



26. ASIM Symposium Simulationstechnik

25.7.-27.7.2022, TU Wien

Herausgegeben von
Felix Breitenecker, Christina
Deatcu, Umut Durak, Andreas
Körner
und Thorsten Pawletta

ASIM SST 2022



Proceedings Langbeiträge





ASIM



ASIM



ASIM



ASIM Books – ASIM Book Series – ASIM Buchreihen

Proceedings

- Proceedings Langbeiträge ASIM SST 2022 -26. ASIM Symposium Simulationstechnik**, TU Wien, Juli 2022
F. Breitenecker, C. Deatcu, U. Durak, A. Körner, T. Pawletta (Hrsg.), ARGESIM Report 20; ASIM Mitteilung AM 181
ISBN ebook 978-3-901608-97-1, DOI 10.11128/arep.20, ARGESIM Verlag Wien, 2022; ISBN print 978-3-903311-19-0, TU Verlag
- Proceedings Kurzbeiträge ASIM SST 2022 -26. ASIM Symposium Simulationstechnik**, TU Wien, Juli 2022
F. Breitenecker, C. Deatcu, U. Durak, A. Körner, T. Pawletta (Hrsg.), ARGESIM Report 19; ASIM Mitteilung AM 179
ISBN ebook 978-3-901608-96-4, DOI 10.11128/arep.19, ISBN print 978-3-901608-73-5, ARGESIM Verlag Wien, 2022
- Simulation in Production and Logistics 2021 – 19. ASIM Fachtagung Simulation in Produktion und Logistik**
Online Tagung, Sept. 2021, J. Franke, P. Schuderer (Hrsg.), Cuvillier Verlag, Göttingen, 2021,
ISBN print 978-3-73697-479-1; ISBN ebook 978-3-73696-479-2; ASIM Mitteilung AM177
- Proceedings ASIM SST 2020 – 25. ASIM Symposium Simulationstechnik**, Online-Tagung
14.-15.10.2020; C. Deatcu, D. Lückerrath, O. Ullrich, U. Durak (Hrsg.), ARGESIM Verlag Wien, 2020;
ISBN ebook: 978-3-901608-93-3; DOI 10.11128/arep.59; ARGESIM Report 59; ASIM Mitteilung AM 174
- Simulation in Production and Logistics 2019 – 18. ASIM Fachtagung Simulation in Produktion und Logistik**
Chemnitz, 18.-20. 9. 2019; M. Putz, A. Schlegel (Hrsg.), Verlag Wissenschaftliche Skripten Auerbach, 2019,
ISBN print 978-3-95735-113-5, ISBN ebook 978-3-95735-114-2; ASIM Mitteilung AM172
- Tagungsband ASIM SST 2018 - 24. ASIM Symposium Simulationstechnik**, HCU Hamburg, Oktober 2018
C. Deatcu, T. Schramm, K. Zobel (Hrsg.), ARGESIM Verlag Wien, 2018; ISBN print: 978-3-901608-12-4;
ISBN ebook: 978-3-901608-17-9; DOI 10.11128/arep.56; ARGESIM Report 56; ASIM Mitteilung AM 168

Book Series Fortschrittsberichte Simulation – Advances in Simulation

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

ASIM SST 2022

Proceedings Langbeiträge

26. ASIM Symposium Simulationstechnik
25.7.-27.7.2022, TU Wien

Herausgegeben von
Felix Breitenecker, Christina Deatcu, Umut Durak,
Andreas Körner und Thorsten Pawletta

ISBN print
ISBN 978-3-903311-19-0
TU Verlag, Wien, 2022
Print on Demand
www.tuverlag.at

ARGESIM Report 20

ISBN ebook
ISBN 978-3-901608-97-1
ARGESIM Verlag, Wien, 2022
www.argesim.org

ASIM Mitteilung 180

Bibliographic Data:

Publisher: ARGESIM Publisher, Vienna

Title: Proceedings Langbeiträge ASIM SST 2022

Subtitle: 26. ASIM Symposium Simulationstechnik, 25.7.-27.7.2022, TU Wien

Author(s):

Editor(s): Felix Breitenecker, Christina Deatcu, Umut Durak, Andreas Körner, Thorsten Pawletta

Series: ARGESIM Reports

Series Editors: Felix Breitenecker, Thorsten Pawletta, ASIM

Volume: ARGESIM Report no. 20

ISBN print: 978-3-903311-19-0, TU Verlag, Print on Demand

ISBN ebook: 978-3-901608-97-1 ARGESIM Verlag

DOI: 10.11128/arep.20

ASIM ID: ASIM Mitteilung 180

Publication Date: July 20, 2022

Number of Pages: 268 +viii pages

Cover: 2 reproductions of algorithmic art graphic series 'Matamorphosis' by Vlatko Ceric, www.vceric.net,
with permission of the artist

Copyright © 2022 ARGESIM Publisher

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, MATHMOD, and EU-ROSIM (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 EU-ROSIM, 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
- Administration and support of ASIM (German Simulation Society → www.asim-gi.org) and of EUROSIM (Federation of European Simulation Societies → www.eurosim.info)

ARGESIM – Arbeitsgemeinschaft Simulation News – Working Committee Simulation News – SNE Publication
Mommengasse 19/8, 1040 Vienna, Austria; Tel +43-1-58801-10111, -10115; Fax +43-1-58801-910111
Email: office@argesim.org, office@sne-journal.org; WWW: www.argesim.org, www.sne-journal.org
Incorporated Austrian Society ZVR No 213056164 – EU VAT ID No ATU 72054279
Bank Account: ARGESIM, IBAN AT07 2011 1828 9115 0800, BIC GIBAATWWXXX, ERSTE BANK VIENNA

VORWORT

ASIM lädt zu einer Tagung ein, wie es früher einmal war – lautet das Motto der Veranstalter von **ASIM SST 2022**, dem **26. ASIM Symposium Simulationstechnik**. Nach einigen virtuellen und hybriden Tagungen war es der erklärte Wunsch, wieder einmal eine ASIM-Präsenztagung zu veranstalten.

Traditionell findet das **ASIM Symposium Simulationstechnik** alle zwei Jahre im Herbst statt. Das Jubiläumssymposium, das **25. ASIM Symposium Simulationstechnik** konnte im Herbst 2020 nur virtuell veranstaltet werden. Planungen für das 26. Symposium sahen einerseits den Herbst 2022 vor, andererseits entstand die Idee, den Frühjahrs-Workshop der ASIM-Fachgruppen GMMS und STS gemeinsam mit dem Symposium Simulationstechnik zu veranstalten, und darüber hinaus auch noch mit der Tagung **MATHMOD Vienna** zu verbinden, deren Co-Sponsor ASIM ist. Die **10th MATHMOD Vienna** musste Corona-bedingt bereits von Februar 2021 auf Februar 2022 verschoben werden – und daher entschieden sich die Organisatoren für **ASIM SST 2022** im Februar 2022, direkt vor der **MATHMOD 2022**.

Wieder machte Corona einen Strich durch die Planung, ein weiteres Verschieben auf den Herbst 2022 erwies sich als ungünstig, da der Herbst 2022 mit Tagungen vollgepfastert ist (sehr viele Corona-bedingte Verschiebungen) – eine gemeinsame Planung mit den **MATHMOD** Organisatoren (teilweise in Personalunion mit den **ASIM SST 2022** Organisatoren) führte daher zu **ASIM SST 2022** und **MATHMOD Vienna 2022** im Juli 2022 - **Tagungen, wie es früher einmal war - conference as it used to be**.

ASIM SST 2022 bietet zwei Arten von Beiträgen, die alle einem Peer Review Prozess unterzogen wurden und sich in der Publikationsform unterscheiden:

- **ASIM SST 2022 Langbeiträge** (6-10 Seiten, deutsch oder englisch) mit individueller DOI für jeden Beitrag im Print- und e-Proceedings Band:
Proceedings Langbeiträge ASIM SST 2022, ARGESIM Report 20, ASIM Mitteilung 180, ISBN ebook 978-3-901608-97-1 ARGESIM Verlag,
ISBN print 978-3-903311-19-0, TU Verlag, Print on Demand
- **ASIM SST 2022 Kurzbeiträge** (2-5 Seiten, deutsch oder englisch), mit Tagungsband-DOI:
Proceedings Kurzbeiträge ASIM SST 2022, ARGESIM Report 19, ASIM Mitteilung 179, ISBN ebook 978-3-901608-96-4 ARGESIM Verlag,
ISBN print 978-3-901608-73-5, ARGESIM Verlag

Wie bei allen ASIM SST und MATHMOD Tagungen bietet **ASIM SST 2022** auch (gemeinsame) Tutorials an. Zudem haben die Organisatoren eine Einrichtung früherer ASIM-Tagungen wiedereingeführt, das **Forum Simulationstechnik**: kontroverse Impulsvorträge zu Entwicklungen in der Simulationstechnik von Simulationsexperten (aus ASIM und Gäste; s. Seite iv) laden zu einer regen Diskussion ein, verbunden mit einer „Strudelsitzung“. Und natürlich freut es uns, dass wir zwei ASIM-Mitglieder für Hauptvorträge gewinnen konnten, Dirk Zimmer (DLR Oberpfaffenhofen; Modelica Mitentwickler), und Niki Popper (TU Wien & dwh Simulation Services; Mitglied des Corona-Beratungsgremiums der österreichischen Bundesregierung).

Zum **ASIM SST 2022** haben sich etwa 80 Fachleute zusammengefunden, zwei Drittel davon ASIM-Mitglieder. Dieser Tagungsband, die *Proceedings Langbeiträge ASIM SST 2022* publiziert die 33 angenommenen Langbeiträge – mit individueller DOI, gruppiert nach thematischen Sessions. Wie bei jeder ASIM-Tagung, ist die Möglichkeit für Postconference Publications in EUROSIM's bzw. ASIM's Scientific Journal *Simulation Notes Europe SNE* vorgesehen.

ASIM lässt bei Tagungen und Workshops den Autorinnen und Autoren bewusst die Wahl der Sprache frei: Englisch für weitere Publikationsverbreitung, oder Deutsch für weitere Kommunikation innerhalb der ASIM-Fachgruppen. Daher sind in den Proceedings beide Sprachen bunt gemischt vertreten.

Virtuelle und hybride Tagungen haben zu einer Verarmung der gesellschaftlichen Aspekte von Tagungen geführt. Der Wunsch nach einer Präsenztagung für **ASIM SST 2022** entstand daher auch aus dem Bedürfnis, diese „Verarmung“ zu beenden, und bei Tagungen wieder wissenschaftlich-gesellschaftliche Kontakte pflegen zu können. Daher bietet **ASIM SST 2022** auch wieder Gesellschaftliches: *Abendveranstaltungen*, das *Cafe Simulation*, *Kunst*, und traditionellerweise *besondere Tagungstaschen* und den *ASIM Tagungswein*, sowie das *Forum Simulationstechnik mit Begleitung*:

- **Tagungsabend Café Resselpark:** ASIM lädt dazu in ein geschichtsträchtiges Lokal am Karlsplatz ein (Teilnehmer können Harry Lime / Orson Welles treffen), zu Wiener Schmankeleien ein, und zur Verkostung der Tagungsweine.
- **Farewell Party ASIM – Welcome Party MATHMOD:** **ASIM SST 2022** Teilnehmer und **MATHMOD 2022** Teilnehmer haben die Gelegenheit, im *Foyer Gusshaus* der TU Wien bei Snacks und Getränken Erfahrungen auszutauschen
- **Cafe Simulation – Ausstellung Algorithmic Art:** Nahe dem Tagungsbüro bietet das *Cafe Simulation* ein Kommunikationszentrum an, mit typischem Wiener Kaffee von Mokka bis Melange (neben den bewusst langen Kaffeepausen), mit Kostproben des Tagungsweins, und mit Algorithmic Art – Grafiken von Vlatko Čerić (die auch auf den **ASIM SST 2022** Publikationen, auf den Tagungsbadges, etc. zu finden sind).
- **Tagungstaschen:** ASIM- und MATHMOD-Tagungen in Wien pflegen eine besondere Tradition für Tagungstaschen/Tagungsboxen, sie werden von Behindertenwerkstätten angefertigt. Die Rucksack-ähnlichen Sportbeutel als Tagungstaschen für **ASIM SST 2022** und **MATHMOD 2022** haben betreute Personen der Behindertenwerkstatt Ternitz mit den Tagungsnamen bestickt.
- **Tagungswein:** Bereits bei frühen ASIM-Tagungen in Wien (z.B. 1984), beim EUROSIM Congress 1995 in Wien, und bei vielen MATHMOD-Tagungen hat der Tagungswein Gefallen gefunden – bei Kostproben und als Mitgabe für jeden Tagungsteilnehmer. An diese Tradition schließt **ASIM SST 2022** an, mit Wein vom Weingut Zadravil aus Retz (NÖ), von Weinsorten *Grüner Veltliner* bis *Gemischter Satz*, und mit der Weinsorten-Besonderheit *Gemischte Katz*.
- **Strudelsitzung:** Warum nicht Wissenschaft mit Kaffee und Kuchen verbinden – *Das Forum Simulationstechnik*, der letzte Programmpunkt von **ASIM SST 2022**, hat als Begleitung eine „Strudelsitzung“: Diskutanten können sich an ‚Ausgezogenen Strudeln‘ und Kaffee und Getränken stärken.

Wir Organisatoren möchten allen herzlich danken, die das **ASIM SST 2022** zu einem Erfolg machen - *zu einer Tagung ein, wie es früher einmal war*: den Teilnehmerinnen und Teilnehmern, den Autoren der Beiträge, dem Programmkomitee für das Reviewing, dem ARGE-SIM Verlag für die Publikationen, und unseren Tagungsmitarbeitern.

*Felix Breitenacker, Christina Deatcu, Umut Durak,
Andreas Körner und Thorsten Pawletta, Juli 2022*

ASIM SST 2022

26. ASIM Symposium Simulationstechnik, 25.7.-27.7.2022, TU Wien

Sponsor und Organisation - Sponsored and Organized by

TU Wien, Inst. f. Analysis and Scientific Computing

Co-Sponsoren - Co-Sponsors

- **ASIM** – German Simulation Society (Arbeitsgemeinschaft Simulation) → www.asim-gi.org
- **ARGESIM** Vienna – Arbeitsgemeinschaft Simulation News, Wien → www.argesim.org
- **EUROSIM** – Federation of European Simulation Societies → www.eurosim.info

Programmkomitee – Programme Committee

Umut Durak, DLR (Chair)

Felix Breitenecker, TU Wien, (Co-Chair)

Christina Deatcu, HS Wismar, (Co-Chair)

Kurt Chudej, Universität Bayreuth

Walter Commerell, TH Ulm

Umut Durak, DLR

Leo Gall, LTX Simulation GmbH, München

Lukas Hollenstein, HS Zürich

Andreas Körner, TU Wien

Christoph Laroque, Hochschule Zwickau

Xiabo Liu-Henke, Hochschule Ostfalia

Daniel Lückerath, Fraunhofer IAIS

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

Thorsten Pawletta, Hochschule Wismar

Nicolas Popper, dwh GmbH, Wien

Oliver Rose, Univ. der Bundeswehr München

Thomas Schramm, HCU Hamburg

Michael Striebel, HTWG Konstanz

Oliver Ullrich, Fraunhofer IAIS

Sigrid Wenzel, Universität Kassel

Jochen Wittmann, HTW Berlin

Organisationskomitee – Organizing Committee

Andreas Körner (Chair)

Felix Breitenecker (Co-Chair)

Brigitte Ecker (Conf. Ass.)

ASIM 2022 Hauptvorträge - ASIM 2022 Invited Lectures

- Robuste Modellierung und informationsverlustbehaftete Idealisierung in objekt-orientierter Modellierung physikalischer Systeme
Dirk Zimmer, DLR
- Agent-based COVID-19 Simulation: General Analysis and Guideline for Politicians !?
Niki Popper, TU Wien / DWH Simulation Services

ASIM 2022 Forum Simulationstechnik - ASIM 2022 Forum Simulation Technique

Die ASIM 2022, das 26. ASIM Symposium Simulationstechnik, setzt mit dem **Forum Simulationstechnik** eine Tradition früherer ASIM-Tagungen fort. Das Forum bringt (kontroverse) Impulsvorträge zu Entwicklungen in der Simulationstechnik von Simulationsexperten (aus ASIM und Gäste); Teilnehmer sind zu eigenen Impulsvorträgen und/oder zur Diskussion der Impulsvorträge eingeladen:

- Lieber KI mit vielen simulierten Daten als KI mit wenig realen Daten !
O. Rose
- System Dynamics - Modellbildung nur für Dummies ?
P. Junglas
- Co-Simulation - Methode zur Unterstützung der Faulheit in der Modellierung ?
F. Breitenecker, I. Hafner, N. Popper
- Reinforcement Learning - Wettstreit Model-based vs Model-free ?
D. Pasterk, M. Wastian
- Daten für Corona-Simulationen - von exzellent über heimtückisch bis unbrauchbar
N. Popper
- ARGESIM Benchmarks - Out of Date oder Tool für die Simulationslehre ?
F. Breitenecker, P. Junglas, C. Deatcu
- und weitere

ASIM 2022 Thematische Sessions - ASIM 2022 Thematic Sessions

- Simulation in Produktion und Logistik - Simulation in Production and Logistics
- Diskrete Systeme - Discrete Systems
- Simulation Technischer Systeme - Simulation of Technical Systems
- Simulation Biologie, Medizin und Umwelt - Simulation in Biology, Medicine, and Environment
- Simulationsmethoden - Methods in Simulation

Inhaltsverzeichnis – List of Content

HAUPTVORTRÄGE - INVITED LECTURES	1
Agent-based COVID-19 Simulation: General Analysis and Guideline for Politicians !? <i>Niki Popper</i>	3
Robuste Modellierung und informationsverlustbehaftete Idealisierung in objekt-orientierter Modellierung physikalischer Systeme. <i>Dirk Zimmer</i>	5
SIMULATION IN PRODUKTION UND LOGISTIK – SIMULATION IN PRODUCTION AND LOGISTICS	7
Literaturbasierte Untersuchung der Prozessmerkmale datengetriebener Prozesse im Anwendungsfeld der Produktion <i>S. P. Christiansen-Lenger, M. Rabe; DOI: 10.11128/arep.20.a2025</i>	9
Optimierung des Designs modularer Montagesysteme <i>S. Bergmann; DOI: 10.11128/arep.20.a2006</i>	15
Untersuchung von Auftragsfreigabestrategien für mehrzonige Kommissioniersysteme im Online-Lebensmittelhandel. <i>S. Galka; DOI: 10.11128/arep.20.a2021</i>	23
Selbstoptimiertes Auftragsmanagement für autonome Transportfahrzeuge <i>J. Zhang, X. Liu-Henke, T. Vietor; DOI: 10.11128/arep.20.a2031</i>	31
Verknüpfung von ereignisdiskreter Simulation und Process Mining in Produktion und Logistik <i>F. Özkul, R. Sutherland, S. Wenzel, U. Jessen, S. Spieckermann DOI: 10.11128/arep.20.a2035</i>	39
Experimentelle Analyse zur Einschleusplanung in der Halbleiterfertigung mittels einer stochastischen Rückwärtssimulation. <i>C. Laroque, M. Leißau, W. Scholl, G. Schneider, P. Preuß DOI: 10.11128/arep.20.a2016</i>	49
Discrete event-based modeling of conveyors for dry bulk material <i>P. Junglas, L. Schmedes; DOI: 10.11128/arep.20.a2003</i>	57
Simulationsbasiertes Deep Reinforcement Learning für Modulare Produktionssysteme <i>N. Feldkamp, S. Bergmann, S. Straßburger; DOI: 10.11128/arep.20.a2007</i>	65
DISPO 40 - Simulationsgestützte Absatzprognoseoptimierung in der Investitionsgüter-industrie. <i>A. Schmid, F. Kamhuber, T. Sobottka, W. Sihn; DOI: 10.11128/arep.20.a2017</i>	73
Towards Increasing Resilience of Public Transit Infrastructure – A Bi-Modal Simulation Model <i>D. Lücknerath, O. Ullrich; DOI: 10.11128/arep.20.a2015</i>	81
DISKRETE SYSTEME – DISCRETE SYSTEMS	91
A Simulator for NSA-DEVS in Matlab <i>D. Jammer, P. Junglas, T. Pawletta, S. Pawletta; DOI: 10.11128/arep.20.a2005</i>	93
Adapting to Change of Model Transitions in Proxel Based Simulation of CHnMMs <i>D. Bodnár, C. Krull; DOI: 10.11128/arep.20.a2019</i>	101
A Hybrid User Model for Virtual Stochastic Sensors <i>C. Krull; DOI: 10.11128/arep.20.a2030</i>	109

SIMULATION TECHNISCHER SYSTEME – SIMULATION OF TECHNICAL SYSTEM	117
Manöverbasierte Validierung einer Längsdynamiksimulation anhand generisch erzeugter Reallastverläufe	119
<i>S. Cortès, C. Dettmann, L. Heinzelmann, O. Haidar</i> ; DOI: 10.11128/arep.20.a2001	
Robustheitsanalyse einer Längsdynamiksimulation auf Basis verschiedener Qualitätskriterien	129
<i>S. Cortès, C. Dettmann, J. Heib, F. Millberg</i> ; DOI: 10.11128/arep.20.a2002	
Modellierung des Betriebsverhaltens eines Synchrongenerators als Stromerzeuger	139
<i>B. H. Herrmann, M. Ringel, D. Jörss, C. Fink</i> ; DOI: 10.11128/arep.20.a2009	
Systematischer modellbasierter Entwurf einer Reinforcement Learning-basierten neuronalen adaptiven Geschwindigkeitsregelung	147
<i>X. Liu-Henke, S. Jacobitz, Or Aviv Yarom, J. Fritz</i> ; DOI: 10.11128/arep.20.a2011	
Model-based Development of an Automated and Remotely-Controlled Demolition Excavator	155
<i>C. Richter, M. Shakoorianfard, V. Waurich, F. Will</i> ; DOI: 10.11128/arep.20.a2008	
Efficient state estimation for gas pipeline networks via low-rank approximations	163
<i>N. Stahl, N. Marheineke</i> ; DOI: 10.11128/arep.20.a2029	
Robotics simulation – A comparison of two state-of-the-art solutions	171
<i>M. Zwingel, C. May, M. Kalenberg, J. Franke</i> ; DOI: 10.11128/arep.20.a2033	
Parameter tables for PID controllers for time delayed systems optimized with a learning method.	179
<i>R. Büchi</i> ; DOI: 10.11128/arep.20.a2013	
Interacting-Multiple-Model Partikelfilter zur Fehleridentifikation.	187
<i>A. Lepp, D. Weidemann</i> ; DOI: 10.11128/arep.20.a2024	
SIMULATION IN BIOLOGIE, MEDIZIN UND UMWELT – SIMULATION IN BIOLOGY, MEDICINE, AND ENVIRONMENT	195
Ein Fitnesscoach für Menschen mit Querschnittslähmung	197
<i>K. Nickel, J. Bolz</i> ; DOI: 10.11128/arep.20.a2026	
SIMULATIONSMETHODEN - METHODS IN SIMULATION	203
A time domain approach for system identification using hill climbing	205
<i>R. Büchi</i> ; DOI: 10.11128/arep.20.a2014	
Using the Proxel Method to build EHMM for behaviour reconstruction	211
<i>P. Krenckel, C. Krull</i> ; DOI: 10.11128/arep.20.a2037	
An Extension for the Specification and Automated Selection of System Variants Based on the System Entity Structure Using a Problem from Process Industry	219
<i>H. Folkerts, T. Pawletta, U. Durak</i> ; DOI: 10.11128/arep.20.a2018	
Analysis of Selected Literature on Co-simulation	227
<i>I. Hafner, S. Emrich, N. Popper</i> ; DOI: 10.11128/arep.20.a2010	
Coupling of Simulation Tools for Obtaining Local Fatigue in Combination with Experimental Data.	235
<i>N. Adamchuk, B. Rösch, W. Schlüter, F.-C. Reissner, J. Baumgartner</i> DOI: 10.11128/arep.20.a2023	
Hochaufgelöste Energieprofile durch hybride Simulation	243
<i>B. Woerrlein, S. Strassburger</i> ; DOI: 10.11128/arep.20.a2004	
Modular platform for route guidance in the cyber-physical laboratory test field	253
<i>M. Göllner, S. Jacobitz, T. Li, X. Liu-Henke</i> ; DOI: 10.11128/arep.20.a2032	
Service-oriented Architecture for Automatic Evaluation of Urban Logistical Concepts	263
<i>R. Pump, L. Bienzeisler, T. Lelke, V. Ahlers, A. Koschel</i> ; DOI: 10.11128/arep.20.a2036	

Autorenindex – Index of Authors

N. Adamchuk 235	B. H. Herrmann 139	M. Ringel 139
V. Ahlers 263	H. Hillenhagen (K: 53)	B. Rösch 235
M. Bachler (K: 63)	S. Jacobitz 147, 253	M. Rößler (K:33, 47)
V. Barzal (K:33)	D. Jammer 93	U. Rüde (K: 73)
J. Baumgartner 235	U. Jessen 39	A. A. Scheidler (K:81)
L. Berger (K: 77)	D. Jörss 139	W. Schlüter 235, (K: 21, 43)
S. Bergmann 15, 65	P. Junglas 57, 93	L. Schmedes 57
J. Bernhardt (K: 53)	M. Kalenberg 171	A. Schmid 73
L. Bienzeisler 263	F. Kamhuber 73	G. Schneider 49
D. Bodnár 101	H. Kastinger (K: 63)	W. Scholl 49
J. Bolz 197	A. Körner (K: 37, 67)	S. Scholz (K: 77)
F. Breiteneker (K:33, 47, 67)	A. Koschel 263	S. Schreiber (K: 57)
A. Buchele (K: 21, 43)	P. Krenckel 211	C. Schwarz (K: 73)
R. Büchi 179, 205	C. Krull 101, 109, 211	M. Shakoorianfard 155
T. Christiansen (K: 57)	M. J. Kühn (K: 73)	W. Sihn 73
S. P. Christiansen-Lenger 9	K. Langenbach (K:81)	T. Sobottka 73
S. Cortès 119, 129	C. Laroque 49	S. Spieckermann 39
F. Cramer (K: 17)	M. Leißau 49	N. Stahl 163
J. Dettelbacher (K: 21)	T. Lelke 263	S. Straßburger 65, 243
C. Dettmann 119, 129	P. Leleux (K: 73)	R. Sutherland 39
U. Durak 219	A. Lepp 187	O. Ullrich 81
M. Ehrmann (K: 47)	X. Liu-Henke 31, 147, 253	T. Vietor 31
M. Engelmann (K: 25)	D. Lückerrath 81	D. Wagner (K: 21)
S. Emrich 227	N. Marheineke 163	C. Wahmkow (K: 53)
N. Feldkamp 65	L. März (K: 9,13)	M. Wastian (K: 33)
C. Fikar (K: 17)	C. May 171	V. Waurich 155
C. Fink 139	F. Millberg 129	D. Weidemann 187
H. Folkerts 219	K. Nickel 197	D. Welsch (K: 43)
J. Franke 171	F. Özkul 39	S. Wenzel 39
A. Freißlinger (K: 67)	D. Pasterk (K: 37)	F. Will 155
J. Fritz 147	S. Pawletta 93	J. Wittmann (K: 25, 57)
S. Galka 23	T. Pawletta 93, 219	B. Woerrlein 243
M. Göllner 253	N. Popper 3, 227 (K: 3, 33)	V. Wohlgemuth (K: 25)
A. Grüning (K: 53)	P. Preuß 49	Or Aviv Yarom 147
I. Hafner 227	R. Pump 263	K. Zacharias (K: 43)
O. Haidar 119	M. Rabe 9 (K:81)	J. Zhang 31
J. Heib 129	F.-C. Reissner 235	Dirk Zimmer 5
L. Heinzelmann 119	C. Richter 155	M. Zwingel 171, (K: 5)

Die Kurzbeiträge (K) sind zu finden im **ARGESIM Report 19, Proceedings Kurzbeiträge ASIM SST 2022**

ISBN ebook 978-3-901608-96-4, DOI: 10.11128/arep.19, ASIM Mitteilung AM 179

Hauptvorträge

Invited Lectures

Agent-based COVID-19 Simulation: General Analysis and Guideline for Politicians !?
(Abstract)

Niki Popper, TU Wien / DWH Simulation Services

Robuste Modellierung und informationsverlustbehaftete Idealisierung in
objekt-orientierter Modellierung physikalischer Systeme (Abstract)

Dirk Zimmer, DLR

Agent-based COVID-19 Simulation: General Analysis and Guideline for Politicians !?

Niki Popper

dwh Simulation Services, Neustiftgasse 57-59, 1070 Wien &
TU Wien, Inst.f. Information and Software Engineering, Favoritenstraße 11, 1040 Wien

Abstract. Das Gesundheitssystem ist ein Beispiel für hochkomplizierte dynamische Systeme, die aus Eigendynamik verschiedener Bereiche, den Interessen unterschiedlicher Akteure und Kontrollmechanismen ein kaum nachvollziehbares Amalgam an dynamischen Prozessen und Subsystemen ausbilden. Die COVID-19 Krise und der Umgang damit ist nur ein Beispiel dafür. Modellierung versucht hier ein wenig Licht ins Dunkel zu bringen. Methodische Herausforderungen sind dabei speziell die Kombination komplexer Datenprozesse und unterschiedlicher Modellierungsmethoden. Im Vortrag werden Herausforderungen beleuchtet und Best Practice Beispiele, wie etwa der COVID-19 European Scenario Hub der ECDC präsentiert.

Robuste Modellierung und informationsverlustbehaftete Idealisierung in objekt-orientierter Modellierung physikalischer Systeme

Dirk Zimmer

Deutsches Zentrum für Luft- und Raumfahrt (DLR), Institut für Systemdynamik und Regelungstechnik,
Münchner Straße 20, 82234 Oberpfaffenhofen-Wessling, Deutschland

Abstract. Die objekt-orientierte Modellierung hat zu einer großen Ansammlung von Wissen für cyber-physikalische Systeme geführt. Doch auch nach über 25 Jahren halten sich die Probleme der Skalierung in der täglichen Anwendung hartnäckig. Ohne große Expertise scheitert die Simulation komplexer Systeme immer noch zu häufig. Die Ursachen liegen dabei im Informationsverlust bei der Idealisierung technischer Systeme.

Betrachtet man die Formen der Idealisierung ganz allgemein, fallen gutartige und bösartige Formen auf. Mögen wir lernen, die bösartigen Formen zu erkennen und die gutartigen Formen in methodischer Form zu nutzen. Robuste Modellierungsbibliotheken wie die DLR ThermoFluid Stream Library zeigen die für den Anwender nützlichen Auswirkungen dieser grundlegenden Betrachtungen.

Simulation in Produktion und Logistik

Simulation in Production and Logistics

Optimierung des Designs modularer Montagesysteme (L)

S. Bergmann

Untersuchung von Auftragsfreigabestrategien für mehrzonige Kommissioniersysteme im Online-Lebensmittelhandel (L)

S. Galka

Selbstoptimiertes Auftragsmanagement für autonome Transportfahrzeuge (L)

J. Zhang, X. Liu-Henke, T. Vietor

Operative, simulationsgestützte Leistungssteuerung von Springern in einer getakteten Endmontagelinie (K)

L. März

Simulationsgestütztes Line Balancing (K)

L. März

Verknüpfung von ereignisdiskreter Simulation und Process Mining in Produktion und Logistik (L)

F. Özkul, R. Sutherland, S. Wenzel, U. Jessen, S. Spieckermann

Experimentelle Analyse zur Einschleusplanung in der Halbleiterfertigung mittels einer stochastischen Rückwärtssimulation (L)

C. Laroque, M. Leißau, W. Scholl, G. Schneider, P. Preuß

Discrete event-based modeling of conveyors for dry bulk material (L)

P. Junglas, L. Schmedes

Simulationsbasiertes Deep Reinforcement Learning für Modulare Produktionssysteme (L)

N. Feldkamp, S. Bergmann, S. Straßburger

Simulating the Viability of Crowd Logistics in Short Food Supply Chains (K)

Florian Cramer, Christian Fikar

Kopplung einer Material- und Energieflusssimulation mit Reinforcement-Learning-Algorithmen (K)

J. Dettelbacher, D. Wagner, A. Buchele, W. Schlüter

DISPO 40 - Simulationsgestützte Absatzprognoseoptimierung in der Investitionsgüterindustrie (L)

A. Schmid, F. Kamhuber, T. Sobottka, W. Sihn

Towards Increasing Resilience of Public Transit Infrastructure – A Bi-Modal Simulation Model (L)

D. Lückcrath, Oliver Ullrich

Erstellung eines Verkehrssimulationsmodells für Berlin in SUMO zur Simulation der möglichen Emissionen und daraus resultierendem Routing (K)

M. Engelmann, J. Wittmann, V. Wohlgemuth

Die Kurzbeiträge (K) sind zu finden im **ARGESIM Report 19**,

Proceedings Kurzbeiträge ASIM SST 2022

ISBN ebook 978-3-901608-96-4, DOI: 10.11128/arep.19, ASIM Mitteilung AM 179

Literaturbasierte Untersuchung der Prozessmerkmale datengetriebener Prozesse im Anwendungsfeld der Produktion

Sean Paul Christiansen-Lenger^{1*}, Markus Rabe¹

¹Fachgebiet IT in Produktion und Logistik, Technische Universität Dortmund, Leonhard-Euler-Str. 5, 44227 Dortmund, Germany; *sean.christiansen@tu-dortmund.de

Abstract. Datengetriebene Prozesse sind heutzutage für viele Anwendungen im produktionslogistischen Bereich relevant. Durch die uneinheitliche Verwendung des Begriffs datengetrieben bzw. des englischen Begriffs Data-driven zur Prozessbezeichnung in der Literatur, ist keine eindeutige Klassifizierung für Prozesse als datengetrieben möglich. Um diesbezüglich eine Klassifikation zu ermöglichen, wurden aus Prozessbeschreibungen von datengetriebenen Prozessen aus der Fachliteratur Merkmale zur Klassifikation abgeleitet und als Checkliste dargestellt.

Einleitung

Um die Ansätze des Smart Manufacturing oder Digitaler Zwillinge zu realisieren, müssen große Datenmengen echtzeitnah verarbeitet werden. Datengetriebene Prozesse bilden die Grundlage für Smart Manufacturing [1; 2]. In der Fachliteratur werden im Bereich der Produktion unterschiedliche als datengetrieben klassifizierte Prozesse beschrieben, u. A. im Kontext der Produktionsplanung, der -überwachung und der -optimierung [1] sowie in produktionsnahen Bereichen wie der Simulation [3] und des Supply Chain Management [4; 5].

Ähnlich wie für den Begriff Big Data, für den sich auch bis in das Jahr 2022 noch keine einheitliche Definition durchgesetzt hat, gibt es auch für datengetriebene Prozesse keine allgemeingültige Definition. Synonym werden beispielsweise Begriffe wie *knowledge-based methods* [6] oder *big data driven* [4; 7] verwendet und es finden sich auch Bezüge zu Begriffen wie *twin driven* [8].

Da die in der Literatur uneinheitliche Verwendung der Prozessbezeichnung datengetrieben eine entsprechende Klassifizierung von Prozessen erschwert, werden im Folgenden die gemeinsamen Merkmale der datengetriebenen Prozesse in der Literatur hergeleitet. Diese Merkmale dienen als Grundlage für eine Klassifizierung

von Prozessen als datengetrieben. Um in der Praxis die Klassifizierung zu erleichtern, werden die Merkmale in eine Checkliste überführt, die eine Überprüfung eigener Prozesse bezüglich der Einordnung als datengetrieben ermöglicht. Zur besseren Lesbarkeit wird im Folgenden das deutsche Adjektiv *datengetrieben* für den englischen Begriff *Data-driven* verwendet. Für feststehende Fachbegriffe wird auf eine Übersetzung verzichtet.

1 Merkmale und Begriffsfindung

Das Verständnis für IT-Prozesse mit der Klassifikation datengetrieben hat sich im Laufe der Zeit gewandelt [9] und der Begriff datengetrieben wird beispielsweise auch in Fachdisziplinen des Journalismus [10] und der Linguistik [11] verwendet. Aus diesen Gründen werden zu der Ableitung einer Verständnisgrundlage für den produktionslogistischen Kontext ausschließlich aktuelle Veröffentlichungen aus diesem Fachbereich herangezogen.

Xu et al. beschreiben ein auf datengetriebene Prozesse ausgerichtetes Produktionssystem, an dem sich dieser wissenschaftliche Beitrag orientiert [12]. In dem von Xu et al. dargestellten Produktionssystem werden datengetriebene Prozesse direkt mit dem Fertigungsprozess verknüpft. Alle relevanten Prozessparameter werden beispielsweise durch Sensoren im Prozess oder Produktionssystem aufgenommen. Die aufgenommenen Messwerte werden in Echtzeit analysiert und interpretiert. Auf Basis dieser Interpretationen werden Entscheidungen getroffen und direkt in der realen Fertigung umgesetzt. Diese Form der Organisation eines Produktionssystems wird als datengetriebene Fertigung bezeichnet und kann durch die direkte Rückführung der Erkenntnisse des datengetriebenen Prozesses in den physikalischen Prozess

als Kreislauf dargestellt werden [12].

Dieser Ablauf mit Bezug auf ein umgebenes Framework wird in [12] detailliert beschrieben, ein weiteres Framework mit umfangreicherem Bezug zu dem Produktdaten-Lebenszyklus in [1]. In **Abbildung 1** wird in einer Weiterentwicklung einer Darstellung aus [12] der Ablauf eines datengetriebenen Prozesses sowie dessen Einbindung in der datengetriebenen Fertigung dargestellt.

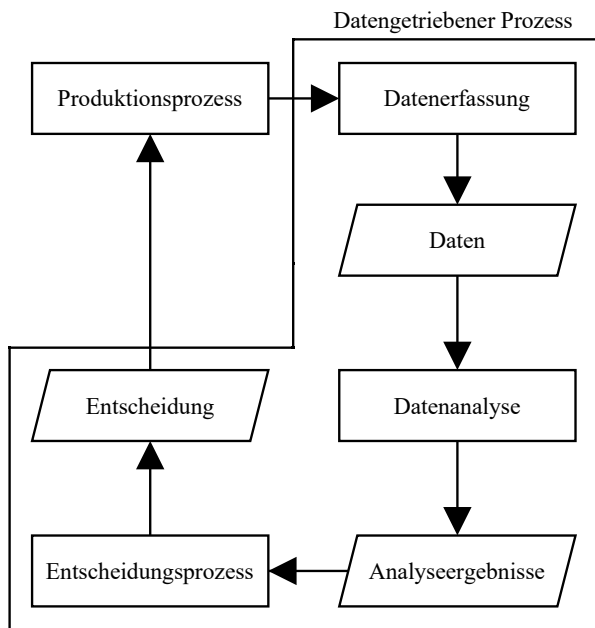


Abbildung 1: Ablauf datengetriebener Prozesse in der datengetriebenen Produktion

Basierend auf dem zuvor beschriebenen Ablauf werden im Folgenden die wichtigsten Merkmale diskutiert, die datengetriebenen Prozessen in der Literatur zugeordnet werden. Merkmale werden im Sinne der DIN 55350 verstanden, die ein Merkmal als „kennzeichnende Eigenschaft“ definiert, welche sowohl qualitativ als auch quantitativ sein kann [13].

Um mögliche Merkmale aktueller datengetriebener Prozesse darzulegen, wurden basierend auf der Anzahl der Zitationen und dem Journal Index Factor die relevantesten Veröffentlichungen mit Bezug auf Produktion und Logistik aus den Jahren 2015 bis 2021 analysiert und die gemeinsamen Merkmale als definierend eingestuft.

Datenzentrierte Prozessbetrachtung

Beispielsweise in [14] wird explizit darauf hingewiesen, dass es zur Gewinnmaximierung für moderne produzierende Unternehmen ein Schlüsselement ist, aus Daten Erkenntnisse zu gewinnen, um Entscheidungsprozesse

effektiv zu unterstützen [1; 2; 14]. In den aus dieser Notwendigkeit resultierenden datengetriebenen Prozessen [14] nehmen Daten in der Prozessbetrachtung eine zentrale Rolle ein, beispielsweise in [1; 6; 12; 14–16].

Datenerfassung

In der Mensch-Maschine-Material-Umgebung der Fertigung werden Daten aus Fertigungs- [1; 6; 14; 17] und Logistikprozessen [1; 6; 17; 18] durch Sensoren sowie Internet-der-Dinge-Technologien wie RFID, AutoID etc. [18–20] und von Werkern [1] direkt aufgenommen sowie indirekt aus Informationssystemen [1] oder aus historischen Daten [6; 14] übernommen. Historische Daten können dabei sowohl zum Training von Machine-Learning-Anwendungen [1; 6; 14] als auch zur teilweise automatischen Bewertung des Status oder zur Vorhersage zukünftiger Störungen in dem überwachten Prozess oder der Maschine genutzt werden [1]. Besondere Bedeutung wird in vielen Quellen auch der Echtzeitfähigkeit dieser Prozesse beigemessen, z. B. in [1; 16–18].

Zielgerichtete Wissensgenerierung

Die zentrale Aufgabe von datengetriebenen Prozessen ist, Wissen aus Daten zu generieren [1; 4; 6; 12; 14; 15; 21]. Wissensentdeckung kann in zwei Arten eingeteilt werden. Sie kann deskriptiv sein, also der Zusammenfassung oder dem ungerichteten Entdecken von Wissen dienen oder prädiktiv sein, um zielgerichtet Wissen zu entdecken, um im Idealfall direkt Strategien zu entwickeln, die als Entscheidungsgrundlagen dienen können [14].

Anders als bei Knowledge-Discovery-Prozessen ist bei vielen datengetriebenen Prozessen das Ziel, eine vor Anwendung des Prozesses definierte Art von Wissen zu extrahieren, die das Lösen einer vorher definierten Aufgabe ermöglichen soll [1; 6; 12; 15; 16]. Datengetriebene Prozesse lassen sich demnach als hypothesengetrieben bezeichnen und können rein deskriptiv oder rein prädiktiv sowie eine Kombination aus beidem sein.

Checkliste

Die abgeleiteten Prozessmerkmale werden in **Tabelle 1** zusammengefasst. Diese ist in Form einer Checkliste gehalten, die eine Einordnung von Prozessen als datengetrieben ermöglichen soll. Dafür muss der Prozess jedes der drei Kriterien erfüllen. Für das Kriterium 2 müssen dafür beide Unterkriterien erfüllt sein, für das Kriterium 3 genügt die Erfüllung eines Unterkriteriums. Zur Erfüllung eines Unterkriteriums reicht jeweils aus, wenn eine der

beschriebenen Merkmalsausprägungen erfüllt wird.

Tabelle 1: Checkliste zur Einordnung von IT-Prozessen als datengetrieben

Kriterium	Quellen
1. Prozessbetrachtung	
• Fokussierung auf Daten zur Entscheidungsunterstützung	[1; 6; 12; 14–16]
2. Wissensgenerierung	
• Hypothesengetrieben	[1; 4; 6; 12; 14; 15; 21]
• Form der Wissensextraktion	
○ Deskriptiv	[1; 4; 6; 12; 14; 15; 21]
○ Prädiktiv	[1; 4; 6; 12; 14; 15; 21]
○ Mischform deskriptiv-prädiktiv	[1; 4; 6; 12; 14; 15; 18; 21]
3. Datenerfassung	
• Direkte Datenerfassung	[1; 6; 14; 16; 17]
• Indirekte Datenerfassung	
○ Erfassung von korrelierten Daten	[12]
○ Übernahme aus IT-Systemen, z. B.	
▪ Prozessdaten	[1]
▪ Historische Daten	[1; 6; 14; 16–18]

2 Datengetriebene Prozesse im Kontext der Simulation

Durch die angestrebte Anwendung im Digitalen Schatten [17] und Digitalen Zwilling [18; 22] werden datengetriebene Prozesse auch für die Simulation immer relevanter. In der Vergangenheit konnten Daten in Produktionssystemen kaum in Echtzeit gesammelt oder verarbeitet werden [3]. Deshalb wurden historische Daten oder Expertenwissen zur Erstellung von Modellen verwendet [3; 12; 23]. Diese Modelle kommen überwiegend zur Entscheidungsunterstützung im langfristigen Planungshorizont

zum Einsatz [3; 23]. Dahingegen werden Simulationsmodelle in digitalen Zwillingen, die reale Systeme simulieren, mit Echtzeitdaten synchronisiert und können somit auch im operativen Geschäft Grundlagen für Entscheidungen bilden [3].

Durch die Verbindung eines Modells mit dem entsprechenden physikalischen Prozess durch einen datengetriebenen Prozess sollen sogenannte datengetriebene Modelle im dynamischen Fertigungsumfeld Entscheidungen treffen [21] oder den Entscheidungsträgern Alternativen empfehlen können. Modelle sind dadurch nicht länger statisch, sondern werden durch Echtzeitdaten aktualisiert und teilweise angepasst [3].

Um die Einsatzmöglichkeiten der Checkliste zur Einordnung von Prozessen als datengetrieben anhand der jeweiligen Prozessbeschreibung zu demonstrieren, wird diese in einem Use Case angewandt. Als Use Case wurde die von Qiao et al. [24] beschriebene Anwendung eines datengetriebenen Prozesses mit einem numerischen Modell zur Vorhersage von Werkzeugverschleiß ausgewählt, da diese Veröffentlichung die aktuell relevante Verknüpfung von datengetriebenen Prozessen mit der Simulation und dem Digitalen Zwilling beschreibt.

In diesem Use Case werden in einem Fräsprozess die Prozesskräfte, Vibrationen und Lautstärkeemissionen mit Sensoren in Echtzeit gemessen. Diese Daten werden in einem Modell zur Vorhersage von Werkzeugverschleiß analysiert. Das Modell berechnet durch die Verwendung eines Rekurrenten Neuronalen Netzes (RNN) den Verschleiß des Werkzeugs basierend auf den oben genannten Prozessdaten. Das RNN wurde mit gespeicherten Prozessdaten und korrespondierenden Verschleißdaten der Freiflächen des Fräskopfs trainiert [24].

Basierend auf der Prozessbeschreibung von Qiao et al. wurde die in der vorliegenden Veröffentlichung erstellte Checkliste bezüglich der Klassifizierung des Prozesses aus [24] wie in **Tabelle 2** dargestellt ausgefüllt. Die Prozessbetrachtung ist durch das Training des RNN durch Prozessdaten und die Verwendung von Echtzeitprozessdaten zur Vorhersage von Verschleiß primär auf Daten ausgerichtet (Kriterium 1). Die Wissensgenerierung ist hypothesengetrieben, da davon ausgegangen wird, dass der Verschleiß durch die aufgenommenen Prozessdaten vorhersagbar ist und stellt eine Mischung aus deskriptiver und prädiktiver Wissensgenerierung dar (Kriterium 2). Die Datenerfassung findet überwiegend direkt im Prozess durch Sensoren statt. Da die Verschleißanalyse des Werkzeugs, die getrennt vom Prozess und der direkten Datenaufnahme stattfindet, lediglich

zum Training des RNN verwendet wird, wird auch Kriterium 3 als erfüllt angesehen.

Tabelle 2: Basierend auf der Prozessbeschreibung aus [24] ausgefüllte Checkliste zur Einordnung von IT-Prozessen als datengetrieben

Kriterium	Erfüllung?
	✓/x
1. Prozessbetrachtung	
• Fokussierung auf Daten zur Entscheidungsunterstützung	✓
2. Wissensgenerierung	
• Hypothesengetrieben	✓
• Form der Wissensextraktion	
○ Deskriptiv	
○ Prädiktiv	
○ Mischform deskriptiv-prädiktiv	✓
3. Datenerfassung	
• Direkte Datenerfassung	✓
• Indirekte Datenerfassung	
○ Erfassung von korrelierten Daten	
○ Übernahme aus IT-Systemen, z. B.	
▪ Prozessdaten	
▪ Historische Daten	✓

3 Diskussion und Ausblick

Um Smart Manufacturing zu realisieren, reicht das Internet der Dinge nicht aus, da dieses lediglich Transparenz der verfügbaren Informationen bietet [25]. Dadurch werden datengetriebene Prozesse relevant, da diese basierend auf Wissensgenerierung aus den immer größer werdenden Datenmengen Entscheidungsgrundlagen bilden können oder Entscheidungen autonom treffen können. Da der Begriff datengetriebener Prozess in der Literatur uneinheitlich verwendet wird, wurden zur Bildung einer Klassifikationsgrundlage die relevantesten Veröffentlichungen aus dem produktionslogistischen Kontext von 2015 bis 2021 herangezogen und gemeinsame Merkmale der dargestellten Prozesse abgeleitet. Diese Merkmale wurden zur Klassifikation von Prozessen in Form einer Checkliste dargestellt. Die Anwendung dieser Checkliste wurde im weiteren Verlauf gezeigt.

In der Merkmalsableitung wird besonders die Abgrenzung zu Big Data Analytics deutlich. Während im Kontext von Big Data große vorhandene Datenmengen mit dem Ziel einer deskriptiven Wissensentdeckung analysiert werden, wird bei der Auslegung von datengetriebenen Prozessen bereits ein konkretes Ziel verfolgt. Zur Lösung einer oder mehrerer klar definierter Aufgaben werden die zu analysierenden Daten gezielt ausgewählt oder, wie in [12] ausführlich dargestellt, direkt oder indirekt aufgenommen. Dabei bedienen sich datengetriebene Prozesse unter anderem Anwendungen aus dem Big-Data-Kontext, wie z. B. Data Analytics [4] oder Data Mining [21]. Der Unterschied zwischen den Begriffen zeigt sich auch bei der Datenaufnahme und der Bedeutung einzelner Messdaten, welchen im Kontext von Big Data wesentlich weniger Bedeutung beigemessen wird, da diese, im Gegensatz zu der Verwendung in datengetriebenen Prozessen, einzeln trivial und ungenau sind [12].

Weiterer Forschungsbedarf bezüglich eines einheitlichen Definitionsvorschlags für datengetriebene Prozesse wird dadurch bedingt, dass sich das Begriffsverständnis weiter im Wandel befindet. Ein Beispiel dafür liegt im Bereich der Datenerfassung (Kriterium 2 der Checkliste). Da im Kontext datengetriebener Prozesse häufiger die Datenaufnahme und -verarbeitung in Echtzeit beschrieben wird, muss das Kriterium „Datenerfassung“ der Checkliste in Zukunft gegebenenfalls um Echtzeitfähigkeit erweitert werden.

Wie auch im Anwendungsbeispiel der Checkliste gezeigt, werden datengetriebene Prozesse im Bereich der Simulation immer relevanter. Bei diesen datengetriebenen Modellen ist weitere Forschung in Bezug auf Verifikation und Validierung (V&V) notwendig. In allen gängigen Vorgehensmodellen kommt V&V eine wichtige Rolle zu, sodass ohne begleitende V&V ein Modell nicht als glaubwürdig gelten kann. Dies stellt in Bezug auf datengetriebene Modelle eine Herausforderung dar, da V&V-Techniken überwiegend von Experten durchgeführt werden, auch wenn bei einzelnen Tests und Techniken durchaus Automatisierungspotential besteht [26]. Besonders ist zu diskutieren, welche V&V-Techniken bei Modellen, die beispielsweise durch Machine Learning entstanden sind, sinnvoll anwendbar sind. Auch ist die Frage zu klären, ob die stichprobenartige V&V eines Modells ausreichend ist, oder ob eine durchgängige V&V auch bei kleineren Veränderungen im System weiterhin notwendig ist.

Wie bei der V&V ist es auch in weiteren Bereichen von datengetriebenen Prozessen noch nicht möglich,

komplett auf Expertenwissen zu verzichten. Da datengetriebene Prozesse hypothesengetrieben sind, müssen die auszuwertenden Daten oder Datenquellen zunächst basierend auf einem vermuteten Zusammenhang ausgewählt werden. Dies wird im Anwendungsbeispiel in Abschnitt 2 durch die Auswahl der zu messenden Prozessparameter deutlich.

Auch die Datenanalyse in datengetriebenen Prozessen kann nicht vollständig automatisiert werden. Beispielsweise ist bei Prozessen zur Fehlererkennung Expertenwissen beim Training von Neuronalen Netzen weiter essentiell, da ohne Training der Neuronalen Netzen neu auftretende Fehlertypen fälschlicherweise bereits bekannten Fehlertypen zugeordnet werden können [6].

Vorarbeiten zu diesem Paper basieren auf der Forschungsarbeit [27].

Literaturverzeichnis

- [1] Tao, F.; Qi, Q.; Liu, A.; Kusiak, A. Data-driven smart manufacturing. *Journal of Manufacturing Systems*. 2018; 48: S. 157–169. doi: 10.1016/j.jmsy.2018.01.006.
- [2] Shao, G.; Jain, S.; Shin, S. Data analytics using simulation for smart manufacturing. In: Tolk, A.; Diallo, S.Y.; Ryzhov, I.O.; Buckley, S.J.; Miller, J.A.; Yilmaz, L. (Hrsg.): *Proceedings of the 2014 Winter Simulation Conference*, 07.12.2014–10.12.2014, Savannah, GA. Piscataway, NJ: IEEE Press: S. 2192–2203. doi: 10.1109/WSC.2014.7020063.
- [3] Jeong, Y.; Singh, A.; Zafarzadeh, M.; Wiktorsson, M.; Baalsrud Hauge, J. Data-driven manufacturing simulation: towards a CPS-based approach. In: Säfsen, K.; Elgh, F. (Hrsg.): *SPS2020: Proceedings of the Swedish Production Symposium*, 07.10.2020–8.10.2020, Jönköping, Sweden. Amsterdam: IOS Press Incorporated: S. 587–596. doi: 10.3233/ATDE200197.
- [4] Kamble, S.S.; Gunasekaran, A. Big data-driven supply chain performance measurement system: a review and framework for implementation. *International Journal of Production Research*. 2020; 58(1): S. 65–86. doi: 10.1080/00207543.2019.1630770.
- [5] Nguyen, D.T.; Adulyasak, Y.; Cordeau, J.-F.; Ponce, S.I. Data-driven operations and supply chain management: established research clusters from 2000 to early 2020. *International Journal of Production Research*. 2021; 60(59): S. 1–25. doi: 10.1080/00207543.2021.1956695.
- [6] Wen, L.; Li, X.; Gao, L.; Zhang, Y. A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Transactions on Industrial Electronics*. 2018; 65(7): S. 5990–5998. doi: 10.1109/TIE.2017.2774777.
- [7] Qiao, F.; Liu, J.; Ma, Y. Industrial big-data-driven and CPS-based adaptive production scheduling for smart manufacturing. *International Journal of Production Research*. 2021; 59(23): S. 7139–7159. doi: 10.1080/00207543.2020.1836417.
- [8] Tao, F.; Cheng, J.; Qi, Q.; Zhang, M.; Zhang, H.; Sui, F. Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology*. 2018; 94(9-12): S. 3563–3576. doi: 10.1007/s00170-017-0233-1.
- [9] Na, Y.; Peng, S. A data-driven approach for identification and detection of intermittent faults. In: Liu, M.; Nogueiras Melendez, A.A.; Zhang, B.; Liu, X. (Hrsg.): *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 10.05.2021 - 12.05.2021, Victoria, BC, Canada. Piscataway, NJ: IEEE Press: S. 455–460. doi: 10.1109/ICPS49255.2021.9468167.
- [10] Hammond, P. From computer-assisted to data-driven: Journalism and Big Data. *Journalism*. 2017; 18(4): S. 408–424. doi: 10.1177/1464884915620205.
- [11] Men, H. Data-driven learning in enhancing learners' language idiomaticity. *International Journal of Emerging Technologies in Learning (iJET)*. 2020; 15(23): S. 27. doi: 10.3991/ijet.v15i23.19023.
- [12] Xu, K.; Li, Y.; Liu, C.; Liu, X.; Hao, X.; Gao, J.; Maropoulos, P.G. Advanced data collection and analysis in data-driven manufacturing process. *Chinese Journal of Mechanical Engineering*. 2020; 33(1): S. 1–21. doi: 10.1186/s10033-020-00459-x.
- [13] Deutsche Industrie Norm DIN 55350:2021-10, Begriffe zum Qualitätsmanagement. Berlin: Beuth.
- [14] Cerquitelli, T.; Pagliari, D.J.; Calimera, A.; Bottaccioli, L.; Patti, E.; Acquaviva, A.; Poncino, M. Manufacturing as a data-driven practice: methodologies, technologies, and tools. *Proceedings of the IEEE*. 2021; 109(4): S. 399–422. doi: 10.1109/JPROC.2021.3056006.
- [15] Subramanian, M.; Skoogh, A.; Salomonsson, H.; Bangalore, P.; Bokrantz, J. A data-driven algorithm to predict throughput bottlenecks in a production system based on active periods of the machines. *Computers & Industrial Engineering*. 2018; 125: S. 533–544. doi: 10.1016/j.cie.2018.04.024.
- [16] Zhang, Y.; Cheng, Y.; Wang, X.V.; Zhong, R.Y.; Zhang, Y.; Tao, F. Data-driven smart production line and its common factors. *The International Journal of Advanced Manufacturing Technology*. 2019; 103(1-4): S. 1211–1223. doi: 10.1007/s00170-019-03469-9.
- [17] Roh, P.; Kunz, A.; Netland, T. Data-driven detection of moving bottlenecks in multi-variant production lines. *IFAC-PapersOnLine*. 2018; 51(11): S. 158–163. doi: 10.1016/j.ifacol.2018.08.251.
- [18] Ivanov, D.; Dolgui, A. A digital supply chain twin for managing the disruption risks and resilience in the era of Industry 4.0. *Production Planning & Control*. 2021; 32(9): S. 775–788. doi: 10.1080/09537287.2020.1768450.

- [19] Zhang, Y.; Zhang, G.; Wang, J.; Sun, S.; Si, S.; Yang, T. Real-time information capturing and integration framework of the internet of manufacturing things. *International Journal of Computer Integrated Manufacturing*. 2015; 28(8): S. 811–822. doi: 10.1080/0951192X.2014.900874.
- [20] Tao, F.; Cheng, J.; Qi, Q. IIHub: an industrial internet-of-things hub toward smart manufacturing based on cyber-physical system. *IEEE Transactions on Industrial Informatics*. 2018; 14(5): S. 2271–2280. doi: 10.1109/TII.2017.2759178.
- [21] Kuo, Y.-H.; Kusiak, A. From data to big data in production research: the past and future trends. *International Journal of Production Research*. 2019; 57(15-16): S. 4828–4853. doi: 10.1080/00207543.2018.1443230.
- [22] Zhang, C.; Zhou, G.; He, J.; Li, Z.; Cheng, W. A data- and knowledge-driven framework for digital twin manufacturing cell. *Procedia CIRP*. 2019; 83: S. 345–350. doi: 10.1016/j.procir.2019.04.084.
- [23] Gutenschwager, K.; Rabe, M.; Spieckermann, S.; Wenzel, S. Simulation in Produktion und Logistik. Berlin, Heidelberg: Springer. 2017.
- [24] Qiao, Q.; Wang, J.; Ye, L.; Gao, R.X. Digital twin for machining tool condition prediction. *CIRP Procedia*. 2019; 81: S. 1388–1393. doi: 10.1016/j.procir.2019.04.049.
- [25] Bi, Z.; Da Xu, L.; Wang, C. Internet of things for enterprise systems of modern manufacturing. *IEEE Transactions on Industrial Informatics*. 2014; 10(2): S. 1537–1546. doi: 10.1109/TII.2014.2300338.
- [26] Rabe, M.; Wenzel, S.; Spieckermann, S. Verifikation und Validierung für die Simulation in Produktion und Logistik. Berlin, Heidelberg: Springer. 2008.
- [27] Heimeshoff, L. Data Driven - Untersuchung datengetriebener Prozesse im Umfeld von Produktion und Logistik [Bachelorarbeit]. [Fakultät Maschinenbau]. Technische Universität Dortmund, 2021.

Optimierung des Designs modularer Montagesysteme

Sören Bergmann^{1*}

¹Fachgebiet Informationstechnik in Produktion und Logistik, TU Ilmenau, Max-Planck-Ring 12, 98693 Ilmenau, Deutschland; *soeren.bergmann@tu-ilmenau.de

Abstract. Der Wunsch nach mehr Flexibilität in Fertigungssystemen, insbesondere, wenn verschiedene Produkte bzw. viele Produktvarianten in einem Produktionssystem gefertigt werden, führt, besonders in der Automobilindustrie, zur Abkehr vom Fertigungsprinzip der klassischen Linienfertigungen hin zu eher flexiblen und werkstatorientierten Produktionssystemen. Eine der Herausforderungen in diesen so genannten modularen Montage- bzw. Produktionssystemen ist das Systemdesign, insbesondere die Zuordnung der Tätigkeiten auf die einzelnen Fertigungsinseln. Einen Ansatz, diese Zuordnung zu verbessern, bietet die simulationsbasierte Optimierung. In diesem Beitrag wird ein Konzept zur simulationsbasierten Optimierung der Tätigkeitszuordnung zu einzelnen Fertigungsinseln in modularen Montagesystemen vorgestellt und anhand einer Fallstudie demonstriert. Zum Einsatz kommen hierbei genetische Algorithmen, speziell der NSGA-II-Algorithmus, welcher auch mehrkriterielle Optimierung ermöglicht.

Einleitung

Der in den letzten Jahren zunehmende Wunsch des Kunden zu mehr Individualität stellt Unternehmen vor neue Herausforderungen, welche sich durch kürzere Innovationszyklen und damit einhergehend oft auch kürzere Produktzyklen sowie durch äußere Einflüsse durch den Markt, Staat etc. noch verstärkt werden [12].

Dem folgend besteht eine wesentliche Anforderung an moderne Produktionssysteme, insbesondere in der Endmontage, darin, flexibel auf sich ändernde Bedingungen reagieren zu können, hierbei aber die Wirtschaftlichkeit der Produktion für die zum Teil große Zahl der durch ein Produktionssystem zu fertigenden Produkte bzw. Produktvarianten zu wahren [5, 16, 18].

Eine besonders in der Automobilindustrie, aber nicht beschränkt auf diese, gewonnene Erkenntnis ist, dass klassische Linienfertigungen an ihre Grenzen zu stoßen

drohen, wodurch eine Abkehr von diesem Fertigungsprinzip hin zu flexibleren Systemen zu untersuchen ist. Erste Pilotprojekte sind hierbei bereits gestartet [9]. In der Praxis sind verschiedene Bezeichnungen und Spielarten, wie beispielsweise „modulare Montage“ [2], „Flexi-Line“ [14] oder „Full-Flex-Werk“ [4] zu finden. Im Folgenden werden solche Konzepte unter dem Sammelbegriff der modularen Produktions- bzw. Montagesysteme subsumiert.

Ein solches modulares Produktions- bzw. Montagesystem besteht aus anpassungsfähigen Arbeitsstationen (Fertigungsinseln) mit spezifischen Werkzeugen, an denen ein oder oft mehrere unterschiedliche Produktions- bzw. Montageschritte durchgeführt werden können und welche durch fahrerlose Transportsysteme (FTS) verbunden sind. Hierbei entscheidet das System ad hoc, unter Berücksichtigung der vorhandenen technischen Restriktionen und der individuellen Aufgabenpakete der zu fertigenden Produkte bzw. Produktvarianten, welchen Weg das Produkt durch das Produktionssystem nimmt [5].

Neben der großen Herausforderung der Steuerung solcher Systeme ist das Systemdesign entscheidend für die Performance des Produktionssystems in Gänze. Wesentliche Punkte innerhalb des Systemdesigns sind in der Festlegung der Zahl der Fertigungsinseln sowie der Aufgabenverteilung auf diese zu sehen [11]. Ein möglicher Ansatz zur Lösung solcher komplexen dynamischen Problemstellungen stellt die simulationsbasierte Optimierung, z. B. unter Nutzung genetischer Algorithmen, dar [17]. Die Konzeption und Implementierung eines solchen Ansatzes sowie eine kleine Fallstudie stehen im Fokus dieses Beitrags.

Der Beitrags ist hierbei wie folgt aufgebaut: nach der Einleitung folgt eine Definition sowie ein Überblick zur modularen Produktion/Montage auf der einen Seite und zur simulationsbasierten Optimierung auf der anderen Seite. Danach wird das Konzept zur simulationsbasierten Optimierung der Zahl der Fertigungsinseln sowie der

Aufgabenverteilung auf diese, als wesentliche Komponente des Systemdesigns eines modularen Montagesystems, vorgestellt. Das Konzept wird im nächsten Abschnitt anhand einer akademischen Fallstudie demonstriert. Ein Kapitel mit Fazit und Ausblick schließt den Beitrag ab.

1 Grundlagen und Stand der Forschung

1.1 Modulare Montage

Beim Einsatz des Konzepts der modularen Montage, welches hier synonym für eine Reihe ähnlicher Begriffe genutzt wird, wird das klassische Prinzip der Fließfertigung zugunsten einer eher werkstatorientierten Montage aufgelöst. Die Fertigung, oft wird die Endmontage von z. B. Automobilen betrachtet, erfolgt in der modularen Montage auf Fertigungsinseln bzw. Stationen. Diese können jeweils sehr flexibel und abhängig des jeweilig zu fertigenden Produktes bzw. der Produktvariante verschiedene Aufgaben verrichten. Der Transport der Werkstücke zwischen den Fertigungsinseln sowie auch der Transport von Material wird von FTS übernommen.

Zusammenfassend ist eine modulare Montage durch folgende Charakteristiken gekennzeichnet:

- Entkoppelte Arbeitsstationen mit individuellen Taktzeiten
- Mehrere Tätigkeiten / Fähigkeiten je Fertigungsinsel
- Flexible Materialflüsse mittels FTS

Zu beachten ist, dass keine fixen Arbeitsschrittfolgen definiert werden, sondern die Wahl des nächsten Arbeitsschrittes basierend auf den für das konkrete Produkt technisch möglichen nächsten Schritten, dem Systemdesign, sowie dem aktuellen Zustand des Produktionssystems erfolgen. Die Definition der technisch möglichen nächsten Schritte erfolgt in der Regel mittels Vorranggraphen. Im Rahmen der Zustandsanalyse des Produktionssystems können verschiedenste Kennwerte herangezogen werden. Diese reichen von eher einfachen Erhebungen, z. B. bzgl. den Warteschlangenlängen vor möglichen nächsten Stationen zum Entscheidungszeitpunkt, über Daten bzgl. des eigenen Produktionsfortschrittes bis hin zu komplexen Betrachtungen im Zusammenspiel mit den konkurrierenden Aufträgen im System.

Das Systemdesign selbst hat verschiedene Gestaltungsdimensionen vom grundlegenden Layout d. h. der

grundlegenden räumlichen Struktur möglicher Fertigungsinseln über die Aufgaben und Tätigkeitsverteilung bis hin zur Bestimmung der Größe der FTS-Flotte, der Wahl der Steuerungsalgorithmen, insbesondere für das Routing der FTS, und dem Materialversorgungskonzept. Einzelne Teilaspekte des Systemdesigns werden in der Praxis durch eine Vielzahl technischer und organisatorischer Restriktionen eingeschränkt, z. B. können technische Parameter verbieten, dass bestimmte Tätigkeitskombinationen an einer Fertigungsinsel durchgeführt werden können [5, 15].

Das Konzept der modularen Montage führt zu einer inhärenten Flexibilität der Fertigung, sowohl in Form von Ressourcenflexibilität, d. h. eine Station kann verschiedene Aufgaben wahrnehmen, als auch in Form von Routenflexibilität, da Produkte in verschiedenen Arbeitsschrittreihenfolgen bzw. selbst bei gleichen Reihenfolgen ggf. auf verschiedenen Fertigungsinseln produziert werden können.

Die beiden Flexibilitätsdimensionen werden maßgeblich durch das Systemdesign, insbesondere die Tätigkeitsverteilung, bestimmt. Sowohl Ressourcen- als auch Routenflexibilität sollen im Folgenden neben eher klassischen Kennzahlen zur Bewertung des Designs dienen. Folgende Kennzahlen sind unter anderem zur Bewertung des Systemdesigns denkbar [6, 9]:

- Durchsatz
- Liefertreue je Produkt/ Produktvariante
- Durchlaufzeiten je Produkt/ Produktvariante
- Flächenproduktivität (Durchsatz bzgl. Fläche)
- Mittlere Auslastung der Fertigungsinseln
- Zurückgelegte Distanz aller FTS
- Zahl der möglichen Routen im System für alle Produkte/Produktvarianten (globale Routenflexibilität)
- Mittlere Zahl der Entscheidungsmöglichkeiten für den nächsten Arbeitsschritt – EMNS (lokale Routenflexibilität)
- Diverse Kosten etc.

1.2 Simulationsbasierte Optimierung

Simulation ist ein etabliertes Werkzeug zur Modellierung, Planung und Steuerung von Produktionssystemen. Die Kopplung von Simulation und Optimierungsalgorithmen ist eine durchaus verbreitete und vielfach untersuchte Methode. Ziel ist es, für ein zu betrachtendes System, bzw. dessen Nachbildung in Form eines Simulationsmodells, die beste (oder zumindest eine gute) Kombination von Systemeinstellungen (Faktoren) zu finden.

Hierbei werden im Prozess der simulationsbasierten Optimierung durch den Optimierungsalgorithmus Faktorwerte (und somit Systemkonfigurationen) zielgerichtet gesetzt, die Simulation wird anschließend zur Bestimmung der Fitness bzw. des Zielwertes/ der Zielwerte unter Einstellung der Faktorwerte genutzt [1, 3, 7, 13].

Im Rahmen der simulationsbasierten Optimierung kommen in der Literatur sehr verschiedene Algorithmen zum Einsatz. Die Wahl des geeigneten Algorithmus hängt hierbei stark von dem zu untersuchenden Anwendungsfall ab [1, 13]. In entsprechenden Veröffentlichungen häufig genutzte Algorithmen sind beispielsweise: genetische Algorithmen (GA), Simulated Annealing, Tabu-Suche, Scatter Search, Ameisenalgorithmen oder auch die Partikelschwarmoptimierung [1, 3]. Zur Optimierung des Designs modularer Montagesysteme wurden vom Autor verschiedene Algorithmen untersucht, im Rahmen dieses Beitrags soll sich auf einen Algorithmus beschränkt werden. Diese Entscheidung basiert auf den in der Literatur beschriebenen allgemein guten Eigenschaften, der Verbreitung der GA sowie der Fähigkeit, je nach genutzter Variante des Optimierungsalgorithmus sogar mehrkriterielle Optimierung zu betreiben. Des Weiteren waren die Ergebnisse der hier aus Platzgründen nicht im Detail beschreibbaren Vorversuchen, bei der Nutzung von genetischen Algorithmen, hier im speziell den für mehrkriterielle Optimierung geeignete non dominated sorting genetic algorithm II (NSGA-II), am vielversprechendsten.

Genetische Algorithmen im Allgemeinen sind Algorithmen aus der Klasse der evolutionären Algorithmen. Hierbei werden einzelne Lösungen (Individuen) durch ihre Faktorwerte (Gene) beschrieben und können z. B. mittels der Simulation bzgl. einer Zielgröße (Fitness) bewertet werden. Zum Start der Optimierung wird eine definierte Menge zufälliger Lösungskandidaten erzeugt (1. Generation der Population) und deren Fitnesswerte ermittelt. Anschließend werden folgende Schritte für die aktuelle Population (Generation) bis zum Erreichen eines Abbruchkriteriums (z. B. bestimmte verstrichene Zeit, bestimmte Anzahl an Generationen erreicht, keine Verbesserung in den letzten Schritten erzielt) durchlaufen:

1. Rekombination/Mutation: Hinzufügen neuer Lösungskandidaten durch Kombination zweier Individuen der aktuellen Generation bzw. zufällige Veränderung eines solchen Kandidaten
2. Bewertung: Ermittlung der Fitnesswerte für alle neuen Lösungskandidaten
3. Selektion: Bestimmung der nächsten Generation der

Population durch Auswahl der besten Lösungskandidaten.

Im Kontext der Optimierung des Designs modularer Montagesysteme soll meist ein Problem mit mehreren ggf. konkurrierenden oder gar gegensätzlichen Zielgrößen, z. B. Auslastung der Fertigungsinseln vs. mittlere Durchlaufzeit der Produkte, gelöst werden. Einer der hierbei meist genutzten (genetischen) Algorithmen ist der NSGA-II, welcher Kompromisslösungen (so genannte Pareto-optimale Lösungen) ermittelt, aus welchen der Nutzer, basierend auf eigenen Präferenzen, die geeignetste auswählen kann. Pareto-optimale Lösungen sind hierbei Lösungen, bei denen sich eine Zielgröße nicht weiter verbessern lässt ohne eine andere zu verschlechtern (nicht dominierte Eigenschaft). Der NSGA-II Algorithmus bewertet nun im Rahmen der Selektion der fittesten Faktorkombinationen (Individuen) aktuell nicht dominierte Lösungen besonders gut [19].

Bezüglich weiteren Details zu genetischer Algorithmen, insbesondere den grundlegenden Operatoren, wird an dieser Stelle auf entsprechende Literatur, z. B. [8], verwiesen. Details des NSGA-II Algorithmus sind unter anderem in [19] zu finden.

2 Konzept zur simulationsbasierten Optimierung der Tätigkeitszuordnung in modularen Montagesystemen

Das Konzept zur simulationsbasierten Optimierung des Systemdesigns modularer Montage- bzw. Produktionssysteme (vgl. Abbildung 1) ist möglichst generisch gestaltet worden, um in vielen Anwendungsfällen nutzbar zu sein. So muss als Basis nur ein geeignetes Simulationsmodell mit dem entsprechenden Grundlayout der Produktion vorhanden sein und geeignete Schnittstellen zu diesem ausimplementiert werden, alle sonstigen Rahmenbedingungen sowie die Parameter der Optimierung werden über hinterlegte jederzeit anpassbare Datenstrukturen beschrieben.

Ziel des Konzeptes ist es, dem Planer von modularen Produktions- bzw. Montagesystemen bei der Zuweisung von Tätigkeiten/Fähigkeiten zu Fertigungsinseln zu unterstützen und somit die Erzeugung eines möglichst guten Systemdesigns zu unterstützen. Mittelbar kann auch die Zahl der benötigten Fertigungsinseln bestimmt wer-

den, wobei eine Maximalzahl durch das Layout vorgegeben wird und sich der tatsächliche Bedarf durch Maximalzahl minus ggf. nicht mit Tätigkeiten beauftragten Stationen ergibt.

Wie Abbildung 1 zu entnehmen ist, basiert das Konzept auf vier Hauptbestandteilen: Eingangsdaten, Datenbanken, Simulationsmodell, Optimierungsbaustein, welche im folgendem kurz diskutiert werden.

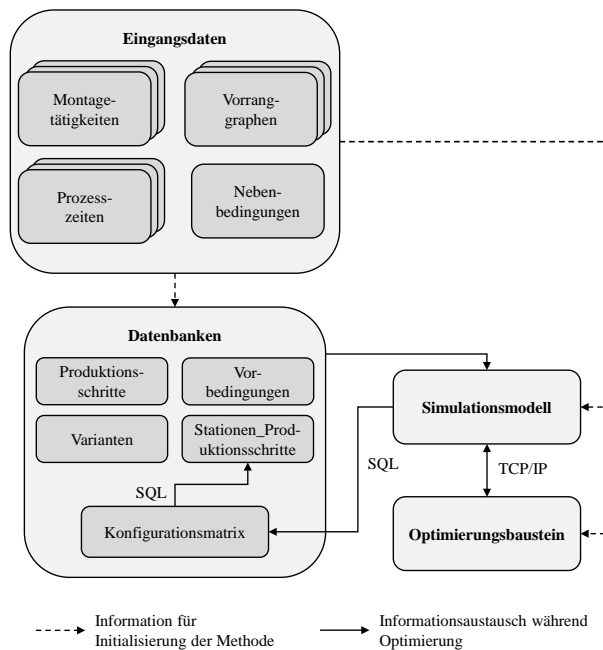


Abbildung 1: Grobkonzept zur simulationsbasierten Optimierung des Systemdesigns modularer Produktionssysteme.

Die wichtigsten Eingangsdaten sind, wie bereits in Abschnitt 1.1 angerissen, die benötigten Prozessschritte der einzelnen Produkte bzw. Produktvarianten, deren Abhängigkeiten zueinander in Form von Vorranggraphen, die Prozesszeiten, sowie Nebenbedingungen wie der Ausschluss bestimmter Tätigkeiten an einer Station (Inkompatibilität) mittels Kompatibilitätsmatrizen, die Zahl der max. verfügbaren FTS sowie die max. erlaubte Zahl von Tätigkeiten je Fertigungsinsel. Neben diesen jeweils zu definierenden Nebenbedingungen sind aktuell einige Bedingungen im Prototypen fest definiert. So ist festgeschrieben, dass jede benötigte Tätigkeit mindestens einer Fertigungsinsel zugewiesen sein muss, um eine gültige Lösung darzustellen. Jede Fertigungsinsel kann zudem nur ein Produkt zur gleichen Zeit bearbeiten (keine Parallelstationen). Des Weiteren wird hier davon ausgegangen, dass jedes FTS nur ein Produkt transportiert und

die Bearbeitung inline erfolgt, d. h. die Bearbeitung erfolgt auf dem FTS, somit wird ein FTS (wenn vorhanden) beim Eintreffen des Produktes im System allokiert und erst nach dessen vollständiger Bearbeitung wieder freigegeben. Die Steuerung der FTS erfolgt aktuell nach einer einfachen Heuristik, bei welcher immer die kürzest mögliche Warteschlange angesteuert wird. Andere Steuerungsalgorithmen sind aber implementierbar wobei dann der Steuerungsalgorithmus auch als weiterer Faktor im Optimierungsprozess genutzt werden könnte.

Die Eingangsdaten aus verschiedenen Quellen werden zunächst während der Initialisierungsphase, zur leichten Weiterverarbeitbarkeit, in der Datenbankkomponente in einzelne Datentabellen überführt, zusätzlich werden in entsprechenden Tabellen (Konfigurationsmatrizen) die jeweils aktuell betrachteten Systemdesigns gespeichert. Eine Konfigurationsmatrix beinhaltet hierbei für jede Station die zugeordneten Tätigkeiten. Ein Beispiel für eine solche Matrix ist in Abbildung 2 zu sehen.

konfigurations_matrix					
	produktionsschritte	station1	station2	station3	station4
1	A	1	0	1	0
2	B	1	1	0	1
3	C	0	1	1	0
4	D	0	0	1	1

Abbildung 2: Beispiel einer Konfigurationsmatrix.

Die in der Datenbankkomponente gespeicherten Informationen bilden zudem u.a. die Basis zur Initialisierung des Simulationsmodells und des Optimierungsbausteins.

Zur Simulation wird hier auf einen agentenbasierten Ansatz, implementiert in AnyLogic, zurückgegriffen, welcher bereits in [5] als Vergleichsimplementierung herangezogen wurde (siehe Abbildung 3).



Abbildung 3: Screenshot des AnyLogic Simulationsmodells der modularen Montage.

Auch für die Optimierungskomponente wird auf eine bestehende Software zurückgegriffen, die Wahl fiel auf HeuristicLab [10], da diese zum einen viele Optimierungsalgorithmen unterstützt, das Optimierungsproblem leicht zu parametrieren ist und zudem mittels C# Programmierung leicht Erweiterungen implementiert werden können. Erweiterungen wurden hier konkret sowohl für die Kommunikation des Optimierers mit dem Simulationsmodell per TCP/IP, als auch für die Überprüfung der Nebenbedingungen und ggf. nötigen Korrekturmechanismen entwickelt. Im Prototyp sind sechs verschiedene Zielfunktionen hinterlegt, welche aber jederzeit durch weitere ergänzt werden können.

Aktuell sind folgende Zielfunktionen verfügbar:

- Max. des Durchsatzes
- Max. der Auslastung der Montagestationen
- Min. der zurückgelegten Gesamtdistanz der FTS
- Max. der Auslastung der FTS
- Max der Flexibilität (EMNS)
- Max der Routenflexibilität

Der schematische Ablauf der Optimierung ist Abbildung 4 zu entnehmen.

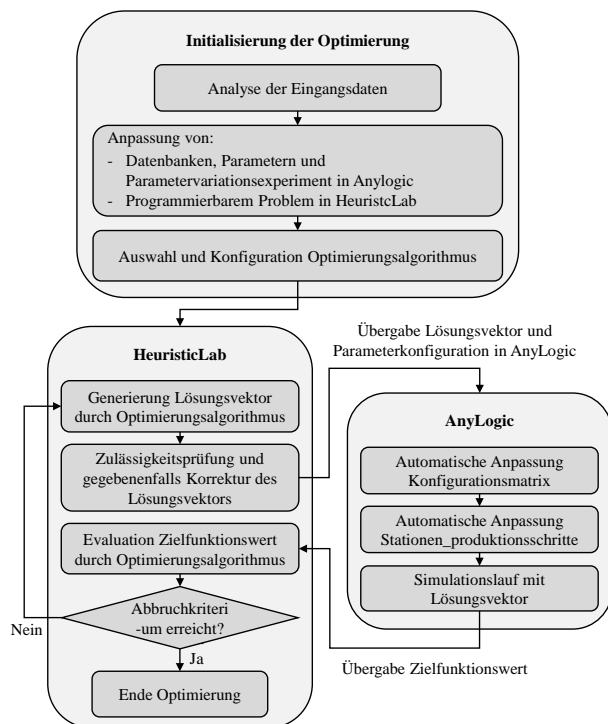


Abbildung 4: Schematischer Ablauf der Optimierung.

3 Fallstudie

Zur Evaluierung des Konzeptes wurden verschiedene Szenarien definiert und mittels verschiedener Optimierungsmethoden, Zielfunktionen etc. untersucht.

In diesem Beitrag wird zur Illustration ein fiktives Produktionssystem herangezogen, welches in allen Aspekten an die in der Literatur beschriebenen Realweltsystemen angelehnt ist, aber eine geringere Anzahl von verschiedenen Produktvariationen, Tätigkeiten und Fertigungsinseln aufweist. Ein skalieren auf Systemgrößen, welche in der Praxis auftreten, wird unterstellt, wurde aber bisher nicht untersucht.

In der Fallstudie werden zwei Produkte (P1 und P2) in verschiedenen Varianten gefertigt. Insgesamt sind 7 verschiedene Tätigkeiten (A bis G) auf bis zu 9 Fertigungsinseln (Station 1 – 9) zu verteilen. Die entsprechenden Vorranggraphen der Produkte sind Abbildung 5 zu entnehmen. Produktvarianten ergeben sich aus den optionalen Tätigkeiten. Die dreiecksverteilten Prozesszeiten der einzelnen Tätigkeiten sind in Tabelle 1 aufgeführt. Ein Produktionsauftrag für eine der Produktvarianten trifft alle 120 Sekunden im System ein. Die Häufigkeit der Produkte bzw. Produktvarianten ist hierbei für alle gleich.

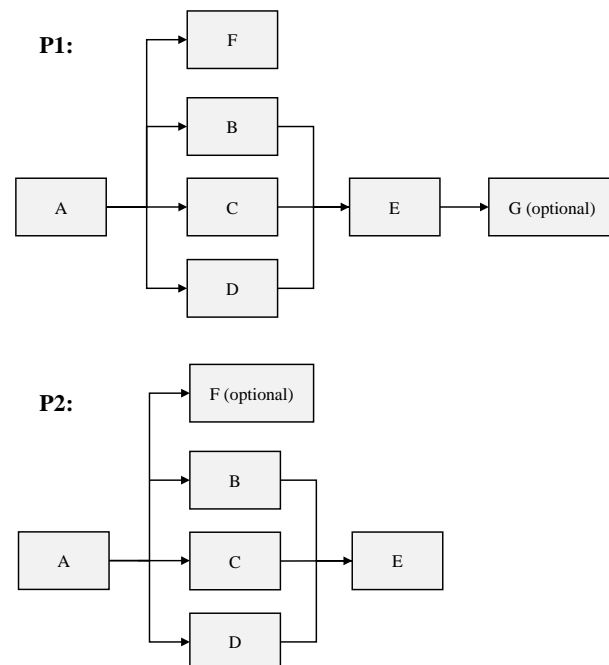


Abbildung 5: Vorranggraphen der Produkte P1 und P2.

Als Fertigungslayout wird eine Matrixstruktur angenommen, welche ein vollständig 2-spuriges Wegenetz aufweist. Die Materialversorgung erfolgt per Warenkörben, welche analog dem Produkt fest mit dem FTS verbunden sind.

Montage-tätigkeit	Prozesszeit in Sekunden (min Wert, wahrscheinlichster Wert, max Wert)
A	90, 100, 110
B	180, 200, 220
C	50, 70, 90
D	60, 80, 100
E	200, 220, 240
F	80, 100, 120
G	90, 100, 110

Tabelle 1: Prozesszeiten (dreiecksverteilt).

Als Nebenbedingungen wurden unter anderem einige Inkompatibilitäten zwischen einzelnen Tätigkeiten definiert, diese sind in Tabelle 2 zu sehen. Die Prozesskompatibilitätsmatrix ist dabei so zu lesen, dass sich Tätigkeitskombinationen mit dem Wert 0 an einer Fertigungsinsel ausschließen, hier z. B. die Tätigkeiten A und F. Die maximal zulässige Zahl von Tätigkeiten auf einer Fertigungsinsel wurde zentral auf 4 festgelegt.

	A	B	C	D	E	F	G
A	0	1	1	1	1	0	0
B	1	0	1	1	1	0	0
C	1	1	0	1	1	0	0
D	1	1	1	0	1	0	0
E	1	1	1	1	0	1	1
F	0	0	0	0	1	0	1
G	0	0	0	0	1	1	0

Tabelle 2: Prozesskompatibilitätsmatrix.

Das skizzierte System wurde mit dem NSGA-II-Algorithmus optimiert. Folgende Werte für die Hyperparameter wurden genutzt:

- Populationsgröße: 100
- Anzahl von Generationen: 6
- Simulationsdauer: 2 Stunden

- Selektionsverfahren: Crowded Tournament Selector
- Rekombinationsverfahren: Rounded Average Crossover
- Rekombinationswahrscheinlichkeit: 90 %
- Mutationsverfahren: Uniform Some Positions Manipulator
- Mutationswahrscheinlichkeit: 5 %

Zunächst wurden der Gesamtdurchsatz und die Gesamtauslastung der Fertigungsinseln als Zielfunktionen untersucht. Die Idee hinter der Wahl der Zielfunktionen war es Lösungen zu finden, die zwar einen möglichst hohen Durchsatz erreichen, aber nicht einfach alle Tätigkeiten maximal oft zuweisen. Idealerweise sind ggf. sogar Lösungen zu finden, in denen weniger als die 9 maximal verfügbaren Fertigungsinseln benötigt werden.

Die Ergebnisse sind in Tabelle 3 aufgeführt. Zu erkennen ist, dass Durchsatz im Pareto-Optimum 2 am höchsten ist, wobei allen Fertigungsinseln min. eine Tätigkeit zugewiesen wurde. Pareto-Optimum 3 erscheint hingegen als guter Kompromiss zwischen der Zahl der benötigten Fertigungsinseln (6 der max. 9) und im Vergleich zu den anderen Lösungen auch der Auslastung der Fertigungsinseln sowie dem Durchsatz, der moderat niedriger ist als in Pareto-Optimum 2.

	Pareto-Optimum 1	Pareto-Optimum 2	Pareto-Optimum 3
Durchsatz	15	18	17
Auslastung	32,14 %	28,81 %	31,85 %
Anzahl FTFs	5	5	5
Station 1	F	A	F
Station 2	G	C	G
Station 3	-	E	-
Station 4	-	G	-
Station 5	-	D	E
Station 6	-	F	-
Station 7	E	B	E
Station 8	A, B	A, B	A, B
Station 9	C, D	C, D	C, D

Tabelle 3: Lösungen des NSGA-II-Algorithmus für die Zielgrößen Durchsatz und Auslastung.

	Pareto-Optimum 1	Pareto-Optimum 2	Pareto-Optimum 3	Pareto-Optimum 4
Durchsatz	13	19	15	17
Routenflexibilität	548	211	454	267
Anzahl FTFs	5	5	5	5
Station 1	A, C, D, E	A	B, E	A
Station 2	B	B	D, E	C
Station 3	E, F, G	F	E, F, G	F
Station 4	A, C, D, E	C	D	E
Station 5	A, D	D	A, E	E
Station 6	B, C, D	E	G	B, E
Station 7	E, G	G	D, E	B, D
Station 8	E, F, G	-	A, B, C	E
Station 9	B, C	-	C	G

Tabelle 4: Lösungen des NSGA-II-Algorithmus für die Zielgrößen Durchsatz und Routenflexibilität.

Als zweites Beispiel wurde hier, ebenfalls mit dem NSGA-II-Algorithmus, eine Optimierung bzgl. des Gesamtdurchsatzes und der Routenflexibilität vorgenommen. Hintergrund dieser Wahl der Zielfunktionen ist es, ein Systemdesign zu finden, das zum einen als notwendige Bedingung einen guten Durchsatz hat, zum anderen aber auch eine hohe Flexibilität aufweist. Eine Annahme ist hierbei, dass eine solche hohe Flexibilität ggf. eine gewisse Robustheit gegen Störungen im System, aber auch gegen schwankende Verteilungen der Häufigkeiten der beiden Produkte sichert. Des Weiteren ist ggf. auch die Erweiterung des Systems, z. B. bei der Einführung weiterer Produkte, leichter zu realisieren.

Wie Tabelle 4 zu entnehmen, wurden auch hier Lösungen gefunden, die einen hohen Durchsatz realisieren. So wurde im Pareto-Optimum 2 ein Durchsatz von 19 erreicht. Zu bemerken ist hier aber, dass dies eine Lösung ist, die im Vergleich zu den anderen Lösungen eine eher geringe Routenflexibilität aufweist. Allgemein ist zu sehen, dass Lösungen mit einer hohen Routenflexibilität offenbar eher moderate Durchsätze realisieren. Zudem

geht, wie auch zu vermuten war, eine hohe Routenflexibilität (z. B. im Pareto-Optimum 1) mit der Zuweisung von recht vielen Tätigkeiten je Station und der Nutzung möglichst vieler Stationen einher.

Zusammenfassend lässt sich sagen, dass je nach den Präferenzen des Entscheiders durchaus sehr verschiedene Lösungen relevant werden könnten. Nimmt man an, dass ein hoher Durchsatz wünschenswert ist und auch der Verzicht auf die Nutzung ganzer Fertigungsinseln zum einen Kosten spart aber zum anderem auch positiv im Fall möglicher Erweiterungen ist, so ist das Pareto-Optimum 2 der NSGA-II-Optimierung nach den Zielgrößen Durchsatz und Routenflexibilität zu präferieren. In dieser Lösung wird mit 19 der höchste Durchsatz aller hier ermittelten Lösungen erreicht und dies wobei zwei der möglichen Fertigungsinseln nicht benötigt werden. Allein die ermittelte Routenflexibilität ist im Vergleich zu den anderen hier aufgeführten Lösungen eher gering.

4 Fazit und Ausblick

In diesem Beitrag wurde ein Konzept zur simulationsbasierten Optimierung der Tätigkeitszuordnung zu Fertigungsinseln als wesentlicher Aspekt des Systemdesigns von modularen Produktionssystemen unter Nutzung des NSGA-II-Algorithmus vorgestellt. Hierzu wurden zunächst die Besonderheiten von modularen Produktionssystemen diskutiert, bevor die Grundzüge der simulationsbasierten Optimierung, insbesondere mittels dem NSGA-II-Algorithmus, beschrieben wurden.

Anhand eines einfachen akademischen Fallbeispiels wurde das Konzept bzw. dessen prototypischen Implementierung demonstriert. Es konnte gezeigt werden, dass durchaus Optimierungspotentiale existieren und auch mit dem Konzept realisiert werden konnten.

In Zukunft sind diverse Richtungen für weitere Forschungen denkbar. So sind weitere Optimierungsalgorithmen sowie das Erweitern bzw. Anpassen der Nebenbedingungen, z. B. Platzbedarfe von Tätigkeiten, möglich. Des Weiteren sind auch Anpassungen an den Korrekturmechanismen bei ungültigen Lösungen ein möglicher Forschungsgegenstand. Schließlich können auch weitere Gestaltungsdimensionen der modularen Produktion einbezogen werden, so wären die Betrachtung von Materialbereitstellungskonzepten oder alternativen Grundlayouts möglich. Schließlich wäre es zudem spannend das Konzept an Echtweltszenarien zu testen.

Literaturverzeichnis

- [1] Amaran, S., Sahinidis, N. V., Sharda, B., and Bury, S. J. 2016. Simulation optimization: a review of algorithms and applications. *Annals of Operations Research* 240, 1, 351–380.
- [2] Audi AG. 2019. *Die Modulare Montage - Fertigungsinseln statt Fließband*. <https://www.audi-illustrated.com/de/smart-factory/Die-Modulare-Montage>. Accessed 28 January 2019.
- [3] Beham, A., Affenzeller, M., and Wagner, S. 2008. Simulation optimization with HeuristicLab. In *Proceedings of the 20th European Modeling and Simulation Symposium, EMSS 2008*. DIPTeM, Genova, 75–80.
- [4] Daimler AG. 2018. *Factory 56 – Mercedes-Benz Cars increases flexibility and efficiency in operations*. <https://www.daimler.com/innovation/production/factory-56.html>.
- [5] Feldkamp, N., Bergmann, S., and Straßburger, S. 2019. Modellierung und Simulation von modularen Produktionssystemen. Modelling and Simulation of Modular Production Systems. In *Tagungsband der 18. ASIM Fachtagung Simulation in Produktion und Logistik 2019*. Wissenschaftliche Scripten, Auerbach, 391–401. DOI=10.1515/9783110924992-003.
- [6] Foith-Förster, P., Eising, J.-H., and Bauernhansl, T. 2017. Effiziente Montagesysteme ohne Band und Takt. *wt Werkstattstechnik online* 107, 3, 169–175.
- [7] Fu, M. C. 2015. *Handbook of simulation optimization*. International Series in Operations Research & Management Science 216. Springer Science+Business Media, New York.
- [8] Gerdes, I., Klawonn, F., and Kruse, R. 2004. *Evolutive Algorithmen. Genetische Algorithmen - Strategien und Optimierungsverfahren - Beispielanwendungen*. Computational intelligence. Vieweg, Wiesbaden.
- [9] Göppert, A., Hüttemann, G., Jung, S., Grunert, D., and Schmitt, R. 2018. Frei verkettete Montagesysteme. *ZWF* 113, 3, 151–155.
- [10] 2021. *HeuristicLab*. <https://dev.heuristiclab.com/trac.fcgi/>. Accessed 25 October 2021.
- [11] Kern, W., Rusitschka, F., and Bauernhansl, T. 2016. Planning of Workstations in a Modular Automotive Assembly System. In *Proceedings of the 49th CIRP 2016. Factories of the Future in the Digital Environment*. Elsevier BV, Amsterdam, 327–332. DOI=10.1016/j.procir.2016.11.057.
- [12] Koren, Y. 2010. *The Global Manufacturing Revolution. Product-Process-Business Integration and Reconfigurable Systems*. Wiley Series in Systems Engineering and Management 1. John Wiley & Sons, New York, NY.
- [13] März, L., Krug, W., Rose, O., and Weigert, G., Eds. 2011. *Simulation und Optimierung in Produktion und Logistik. Praxisorientierter Leitfaden mit Fallbeispielen*. VDI-Buch. Springer, Heidelberg.
- [14] Mayer, B. 2018. *Ich bin ein Fan von Effizienz*. <https://www.automobil-produktion.de/interviews-734/porsche-produktionsvorstand-reimold-wir-ziehen-alle-register-126.html?page=4>. Accessed 28 January 2019.
- [15] Poll, D. 2016. *Das Ende des Fließbands*. <https://www.produktion.de/trends-innovationen/das-ende-des-fließbands-306.html>. Accessed 23 April 2019.
- [16] Spath, D., Ed. 2013. *Produktionsarbeit der Zukunft - Industrie 4.0*. Fraunhofer-Verl., Stuttgart.
- [17] Völker, S. and Schmidt, P.-M. 2010. Simulationsbasierte Optimierung von Produktions- und Logistiksystemen mit Tecnomatix Plant Simulation. Simulation-based Optimization of Production and Logistics Systems using Tecnomatix Plant Simulation. In *Tagungsband der 14. ASIM-Fachtagung Simulation in Produktion und Logistik - Integrationsaspekte der Simulation: Technik, Organisation und Personal*. ASIM-Mitteilung 131. Technische Informationsbibliothek u. Universitätsbibliothek; KIT Scientific Publ, Hannover, Karlsruhe, 93–100.
- [18] Wack, K.-J. 2019. *Interdisziplinäre Absicherung der Produktionsplanung in der Automobilindustrie*. Dissertation. Universitätsverlag, Ilmenau.
- [19] Yusoff, Y., Ngadiman, M. S., and Zain, A. M. 2011. Overview of NSGA-II for Optimizing Machining Process Parameters. *Procedia Engineering* 15, 3978–3983.

Untersuchung von Auftragsfreigabestrategien für mehrzonige Kommissioniersysteme im Online-Lebensmittelhandel

Stefan Galka^{1*}

¹Ostbayerische Technische Hochschule Regensburg,
Galgenbergstr. 30, D-93053 Regensburg, *stefan.galka@oth-regensburg.de

Abstract. This paper briefly presents the challenges for order control and release in multi-zone order picking systems. On the one hand, the order control must ensure that all orders are processed on time, and on the other hand, the space requirements (buffer) and the utilisation of the zones must be considered.

Within the framework of a case study, different strategies for order release were developed. The paper shortly describes the ideas of the strategies and presents results of a case-based simulative evaluation of the strategies. The findings of the simulation study are the basis for the development of a digital twin for the operational control of multi-zone picking systems. In addition to the order release, the digital twin should also supervise workforce disposition and the allocation of staging locations for products depending on anticipated sales.

Einleitung

Die Kommissionierung für den Online-Lebensmittelhandel erfolgt in der Regel in mehrzonigen Kommissioniersystemen, da unterschiedliche Anforderungen an die Lagerung gegeben sind (z.B. Temperatur). Um kurze Durchlaufzeiten zu ermöglichen, ist eine parallele Bearbeitung der Teilaufträge in den einzelnen Zonen notwendig. Dies hat zur Konsequenz, dass die Teilaufträge vor der Auslieferung konsolidiert werden müssen. Um die Konsolidierung und Kommissionierung voneinander zu entkoppeln, werden die Teilaufträge in einem Puffer zusammengeführt. Wenn alle Teilaufträge im Puffer vereinnahmt wurden, kann die Konsolidierung beginnen. Durch unterschiedliche Durchlaufzeiten in den einzelnen Kommissionierzonen kann es sein, dass die Verweilzeit von einzelnen Teilaufträgen im Puffer sehr groß ist, was eine entsprechende Pufferkapazität voraussetzt. In diesem Beitrag wird anhand eines Fallbeispiels aufgezeigt,

welchen Einfluss unterschiedliche Strategien für die Freigabe der Teilaufträge auf die Pufferkapazität und die Termintreue hat. Dazu werden unterschiedliche Strategien kurz vorgestellt und mit Hilfe eines ereignisdiskreten Simulationsmodells untersucht.

1 Problemstellung

Kommissioniersysteme bestehen häufig aus verschiedenen Bereichen (Zonen), in denen die Artikel für einen Auftrag kommissioniert werden. Die Gründe hierfür sind vielfältig. Zum einen können unterschiedliche Kommissioniertechniken zum Einsatz kommen, um den Anforderungen an die Auftragsbearbeitung besser gerecht zu werden. Auch können Produktanforderungen wie eine temperierte Lagerung Ursachen für eine Aufteilung sein. Zum anderen kann das Streben nach kurzen Durchlaufzeiten ein Beweggrund für eine Zonierung und die parallele Bearbeitung von Teilaufträgen sein. Die VDI-Richtlinie 3590 beschreibt die Möglichkeiten für die Aufbau- und Ablauforganisation von Kommissioniersystemen [1].

Im stark wachsenden Online-Lebensmittelhandel lassen sich mehrzonige Kommissioniersysteme nicht vermeiden, da das zu handhabende Sortiment unterschiedliche Anforderungen an die Temperatur stellt. Weiterhin ist eine sehr schnelle und termingerechte Bearbeitung der Aufträge notwendig, um die Kunden teilweise am Tag der Bestellung zu beliefern und die kleinen Anlieferzeitfenster beim Kunden einzuhalten. [2,3] Der Grundstein hierfür wird durch eine hohe Termintreue im Distributionszentrum gelegt.

Einer der kritischsten Bereiche in einem Kommissioniersystem für den Online-Lebensmittelhandel ist die Konsolidierung der Teilaufträge aus den unterschiedlichen Kommissionierbereichen. Dabei muss sichergestellt

werden, dass alle Teilaufträge termingerecht für den Verpackprozess (Konsolidierung) im Konsolidierungspuffer bereitstehen. Durch unterschiedliche Durchlaufzeiten in den einzelnen Bereichen können sich die Ankunftszeiten auch bei einer gleichzeitigen Auftragsfreigabe in allen Bereichen unterscheiden. Der Konsolidierungspuffer kann die unterschiedlichen Ankunftszeiten kompensieren. Allerdings ist die Pufferkapazität beschränkt, da dies Auswirkungen auf den Flächenbedarf und die Investitionen (bei automatisierten Puffern) hat.

Um die notwendige Pufferkapazität zu reduzieren, kann die Durchlaufzeit harmonisiert werden. Neben der Auswahl des geeigneten Kommissionierverfahrens in der Planung kann die schwankende Auftragslast in den Zonen und die damit verbundene Variation der Durchlaufzeit durch eine geeignete Personaldisposition während des Betriebes ausgeglichen werden. Ein weiterer Ansatz ist die Entwicklung einer geeigneten Auftragsfreigabestrategie, die die erwartete Ankunftszeit der Teilaufträge im Konsolidierungspuffer angleicht und dabei die aktuelle Auftragssituation in den Zonen beachtet.

In diesem Paper werden Ergebnisse einer Simulationsstudie vorgestellt, mit der unterschiedliche Auftragsfreigabestrategien für ein neues Distributionszentrum eines Lebensmittel-Onlinehändlers untersucht wurden. Diese Untersuchung ist Teil einer Potenzialanalyse für einen digitalen Systemzwillings, der die operative Steuerung des Distributionszentrums unterstützen soll. Neben der Auftragsfreigabe soll dieser Digitale Zwillings auch die kurzfristige Personaleinsatzplanung und die Auswahl geeigneter Bereitstellbereiche für Waren übernehmen.

2 Stand der Technik

Zahlreiche wissenschaftliche Publikationen beschäftigen sich mit der Systemauswahl und Optimierung von Kommissioniersystemen [4]. Neben analytischen Modellen wird für die Untersuchung häufig auch die Simulation genutzt. [5,6]

In den Veröffentlichungen von Ulbrich und Venn wird der Einsatz der Simulation bei der Grobplanung von Kommissioniersystemen beschrieben [7,8]. Neben der Grobplanung wird vor allem bei (teil-)automatisierten Kommissioniersystemen vor der Realisierung eine Simulationsstudie zur Absicherung der Planung durchgeführt. [9] Die erstellten Simulationsmodelle werden in der Regel nicht genutzt, um die operative Planung zu unterstützen.

Zellerhoff stellt Ansätze für eine dynamische Anpassung von Kommissionierstrategien und der genutzten Ressourcen vor. Mit diesem Ansatz soll der spätere Betrieb unterstützt werden. [10] Auch Scholl setzt sich mit dem Thema der Personaleinsatzplanung in der Kommissionierung auseinander. [11] Die konzeptionellen Fragestellungen bei der Entwicklung eines Digitalen Zwillings für die operative Steuerung von Kommissioniersystemen präsentiert Kauke in seinem Beitrag. Allerdings werden keine konkreten Strategien aufgezeigt. [12]

Digitale Zwillinge für Intralogistiksysteme werden in verschiedenen Veröffentlichungen vorgestellt. Die Veröffentlichungen adressieren z.B. einen Auftragszuweisung für Gabelstapler mit einer integrierten Routenplanung. [13] Die operative Freigabe von Aufträgen in einem mehrzonigen Kommissioniersystem wird in keiner Veröffentlichung adressiert.

Die Auftragsfreigabe ist eine elementare Funktion eines Warehousemanagement-Systems (WMS). Die Systemanbieter bieten unterschiedliche Strategien für diese Funktion an. [14] Die dabei herangezogenen Informationen unterscheiden sich beginnend von Planwerten (wie z.B. eine geplante Auslieferzeit) bis hin zu einer Kalkulation der aktuellen Arbeitsbelastung in den einzelnen Bereichen. Die Auswahl einer geeigneten Strategie erfolgt bei der Einführung des WMS meist auf Grundlage von Einschätzungen oder qualitativer Bewertungen durch das Projektteam.

Die vorgestellte Simulationsstudie unterstützt die Entwicklung einer angepassten Auftragsfreigabe-Strategie für den Online-Lebensmittelhandel. Wobei die Erkenntnisse auf mehrzonige Kommissioniersysteme in anderen Anwendungsgebieten übertragen werden können.

Bezeichnung	Kommissionierverfahren	Hinweise
Gemüse	Kommissioniernest	Single-Order-Picking
Kühlbereich	Person-zur-Ware	Multi-Order-Picking
Trockensortiment	Person-zur-Ware	Multi-Order-Picking
Backwaren	Person-zur-Ware	Multi-Order-Picking

Tabelle 1: Übersicht der Kommissionierzonen

3 Rahmenbedingungen

Im Rahmen der Fallstudie werden unterschiedliche Freigabestrategien entwickelt und mit Hilfe eines geeigneten Simulationsmodells untersucht.

Das untersuchte Distributionszentrum besteht aus vier unterschiedlichen Kommissionierzonen, die in der Tabelle 1 aufgeführt sind. Die Positionen des Kundenauftrags (Artikel) können eindeutig einer dieser vier Zonen zugeordnet werden. Aus dem Kundenauftrag wird für jede Zone ein eigener Kommissionierauftrag abgeleitet. Die einzelnen Kommissionieraufträge werden für dezidierte Zonen zu Batches zusammengefasst und durch den Kommissionierer gemeinsam auf einer Tour durch die Zone abgearbeitet (Multi-Order-Picking). Dabei werden die Artikel direkt dem Auftrag(-sbehälter) zugeordnet. Am Ende stellt der Kommissionierer die Behälter auf eine Fördertechnik, die den Transport zum Konsolidierungspuffer übernimmt. Im Konsolidierungspuffer werden die Behälter eingelagert. Die Auslagerung erfolgt, wenn alle Aufträge einer Auslieferung (Fahrt) vollständig sind. Da an einer Packstation direkt die Rollgestelle für eine Auslieferungsfahrt vorbereitet werden, findet die Bearbeitung aller Aufträge einer Fahrt sukzessive am selben Packplatz statt. Wenn ein Packplatz eine Auslieferung vollständig bearbeitet hat, wird dem Platz eine neue Auslieferung zugewiesen und die Auslagerung der Behälter des ersten Auftrages aus dem Konsolidierungspuffer angestoßen. Der schematische Aufbau des Kommissioniersystems und der grobe Ablauf sind in Abbildung 1 dargestellt.

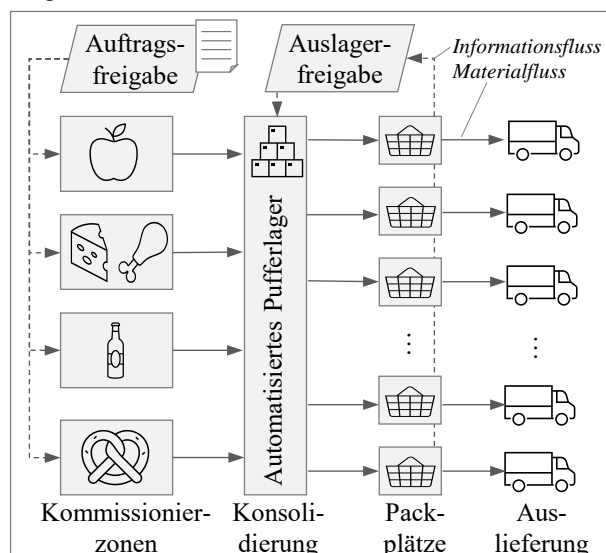


Abbildung 1: Schematischer Aufbau des Systems

Die Auslieferung der Aufträge erfolgt von Montag bis Samstag im Zeitraum von 6:00 bis 22:00 Uhr. Die Kommissionierung beginnt bereits um 3:00 Uhr mit der Bearbeitung der Aufträge und beendet die Arbeit spätestens um 20 Uhr. Die Konsolidierung beendet die Arbeit, wenn alle Aufträge im Puffer abgearbeitet wurden. Die Anzahl der Aufträge variiert zwischen den Wochentagen und den Auslieferzeitfenstern. Die genutzten Auftragsdaten wurden auf der Basis von Vergangenheitsdaten generiert. In der Tabelle 2 sind die Anzahl der Aufträge und deren Verteilung über den Tag aufgeführt.

Wochentag	Anzahl Aufträge	Morgens	Mittags/ Nachmittags	Abends
Mo	1.000	30%	30%	40%
Di	800	30%	30%	40%
Mi	700	30%	30%	40%
Do	900	30%	30%	40%
Fr	1.200	30%	40%	30%
Sa	900	50%	40%	10%

Tabelle 2: Verteilung der mittleren Auftragsanzahl über die Wochentage und Auslieferzeiträume

Die meisten Aufträge umfassen Positionen aus mindestens drei Zonen. Dabei ist die Anzahl der zu kommissionierenden Artikel in den einzelnen Zonen unterschiedlich. Auch die durchschnittlich bestellte Menge eines Artikels unterscheidet sich. Die durchschnittliche Anzahl von Auftragspositionen pro Zonen und die mittlere Entnahmemenge sind in Tabelle 3 zusammengefasst.

Bezeichnung	Aufteilung der Positionen eines Auftrages	Menge pro Position
Obst und Gemüse	35%	2 EE
Kühlbereich	25%	1,5 EE
Trockensortiment	30%	1 EE
Backwaren	10%	1,7 EE

Tabelle 3: Auftragseigenschaften, Verteilung der Auftragspositionen auf die Kommissionierzonen und mittlere Entnahmemenge (EE) pro Position

Der Lebensmittel-Onlinehändler möchte eine Termintreue von 95 Prozent sicherstellen. Mit Hilfe der Untersuchung soll eine sinnvolle Strategie für die

Auftragsfreigabe identifiziert werden. Weiterhin soll ermittelt werden, welche weiteren Aspekte die Termintreue beeinflussen und somit durch ein Steuerungssystem überwacht und hinsichtlich der aktuellen Anforderungen angepasst werden sollten.

4 Strategien für die Freigabe

In diesem Abschnitt sollen die untersuchten Freigabestrategien vorgestellt werden. Die Strategien wurden gemeinsam in Workshops ausgearbeitet. Dabei wurde die Umsetzbarkeit im Warehouse-Management-System (WMS) des Lebensmittelhändlers berücksichtigt. Bei allen entwickelten Strategien kann der früheste Freigabezeitpunkt festgelegt werden (z.B. maximal vier Stunden vor der gewünschten Lieferzeit). Zum Zeitpunkt der Freigabe muss bereits die Tourenplanung erfolgt und damit ein Kundenauftrag einer Ausliefertour zugeordnet sein. Um Expresslieferungen in den normalen Ablauf zu integrieren, soll die Tourplanung möglichst spät erfolgen. Für die Untersuchung wurden die folgenden vier Strategien identifiziert.

Standard-Release Time (SRT). Bei dieser Strategie werden alle Teilaufträge eines Auftrages zur gleichen Zeit für die Bearbeitung in den relevanten Zonen freigegeben. Der Freigabezeitpunkt kann über Parameter eingeschränkt werden. Hierbei wird für jede Zone eine maximale Anzahl an offenen Aufträgen im Auftragspuffer (freigegeben) festgelegt. Wenn für alle relevanten Zonen der Grenzwert nicht überschritten wird, erfolgt die Freigabe aller Teilaufträge. Andernfalls wird die Freigabe verzögert.

Fixer Offset (FO). Dem Betreiber des Kommissioniersystems ist in der Regel bekannt, welche Zone den Engpass bildet. Die Idee dieser Strategie ist es, dass zuerst der Auftrag für die Engpasszone freigegeben wird. In einem für jede Zone fest definierten Zeitabstand erfolgt auch die Freigabe der anderen Teilaufträge. Damit soll die notwendige Pufferkapazität reduziert werden. Im späteren Betrieb können die Offset-Parameter regelmäßig angepasst werden. (In dieser Studie wird keine Anpassung im Simulationszeitraum vorgenommen)

Process Trigger (PT). Die im Vorhinein festgelegte zeitliche Staffelung der Auftragsfreigabe kann nur bedingt die aktuelle Belastung des Kommissioniersystems abbilden. Bei dieser Strategie erfolgt zuerst die Freigabe für die Engpasszone. Wenn ein festgelegter

Prozessschritt (in der Studie der Start der Auftragsbearbeitung in der Engpasszone) begonnen wird, werden die anderen Teilaufträge für die Bearbeitung freigegeben. Es wird bei dieser Strategie unterstellt, dass eine der Hauptursachen für die unterschiedlich langen Durchlaufzeiten die Wartezeit (im Auftragspuffer der Zone) bis zum Start der eigentlichen Auftragsbearbeitung ist. Durch die Freigabe zum Bearbeitungsstart soll der Unterschied reduziert werden. In der Untersuchung triggert nur der Engpassprozess die anderen Teilaufträge. Es ist aber auch denkbar, dass mehrere Trigger eingesetzt werden können.

Estimated Time of Arrival (ETA). Die Idee für diese Strategie beruht auf der Annahme, dass die aktuelle Belastung der Zonen und damit die erwartete Durchlaufzeit berechnet werden kann. Zuerst wird für die bereits freigegebenen Aufträge die Planbearbeitungszeit bestimmt. Dabei werden die Auftragspositionen und die Entnahmemenge berücksichtigt. Die Planbearbeitungszeiten beruhen dabei auf Vorgabezeiten die mittels Methods-Time-Measurement (MTM) [15,16] analysiert wurden. Für die Zone mit der längsten zu erwartenden Durchlaufzeit erfolgt die Freigabe. Die erwartete Durchlaufzeit für diesen Auftrag bildet die Zielzeit für die anderen Aufträge. Ausgehend von der Zielzeit findet eine Rückwärtsterminierung statt und es wird ein spätester Freigabezeitpunkt für jeden Teilauftrag festgelegt. Die individuelle Freigabe der Teilaufträge erfolgt in einer definierten Zeitspanne vor dem spätesten Freigabezeitpunkt. (In der Untersuchung sind dies 10 Minuten)

Bei der Entwicklung der Strategien müssen verschiedene Zielstellungen beachtet werden. Aus Sicht der Termintreue wäre eine möglichst frühe Auftragsfreigabe für alle Zonen hilfreich. Dies führt unwiderruflich zu einem hohen Platzbedarf im Konsolidierungspuffer. Um die notwendige Kapazität zu reduzieren, kann eine dezidierte Freigabe der Teilaufträge helfen. Die gestaffelte Freigabe wirkt sich gegebenenfalls auf die Auslastung der einzelnen Zonen aus. Aus wirtschaftlichen Gründen müssen die Kommissionierer gut ausgelastet werden. Dies spricht wieder für eine möglichst frühe Auftragsfreigabe. Um auch bei einer gestaffelten Freigabe eine gute Auslastung der Kommissionierer zu ermöglichen, sollte auch die Mitarbeiterdisposition berücksichtigt werden. Dieser Aspekt soll in einer späteren Simulationsstudie untersucht werden.

5 Modell und Experimente

Für die simulative Untersuchung der Auftragsfreigabe werden nur der Kommissionier- und Verpackungsprozess betrachtet. Weitere Prozesse wie der Wareneingang, der Nachschub oder die Auslieferung an den Kunden werden nicht im Simulationsmodell abgebildet. Für die Untersuchung wurden verschiedene Bausteine (Klassen) in PlantSimulation entwickelt. Jede **Kommissionierzone** wird durch ein Netzwerk repräsentiert. Dieses Netzwerk kann über Parametertabellen angepasst werden. Neben der Ressourcenanzahl können auch Prozessparameter für jede Zone individuell festgelegt werden. Der eigentliche Kommissionierprozess ist abstrakt modelliert. Dies bedeutet, dass der Kommissionierprozess in Abhängigkeit der gerade zu bearbeitenden Aufträge und der verfügbaren Ressourcen als Zeitbedarf modelliert wird. Bei diesem Ansatz können keine Blockierungen und Behinderungen zwischen den Kommissionierern auftreten. Diese Aspekte wurden für diese Untersuchung als nicht relevant klassifiziert, da die Gassen ausreichend breit sind und die Anzahl der Mitarbeiter im Verhältnis zur Fläche als gering eingeschätzt wird.

Ein weiterer Baustein bildet den **Konsolidierungspuffer** ab. Neben der Abbildung der Ein- und Auslagervorgänge überwacht dieser Baustein die Vollständigkeit der Aufträge und Auslieferungen. Wenn ein Packplatz eine neue Auslieferung bearbeiten kann, wird diese anhand der Vollständigkeit und der Auslieferzeit ausgewählt sowie die Auslagerung angestoßen.

Die eigentliche Konsolidierung wird durch den **Packplatz**-Baustein dargestellt. Bei diesem Baustein wurde auf eine detaillierte Abbildung des Packprozesses verzichtet. Vielmehr wird der Zeitverbrauch in Abhängigkeit des Auftrages bestimmt und die Ressource für diesen Zeitraum belegt.

Neben den beschriebenen Bausteinen umfasst das Modell noch weitere Funktionen, die für die Auftragssteuerung, die Protokollierung von Kennwerten und die Steuerung der Experimentreihen notwendig sind. In der Abbildung 2 ist das Hauptnetzwerk des Modells zu erkennen. Das Modell besteht aus den vier Kommissionierzonen, einem Konsolidierungspuffer und 10 Packplätzen. Die Bereiche sind über Gebindefördertechnik miteinander verbunden.

Die Simulationsstudie kann in zwei Phasen eingeteilt werden. In der ersten Phase wird der benötigte Ressourceneinsatz in den einzelnen Bereichen des

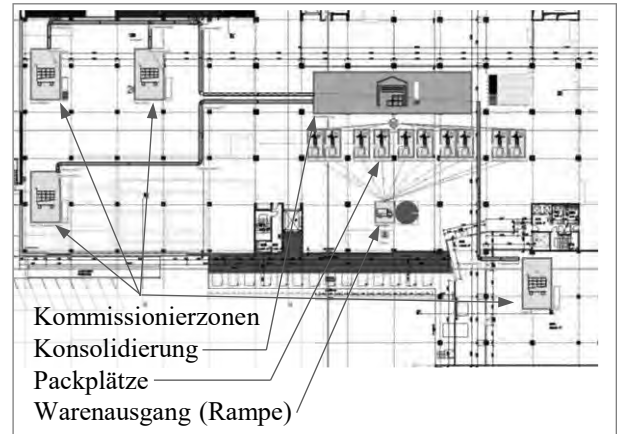
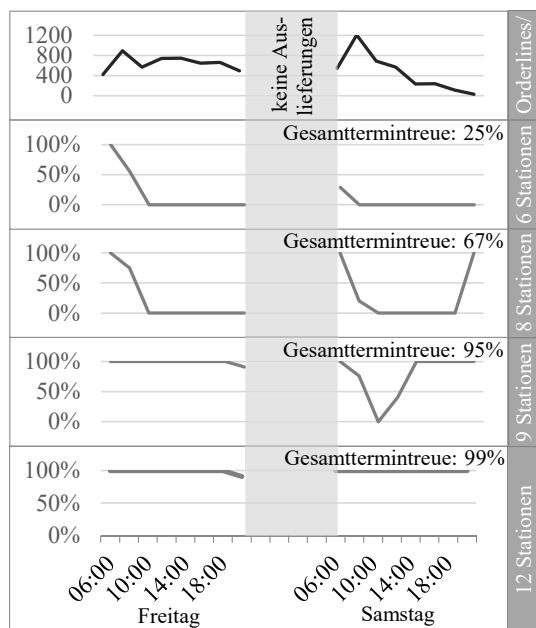


Abbildung 2: Hauptnetzwerk des Simulationsmodells in PlantSimulation

Systems untersucht. Die festgelegten Werte finden in allen weiteren Experimenten Anwendung. In der zweiten Phase stehen die vorgestellten Strategien im Mittelpunkt der Untersuchungen. Dazu wurden insgesamt 36 Experimente mit jeweils vier Replikationen durchgeführt, wobei die Replikationen unterschiedliche Auftragslisten nutzen. Für die Strategien „Estimated Time of Arrival“ und „Standard-Release Time“ wurde jeweils nur ein Experiment durchgeführt, da für diese Strategien keine Parameter festgelegt werden müssen. Für die Untersuchung der Strategie „Process Trigger“ wurde jede der vier Kommissionierzonen in einem Experiment als „Engpass“ festgelegt (Annahme) und triggert somit den Prozessstart in den anderen Zonen. Die verbleibenden 30 Experimente entfallen auf die Strategie „Fixer Offset“. Als Werte für den Offset wurden 0, 5, 10, 20 und 30 Minuten eingesetzt. Für die Untersuchung wurden 30 Kombinationen ausgewählt. Bei jeder dieser Kombinationen ist einer Zone der Wert null zugeordnet. Dies bedeutet, dass der entsprechende Teilauftrag in dieser Zone zuerst freigegeben wird (Engpasszone).

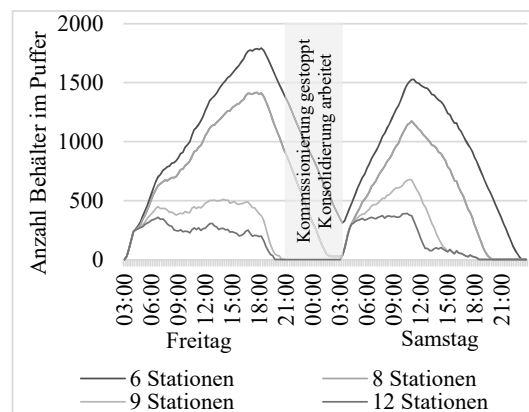
6 Ergebnisse

In den ersten Experimenten (1. Phase) wurden die Bearbeitungskapazitäten des Systems an die geforderte Systemlast angepasst. In dieser Fallstudie wird davon ausgegangen, dass die Kapazität während der Simulation konstant bleibt. Exemplarisch ist in der Abbildung 3 die Veränderung der Termintreue in Abhängigkeit der Anzahl an Packstationen dargestellt. Es ist zu erkennen, dass ab neun Packstationen die geforderte Termintreue von 95 Prozent eingehalten werden kann.


Abbildung 3: Verlauf der Termintreue

Die Simulationsergebnisse aus der ersten Phase haben gezeigt, dass die Packstationen am schlechtesten Schwankungen der Auftragslast kompensieren können. In Abbildung 3 ist dies durch einen kurzfristigen Einbruch der Termintreue am Samstag erkennbar. Vor allem Lastspitzen am Tagesanfang sind kritisch. Auch wenn die Aufträge bereits am Vortag bekannt sind, kann die Bearbeitung aufgrund der Anforderungen an die Kühlkette nicht vorgezogen werden. Um die Termintreue zu verbessern, muss kurzfristig die Anzahl „offener“ Packstationen erhöht werden.

Die Auswirkungen einer zu geringen Bearbeitungskapazität im Konsolidierungsbereich (Packstationen) wirken sich erwartungsgemäß auf den notwendigen Pufferplatz in der Konsolidierungszone aus. Der Pufferverlauf ist in der Abbildung 4 ersichtlich. Bei neun Stationen müssen maximal 680 Behälter gepuffert werden. Der Bestandsverlauf lässt die Schlussfolgerung zu, dass bei neun Stationen der Zufluss und Abfluss aus dem Konsolidierungspuffer annähernd gleich sind. Der Samstag stellt aus den bereits diskutierten Gründen eine Ausnahme dar. Die Kommissionierbereiche stellen die Arbeit ein, wenn alle Aufträge abgearbeitet wurden, spätestens aber um 20 Uhr. Die Konsolidierung muss aufgrund der bereits erwähnten Kühlkette alle Aufträge abarbeiten und beendet die Arbeit erst, wenn der Puffer leer ist.


Abbildung 4: Pufferverlauf für Freitag und Sonntag

In der ersten Phase der Simulationsstudie wurde für jeden Bereich eine wirtschaftlich sinnvolle Kapazität bestimmt. Im Mittel beträgt die Auslastung der Kommissionierer 72 bis 87 Prozent. Für die weiteren Experimente wird der in Tabelle 5 beschriebende Ressourceneinsatz zugrunde gelegt.

Bezeichnung	Anzahl Mitarbeiter/Stationen
Obst und Gemüse	7
Kühlbereich	5
Trockensortiment	7
Backwaren	3
Konsolidierung	9

Tabelle 5: eingesetzte Ressourcen nach Zonen

Die Tourplanung für die Auslieferung soll möglichst spät erfolgen. Die Bearbeitung im Distributionszentrum kann erst gestartet werden, wenn die Tourplanung erfolgt ist. Aus diesem Grund wurde der Einfluss des frühestmöglichen Auftragsfreigabezeitpunkts untersucht. Hierbei wurde der Zeitraum für die Freigabe zwischen einer Stunde und sechs Stunden variiert. Die Ergebnisse, welche in der Abbildung 5 dargestellt sind, zeigen, dass die Freigabe mindestens vier Stunden vor der Auslieferung erfolgen muss. Andernfalls muss die Anzahl der Ressourcen in allen Bereichen erhöht werden.

Eine frühere Freigabe der Aufträge erlaubt eine bessere Kompensation von Lastspitzen. Der Lastspitze am Samstag kann aber auch nicht mit einer Freigabe von 6 Stunden vor Auslieferung begegnet werden, da die Spitze am Tagesanfang auftritt und eine Bearbeitung am Vortag einen noch größeren Vorlauf nötig macht.

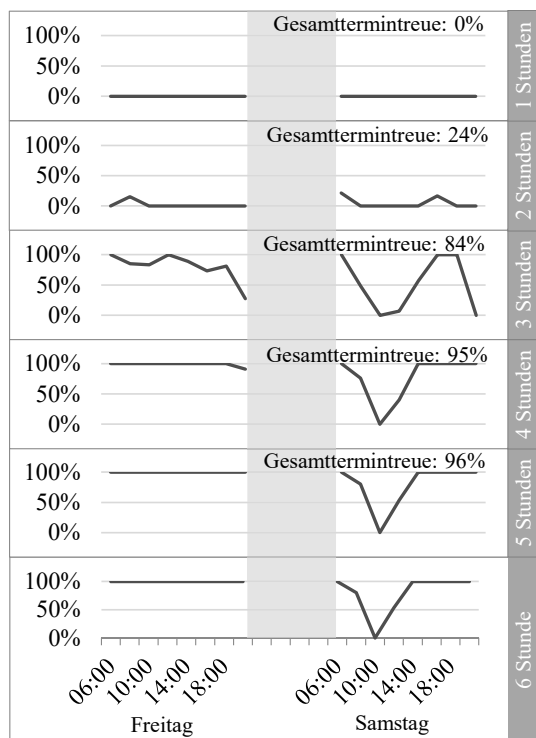


Abbildung 5: Verlauf der Termintreue in Abhängigkeit des frühesten Freigabezeitpunktes

Für die weiteren Untersuchungen wird der früheste Freigabezeitpunkt auf 4 Stunden vor Auslieferung festgelegt. Neben der erreichbaren Termintreue sind die Anforderungen an die Kühlkette und Tourplanung dafür entscheidend.

Die ersten Experimente haben gezeigt, dass die Packstationen einen Engpass darstellen. Permanent mehr als neun Stationen mit Mitarbeitern zu besetzen, ist nicht wirtschaftlich. In diesem Bereich muss der Kapazitätsbedarf dynamisch angepasst werden. Da in dieser Studie die Auswirkungen der Freigabestrategie untersucht werden sollen, wird das Potenzial der Strategien anhand der Ankunftszeit im Konsolidierungspuffer bewertet. Dadurch werden die Ergebnisse nicht durch zeitweilige Engpässe im Packbereich beeinflusst. Als Vergleichswert wird die Zeitspanne zwischen der Ankunft des ersten und des letzten Teilauftrages im Puffer herangezogen. Exemplarisch ist für die Strategie ETA in der Abbildung 6 ein Histogramm abgebildet, welches die Verteilung der Zeitspannen zwischen der Ankunft des ersten und letzten Teilauftrages darstellt.

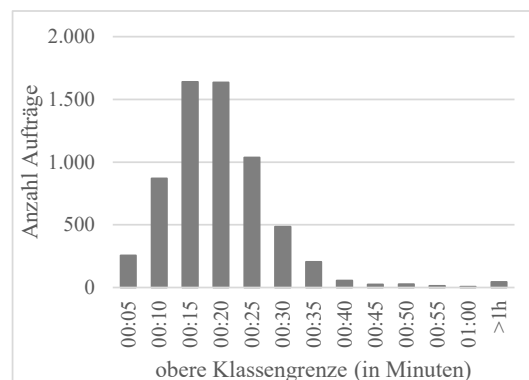


Abbildung 6: Verlauf der Termintreue in Abhängigkeit des frühesten Freigabezeitpunktes.

Für eine bessere Vergleichbarkeit der Experimentergebnisse wird für jedes Experiment die mittlere Zeitspanne und Standardabweichung bestimmt. Die Ergebnisse der relevanten Experimente sind in der Tabelle 6 aufgeführt. In den Spalten „Obst“, „Kühlung“, „Trocken“ und „Bäckerei“ sind die eingesetzten Parameter für die jeweilige Zone ersichtlich. Die Tabelle zeigt nur die relevanten Experimente.

Strategie	Obst	Kühlung	Trocken	Bäckerei	Mittlere Zeitspanne	Standardabweichung	max. Pufferbestand
SRT					0:35:18	0:04:11	1183
FO	E	10	10	10	0:38:11	0:03:15	1078
FO	E	20	5	20	0:32:41	0:03:05	988
FO	E	30	5	30	0:32:18	0:03:01	976
FO	10	10	E	10	0:24:25	0:02:28	799
FO	10	20	E	20	0:24:48	0:02:14	805
FO	10	30	E	30	0:24:51	0:02:19	827
FO	20	20	E	20	0:23:01	0:02:04	764
FO	20	30	E	30	0:22:51	0:01:48	735
PT	T				0:24:42	0:01:59	812
PT			T		0:20:02	0:01:35	720
ETA					0:19:48	0:01:07	674

Tabelle 6: Ergebnisse der relevanten Experimente zur Untersuchung der Strategien. (E-Engpasszone, T: Triggerzone, Angabe des Offsets in Minuten)

Die Ergebnisse der Studie haben gezeigt, dass die Strategie ETA am besten auf die aktuelle Systemlast reagiert. Ähnlich gute Ergebnisse zeigt die Strategie PT.

Größere Unterschiede in den Zeitspannen konnten bei der Strategie FO beobachtet werden. Die mittlere Zeitspanne ist bei gut gewählten Parametern ähnlich zu den vorher genannten Strategien. Wenn alle Teilaufträge zur gleichen Zeit freigegeben werden (SRT), führt dies zu höheren Zeitspannen zwischen der Ankunft des ersten und des letzten Teilauftrages und somit zu einem höheren maximalen Pufferbestand.

Die fallbasierte Simulationsstudie hat gezeigt, dass die Kapazität der einzelnen Kommissionierbereiche und vor allem die Leistungsfähigkeit der Konsolidierung wesentlich die Termintreue und auch die Puffergröße beeinflussen. In einem Kommissioniersystem können sich Engpassbereiche verschieben, da die Arbeitslast in den Bereichen variiert. Aus diesem Grund ist es notwendig, dass neben der Wahl der richtigen Auftragsfreigabestrategie auch die Bearbeitungskapazität der Zonen überwacht und bedarfsgerecht angepasst werden kann. Wie zu Beginn des Beitrages aufgezeigt, sollen Strategien für die dynamische Personaleinsatzplanung in weiterführenden Untersuchungen analysiert werden.

7 Zusammenfassung

Die Untersuchung der Freigabestrategien für die Teilaufträge in einem mehrzonigen Kommissioniersystem hat gezeigt, dass die sogenannte „Estimated time of arrival“-Strategie für das Fallbeispiel die besten Kennwerte lieferte und hinsichtlich einer Umsetzung weiter detailliert werden soll. Durch die Simulationsstudie kristallisierte sich heraus, dass die Freigabestrategie nicht autark betrachtet werden kann. Vielmehr muss auch die Einsatzplanung für die Mitarbeiter durch die Steuerungsstrategien erfolgen. Durch diese ganzheitliche Betrachtung kann das System ausballanciert werden. Aus wirtschaftlichen Gesichtspunkten ist der Einsatz der Ressourcen beschränkt. Vor allem bei Lastspitzen müssen Strategien Anwendung finden, die die Last nivellieren. Da die Einhaltung der Kühlkette kein uneingeschränktes Vorziehen der Auftragsbearbeitung ermöglicht, müssen andere Maßnahmen ergriffen werden. Zum Beispiel kann über das Marketing für den Kunde bei der Wahl des Liefertermins beeinflusst werden.

In weiterführenden Untersuchungen sollen das Thema Personaldisposition untersucht und die Strategien für die Umsetzung eines Steuerungssystems mit integrierter Prozesssimulation (Digitaler Zwilling) identifiziert werden.

References

- [1] VDI Richtlinie 3590. Kommissionierung. Berlin: Beuth Verlag; 1994.
- [2] Thoma K. Akzeptanz des Online-Lebensmittelhandels. Hof: HSH; 2019.
- [3] Warschun, M., Limper-Menapace, A., Neumann, F., Pins, A. Online-Handel mit Lebensmitteln: Fokus als Schlüssel zum Erfolg. AT Kearney. Internet ohne Orstange. Aufruf 20.03.2022.
- [4] De Koster, R., Le-Duc, T., Roodbergen, K.J. Design and control of warehouse order picking: A literature review. *European journal of operational research* 182.2; 2007.
- [5] Bučková, M., Krajčovič, M., Edl, M. Computer simulation and optimization of transport distances of order picking processes. *Procedia engineering* 192 (2017).
- [6] Klodowski, M., et al. Simulation analysis of order picking efficiency with congestion situations. *International Journal of Simulation Modelling* 17.3, 2018: 431-443.
- [7] Ulbrich, A. S.: Simulationsgestützte Grobplanung von Kommissioniersystemen. München: fml; 2010.
- [8] Venn, E.: Beitrag zur simulationsgestützten Konzeptplanung von heterogen strukturierten Kommissioniersystemen. Dortmund: Verl. Praxiswissen, 2011.
- [9] Jaghbeer, Y., Hanson, R., Johansson, M.I. Automated order picking systems and the links between design and performance: a systematic literature review. *International Journal of Production Research* 58.15 2020: 4489-4505.
- [10] Ten Hompe, M.; Zellerhoff, J. Simulation zur Bestimmung des Leistungsgewinns bei kombiniertem Einsatz mehrerer Kommissionierstrategien – in Abhängigkeit von Layout und Auftragslast. In *Integrationsaspekte der Simulation: Technik, Organisation und Personal*. Karlsruhe: KIT Scientific Publishing, 2010
- [11] Scholl, P.; Deuse, J.: Optimale Personalstrategie in der Kommissionierung-Dynamische Entscheidungsmodell zur kostenoptimalen Kommissionierung bei hohen Absatzschwankungen. In: *wt online*, 2010.
- [12] Kauke, D., Galka, S., Fottner, J. Digital Twins in Order Picking Systems for Operational Decision Support. 54th Hawaii International Conference on System Sciences, 2021 Hawaii, USA, 1655 – 1664
- [13] Braglia, R., Gabbriellini, M., Frosolini, L., Marrazzini Padellini. Using rfid technology and discrete-events, agent-based simulation tools to build digital-twins of large warehouses. *IEEE International Conference on RFID Technology and Applications*. 2019; 464-469
- [14] Ten Hompe, M., Schmidt, T. Warehouse management. Springer Berlin, 2008.
- [15] MTM -Institut: Schulungsunterlagen Standardvorgänge Logistik, Zeuthen, 2009.
- [16] Galka, S., Günthner, W.A. Planung und Auswahl von Kommissioniersystemen. In *MTM in einer globalisierten Wirtschaft*. Berlin: mi-Verlag. 2013; 201-220

Selbstoptimiertes Auftragsmanagement für autonome Transportfahrzeuge

Jie Zhang^{1*}, Xiaobo Liu-Henke¹, Thomas Vietor²

¹Institut für Mechatronik Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel, Deutschland; *jie.zhang@ostfalia.de

²Institut für Konstruktionstechnik, Technische Universität Braunschweig, Hermann-Blenk- Str. 42, 38108 Braunschweig, Deutschland

Abstract. Der folgende Beitrag beschreibt eine Assistenzfunktion zur Unterstützung des Auftragsmanagementsystems bei der Verwaltung der Transportaufträge (TA) auf Basis von genetischem Algorithmus (GA), so dass die TA unter Berücksichtigung von deren Priorität und vordefinierten Deadline klassifiziert und in einer bestimmten Reihenfolge zu einem neuen Auftrag Transportfahrzeug (FTF) umgruppiert werden. Das Ziel ist, die Intralogistik in der Situation von mangelnder Transportleistung mittels FTF im Vorfeld vor der Ausführung des TA zu optimieren, insbesondere die Transportverzögerung zu minimieren und damit die Pünktlichkeitsrate des Gütertransportes zu erhöhen.

Einleitung

Aufgrund des Entwicklungstrends der zukünftigen Produktion nach kundenspezifischen Anforderungen und dem Einsatz von I4.0-Lösungen zur Optimierung der Prozesse eines Produktionsunternehmens, ist die weitere Erhöhung des Automatisierungsgrades der Produktionsakteure, insbesondere Produktionsmaschine zur Produktfertigung und FTF in der Intralogistik, erforderlich, um die Flexibilität des gesamten Produktionssystems zu

gewährleisten [1, 2, 3]. Um die Ziele zu erreichen und damit die Anforderungen an die zukünftige kundenindividuelle Produktion zu erfüllen, wird an der Ostfalia ein autonomes FTF mit Mecanumrädern entwickelt, welche gegenüber des konventionellen FTF über IoT-basierte Kommunikationstechnologie mit anderen Produktionsakteuren vernetzen und anhand der ausgetauschten Informationen selbst Entscheidungen treffen ohne Eingriff von Bedienern [4].

Im vorliegenden Beitrag wird eine Assistenzfunktion auf Basis von GA zur Klassifizierung und Umgruppierung der TA konzipiert und entwickelt, welche beim autonomen Gütertransport mittels des dezentralen Auftragsmanagementsystems von FTF unterstützt. Dadurch lassen sich mehrere Ziele der Intralogistik gleichzeitig in der Situation von mangelnden Transportleistung mittels FTF optimieren.

1 Stand des Wissens

1.1 Dezentrales Auftragsmanagementsystem von autonomen FTF

Wie bereits erwähnt, um die Flexibilität und den Auto-

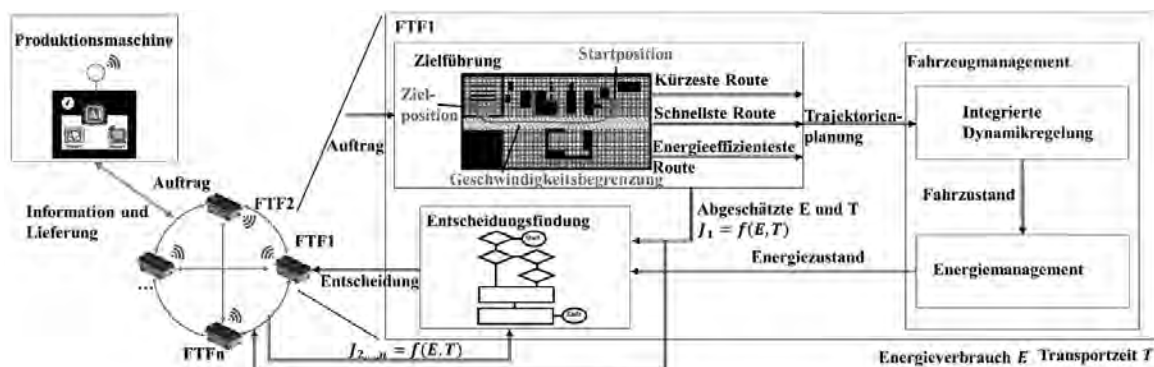


Abbildung 1: Konzept des dezentralen Auftragsmanagementsystems für autonome FTF

matisierungsgrad des FTF zu erhöhen sowie damit den autonomen Gütertransport zu realisieren, werden die Leitsteuerung der konventionellen FTF dezentralisiert [5] und die FTF mit IoT-basierter Kommunikationstechnologie, intelligenter Hard- und Software ausgebaut. An der Ostfalia werden autonome FTF entworfen und das dezentrale Auftragsmanagementsystem zur Vernetzung der FTF mit der Produktionslinie entwickelt.

Abbildung 1 stellt das Konzept des dezentralen Auftragsmanagementsystems und auch die Funktionsstruktur der autonomen FTF vor [6]. Alle Produktionsakteure, z.B. Produktionsmaschinen FTF, in einer Produktionsanlage vernetzen sich direkt miteinander mittels IoT-basierter Kommunikationstechnologie und mithilfe der eigenen Intelligenz treffen selbständig Entscheidungen zur Durchführung der Produktion und Transportabwicklung. Anhand der TA, die von den Produktionsmaschinen erstellt und anschließend über Kommunikationsmodule die FTF mitgeteilt werden, können die FTF eigenständig eine kürzeste Route zwischen der aktuellen Position und Zielposition ohne Konflikt mit den stationären Hindernissen ermittelt und dann eine Trajektorie unter Berücksichtigung des dynamischen und kinematischen Verhaltens des FTF berechnet. Mit der abgeschätzten Transportzeit und Energiekosten der TA wird die Fahrkosten J mithilfe der künstlichen Intelligenz bestimmt und die FTF konkurrieren sich miteinander, um TA zu gewinnen. Das heißt, die FTF vergleichen die Fahrkosten durch das Kommunikationsmodul miteinander zu einer gemeinsamen Entscheidungsfindung, welches FTF diesen TA übernehmen und durchführen soll. Das dezentrale Auftragsmanagement und die jeweiligen Fahrfunktionen des FTF werden modellbasiert entwickelt und deren Funktionalität in einer realitätsnahen Simulationsplattform abgesichert [7].

1.2 Genetischer Algorithmus

Genetischer Algorithmus (GA) ist in den letzten Jahrzehnten weit verbreitet zur Lösung der Optimierungsprobleme, welcher nicht-deterministische stochastische Suchmethode ist und nach der Evolutionsprinzip „Survival of the fittest“ von Charles Darwin entwickelt wird. GA generiert Populationen der Individuen und damit die gute Genfragmente mittels genetischen Evolutionsprozesses bzw. Selektion, Kreuzung und Mutation beibehalten, sodass die Lösungsqualität fortlaufend verbessert wird und es zu einer optimalen Lösung schnell konvergiert, ohne jede mögliche Lösung zu untersuchen.

Das grundlegende Konzept von GA ist, dass die Stärkeren dazu neigen, sich anzupassen und zu überleben, während die Schwächeren zum Aussterben neigen. Das heißt, die Optimierung basiert auf der Evolution und dem Konzept des Überlebens der Stärkeren im iterativen Evolutionsprozess. GA hat die Fähigkeit, eine anfängliche Population realisierbarer Lösungen zu erstellen und diese dann so zu rekombinieren, dass ihre Suche nur auf die vielversprechendsten Bereiche des Lösungsraums gelenkt wird.

Abbildung 2 verdeutlicht den grundlegenden Ablauf des GA. Nachdem das zu optimierende Problem unter Berücksichtigung des Anwendungsfalls individuell kodiert wird, wird eine Population von Individuen bzw. den möglichen Lösungskandidaten des Optimierungsproblems zufällig erzeugt und initialisiert, die von der Generation 0 genannt wird. Um besser angepasste Individuen in einer neuen Generation zu erstellen, wird die Güte jedes Individuums mithilfe einer speziell für die Aufgabenstellung entworfenen Fitnessfunktion in der Anwendung bestimmt. Die Fitness eines Individuums wird als ein Maß der Fähigkeit zum Überleben und zur Produktion der Nachkommen bezeichnet. Durch die genetische Manipulation nach probabilistischen Regeln, bzw. Selektion, Kreuzung und Mutation evolviert sich die Individuen, gemessen an der Fitnessfunktion, in die nächste Generation. Der genetische Prozess wiederholt sich, bis sich ein Individuum mit den gewünschten Eigenschaften entwickelt hat oder die vordefinierte Abbruchbedingung erfüllt ist. Das Konzept und der Entwicklungsprozess der Assistenzfunktion auf Basis von GA wird in Abschnitt 3 und 4 detailliert dargestellt.

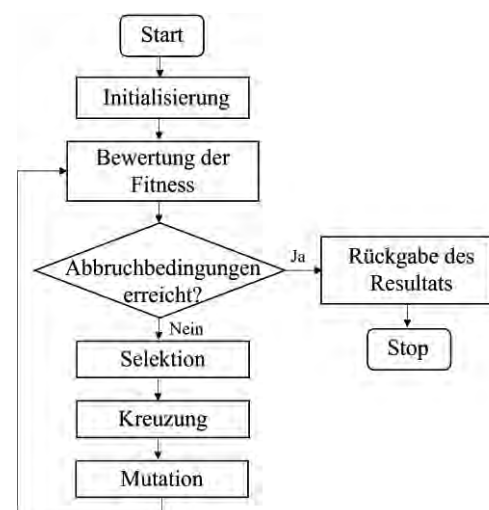


Abbildung 2: Entwicklungsablauf von GA [8]

2 Methodik

Die durchgängig modellbasierte und verifikationsorientierte Funktionsauslegung und -absicherung vernetzter mechatronischer Systeme nach [9] hat sich in zahlreichen Anwendungen in Forschung und Industrie als zeit- und kosteneffizient erwiesen.

Abbildung 3 illustriert den modellbasierten mechatronischen Entwicklungskreislauf, welcher zur Absicherung Model-in-the-Loop- (MiL-), Software-in-the-Loop- (SiL-) und Hardware-in-the-Loop- (HiL-) Simulationen sowie die Echtzeiteralisierung durch Prototypen beinhaltet. Basierend auf den Anforderungen, welche durch die systematische Betrachtung des Zusammenspiels dessen Komponenten aus verschiedenen Domänen abgeleitet und konkretisiert werden [10, 11], erfolgen bereits in frühen Entwicklungsphasen Optimierungen, Tests und Verifikationen. In MiL-Simulationen werden dabei Funktionsmodelle gegen komplexe Modelle der Umgebung, in einer realitätsnahen Simulationsumgebung getestet. Ist ein hinreichender Funktionsstand erreicht, erfolgt die automatische Generierung von C-Code, welcher in SiL-Simulationen mit Hilfe des Streckenmodells verifiziert wird. Unter Echtzeitbedingungen erfolgen schließlich weitere Optimierungen und Verifikationen mittels HiL-Simulationen unter Verwendung echtzeitfähiger Streckenmodelle, welche durch reale Komponenten ergänzt werden. Letztendlich erfolgen Feldtests unter echtzeitigen Bedingungen durch den Einsatz von Prototypen.

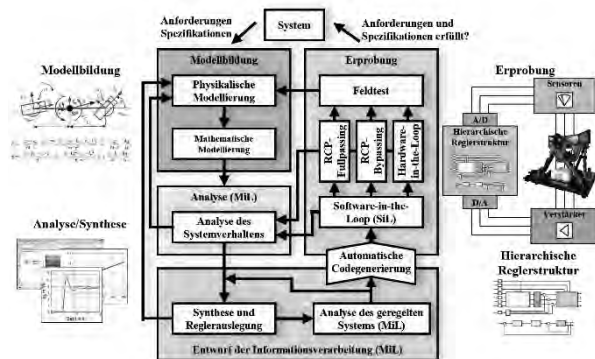


Abbildung 3: Mechatronischer Entwicklungskreislauf

Mithilfe des mechatronischen Entwicklungskreislaufes wird eine durchgängige Funktionsentwicklung und deren Integration bis zum Gesamtsystem sichergestellt. Dieser Beitrag stellt aus der Methodik schwerpunktmäßig die Entwicklung und Verifikation der Assistenzfunktion MiL-Simulation dar.

3 Konzeption der Assistenzfunktion zur Unterstützung des dezentralen Auftragsmanagementsystems

Dieser Abschnitt stellt die Anforderungen an die Assistenzfunktion zur Unterstützung des dezentralen Auftragsmanagementsystems bei der Klassifikation und Umgruppierung der TA in einer Situation von mangelnder Transportleistung vor und beschreibt das Konzept der Assistenzfunktion.

3.1 Problemstellung und Anforderungen

Die Funktionsweise des dezentralen Auftragsmanagementsystems in einer Situation von genügender Transportleistung mittels autonomen FTF, wobei FTF die TA einer Produktionsanlage termingerecht ausführen können, wird erfolgreich getestet und in [6] aufgezeigt. Aber die weiter erhöhte Nachfrage an die Transportleistung mit einer bestimmten Anzahl von FTF führt zur Verzögerung der Ausführung der TA und sogar zum Stillstand der Produktion infolge der mangelnden Materialien am Eingang der Produktionsmaschinen. Während das Problem in Mangel an Transportleistung sich durch Einsatz neuen FTF leicht überwinden lassen kann, erhöht dies jedoch die Ausrüstungsinvestitionen. Es wäre besser, dass die TA, dessen Datenstruktur wie in der Abbildung 4 dargestellt wird, vor Ausführung durch FTF unter Berücksichtigung von der Priorität und vordefinierten Deadline klassifiziert und umgruppiert werden. Damit wird die Priorität der TA nach der entsprechenden Deadline dynamisch angepasst und die Reihenfolge zur Bearbeitung der TA im Voraus optimiert. Außerdem liegen die weiteren Vorteile insbesondere darin, dass die Verzögerung des Gütertransportes reduziert und sogar vermieden ohne Einsatz von neuen FTF werden kann.

Anhand der definierten Ziele werden die folgenden wesentlichen Anforderungen mittels des modellbasierten Anforderungsmanagements anhand [11] an die Assistenzfunktion abgeleitet und definiert:

- [R1] Dynamische Anpassung der Priorität der zu bearbeitenden TA in der Wartelist.
- [R2] Festlegung einer optimalen Reihenfolge der abzuwickelnden TA
- [R3] Integration ins dezentrale Auftragsmanagementsystem.

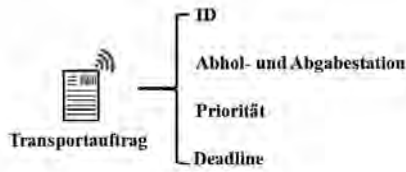


Abbildung 4: Datenstruktur eines TA

3.2 Entwurf der Assistenzfunktion

Nach dem Informationsfluss des dezentralen Auftragsmanagementsystems und der definierten Anforderungen werden für die Assistenzfunktion die folgenden Funktionsmodule konzipiert, deren Entwicklungsprozess im Abschnitt 4 detailliert wird:

- [F1] Dynamische Anpassung der Priorität von TA: da in der Situation mit mangelnder Transportleistung die TA nicht zeitnahe bearbeitet werden kann und damit es zu einer großen Ansammlung von TA infolge Blockaden führt, soll die Priorität der TA dynamisch über die Zeit angepasst und dann die TA mit gleicher Priorität in einer Gruppe zusammengesammelt werden. Hierdurch werden die TA mit hoher Priorität vorrangig ausgeführt und somit lässt sich die Verzögerung des Gütertransportes minimieren. Die neu priorisierten TA werden in einer Warteliste gesammelt und stellen als Eingangsdaten für Klassifizierung und Umgruppierung zur Verfügung.
- [F2] Klassifikation und Umgruppierung der TA: die TA in einer Gruppe sollen anhand der Auftragsinformationen z.B. die Abhol- und Abgabestation klassifiziert und sich miteinander in einer bestimmten Reihenfolge als einen neuen TA für FTF verknüpfen, sodass die Umwege des FTF aufgrund der unterschiedlichen Abhol- und Abgabestationen von TA minimiert und die Intralogistik sowie der Produktionsprozess beschleunigt werden kann.

4 Entwicklung der Assistenzfunktion mittels GA

Die im Abschnitt 3 konzipierte Assistenzfunktion soll modellbasiert entwickelt werden, deren Entwicklungsprozess wie im Folgenden detailliert beschrieben wird.

4.1 Initialisierung und Kodierung

Um das Optimierungsproblem mittels GA zu lösen bzw. die optimale Klassifizierung und Umgruppierung der TA zu ermöglichen, wird das Optimierungsproblem zuerst kodiert. Entlehnt von der Genetik wird folgendes

Vokabular speziell für das Optimierungsproblem definiert und initialisiert:

- Individuum S : Jedes Individuum wird als eine mögliche Lösung des Optimierungsproblems bzw. eine optimale Umgruppierung der TA für FTF in einer bestimmten Reihenfolge bezeichnet.
- Länge eines Individuums n : Hierbei bezieht sich die Länge n auf die Anzahl der zu bearbeitenden TA.
- Gen: Eine bestimmte Stelle bzw. Sequenz eines Chromosoms wird als Gen bezeichnet. Unter Berücksichtigung der Datenstruktur eines TA wird in diesem Beitrag das TA-ID als Gen verwendet.
- Population i und Generation t : Eine Menge von Individuen, die in einem bestimmten Evolutionsprozess entstehen, wird als eine Population und die Population als Generation bezeichnet.

Durch Zuweisung der entsprechenden Werten zu Genen eines Individuums wird das Optimierungsproblem kodiert. Die Abbildung 5 stellt ein kodiertes Individuum S_i^t in einer Generation t vor, welches als eine Reihe des ID von n TA in bestimmter Reihenfolge dargestellt wird.



Abbildung 5: Kodierung des Optimierungsproblems

4.2 Fitnessfunktion und Abbruchbedingung

Durch die Fitnessfunktion wird die Güte eines Individuums bezüglich der zu optimierenden Aufgabe bewertet und quantifiziert. Zur Festlegung der optimalen Reihenfolge von TA werden die folgenden Zielkriterien mit entsprechenden Kennwerten definiert, durch welche die möglichen Lösungen bzw. die Individuen S_i^t einer Generation quantitativ bewertet:

- Transportzeit $t(S_i^t)$ von FTF zur Ausführung S_i^t
- gesamte Länge der Fahrwege $l(S_i^t)$ von FTF zur Ausführung S_i^t
- Pünktlichkeitsrate des Gütertransportes $u(S_i^t)$

$$u(S_i^t) = \frac{n_p(S_i^t)}{n_{all}(S_i^t)} \quad (1)$$

n_p : Anzahl der pünktlich ausgeführten TA

n_{all} : Anzahl der gesamten originalen TA

- gesamte Verzögerungszeit des Gütertransportes t_{gvz} :

$$t_{gvz}(S_i^t) = \sum_{j=1}^{n_{all}-n_p} t_{vz}(S_i^t) \quad (2)$$

Weil die Zielkriterien unterschiedlich skaliert werden, sollen die Kennwerte der entsprechenden Zielkriterien mit einer geeigneten Skalierung normiert werden und somit die können miteinander verglichen werden. Deren Normierung wird, beispielsweise für den Kennwert Transportzeit $t(S_i^t)$, wie folgend aufgezeigt:

$$t_\alpha = \frac{t_{\max} - t(S_i^t)}{t_{\max} - t_{\min}} \quad (3)$$

$$t_{\max} = \max(t(S_1^t), t(S_2^t), \dots, t(S_n^t))$$

$$t_{\min} = \min(t(S_1^t), t(S_2^t), \dots, t(S_n^t))$$

Um die erwähnten Zielkriterien gleichzeitig zu optimieren, wird die folgende Fitnessfunktion $F(S_i^t)$ für jeweiliges Individuum S_i^t formuliert, bei der die Gewichtungsfaktoren bzw. a, b, c und d für unterschiedene Zielkriterien unter Berücksichtigung von dem Anwendungsfall individuell ermittelt wird:

$$F(S_i^t) = a \cdot t_\alpha + b \cdot l_\alpha + c \cdot u_\alpha + d \cdot t_{gvz_\alpha} \quad (4)$$

Wie bereits erwähnt, wiederholt sich der iterative Verlauf des GA, bis sich ein Individuum mit den gewünschten Eigenschaften entwickelt hat oder die vordefinierte Abbruchbedingung erfüllt ist. Zur Begrenzung der Rechenkosten wird hierbei die Abbruchbedingung verwendet. D.h., die Iterationsschleife bricht ab, wenn sie eine vorgegebene Anzahl der Generation erreicht.

4.3 Selektion, Kreuzung und Mutation

Nach dem GA entwickeln sich die neuen Individuen (Kinder) in der Generation $t + 1$ von den Individuen (Eltern) in der vorherigen Generation t . Um die Gesamtfitness der Population ständig zu verbessern, werden anhand genetischer Manipulation nach probabilistischen Regeln die besten Individuen in der Population mithilfe der Selektion während des iterativen Verlaufs behalten und die schwachen Individuen mit niedriger Fitness verworfen. D.h., die Individuen mit größerer Fitness haben höhere Wahrscheinlichkeit zur Rekombination der neuen Individuen. In diesem Beitrag wird die fitnessproportionale Selektionsmethode durch Wettkampfselektion durchgeführt, um die Vielfalt der Individuen einer Generation zu gewährleisten und die vorzeitige Konvergenz in der frühen Phase des GA zu vermeiden. Nach der Wettkampfselektion werden n Individuen zufällig aus der Generation als eine kleine Gruppe ausgewählt und jedes Individuum hat die gleiche Wahrscheinlichkeit zur Rekombination. Anschließend wird das Individuum mit der höchsten Fitness von der Gruppe zu Rekombination als Eltern-Individuum bestimmt.

Dieser Vorgang wird fortgesetzt, bis die erforderliche Anzahl der Individuen als Eltern erreichen.

Durch Kreuzung (Abbildung 6) werden zwei neue Individuen S_i^{t+1} und S_j^{t+1} in Generation $t + 1$ aus zwei ausgewählten Individuen S_i^t und S_j^t der Generation t durch Austausch der Gene erzeugt. Die Eltern-Individuen werden zufällig ausgewählt, was durch die Kreuzungsrate P_c repräsentiert. In dieser Arbeit wird die sogenannte Zwei-Punkt-Kreuzung-Methode verwendet. Hierbei werden die beiden Positionen für Kreuzung zufällig ermittelt und dann die ausgewählte Genfragmente der Eltern-Individuen miteinander ausgetauscht, sodass die beiden Kinder-Individuen sich generieren.

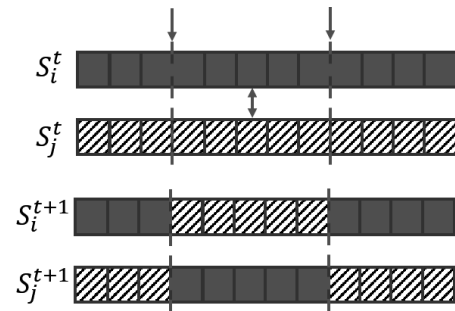


Abbildung 6: Zwei-Punkt-Kreuzung

Durch Mutation wird die zusätzliche Variabilität von Individuen in einer Generation initiiert, was durch die passende Einstellung der Mutationsrate P_m erfolgt. In Analogie zu der Genetik, da die Mutation in der Natur sehr selten passiert, soll die Mutationsrate gegenüber der Kreuzungsrate einen kleineren Wert genommen werden. Abbildung 7. stellt die in diesem Beitrag genutzte Mutationsoperation dar, wobei ein neues Individuum $S_{i_new}^{t+1}$ durch Umkehrung der Gene eines zufällig ausgewählten Mutationsbereiches auf Individuum S_i^{t+1} erstellt wird.

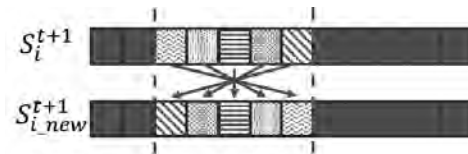


Abbildung 7: Mutationsoperation

4.4 Elitismus Strategie

Im GA ersetzen die Nachkommen der neuen Generation $t + 1$ nach dem genetischen Evolutionsprozess die gesamte Population der vorherigen Generation t . Aber es ergibt sich jedoch die Gefahr, dass die besten Individuen durch zufällige Kreuzung und Mutation sich ver-

lieren, was zur Verschlechterung der Individuenqualität und Verlangsamung der Konvergenz des Optimierungsproblems führt. Um das erwähnte Risiko zu vermeiden, wird die sogenannte Elitismus-Strategie in den genetischen Entwicklungsprozess eingesetzt, wobei die schlechtesten Nachkommen mit kleinster Fitness der aktuellen Generation durch die besten Individuen mit größter Fitness der vorherigen Generation ersetzt werden sollen, wie in der Abbildung 8 veranschaulicht [12].

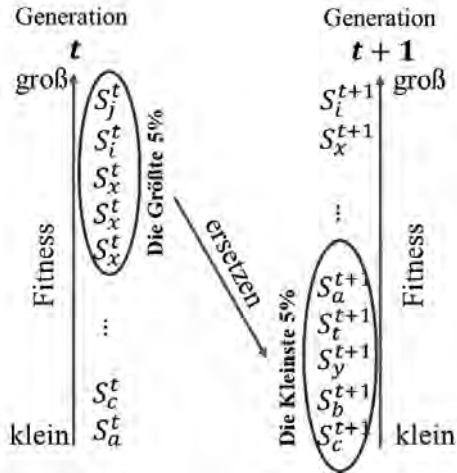


Abbildung 8: Elitismus Strategie

4.5 Dynamische Kreuzung- und Mutationsrate

Aufgrund der Konvergenz von GA und der eingesetzten Elitismus-Strategie verliert sich die Population die Diversität der Individuen während des iterativen Evolutionsprozesses insbesondere in dessen späten Phase [13]. Um die Diversität der Individuen in der späten Phase des Evolutionsprozesses zu behalten und die Gefahr der Konvergenz zu einem lokalen Optimum, sollen die Kreuzung- und Mutationsrate bzw. P_c und P_m sich im Zug des Evolutionsprozesses dynamisch anpassen [14]. Falls der prozentuale Anteil der Individuen mit höherer Fitness bzw. f_{cur} den vordefinierten Schwellenwert f_w überschreitet, vergrößern sich die Kreuzung- und Mutationsrate, um die Diversität der Individuen zu erhöhen:

$$P_c = \begin{cases} P_c^1, & f_{cur} < f_w, P_c^1 < P_c^2 \\ P_c^2, & f_{cur} \geq f_w \end{cases} \quad (5)$$

$$P_m = \begin{cases} P_m^1, & f_{cur} < f_w, P_m^1 < P_m^2 \\ P_m^2, & f_{cur} \geq f_w \end{cases} \quad (6)$$

5 Funktionsabsicherung mittels MiL-Simulation

Anhand des mechatronischen Entwicklungsprozesses wird die entwickelte Assistenzfunktion mit einem Test-szenario mittels MiL-Simulation abgesichert und demonstriert.

5.1 Szenariobeschreibung

Zur realitätsnahen Entwicklung und vor allem Erprobung der beschriebenen Funktionalitäten eines einzelnen FTF, der IoT-basierten Kommunikation in Produktionsanlage sowie der selbstoptimierten Intralogistik und Produktionsplanung wird ein cyber-physisches Industrie-4.0-Labortestfeld an der Ostfalia konzipiert und entwickelt, welche Untersuchungen in einer realen Umgebung unter Echtzeitbedingungen ermöglicht. Abbildung 9 stellt die Konfiguration des Testfeldes dar, wobei eine Produktionsanlage mit drei Produktionsmaschinen, einem Lager, einem Warenausgang und vier autonomen FTF simuliert wird.

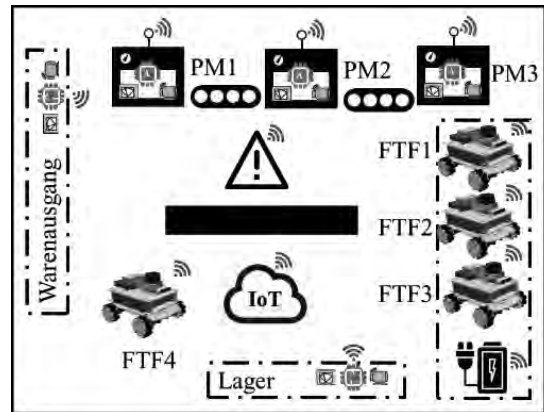


Abbildung 9: Cyber-physisches Industrie-4.0-Labortestfeld

Zur experimentellen Überprüfung der Funktionalität der entwickelten Assistenzfunktion werden 40 TA anhand der Konfiguration des Labortestfeldes zufällig generiert. Ausgehend von den TA werden die Transportzeit und die Länge des Fahrweges von jeweiligem TA mittels der entwickelten intelligenten Fahrfunktionen bzw. Zielführung und konfliktfreien Trajektorienplanung berechnet [6]. Aufbauend auf die Transportzeit und vordefinierte Deadline des TA können die Pünktlichkeit des Gütertransportes überprüft und die verzögerte Transportzeit bestimmt werden.

5.2 Simulationsergebnisse

Die Assistenzfunktion wird nach der Szenariobeschreibung mittels CAE-Werkzeug Matlab/Simulink durchgeführt. Die Simulationsergebnisse werden durch die folgenden Abbildungen veranschaulicht.

Abbildung 10 zeigt die Umgruppierung der TA als die endgültige Lösung des Optimierungsproblems für vier FTF auf, die durch das Gantt-Diagramm visualisiert wird. Bei entsprechender Gruppe verbinden sich die originalen TA miteinander in einer bestimmten Reihenfolge. Die Ziffer auf jeweiligem Balken bezieht sich auf den ID des originalen TA und die Länge des Balkens repräsentiert die Größe der Transportzeit.

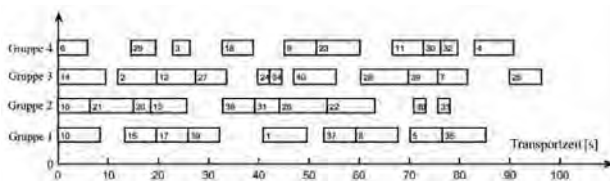


Abbildung 10: Gantt-Diagramm der neue umgruppierten TA

Die folgenden vier Abbildungen stellen die Konvergenzdiagramme der entsprechenden Zielkriterien bzw. Länge der Fahrwege l , Transportzeit t , Pünktlichkeitsrate u und Verzögerungszeit des Gütertransportes t_{gvz} bis zu 200 Generationen dar, wobei die Konvergenztendenz im Zug des Evolutionsprozesses von GA verdeutlicht. Die Schwankungen, die in den Abbildungen darunter erscheinen, sind auf die Tatsache zurückzuführen, dass das GA mehrere Ziele gleichzeitig optimiert. Aber solche Ziele können in Konflikte zueinander stehen und ein Kompromiss untereinander muss getroffen werden, sodass die Schwankungen unvermeidlich sind.

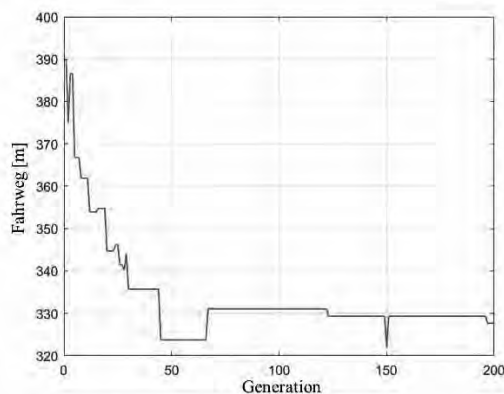


Abbildung 11: Konvergenzdiagramm für das Ziel zur Optimierung des Fahrweges

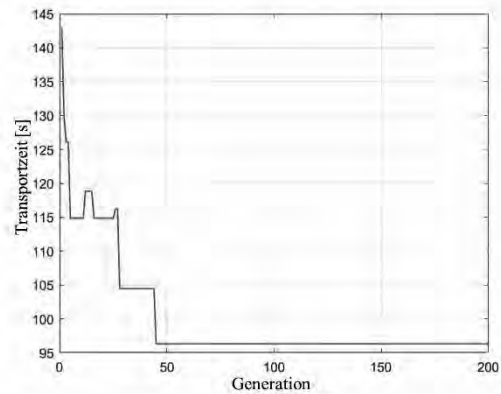


Abbildung 12: Konvergenzdiagramm für das Ziel zur Optimierung der Transportzeit

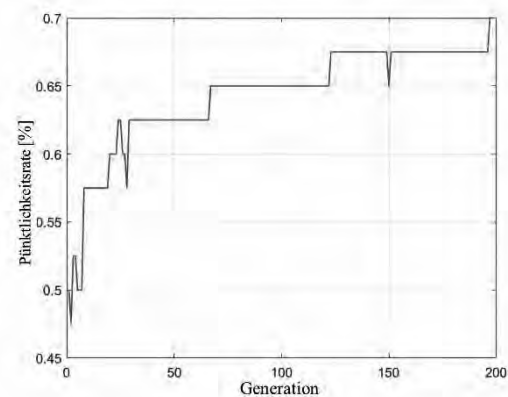


Abbildung 13: Konvergenzdiagramm für das Ziel zur Optimierung der Pünktlichkeitsrate

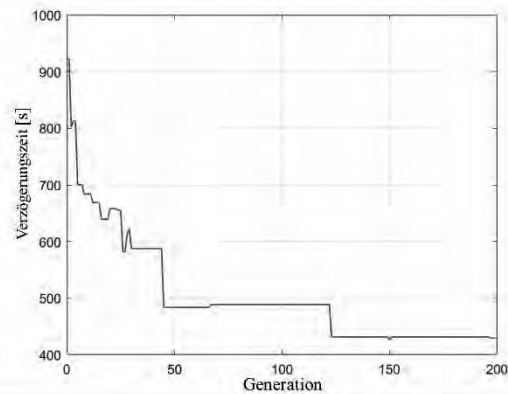


Abbildung 14: Konvergenzdiagramm für das Ziel zur Optimierung der Verzögerungszeit

6 Zusammenfassung und Ausblick

Im vorliegenden Beitrag wird die Assistenzfunktion mittels GA zur Klassifikation und Umgruppierung der

TA konzipiert und entwickelt, sodass die Multi-Ziele insbesondere Erhöhung der Pünktlichkeit und Minimierung der Verzögerung des Gütertransportes in einer Situation von mangelnder Transportleistung gleichzeitig optimiert werden können. Gegenüber des konventionellen GA, um die Konvergenz zu der globaloptimalen Lösung zu beschleunigen und die vorzeitige Konvergenz zu einer lokaloptimalen Lösung zu vermeiden, werden die Optimierungsansätze bzw. Elitismus-Strategie und adaptiv anpassende Kreuzung- und Mutationsrate eingesetzt. Mithilfe des Testszenarios wird die Funktionalität der Assistenzfunktion in der Simulationsumgebung validiert und demonstriert.

In den nächsten Schritten soll die Assistenzfunktion weiter optimiert und entwickelt werden. Obwohl die möglichen Konflikte zwischen FTF während des Gütertransportes mithilfe der Fahrfunktion Trajektorienplanung gelöst werden, sind die Rechenkosten von FTF ziemlich hoch [7]. Es wäre besser, solche möglichen Konflikte könnten auch in der Umgruppierung der TA im Vorfeld identifiziert und ggf. gelöst werden. Außerdem soll die unter Echtzeitbedingungen mittels HiL-Simulation erprobt und optimiert.

Danksagung

Dieser Beitrag wurde im Rahmen des Forschungsprojektes SMILE (*Simulationsbasierte Methodik zur Integration verteilter IoT-basierter, smarter Systeme mit dezentraler Intelligenz im lokalen Energienetz*) durch den Europäischen Fonds für regionale Entwicklung (EFRE) unter dem Förderkennzeichen ZW 6-85037374 gefördert. Die Verantwortung für den Inhalt liegt bei den Autoren.



Literatur

- [1] Matt DT, Modrák V, Zsifkovits H. *Industry 4.0 for SMEs: Challenges, Opportunities and Requirements*. Palgrave Macmillan, 2020, ISBN 978-3-030-25424-7.
- [2] VDI-Handbuch Technische Logistik, Band 2. *Automated Guided Vehicle Systems (AGVS)*. VDI 2510, Beuth Verlag, Düsseldorf, 2005.
- [3] Sundermeier J, Gehlhoff F, Fay A. *Development of a Simulation Model to Analyze the Performance of Decentral Rescheduling Algorithms in Production Systems*. SNE Simulation Notes Europe 30(1), Mar. 2020, pp. 15-22, doi: 10.11128/sne.30.tn.10504.
- [4] Liu-Henke X, Yarom OA, Göllner M. *Concept for the simulation-based description of intelligent mechatronic components and their networking in an Industry 4.0 (I4.0) production line*. The International Congress and Exhibition on Industrial and Manufacturing Engineering (ICEIME), Prague, Czech Republic, Aug. 2019.
- [5] Munkelt T, Krockert M. *An Approach to a Self-organizing Production in Comparison to a Centrally Planned Production*. SNE Simulation Notes Europe 30(1), Mar. 2020, pp. 23-30, doi: 10.11128/sne.30.tn.10504.
- [6] Zhang J, Yarom OA, Liu-Henke X. *Decentralized, Selfoptimized Order-acceptance Decision of Autonomous Guided Vehicles in an IoT-based Production Facility*. International Journal of Mechanical Engineering and Robotics Research, vol. 10, no.1, pp. 1-6, 2021.
- [7] Liu-Henke X et al. *Cyber-physical Industry 4.0 laboratory test field to simulate self-optimizing intralogistics*. The 19th International Conference on Mechatronics – Mechatronika (ME), pp.1-6, Dec. 2020.
- [8] Mousavi M, Yap HJ, Musa SN, Tahriri F, Dawal SZMd. *Multi-objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization*. PLoS ONE 12(3): e0169817. <https://doi.org/10.1371/journal.pone.0169817>.
- [9] Liu-Henke X, Jacobitz S, Scherler S, Göllner M, Yarom OA, Zhang J. *A Holistic Methodology for Model-Based Design of Mechatronic Systems in Digitized and Connected System Environments*. The 16th International Conference on Software Technologies (ICSOT), Online, July 6-8, 2021.
- [10] Inkermann D, Huth T, Vietor T, Grewe A, Knieke C, Rausch A. *Model-Based Requirement Engineering to Support Development of Complex Systems*. Procedia CIRP, 84, pp 239-244, 2019 doi: 10.1016/j.procir.2019.04.345
- [11] Yarom OA, Zhang J, Raulf C, Liu-Henke X, Vietor T. *Anforderungsmanagement für die modellbasierte Entwicklung mechatronischer Systeme im digitalisierten und vernetzten Umfeld*. 25. Symposium Simulationstechnik ASIM 2020, virtuelle Tagung, October 14 – 15 2020, pp 213-220, doi:10.11128/arep.59.a59029
- [12] Sun MH, Cui HT, Wen WD. *Topology optimization of continuum structure under multiple constraints using genetic algorithm with elitist strategy*. Journal of Aerospace Power, 2016, vol. 21 No. 4, pp. 732-737.
- [13] Chen P. *An improved genetic algorithm for solving the Traveling Salesman Problem*. 2013 Ninth International Conference on Natural Computation (ICNC), 2013, pp. 397-401, doi: 10.1109/ICNC.2013.6818008.
- [14] Cao K, Chen G, Jiang H, Ma H. *Guided self-adaptive evolutionary genetic algorithm*. Journal of Electronics & Information Technology, 2014, vol. 36(8), pp. 1884–1890, doi: 10.3724/SP.J.1146.2013.01446.

Verknüpfung von ereignisdiskreter Simulation und Process-Mining in Produktion und Logistik

Felix Özkul¹, Robin Sutherland¹, Sigrid Wenzel^{1*}, Ulrich Jessen¹, Sven Spieckermann^{2**}

¹Fachgebiet Produktionsorganisation und Fabrikplanung, Universität Kassel, Kurt-Wolters-Straße 3, 34125 Kassel, Deutschland; *s.wenzel@uni-kassel.de

² SimPlan AG, Sophie-Scholl-Platz 6, 63452 Hanau, Deutschland; **Sven.Spieckermann@simplan.de

Abstract. Der vorliegende Beitrag thematisiert den kombinierten Einsatz von ereignisdiskreter Simulation und Process-Mining im Anwendungskontext von Produktions- und Logistiksystemen. Hierbei werden – neben dem in der einschlägigen wissenschaftlichen Literatur dargelegten Stand der Technik – vor allem anwendungsbezogene Interoperabilitätsszenarien aufgegriffen und in der spezifizierten Domäne skizziert. Der Schwerpunkt der Ausführungen wird aus der Simulation heraus begründet. Aktuelle Herausforderungen und zukünftige Forschungsbedarfe schließen den Beitrag ab.

Einleitung

Die Notwendigkeit des Einsatzes der ereignisdiskreten Simulation (*Discrete Event Simulation*, DES) in der Planung, der Realisierung und im Betrieb von Produktions- und Logistiksystemen [1-4] ergibt sich vor allem dadurch, dass Simulationsmodelle im Hinblick auf zu untersuchende Ziele und Fragestellungen die Realität hinreichend genau abbilden. Mit ihnen sind experimentelle Untersuchungen möglich, ohne Einschränkungen aufgrund struktureller und stochastischer Komplexität oder des dynamischen Zeitverhaltens der zu untersuchenden Systeme vornehmen zu müssen, welche die Aussagekraft der Modelle begrenzen. Weiterhin ist es mithilfe der DES möglich, verschiedene Szenarien in Systemen im Rahmen von „Was-wäre-wenn“-Fragestellungen experimentell zu untersuchen [5], ohne (kosten- und zeitintensive) Veränderungen am realen System vornehmen zu müssen. Im Zusammenhang mit komplexer werdenden Anwendungssystemen der Produktion und Logistik [6, 7], digitalisierungsgetriebenen disruptiven Umbrüchen [8] sowie kürzer werdenden Produktlebenszyklen [9] nimmt die Bedeutung der DES als Planungs- und Betriebswerkzeug einen immer größer werdenden Stellenwert ein und

stellt vor allem auch die Grundlage für sogenannte Digitale Fabrikzwillinge dar [10].

Process-Mining stellt einen Ansatz zur Gewinnung von Erkenntnissen zu Unternehmensprozessen auf Grundlage von Ist-Daten (aus Eventlogs) dar [11]. Anders als das Data-Mining besitzt Process-Mining einen ausschließlich prozessbezogenen Fokus (Abgrenzungen zum Data-Mining und den Methoden des Business Process Managements finden sich in [11]). Die Hauptaufgaben des Process-Mining umfassen die Prozessextraktion (*process discovery*), die Konformitätskontrolle (*conformance checking*) und die (Prozess-)Modellerweiterung (*process enhancement*) [12].

Beide Methoden werden in der Praxis typischerweise getrennt voneinander eingesetzt. Grundlage für die Anwendung des Process-Mining sind Algorithmen (für ausgewählte Prozessextraktionsalgorithmen siehe [13 - 19]), die vielfach in Open-Source-Werkzeugen [20, 21] und kommerziellen Softwarewerkzeugen [22] implementiert vorliegen. Für die DES in Produktion und Logistik existiert ebenfalls eine Vielzahl an unterschiedlichen Simulationswerkzeugen, die sich im Hinblick auf das implementierte Modellierungskonzept (beispielsweise bausteinorientiert, agentenbasiert) unterscheiden [5]. Der Einsatz beider Methoden erfordert die Einbringung von Expertenwissen – eine ganzheitliche Methodik für den verknüpften Einsatz, der die Charakteristiken der Planung von Produktions- und Logistiksystemen berücksichtigt, existiert nicht. Allerdings kann gerade durch das Zusammenwirken beider Methoden zusätzlicher Mehrwert in der Anwendung realisiert werden.

Bezugnehmend auf die beschriebene Ausgangssituation werden im Folgenden zunächst der Stand der Technik des Process-Mining im Kontext von DES näher beschrieben (Kapitel 1), bevor ausgewählte anwendungsbezogene Interoperabilitätsszenarien und die damit verbun-

denen Herausforderungen erläutert werden. Zuletzt werden weitere Forschungsbedarfe zusammenfassend dargestellt.

1 Stand der Technik des Process-Mining im Kontext von DES

Während das Process-Mining primär vergangenheitsbezogene Analysen zu Prozessen auf Grundlage von Ist-Daten ermöglicht, ist die DES besonders auf die Untersuchung zukunftsbezogener Szenarien ausgerichtet. Der verknüpfte Einsatz beider Methoden erscheint daher naheliegend, um einerseits die datengetriebenen zumeist automatisierbaren Verfahren des Process-Mining mit der Zukunftsorientierung von DES nutzen zu können. So werden in der Literatur bereits verschiedene Konzepte für den kombinierten Einsatz beider Methoden vorgestellt [23-34]. Darüber hinaus diskutiert van der Aalst (2018) verschiedene Kombinationen von DES und Process-Mining und analysiert diese theoretisch in Bezug auf Qualitätskonzepte wie die Richtig-positiv-Rate (*recall*), die Genauigkeit (*precision*) und die Generalisierung [35]. Er bezieht sich mit seinen Theorien nicht auf die im Bereich Produktion und Logistik priorär verwendeten strukturorientierten Ansätze, sondern beschränkt sich ausschließlich auf die prozessorientierte DES.

Aus der Simulationssicht kann das Process-Mining genutzt werden, um den Modellierungsprozess zu verbessern und einen Vergleich zwischen den Simulationsläufen und dem tatsächlichen Systemverhalten herzustellen [35]. Eine systematische Aufarbeitung und Bewertung der dadurch entstehenden Potenziale erfolgen in [27, 30]. In der konzeptionellen Arbeit von Martin et al. (2014) werden die Potenziale der Verwendung von Eventlogs zur Entwicklung eines Simulationsmodells bezogen auf Entitäten, Aktivitäten, Ressourcen und dem Kontrollfluss dargelegt [30].

Einer der ersten Ansätze zur Verknüpfung der beiden Methoden ist von Rozinat et al. (2009). In diesem Ansatz wird das Process-Mining angewendet, um aus Eventlogs Geschäftsprozesse zu ermitteln und zu analysieren. Ergänzend liegt ein Simulationsmodell in Form eines Petri-Netzes vor und kann zur Prozessanalyse und Leistungsbewertung verwendet werden. Durch dieses Modell können Gestaltungsalternativen zur Verbesserung der Res-

sourcennutzung und Verringerung der Durchlaufzeit simuliert werden [34]. Die Verknüpfung der beiden Methoden zum Entwurf von Petri-Netzen wird in den Ansätzen von Märuster et al. (2009) und Khodyrev et al. (2014) aufgegriffen [31, 28]. Der Fokus liegt auf der einen Seite in der Prozessneugestaltung [31] und auf der anderen Seite in der kurzfristigen Vorhersage von Leistungskennzahlen [28]. Darüber hinaus ermitteln Gawin und Marcinkowski (2015) durch den Einsatz von Process-Mining die Prozessmodellstruktur, die Bearbeitungszeiten der Aktivitäten und eine vereinfachte Entscheidungslogik [26]. In der jüngeren Vergangenheit zeigen Pourbafrani et al. (2020), dass die Entwicklung von System Dynamics-Modellen mithilfe der Extraktion von Leistungsparametern aus Eventlogs unterstützt werden kann [33].

Die vorgestellten Arbeiten haben fast alle gemein, dass sie Verbesserungen in den Simulationsstudienphasen der Systemanalyse (Konzeptmodell), der Modellformalisierung (formales Modell) sowie der Experimentdurchführung fokussieren. Dabei bildet die Ermittlung der Prozessmodelle aus den Eventlogs mithilfe des Process-Mining die Grundlage und wird anschließend durch eine Parametrisierung des Modells ergänzt (z. B. Ankunftsrate oder Bearbeitungszeiten) [25]. Im Gegensatz zur automatisierten Prozessmodellierung erfolgt die genaue Abstimmung (Parametrisierung) von Simulationsmodellen in den meisten Fällen weiterhin durch Anwenderinnen und Anwender in Abhängigkeit vom gewünschten Detaillierungsgrad, den das Modell für ein bestimmtes Untersuchungsziel aufweisen muss. Dieser manuelle Arbeitsschritt wird in der Methodik von Camargo et al. (2020) durch eine iterative Hyperparameteroptimierung automatisiert [25]. Mithilfe dieses Verfahrens wird ein möglicher Lösungsraum im Hinblick auf die Maximierung der Ähnlichkeit des simulierten und des realen Verhaltens (beschrieben in den Eventlogs aus dem Untersuchungssystem), durchsucht [25].

Für den Vergleich von Gestaltungsalternativen und die Beantwortung von „Was-wäre-wenn“-Fragestellungen erweist sich der Modellierungsprozess von komplexen Produktions- und Logistiksystemen als zeitintensiv [5]. Dieser Aufwand kann durch die Anwendung von Process-Mining im Rahmen der Datenvorverarbeitung reduziert werden. So entwickelten Langer et al. (2021) eine Methodik, welche die Anwendung von Simulationsmodellen im Planungsprozess (insbesondere in der Umplanung von Produktions- und Logistiksystemen) unterstützt [29]. Die gewonnenen Erkenntnisse bestätigen,

dass eine durch das Process-Mining geschaffene fundierte Datenbasis für Simulationsmodelle Potenziale in den Bereichen der Datenerfassung, Datenqualität und Modellbildung eröffnet.

Im Gegensatz zu den bisher vorgestellten Ansätzen ist die Methodik von Mesabbah und McKeever (2018) nicht auf ein konkretes Anwendungsfeld beschränkt, sondern kann, unter Voraussetzung vorhandener Eventlogs, in unterschiedlichen Bereichen angewendet werden [32]. Ihr generischer Ansatz soll es Führungskräften aller Anwendungsfelder und Branchen ermöglichen, aufwandsarm Prozessvarianten mithilfe von DES und Process-Mining zu untersuchen [32].

Die vorgestellten Forschungsarbeiten, die sich durch eine Verknüpfung von DES und Process-Mining ergeben, erweitern die bisherigen Analysemöglichkeiten und eröffnen damit Nutzenpotenziale. Zur Erschließung dieser Potenziale werden erarbeitete anwendungsbezogene Interoperabilitätsszenarien im Folgenden vorgestellt.

2 Interoperabilitätsszenarien und ihre Herausforderungen

Bei der Verknüpfung von DES und Process-Mining kristallisieren sich verschiedene Forschungsbedarfe für ein effektives methodisches Zusammenwirken heraus. Aufbauend auf der vorgestellten Literatur beschreiben die folgenden Abschnitte ausgewählte identifizierte Szenarien

für die methodische Interoperabilität sowie Herausforderungen im Anwendungskontext.

2.1 Anwendungsbezogene Interoperabilitätsszenarien

Die folgende Abbildung 1 visualisiert das Zusammenwirken zwischen DES und Process-Mining in Form von möglichen Interoperabilitätsszenarien.

Mithilfe von Ist-Daten aus informationstechnischen (IT) Systemen (wie bspw. Warehouse Management Systemen) werden zu den zu untersuchenden Produktions- und Logistiksystemen Eventlogs erzeugt (1.), aus denen im Anschluss durch die Anwendung von Prozessextraktionsalgorithmen des Process-Mining Ist-Prozessgraphen generiert werden. Auf Grundlage der Ist-Prozessgraphen können danach strukturorientierte Simulationsmodelle erzeugt werden (2.). Hierfür sind die aus den Eventlogs relevanten Informationen für die DES festzulegen und mithilfe eines Mapping-Mechanismus hinsichtlich der semantischen Zusammenhänge auf den strukturorientierten DES-Modellierungsansatz abzubilden. Auf syntaktischer Ebene ist der rein technische Datenaustausch zwischen den Werkzeugen zur DES und zum Process-Mining zu gewährleisten. Mithilfe des Simulationsmodells können anschließend „Was-wäre-wenn“-Analysen im Rahmen von Experimenten durchgeführt und verschiedene Szenarien näher betrachtet werden. Das Process-Mining kann dabei genutzt werden, um implementierte

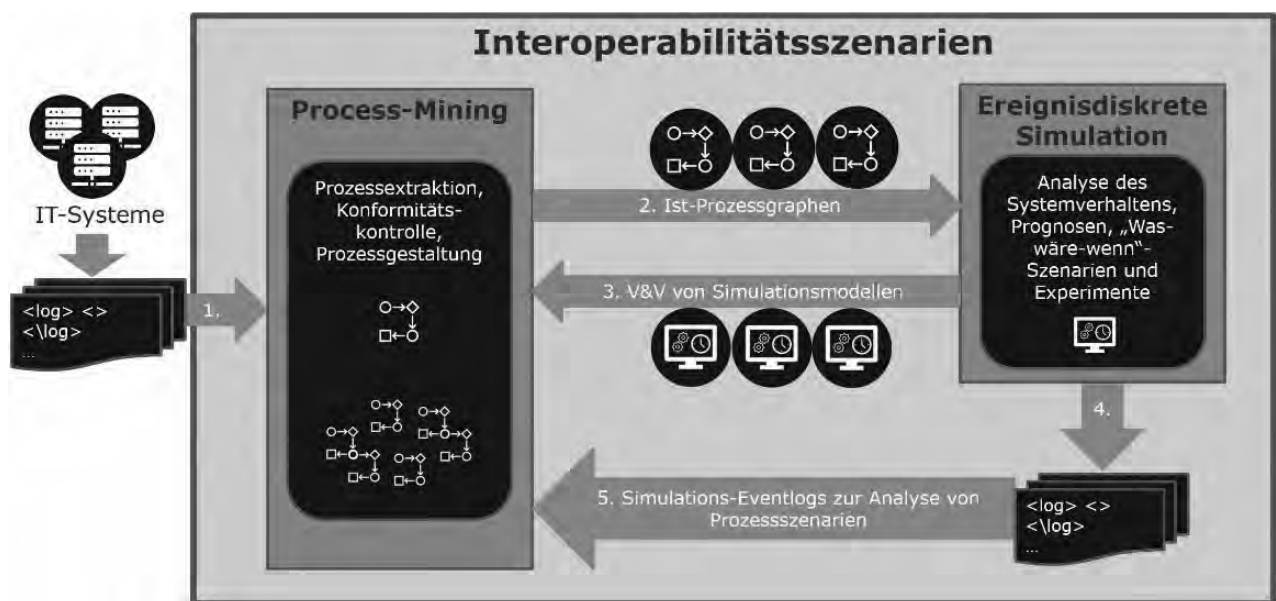


Abbildung 1: Interoperabilitätsszenarien zwischen DES und Process-Mining (eigene Darstellung)

Simulationsmodelle im Rahmen der Verifikation und Validierung (V&V) zu überprüfen (3.). Weiterhin können auf Basis von Ergebnisdaten aus der DES wiederum Eventlogs erzeugt werden (4.). Diese werden im Anschluss durch die Anwendung des Process-Mining zur Generierung von Soll-Prozessmodellen genutzt (5.), so dass verschiedene Szenarien formalisiert und bspw. für Konformitätskontrollen oder Prozessmodellerweiterungen verwendet werden können. Hierauf aufbauend können per DES abgesicherte, zukunftsorientierte Prozessanalysen vorgenommen und konkrete Bedarfe hinsichtlich der Prozessgestaltung im Untersuchungssystem identifiziert werden.

Abbildung 2 konkretisiert ein Interoperabilitätsszenario anhand eines entwickelten Software-Demonstrators.

Auf Grundlage von Eventlogs aus verschiedenen IT-Systemen kann zunächst mithilfe eines webbasierten Frontends eine formale Prozessbeschreibung für das zu untersuchende Produktions- oder Logistiksystem – in Form eines parametrisierten Prozessmodells – generiert werden.

Hierbei werden Eventlogs über eine Datenschnittstelle im Frontend hochgeladen. Als Schnittstellen- bzw. Eventlogformate stehen XES (eXtensible Event Stream) und CSV (Comma Separated Values) zur Verfügung. XES existiert als Schema für die Struktur von Eventlogs in mehreren leicht verschiedenen Implementierungen (im Rahmen dieser Forschungen wird der IEEE-Standard aus [36] zugrunde gelegt), ist aufgrund des XML-Schemas (XML: Extensible Markup Language) serialisierbar und wird von gängigen kommerziellen Process-Mining-Werkzeugen unterstützt. Der Nachteil von XES im Anwendungskontext Produktion und Logistik ist die geringe native Unterstützung von Eventlogs im XES-Schema in den relevanten Informationssystemen. Hierfür eignet sich wiederum eher CSV als generisches Format.

Auf Grundlage des hochgeladenen Eventlogs wird anschließend mithilfe von Prozessextraktionsalgorithmen ein initiales Prozessmodell nach den Regeln der Business Process Model and Notation (BPMN) generiert (zur Modellierung mit BPMN vgl. [37]). Abhängig von

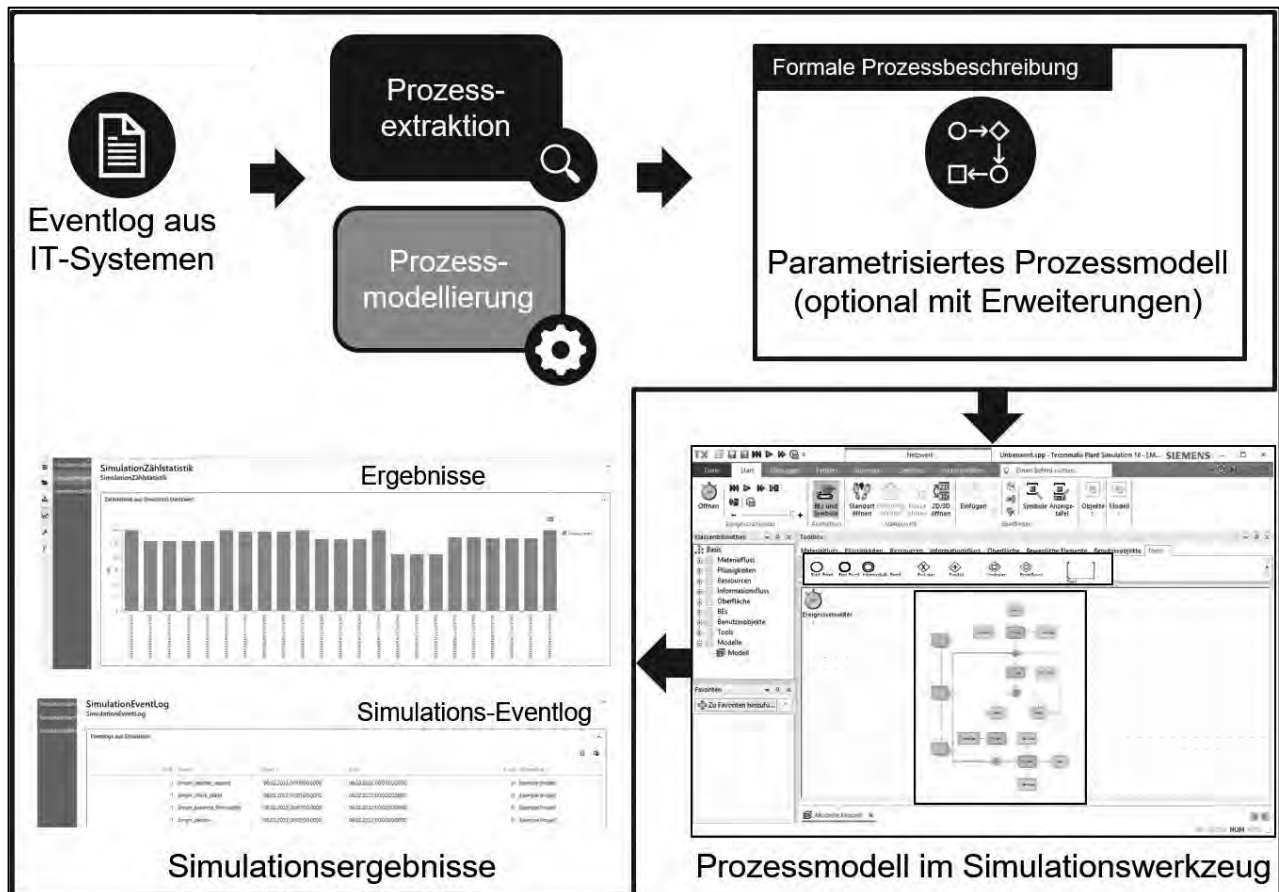


Abbildung 2: Beispielhaftes Interoperabilitätsszenario zwischen DES und Process-Mining (eigene Darstellung)

den Informationen innerhalb der Eventlogs werden einzelnen Aktivitäten an dieser Stelle bereits Parameter (wie bspw. Bearbeitungszeiten) automatisiert zugewiesen.

Das initiale Modell kann optional manuell mithilfe einer grafischen Bedienoberfläche und unter Verwendung von Prozessbausteinen im Rahmen einer Prozessmodellierung erweitert werden; die Berücksichtigung des Expertenwissens ist hierbei besonders im Hinblick auf die Akzeptanz und eine für die Aufgabe hinreichende Realitätsstreuung der Prozessbeschreibung zweckmäßig.

Aufbauend auf dem parametrisierten Prozessmodell wird auf einem Server im Hintergrund ein Simulationsmodell in einem DES-Werkzeug generiert. Mithilfe anschließender Experimente können Aussagen über das Systemverhalten anhand der zu betrachtenden Leistungskennzahlen (Simulationsergebnisse) getroffen und verschiedene Prozessvarianten analysiert werden. Darüber hinaus können über die DES neue Eventlogs erstellt werden, die wiederum als Grundlage für die Anwendung des Process-Mining dienen. Beispielsweise kann auf Grundlage der Simulations-Eventlogs für eine – im Hinblick auf die betrachteten Simulationsergebnisgrößen – vielversprechende Modellvariante ein initiales Soll-Prozessmodell extrahiert werden, das beurteilt und zu einem normativen Modell erweitert werden kann. Das normative Modell kann bei Bedarf in der DES weiter untersucht werden, wodurch eine iterative und experimentelle Soll-Prozessgestaltung und -validierung möglich wird.

Die beschriebenen Schnittstellen zwischen DES und Process-Mining erlauben es einerseits, die per DES erzeugten Eventlogs durch Process-Mining-Werkzeuge zu analysieren, und andererseits können die durch das Process-Mining erstellten Prozessmodelle als Basis für die Simulationsmodellerstellung verwendet werden. Der Austausch kann dabei über die Eventlogs (unter Verwendung des XES-Standards) erfolgen.

2.2 Herausforderungen

Losgelöst von spezifischen Interoperabilitätsszenarien ergeben sich aufgrund der methodischen Charakteristiken von DES und Process-Mining sowie den anwendungsseitig verbreiteten Werkzeugen forschungsseitige Herausforderungen, die im Folgenden näher beschrieben werden.

Strukturbezogenheit vs. Prozessbezogenheit.

Process-Mining-Werkzeuge arbeiten auf Grundlage von Eventlogs, die Informationen über Prozesse enthalten

und aus denen Prozessmodelle (i. e. Modelle, die mithilfe einer Prozessmodellierungssprache – und damit einer festgelegten Syntax – beschrieben werden) algorithmisch extrahiert werden können. In der praktischen Anwendung von Process-Mining wird hierbei häufig auf Basis von Directly Follows Graphs (DFGs) [38] und mit BPMN gearbeitet. DFGs bestehen aus Aktivitäten (Knoten) und Relationen (gewichtete Kanten) und haben einen höheren Abstraktionsgrad als Modellierungstechniken wie Petri-Netze oder BPMN [38].

Besonders in der praktischen Anwendung finden DFGs Verwendung bei der Generierung von Prozessmodellen aus Eventlogs, da die Regeln der Grapheninterpretation im Vergleich zu anderen Modellierungssprachen vereinfacht sind.

Simulationswerkzeuge in Produktion und Logistik verfolgen jedoch primär einen strukturbezogenen Modellierungsansatz, bei dem die Bausteine physische Systemelemente (wie bspw. Fördertechnik oder Arbeitsstationen) und Systemgrenzen (i. e. Quellen und Senken) abbilden [5]. Ablaufsteuerungen bzw. Strategien werden über Parametrisierungen, separate Steuerungsbausteine oder implementierte Algorithmen umgesetzt. Die Abbildung des Zeitfortschritts erfolgt (im originär ereignisgesteuerten Fall) durch Ereignisse, die in einer Ereignisliste zeitlich eingeordnet und in einer linearen Ordnung über das Ausführen von Ereignisroutinen abgearbeitet werden [5]. Über das Eintreten eines Ereignisses und das Ausführen dieser Ereignisroutinen werden Zustandsänderungen in den Simulationsbausteinen und im Simulationsmodell umgesetzt und neue Ereignisse generiert, sodass Simulationsabläufe im Sinne von Prozessen entstehen. Die Ablauflogik eines Prozesses innerhalb eines Simulationsmodells wird von den o.g. Strategien bestimmt; Beispiele für Strategien sind First-In-First-Out an zusammenführenden Bausteinen oder eine zielgerichtete Verteilung nach Produkttypen an Verteilelementen.

Die genannten Simulationsmodellelemente bilden Systemelemente realer Produktions- und Logistiksysteme ab. Letztgenannte Elemente können wiederum grundsätzlich in Eventlogs abgebildet werden (In XES können Elemente, an denen die protokollierten Ereignisse auftreten, beispielsweise mithilfe von Attributen bzw. Extensions ausgewiesen werden, um die Semantik der Eventlogs zu erweitern.).

In Abhängigkeit des anwendungsbezogenen Kontextes der methodischen Verknüpfung ist daher zu klären, in welcher Granularität Daten über Systemelemente in Eventlogs abgebildet werden müssen, um hinreichende

Informationen über Systemstrukturen zu erhalten, damit geeignete Modellelemente und -strukturen festgelegt werden können.

Umsetzung der semantischen Interoperabilität.

Aufgrund der unterschiedlichen Modellierungspadigmen von DES und Process-Mining stellt die Überführung der Modelle dieser beiden Methoden aufeinander eine weitere Herausforderung dar. Zur Umsetzung einer allgemeinen Lösung für die Schaffung einer semantischen Interoperabilität kommt erschwerend hinzu, dass sich auch die Modellierungspadigmen und daraus resultierende Algorithmen einzelner Simulationstools untereinander unterscheiden. Das gilt in ähnlicher Weise für die implementierten Algorithmen einzelner Werkzeuge des Process-Mining. Exemplarisch hierfür sind beispielsweise verschiedene Konzepte zur Modellerstellung oder zur Erzeugung von Zufallszahlen im Bereich der DES [39] und die unterschiedliche (aber dadurch in Teilen inkorrekte) Implementierung von Process-Mining-Algorithmen zum Zwecke der effizienteren Analyse von Eventlogs zu nennen. Die zuletzt genannte Problematik wird in Bezug auf den Algorithmus zur Erzeugung der weitverbreiteten DFGs in [38] thematisiert. Die Herausforderung in der Forschung besteht daher darin, semantische Interoperabilitätslösungen zu schaffen, welche einerseits die Modellierungspadigmen von DES und Process-Mining beibehalten und andererseits

gleichzeitig die sich durch die abweichenden softwareseitigen Implementierungen ergebenden semantischen Restriktionen einbeziehen.

Perspektivisch kann eine Lösung hierfür durch die Konzeptionalisierung, Implementierung und Wartung eines semantischen Modells erzielt werden. Abbildung 3 skizziert beispielhaft einen solchen Ansatz in einer stark vereinfachten OWL-Ontologie (*Web Ontology Language*-Ontologie), welche Relationen zwischen Systemelementen eines realen Untersuchungssystems in Produktion und Logistik sowie der Abbildung der Elemente in einem Eventlog und in der DES herstellt. Im linken Bildteil sind Klassen (Konzepte) der Domänen „Untersuchungssystem“, „DES“ und „Process-Mining“ abgebildet, die mittels der Relationen (Pfeile) „hat Subklasse“ und „hat Individuum“ eine Taxonomie der Klassen sowie eine Zuordnung individueller Instanzen (Individuen) zu diesen Klassen darstellen. Im rechten Bildteil sind die Individuen „Qualitätskontrolle“, „Informationssystem1“, „StationPS1“, „Arbeitsstation1DOS3“ und „XES1“ dargestellt, die Instanzen ihrer zugehörigen Klassen sind (ausgedrückt durch „hat Individuum“). Im gezeigten Beispiel wird die „Qualitätskontrolle“ dem Individuum „XES1“ (der Klasse „XESEventlog“) mittels der Relation „istAbgebildetInXES“ zugeordnet; die inverse (i. e. die entgegengesetzt gerichtete) Relation ist „bildetMFSysystemelementAb“. Gleichzeitig wird die Qualitätskontrolle zu den Simulationsbausteinindividuen „Station1PS“ und „Arbeitsstation1DOS3“ durch die Relation „istAbgebildetInDES“ in Bezug gesetzt.

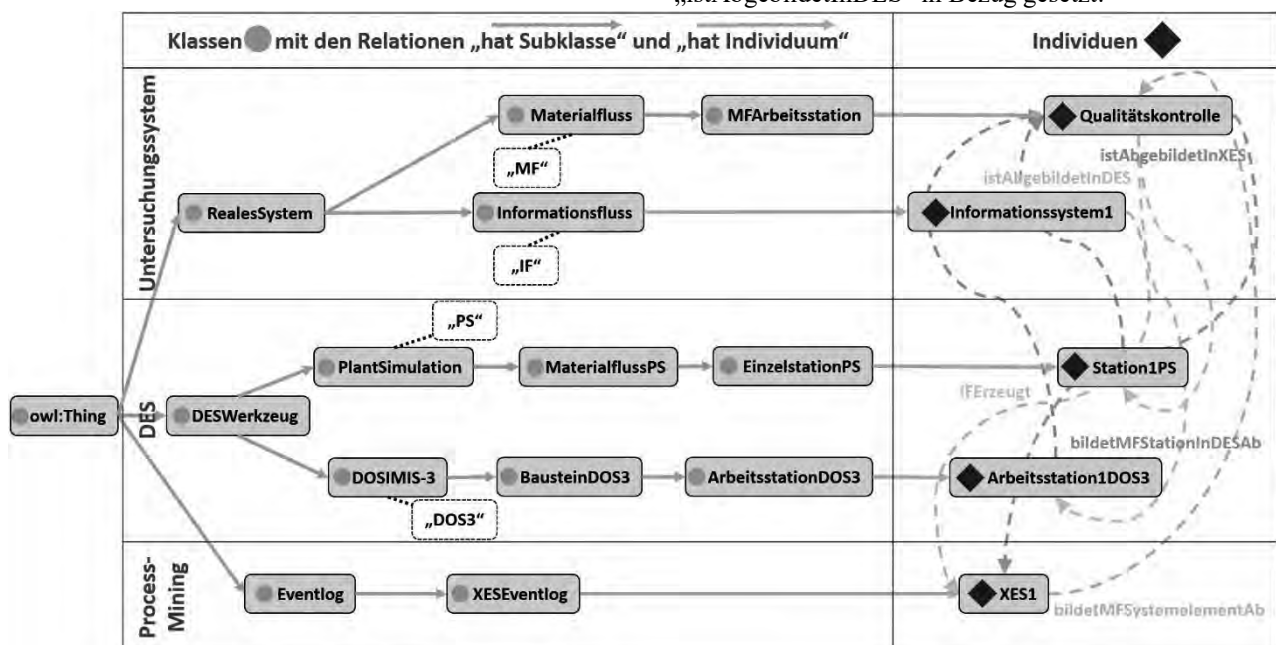


Abbildung 3: Semantisches Modell (Ausschnitt) für die Interoperabilität zwischen DES und Process-Mining (eigene Darstellung)

Mithilfe einer Inferenzmaschine (*Reasoner*) kann im dargestellten Fall (transitiver Relationen) automatisiert geschlussfolgert werden, dass die im XES-Eventlog abgebildete Qualitätskontrolle in den beiden Simulationswerkzeugen durch die Simulationsbausteine Einzelstation bzw. Arbeitsstation abgebildet wird. Weiterführendes zum Ontologie-Einsatz im Kontext DES stellen Stolipin und Wenzel (2019) vor [40].

Im Zusammenhang mit dem ontologiegestützten Ansatz für die hier beschriebenen Interoperabilitätsszenarien ist im Rahmen weiterführender Forschung insbesondere zu klären, wie mit der hohen Komplexität des Modells umgegangen werden kann.

3 Weitere Forschungsbedarfe

Die Verknüpfung von DES und Process-Mining schafft im Kontext der Digitalen Transformation in Produktion und Logistik einen signifikanten Mehrwert in Bezug auf die verbesserte Nutzung realer Daten in Simulationsmodellen, die Unterstützung eines automatisierten Modellierungsprozesses für die DES und die vergleichende Bewertung von realem und simuliertem Prozess. Zur Erreichung einer durchgängigen Interoperabilität müssen jedoch noch wesentliche simulationsseitige Herausforderungen überwunden und zusätzliche Szenarien näher erforscht werden. Weiterführende konkrete Forschungsbedarfe werden nachfolgend beschrieben.

Im Rahmen einer systematischen Simulationsstudie werden phasenübergreifende V&V-Aktivitäten durchgeführt, damit fehlerhafte Erkenntnisse in Bezug auf das zu untersuchende System vermieden werden [5, 41]. V&V-Aktivitäten können dabei durch den Einsatz verschiedener Methoden unterstützt werden und müssen – entsprechend den verschiedenen Phaseergebnissen – differenziert eingesetzt werden (für ausführliche Beschreibungen hierzu siehe [41]). Im Rahmen eines strukturierten Untersuchungs- und Evaluationsvorgehens ist daher zu klären, welche Phaseergebnisse einer Simulationsstudie durch den Einsatz von Process-Mining als V&V-Methode besonders profitieren können und wie ein geeignetes Vorgehen zur Anwendung des Process-Mining als Methode der V&V gestaltet werden kann.

Der verknüpfte Einsatz von DES und Process-Mining setzt voraus, dass die Interoperabilität in der Anwendung akzeptiert wird. Wie eingangs beschrieben, existieren relevante Softwarelösungen für beide Methoden in der Pra-

xis derzeit getrennt voneinander und bedienen (bis zu einem gewissen Grad) die Anforderungen der Anwendung. Anwendungssoftwarewerkzeuge, die im Rahmen des beschriebenen Forschungsvorhabens erarbeitet werden, müssen neben ihrer Konformität in Bezug auf Gestaltungsrichtlinien der Softwareentwicklung vor allem auf ihre Akzeptanz hin beurteilt werden. Hierzu muss – neben der glaubwürdigen Darlegung eines anwendungsbezogenen Mehrwerts – ein geeignetes Evaluationsvorgehen entwickelt und angewendet werden.

Neben umfassenden Forschungsanstrengungen im Kontext der DES (derzeit besonders relevante Handlungsfelder sind zusammengefasst in [42]), zeichnet sich das Forschungsgebiet des Process-Mining ebenfalls durch seine Schnelligkeit und eine schwerpunktbezogene Vielfältigkeit hinsichtlich seiner Forschungsfelder aus. Eine besondere methodische Entwicklung, die im Rahmen der Weitererforschung zur Interoperabilität zu berücksichtigen sein wird, ist das objektzentrierte Process-Mining (*Object-centric Process Mining*) [43-45]. Ein objektzentriertes Process-Mining berücksichtigt die Umstände, dass ein Ereignis zu mehreren Prozessinstanzen gehören (*convergence*) und dass eine Aktivität innerhalb einer Prozessinstanz (planmäßig) mehrfach auftreten kann (*divergence*) [43] – beides kann zu irreführenden Ergebnissen in der Anwendung von Process-Mining führen, die durch derzeitige Abbildungen in gängigen Eventlogformaten (i. e. XES) unvermeidbar sind. Im Zusammenhang mit der Interoperabilität zwischen DES und Process-Mining ist daher zu prüfen, welche Szenarien durch den Einsatz des objektzentrierten Process-Mining in Produktion und Logistik neu formalisiert bzw. anwendungsbezogen aufgewertet werden können.

Danksagung

Dieser Beitrag entstand im Rahmen des Forschungsprojektes „SimProve – Verknüpfung von DES und Process-Mining zur effizienteren Analyse und Gestaltung von Produktions- und Logistiksystemen“, das durch Mittel des Landes Hessen im Rahmen des Distr@l-Programms gefördert und zwischen September 2021 und August 2023 unter Konsortialführerschaft der SimPlan AG durchgeführt wird.

Literatur

- [1] VDI. *VDI 3633: Blatt 1. Simulation von Logistik-, Materialfluss- und Produktionssystemen*. Berlin: Beuth; 2014.

- [2] Bergmann S, Straßburger S. Automatische Modellgenerierung – Stand, Klassifizierung und ein Anwendungsbeispiel. In: Mayer G, Pöge C, Spieckermann S, Wenzel S. *Ablaufsimulation in der Automobilindustrie*. 1. Aufl. Berlin, Heidelberg: Springer Vieweg; 2020. S. 333–347.
- [3] Sprock T, Hilmer F. Bereitstellungssimulation. In: Mayer G, Pöge C, Spieckermann S, Wenzel S. *Ablaufsimulation in der Automobilindustrie*. Berlin, Heidelberg: Springer; 2020. S. 189–204.
- [4] Mayer G, Mieschner M. Industrie 4.0 – Chance oder Risiko für die Ablaufsimulation. In: Wenzel S, Peter T. *Simulation in Produktion und Logistik*. Kassel: Kassel University Press; 2017. S. 177–187.
- [5] Gutenschwager K, Rabe M, Spieckermann S, Wenzel S. *Simulation in Produktion und Logistik: Grundlagen und Anwendungen*. Berlin, Heidelberg: Springer Vieweg; 2017.
- [6] Schmid A, Sobottka T, Lielacher M. Simulationsbasierte Optimierung von Bestelllosgrößen in der verbrauchsge- steuerten Materialdisposition der Investitionsgüterindustrie. In: Franke J, Schuderer P. *Tagungsband Simulation in Produktion und Logistik*. Göttingen: Cuvillier Verlag; 2021. S. 475–485.
- [7] Meyer T, Pöge C. Die Materialflusssimulation im Kontext eines Produktlebenszyklusmanagements. In: Franke J, Schuderer P. *Tagungsband Simulation in Produktion und Logistik*. Göttingen: Cuvillier Verlag; 2021. S. 247–257.
- [8] Matysczok C, Seewaldt M. VDI Fachausschuss Digitale Fabrik – Darstellung der aktuellen Aktivitäten und Richtlinien. In: Putz M, Schlegel A. *Tagungsband Simulation in Produktion und Logistik*. Auerbach: Verlag Wissenschaftliche Scripten; 2019. S. 39–49.
- [9] Massow BB, Hausberger T, Geiger DM, Klotz M, Schlegel A, Putz M. Remote-Lernfabrik – Simulationsmodelle im Anlagenentstehungsprozess und deren Integration in die Virtuelle Inbetriebnahme. In: Franke J, Schuderer P. *Tagungsband Simulation in Produktion und Logistik*. Göttingen: Cuvillier Verlag; 2021. S. 575–585.
- [10] Shao G, Jain S, Laroque C, Lee LH, Lendermann P, Rose O. Digital Twin for Smart Manufacturing: The Simulation Aspect. In: Mustafee N. *WSC '19: Proceedings of the 2019 Winter Simulation Conference*. Piscataway, NJ: IEEE Press; 2019. S. 2085–2098.
- [11] van der Aalst WMP. *Process Mining: Data Science in Action*. Berlin, Heidelberg: Springer; 2016.
- [12] van der Aalst WMP. Fokus Prozesse: Von den Anfängen bis zur Verbesserung von Arbeitsabläufen in der Gesundheitsversorgung. *Wissensmanager: Das KMS Magazin für die Gesundheitswirtschaft*. 2019. S. 4–10.
- [13] van der Aalst WMP, Weijters T, Maruster L. Workflow mining: discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*. 2004; 16 (9): S. 1128–1142. doi: 10.1109/TKDE.2004.47.
- [14] de Medeiros AKA, Weijters AJMM, van der Aalst WMP. Genetic process mining: an experimental evaluation. *Data Min Knowl Disc*. 2007; 14 (2): S. 245–304. doi: 10.1007/s10618-006-0061-7.
- [15] van Dongen BF, de Medeiros AKA, Wen L. Process Mining: Overview and Outlook of Petri Net Discovery Algorithms. In: Jensen K, van der Aalst WMP. *Transactions on Petri Nets and Other Models of Concurrency II: Special issue on concurrency in process-aware information systems*. Berlin, Heidelberg: Springer; 2009. S. 225–242.
- [16] Weijters AJMM, Ribeiro JTS. Flexible Heuristics Miner (FHM). In: IEEE. *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2011)*; Apr 2011; Paris, Frankreich. Piscataway, NJ: IEEE, S. 310–317.
- [17] Leemans SJJ, Fahland D, van der Aalst WMP. Discovering Block-Structured Process Models from Event Logs - A Constructive Approach. In: Colom J-M, Desel J. *Proceedings. Application and Theory of Petri Nets and Concurrency - 34th International Conference*; Jun 2013; Mailand, Italien. Berlin, Heidelberg: Springer; 2013. S. 311–329. doi: 10.1007/978-3-642-38697-8_17.
- [18] Leemans SJJ, Fahland D, van der Aalst WMP. Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour. In: Lohmann N, Song M, Wohed P. *Revised Papers. Business Process Management Workshops. BPM 2013 International Workshops*; Aug 2013; Beijing, China. Basel, Schweiz: Springer International Publishing; 2014. S. 66–78.
- [19] Tax N, Sidorova N, van der Aalst WMP. Discovering more precise process models from event logs by filtering out chaotic activities. *Journal of Intelligent Information Systems*. 2019; 52 (1): S. 107–139. doi: 10.1007/s10844-018-0507-6.
- [20] van Dongen BF, de Medeiros AKA, Verbeek HMW, Weijters AJMM, van der Aalst WMP. The ProM Framework: A New Era in Process Mining Tool Support. In: Ciardo G, Darondeau P. *Applications and theory of Petri Nets 2005. 26th international conference, ICATPN 2005*; Jun 2005; Miami, Florida. Berlin, Heidelberg: Springer; 2005. S. 444–454.
- [21] Berti A, van Zelst SJ, van der Aalst WMP. *Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science*. 2019. Online verfügbar unter <http://arxiv.org/pdf/1905.06169v1>., zugegriffen am 20.04.2022
- [22] Stierle M, Viner D, Matzner M. *FAU 2021 Process Mining Survey*. Erlangen-Nürnberg: FAU University Press; 2021. Online verfügbar unter <https://www.processmining-software.com/resources/>, zugegriffen am 20.04.2022
- [23] Abohamad W, Ramy A, Arisha A. A hybrid process-mining approach for simulation modeling. In: Chan

- WKV, D'Ambrogio A, Zacharewicz G, Mustafee N, Wainer G, Page E. *WSC '17: Proceedings of the 2017 Winter Simulation Conference*; Dez 2017; Las Vegas, Nevada. Piscataway, NJ: IEEE Press; 2017. S. 1527–1538.
- [24] Aguirre S, Parra C, Alvarado J. Combination of Process Mining and Simulation Techniques for Business Process Redesign: A Methodological Approach. In: Cudré-Mauroux P, Ceravolo P, Gašević D. Data-driven process discovery and analysis. *International Symposium, SIMPDA 2012*; Jun 2012; Campione d'Italia, Italien. Berlin, Heidelberg: Springer. 2012. S. 24–43.
- [25] Camargo M, Dumas M, González-Rojas O. Automated discovery of business process simulation models from event logs. *Decision Support System*. 2020; 134: S. 113284. doi: 10.1016/j.dss.2020.113284.
- [26] Gawin B, Marcinkowski B. How Close to Reality is the „as-is“ Business Process Simulation Model? *Organizacija*. 2015; 48 (3): S. 155–175. doi: 10.1515/orga-2015-0013.
- [27] Jadrić M, Pašalić IN, Ćukušić M. Process Mining Contributions to Discrete-event Simulation Modelling. *Business Systems Research Journal*. 2020; 11 (2): S. 51–72. doi: 10.2478/bsrj-2020-0015.
- [28] Khodyrev I, Popova S. Discrete Modelling and Simulation of Business Processes Using Event Logs. *Procedia Computer Science*. 2014; 29: S. 322–331. doi: 10.1016/j.procs.2014.05.029.
- [29] Langer A, Ortmeier C, Martin NL, Abraham T, Herrmann C. Combining Process Mining And Simulation In Production Planning. In: Nyhuis P, Herberger D, Hübner M. *Proceedings of the 2nd Conference on Production Systems and Logistics (CPSL 2021)*. Hannover: Institutionelles Repositorium der Leibniz Universität Hannover; 2021. S. 264-273.
- [30] Martin N, Depaire B, Caris A. The use of process mining in a business process simulation context: Overview and challenges. In IEEE. *2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2014)*; Dez 2014; Orlando, Florida. Piscataway, NJ: IEEE; 2014. S. 381–388.
- [31] Mărușter L, van Beest NRTP. Redesigning business processes: a methodology based on simulation and process mining techniques. *Knowledge and Information Systems*. 2009; 21 (3): S. 267–297. doi: 10.1007/s10115-009-0224-0.
- [32] Mesabbah M, McKeever S. Presenting a Hybrid Processing Mining Framework for Automated Simulation Model Generation. In: Rabe M, Juan AA, Mustafee N, Skoogh A, Jain S, Johansson B. *WSC '18: Proceedings of the 2018 Winter Simulation Conference*; Dez 2018; Gothenburg, Schweden. Piscataway, NJ: IEEE Press; 2018. S. 1370–1381.
- [33] Pourbafrani M, van Zelst SJ, van der Aalst WMP. Supporting Automatic System Dynamics Model Generation for Simulation in the Context of Process Mining. In: Klein G. *Business Information Systems. 23rd International Conference, BIS*; Jun 2020; Colorado Springs, Colorado. Basel, Schweiz: Springer International Publishing. 2020. S. 249–263. 2020
- [34] Rozinat A, Mans RS, Song M, van der Aalst WMP. Discovering simulation models. *Information Systems*. 2009; 34 (3): S. 305–327. doi: 10.1016/j.is.2008.09.002.
- [35] van der Aalst WMP. Process Mining and Simulation: A Match Made in Heaven! In: *SummerSim '18: Proceedings of the 50th Computer Simulation Conference*; Jul 2018; Bordeaux, Frankreich. San Diego, CA: Society for Computer Simulation International. 2018. S. 1–12.
- [36] IEEE XES Working Group. *IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams*. New York: IEEE Standard; 2016 doi: 10.1109/IEEESTD.2016.7740858
- [37] Freund J, Rücker B. *Praxishandbuch BPMN. Mit Einführung in DMN. 6. aktualisierte Auflage*. München: Hanser; 2019.
- [38] van der Aalst WMP. A practitioner's guide to process mining: Limitations of the directly-follows graph. *Procedia Computer Science*. 2019; 164: S. 321–328. doi: 10.1016/j.procs.2019.12.189.
- [39] L'Ecuyer P. Random numbers for simulation. *Commun. ACM*. 1990; 33 (10): S. 85–97. doi: 0.1145/84537.84555.
- [40] Stolipin J, Wenzel S. Ontologiebasierte Methodik zur Unterstützung der Nachnutzung von Simulationswissen. In: Putz M, Schlegel A. *Tagungsband Simulation in Produktion und Logistik*. Auerbach: Verlag Wissenschaftliche Scripten; 2019. S. 193–202
- [41] Rabe M, Spieckermann S, Wenzel S. *Verifikation und Validierung für die Simulation in Produktion und Logistik. Vorgehensmodelle und Techniken*. 1. Aufl. Berlin, Heidelberg: Springer; 2008.
- [42] Furmans K, Hanschke T, Möller DPF, Rabe M, Wenzel S, Zabel A, Zisgen H. *Simulation für Produktion und Logistik – VDI-Thesen und Handlungsfelder*. Düsseldorf: VDI; 2021.
- [43] van der Aalst WMP. Object-Centric Process Mining: Dealing with Divergence and Convergence in Event Data. In: Ölveczky PC, Salaün G. *Software Engineering and Formal Methods. 17th International Conference, SEFM 2019*; Sep 2019; Oslo, Norwegen. Basel, Schweiz: Springer International Publishing; 2019. S. 3–25.
- [44] Ghahfarokhi AF, Park G, Berti A, van der Aalst WMP. OCEL: A Standard for Object-Centric Event Logs. In: Bellatreche L, Dumas M, Karras P, Matulevičius R, Awad A, Weidlich M, Ivanovic M, Hartig O. *New Trends in Database and Information Systems, Bd. 1450*. Basel, Schweiz: Springer International Publishing; 2021. S. 169–175.

- [45] Adams JN, van der Aalst WMP. Precision and Fitness in Object-Centric Process Mining. In: Di Ciccio C, Di Francescomarino C, Soffer P. *2021 3rd International Conference on Process Mining (ICPM)*; Nov 2021; Eindhoven, Niederlande. Piscataway, NJ: IEEE; 2021. S. 128-135. doi: 10.1109/icpm53251.2021.9576886.

Experimentelle Analyse zur Einschleusplanung in der Halbleiterfertigung mittels einer stochastischen Rückwärtssimulation

Christoph Laroque^{1*}, Madlene Leißau¹,
Wolfgang Scholl², Germar Schneider², Patrick Preuß²

¹Industry Analytics, Westsächsische Hochschule Zwickau, Kornmarkt 1, 08056 Zwickau, Deutschland;
*christoph.laroque@fh-zwickau.de

²Infineon Technologies Dresden GmbH & Co. KG, Königsbrücker Straße 180, 01099 Dresden, Deutschland

Abstract. Manufacturing companies are experiencing many challenges regarding customer-oriented and on-time production, especially in the context of an intensified global business and while advancing the digital transformation. Accordingly, the ongoing development and deployment of Industry 4.0 solutions for customisable products in small batch sizes, not only pose new problems for work preparation, but also for operative production planning in connection with high cost, time, and quality pressure. Modern, complex, and highly automated production systems (in this case from the semiconductor domain) must be operated close to an optimal operating state to be economically reasonable. Promised delivery dates and throughput times specified in contracted service agreements must be ensured and require a permanent and effective adjustment of production planning and control in daily execution. All other general conditions of an economic production remain unchanged. Today, the question of whether the production target is realistic and whether all promised delivery dates are met are still answered with rather simple backward-oriented approaches, mostly without considering uncertainties, stochastic behaviour of the manufacturing system or alternatives that arise during operation. As shown in previous publications by the authors, these questions can be answered in more detail and more resilient using a backward-oriented discrete event-based simulation approach (SimBack). This article presents additional findings. The results show and deepen the impression, that the SimBack approach can successfully solve scheduling questions for customer-specific orders in a real-world environment.

1 Motivation

Der intensive globale Wettbewerb und die fortschreitende digitale Transformation sowie eine kundenorientiertere und termingerechtere Produktion und Lieferung sind entscheidende Wettbewerbsfaktoren für produzierende Unternehmen über viele Branchen hinweg. Die stetige Weiterentwicklung der Vision von Industrie 4.0 – zukünftig auch Industrie 5.0 [1] – und das Konzept einer "smarten Fabrik" für individualisierbare Produkte in kleinen Losgrößen stellen die Arbeitsvorbereitung sowie die operative Produktionsplanung vor immer neue Herausforderungen. Hochautomatisierte Produktionsanlagen müssen in einem Betriebszustand nahe am Optimum betrieben werden, um wirtschaftlich erfolgreich zu sein. Darüber hinaus müssen zugesagte Liefertermine und Durchlaufzeiten aus Rahmenverträgen (Service Level Agreements) eingehalten werden und erfordern eine permanente (effektive) Anpassung der Produktionsplanung und -steuerung im täglichen Betrieb.

Zur Herstellung und Erhaltung eines solchen (weitmöglichst) „optimalen Betriebszustands“ müssen sich Unternehmen fortwährend über die konkreten Stell-schrauben einer wirksamen Anpassung der Produktionsplanung und -steuerung bewusst sein. Ansonsten können kleinste Unsicherheiten im Rahmen von Produktionsplanung und -steuerung sowie daraus resultierender Anpassungen unerwartete Rückwirkungen auf die Leistung des gesamten (Produktions-)Systems sowie eine Fehlnutzung monetärer und zeitlicher Ressourcen zur Folge haben. Produktionsplanung und -steuerung müssen aber auch anpassbar bleiben, um gleichermaßen eine bestimmte Flexibilität in Bezug auf die zu planenden

Aufträge zu erhalten, um auf kurzfristige Kunden- und Marktänderungen noch reagieren zu können. Die damit verbundenen Risiken sind im laufenden Betrieb für Unternehmen zumeist nicht vollumfänglich oder gar nicht abzuschätzen; Methoden und Werkzeuge simulationsgestützter Optimierung erfahren hier mehr und mehr Bedeutung. Die andauernde Digitalisierung der Produktionsplanung und -steuerung unterstreicht in diesem Zusammenhang den stetig wachsenden Stellenwert der Simulation [2][3][4]. Demgemäß „zeigt sich regelmäßig, dass der Einsatz der Simulation bei der Planung von komplexen dynamischen Produktions- und Logistiksystemen zu abgesicherten und besser nachvollziehbaren Planungsergebnissen führt“ [4].

Im Rahmen des EU-ECSEL-Forschungsprojektes iDEV40 (Präsentation des Gesamtprojektes unter www.idev40.eu) wurden Einsatzszenarien der rückwärtsorientierten Materialflusssimulation in der Halbleiterfertigung erschlossen. Während die prinzipielle Machbarkeit des Ansatzes auf Fabrikebene bereits mehrfach nachgewiesen werden konnte [5][6], galt es im Vorhaben, die Methode an die Besonderheiten der Halbleiterfertigung anzupassen und in einem praxisnahen Modell zu verproben.

Gegenüber anderen Branchen sind die Produktionssysteme und -prozesse der in dieser Arbeit adressierten Halbleiterfertigung durch ein überdurchschnittlich hohes Maß an Komplexität charakterisiert. Im Hinblick auf die Prozessstabilität gelten die eingesetzten Fertigungstechnologien im Mikro- und Nanometerbereich als besonders sensitiv und beinhalten daraus resultierend sehr komplexe Steuerungslogiken. In Abhängigkeit verschiedener im Vorfeld festgesetzter Produkt- und Prozesskriterien durchlaufen einzelne Produktionslose zum Teil mehr als 1000 Prozessschritte. Einzelne Produktionslose werden mitunter mehrfach sowie mit hohem Automationsgrad unter Reinraumbedingungen über spezielle und teilweise dieselben Maschinen und Transportrouten prozessiert (Re-Entry-Cycles). Die resultierende Komplexität der Produktionssysteme und -prozesse führt folglich nicht selten zu einem Ausschuss von Zwischenprodukten in relevanter Größenordnung, den es dann (kurzfristig) durch zusätzliche Einschleusungen auszugleichen gilt.

Nach einer kurzen Darstellung des wissenschaftlichen Stands der Technik sowie einer Erläuterung des prinzipiellen Lösungsansatzes sollen in diesem Beitrag im Schwerpunkt aktuelle Experimentiererergebnisse eines „real-world use-case“ unter Verwendung des Simulationswerkzeuges AutoSched AP detailliert beschrieben

und dargestellt werden. Die abschließende Zusammenfassung beschreibt das weitere Vorgehen der Autoren nach Abschluss des EU-Projektes im Oktober 2021.

2 Rückwärtssimulation

In Ergänzung zu konventionellen Lösungsverfahren der gemischt-ganzzahligen Optimierung, verschiedenen Heuristiken oder der einfachen Vorwärts- und Rückwärtsterminierung (mit oder ohne Kapazitätsbeschränkungen) beschäftigt sich dieser Beitrag mit einem Ansatz der diskreten ereignisorientierten Simulation, wobei Entscheidungs- und Planungsprobleme einer rückwärts gerichteten Terminierung unterzogen werden. Ein solcher Ansatz wird nachfolgend als Rückwärtssimulation verstanden und meint die Umkehrung des Materialflusses einer Simulation, seiner Ablauflogik samt der implementierten Steuerungs- und Prioritätsregeln (Figure 1) und die daraus resultierende rückwärtsgerichtete Ausführung des Simulationslaufes selbst. Anknüpfend an eine rückwärtsgerichtete Ausführung einer Simulation lassen sich nach Jain und Chan [7] sowie Laroque [8] fundierte Aussagen über zu erreichende Zielgrößen im Kontext zugesagter Liefertermine treffen. Die Rückwärtssimulation ist ein effizientes Werkzeug für die Implementierung der Verfahrensweisen einer (einfachen) Rückwärtsterminierung. Demgegenüber kann die Rückwärtssimulation die Lösungsqualität eines herkömmlichen Produktionsplanungs- und Terminierungsmechanismus verbessern und die Vorteile bezüglich der Ausführungsgeschwindigkeit simulationsbasierter Terminierungsansätze (bspw. gegenüber optimierenden Verfahren) wirksam werden lassen [7].

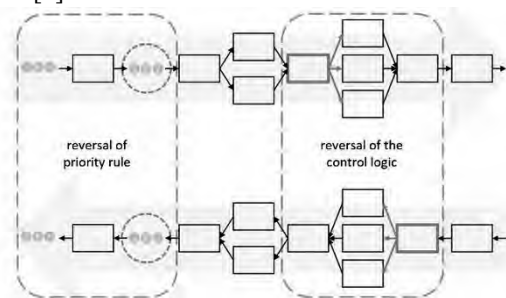


Figure 1: Umkehrung der Ablauflogik einer Simulation samt der implementierten Steuerungs- und Prioritätsregeln

Die vorwärtsgerichtete Ausführung einer Simulation und die Termin- und Reihenfolgeplanung orientieren sich an denselben Prinzipien wie die Verfahrensweise einer

(einfachen) Vorwärtsterminierung. Entsprechend bestimmen eine Termin- und Reihenfolgeplanung durch fortlaufende Addition produktspezifischer Bearbeitungszeiten, d. h. einzelner Vorgangslaufzeiten die frühestmöglichen Zwischen- und Endtermine einzelner Arbeitsvorgänge und die möglichen Fertigstellungstermine einzelner Aufträge [9][10]. Im Gegensatz dazu beginnen Termin- und Reihenfolgeplanung einer rückwärtsorientierten Simulation mit dem Ziel einer Nullverspätung und danach einer zeitlichen Realisierbarkeit zugesagter Liefertermine mit dem spätestmöglichen Fertigstellungstermin sowie der höchsten Dispositions-/Produktionsstufe [6][9][10][11]. Für die Art und Weise einer Auftragsfreigabe an einer Maschine beziehungsweise Ressource werden Aufträge, welche innerhalb einer rückwärtsgerichteten Ausführung an erster Stelle ausgewählt und bearbeitet werden, in einer vorwärtsgerichteten Ausführung auf der Zeitachse später betrachtet [11].

Eine solche Inversion oder Linksverschiebung auf der Zeitachse im Sinne einer vorwärtsgerichteten Simulation ist nach der rückwärtsgerichteten Simulation zur Generierung eines Einschleusplans durchzuführen, um die resultierende Lösungsmenge validieren zu können. Entsprechend lassen heutige Simulationswerkzeuge und der implementierte (vorwärtsorientierte) Planungszeitraum keine eins-zu-eins beziehungsweise zeitlich absteigende Einlastung im Sinne einer rückwärtsgerichteten Simulation zu. So kann angenommen werden, dass ein Auftrag *A1* mit einem (originären) Fertigstellungstermin in 5 Tagen in einer vorwärtsgerichteten Simulation Vorrang gegenüber einem Auftrag *A2* mit einem Fertigstellungstermin in 15 Tagen besitzt, diese Aufträge gemäß einer Rechtsverschiebung und einer rückwärtsgerichteten Simulation jedoch in umgekehrter Reihenfolge (ausgehend des spätestmöglichen Fertigstellungstermins) zu betrachten sind. Im Anschluss konkretisiert eine Linksverschiebung auf der Zeitachse im Sinne einer vorwärtsgerichteten Simulation und ferner einer Anwendung und Validierung der resultierenden Lösungsmenge die Differenz eines (originären) Fertigstellungstermins und einer simulierten Durchlaufzeit (nach rückwärtsgerichteter Ausführung). Eine Nullverspätung und danach eine zeitliche Realisierbarkeit zugesagter Liefertermine ist erreicht, wenn der Zeitpunkt der (simulierten) Fertigstellung früher liegt als der zugesagte Liefertermin [11][12]. Erste Anwendungsstudien, bei denen Aufträge mit Hilfe einer Rückwärtssimulation zeitlich rückwärtsgerichtet eingeplant werden, sind bereits seit den 1990er Jahren verfügbar [7][12][13][14].

Das Vorhaben und die anschließende Modellierung einer rückwärtsgerichteten Ausführung einer Simulation verlangen im Vorfeld einige sorgfältige Überlegungen, um eine korrekte Umkehrung des Simulationsmodells vornehmen zu können und sich von der Denkweise einer vorwärtsgerichteten Modellierung zu lösen. Diese Überlegungen beziehen sich insbesondere auf eine Umkehrung einzelner Produktionsprozessfolgen, beispielsweise einer Montage in eine Demontage [7]. Zum abstrahierten Nachempfinden des Verhaltens eines bereits bestehenden und/oder zukünftig beabsichtigten (realen) Systems innerhalb einer vorwärtsgerichteten Simulation sind in der rückwärtsgerichteten Modellierung ebenso die implementierten Steuerungs- und Prioritätsregeln zu übertragen. Steuerungs- und Prioritätsregeln lassen sich nicht immer eins-zu-eins in das korrespondierende Rückwärts-Pendant invertieren; hier sind spezifische Regeln für das Rückwärtsmodell zu entwickeln und das resultierende Modell auch diesbezüglich durch die vorwärtsgerichtete Simulation zu validieren. Eine experimentelle Untersuchung kann die Lösungsqualität eines solchen Ansatzes bezogen auf zeitlich rückwärtsgerichtete Entscheidungs- und Planungsprobleme weiter steigern. Ferner lassen sich nach Graupner et al. [6] die Vorteile einer vorwärts- und rückwärtsgerichteten Simulation in einer kombinierten Ausführung vereinen und mögliche Modellierungsdiscrepanzen bezogen auf die rückwärtsgerichtete Modellierung ausgleichen.

Die Modellierung einer rückwärtsorientierten Ausführung einer Ablaufsimulation zieht immer eine Umkehrung der Quelle-Senke-Beziehung nach sich. Konkret heißt das, dass Aufträge oder Lose an den Stellen in das System eingeschleust werden, an der sie dieses in der vorwärtsorientierten Ausführung verlassen. Andersherum verlassen sie das Rückwärts-Pendant an den Einschleusstellen der vorwärtsgerichteten Ablaufsimulation („from product to raw material“). Die Rückwärtssimulation ist dennoch nicht als reine ‚Umkehrfunktion‘ der Vorwärtssimulation zu verstehen. So verweisen Ying und Clark [14] darauf, dass Vorwärts- und Rückwärtssimulation zur selben jeweils berechneten Simulationszeit nicht denselben Zustand aufweisen müssen (die Rückwärtssimulation ist kein „Zurückspulen“ des Vorwärtsmodells, sondern eine eigenständige Abstraktion).

Neben der Verkürzung der Durchlaufzeit und einer Erhöhung des Durchsatzes sowie der aktuellen Verfügbarkeit und Wertschöpfung einzelner Produktionsanlagen (Overall Equipment Effectiveness) gilt es im Zuge einer rückwärtsberechneten Ausführung einer

Simulation unerwünschte Wartezeiten, die im Verlauf einer reinen vorwärtsgerichteten Simulation entstehen, auf ein Minimum zu reduzieren [6][7].

3 Integration in betriebliches Simulationswerkzeug und aktuelle Ergebnisse

Nachdem in Vorarbeiten bereits ein vereinfachtes Modell des nachfolgend beschriebenen „real-world use case“ unter Verwendung des Simulationswerkzeuges AnyLogic modelliert und bezüglich einer rückwärtsorientierten Materialflusssimulation untersucht wurde [15][16], galt es den methodischen Ansatz zur Generierung einer Einschleusplanung durch Rückwärtssimulation innerhalb des Simulationswerkzeuges AutoSched AP beim Praxispartner zu integrieren und zu erproben.

Das betrachtete Modell des Vormontageprozesses besteht im Wesentlichen aus 13 Stationengruppen mit bis zu 27 Stationen pro Stationengruppe und einer verfügbaren Kapazität von bis zu 8 Losen pro Station. Eine zusätzliche Stationengruppe *DUMMY* ist eine starke Abstraktion der abzubildenden Realität und bezogen auf die Anzahl baugleicher Anlagen eine Ausnahme (Stationenanzahl = 1000); sie wurde gewählt, um künstlich entstehende Engpässe in der Simulation des Materialflusses zu vermeiden.

Entsprechend der implementierten Steuerungs- und Prioritätsregeln des realen Vorbildes gelten für das nachempfundene System und gemäß einer Bearbeitungsliste innerhalb einer Stationengruppe (*FWLRANK*) und einer Bearbeitungsliste einzelner Stationen (*RULE*) konkrete Prioritätsregelverfahren. Bezogen auf das Prioritätsregelverfahren *Earliest Due Date* (*EDD*, *rank_EDD*) haben diejenigen Lose Vorrang, die den zeitlich frühesten Termin der Fertigstellung aufweisen, während das Prioritätsregelverfahren *First In – First Out* (*FIFO*, *rule_First*) die Lose gemäß des spezifischen Eintreffens an der Anlage bearbeitet. Das Prioritätsregelverfahren *Same Setup* (*SSU*, *rule_SSU*) sortiert die Lose in der Warteschlange in Abhängigkeit des Rüstzustands der Anlagen.

Der Prozessfluss eines jeden Loses und die eindeutige Aufeinanderfolge von Operationen innerhalb des Produktionsprozesses werden über einen Arbeitsplan bestimmt. Entsprechend werden 92 Arbeitspläne mitsamt dazugehöriger Operationen und Prozesszeiten definiert. Während ein solcher Arbeitsplan bis zu 21 Operationen

umfassen kann, sind die Operationen eindeutig einem Workcenter zuordenbar. Die Workcenter sind im Gegensatz zu einer Abstraktion der Stationengruppe *DUMMY* detailliert im Modell abgebildet und können im Anschluss an die Simulation über einen speziellen Report (subset.rdf) ausgewertet werden.

Die Zeit, die für die Vorbereitung von Anlagen für die Bearbeitung einzelner Lose gemäß eines Prioritätsregelverfahrens *Same Setup* erforderlich sein kann, ist über eine Zuordnung einer bestimmten Konfiguration zu einem gleichnamigen Arbeitsplan und einer Zeitmenge gegeben. Darüber hinaus werden auch Verfügbarkeit und Ausfallzeiten einer Stationengruppe mit Hilfe einer dafür vorgesehenen Kalenderzuweisung definiert. Während für das Modell des zugrunde liegenden Vormontageprozesses lediglich (stochastische) Ausfallzeiten zu definieren sind, lassen sich auf diese Weise, beispielsweise, auch (vorbeugende) Wartungszeiten und Arbeitsschichten definieren.

Neben der Zuordnung eines Arbeitsplanes besitzen Lose und die darin enthaltenen Wafer (Produkte) einen Grundtyp (*Basic Type*): eine prinzipielle Beschreibung der Größe und Anzahl der auf einem Wafer enthaltenen Chips. Insgesamt sind 605 Grundtypen auf die 92 Arbeitspläne verteilt.

Die nachstehende Abbildung (Figure 2) soll die Komplexität des zugrunde liegenden Produktionsprozesses anhand der implementierten Arbeitspläne über die Workcenter noch einmal veranschaulichen. Hierbei werden auch die einleitend erwähnten Re-Entry-Cycles deutlich.

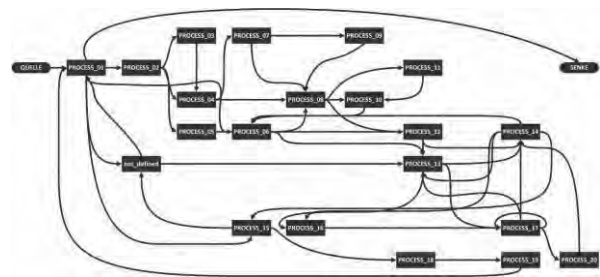


Figure 2: Prozessabfolge über die Workcenter gemäß der zugrunde liegenden Arbeitspläne

Ausgangspunkt des später beschriebenen vollfaktoriellen Experimentdesigns ist die Betrachtung der Kapazitäten an den Stationen von drei zusammengehörenden Stationengruppen und der Auswirkung der Kapazitätsbeschränkungen auf die Gesamtleistung des Systems, beispielsweise aufgrund von unvorhergesehenen Ausfällen

oder (vorbeugenden) Wartungszeiten. Die betrachteten Stationengruppen bilden Prüf- und Inspektionsstationen ab, wo mehrere manuelle und/oder automatische Inspektionsschritte ablaufen (darum bietet sich eine gemeinsame Betrachtung hier an).

Im Folgenden wird ein vollfaktorieller Versuchsplan aufgestellt, durch den sich die ganzheitliche Untersuchung aller Wechselwirkungsbeziehungen der Einflussgrößen unter sonst gleichen Bedingungen anstellen lässt (*ceteris paribus*). Gemäß der Einflussgrößen *STNCAP_01*, *STNCAP_02* und *STNCAP_03*, die im Folgenden die zur Verfügung stehenden Kapazitäten an den Stationen der gleichnamigen Stationengruppen beschreiben, werden je 3 Faktoreinstellungen (*gering*, *mittel* und *hoch*) konkretisiert. Während ein Faktor *gering* danach etwa 42,86 beziehungsweise 50,00 Prozent und eine Stellgröße *mittel* etwa 71,43 beziehungsweise 75,00 Prozent zur Verfügung stehende Kapazitätseinheiten annehmen, beziffert eine Stellgröße *hoch* 100 Prozent zur Verfügung stehender Kapazitätseinheiten. Darauf aufbauend zeigt die nachstehende Tabelle (Table 1) noch einmal eine Übersicht über die verwendeten Einflussgrößen sowie die jeweils gewählten Stellgrößen mitsamt einer Definition gemäß einer zur Verfügung stehenden Kapazität.

Einflussgrößen	Stellgrößen des Faktors
STNCAP_01	gering (3), mittel (5), hoch (7)
STNCAP_02	gering (3), mittel (5), hoch (7)
STNCAP_03	gering (4), mittel (6), hoch (8)

Table 1: Definition der Stellgrößen eines Faktors je Einflussgröße entsprechend der zur Verfügung stehenden Kapazität an den Stationen einer Stationengruppe

Durch die im Modell implementierten (stochastischen) Ausfallzeiten an den Stationengruppen werden die Versuchsläufe des jeweiligen Versuchspunktes des vollfaktoriellen Versuchsplans repliziert und unter Verwendung verschiedener Seedwerte durchgeführt. Die verschiedenen Versuchspunkte verwenden jeweils diesselben Seedwerte. Eine im Vorfeld getätigte Durchführung für mehrerer Konfidenzläufe für einzelne Versuchspunkte und eine Auswertung dieser zeigte in diesem Modell aber auch nur geringe Schwankungen.

Die Versuchsläufe werden mittels des auf die Halbleiterfertigung spezialisierten Simulationswerkzeuges AutoSched AP auf einem Quad-Core-Prozessor Intel

Core i5-9300H und einem physischen Arbeitsspeicher von 16 GB gestartet. Die resultierenden Simulationsdaten werden im Falle der rückwärtsgerichteten Ausführung des Simulationsmodells manuell und im Falle der vorwärtsgerichteten, validierenden Ausführung des Simulationsmodells automatisiert über einen KNIME-Workflow (Figure 3) aufbereitet und in einem Bericht (*Report*) ausgewertet.

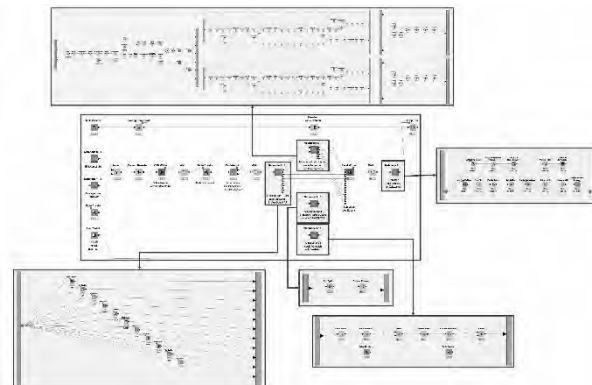


Figure 3: KNIME-Workflow für die vorwärtsgerichtete Ausführung des Simulationsmodells und eine nachstehende Datenaufbereitung sowie -auswertung

Die nachfolgenden Ergebnisse beziehen sich auf einen kurzfristigen Planungshorizont gemäß der formalen Zusage von Lieferterminen (*DUE*) über einen Zeitraum vom 08. März 2021 bis zum 22. März 2021. Für die reale Einlastung (*REAL*) und ein Zeitintervall von je 2 Tagen um den *DUE* $(-48, 48]$ ließen sich in der Praxis insgesamt 652 von 1197 Aufträge (rund 54,47 Prozent) fertigstellen. Ein solches Zeitintervall $(-48, 48]$ umfasst sämtliche Aufträge mit einer Abweichung von > -48 bis ≤ 48 Stunden und kann (laut Praxispartner) als hinreichend gegenüber der Einhaltung zugesagter Liefertermine angesehen werden. Für ein Zeitintervall $(-12, 12]$ und damit unmittelbar um eine Nullverspätung herum lassen sich insgesamt 158 von 1197 Aufträge (rund 13,20 Prozent) abschließen. Im Vergleich dazu können bei einer kombinierten Ausführung des Simulationsmodells im Sinne einer stochastischen Rückwärtssimulation insgesamt 923 von 1197 Aufträge (rund 77,11 Prozent) in einem Zeitintervall $(-48, 48]$ im Mittel und bei voller Kapazität fertiggestellt werden. In dem Zeitintervall $(-12, 12]$ lassen sich in dieser Konfiguration 211 von 1197 Aufträgen (rund 17,62 Prozent) fertiggestellt werden. Entsprechend können gegenüber einer realen Einlastung (*REAL*) im Mittel etwa 263 Aufträge (rund 22,04 Prozent) mehr gemäß einer Abweichung von > -48 bis ≤ 48 Stunden abgeschlossen

werden. In Summe heißt dies, dass sich in einem Zeitintervall $(-48, 48]$ im Mittel insgesamt 915 von 1197 Aufträgen (rund 76,51 Prozent) fertigstellen lassen. Weiterhin werden in einem Zeitintervall $(-12, 12]$ im Mittel insgesamt 183 von 1197 Aufträge (rund 15,34 Prozent) abgeschlossen.

Die Anwendung und die Ergebnisse des methodischen Ansatzes der Rückwärtssimulation zeigen damit auf, dass die stochastische Rückwärtssimulation unter den Spezifika der Halbleiterfertigung die avisierte Leistungsfähigkeit als Instrument zur Produktionsplanung erbringen kann. Die zu fertigenden Aufträge lassen sich mittels einer Einschleusplanung durch die Rückwärtssimulation in Summe in einem deutlich geringeren Zeitintervall, d. h. mit einer Abweichung von > 24 bis ≤ 96 Stunden fertigstellen. Bei einer realen Einlastung werden demgegenüber im gleichen Zeitintervall lediglich 979 (rund 81,79 Prozent) abgeschlossen.

In der nachstehenden Abbildung (Figure 4) ist die Ist-Fertigstellung (*REAL* und *SIMBACK*) gegenüber der Plan-Fertigstellung einmal grafisch dargestellt. Der grau markierte Bereich kennzeichnet dabei das Zeitintervall $(-48, 48]$, das als hinreichend („pünktlich“) angenommen werden kann.

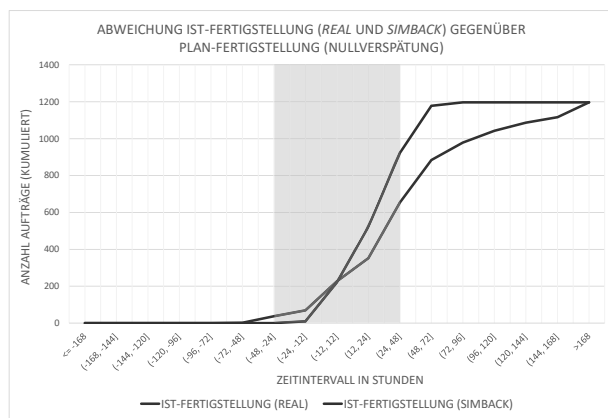


Figure 4: Abweichung einer Ist-Fertigstellung (*REAL* und *SIMBACK*) gegenüber einer Plan-Fertigstellung (Nullverspätung)

Die folgende Abbildung (Figure 5) soll darüber hinaus noch einmal die Ist-Fertigstellung (*REAL* und *SIMBACK*) und eine Plan-Fertigstellung bezogen auf konkreter Aufträge im Zeitverlauf darstellen und die resultierende Abweichung gegenüber einer Plan-Fertigstellung verdeutlichen.

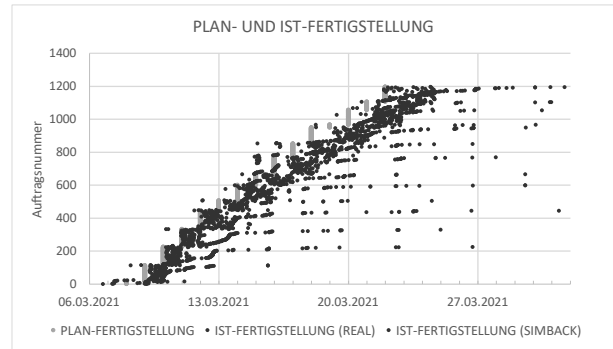


Figure 5: Abweichung einer Ist-Fertigstellung (*REAL* und *SIMBACK*) gegenüber einer Plan-Fertigstellung (Nullverspätung) anhand konkreter Aufträge

Die Ergebnisse nach den durchgeführten Versuchsläufen des vollfaktoriellen Versuchsplanes lassen in Summe darauf schließen, dass eine solche kombinierte Ausführung im Sinne einer stochastischen Rückwärtssimulation sinnvoll angewendet werden kann, Kapazitätsbeschränkungen sinnvoll berücksichtigt und damit als Instrument zur Produktionsplanung geeignet sein kann.

Die Versuchsläufe des vollfaktoriellen Versuchsplanes deuten auch im Mittel auf eine Verbesserung der Termintreue und einer in Summe geringeren Abweichung gegenüber einer Plan-Fertigstellung (Nullverspätung) hin.

Aus den Beobachtungen bezogen auf einen kurzfristigen Planungshorizont über einen Zeitraum von 2 Wochen und den Versuchsläufen des vollfaktoriellen Versuchsplanes lässt sich feststellen, dass die gewählten Kapazitätsbeschränkungen keine signifikante Wirkung auf eine Termintreue zugesagter Liefertermine zeigen. Der Interquartilsabstand (engl. Interquartile Range, IQR), d. h. der Abstand zwischen dem dritten und dem ersten Quartil zeigt eine gemessene Abweichung von 1,00 Prozent für eine Abweichung von > -48 bis ≤ 48 Stunden und 0,25 Prozent für eine Abweichung von > -72 bis ≤ 72 Stunden. Entsprechend ordnen sich zwischen dem dritten und dem ersten Quartil 50 Prozent der gemessenen Ergebnisse ein.

4 Ausblick

Die auf Basis eines “real-world use-case” erzeugten Ergebnisse zeigen, dass der methodische Ansatz zur Generierung einer Einschleusplanung durch Rückwärtssimulation unter auch unter Berücksichtigung stochastischer Modell-Einflüsse funktioniert und vielversprechende Ergebnisse liefern kann. Aufbauend auf Vorarbeiten

[15][16][17] mit dem Simulationswerkzeug AnyLogic zeigt dieser Beitrag die Anwendbarkeit des methodischen Ansatzes der Rückwärtssimulation in einem weiteren Simulationswerkzeug (AutoSched AP) anhand eines detaillierteren Modells des betrachteten Vormontageprozesses im praktischen Kontext. Das Modell grenzt sich durch eine höhere Komplexität und einen höheren Detaillierungsgrad, u. a. durch die implementierten Steuerungs- und Prioritätsregeln, von den Modellen in Vorarbeiten ab und kann dennoch die dort beschriebenen Ergebnisse hinsichtlich der Leistungsfähigkeit des methodischen Ansatzes der Rückwärtssimulation bestätigen.

Die Erprobung und Anwendung mehrerer Simulationsmodelle sowie die Durchführung und Analyse verschiedener Experimente im Rahmen des europäischen Forschungsprojektes iDEV40 zeigen, dass eine Rückwärtssimulation im Sinne einer kombinierten Ausführung des Simulationsmodells ein leistungsfähiges Werkzeug für die operative Produktionsplanung darstellen kann.

Im Anschluss an das Forschungsprojekt iDEV40 soll nun die Entwicklung des methodischen Ansatzes der Rückwärtssimulation hin zu einer zielgerichteten Datengenerierung und -auswertung anhand der Vorgehensweise des *Data Farming* entwickelt werden. Ziel ist die Verbesserung der Lösungsqualität, insbesondere in Bezug auf die Einhaltung zugesagter Liefertermine und weiterer relevanter Leistungskennzahlen. Die Verwendung der resultierenden Ergebnisdaten als Teil einer Trainingsphase von Verfahren des maschinellen Lernens (engl. *Machine Learning*) und die Bereitstellung eines leistungsfähigen Werkzeuges (Anwendungsphase) als operative Entscheidungsunterstützung für die Termin- und Reihenfolgeplanung in der Halbleiterindustrie sollen die Anwendbarkeit des Lösungsansatzes für unmittelbare Entscheidungsträger sicherstellen und den Zeit- und Ressourcenbedarf minimieren.

Ein Erproben der Methode des Transferlernens (engl. *Transfer Learning*) adressiert die Schwierigkeit, die auf maschinellen Lernen basierende Entscheidungsunterstützung und das dahinterstehende Vorhersagemodell neu mit problemspezifischen Daten (Simulations- und Echt-daten) trainieren zu müssen, sobald sich das zugrundeliegende System entscheidend verändert und/oder ähnliche Anwendungsfälle betrachtet werden sollen.

Danksagung

Der vorliegende Beitrag wurde teilweise durch das europäische Forschungsvorhaben iDEV40 finanziert, das vom ECSEL Joint Undertaking (JU) unter der Fördervereinbarung Nr. 783163 gefördert wird. Das JU erhält Unterstützung aus dem Forschungs- und Innovationsprogramm Horizon 2020 der Europäischen Union. Es wird von den Konsortiumsmitgliedern sowie Zuschüssen aus Österreich, Deutschland, Belgien, Italien, Spanien und Rumänien mitfinanziert. Die in dieser Veröffentlichung enthaltenen Informationen und Ergebnisse sind die der Autoren und spiegeln nicht zwingend die Meinung des ECSEL Joint Undertaking wider.

References

- [1] Breque M, De Nul L, Petridis A. *Industry 5.0 – Towards a sustainable, human-centric and resilient European Industry*. Luxembourg: Publications Office of the European Union; 2021.
- [2] Kühn, W. *Digitale Fabrik: Fabriksimulation für Produktionsplaner*. München/Wien: Carl Hanser Verlag; 2006.
- [3] März L, Weigert G. Simulationsgestützte Optimierung. In: März L, Krug W, Rose M, Weigert G, editors. *Simulation und Optimierung in Produktion und Logistik: Praxisorientierter Leitfaden mit Fallbeispielen*. Berlin: Springer; 2011, p 3-12.
- [4] Gutenschwager K, Rabe M, Spieckermann S, Wenzel S. *Simulation in Produktion und Logistik: Grundlagen und Anwendungen*. Berlin/Heidelberg: Springer Vieweg; 2017.
- [5] Arakawa M, Fuyuki M, Inoue I. *A Simulation-based Production Scheduling Method for Minimizing the Due-date-deviation*. International Transactions in Operational Research 9 (2); 2002. p 153-167.
- [6] Graupner TD, Bornhäuser M, Sihn W. Backward simulation in food industry for facility planning and daily scheduling. *Proceedings of the 16th European Simulation Symposium (ESS)*, 17th-20th October, Budapest (Hungary); 2004.
- [7] Jain S, Chan S. Experiences with Backward Simulation Based Approach for Lot Release Planning. *Proceedings of the 29th Winter Simulation Conference (WSC)*, 7th – 10th December, Atlanta (USA); 1997. p 773-780.
- [8] Laroque C. *Ein mehrbenutzerfähiges Werkzeug zur Modellierung und richtungsoffenen Simulation von wahlweise objekt- und funktionsorientiert gegliederten Fertigungssystemen* [dissertation]. Paderborn: Heinz-Nixdorf-Institut; 2007.
- [9] Dangelmaier W, Warnecke HJ. *Fertigungslenkung: Planung und Steuerung des Ablaufs der diskreten Fertigung*. Berlin: Springer; 1997.

- [10] Kurbel K. *Produktionsplanung und -steuerung im Enterprise Resource Planning und Supply Chain Management*. 6th Edition. München: Oldenbourg Wissenschaftsverlag; 2005.
- [11] Huang C, Wang H. Backward Simulation with Multiple Objectives Control. *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS)*, 18th -20th March 2009; Vol. II.
- [12] Watson EF, Medeiros DJ, Sadowski RP. Generating Component Release Plans with Backward Simulation. *Proceedings of the 25th Winter Simulation Conference (WSC)*, 12th-15th December, Los Angeles (USA); 1993. p 930-938.
- [13] Watson EF, Medeiros DJ, Sadowski RP. A simulation-based backward planning approach for order-release. *Proceedings of the 29th Winter Simulation Conference (WSC)*, 7th-10th December, Atlanta (USA); 1997. p 765-772.
- [14] Ying CC, Clark GM. Order release planning in a job shop using a bidirectional simulation algorithm. *Proceedings of the 26th Winter Simulation Conference (WSC)*, 11th-14th December, Orlando (USA); 1994. p 1008-1012.
- [15] Laroque C, Leißau M, Scholl W, Schneider G. Rückwärtssimulation als Instrument zur Produktionsplanung – Erkenntnisse aus einer praxisbezogenen Fallstudie. In: Franke J, Schuderer P, editors. *Simulation in Produktion und Logistik 2021*. Göttingen: Cuvillier Verlag; 2021, p 295-304.
- [16] Laroque C, Leißau M, Scholl W, Schneider G. Backward Simulation for Production Planning – Recent Advances in a Real-World Use-Case. *Proceedings of the 2021 Winter Simulation Conference (WSC)*, 13.-17. Dezember, Phoenix (AZ, USA); 2021.
- [17] Laroque C, Löffler C, Scholl W, Schneider G. Einsatzmöglichkeiten der Rückwärtssimulation zur Produktionsplanung in der Halbleiterfertigung. *Proceedings ASIM SST 2020*. Wien: RGE SIM Report AR 59; 2020. p 397-401.

Discrete event-based modeling of conveyors for dry bulk material

Peter Junglas^{1*}, Lukas Schmedes²

¹PHWT-Institut, PHWT Vechta/Diepholz, Thüringer Straße 3, 49356 Diepholz, Germany;

*peter@peter-junglas.de

²GRIMME Landmaschinenfabrik GmbH & Co. KG, Hunteburger Straße 32, 49401 Damme, Germany

Abstract. The simulation of transport processes, though inherently continuous, is often done in a discrete-event simulation environment. In the case of conveyors for dry bulk material, this can lead to modeling difficulties, especially regarding the coupling of two conveyors with different velocities. We will present a modeling approach solving such problems, describe an implementation in SimEvents and present results of systematic tests.

Introduction

In the simulation of production and logistic processes, the modeling of materials handling is of paramount importance. Though the detailed description of a transport process uses continuous functions of time, such as positions or mass flows, in the context of a complex simulation it is often simplified and modeled using only discrete events. But this reduction of complexity can lead to problems, because:

It is often important to model such entity transfer accurately since studies have shown that delays and inefficiencies in operations might be caused more by the need just to move things around rather than in actually doing the work. [1, p.345]

A simple conveyor belt moving discrete unit loads with constant velocity generally can be modeled easily enough. But building an adequate model for the transport of dry bulk material with a wide range of granularity and changing velocities of one belt or between belts is much more difficult.

Modifying the belt velocity is a standard method to adapt to a varying input mass flow. This can be used to utilise the full capacity of the conveyor at a lower speed, which will often decrease the power consumption [2], or to speed up in order to shorten the trans-

portation time. On the other hand, when transporting damageable goods like apples or potatoes, one could try to slow down the conveyor to guarantee a high quality of the goods.

In the following we will describe a discrete-event model of a conveyor for dry bulk material, which has a control input to change the velocity. A special focus will be on the coupling of conveyors running with different velocities, since this leads to modeling problems in a discrete environment. Finally the model will be implemented in SimEvents from Mathworks [3] and tested systematically.

The acceleration or deceleration of a highly loaded conveyor creates considerable tension in the belt, leading to local stretching or even breaking of the belt [4]. In this study we will neglect this effect and treat the belt as a rigid body with the same velocity everywhere.

1 Modeling of a single conveyor

A discrete-event model of a conveyor has to implement the delays of the incoming entities given by the conveyor length l and the velocity v . In addition it has to store the positions of all entities at the current (discrete) time in order to cope with entities of varying size l_E or with a time-dependent velocity (cf. Figure 1).

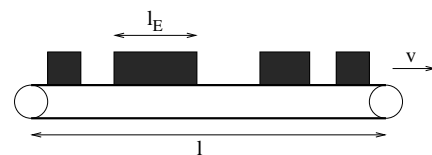


Figure 1: Simple conveyor model.

Such a component exists in many commercial discrete simulation environments such as Arena [1], SimEvents [3] or PlantSimulation [5]. The length of

the entities can either be defined as a fixed parameter or as an entity-specific attribute. The various programs have different additional features like a minimal distance between entities, accumulation of entities in case of blocking or incorporating acceleration phases at the beginning – all of which will not be considered in the following. Another difference is the exact procedure, how an entity enters (or leaves) the conveyor. For concreteness we will define that a (right-going) entity starts at position $x = 0$ with its left edge coinciding with the left edge of the conveyor, and leaves at position $x = l$, when its left edge coincides with the right edge of the conveyor.

For the transportation of dry bulk material the situation is a bit more complicated, because there are no easily identifiable entities and the transport process is inherently continuous: Due to a usually non-uniform production process and the inhomogeneity of the material the conveyor is filled with an incoming mass flow $\dot{m}_{in}(t)$, which leads to a line load $\lambda := \frac{\partial m}{\partial x}$ given by

$$\lambda(t, x) = \frac{1}{v} \dot{m}_{in}\left(t - \frac{x}{v}\right) \quad (1)$$

for a constant velocity v (cf. Figure 2).

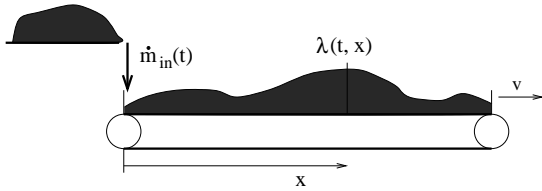


Figure 2: Conveyor model for bulk dry material.

The granularity of dry bulk material varies widely in practical applications, ranging from almost microscopic particles (powder) over small or medium sized particles (rice, apples) to large lumps (ore). The modeling methods used vary accordingly. Two extreme methods are described and compared in [6]: the microscopic description, where the movement of each particle is described separately, and a macroscopic representation, using a mass density and differential equations describing the conservation laws.

Both methods are computationally intensive, therefore in [7] a “mesoscopic” approach has been utilised, which is well suited for medium sized granularity and can be easily incorporated into standard discrete modeling environments. Basically, the continuous line load

is replaced by discrete entities E_i with mass m_i given by

$$m_i = \int_{(i-1)\Delta t}^{i\Delta t} \dot{m}_{in}(t) dt, \quad i \in \mathbb{N}^+$$

for an arbitrary fixed time interval Δt . Using (1) one gets for constant velocity v :

$$\begin{aligned} m_i &= v \int_{(i-1)\Delta t}^{i\Delta t} \lambda(i\Delta t, v(i\Delta t - t)) dt \\ &= \int_0^{v\Delta t} \lambda(i\Delta t, x) dx \end{aligned}$$

This shows that the conveyor is divided into compartments of equal length $l_E = v\Delta t$. For simplicity, one often chooses $l_E = l/N$ for $N \in \mathbb{N}^+$, so that entity E_i enters the conveyor at time $i\Delta t$, i. e. when its compartment is filled, and leaves at $(N + i)\Delta t$.

An alternative approach could be to use compartments of equal mass instead of equal length. This would lead to a more complex timing of events, which makes the interpretation of results more involved. Furthermore, some simulation environments (e. g. Arena) use conveyor components with a fixed cell size, which would make the implementation of this approach quite ugly.

2 Coupling of conveyors with different velocities

We will now analyse how one can combine two single conveyors with lengths l_1, l_2 and velocities v_1, v_2 . A mathematical description of the continuous process with additional input and output reservoirs at the end of each conveyor has been given in [8]. Considering only the case of constant (but different) velocities, one has:

$$\begin{aligned} \lambda_1(t, x) &= \frac{1}{v_1} \dot{m}_{in,1}\left(t - \frac{x}{v_1}\right) \\ \lambda_2(t, x) &= \frac{1}{v_2} \dot{m}_{in,2}\left(t - \frac{x}{v_2}\right) \end{aligned}$$

Connecting the conveyors directly, the output of conveyor 1 is the input of conveyor 2, therefore:

$$\begin{aligned} \dot{m}_{in,2}(t) &= v_1 \lambda_1(t, l_1) \\ \Rightarrow \lambda_2(t, x) &= \frac{v_1}{v_2} \lambda_1\left(t - \frac{x}{v_2}, l_1\right) \end{aligned}$$

For a continuous model one simply changes the line load λ_2 by the factor v_1/v_2 . This happens automatically in real life, when the bulk material gets thinned out or condensed on the second conveyor – as long as the capacity of the second belt is not exceeded.

But our discrete model runs into problems with the timing of the entities: Conveyor 1 sends entities at time intervals $\Delta t_1 = l_{E,1}/v_1$ to the entrance of conveyor 2, which in turn delivers entities at its output at generally different time intervals $\Delta t_2 = l_{E,2}/v_2$. Therefore one cannot maintain the idea of an entity defined as a fixed set of particles with given mass. Instead one identifies an entity with the content of a given compartment on a conveyor. Such a compartment is created and filled at the entrance of a conveyor, and emptied and destroyed at its exit.

The remaining task is now to compute the mass of such a newly created entity. According to the ratio

$$k := \frac{\Delta t_2}{\Delta t_1} = \frac{l_{E,2}}{l_{E,1}} \cdot \frac{v_1}{v_2}$$

one needs different strategies, how to cope with this problem. Such strategies should fulfil two requirements:

- mass conservation, i. e. incoming and outgoing masses should balance on a short time scale,
- homogeneity, i. e. the output mass distribution should closely follow the input mass values. For a constant incoming distribution this means, that the outgoing values shouldn't vary much.

If k is integer, one simply adds up the masses of k incoming entities to create an outgoing one, whereas if $1/k$ is integer, one distributes the mass of one incoming entity among $1/k$ outgoing entities. In all other cases one has to account for the unbalanced timing of input and output entities. Since the problem appears only at the connection of the two conveyors, we can concentrate on the second conveyor with its incoming values $m_{in,i}$ at times $i\Delta t_1$ and the corresponding outgoing values $m_{out,j}$ at times $j\Delta t_2$, disregarding the delay time of the second conveyor.

If $k > 1$ one can apply a simple collection strategy using a virtual bin, which accumulates all incoming masses into m_{acc} . A new output entity then empties the bin and gets the total accumulated mass. Table 1 shows how this works in an example with equal entity lengths $l_{E,1} = l_{E,2} = 1$ m, velocities $v_1 = 2.5$ m/s, $v_2 = 1$ m/s and constant incoming masses $m_i = 1$ kg.

i	j	t	m _{in}	m _{acc}	m _{out}
1	-	0.4	1	1	-
2	-	0.8	1	2	-
-	1	1.0	-	0	2
3	-	1.2	1	1	-
4	-	1.6	1	2	-
5	2	2.0	1	0	3
6	-	2.4	1	1	-

Table 1: Times and masses for example 1 ($k = 2.5$).

For $k < 1$ one has to use a partition strategy instead. The following strategy “A” defines the mass of a partition as

$$m_p = k m_{in}$$

each time a new entity enters, and sets outgoing entities accordingly. This simple scheme leads to a problem due to the timing, as can be seen in Table 2, which uses $v_1 = 1$ m/s, $v_2 = 1/0.35$ m/s: At $t = 1.75$ there is not enough mass available for the outgoing entity E_5 . But this can be cured easily by setting

$$m_{out} = \min(m_p, m_{acc})$$

Unfortunately strategy A has a serious drawback, as Ta-

i	j	t	m _{in}	m _p	m _{acc}	m _{out}
1	-	1.0	1	0.35	1	-
-	3	1.05	-		0.65	0.35
-	4	1.4	-		0.3	0.35
-	5	1.75	-		0	0.30
2	-	2.0	1	0.35	1	-

Table 2: Times and masses for example 2 ($k = 0.35$).

ble 3 shows using $v_1 = 1$ m/s, $v_2 = 1/0.8$ m/s: Though the mean ratio of input entities to output entities is 0.8, for a while the actual ratio is 1. Therefore the accumulated mass m_{acc} , which is just the difference between total input and total output mass, grows. This is in conflict with the primary goal of mass conservation on a short time scale. Even worse: When the following input entities are empty (i. e. $m_{in} = 0$), m_p is set to 0 and the accumulated mass stays in the internal bin. Strategy “B” tries to solve this problem by changing the value of the partition mass to

$$m_p = k m_{acc},$$

where the value is only computed, when an input entity

i	j	t	m _{in}	m _p	m _{acc}	m _{out}
1	-	1.0	1	0.8	1	-
-	2	1.6	-		0.2	0.8
2	-	2.0	1	0.8	1.2	-
-	3	2.4	-		0.4	0.8
3	-	3.0	1	0.8	1.4	-
-	4	3.2	-		0.6	0.8

Table 3: Times and masses for example 3 ($k = 0.8$), strategy A.

arrives. This will distribute the surplus value of m_{acc} among the next outgoing entities, thereby reducing the total mass imbalance, as can be seen in Table 4 for the values of example 3.

i	j	t	m _{in}	m _p	m _{acc}	m _{out}
1	-	1.0	1	0.800	1.000	-
-	2	1.6	-		0.200	0.800
2	-	2.0	1	0.960	1.200	-
-	3	2.4	-		0.240	0.960
3	-	3.0	1	0.992	1.240	-
-	4	3.2	-		0.248	0.992

Table 4: Times and masses for example 3 ($k = 0.8$), strategy B.

We will finally provide a mathematical description of the distribution strategy B to clarify possible open points and to guide the implementation. Starting point are the two positive time intervals Δt_1 , $\Delta t_2 = k \Delta t_1$ between arrival or departure of entities at the virtual connecting bin and the positive end time t_{end} of the simulation. We now define the sets

$$T_{in} = \{i \Delta t_1 \mid i \in \mathbb{N}^+\} \cap [0, t_{end}]$$

$$T_{out} = \{j \Delta t_2 \mid j \in \mathbb{N}^+\} \cap [0, t_{end}]$$

The function $m_{in}(t)$ is given for $t \in T_{in}$ (by a production process) and constant elsewhere. The functions m_{acc} , m_p and m_{out} will be defined on $T_{in} \cup T_{out}$, they are constant elsewhere. For simplicity we denote

$$f(t^-) := f(t - \varepsilon) \quad (\varepsilon > 0 \text{ sufficiently small}),$$

where “sufficiently small” means “smaller than the size of any open time interval from $T_{in} \cup T_{out}$ ”. We now start with

$$m_{acc}(0) = 0 \text{ kg}$$

and define:

$$m_p(t) = \begin{cases} k(m_{acc}(t^-) + m_{in}(t)) & | t \in T_{in} \\ \text{const.} & | \text{otherwise} \end{cases}$$

$$m_{out}(t) = \begin{cases} \min(m_p(t), m_{acc}(t^-)) & | t \in T_{out} \setminus T_{in} \\ \min(m_p(t), m_{acc}(t^-) + m_{in}(t)) & | t \in T_{in} \cap T_{out} \\ \text{const.} & | \text{otherwise} \end{cases}$$

$$m_{acc}(t) = \begin{cases} m_{acc}(t^-) + m_{in}(t) & | t \in T_{in} \setminus T_{out} \\ m_{acc}(t^-) - m_{out}(t) & | t \in T_{out} \setminus T_{in} \\ m_{acc}(t^-) + m_{in}(t) - m_{out}(t) & | t \in T_{in} \cap T_{out} \\ \text{const.} & | \text{otherwise} \end{cases}$$

One easily checks that these definitions reproduce the collection strategy and partition strategy B. Strategy A is a bit simpler and can be easily defined in a similar way.

3 Implementation in SimEvents

SimEvents [9] is a blockset for the Simulink environment from Mathworks [10] that enables discrete event modeling. It uses the transaction-based approach, which describes entities that are handled by fixed components. It contains the usual components like an entity generator, a server, a queue and several routing blocks. As stated above, a basic Conveyor System component is available that transports discrete entities of given length. Many components include so-called “action”-functions, which are called at the entry or exit of an entity, and can be defined using Simulink function blocks.

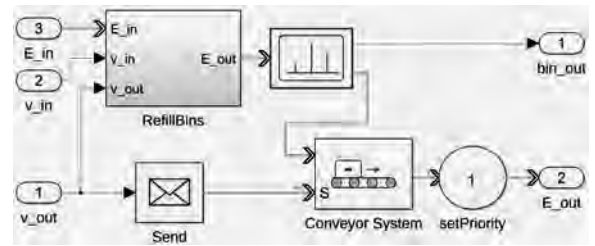


Figure 3: Conveyor model.

The conveyor for dry bulk material (cf. Figure 3) is defined as a component with an input and output port for the entities, inputs for the incoming and outgoing

velocities and an additional output to display the entities leaving the internal bin. The length l_E of the compartments and the total length l are provided as parameters. Incoming and outgoing entities have attributes describing their length and their mass. The block uses the predefined `Conveyor System` and a component `RefillBins` that handles the adaptation of the different velocities.

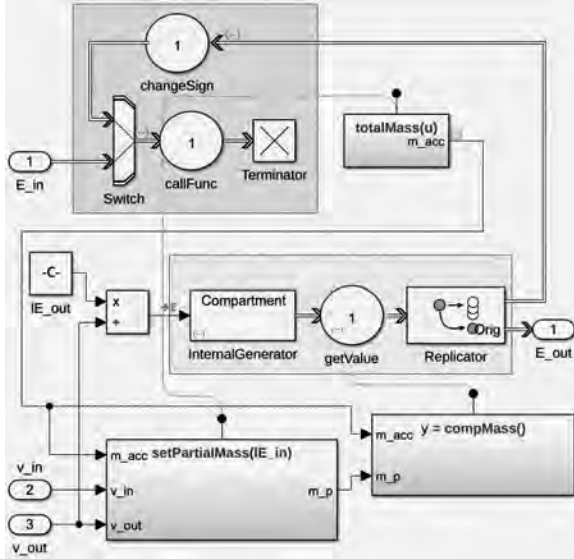


Figure 4: Implementation of the `RefillBins` component.

The central block `RefillBins` (cf. Figure 4) implements the formulae described in section 2 that define strategy B. Incoming entities are routed through a server, which calls the Simulink functions `totalMass` to compute m_{acc} and `setPartialMass` to compute m_p , and destroyed afterwards. An internal generator creates new entities at times in T_{out} and sends them to a server that sets the mass attribute to m_{out} , which is computed with the function `compMass`. Before an entity leaves the block, a copy is created and sent back to the input server, so that its mass can be subtracted from m_{acc} . The alternative strategy A can be implemented easily in an analogous way.

As usual for transaction-based modeling, one has to make sure that concurrent events are handled in the correct order to make things work. If $t \in T_{in} \cap T_{out}$, this means that the incoming entity has to be processed before the internally created one to guarantee the correct computation of m_p and m_{out} . For this purpose entities enter the conveyor with a high priority (low value), while the internal generator creates entities with low

priority, which is raised, when an entity leaves the conveyor.

A more subtle timing problem has lead to the inclusion of the server `getValue` behind the internal generator: In principle the call of the function `compMass` could have been done immediately inside the generator. But then the order of the mass computation and the processing of a concurrent incoming entity are not defined! The priority only affects events and messages, not internal function calls.

4 Test results

To compare the performance of the strategies A and B in detail, a set of tests have been carried out that concentrate on two key figures: the mean value over time $\overline{m_{acc}}$ of the internally accumulated mass, which shows the short-time mass conservation, and the standard deviation σ_{out} of the output mass, which measures the homogeneity of the outgoing mass distribution.

All tests use constant entity lengths $l_{E,1} = l_{E,2} = 1$ m and outgoing velocity $v_2 = 1$ m/s. The input velocity is given as $v_1 = k v_2$, where different values of k and different input mass distributions will be analysed. All results are compiled in Table 5 and are referenced by their number in the following.

The first group (1 – 9) consists of tests with constant input mass $m_i = 1$ kg and varying k . For k or $1/k$ integer, optimal procedures have been given in section 2, which lead to a constant output mass, i. e. $\sigma = 0$. For these cases the average value of m_{acc} can easily be computed to be

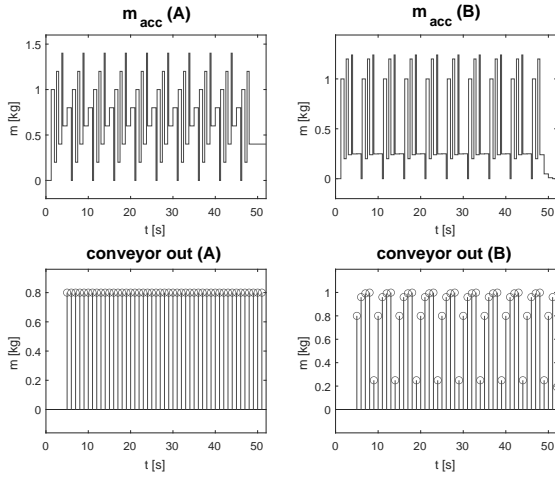
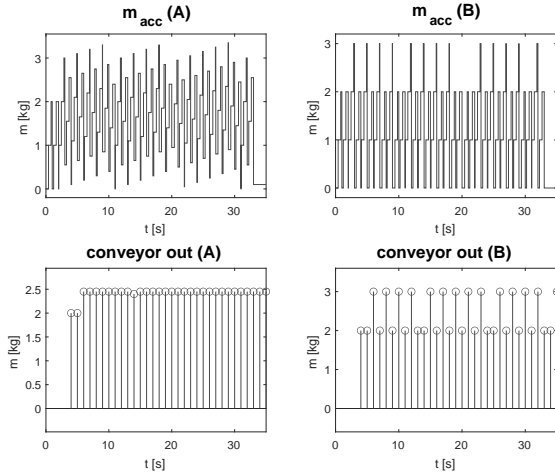
$$\overline{m_{acc}} = \frac{n-1}{2n} \quad | \quad n \equiv 1/k \in \mathbb{N}^+$$

$$\overline{m_{acc}} = \frac{n-1}{2} \quad | \quad n \equiv k \in \mathbb{N}^+$$

Test results 1 – 4 show that both implementations reproduce these values, minor differences are due to a short initial period.

Results for other values of k are shown in tests 5 – 9, among them the examples from section 2. The plots in Figure 5 display the function $m_{acc}(t)$ and the conveyor output over time for both strategies, they reproduce the results for $k = 0.8$ from Tables 3 and 4. The figures from Table 5, no. 6, show that the mass balance of strategy B is better by a factor of 1.5 than that of strategy A.

Very instructive is the comparison of tests 8 and 9: Changing k from 2.5 to 2.45 leads to a much worse lo-

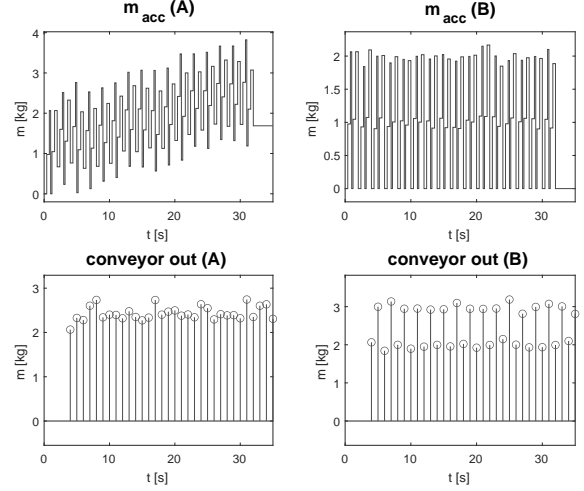

 Figure 5: Test result 6, constant input, $k = 0.8$.

 Figure 6: Test result 8, constant input, $k = 2.45$.

cal mass balance, especially for strategy A. The reason for this behaviour can be seen in Figure 6: The accumulated mass rises slowly over fast cycles of 2 s, but is reset with a longer period of 20 s. A look at Table 1 shows that for $k = 2.5$ a much shorter period of 5 s appears, so that m_{acc} can't grow as much. The length of the period is given by the representation $k = p/q$ with coprime natural numbers p, q :

$$k = \frac{p}{q} = \frac{\Delta t_2}{\Delta t_1} \Rightarrow q\Delta t_2 = p\Delta t_1,$$

where $\Delta t_2 = 1$ s in all our tests. A rational k with a large denominator therefore leads to a long period, which can

possibly produce a long time accumulation and a bad mass balance. The problem is less severe for strategy B, since it gets rid of short time accumulations as fast as possible.


 Figure 7: Test result 14, uniform input, $k = 2.5$.

In real applications the input flow is usually not constant, at least it fluctuates due to the granularity of the material. To model this, the next tests (10 – 14) take mass input values $m_i(t)$ using a uniform distribution on the interval $[0.9, 1.1]$ kg, which has a standard deviation of $\sigma_{in} = 0.058$ kg.

The corresponding results in Table 5 are generally similar to the previous ones, but the standard deviations seem to be too small, they are sometimes smaller than σ_{in} . This is due to two effects: Firstly, the mean value of the output mass is not 1, but k , and σ_{out} has to be scaled accordingly. Secondly, the internal accumulation process smoothes the incoming values, thereby reducing the standard deviation.

A striking result is the mass balance of strategy A in test 14 ($k = 2.5$), which is much larger than expected. Figure 7 shows that the accumulation of rest masses, which was limited before due to the periodic behaviour, now grows apparently unbounded.

For the last tests a macroscopic change has been added to the small scale fluctuations: In tests 15 – 19 the mean value of the input mass is reduced by a factor two in the middle of the measurement, in tests 20 – 24 it is doubled. If one takes the changing mass scale into account, the results are similar to the last ones. It is interesting to note, what happens to the previous problematic case $k = 2.5$: As can be seen in Figure 8 the

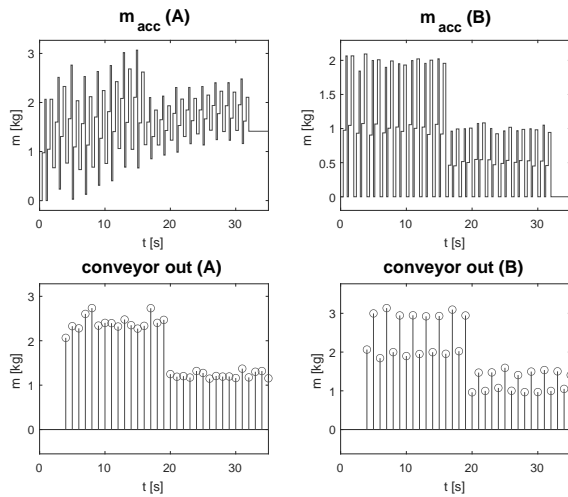


Figure 8: Test result 19, falling input, $k = 2.5$.

accumulation of “residual mass” continues, even if the mean value drops. The same happens, if the mean value rises (cf. Table 5, no. 24).

5 Conclusion

We have presented a simple discrete event-based model of a conveyor system for the transport of dry bulk material, which allows for the coupling of conveyors with different velocities. Two strategies have been compared to cope with the timing problems, where strategy B is much better, if the short time mass balance is of highest concern, while strategy A provides a better homogeneity of the outgoing masses.

While discretisation of continuous systems is important to reduce computation times drastically, it creates problems of its own. To solve them, a precise mathematical description is of uttermost importance, not only to precisely define the model, but also to guide and thereby simplify the implementation process. A typical implementation problem, which had to be solved here, was to ensure the correct ordering of concurrent events. While using priorities is a standard way to cope with it, one had to dig deeply into internal features of SimEvents to come up with a final solution. Since such details vary between different simulation environments [11], a precise (mathematical!) definition of the exact behaviour of SimEvents would have been helpful.

Though the model has shown its principle validity in a series of tests, the real proof of its usefulness would be

No.	k	$\overline{m_{acc}}(A)$ [kg]	$\overline{m_{acc}}(B)$ [kg]	$\sigma_{out}(A)$ [kg]	$\sigma_{out}(B)$ [kg]
1	0.33	0.3300	0.3300	0.0000	0.0000
2	0.14	0.4243	0.4243	0.0000	0.0000
3	3.00	1.0000	1.0000	0.0000	0.0000
4	7.00	3.0000	3.0000	0.0000	0.0000
5	0.35	0.5315	0.4460	0.0145	0.0394
6	0.80	0.6644	0.4555	0.0000	0.2930
7	1.20	0.8429	0.5000	0.0672	0.3966
8	2.45	1.5315	1.1300	0.1106	0.5040
9	2.50	1.1714	0.9143	0.0884	0.5080
10	0.35	0.5331	0.4440	0.0241	0.0386
11	0.80	0.6825	0.4499	0.0455	0.2933
12	1.20	0.8351	0.4933	0.0867	0.3948
13	2.45	1.3826	1.1141	0.1690	0.4864
14	2.50	1.8598	0.9123	0.1552	0.5136
15	0.35	0.4111	0.3237	0.0875	0.0929
16	0.80	0.6621	0.3391	0.2048	0.3075
17	1.20	0.6865	0.3629	0.2910	0.4185
18	2.45	1.1876	0.8426	0.5983	0.6979
19	2.50	1.5818	0.6838	0.6132	0.7500
20	0.35	0.8212	0.6846	0.1764	0.1809
21	0.80	0.9939	0.6715	0.4006	0.5983
22	1.20	1.3305	0.7239	0.6296	0.9080
23	2.45	2.0568	1.6572	1.2454	1.4854
24	2.50	2.4205	1.3695	1.2898	1.5003

Table 5: Test results comparing mass conservation and homogeneity of both strategies.

seen in the integration with a controller. The coupling with a continuous controller should work in principle, but for several reasons – practical as well as theoretical –, a discrete controller with a finite set of velocities would be more adequate in a lot of applications [12, 13]. Whether the simple strategies proposed here are useful in such a context, or whether one needs more complex strategies, which are adapted to the controller algorithm, is a question for future research.

Acknowledgement

The authors like to thank GRIMME Landmaschinenfabrik GmbH & Co. KG for the permission to publish results of internal research.

References

- [1] W David Kelton NBZ Randall P Sadowski. *Simulation with Arena*. New York: McGraw-Hill, 6th ed. 2015.

- [2] Hiltermann J, Lodewijks G, Schott DL, Rijsenbrij J, Dekkers J, Pang Y. A methodology to predict power savings of troughed belt conveyors by speed control. *Particulate science and technology*. 2011;29(1):14–27.
- [3] The MathWorks. *SimEvents: Model and simulate discrete-event systems*. www.mathworks.com/products/simevents.html.
- [4] He D, Pang Y, Lodewijks G. Belt conveyor dynamics in transient operation for speed control. *International Journal of Civil, Environmental, Structural, Construction and Architectural Engineering*. 2016; 10(7):828–833.
- [5] Bangsow S. *Tecnomatix Plant Simulation - Modeling and Programming by Means of Examples*. Cham: Springer, 2nd ed. 2020.
- [6] Göttlich S, Hoher S, Schindler P, Schleper V, Verl A. Modeling, simulation and validation of material flow on conveyor belts. *Applied mathematical modelling*. 2014; 38(13):3295–3313.
- [7] Fioroni MM, Franzese LAG, Zanin CE, Furia J, de Toledo Perfetti L, Leonardo D, da Silva NL. Simulation of continuous behavior using discrete tools: Ore conveyor transport. In: *2007 Winter Simulation Conference*. IEEE. 2007; pp. 1655–1662.
- [8] Pihnastyi O, Kozhevnikov G, Khodusov V. Conveyor model with input and output accumulating bunker. In: *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. IEEE. 2020; pp. 253–258.
- [9] Li W, Mani R, Mosterman PJ. Extensible discrete-event simulation framework in SimEvents. In: *Proc. 2016 Winter Simulation Conference*. Arlington: IEEE. 2016; pp. 943–954.
- [10] The MathWorks. *Simulink: Simulation and Model-Based Design*. <https://www.mathworks.com/products/simulink/>.
- [11] Schriber TJ, Brunner DT, Smith JS. How discrete-event simulation software works and why it matters. In: *Proceedings of the 2012 Winter Simulation Conference (WSC)*. IEEE. 2012; pp. 1–15.
- [12] Pang Y, Lodewijks G, Schott DL. Fuzzy Controlled Energy Saving Solution for Large-Scale Belt Conveying Systems. In: *Energy, Environment and Sustainable Development*, vol. 260 of *Applied Mechanics and Materials*. Trans Tech Publications Ltd. 2013; pp. 59–64.
- [13] Reutov AA. Simulation of load traffic and steeped speed control of conveyor. In: *IOP Conference Series: Earth and Environmental Science*, vol. 87. IOP Publishing. 2017; p. 082041.

Simulationsbasiertes Deep Reinforcement Learning für Modulare Produktionssysteme

Niclas Feldkamp^{1*}, Sören Bergmann¹, Steffen Straßburger¹

¹Informationstechnik in Produktion und Logistik, TU Ilmenau, Max-Planck-Ring 12, 98693 Ilmenau; *niclas.feldkamp@tu-ilmenau.de

Abstract. Modulare Produktionssysteme sollen die traditionelle Fließbandproduktion in der Automobilindustrie ablösen. Die Idee dabei ist, dass sich hochgradig individualisierte Produkte dynamisch und autonom durch ein System flexibler Arbeitsstationen bewegen können. Dieser Ansatz stellt hohe Anforderungen an die Planung und Organisation solcher Systeme. Da jedes Produkt seinen Weg durch das System frei und individuell bestimmen kann, kann die Implementierung von Regeln und Heuristiken, die die Flexibilität des Systems zur Leistungssteigerung ausnutzen, in diesem dynamischen Umfeld schwierig sein. Transportaufgaben werden in der Regel von fahrerlosen Transportsystemen (FTS) ausgeführt. Daher bietet die Integration von KI-basierten Steuerungslogiken eine vielversprechende Alternative zu manuell implementierten Entscheidungsregeln für den Betrieb der FTS. In diesem Beitrag wird ein Ansatz für den Einsatz von Reinforcement Learning (RL) in Kombination mit Simulation vorgestellt, um FTS in modularen Produktionssystemen zu steuern. Darüber hinaus werden Untersuchungen zu dessen Flexibilität und Skalierbarkeit durchgeführt.

Einleitung

Insbesondere in der Automobilindustrie stellen modulare Produktionssysteme den nächsten Evolutionsschritt in der Produktionsorganisation dar, welcher die klassische Linienfertigung vielerorts abzulösen versucht. Heutzutage wird die Bedeutung der Flexibilität immer wichtiger, da die Anforderungen an Qualität, Vielfalt der Produktpalette und Individualisierbarkeit steigen [6]. Viele Automobilhersteller haben daher Pilotprojekte gestartet, um die Modernisierung des traditionellen Fließbandkonzepts hin zu einem flexibleren, werkstattähnlichen Produktionsschema zu voranzutreiben. Dies lässt sich zusammenfassen unter dem Begriff des modularen Produktionssystems [15–17]. In der Praxis finden sich viele weitere Begriffe, die alle mehr oder weniger den gleichen Ansatz beschreiben, darunter "modulare Montage" [3], "Flexi-Line" [22], "Full-Flex-Plant" [5] oder "Factory 56" [4]. Das Ziel dieser Ansätze ist es, nahezu jeden Fahrzeugtyp bzw. Variante im selben Werk zu produzieren, so dass auch bei der Einführung neuer Produkte kein langwieriger Um- oder Neubau der Produktion erforderlich ist.

Im besten Fall können neue Produkte sogar im laufenden Betrieb eingeführt werden, ohne dass die Produktion unterbrochen werden muss. Erreicht werden soll dies durch die Umsetzung einer selbstregulierenden Produktion im Sinne des Grundgedankens von Industrie 4.0. Hierbei werden die Produktionsaufträge von fahrerlosen Transportsystemen (FTS) durch ein System von matrixförmig angeordneten Stationen geleitet. Diese Stationen sind unterschiedlich ausgestattet und können daher jeweils verschiedene Produktionsschritte anbieten. Produkte können dynamisch und individuell entscheiden, welchen Produktionsschritt sie als nächstes ausführen lassen wollen und welche Station diesen ausführen soll. Die gesamte Steuerungsstrategie wird nur durch technische Restriktionen des Produktionsprogramms hinsichtlich der Reihenfolge einiger Produktionsschritte limitiert. Zur Abbildung der Entscheidungsfindung der FTS ist die einfachste Methode die Implementierung von vordefinierten Heuristiken. Die Einbeziehung der sich dynamisch verändernden Umgebung und des Systemzustands in solche heuristischen Regeln kann in einem komplexen System recht schwierig sein und führt nicht in jedem möglichen Szenario zur besten Lösung. Daher ist die Verwendung einer auf künstlicher Intelligenz (KI) basierenden Steuerungslogik, die an einer großen Anzahl von Beispielen trainiert wird und in Echtzeit entscheiden kann, ein vielversprechender Ansatz. Auf der anderen Seite ist die ereignisdiskrete Simulation eine etablierte Methodik für die Modellierung, Planung und Steuerung von Produktionssystemen, insbesondere in der Automobilindustrie. Daher liegt es nahe, eine KI-basierte Steuerungslogik mit Hilfe eines geeigneten Simulationsmodells zu trainieren. In einem vorherigen Beitrag wurde hierzu ein Ansatz vorgestellt, welcher Reinforcement Learning in Verbindung mit einem Simulationsmodell zum Erlernen einer Steuerungslogik für die FTS nutzt [9]. Dieser Ansatz wird in diesem Beitrag weiter untersucht. Im Zentrum steht hierbei die Frage, wie flexibel und skalierbar die gelernte Strategie bei Änderungen im System

ist bzw. ob bei Änderungen des Ausgangssystems die Strategie neu angelernt werden muss.

1 Grundlagen und Stand der Forschung

1.1 Modulare Produktionssysteme

Der Begriff des modularen Produktionssystems ist ein Synonym für eine Reihe von Terminologien, die alle das gleiche Prinzip beschreiben. Bei diesem Ansatz ist die Flexibilisierung der Montage, z. B. in der Automobilindustrie, das Hauptziel. Das traditionelle Prinzip des festen Fließbandes wird zugunsten einer eher werkstatt-ähnlichen Produktion aufgelöst [15, 17]. Die eigentliche Montage der Produkte erfolgt an Montagestationen (oft auch als Fertigungsinseln bezeichnet), die sehr flexibel sind und verschiedene Aufgaben ausführen können, abhängig von ihrer Funktion, den Werkzeugen und dem zur Verfügung stehenden Personal. Jedes Produkt ist durch Aufgaben definiert, so dass die Produkte einen sehr hohen Grad an Individualisierung aufweisen können. Einzelne Produkte müssen also unterschiedliche Stationen durchlaufen und können daher individuelle Wege durch das System gehen. Die Montagestationen sind nicht durch ein festes Fördersystem verbunden. Das heißt, sowohl der Transport der Produkte als auch der Materialtransport wird durch FTS durchgeführt [21]. Hieraus ergibt sich ein hoher Grad an Flexibilität auf unterschiedlichen Ebenen. Die auszuführenden Montageschritte können sehr unterschiedlich sein und sind nicht an eine global einheitliche Systemzykluszeit gebunden, wie in Abbildung 1 dargestellt.

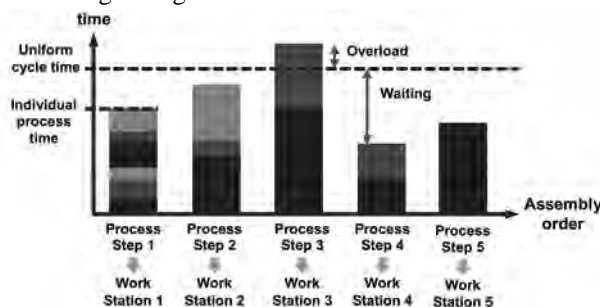


Abbildung 1: Individuelle Prozesszeiten gegenüber einheitlicher Zykluszeit [14, S.161].

Die verfügbaren Stationen bieten unterschiedliche Aufgaben an, die sie erfüllen können. Zudem können sich die Produkte infolge des Wegfalls eines festen Fördersys-

tems frei bewegen, so dass die Montageschritte nicht unbedingt in einer festen Reihenfolge ausgeführt werden müssen, sofern keine technischen Einschränkungen gelten. Daraus ergibt sich ein hoher Freiheitsgrad für die Produkte, was die Flexibilität des Systems erhöht und die Kapazitätsauslastung der Stationen verbessert [14, 17]. Auf der anderen Seite wird dieser hohe Freiheitsgrad mit einer erhöhten Komplexität und Herausforderungen bei der Koordination, Kommunikation und Organisation zwischen den Systemkomponenten erkauft. Dies bezieht sich auf einen der Grundgedanken des Industrie 4.0-Konzepts, dass eine Vernetzung zwischen Maschinen, Geräten, Sensoren und Menschen proklamiert [19]. Einige Veröffentlichungen, welche sich bereits mit den simulationsbezogenen Aspekten und Modellierungstechniken modularer Produktionssysteme befasst haben [8, 26], kamen zu dem Schluss, dass ein geeignetes Simulationsmodell für die Entwicklung von Heuristiken zur Optimierung von Planungs- und Steuerungsproblemen in solchen Systemen unerlässlich ist und dies auch nach wie vor eine Herausforderung in Forschung und Praxis darstellt [10, 24]. Die Verwendung von KI-gestützten Steuerungslogiken trägt daher zur Lösung dieser Herausforderungen bei.

1.2 Deep Reinforcement Learning

Die Grundidee von Reinforcement Learning ist, dass ein Agent mit seiner Umgebung interagieren kann und auf der Grundlage des von ihm wahrgenommenen Zustands der Umgebung eine Aktion aus einer gegebenen Menge bestimmter Aktionen auswählt. Als Folge seiner Aktion erhält der Agent im Gegenzug eine Rückmeldung von seiner Umgebung. Diese Rückmeldung wird als Belohnung gewährt. Die Belohnung wird durch eine vorher ausgestaltete Formel berechnet und kann positiv oder negativ sein (Bestrafung statt Belohnung), abhängig vom gegebenen Zustand der Umwelt und der Aktion, die der Agent gewählt hat [28]. Diese Beziehung kann als ein sogenannter Markov-Entscheidungsprozess formalisiert werden, wie in Abbildung 2 dargestellt. Basierend auf der Wahrnehmung des Umgebungszustands S_t aus allen möglichen Zuständen S zu diskreten Zeitpunkten $t = 1..n$, wählt der Agent eine Aktion A_t aus der Menge der möglichen Aktionen A . Dies führt zur Änderung des Zustands zu S_{t+1} . Der Agent erhält dann den neuen Zustand sowie die Belohnung R_{t+1} .

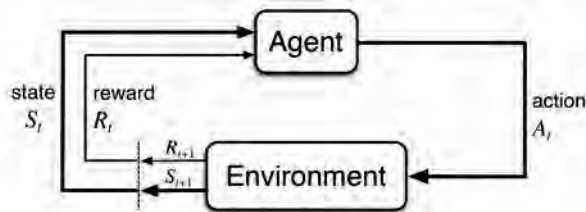


Abbildung 2: Agent-Umgebungsbeziehung in einem Markov-Entscheidungsprozess [28, S.48].

Der Agent strebt an, seine Handlungen so zu wählen, dass er die Summe seiner Belohnungen über den betrachteten Zeitraum maximieren kann und entwickelt so letztlich eine Strategie zur Lösung des Problems [28, 31]. Tatsächlich ist Reinforcement Learning aber ein Oberbegriff für eine Vielzahl unterschiedlicher Methoden und Algorithmen. Die meisten dieser Algorithmen lassen sich in zwei Kategorien einteilen: Modellbasierte und modellfreie Ansätze. Modellbasierte Ansätze zielen darauf ab, ein Modell der zugrundeliegenden Regeln und der Dynamik der jeweiligen Umgebung zu erlernen, um daraus eine optimale Strategie abzuleiten. Modellfreie Ansätze hingegen zielen darauf ab, die beste Aktion für jeden Zustand zu finden, um eine globale Strategie über alle Zustände zu erstellen oder einen gewünschten Zustand so schnell wie möglich zu erreichen [1, 18].

Für komplexe und dynamische Systeme, die mit stochastischer Unsicherheit konfrontiert sein können (wie modulare Produktionssysteme), sind modellbasierte Ansätze in der Regel nicht sehr geeignet, da der Zustandsraum zu groß ist, die interne Modellbildung des Algorithmus zu ungenau sein könnte und der Rechenaufwand zu hoch ist [12]. Einer der bekanntesten Vertreter der modellfreien Ansätze ist Q-Learning. Hier lernt der Agent für jedes Zustands-Aktions-Paar einen Wert (sog. Q-Wert), so dass er eine Wertfunktion $Q(s,a)$ approximieren kann, aus der er eine optimale Strategie ableiten kann [30]. Eine adäquate Erweiterung des traditionellen Q-Learnings ist Deep-Q-Learning (DQL), bei dem die Beziehung zwischen Zuständen und Aktionen durch ein künstliches neuronales Netz abgebildet wird. Dies ermöglicht das Prinzip des Q-Learning auch für große Zustands- und Aktionsräume [23, 27].

1.3 DQL in Verbindung mit ereignisdiskreten Simulationsmodellen

Gabel und Riedmiller konnten zeigen, dass DQL in der Lage ist, Standard-Optimierungsprobleme im Kontext des Produktionsmanagements zu lösen. In einem Job-Shop-Scheduling-Szenario mit parallelen Maschinen

konnten sie klassische heuristische Regeln schlagen [11]. Über die Kombination von DQL und ereignisdiskreten Simulationsmodellen, der bevorzugten Simulationstechnik für die Modellierung von Produktionssystemen [8], gibt es nur wenig Forschung, so dass hier insgesamt noch viel Forschungsbedarf besteht. Gosavi weist in diesem Zusammenhang auf die komplexe Herausforderung der Synchronisation von Simulationsfortschritt und Lernalgorithmus hin [12]. Zur Lösung einfacher Optimierungsprobleme wurden bereits erfolgreich DQL-Algorithmen in Verbindung mit Simulationsmodell angewendet [12, 13]. Dies bezog sich allerdings auf das Finden einer globalen optimalen Lösung für ein festgelegtes Problem.

DQL bietet in Kombination mit Simulation aber auch vielversprechende Möglichkeiten für die dynamische Entscheidungsfindung in Echtzeit in einer sich dynamisch verändernden und unsicheren Umgebung. Zhang et al. setzten DQL und Simulation erfolgreich für die Bestimmung des nächsten auf einer Maschine zu verarbeitenden Auftrags [32, 33] sowie für Losgrößenbestimmung in Echtzeit ein [34]. In diesem Zusammenhang weisen Xie et al. darauf hin, dass es viele Herausforderungen bei der Kombination von Simulation und DQL gibt, die noch nicht abschließend und allgemeingültig gelöst sind, zum Beispiel die Quantifizierung von Zuständen und Aktionen, die Gestaltung der Belohnungsfunktion und die Sicherstellung der Konvergenz des Lernalgorithmus [32].

2 Reinforcement Learning und Simulation zur Steuerung von FTS in modularen Produktionssystemen

2.1 Konzeptionelle Überlegungen

Im Folgenden wird unterstellt, dass ein FTS ein Produkt/Auftrag an der Quelle aufnimmt und wieder freigibt, nachdem alle erforderlichen Montageaufgaben ausgeführt wurden, bevor es wieder ein neues Produkt aufnimmt. Das Produkt und das FTS bilden insofern eine Einheit. D. h., das Produkt überlässt die gesamte Entscheidungsfindung im Prozess dem FTS und seiner Steuerungslogik.

Der Entscheidungsfindungsprozess für FTS kann wiederum in mehrere Ebenen unterteilt werden, wie in Abbildung 3 dargestellt.



Abbildung 3: Hierarchie der Entscheidungsfindung für FTS in modularen Produktionssystemen [7, S.397].

Die erste Entscheidungsebene ist für die Auswahl der nächsten Aufgabe (welcher Montageschritt als nächstes auszuführen ist) und des Ziels (welche Station für die Ausführung der Aufgabe zu wählen ist) verantwortlich. Diese Entscheidung führt kaskadenartig zu untergeordneten Entscheidungen in Bezug auf Routing und Wegfindung, Kollisionserkennung und -vermeidung bis hin zu den eigentlichen Sensoren und Aktoren des FTS. Der im Folgenden vorgestellte Ansatz zielt auf die erste Ebene der Entscheidungsfindung ab und überlässt die unteren Ebenen anderen Systemen bzw. dem Simulator. Dies schließt jedoch nicht aus, dass diese Ebenen auch durch KI-gestützte Systeme gesteuert werden können. Das FTS muss folglich nach jeder abgeschlossenen Aufgabe oder nach Aufnahme eines neuen Auftrags an der Quelle eine Entscheidung treffen. Wenn eine Entscheidung ansteht, wird die Simulation angehalten und der aktuelle Systemzustand an den RL-Agenten gesendet. Der Agent trifft dann die Entscheidung über die nächste anzufahrende Station und sendet sie zurück an die Simulation, woraus dann das nächste Ziel für das FTS festgelegt und die Simulation fortgesetzt wird, bis eine neue Entscheidung erforderlich ist. Hierbei wird zur Entscheidungsfindung, wie oben erläutert, DQL für das Lernen des RL-Agenten verwendet. Das bedeutet, dass ein mehrschichtiges neuronales Netz verwendet wird, um die Wertfunktion zwischen Systemzuständen und Aktionen zu modellieren. Der Input des Netzes wird durch den aktuellen Systemzustand repräsentiert. Theoretisch könnte eine große Anzahl von möglichen Systemparametern übergeben werden. In unserem Ansatz wurde der Input reduziert auf die Aufgaben, die derzeit ausgeführt werden können, die verfügbaren Stationen (d. h., jene Station, die ausgestattet sind, um eine der verfügbaren Aufgaben auszuführen) und die aktuelle Länge der einzelnen Warteschlangen. Die Ausgabeschicht des Netzes stellt dann die Stationen dar, so dass das Ausgangsneuron mit dem höchsten Wert die nächste Station für die aktuelle Entscheidung bestimmt.

Um die Leistung des Lernalgorithmus weiter zu steigern, wird Double Deep Q-Learning (DDQN) eingesetzt.

Dabei werden zwei identische neuronale Netze verwendet: Ein lokales Netz, das die nächste Aktion vorhersagt, und ein Zielnetz, das den Q-Wert für den Trainingsschritt berechnet. Diese Methode verhindert, dass der Agent suboptimale Strategien überschätzt und führt zu einer besseren Gesamtlernleistung [29].

Der Agent lernt im Laufe mehrerer Episoden. Eine Episode endet, wenn der Agent eine ungültige Entscheidung trifft oder das Zeitlimit der Simulation erreicht ist. Der eigentliche Trainingsprozess ist angelehnt an das bekannte Lernproblem, bei dem ein Agent versucht, den Ausgang eines Labyrinths zu finden, ohne in eine Falltür zu geraten [2]. Für jede richtige Entscheidung, die der Agent trifft, erhält er eine Belohnung von 0. Wenn ein Produkt vollständig zusammengebaut und freigegeben wird, erhält er eine Belohnung von 1 (der Ausgang des Labyrinths gilt als gefunden), was den Agenten dazu bringt, so schnell wie möglich zum Ausgang zu gelangen. Wenn eine ungültige Aktion gewählt wird (was bedeutet, dass die gewählte Station keine der aktuell verfügbaren Aufgaben ausführen kann), wird eine Belohnung von -1 gegeben, die Episode endet und die Simulation beginnt von vorne. Auf diese Weise wird sichergestellt, dass die korrekten Produktionsregeln und -beschränkungen erlernt werden. Darüber hinaus wurden der Belohnungsfunktion zusätzliche leistungsbezogene Komponenten hinzugefügt, nachdem sich ein stabiler Lernprozess etabliert hatte. Eine Aktualisierung der Gewichte der neuronalen Netze und damit der Q-Funktion direkt nach jeder Entscheidung ist jedoch nicht ratsam. Zeitliche Korrelationen zwischen dem Lernen aus Erfahrungen und dem Sammeln von Erfahrungen können zu unerwünschten Verzerrungen führen, die den Agenten dazu neigen lassen, ältere Vorkommnisse bestimmter Zustands-/Aktionspaare zu vergessen. Aus diesem Grund wurde ein Mechanismus implementiert (sog. Memory-Replay-Buffer), der jede Erfahrung, die der Agent macht, in einem großen Puffer speichert. Gespeichert wird hierbei der jeweilige Systemstatus, die Aktionen des Agenten, den neuen Systemstatus nach der Aktion und die gewonnene Belohnung. Nach jeder Episode wird eine Zufallsstichprobe von Erinnerungen aus diesem Puffer entnommen, um das neuronale Netz zu trainieren [20]. Nach erfolgreichem Training kann dann das Echtssysteme an die Stelle des Simulationsmodells treten, sodass der RL-Agent die realen FTS steuern kann.

2.2 Implementierung

Für die Erprobung der Implementierung anhand einer

Fallstudie wurde ein Beispielmodell entwickelt, welches ein typisches Layout für modulare Produktionssysteme in der Automobilindustrie darstellt. Abbildung 4 zeigt einen Screenshot des Modells, welches mit Siemens Plant Simulation implementiert wurde. Das Modell enthält eine 50 m x 35 m große Fabrikhalle mit 9 Montagestationen. Die FTS können sich vor den Stationen in eine Warteschlange einreihen, wenn sie auf die Bearbeitung warten. Es gibt 4 verschiedene Produkttypen, die jeweils bis zu 20 einzelne Montageaufgaben mit unterschiedlichen Bearbeitungszeiten benötigen. Jede der Stationen bietet ein Portfolio von 2 bis 4 Aufgaben, welche sie ausführen kann. Für jedes Produkt gibt es spezifische Vorranggraphen für die Montage, so dass manche Aufgaben in einer bestimmten Reihenfolge ausgeführt werden müssen, während die Reihenfolge für andere Aufgaben frei gewählt werden kann. Die Art des Auftrags wird zufällig ausgewählt, bevor er einem freien FTS zugewiesen wird und von diesem an der Quelle abgeholt wird.

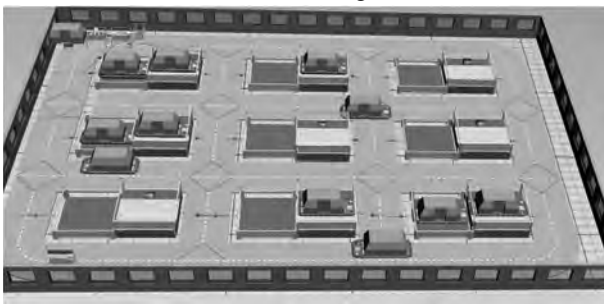


Abbildung 4: Screenshot des Plant Simulation Modells.

Die Implementierung des DQL-Algorithmus wurde in der Python-Bibliothek Tensorflow/Keras vorgenommen. Die Verbindung zwischen Plant Simulation und Python wurde über eine TCP/IP-basierte Schnittstelle realisiert. Dabei fungiert die Python-Seite als Server, das Plant-Simulation-Model als Client. Beim Starten des Servers werden die erforderlichen Elemente, einschließlich des neuronalen Netzes, initialisiert. Der Server wartet dann auf eingehende Anfragen des Simulators. Bei jeder Anfrage werden eine Reihe von Daten übertragen, darunter der Zustand des Systems sowie einige Metadaten wie Simulationszeit und Systemkennzahlen. Dieser Zustandsvektor wird an den RL-Algorithmus weitergeleitet, welcher dann eine Aktion auswählt, die dann wiederum als Antwort auf die TCP/IP-Anfrage an den Simulator zurückgesendet wird. Die Serverseite ist auch für die Berechnung von Belohnungen sowie für das Anhalten, Zurücksetzen und Starten der Simulation über in die

TCP/IP-Antwort eingebettete Schlüsselwörter verantwortlich. Für die Feinabstimmung des Versuchsaufbaus wurden einige Vorversuche durchgeführt. Eine der größten Herausforderungen ist beispielsweise die Verwaltung des Memory-Buffers in Verbindung mit der Abfolge der Lernepisoden. Beim Lernen befinden sich RL-Agenten in einem Dilemma zwischen Exploration und Exploitation. Exploration bedeutet, dass der Agent neue Aktionen ausprobiert, während Exploitation bedeutet, dass der Agent an bereits gelernten Strategien festhält. Um Exploration zu gewährleisten und zu verhindern, dass der Agent in suboptimalen Strategien stecken bleibt, sollten Entscheidungen mit einer bestimmten Wahrscheinlichkeit ϵ zufällig gewählt werden. Diese Wahrscheinlichkeit sollte anfangs hoch sein (in der Regel 90 %) und dann über die Dauer der Episoden abnehmen.

2.3 Ergebnisse

Um die Performanz der erlernten Strategie des Agenten bewerten und in Kontext setzen zu können, wurde diese zum einen mit zufälligem Entscheiden der nächsten Aktion sowie einer einfachen, heuristischen Regel verglichen. Bei ersterem wählt das FTS aus der Menge der möglichen Ziele ein zufälliges Ziel aus. Für die Heuristik wird die nächste gültige Station mit der kürzesten Warteschlange und der geografisch nächstgelegenen Station ausgewählt, wenn mehrere Stationen zum Zeitpunkt der Entscheidungsfindung die gleiche Warteschlangenlänge haben. Für den Vergleich wurden die benötigten Durchlaufzeiten gemessen, jeweils mit einem Stichprobenumfang von 100. Das Ergebnis dieses Vergleichs ist in Abbildung 5 in Form eines Box-Whisker-Plots dargestellt. Hierbei sind die Ergebnisse mit 12 FTS sowie mit 14 FTS dargestellt. Es hat sich gezeigt, dass bei weniger als 12 FTS die Auslastung der Stationen zu gering ist, als dass irgendeine gesonderte Strategie einen nennenswerten Effekt auf die Durchlaufzeit hätte. Auf der anderen Seite führen mehr als 14 FTS zu Stau- und Blockadesituationen, sodass das System mit mehr als 14 FTS nicht sinnvoll zu betreiben ist. Wie in Abbildung 5 zu erkennen ist, ist die zufällige Entscheidungsauswahl in beiden Szenarien am schlechtesten. Im 12-FTS-Szenario ist die RL-basierte Strategie im Durchschnitt gleichwertig mit der heuristischen Regel. Allerdings ist sie in einigen Situationen sogar in der Lage gewesen, deutlich kürzere Durchlaufzeiten zu produzieren. Zudem sind die Ausreißer nach oben etwas weniger stark vertreten als bei der heuristischen Regel. Dies lässt vermuten, dass die heuristische

Regel im Schnitt schon eine sehr gute Grundstrategie darstellt, es aber Situation gibt, in denen dies eben nicht die beste Strategie darstellt und dieses Potenzial entsprechend vom RL-Agenten ausgenutzt wird. Im 14-FTS-Szenario nähern sich alle drei Strategien in der Spannweite aller Beobachtungen an, da hier offenbar generell weniger Optimierungspotenzial, welches durch eine geschickte Strategie ausgenutzt werden kann, besteht.

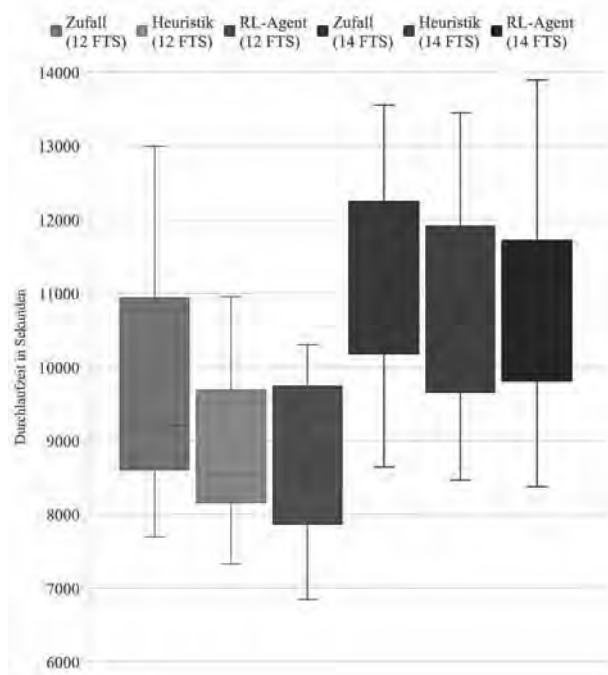


Abbildung 5: Vergleich von Durchlaufzeiten (in Sekunden) für unterschiedliche Steuerungsmechanismen und Szenarien.

Da sich der durchschnittliche Durchsatz zwischen den Szenarien nur unwesentlich unterscheidet (Abbildung 6), scheint der Einsatz von 14 FTS im gegebenen System nicht sinnvoll zu sein. Es existiert wenig Optimierungspotential, welches über das der heuristischen Regel hinausgeht. Interessant ist dennoch, dass im 14-FTS-Szenario Mittelwert und Median der gemessenen Durchlaufzeiten der RL-Agenten-Strategie deutlich unter denen der Heuristik liegen. Durch die zum Teil deutlich höheren Ausreißer liegt der durchschnittliche Durchsatz aber nur unwesentlich höher. Die RL-Agenten-Strategie scheint sich also bezogen auf die Entscheidungsfindung von der Heuristik zu unterscheiden, kann in Summe jedoch keine bessere Performanz erreichen.

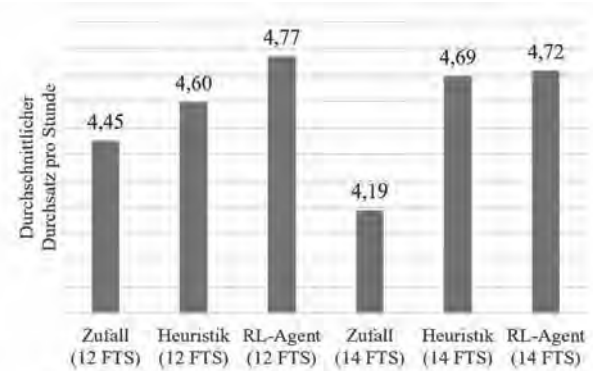


Abbildung 6: Durchschnittlicher Durchsatz von gefertigten Produkten pro Stunde.

Im nächsten Schritt wurde untersucht, wie flexibel der Agent mit seiner erlernten Strategie auf veränderte Rahmenbedingungen, d. h. eine veränderte Anzahl von FTS im System, reagiert. Dies ist in Abbildung 7 dargestellt.

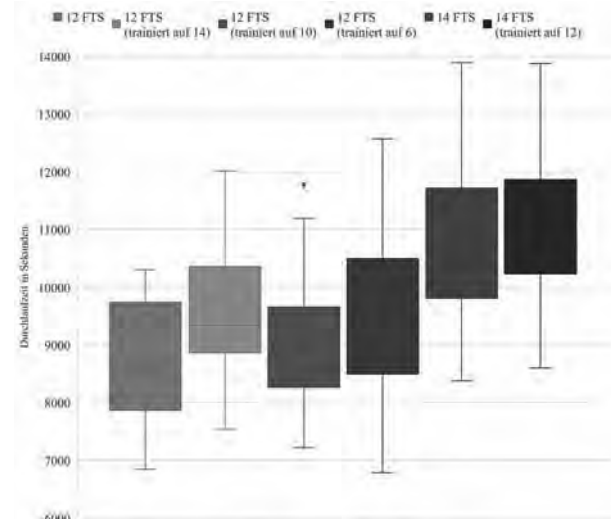


Abbildung 7: Vergleich der Durchlaufzeiten (in Sekunden) verschiedener RL-Agenten bei geänderte Rahmenbedingungen.

Hierbei wurde ein jeweils auf einer bestimmten Anzahl von FTS trainierter Agent in ein neues Umfeld mit veränderter FTS-Zahl eingesetzt, sowohl für das Szenario mit 12 FTS als auch für das Szenario mit 14 FTS. Hierbei wurde deutlich, dass der Agent in keiner der Varianten in der Lage war, mit der Performanz der für das eigentliche Szenario antrainierten Strategie mitzuhalten. Dies zeigt also, dass die jeweils antrainierten Strategien im Hintergrund keine allgemein gültige, optimierte Strategie gefunden haben, vielmehr ist diese vermutlich sehr stark auf das jeweilige Szenario spezialisiert. In der Konsequenz heißt dies, dass hier generell bei Änderungen am Szenario jedes Mal neu antrainiert werden sollte, um eine

bestmögliche Performanz zu erzielen.

3 Fazit und Ausblick

In diesem Beitrag wird gezeigt, dass die Entscheidungsfindung von FTS in modularen Produktionssystemen mittels Deep-Reinforcement-Learning in Kombination mit ereignisdiskreter Simulation verbessert werden kann. Die erlernte Strategie ist in der Lage, einfache heuristische Entscheidungsregeln zu übertreffen. Grundsätzlich ist die erlernte Strategie jedoch spezialisiert auf das zugrundeliegende System und reagiert empfindlich auf veränderte Rahmenbedingungen. Ein trainierter Agent sollte daher immer nur in demselben Umfeld eingesetzt werden, für welches er trainiert wurde. Grundsätzlich zeigt dieser Beitrag das Potenzial für KI-basierte Steuerung in modularen Produktionssystemen auf, es bietet sich allerdings auch noch viel Raum für weitere Forschung. Beispielsweise hängt der Lernerfolg des RL-Agenten stark von der Ausgestaltung der Belohnungsfunktion ab, welche wiederum vom angestrebten Optimierungsbedarf abhängt. In der hier gezeigten Fallstudie wurde sich auf das Ziel der Verbesserung der Durchlaufzeiten konzentriert. Mehrdimensionale Zielsysteme, wie Durchsatz, Durchlaufzeiten und Systemauslastung sind deutlich schwieriger darzustellen nicht nur bezogen auf die Komplexität der Belohnungsfunktion, sondern auch auf die Menge an Informationen, die der RL-Algorithmus verarbeiten muss. Ein weiteres Problem ist, dass der RL-Agent immer nur auf Grundlage seines individuellen Optimums trainiert und dabei eine mögliche globale Strategie, die zu besseren Ergebnissen führen könnte, vernachlässigt wird. Für weitere Forschung hierzu gibt es grundsätzlich zwei Möglichkeiten, dieses Problem anzugehen: Die erste Option wäre, einen Agenten zu trainieren, der nicht nur ein individuelles FTS, sondern das gesamte System auf einmal steuert. Dies würde theoretisch zu einer globalen Strategie führen, aber auch die Komplexität des Eingangsvektors des neuronalen Netzes stark erhöhen. Zudem wäre hier die Synchronisation der Entscheidungsanforderung und -findung zwischen Simulation und RL-Agent deutlich schwieriger darzustellen. Die andere Option wäre die Implementierung von Elementen klassischer Multiagentensysteme, so dass ein Agent mit anderen Agenten interagieren und verhandeln kann. Diese Ansätze werden aktuell als Multi-Agent Deep Reinforcement Learning (MADRL) zusammengefasst und wurden bereits in Anwendungen wie Flotten- und Verkehrsma-

nagement oder auch Energieverteilungsoptimierung eingesetzt [25]. Eine Nutzung solcher Algorithmen für die FTS-Entscheidungsfindung in modularen Produktionssystemen verspricht daher einen vielversprechenden Anknüpfungspunkt.

References

- [1] Achiam, J. and Morales, M. 2018. *OpenAI Spinning Up – Part 2: Kinds of RL Algorithms*. <https://spinningup.openai.com/en/latest/etc/author.html>. Accessed 14 April 2020.
- [2] Atienza, R. 2020. *Advanced Deep Learning with TensorFlow 2 and Keras*. Packt Publishing, Birmingham, UK.
- [3] Audi AG. 2019. *Die Modulare Montage - Fertigungsinseln statt Fließband*. <https://www.audi-il-lustrated.com/de/smart-factory/Die-Modulare-Montage>. Accessed 28 January 2019.
- [4] Daimler AG. 2018. *Factory 56 – Mercedes-Benz Cars increases flexibility and efficiency in operations*. <https://www.daimler.com/innovation/production/factory-56.html>.
- [5] Daimler AG. 2018. *Groundbreaking for first Full-Flex Plant*. <https://www.daimler.com/innovation/case/connectivity/full-flex-plant.html>. Accessed 28 January 2019.
- [6] ElMaraghy, H., Schuh, G., ElMaraghy, W., Piller, F., Schönsleben, P., Tseng, M., and Bernard, A. 2013. Product variety management. *CIRP Annals* 62, 2, 629–652.
- [7] Feldkamp, N., Bergmann, S., and Straßburger, S. 2019. Modellierung und Simulation von modularen Produktionssystemen. In *Simulation in Produktion und Logistik 2019*, M. Putz and A. Schlegel, Eds. Verlag Wissenschaftliche Scripten, Auerbach, 391–401.
- [8] Feldkamp, N., Bergmann, S., and Strassburger, S. 2019. Modelling and Simulation of Modular Production Systems. In *Simulation in Produktion und Logistik 2019*, M. Putz and A. Schlegel, Eds. Verlag Wissenschaftliche Scripten, Auerbach, 391–401.
- [9] Feldkamp, N., Bergmann, S., and Strassburger, S. 2020. Simulation-Based Deep Reinforcement Learning for Modular Production Systems. In *Proceedings of the 2020 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 1596–1607.
- [10] Filz, M.-A., Herrmann, C., and Thiede, S. 2019. Simulation-Based Data Analysis to Support the Planning of Flexible Manufacturing Systems. In *Simulation in Produktion und Logistik 2019*, M. Putz and A. Schlegel, Eds. Verlag Wissenschaftliche Scripten, Auerbach, 413–422.
- [11] Gabel, T. and Riedmiller, M. 2008. Adaptive Reactive Job-Shop Scheduling with Learning Agents. *International Journal of Information Technology and Intelligent Computing* 24, 4.
- [12] Gosavi, A. 2015. *Simulation-Based Optimization. Parametric Optimization Techniques and Reinforcement*

- Learning*. Operations Research/Computer Science Interfaces Series 55. Springer, New York.
- [13] Gosavi, A. 2015. Solving Markov Decision Processes via Simulation. In *Handbook of Simulation Optimization*, M. C. Fu, Ed. Springer New York, New York, NY, 341–379. DOI=10.1007/978-1-4939-1384-8_13.
- [14] Greschke, P., Schönemann, M., Thiede, S., and Herrmann, C. 2014. Matrix Structures for High Volumes and Flexibility in Production Systems. *Procedia 5CIRP* 17, 160–165.
- [15] Grunert, D., Jung, S., Leipold, T., Göppert, A., Hüttemann, G., and Schmitt, R. 2019. Service-oriented Information Model for the Model-Driven Control of Dynamically Interconnected Assembly Systems. In *Tagungsband des 4. Kongresses Montage Handhabung Industrieroboter*. Springer Berlin Heidelberg, Berlin, Heidelberg, 23–33.
- [16] Hüttemann, G., Buckhorst, A. F., and Schmitt, R. H. 2019. Modelling and Assessing Line-less Mobile Assembly Systems. *Procedia 5CIRP* 81, 724–729.
- [17] Hüttemann, G., Gaffry, C., and Schmitt, R. H. 2016. Adaptation of Reconfigurable Manufacturing Systems for Industrial Assembly – Review of Flexibility Paradigms, Concepts, and Outlook. *Procedia 5CIRP* 52, 112–117.
- [18] Kaelbling, L. P., Littman, M. L., and Moore, A. W. 1996. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research* 4, 1, 237–285.
- [19] Kagermann, H., Helbig, J., Hellinger, A., and Wahlster, W. 2013. *Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Securing the future of German manufacturing industry ; final report of the Industrie 4.0 working group*. Forschungsunion; Geschäftsstelle der Plattform Industrie 4.0, Berlin, Frankfurt/Main.
- [20] Kalyanakrishnan, S. and Stone, P. 2007. Batch reinforcement learning in a complex domain. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems - AAMAS '07*. ACM Press, New York, New York, USA, 1. DOI=10.1145/1329125.1329241.
- [21] Kern, W., Lämmermann, H., and Bauernhansl, T. 2017. An Integrated Logistics Concept for a Modular Assembly System. *Procedia Manufacturing* 11, 957–964.
- [22] Mayer, B. 2018. *Ich bin ein Fan von Effizienz*. <https://www.automobil-produktion.de/interviews-734/porsche-produktionsvorstand-reimold-wir-ziehen-alle-register-126.html?page=4>. Accessed 28 January 2019.
- [23] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540, 529–533.
- [24] Mueller, D., Mieth, C., and Henke, M. 2018. Quantification of Sequencing Flexibility Based on Precedence Graphs for Autonomous Control Methods. In *Proceedings of the 4th International Conference on Industrial and Business Engineering - ICIBE' 18*. ACM Press, New York, New York, USA, 211–220. DOI=10.1145/3288155.3288162.
- [25] Nguyen, T. T., Nguyen, N. D., and Nahavandi, S. 2020. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE transactions on cybernetics*.
- [26] Schönemann, M., Herrmann, C., Greschke, P., and Thiede, S. 2015. Simulation of matrix-structured manufacturing systems. *Journal of Manufacturing Systems* 37, 1, 104–112.
- [27] Shrestha, A. and Mahmood, A. 2019. Review of Deep Learning Algorithms and Architectures. *IEEE Access* 7, 53040–53065.
- [28] Sutton, R. S. and Barto, A. 2018. *Reinforcement Learning: An Introduction*. Adaptive computation and machine learning. The MIT Press, Cambridge, MA, London, England.
- [29] van Hasselt, H., Guez, A., and Silver, D. 2016. Deep Reinforcement Learning with Double Q-Learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16. AAAI Press, 2094–2100.
- [30] Watkins, C. J. C. H. and Dayan, P. 1992. Q-learning. *Machine Learning* 8, 3-4, 280–292.
- [31] Wiering, M. and van Otterlo, M. 2012. *Reinforcement Learning and Markov Decision Processes*. Adaptation, Learning, and Optimization 12. Springer, Heidelberg.
- [32] Xie, S., Zhang, T., and Rose, O. 2019. Online Single Machine Scheduling Based on Simulation and Reinforcement Learning. In *Simulation in Produktion und Logistik 2019*, M. Putz and A. Schlegel, Eds. Verlag Wissenschaftliche Scripten, Auerbach, 59–68.
- [33] Zhang, T., Xie, S., and Rose, O. 2017. Real-time Job Shop Scheduling Based on Simulation and Markov Decision Processes. In *Proceedings of the 2017 Winter Simulation Conference*. IEEE Inc, 3899–3907.
- [34] Zhang, T., Xie, S., and Rose, O. 2018. Real-Time Batching in Job Shops based on Simulation and Reinforcement Learning. In *Proceedings of the 2018 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 3331–3339.

DISPO 4.0 - Simulationsgestützte Absatzprognoseoptimierung in der Investitionsgüterindustrie

Alexander Schmid^{1*,2}, Felix Kamhuber¹, Thomas Sobottka^{1,2}, Wilfried Sihl¹

¹Fraunhofer Austria Research GmbH, Theresianumgasse 7, 1040 Vienna, Austria; *alexander.schmid@fraunhofer.at

²TU Wien, Institute of Management Science, Theresianumgasse 27, 1040 Vienna, Austria

Abstract. This paper presents a demand forecasting approach that automatically selects optimal article specific forecasting methods and optimizes the method parameters, using deterministic simulation and a *Genetic Algorithm* (GA). For an efficient demand forecast, choosing the best forecasting method based on the item-specific historical requirements time series is key. The optimization of the forecast parameters is also crucial for efficient demand planning. Both decisions lack digital method support, leading to suboptimal forecasts in practice and thus inefficient material requirements planning. This paper investigates the optimization potential of an automatically optimizing forecasting approach, featuring a simulation-based comparison of six standard forecasting methods, evaluated using a case-study from the capital goods industry. The methodological core of the optimization is a GA, which improves the underlying, method-specific forecast parameters. The simulation-based optimization provides a rolling-horizon demand forecast for each item, and is determined through the application of a rule-based heuristic. The results show a significant improvement potential through this form of efficient item-specific demand planning.

Einleitung

„Daten und Informationen sind das Öl des digitalen Zeitalters“ [1] und Algorithmen sind der Schlüssel für eine effiziente Verwertung großer Datenmengen [1]. Dies betrifft insbesondere auch die Materialdisposition, die sich in einem volatilen, globalen Marktumfeld mit steigender Komplexität [2] und zunehmender Menge an Informationen und Daten [3] konfrontiert sieht. Disruptionen durch Digitalisierung, kleinere Losgrößen, schwankende Absatzmengen, globalisierte Supply Chains sowie Kostendruck sind wesentliche Komplexitätstreiber in der

Materialdisposition [4]. Materialdisposition umfasst das Koordinieren der Materialeingänge in das Unternehmen und des Lagerbestands, so dass die richtigen Artikel zeitgerecht sowie ortsgerecht in der richtigen Qualität vorhanden sind [5]. Das Themengebiet Materialdisposition gliedert sich in die Bedarfs-, Bestands- sowie Bestellrechnung [5], siehe *Abbildung 1*. Um in einem Umfeld, ständiger Schwankungen, einen Wettbewerbsvorteil erzielen zu können, müssen Unternehmen rechtzeitig Entscheidungen auf genauer Datengrundlage treffen können. Richtige Entscheidungen basieren dabei auf genauen Bedarfsprognosen [6]. Jedes Teilgebiet der Materialdisposition kann durch digitalisierte Prozesse, mit Nutzung optimierender Algorithmen, große Ersparnisse erzielen [7].



Abbildung 1: Teildisziplinen der Materialdisposition

Dieses Paper befasst sich mit der Bedarfsrechnung und speziell mit der Anwendung von Prognosealgorithmen in der verbrauchsgesteuerten Materialdisposition der Investitionsgüterindustrie. Wobei sich diese Arbeit auf Zeitreihenmodelle bezieht, die den quantitativen Prognosemethoden zugeordnet werden [8]. Zeitreihenanalysen basieren immer auf historischen Daten. Dabei wird die historische Nachfrage eines Artikels analysiert, um die zukünftige Nachfrage vorhersagen zu können [8].

Bei der verbrauchsgesteuerten Bedarfsrechnung bilden die Materialverbrauchsdaten der Vergangenheit die Grundlage, um mit Hilfe von Prognosealgorithmen die zukünftigen Bedarfe zu ermitteln [9]. Hierbei wird dann von einer kundenanonymen Absatzplanung gesprochen

[10]. Dabei geht es bei der Bedarfsrechnung darum, unter Berücksichtigung aller verfügbaren Informationen und der vorhandenen historischen Daten, die zukünftigen Bedarfe so genau wie möglich vorherzusagen [11]. Heute existieren in der verbrauchsgesteuerten Materialdisposition vielfältige Verfahren und komplexe Logiken, um die Materialdisposition effizienter zu gestalten. Im betrieblichen Alltag findet jedoch zumeist nur ein sehr geringer Anteil mathematischer Modelle Anwendung, da anwendungsbereite Methoden zur Auswahl und fallspezifischen Parametrierung fehlen [12]. Prognosesysteme erfordern die Entwicklung von Fachwissen über die Ermittlung von Prognoseproblemen, die Anwendung einer Reihe von Prognosemethoden, die Auswahl geeigneter Methoden für jedes Problem und die Bewertung und Verfeinerung von Prognosemethoden im Laufe der Zeit [11]. Die Anwendbarkeit von Prognoseverfahren hängt von den Eigenschaften der zu prognostizierenden Zeitreihen, den Eigenschaften der Prognoseverfahren, vom Verwendungszweck der Prognosen ab [2]. Durch Digitalisierung und Automatisierung eines Prognosesystems kann eine artikelgenaue Materialbedarfsplanung und Ersatzteilbevorratung generiert werden. Genaue Bedarfsvorhersagen führen in der verbrauchsgesteuerten Materialdisposition zu einer Erhöhung der Versorgungssicherheit, zur Reduktion von Lagerbeständen, Lagerhaltungs- und Kapitalbindungskosten sowie zu Verminderung von Verschrottungskosten [12]. Fehlmengen und daraus resultierende Gewinneinbußen lassen sich ebenfalls reduzieren und die Kundenbindung kann durch verbesserte Kundenzufriedenheit gesichert werden [2]. Dieser Anwendungsbeitrag stellt die Entwicklung einer digitalen Planungsmethode vor, die eine artikelspezifisch optimierte Bedarfsrechnung in der verbrauchsgesteuerten Materialdisposition ermöglicht. Dabei werden mittels statischer Simulation und einer regelbasierten Heuristik als Bewertungskriterium artikelspezifisch möglichst optimale Prognoseverfahren ausgewählt und diese automatisiert mittels *Genetischem Algorithmus* (GA) parametrisiert. Das Nutzenpotenzial einer Anwendung von quantitativen Prognosemethoden in der verbrauchsgesteuerten Materialdisposition zur teilautomatisierten Bedarfsrechnung wird in einer Fallstudie evaluiert.

1 Zielsetzung

Zielsetzung dieses Beitrags ist es, mit der entwickelten Methode die artikelgenaue Bedarfsplanung von

verbrauchsgesteuert disponierten Artikeln zu generieren und zu automatisieren. Dabei gilt es, Fachwissen über Prognoseprobleme zur Auswahl einer geeigneten Prognosemethode sowie die Parametrisierung und Optimierung der Prognosemethoden zu berücksichtigen.

Die Forschungshypothese ist, dass die Entwicklung und der Einsatz einer digitalen Planungsmethode der verbrauchsgesteuerten Materialdisposition, mit statischer Simulation und einer regelbasierten Heuristik sowie einer Optimierung von Prognoseparametern mittels GA, unter Nutzung von State-of-the-art Verfahren der Prognoserechnung, eine artikelgenaue Materialbedarfsplanung in Unternehmen der Investitionsgüterindustrie deutlich verbessern kann.

2 Grundlagen Bedarfsermittlung

In einer Literaturanalyse wurden zunächst eine Übersicht der verfügbaren Prognoseverfahren ermittelt, die Algorithmen charakterisiert und die Einsatzmöglichkeiten im betrieblichen Umfeld der Investitionsgüterindustrie evaluiert. Die Analyse der Prognosealgorithmen wurde auf Zeitreihenmodelle der stochastischen Bedarfsrechnung eingeschränkt. Die Literaturanalyse zeigt auf, dass bei den Verfahren der *Exponentiellen* (Exp.) *Glättung* oft keine Unterscheidung hinsichtlich der Unterkategorien gemacht wird, obwohl andere Berechnungsalgorithmen dahinter liegen. **Abbildung 2** zeigt das Ergebnis der ermittelten Verfahren. Die dunkelgrau markierten Verfahren wurden nach einer Häufigkeitsanalyse in der Literatur als die gebräuchlichsten Verfahren ausgewählt und in der entwickelten Planungsmethode berücksichtigt. Diese sechs Algorithmen finden teilweise schon Anwendung in ERP (*Enterprise Resource Planning*)-Systemen. Jedoch fehlt VerantwortungsträgerInnen in den Unternehmen eine Entscheidungsgrundlage, welche der Prognoseverfahren für des Unternehmen die geeignetsten und wie diese optimal und artikelspezifisch zu parametrisieren sind. Die Eignung eines Prognoseverfahrens für einen konkreten Anwendungsfall kann anhand der artikelspezifischen Zeitreihe der historischen Absatzzahlen (Bedarfscharakteristik) eingeschätzt werden [10]. Zeitreihen aus den historischen Absatzzahlen können eine Vielzahl von artikelspezifischen Mustern aufweisen. Daher ist es hilfreich eine Zeitreihe in mehrere Komponenten zu unterteilen, die jeweils eine zugrunde liegende Musterkategorie (Trend, Saisonalität etc.) erfasst [11].

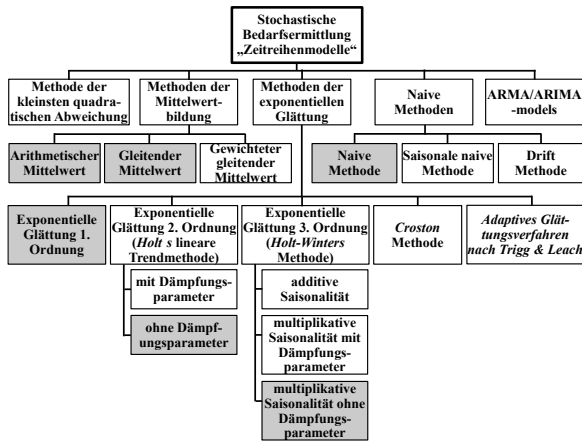


Abbildung 2: Übersicht Zeitreihenmodelle

Zeitreihentypen werden wie in **Abbildung 3** ersichtlich kategorisiert. Schwanken die Vergangenheitswerte um einen Durchschnittswert, nennt man diesen Bedarfsverlauf konstant. Der trendförmige Verlauf zeichnet sich durch einen stetig steigenden oder fallenden Bedarf mit zufälligen Schwankungen um diesen aus und der saisonale Bedarf erscheint regelmäßig in Perioden und schwankt auch stochastisch um den Durchschnittswert [10]. Der sporadische Bedarfsverlauf schwankt ohne erkennbare Regularität [13].

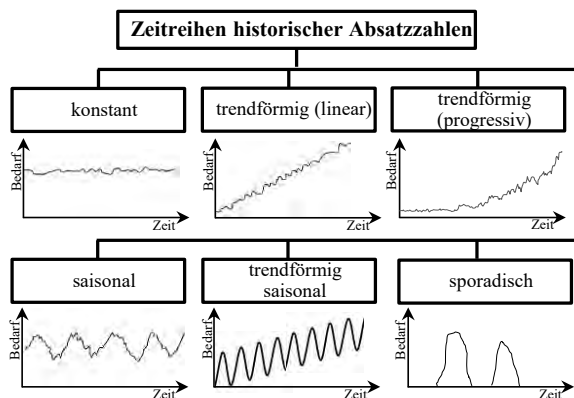


Abbildung 3: Zeitreihen historischer Absatzzahlen (Eigendarstellung in Anlehnung an: [13,14])

Laut Literaturrecherche gibt es keine allgemeingültige Zuordnung, die artikelspezifischen Zeitreihen geeignete Prognosealgorithmen eindeutig zuordnet, siehe **Abbildung 4**. Um in der Anwendungspraxis das bestgeeignete Verfahren artikelspezifisch effizient auswählen zu können und dabei auch auf Veränderungen der Bedarfscharakteristik im Zeitverlauf dynamisch reagieren zu können, wird die hierin vorgestellte Methode entwickelt.

Zeitreihen historischer Absatzzahlen	konstant	trendförmig (linear)	trendförmig (progressiv)	saisonal	trendförmig saisonal	sporadisch
Prognosealgorithmen						
Arithmetischer Mittelwert	●					
Gleitender Mittelwert	●					
Gewogener gleitender Mittelwert	●	○				○
Exponentielle Glättung 1. Ordnung	●					○
Exponentielle Glättung 2. Ordnung (gedämpft)	●	●				
Exponentielle Glättung 2. Ordnung (ungedämpft)	●	●	●			
Exponentielle Glättung 3. Ordnung (additiv)	●	●	●	●		
Exponentielle Glättung 3. Ordnung (multiplikativ gedämpft)	●	●	●	●	●	
Exponentielle Glättung 3. Ordnung (multiplikativ ungedämpft)	●	●	●	●	●	
Croston-Methode						●
Adaptives Glättungsverfahren nach Trigg & Leach		○				○
ARIMA Modell	●	●	●	●	●	○
Legende: ● geeignet ○ bedingt geeignet						

Abbildung 4: Zusammenhang Zeitreihen – Prognosealgorithmen (Eigendarstellung in Anlehnung an: [15,13,10,16])

3 Charakterisierung der Fallstudie

Die Fallstudie wurde anhand eines Unternehmens der Investitionsgüterindustrie betrachtet, welches die Ersatzteildisposition verbrauchsgesteuert abwickelt. Reale Ersatzteilverbräuche der vergangenen 36 Monate und weitere relevanten Unternehmensdaten wurden betrachtet. Von den identifizierten 17.600 aktiven Ersatzteilartikeln mit Absatzzahlen, wurden nur jene Artikel für die Fallstudie herangezogen, die im Betrachtungszeitraum (-36 Monate) zumindest in 24 der 36 Monate Ersatzteilabsätze aufwiesen. Daraus resultieren für die Betrachtung dieser Arbeit 2.002 unterschiedliche Ersatzteilartikel.

Das betrachtete mittelständische Unternehmen wird der Branche der Fahrzeugindustrie zugeordnet. Mit ca. 240 Mitarbeitern erwirtschaftet das Unternehmen einen Jahresumsatz von ca. 89 Mio. € mit 38.800 Kunden aus 22 Ländern. Das jährliche Einkaufsvolumen von ca. 59 Mio. € wird mit 1.260 aktiven Lieferanten aus 28 Ländern umgesetzt. Als zentrales IT-System wird *Infor M3* verwendet. Die Input-Daten werden aus den IT-Systemen des Unternehmens via Schnittstelle eingelesen.

4 Digitale Planungsmethode

In der vorliegenden Arbeit wird ein auf MS (*Microsoft*)-Excel und VBA (*Visual Basic for Applications*) basierende digitale Planungsmethode für eine optimierte Bedarfsrechnung der verbrauchsgesteuerten Materialdisposition in der Investitionsgüterindustrie entwickelt. Mit

dieser Methode soll das Prognostizieren zukünftiger artikelspezifischer Bedarfe von den Erfahrungen einzelner gelöst, bewährte Verfahrensweisen allgemein nutzbar gemacht und die Nachvollziehbarkeit von Entscheidungen bis auf Articlebene auf Basis eines Simulationsmodells erleichtert werden. **Abbildung 5** zeigt den Ablauf des digitalen Planungsmethode im Überblick.

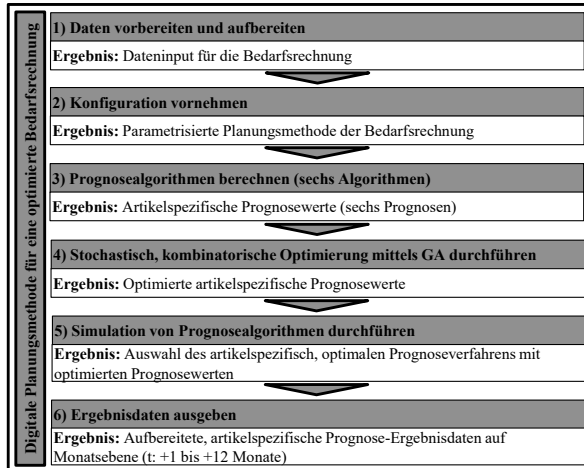


Abbildung 5: Schrittfolge der digitalen Planungsmethode

Die Planungsmethode umfasst sechs Arbeitsschritte, wobei die Rechenzeit v.a. in Schritt vier anfällt. Bis auf diesen werden alle anderen Arbeitsschritte, je nach Umfang des Datensatzes (bis zu ~1 Mio. Artikel), ohne relevanten Zeitverzug ausgeführt.

Die Modellierung der Planungsmethode zielt auf eine möglichst hohe Prognosequalität ab. Dabei sollen sich ändernde Marktbedingungen berücksichtigt werden. Zielsetzung ist eine zukünftige Digitalisierung bzw. Automatisierung der Bedarfsrechnung in der verbrauchsgeordneten Materialdisposition.

Der methodische Kern der Optimierung ist ein GA, der die jeweils zugrundeliegenden, methodenspezifischen Prognoseparameter im Zuge von Schritt vier optimiert wählt. Die simulationsgestützte Optimierung liefert für jede der sechs Methoden eine rollierende Absatzprognose auf Monatsbasis für die mittlere Zukunft sowie den sich daraus in Summe ergebenden Prognosefehler, je betrachteten Artikel. Durch Anwendung einer regelbasierten Heuristik wird das für den Artikel optimale mathematische Prognosemodell ausgewählt. Das Bewertungskriterium ist der, in Bezug auf den Referenzzeitraum, geringste Prognosefehler. Für diese Arbeit wird konkret der *Mean Squared Error* (MSE) herangezogen.

4.1 Daten vorbereiten und aufbereiten

Die in dieser Arbeit angewendeten Verfahren zur Bedarfsermittlung (siehe **Abbildung 2**) unterstellen einen Zusammenhang zwischen dem Verbrauch in der Vergangenheit und dem Bedarf in zukünftigen Perioden. Die Grundlage der stochastischen Bedarfsermittlung bilden effektive Verbrauchsdaten aus der Vergangenheit [9]. Der erste Schritt ist die Aufbereitung und Formatierung der Daten. Dieser Prozess beinhaltet einen standardisierten Bezug historischer Datensätze, das Erkennen fehlender Werte, das Filtern der artikelspezifischen Zeitreihen sowie weitere Vorverarbeitungsaufgaben [11]. Dabei werden unterschiedliche Datenquellen aus dem ERP-System eingespielt und automatisiert aufbereitet. Folgend werden artikelspezifische, tagesaktuelle Verbrauchswerte zu Monatsabsätzen aggregiert.

4.2 Konfiguration vornehmen

Dieser zweite Arbeitsschritt bestimmt die Parametrisierung und Start-Konfiguration der einzelnen Algorithmen und berücksichtigt dabei die Einstellung der folgenden prognose- und optimierungsrelevanten Parameter:

- Überdeckung: Prognosewert in Periode $x >$ Absatzhistorie in Periode x
- Unterdeckung: Prognosewert in Periode $x <$ Absatzhistorie in Periode x
- Einstellung der Startwerte bei den Prognoseparametern α , β und γ : 0,3; 0,5; 0,5
- Einstellung der Grenzwerte je Methode als Nebenbedingung für die anschließende Prognoseparameteroptimierung mittels GA
- Maximale Optimierungs-Laufzeit: drei [s]

Für das Verhältnis von Unter- zu Überdeckung wird für alle Prognosemethoden 2:1 festgelegt. Unterdeckung ist also, aufgrund der negativen Effekte (Pönalen, entgehen des Geschäft), entsprechend stärker gewichtet als Überdeckung. Umfangreiche Datentests haben zur Einstellung der genannten Startwerte für die Prognoseparameter geführt. Folgende weitere Parameter werden gesetzt:

- Anzahl Perioden Absatzhistorie: -36 Perioden
- Anzahl Prognoseperioden: +12 Perioden

Ein längerer Prognosehorizont ist in den meisten Fällen nicht sinnvoll, da veraltete Daten nicht mehr aussagekräftig sind und ein zu langer Prognosezeitraum die

Planungsqualität verringert. In einem separaten Arbeitsblatt je Prognosemethode werden die Monatsabsätze je Artikel als Basis für die folgenden Schritte gespeichert.

4.3 Prognosealgorithmen berechnen

Im Zuge der Berechnung bzw. Optimierung des Prognosefehlers, in diesem Fall der Prognosefehler *MSE*, wird für alle sechs Verfahren artikelweise die Absatzhistorie in die Berechnung übernommen. Für jeden Artikel, für den eine Absatzhistorie vorliegt, werden die folgenden formalisierte sechs Prognoseverfahren berechnet und das artikelspezifische Prognoseergebnis sowie der entsprechende Prognosefehler im Arbeitsblatt je Methode gespeichert.

Die **Naive Methode**, oder *Random Walk*, ist die einfachste Methode der stochastischen Bedarfsermittlung [9]. Aber auch einfache Verfahren können sich als effektiv erweisen, falls die Absatzhistorie ein für die komplexeren Verfahren nicht erkennbares Muster aufweist [11]. Bei der *Naiven Methode* ergeben sich die zukünftigen Prognosewerte direkt aus den Vorperioden-Werten.

Der **Arithmetische Mittelwert** ist die einfachste Form der Berechnung aus den drei Methoden der Mittelwertbildung. Mit diesem Prognosealgorithmus lassen sich konstante Bedarfsverläufe ohne Trend darstellen [17]. Durch die stetige Erhöhung der in die Mittelwertberechnung miteinbezogenen Werte sinkt der Einfluss der aktuellen Perioden immer weiter, was einen entscheidenden Nachteil dieser Methode darstellt [18].

Der **Gleitende Mittelwert** bezieht sich im Gegensatz zum *Arithmetischen Mittelwert* nur auf die letzten *m* aktuellen Perioden und verwirft die älteren Aufzeichnungen, um besser auf den aktuellen Bedarfsschwankungen reagieren zu können [18]. Alle *m* Werte sind gleich gewichtet wie auch beim *Arithmetischen Mittelwert*. Je nachdem, wie viele Perioden für die Berechnung verwendet werden, reagiert der Prognosewert träger oder sensibler. Bei einer hohen Anzahl an betrachteten Perioden reagiert der *Gleitende Mittelwert* nur langsam auf Veränderungen und bei einer geringen Anzahl reagiert er hingegen äußerst sensibel [13].

Die **Exponentielle Glättung 1. Ordnung**, auch *Exponential Smoothing* und *Brown's Model* benannt [19], ist ähnlich den Methoden der Mittelwertbildung. Jedoch wird zusätzlich der Fehler der vergangenen Periode mit einem Glättungsfaktor α multipliziert. Der Fehler der vergangenen Periode ergibt sich durch die Differenz von dem tatsächlich eingetretenen Wert der Vorperioden

X_t mit dem geschätzten Prognosewert X_t^* derselben Periode. Der Glättungsfaktor kann alle Werte zwischen 0 und 1 annehmen, wobei ein hoher Glättungsfaktor die aktuellen Daten höher wertet, als die alten Daten [17]. Basierend auf einer Trendbildung wird bei der *Exp. Glättung 1. Ordnung* die Absatzhistorie der jüngeren Vergangenheit charakteristisch höher gewichtet, siehe **Abbildung 6**. Das Risiko des Verfahrens besteht darin, dass dem Trend verzögert nachgelaufen wird, während sich dieser evtl. bereits abbaut.

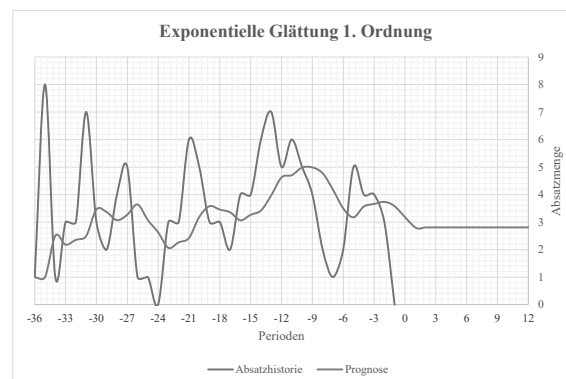


Abbildung 6: Absatzhistorie und Prognose (*Exp. Glättung 1. Ordnung* – Artikel 455 Scheibenwaschbehälter)

Die **Exponentielle Glättung 2. Ordnung** wurde von Holt durch Erweiterung der *Exp. Glättung 1. Ordnung* entwickelt, um Daten mit einem Trendverlauf prognostizieren zu können. Den Glättungsparameter α kann man mit dem der *Exp. Glättung 1. Ordnung* gleich setzen, aber der Trend-Glättungsparameter β legt die Steigung fest. Die beiden Glättungsparameter sind unabhängig voneinander und auch β muss einen Wert zwischen 0 und 1 annehmen. Wenn $\beta = 0$ gesetzt wird, ergibt sich die *Exp. Glättung 1. Ordnung* [10]. Durch die nochmalige Glättung der Vorhersagewerte der *Exp. Glättung 1. Ordnung* über sogenannte Trendkorrekturwerte reagiert die *Exp. Glättung 2. Ordnung* je nach α entsprechend schneller auf Trendänderungen. **Abbildung 7** zeigt exemplarisch die rasche (und angesichts der Historie notwendige) Anpassung der Prognose dieser Methode bei $\alpha = 0,40$.

Die **Exponentielle Glättung 3. Ordnung**, auch *Holt-Winter* Methode ist eine Erweiterung der *Exp. Glättung 2. Ordnung*, um auch die saisonale Abhängigkeit von Daten erfassen zu können. Sie kann additive oder multiplikative Saisonalität einbeziehen [20]. Das Verfahren von *Holt-Winter*, berücksichtigt zusätzlich zum Trend einen (multiplikativen) Saison-Anteil in der Prognose.

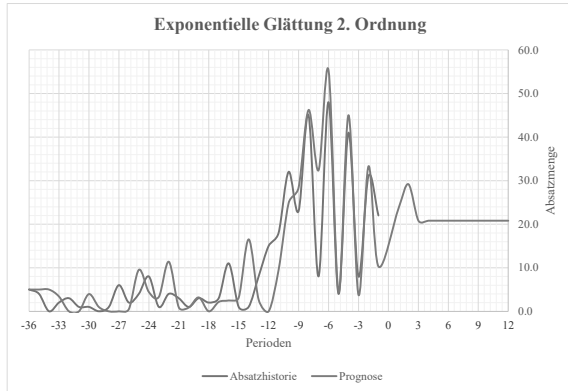


Abbildung 7: Absatzhistorie und Prognose (*Exp. Glättung 2. Ordnung* – Artikel 1181 Set: Heugabel, Schaufel, Stallbesen)

Dieser Anteil berücksichtigt das Verhalten der vergangenen Saison je nach Parametrierung entsprechend stark, siehe **Abbildung 8**. Zu beachten ist, dass die klassischen additiven und multiplikativen *Holt-Winters*-Methoden unzuverlässig werden können, wenn das Rauschen der historischen Zeitreihe die Trend-Komponente β und Saisonale-Komponente γ dominiert [20].

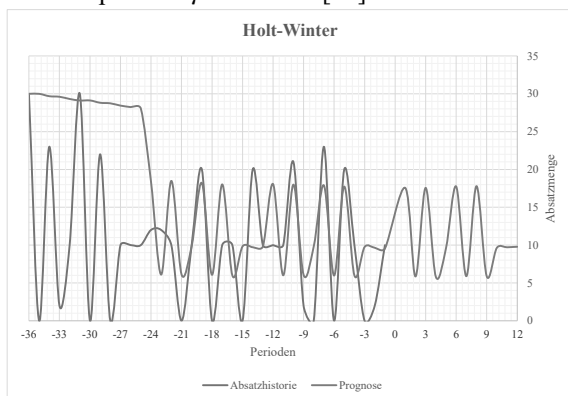


Abbildung 8: Absatzhistorie und Prognose (*Holt-Winter* – Artikel 3 Klemmschelle mit Schraube)

4.4 Optimierung mittels GA durchführen

Bei den Methoden der *Exp. Glättung* handelt es sich um weit verbreitete und vielfach genutzte Verfahren zur Erstellung von Prognosen, indem ein instationärer Prozess abgebildet wird. Dabei erhalten aktuelle Beobachtungen eine höhere Gewichtung als die älteren Beobachtungen. Die Gewichtung der früheren Beobachtungen nimmt exponentiell ab [2], [21]. Die Grundformel der *Exp. Glättung* lautet: Neue Vorhersage = Alte Vorhersage + (Glättungsfaktor x Prognosefehler). Die richtige Wahl des Glättungsfaktors spielt daher bei den Verfahren der *Exp.*

Glättung eine entscheidende Rolle. In der Praxis haben sich folgende Werte bewährt [10]:

$0,05 \leq \alpha \leq 0,3$	für annähernd konstanten Verlauf
$0,3 \leq \alpha \leq 0,5$	für stark schwankenden Bedarfsverlauf
$\alpha \geq 0,5$	nicht empfehlenswert

Die Randwerte der Prognoseparameter bei den anderen *Exp. Glättungsverfahren* liegen für α , β und γ im Intervall $[0;1]$. Die Startlösung basiert auf den Optimierungsergebnisse eines Datensatzes (> 17.000 Artikel) und wird für $\alpha = 0,2$; $\beta = 0,5$; $\gamma = 0,5$ initialisiert.

Die Optimierung der (1 bzw. 3) Prognoseparameter erfolgt für die drei Verfahren der *Exp. Glättung* (1. bis 3. Ordnung). Die anderen drei Methoden verfügen über keine zu optimierenden Prognoseparameter. Für die Parameteroptimierung wird der GA von MS-Excel genutzt, der über einen VBA-Code aufgerufen wird. Darin erfolgt die Übergabe der Zielfunktion mit Minimierung des Prognosefehlers, der Variablenzellen und der Nebenbedingungen, inklusive der dazugehörigen Einstellungen für den Algorithmus, z.B. der Laufzeit/Abbruchkriterien. Die **Abbildung 9** zeigt exemplarisch die Verteilung der Optimierungsparameter (Glättungs-Komponente α , die Trend-Komponente β und Saisonale-Komponente γ) der 506 Artikel der Methode *Exp. Glättung 3. Ordnung*, die in diesem Fallbeispiel Anwendung finden.

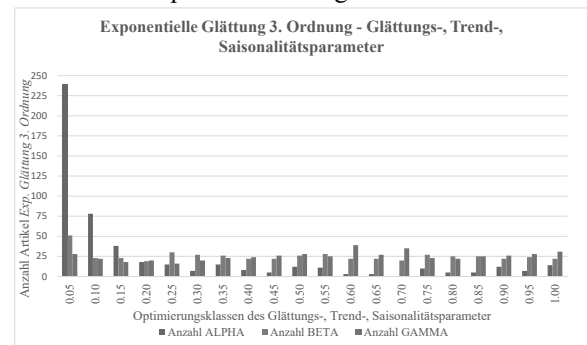


Abbildung 9: *Exp. Glättung 3. Ordnung* - Ergebnis der Parametertoptimierung von Alpha, Beta und Gamma $[0,00-1,00]$

4.5 Simulation von Prognosealgorithmen

Ein wichtiger Punkt ist die Prognosequalität, welche die mit der entsprechenden Methode berechneten Prognosewerte mit den tatsächlich eingetretenen historischen Verbrauchswerten vergleicht, um somit auf eine Aussagekraft über die jeweilige Prognosegüte schließen zu können [22]. Aus den möglichen Prognosefehlern wurde der *MSE* im Rahmen dieser Arbeit zur Bewertung

herangezogen, da durch Quadrieren jene Werte mit größeren Prognoseabweichung eine stärkere Gewichtung bekommen [23]. Ermittelt wird der MSE wie folgt [10]:

$$MSE = \frac{1}{n} * \sum_{t=1}^n (y_t - \hat{y}_{t+h|t})^2 \quad (1)$$

$\hat{y}_{t+h|t}$ Prognosewert für die Periode t

y_t Realer Bedarfswert

n Anzahl der Stichproben des Periodenverlaufs

Im Zuge der Simulation werden pro Artikel die sechs Prognoseverfahren verglichen und die Methode mit dem geringsten Prognosefehler *MSE* für den konkreten Artikel als geeignetste Prognosemethode ausgewählt. **Tabelle 1** stellt eine Übersicht der entsprechend pro Artikel zur Anwendung gewählten Verfahren für den Datensatz dar. Die am häufigsten selektierten Verfahren sind die Verfahren der *Exp. Glättung 1. bis 3. Ordnung*.

Implementierte Prognosealgorithmen	# Artikel	%-Anteil
Arithmetischer Mittelwert	174	8,7 %
Exponentielle Glättung 1. Ordnung	404	20,2 %
Exponentielle Glättung 2. Ordnung ohne Dämpfungsparemeter	743	37,1 %
Exponentielle Glättung 3. Ordnung multiplikative Saisonalität ohne Dämpfungsparemeter	506	25,3 %
Gleitender Mittelwert	48	2,4 %
Naive Methode	118	5,9 %
Nicht prognostizierbar	9	0,4 %
Summe	2.002	100,0%

Tabelle 1: Ergebnisdaten

Das Optimierungspotential des jeweiligen Prognosefehlers beträgt anhand einer Auswertung für den Datensatz aus der Fallstudie im Mittel 32%. Der Wert ist groß, in Anbetracht des Umstands, dass nur drei von sechs Verfahren die Prognoseparameter optimieren. Die drei parameteroptimierten Verfahren werden für 82,6% der Artikel empfohlen. Die detaillierte Auswertung der Optimierungsergebnisse ergibt, dass der Prognosefehler mittels der Prognoseparameter optimierten Methoden wie folgt ggü. der jeweiligen Startlösung reduziert werden konnte:

- *Exp. Glättung 1. Ordnung*: 28,1%
- *Exp. Glättung 2. Ordnung*: 39,6%
- *Holt-Winter-Methode*: 45,6%

Trotz des nicht vorhandenen Optimierungspotentials bei den Methoden *Arithmetischer Mittelwert* und *Gleitender Mittelwert* sowie der *Naiven Methode* wird eine dieser Methoden lt. **Tabelle 1** in ~17% aller Fälle zur Anwendung als Prognosemethode empfohlen. Dies trifft v.a. auf Artikel zu, die keinem Trend bzw. keiner Saisonalität unterliegen. Der Unterschied im Prognosefehler beträgt zwischen der besten und schlechtesten Methode 72,8%. Die Wahl des „richtigen“ Verfahrens für einen bestimmten Artikel zur Prognose ist demnach wirkungsvoll. Der Unterschied zwischen der aktuell zur Prognose angewendeten *Naiven Methode* und der aus der Optimierung resultierenden Prognosemethode beträgt 68,5%.

4.6 Ergebnisdaten ausgeben

Am Ende erfolgt eine, an die Fallstudie angepasste, artikelspezifische Ausgabe der Optimierungsergebnisse. Diese beinhalten insbesondere die Absatzprognose der jeweils geeignetsten Prognosemethode sowie den dazugehörigen Prognosefehler *MSE*. Die ergänzende Prognosefehler *Mean Absolute Deviation*, *Mean Absolute Percentage Error* sowie die Standardabweichung der Prognose werden für eine weiterführende Bestandsrechnung [24] ebenfalls je Artikel berechnet. Außerdem werden die Ergebnisse mit den artikelspezifischen, dispositiven Daten verknüpft und zur weiteren Verwendung ausgegeben.

5 Diskussion

Die zentrale Schlussfolgerung ist, dass die Parameteroptimierung einen signifikanten Einfluss auf die Güte der Absatzplanung hat, weil der Prognosefehler im Mittel um 32% reduziert werden konnte. Die Differenz zwischen der *Naiven Methode* und der durch die Optimierung als geeignetste ermittelte Methode, inklusive Parameteroptimierung, beträgt 68,5%. Die Simulation ermöglicht je Artikel einen Vergleich mehrerer Verfahren und identifiziert je Artikel das bestgeeignete Verfahren. Die Herausforderung für die Materialdisposition besteht darin, für viele zu prognostizierende Artikel das jeweils bestgeeignete Verfahren mit der dazugehörigen Parametrisierung zu wählen. Das vorgestellte digitale Planungsmethode automatisiert diese beiden Entscheidungsfindungsprozesse. Die Auswertung der meistgewählten Verfahren zeigt: Der *Gleitende Mittelwert* wird selten gewählt, da die Verfahren der *Exp. Glättung 1. bis 2. Ordnung* mit optimierten Prognoseparametern deutlich überlegen sind. Die Verfahren mit integrierter Parameteroptimierung

sind in der Fallstudie den einfacheren Verfahren insgesamt überlegen. Das große Optimierungspotential der Methode von *Holt-Winter* (45,6%) lässt sich mit der Parametrisierung der Startlösung erklären, die einer realistischen pauschalen Parametrierung in der Praxis entspricht. Es ist wesentlich schwieriger, eine gute Startlösung für einen Artikel mit drei Prognoseparametern zu ermitteln als für Verfahren mit einem bzw. keinem Prognoseparameter. Die Durchführung der Absatzplanung lediglich mit einer Planungsmethode, wie in der Praxis häufig genutzt, resultiert in suboptimaler Planungsqualität. Dies lässt sich mit der in diesem Beitrag vorgestellten digitalen Planungsmethode für die Investitionsgüterindustrie in Zukunft optimieren. Im Ausblick ist zum einen eine Analyse weiterer Anwendungsfälle zu sehen, um ggf. allgemeingültige Muster der Zuordnung zu erkennen. Zum anderen kann die Möglichkeit untersucht werden, auch über die bislang genutzten Standard-Prognoseverfahren hinaus, intelligente Prognosemethoden zu entwickeln, bspw. Verfahren des *Machine-Learning* nutzend. Forschungsziel ist die Entwicklung einer integrierten, übergreifenden Methode (umfasst Bedarfs-, Bestands- und Bestellrechnung), und damit das Überwinden der sequenziellen Planung der drei Teilplanungen.

Danksagung

Diese Forschung wurde von der Österreichischen Forschungsförderungsgesellschaft (FFG), Förderungsnummer 872787, finanziert. Die Autoren danken den Fallstudienpartnern für ihren Beitrag.

Literatur

- [1] Barton, T., Müller, C., et al., 2018. Digitalisierung in Unternehmen. Springer Fachmedien, Wiesbaden.
- [2] Claus, T., Herrmann, F., et al., 2015. Produktionsplanung und -steuerung: Methoden und deren Anwendungen. Springer, Berlin, Heidelberg.
- [3] Bienhaus, F., Haddud, A., 2018. Procurement 4.0: factors influencing the digitisation of procurement and supply chains. BPMJ 24 (No. 4), S. 965–984.
- [4] Arnolds, H., Heege, F., et al., 2016. Materialwirtschaft und Einkauf. Springer Fachmedien, Wiesbaden.
- [5] Jacob, M., 2013. Management und Informationstechnik: Eine kompakte Darstellung. Springer, Wiesbaden.
- [6] Kolade, O.J. Demand Forecasting and Measuring Forecast Accuracy in a Pharmacy, in: OECONOMICA, AU-DOE, Vol. 15, No. 3/2019, S. 157–169.
- [7] Heiserich, O.-E., Helbig, K., et al., 2011. Logistik: Eine praxisorientierte Einführung, 4. Aufl. Gabler, Wiesbaden.
- [8] Gasparian, M.S., Karmanov, M.V., et al., Modeling of the demand forecasting, in: International Journal of Civil Engineering (IJCIET), Vol. 9, Issue 12, 2018, S. 163–173.
- [9] Pfohl, H.-C., 2018. Logistiksysteme. Springer, Berlin, Heidelberg.
- [10] Schuh, G., Schmidt, C., 2014. Produktionsmanagement. Springer, Berlin, Heidelberg.
- [11] Hyndman, R.J., Athanasopoulos, G., 2021. Forecasting: Principles and Practice: A comprehensive introduction to the latest forecasting methods using R. Learn to improve your forecast accuracy using dozens of real data examples., 3rd edition ed. OTexts, Melbourne (Australia).
- [12] Wischmann, S., Hartmann, E.A. 2018. Zukunft der Arbeit: Eine praxisnahe Betrachtung. Springer, Berlin, Heidelberg.
- [13] Schönsleben, P., 2020. Integrales Logistikmanagement: Operations und Supply Chain Management innerhalb des Unternehmens und unternehmensübergreifend, 8. Aufl. ed. Springer, Berlin, Zürich.
- [14] Waters, D., 2008. Inventory Control and Management. John Wiley & Sons Ltd, West Sussex.
- [15] Lasch, R., 2021. Strategisches und operatives Logistikmanagement: Beschaffung, 3. Aufl. Springer Gabler, Wiesbaden.
- [16] Wannenwetsch, H., 2014. Integrierte Materialwirtschaft, Logistik und Beschaffung, 5. Aufl. Springer, Berlin, Heidelberg.
- [17] Magerhans, A., 2016. Marktforschung: Eine praxisorientierte Einführung. Springer Gabler, Wiesbaden.
- [18] Koether, R., 2018. Distributionslogistik: Effiziente Absicherung der Lieferfähigkeit, 3. Aufl. Springer Gabler, Wiesbaden.
- [19] Babai, M.Z., Tsadiras, A., et al., 2020. On the empirical performance of some new neural network methods for forecasting intermittent demand. Journal of Management Mathematics (IMA) 31 (No. 3), S. 281–305.
- [20] Ferbar Tratar, L., Mojškerc, B., et al., 2016. Demand forecasting with four-parameter exponential smoothing. International Journal of Production Economics 181, S. 162–173.
- [21] Speckenbach, J., 2017. Prognose sporadischer Nachfragen: Ein Vergleichsverfahren. Josef Eul Verlag GmbH, Ingolstadt, Lohmar - Köln.
- [22] Kellner, F., Lienland, B., et al., 2018. Produktionswirtschaft: Planung, Steuerung und Industrie 4.0. Springer Gabler, Berlin, Heidelberg.
- [23] Feindt, M., Kerzel, U., 2015. Prognosen bewerten: Statistische Grundlagen und praktische Tipps. Springer Gabler, Berlin, Heidelberg.
- [24] Schmid, A., Sobottka, T., et al., DISPO 4.0 | Digitalization Of Inventory Calculation In Consumption-Based Material Requirements Planning In The Capital Goods Industry, in: Conference on Production Systems and Logistics, 3rd Conference CPSL 2022, Vancouver, 17.-20. May 2022.

Towards Increasing Resilience of Public Transit Infrastructure – A Bi-Modal Simulation Model

Daniel Lückerrath *, Oliver Ullrich

Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS, Schloss Birlinghoven, 53757 Sankt Augustin, Germany; *daniel.lueckerath@iais.fraunhofer.de

Abstract. With the impact of climate change increasing, public service infrastructure has to become more resilient against extreme weather events as well as human-made disasters. Public transit is a central part of urban infrastructure, often mainly consisting of interacting light-rail as well as express and community bus networks and connected to national rail and individual traffic systems. To increase such a system's resilience against small disturbances and larger outages – as they might result from climate change – service providers need a toolbox of potential measures to mitigate such incidents' impact and to re-establish services as soon as possible after an outage. This paper presents thoughts towards a bi-modal urban transit simulation system covering both light rail and (express and community) bus networks. Important aims of the system are a) to enable operators to evaluate measures against small disturbances and larger outages as they happen, and b) to evaluate what combination of disaster risk management and resilience-building strategies shows most potential to help increasing the resilience of urban transit systems against extreme weather events resulting from climate change as well as other disasters.

Introduction

Urban infrastructure systems are critical for everyday life, their functions serve both the social and economic well-being of urban residents and commuters. Infrastructure is defined as including all types of publicly and privately operated communication, electricity, and water networks, food production, waste treatment, industrial facilities, as well as urban transportation.

Over the last decades, infrastructure systems that were perceived up until then as isolated services have transformed to connected ecosystems; tightly organised networks provided by a multitude of actors, involving a myriad of physical and digital structures, and offering

services to society through all sorts of physical and digital channels. That includes the different modes of urban transit, including light rail systems, express and community buses, and at least partially integrated individual transportation services like taxi cabs, Uber, and Lyft. Out of all commonly available public transit modes, light rail and bus transit have the highest transit performance [1].

To protect their long-term utility, those integrated transit infrastructure components must be resilient against the increasing impact of climate change on urban spaces and systems, including pluvial and fluvial flooding, heat waves, droughts, and windstorms [2].

In case of sudden disasters impacting transit systems, including extreme weather and human-made events, operators have to be able to make decisions fast to a) transfer the infrastructure components into a pre-planned disaster mode and b) to be able to re-establish services as soon as the immediate event has passed. These operators can be assisted with a simulation application covering both light rail and bus transit that executes simulation runs sufficiently fast to enable evaluation and comparison of potential decisions and strategies, thereby contributing to increase the resilience of the transit system.

This paper presents steps towards the design of a bi-modal simulation model representing an urban area's light rail and bus transit network, designed to assist with increasing the transit infrastructure system's resilience against extreme weather events and human-made disasters. A specific focus is put on a) fast execution and b) the representation of operating decisions necessary in disaster risk management situations.

Many models representing urban transit are developed as an extension of already established models of individual traffic [3][4][5]. Generally, many of the more recent simulation models including bus transit use microscopic agent-based modeling approaches

[3][4][6][7], the mesoscopic approach to bus transit simulation proposed by Toledo et al. [5] extends a mesoscopic simulation model for individual traffic based on queuing theory proposed by Burghout [8], which represents the street network as a graph of inter-connected queues and vehicles as individual entities traversing these queues based on speed/density functions.

Especially models utilizing a fine-grained modeling approach generally necessitate the availability of an extensive data basis, including detailed information on origin-destination matrices, vehicular dynamics, signaling strategies, and lane changing rules [9], and include many components which are not immediately interesting for public transit resilience management. This often leads to long runtimes [10][11], thereby rendering those models inadequate for the use case described above. Therefore, this paper builds upon the work presented in [12] by extending a runtime-efficient bus transit model to include light rail transit.

The paper continues by sharing some background on the core components and concepts of urban transit systems and urban transit resilience (Section 1) and then introduces design decisions for a fast bi-modal transit simulation model (Section 2). It concludes with an outlook on necessary further research steps (Section 3).

1 Urban Transit

1.1 Urban Transit Components

Urban transit usually consists of a number of interacting networks, e.g. a light rail system, express and community bus networks, often connected at specific hubs to national rail systems as well as to individual transit systems like taxi cabs, Uber, or Lyft. For the presented model individual transit as well as national rail stations/airports are parameterized and not part of the core model itself.

A mixed light-rail and bus network consists of a network of street and rail segments as well as stops and stations where passenger exchanges take place. These stops and stations are served by a set of transit vehicles executing service trips, i.e., pairings of starting times and sequences of stops, according to a timetable. Each individual vehicle executes several service trips, interspersed with deadheads, over the course of an operational day, which is called a rotation. Such a rotation usually begins with a deadhead from the vehicle's depot

to the first stop of its first service trip and, after a number of service trips, ends with a returning deadhead to the depot. The rotation schedule defines the assignment of specific vehicles to rotations.

While some stops, mainly bus stops, include a bay with capacity for more than one vehicle, many other stops can contain only one vehicle at any given time. Some stops are marked as control points, i.e., locations in the network where control strategies may be employed, e.g., purposely delaying early vehicles until the scheduled departure time is reached. At other stops, vehicles depart as soon as the passenger exchange is completed. Each stop belongs to exactly one station, i.e., a geographically grouped collection of stops which usually share a common name.

Directed paths through the network, connecting two successive stops are called connections. They usually consist of several street and/or rail segments, junctions, and signals, that in turn can be shared by several connections.

Signals control access to individual segments, usually at junctions. Often, two or more signals constitute a signal group with a common scheduling strategy.

Urban transit vehicles generally follow pre-defined line routes, consisting of sequences of stops to be serviced. Often, a line consists of a number of line variants: while a main variant might be served by a majority of vehicles, some variants might contain only stops in the city center but not in the suburbs, others might branch off the trunk route to connect to an outlying commercial area or business park.

In most public transit systems, daily operations are managed by an operations center, with dispatcher personnel managing procedures for the mitigation of small disturbances and larger outages. While the number and intensity of the smaller disturbances might increase from the impacts of climate change, e.g., changing precipitation patterns, many will originate from everyday incidents, like street segments blocked by accidents, or failing transit vehicle doors. Larger outages might result from extreme weather events like pluvial or fluvial flooding, high storms, or excessive heat waves – or from human-caused events like protests or terrorist attacks. In case of any of these events, transit operators have a number of remedies at their disposal to keep services running as long as possible, and to restore them as soon as possible. These include the authority to short-turn or cancel trips, to re-route vehicles, and to deploy extra vehicles.

1.2 Urban Transit Resilience

In the urban transit context, two different understandings of ‘resilience’ are relevant: engineering (or ‘narrow’) resilience and multi-equilibria resilience [13][14][15]. Engineering resilience aims at stability and control, i.e., to withstand shocks and to return to the stable pre-disaster state as fast as possible (‘bouncing back’, see e.g., [16]). Subsequently, the concept of engineering resilience is static and does not take the need for flexibility and adaptation into account. Multi-equilibria resilience [14] on the other hand acknowledges that a disturbed system might not always return to the same stable pre-disaster state and aims at adapting the system to better cope with the disaster (‘bouncing forward’).

For urban transit systems to withstand different types of disasters, transit operators need to design schedules and networks with both resilience concepts in mind. While engineering resilience is useful for mitigating small to medium disturbances that inevitably happen during an operational day (e.g., passengers holding open doors for other passengers), multi-equilibria resilience becomes relevant when addressing medium to large disturbances that might require extensive (temporary) modifications of schedules and vehicle routes. Engineering resilience is usually addressed as part of the medium- to long-term planning (e.g. by designing schedules with high regularity of departure times [17]), multi-equilibria resilience can additionally be addressed in the short- to medium-term planning (e.g. by rerouting vehicles or purposely delaying departure times to keep transfer connections between different transit modes).

Considering accelerating climate change, the associated increase in frequency and intensity of natural disasters, and the subsequent increase in impacts to (urban transit) infrastructure [18][19], it becomes paramount to design new schedules and networks in a resilient and sustainable way, and support operators of existing networks in adapting their services to be more resilient and sustainable.

Simulation models can support this process by enabling decision-makers to assess the implications of alternative decisions faster.

2 Modelling Urban Transit

To represent the described entities and behavior that constitute urban light-rail and bus networks, a bi-modal simulation model based on the event-based approach

[20] is being designed. Thoughts on its design and mechanics are shared in the following sections, highlighting partial models representing physical network components, the logical network, vehicle behavior, operational management, and the necessary randomization. All of these partial models are based on the bus transit model described by Lückerrath and Ullrich [12].

2.1 Physical Network Model

The basis of the model is the representation of the physical transit network as a directed graph. Stops, connections and segments are modeled as nodes of this graph, with their neighborhood relations modeled as edges. Each node has a geographic position, identifying attributes, and a maximum vehicle capacity.

Stops are nodes in the model graph where transit vehicle entities – i.e., busses and light rail cars – stop for boarding and disembarking processes. They always belong to exactly one station and have time-of-day and location-specific stopping times. Different capacity or spatial dimensions of stops are modelled by a maximum number of vehicle entities they can service simultaneously.

Stations group together geographically related stops and give them a uniform name.

Connections are directed paths in the model graph that link two stops. They have a specific length as well as time of day and location specific average travel times. In addition, they are assigned a planned travel time by the timetable. Depending on the transit system to be modeled and the level of detail of the available data, connection nodes also manage model components of the segments, switches and signals belonging to their connection.

Segments represent subsections of connections, representing road or rail segments between two road junctions or between two switches of a rail transport system. Consequently, their corresponding model components have a specific length, a scheduled travel time and manage empirical data on their average travel time. In addition, they have an allowed maximum traversal speed, which can be used, e.g., for microscopic simulation of driving behavior.

To represent the driving behavior of different traffic modes, the model distinguishes between two types of segment nodes: roads and tracks. Road nodes are segment nodes that are used by entities of individual traffic, have an unrestricted vehicle capacity, and do not en-

force a fixed vehicle sequence. Without (detailed) information about lanes, it is assumed that there is sufficient space for overtaking maneuvers on each road node, i.e., travel times of individual vehicle entities can be calculated without considering other entities traveling on the node.

In contrast to road nodes, track nodes are used exclusively by rail vehicle entities and enforce both compliance with a maximum vehicle capacity as well as a fixed vehicle sequence. The latter prevents inadmissible overtaking maneuvers between vehicle entities traveling on the same track node and is realized via the travel time calculation (see Section 2.5): If available, the entity traveling directly ahead is always considered to determine the travel time of a vehicle entity newly arriving on a track node. The calculated simulation time at which the new vehicle entity arrives at the end of the track node can never be earlier than that of the entity directly in front. Without possibilities for overtaking maneuvers, the formation of backlogs - even across neighboring nodes - is considered in the model using the vehicle capacity of nodes.

Switches are locations in rail-based transit systems where track crossings take place without interrupting the journey, i.e. they have a unique geographical position and are related to at least three tracks – at least one each incoming and outgoing. They are modeled as transfer points without spatial extension and are traversed in zero time. Switches can merge several tracks and must be activated to target the correct incoming/outgoing track before an entity can cross them. This is represented in the model by vehicles reserving switches before crossing them and releasing them after a successful transfer.

Signals represent traffic lights of road traffic as well as light signal systems of rail traffic. They usually form a signal group with other signals and have attributes such as switching time or signal status (e.g. green, yellow, red).

Both switches and signals are modelled as additional information layer and not as nodes of the model graph. They can only be found at start or end positions of segment nodes in the model and can be associated with the corresponding nodes based on these positions.

2.2 Logical Network Model

In addition to the physical network components presented so far, logical components such as lines, trips and

timetables have to also be considered to model public transit.

Lines consist of an ordered set of stops, which specifies the route to be followed during regular operation. In the simulation model, this is represented by a reference to a set of corresponding nodes of the model graph. To avoid time-consuming dynamic path finding during the simulation run, lines are additionally supplemented by an ordered set of connection nodes. Furthermore, each line can be assigned to a specific transit mode (e.g., bus or train) and may additionally only be served by vehicle types permitted for it. E.g., a low-floor train may not serve a line whose stops are designed for high-floor trains.

Trips combine ordered sets of stops and connections with a start time and are differentiated into service trips and deadheads. In the model, trips manage references to sets of stop and connection nodes, similar to lines. Service trips additionally refer to the line they serve. Deadheads do not follow a predefined route and therefore do not refer to a line in the model.

2.3 Vehicles

Vehicles are represented as transient entities [20] that encapsulate a significant portion of the event-based simulation logic and move across the model graph during a simulation run. Each vehicle entity has a reference to the trip it is currently serving, i.e., at each simulation time it only has access to the information that is directly relevant for its current activity. All additional information, e.g., about the timetable and the vehicle fleet, is administered by specialized management modules (see Section 2.4).

In the model, vehicles are classified according to their transit mode, their vehicle type, and their individual vehicle characteristics. While transit mode discerns light-rail and bus vehicles, the vehicle type is used for a more detailed subdivision. For example, various types of Vossloh Kiepe GmbH vehicles are in use in the Cologne light rail network, including low-floor vehicles of type K4000 [21] and K4500 [22] and high-floor vehicles of type K5000 [23]. The most detailed classification is based on individual vehicle characteristics. They encapsulate attributes such as passenger capacity, vehicle length, maximum speed, minimum stopping time or boarding rate.

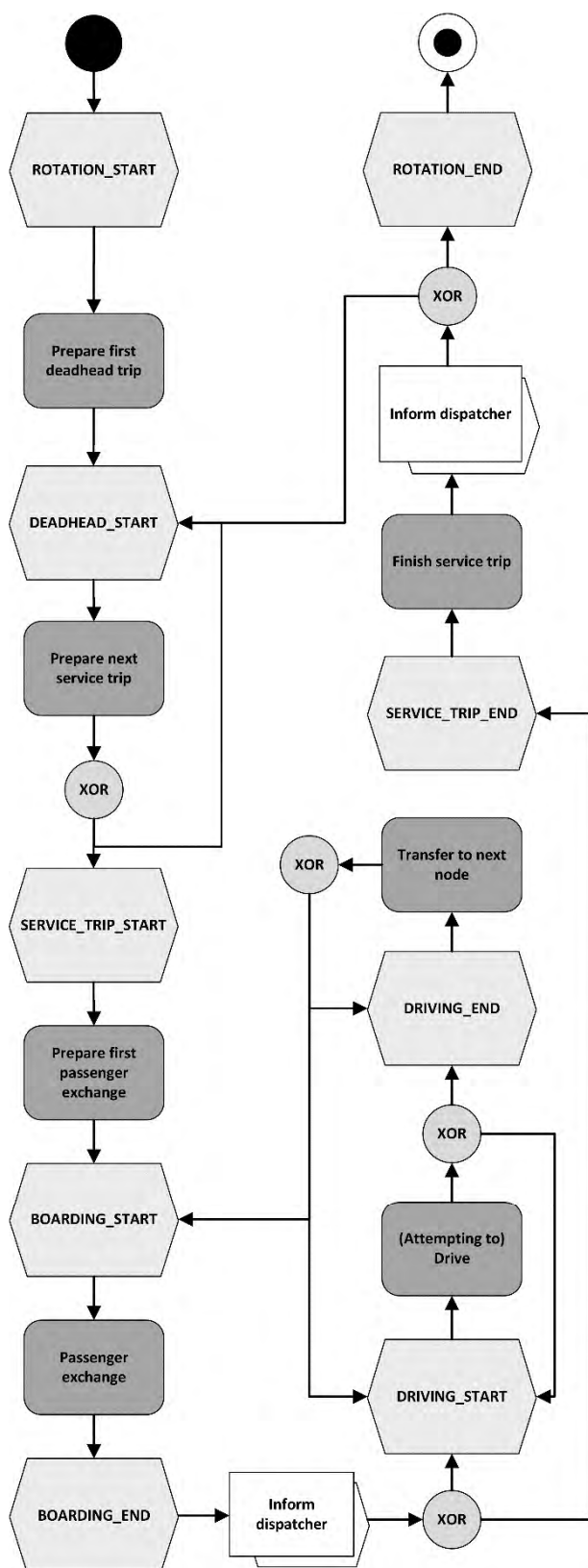


Figure 1: Simulation event types for the light rail and bus vehicles.

Nine simulation event types represent the behavior of bus and light rail vehicles (see Table 1). For a detailed description of the bus-related simulation events see [12]. Figure 1 shows the relationships between the individual event and activity types for light-rail vehicles, based on the associated event process chains. For a detailed discussion of the bus model behavior, see [12].

Simulation event type
ROTATION_START
ROTATION_END
DEADHEAD_TRIP_START
SERVICE_TRIP_START
TRIP_END
BOARDING_START
BOARDING_END
DRIVING_START
DRIVING_END

Table 1: Simulation event types for the light rail and bus vehicles.

2.4 Operational Management

The model components presented so far are sufficient for the representation of elementary functions of public transit systems, but they neglect all higher-level management activities that contribute to the functioning and resilience of transit systems. To allow for management on a higher level than individual trips, the timetable must be supplemented by a rotation schedule, which combines trips into groups (so called rotations) [24] that can be executed by individual vehicles within an operating day. These and other management activities are encapsulated in three management modules: the fleet manager, the line manager, and the dispatcher. Thus, changes to the modeling of individual administrative activities do not affect the modeling of other areas of the simulation model. Work in progress on these modules has been reported in [25].

Fleet manager

The fleet manager administrates the vehicle fleet and allows other components of the simulation model to access the vehicle fleet via defined interfaces, manages which vehicles are currently in use, and is responsible for generating and managing the initial rotation sched-

ule. If no rotation schedule is specified by the user, the fleet manager uses a rotation schedule generator to create an (artificial) rotation schedule.

Line manager

The line manager administrates the lines served as part of a timetable and associates their outward and return directions with each other. In addition, it provides uniform interfaces for accessing individual lines as well as sets of lines. This allows, for example, access to all lines serving a specific stop or a specific connection.

Dispatcher

The dispatcher is the most important and comprehensive management module and can be understood as a model of the operator's decision processes. It holds all the data required for the operational process, such as the current timetables and rotation schedules at a specific point in time. In addition to managing regular operations, the dispatcher also includes the simulation logic required for traffic management. Four different event types address the module's behavior during regular operation (see Table 2).

Simulation event type
OPERATIONAL_DAY_START
OPERATIONAL_DAY_END
BOARDING_END
SERVICE_TRIP_END

Table 2: Simulation event types concerning the dispatcher module.

The event type **OPERATIONAL_DAY_START** models the start of the operating day. As a result of this event type, the dispatcher assigns to the vehicle entities of the vehicle fleet, based on the rotation schedule, the first trip to be served by them. A subsequent event of the type **ROTATION_START** is sent to each assigned vehicle entity.

The end of the operating day is modeled by the event type **OPERATIONAL_DAY_END**. It signals that all service trips have been performed and all vehicle entities have returned to the depot.

Vehicle entities send events of the type **BOARDING_END**, which signal the end of the pure boarding process, to the dispatcher during the simulated operating day. The dispatcher then makes further decisions on

traffic management measures based on this information. For this purpose, the dispatcher can resort to different strategies (see [12] and [25] for detailed descriptions of different strategies). For determining the departure time of a vehicle during regular operations, a location-based departure strategy is employed. Under this strategy, selected stops are defined as control stops at which vehicles always have to wait until their planned departure time, as defined by the timetable, has been reached (e.g., to allow transfers between bus and light rail systems). At all other stops of the network vehicles always depart as soon as the boarding process has been completed, regardless of whether the planned departure time has already been reached or not. If the dispatcher receives an event of the type **BOARDING_END**, it checks whether traffic management measures are to be applied or not. Depending on the result of this check, the waiting time to be added to the entry/exit time is determined. This waiting time is communicated to the affected vehicle entity by sending it a subsequent event corresponding to the end of the waiting time. This subsequent event can be either of the type **SERVICE_TRIP_END** or **DRIVING_START**. The former is the case when the vehicle entity is at the last stop of its current trip. The latter is sent to tell the entity to move to the next node specified in the line route. In addition to traffic management used under 'normal operating conditions', the dispatcher also contains an arsenal of 'emergency traffic management strategies' (as described in [25]), e.g. dynamic rerouting of vehicles in case of blocked segments, shortturning of trips in case of high delay, or temporary splitting of routes.

The last event type relevant for regular operation is **SERVICE_TRIP_END**. It represents the end of a service trip and the subsequent signaling of the control center. As a result of this event, the dispatcher assigns the next service trip to the vehicle entity according to the current schedule and initiates the previously required deadhead. It is ensured that the minimum turnaround time specified by the user is observed between the end of one service trip and the start of the next one. If the finished service trip was the last planned trip of the vehicle entity for the simulated operational day, the dispatcher instructs it to end its rotation. If all trips to be performed on this operating day are completed at the end of the service trip, the dispatcher ends the operating day by scheduling an event of type **OPERATIONAL_DAY_END**.

2.5 Randomization

Two randomized elements are part of the proposed model: the vehicle's traversal time for connections, and the passenger exchange times at stops. Both are directly adapted from [12] with only slight adaptation.

A lognormal distribution is assumed for the **traversal times** for a connection c [26]. Lacking detailed data, the parameters of this distribution, i.e. expectancy value and standard deviation, have to be approximated from the planned traversal times $t_p(c)$. These traversal times usually comprise the planned driving time $t_d(c)$ and the planned passenger exchange time $t_b(c)$, which in turn are comprised of average observed driving/passenger exchange times, standard deviations, and unknown terms (see Equation 1).

$$\begin{aligned} t_p(c) &= t_d(c) + t_b(c) \\ &= (\mu_c^d + \sigma_c^d + \epsilon_c^d) \\ &\quad + (\mu_c^b + \sigma_c^b + \epsilon_c^b) \end{aligned} \quad (1)$$

It can be assumed that the planned traversal time $t_p(c)$ is greater than the average observed traversal time μ_c^d to avoid systematic delays. The average traversal time can then be roughly approximated as follows:

$$\hat{\mu}_c^d = t_p(c) * \gamma, \forall c \in C, 0 < \gamma < 1 \quad (2)$$

The ratio γ has to be determined by the user. The standard deviation σ_c^d can be approximated in the same way. It can be assumed that the standard deviation is only a small fraction of the planned traversal time. This yields Equation 3.

$$\hat{\sigma}_c^d = t_p(c) * \eta, \forall c \in C, 0 < \eta < 1, \eta \ll \gamma \quad (3)$$

The **passenger exchange times** for busses and light-rail vehicles can be modeled following the method first proposed in [27]. This method is suitable for high frequency transit systems like urban light-rail and bus transit, where it can be assumed that passengers arrive randomly during the inter-arrival time of two successive vehicles, instead of arriving in bulk shortly before the planned departure time. Furthermore, the method facilitates the modeling of vehicle bunching, i.e. the effect that two vehicles form an undesired platoon because the vehicle in front takes on more passengers than planned and subsequently suffers longer passenger exchange times, while the rear vehicle takes on fewer passengers as planned and thus catches up to the vehicle in front.

If the number $N_{b,s}$ of passengers entering a vehicle b at a stop s , and the average time I_b a passenger takes to enter vehicle b are known, the passenger exchange time

$T_{b,s}$ can be determined as follows:

$$T_{b,s} = T_b^{min} + I_b + N_{b,s} \quad (4)$$

Here T_b^{min} describes a vehicle specific minimum time, e.g. for opening and closing the vehicle's doors. If the passenger arrival rate a_s at stop s is known, $N_{b,s}$ can be modeled dependent on the basic interval $T_{L(b)}$ of line $L(b)$ currently served by vehicle b . With $N_{b,s} = T_{L(b)} * a_s$ the passenger exchange time can then be approximated as shown in Equation 5.

$$T_{b,s} = T_b^{min} + I_b + T_{L(b)} * a_s \quad (5)$$

If instead of the basic interval between vehicles of the same line, simulated headways between successive vehicles servicing the same stop are used, the model becomes dynamic and thus suitable for a simulation model. If $t_{dep}(b-1, s)$ describes the time a vehicle b 's predecessor has serviced the stop, the passenger exchange time $T_{b,s}(t_{sim})$ can be determined as in shown in Equation 6.

$$\begin{aligned} T_{b,s}(t_{sim}) &= \begin{cases} T_b^{min}, & b \text{ is first vehicle at } s \\ T_b^{min} + (t_{sim} - t_{dep}(b-1, s)) * a_s * I_b, & \text{else} \end{cases} \end{aligned} \quad (6)$$

3 Further Research

This paper introduced steps towards increasing the resilience of public transit infrastructure with a focus on designing and developing a fast bi-modal simulation model covering both light-rail and (express and community) bus transit. For that purpose, the components of public transit infrastructure systems were discussed, followed by a short introduction of resilience concepts and frameworks. Then, thoughts on the development of a bi-modal simulation model were shared, focusing on modelling physical components, the logical network, vehicle behavior, operational decisions, and the necessary randomization.

To complete the project, further research steps are necessary: As a next step, the simulation model and application have to be completed, tested, and validated. An additional goal for that step is to evaluate the model's execution speed – to be useful in real-world applications, development has to focus on fast execution of individual simulation runs. Then, the validated model will be applied to evaluate what combination of disaster risk management and resilience-building strategies

shows most potential to help increasing the resilience of urban transit systems against extreme weather events resulting from climate change as well as other disasters.

Acknowledgements

This paper has been partially supported by the framework of the European project ARCH – Advancing Resilience of historic areas against Climate-related and other Hazards. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no. 820,999. 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 REA nor the European Commission are responsible for any use that may be made of the information contained therein.

References

- [1] Vuchic, V. R. *Urban Transit - Systems and Technology*. Verlag John Wiley & Sons, Hoboken, New Jersey 2007.
- [2] Olfert, A., et al. 2021. *Sustainability and resilience - a practical approach to assessing sustainability of infrastructure in the context of climate change*. G. Hutter, M. Neubert and R. Ortlepp. Building resilience to natural hazards in the context of climate change - Knowledge integration, implementation, and learning. Springer, 2021.
- [3] Behrisch, M., Erdmann, J., Krajzewicz, D. Adding intermodality to the microscopic simulation package SUMO. In: Al-Akaidi, M., editor. MESM’ 2010 – GAMEON-ARABIA’2010. 11th Middle Eastern Simulation Multiconference; 2010 Dec; Alexandria. Alexandria: eurosis. 59-66.
- [4] Kendziorra, A., Weber, M. Extensions for logistics and public transport in SUMO. In: Behrisch, M., Weber, M., editors. SUMO 2015 – Intermodal Simulation for Intermodal Transport. 3rd SUMO User Conference; 2015 May. Berlin: Deutsches Zentrum für Luft- und Raumfahrt, Institut für Verkehrssystemtechnik. 83-90.
- [5] Toledo, T., Cats, O., Burghout, W., Koutsopoulos, H. Mesoscopic simulation for transit operations. *Transport. Res. C-Emer.* 2010; 18(6): 896-908.
- [6] Suzumura, T., Kanezashi, H. Multi-modal traffic simulation platform on parallel and distributed systems. In: Tolk, A., Diallo, S., Ryzhow, I., Yilmaz, L., Buckley, S., Miller, J. Proceedings of the 2014 Winter Simulation Conference. *Winter Simulation Conference*; 2014 Dec; Savannah. Piscataway, NJ, USA: IEEE Press. 769-780.
- [7] Suzumura, T., McArdle, G., Kanezashi, H. A high-performance multi-modal traffic simulation platform and its case study with the Dublin city. In: Yilmaz, L., Chan, W., Moon, I., Roeder, T., Macal, C., Rossetti, M., editors. Proceedings of the 2015 Winter Simulation Conference. *Winter Simulation Conference*; 2015 Dec; Huntington Beach. Piscataway, NJ, USA: IEEE Press. 767-778.
- [8] Burghout, W. *Hybrid microscopic-mesoscopic traffic simulation* [dissertation]. Department of Infrastructure, Royal Institute of Technology, Sweden. University of Stockholm, 2004.
- [9] Ullrich, O., Proff, I., Lückerrath, D., Kuckertz, P., Speckenmeyer, E. Agent-based modeling and simulation of individual traffic as an environment for bus schedule simulation. In: Busch, F., Spangler, M., editors. ITS for Connected Mobility. *mobilitUM*; 2014; Munich. Munich: Schriftenreihe des Lehrstuhls für Verkehrstechnik der Technischen Universität München. 89-98.
- [10] Kastner, K., Keber, R., Pau, P., Samal, M. Real-Time Traffic Conditions with SUMO for ITS Austria. In: Behrisch, M., Knocke, M., editors. 1st SUMO User Conference 2013. *1st SUMO User Conference*; 2013 May, Berlin. Berlin: Deutsches Zentrum für Luft- und Raumfahrt, Institut für Verkehrssystemtechnik. 43-53.
- [11] Kastner, K., Pau, P. Experiences with SUMO in a Real-Life Traffic Monitoring System. In: Behrisch, M., Weber, M., editors. SUMO 2015 – Intermodal Simulation for Intermodal Transport. 3rd SUMO User Conference; 2015 May. Berlin: Deutsches Zentrum für Luft- und Raumfahrt, Institut für Verkehrssystemtechnik. 1-10.
- [12] Lückerrath, D., Rishe, N., Speckenmeyer, E., Ullrich, O. A Mesoscopic Bus Transit Simulation Model Based on Scarce Data. *Simulation Notes Europe (SNE)* 2018; 28(1): 1-10.
- [13] Folke D. Resilience: The emergence of a perspective for social-ecological systems analyses. *Global Environmental Change* 2006; 16(3): 253-267.
- [14] Seelinger, L., Turok, I. Towards Sustainable Cities: Extending Resilience with Insights from Vulnerability and Transition Theory. *Sustainability* 2013, 5(5): 2108-2128.
- [15] Doorn, N. Resilience indicators: opportunities for including distributive justice concerns in disaster management. *Journal of Risk Research* 2017; 20(6): 711-731.
- [16] Holling, C. Engineering resilience versus ecological resilience. In: *Engineering Within Ecological Constraints*, P. Schulze, Ed., Washington D.C., National Academy Press, 1996, pp. 31-44.
- [17] Ullrich, O., Lückerrath, D., Speckenmeyer, E. Do regular timetables help to reduce delays in tram networks? – It depends! *Public Transp* 2016, 8: 39-56.
- [18] Forzieri, G., et al. Escalating impacts of climate extremes in critical infrastructures in Europe. *Global Environmental Change* 2018, 48.
- [19] Forzieri, G., et al. *Resilience of large investments and critical infrastructure in Europe to climate change*. JRC Report, 2016.
- [20] Ullrich, O., Lückerrath, D. An Introduction to Discrete-

Event Modeling and Simulation. *Simulation Notes Europe (SNE)* 2017; 27(1): 9-16.

- [21] Vossloh Kiepe GmbH. *Elektrische Ausrüstung der Niederflur-Stadtbahnwagen K4000 der Kölner Verkehrs-Betriebe AG 2003*. Druckschrift Nr. 00KV7DE.
- [22] Vossloh Kiepe GmbH. *Elektrische Ausrüstung des Niederflur-Stadtbahnwagens K4500 für die Kölner Verkehrs-Betriebe AG 2003*. Druckschrift Nr. 00KN2DE.
- [23] Vossloh Kiepe GmbH. *Elektrische Ausrüstung der Hochflur-Stadtbahnwagen K5000 der Kölner Verkehrs-Betriebe AG 2003*, Druckschrift Nr. 00KB5DE.
- [24] Lückerrath, D., Ullrich, O., Kupicha, A., Speckenmeyer, E. Multi-depot multi-vehicle-type vehicle scheduling for Cologne's tram network. In: Proc. *ASIM-Workshop STS/GMMS 2014*, ARGESIM Report 42, ASIM-Mitteilung AM 149, ARGESIM/ASIM Pub., TU Vienna/Austria, 191-197.
- [25] Lückerrath, D., Bogen, M., Rome, E., Sojeva, B., Ullrich, O., Worst, R., Xie, J. Strategies to Mitigate the Impacts of Climate Change Related Events on Public Transit Networks. In: Proc. *24th Symposium Simulationstechnik (ASIM 2018)*, Hamburg, Germany, October 03-04, 2018, 175-182.
- [26] Andersson, P., Hermansson, A., Tengvald, E., Scalia-Tomba, G. Analysis and simulation of an urban bus route. *Transport. Res. A-Pol.*, 1979, 13(6): 439-466.
- [27] Chapman, R., Michel, J. Modelling the Tendency of Buses to Form Pairs. *Transport. Sci.*, 1978, 12(2): 165-175.

Diskrete Systeme

Discrete Systems

A Simulator for NSA-DEVS in Matlab (L)

D. Jammer, P. Junglas, T. Pawletta, S. Pawletta

Adapting to Change of Model Transitions in Proxel Based Simulation of CHnMMs (L)

D. Bodnár, C. Krull

A Hybrid User Model for Virtual Stochastic Sensors (L)

C. Krull

Analysis of Train Delays using Bayesian Networks (K)

V. Barzal, M. Rößler, M. Wastian, F. Breiteneker, N. Popper

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

D. Pasterk, A. Körner

Die Kurzbeiträge (K) sind zu finden im **ARGESIM Report 19,**
Proceedings Kurzbeiträge ASIM SST 2022

ISBN ebook 978-3-901608-96-4, DOI: 10.11128/arep.19, ASIM Mitteilung AM 179

A Simulator for NSA-DEVS in Matlab

David Jammer^{1,2*}, Peter Junglas², Thorsten Pawletta¹, Sven Pawletta¹

¹Research Group Computational Engineering and Automation, University of Applied Sciences Wismar, Philipp-Müller-Straße 14, 23966 Wismar, Germany;

*david.jammer@cea-wismar.de

²PHWT-Institut, PHWT Vechta/Diepholz, Am Campus 2, 49356 Diepholz, Germany;

Abstract. The PDEVS formalism is widely used for the description and analysis of discrete event systems. But PDEVS has some drawbacks in modeling Mealy behavior. A revised version (RPDEVS) has been invented to resolve them, but it has problems of its own, mainly because its complicated simulator structure. The recently proposed NSA-DEVS scheme tries to unite the advantages of both formalisms by using infinitesimal time intervals.

To further substantiate this claim we describe an abstract simulator for NSA-DEVS, implement it in Matlab and simulate a simple queue-server system. This shows that NSA-DEVS combines the Mealy-like model description of RPDEVS with the simple simulator structure of PDEVS, making it a promising approach to implement an improved modeling and simulation system.

Introduction

The DEVS formalism [1] and its most popular variant PDEVS [2] are a well established approach for the modeling and analysis of discrete event systems. Although a few modeling and simulation tools exist that are using PDEVS [3], the usual formalism does not directly support the implementation of component-based simulation programs.

A few formal problems can be fixed by simple variations of the basic formalism [4], a well-known example being the introduction of input and output ports. A more serious flaw has been found by Preyser et al. [5]: Due to the Moore-like structure of PDEVS the combination of Mealy-type components can sometimes be difficult to implement. Using the standard workaround of transitory states (i. e. states with transition times of 0) the behaviour of a complete system can always be modeled with PDEVS. But the description of the underlying components as individual (“atomic”) blocks can lead to an ordering of concurrent events in the complete system, which does not agree with the intended behaviour.

Therefore Preyser et al. have introduced a Revised PDEVS (RPDEVS) formalism [6] that uses a Mealy-like scheme directly – without the introduction of transitory states – and allows for direct modeling of Mealy-like components, which behave correctly in the context

of a larger system. To make this possible they had to define an abstract simulator for RPDEVS [7] that uses a complicated scheme of internal iterations.

However, for systems with a complex causal structure of concurrent events this iteration leads to problems, as has been shown in [8] using the example of a queue-server system. To solve these problems and to bring the modeling process closer to the underlying ideas of the modeler, the NSA-DEVS (“Non-Standard Analysis DEVS”) formalism has been introduced in [8], which is a variant of RPDEVS and uses concepts of non-standard analysis [9]. Another approach has been suggested to cope with the ordering of concurrent events by augmenting the real time line [10], but it doesn’t address the Mealy-related problems that RPDEVS and NSA-DEVS try to solve.

The objective of the work presented here is to further investigate the soundness and usefulness of the NSA-DEVS formalism by defining a proper abstract simulator. To this end we first review the model and simulator specifications in PDEVS, then shortly introduce the hyperreal numbers and define the NSA-DEVS modeling formalism. Next we describe the abstract NSA-DEVS simulator and highlight some crucial points of its implementation in Matlab. Finally we implement the queue-server system from [8] and demonstrate that it works as intended.

1 Short review of the PDEVS formalism

The Discrete Event System Specification (DEVS) formalisms are divided into model specification and abstract simulator which are explained in more detail in the following for Parallel DEVS (PDEVS). The model specification differentiates between atomic and coupled models, which together form a hierarchical structure. In the following, we introduce a simplified model specification for PDEVS that uses ports instead of input

bags [11, p.108]. The model specification of an atomic model is an 8-tupel $\langle X, S, Y, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta \rangle$ with

X	set of input ports and values,
Y	set of output ports and values,
S	set of sequential states,
$\delta_{int} : S \rightarrow S$	internal transition function,
$\delta_{ext} : Q \times X^+ \rightarrow S$	external transition function,
$\delta_{con} : S \times X^+ \rightarrow S$	confluent transition function,
$\lambda : S \rightarrow Y^+$	output function,
$ta : S \rightarrow \mathbb{R}^{\geq 0} \cup \{\infty\}$	time advance function.

Here $Q = \{(s, e) | s \in S, 0 \leq e < ta(s)\}$ and e is the elapsed time since the last transition. The input and output sets are defined as

$$\begin{aligned} X &= \{(p, v) | p \in P_{in}, v \in X_p\} \\ Y &= \{(p, v) | p \in P_{out}, v \in Y_p\} \end{aligned}$$

where P_{in} and P_{out} are the sets of input and output names and X_p and Y_p are the sets of possible values at input or output port p . Since inputs can arrive simultaneously at different ports, one needs the set

$$X^+ := \{ \{(p_1, v_1), \dots, (p_n, v_n)\} | n \in \mathbb{N}_0, p_i \in P_{in}, p_i \neq p_j \text{ for } i \neq j, v_i \in X_{p_i} \}$$

and similarly Y^+ for simultaneous outputs at several ports. Unlike in [11] simultaneous inputs at the same port are not allowed here. This makes the formulation of external transition functions easier, but prohibits the direct connection of several output ports to one input port. This is not a real limitation though, since one can insert an appropriate atomic component (*multiplexer*) for this purpose.

The formal specification of coupled models has changed several times in the development of DEVS. For practical purposes, the following specification is used in this article: $N = \langle X, Y, D, \{M_d\}, EIC, EOC, IC \rangle$

X	set of input ports and values,
Y	set of output ports and values,
D	set of component names,
$\{M_d\}$	set of dynamic systems with $d \in D$,
EIC	set of external input couplings,
EOC	set of external output couplings,
IC	set of internal couplings.

Furthermore, the PDEVS formalism defines an abstract simulator, which describes the execution of a specified model [11, p.197]. It consists of the modules

root coordinator, *coordinator* and *simulator*. They are combined in a hierarchical structure, which is shown in Fig. 1 for a simple example. The abstract simulator al-

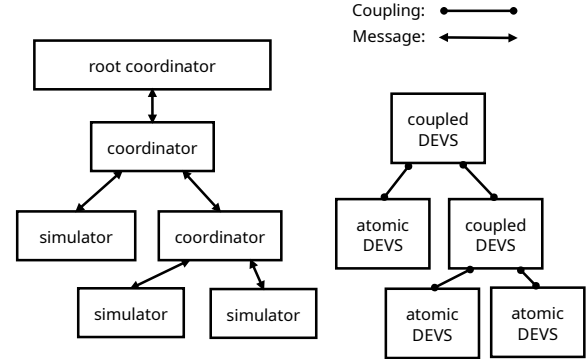


Figure 1: Hierarchical and distributed concept of the abstract simulator.

ways consists of exactly one root coordinator as the top-most instance. This is always followed by a coordinator that is attached to the uppermost coupled model. In addition, a coordinator is assigned to each coupled model of the underlying layers of the hierarchical structure, whereas a simulator is assigned to each atomic model. The coordinators and simulators form a tree structure that parallels the model structure (cf. Fig. 1), where the leaves on the left side are the simulators and on the right (model) side the atomic components.

The simulation is organized with a message concept. Messages are exchanged between root coordinator, coordinators and simulators, all downwards messages contain the current simulation time t . The following message types are used:

- i-message: downwards for initialization,
- *-message: downwards to initiate internal events,
- y-message: upwards to distribute outputs,
- x-message: downwards to trigger events,
- d-message: upwards to return information.

This terminology follows [7], in [11] the d-messages (“done”) are only implicitly mentioned in the pseudocode.

The simulation starts with an i-message that is sent by the root coordinator to the topmost coordinator and distributed downwards. Each simulator initialises its atomic model and returns the time of its next internal

event to its parent coordinator. All coordinators collect the times of their children and report the smallest value upwards, until the root coordinator is reached, which stores the received value as the current simulation time.

Next the root-coordinator sends a *-message, which is forwarded according to the hierarchical structure to all simulators that are *imminent*, i. e. the time of their next event is equal to the current simulation time. Each of these simulators executes the λ -function of its atomic model and sends the output to its coordinator via a y-message. Using the set IC of its coupled model the coordinator distributes the outputs to the appropriate child simulators and coordinators via x-messages and sends additional empty x-messages to the imminent children. Furthermore the coordinator collects the external outputs according to the set EOC of its coupled model and sends them upwards via a y-message.

On receiving an x-message a simulator executes one of the three transition functions of its atomic model, depending on the event type. An empty x-message means an *internal* event, which causes the execution of the δ_{int} function. A non-empty x-message represents an *external* or *confluent* event. If the atomic model is not imminent, δ_{ext} is executed, otherwise δ_{conf} . After the execution of a transition function, the time advance function ta is called to compute the time of the next internal event, which is sent upwards. A coordinator, that receives an x-message, forwards it to its active children, i. e. those that get a new input or are imminent.

As a result, the root coordinator receives the next event time, updates the current simulation time and sends a new *-message. This procedure is repeated until the root coordinator detects a termination condition. A complete description of the abstract simulator using pseudocode is given in [11, p.350-353].

2 The NSA-DEVS modeling formalism

The basic idea of the NSA-DEVS formalism is to start with the RPDEVs description, to add infinitesimal delays at the inputs of all components and to replace transitory states by states with infinitesimal transition times. This has two immediate advantages: Firstly, the complex iteration, that is necessary in the RPDEVs simulator to handle the transport of events through networks of Mealy-type components, is obsolete. Secondly, one can easily define the ordering of concurrent events by using appropriate delay times.

The introduction of infinitesimals to represent small real delays avoids an abundance of unknown additional parameters. Instead one can mainly use a default value ε , using different values only for special needs. Furthermore, the simulator handles the infinitesimal events mainly internally, so that from the user perspective, correct Mealy behaviour can be achieved.

For a precise mathematical description of finite or infinitesimal time delays we use the totally ordered field of hyperreal numbers ${}^*\mathbb{R}$. This is an extension of the real numbers including an infinitesimal $\varepsilon > 0$, which is smaller than any positive real number. Every finite hyperreal a is infinitely close to exactly one real number, called the standard part of a and denoted by $st(a)$. The construction of ${}^*\mathbb{R}$ relies on advanced results from set theory and logic, but its use is rather straightforward. Exact definitions, theorems and proofs can be found in [9]. For the implementation of a simulator, numbers of the form $a + b\varepsilon$ with $a, b \in \mathbb{R}$ are sufficient, they can be stored as a pair of floating point numbers. The standard part then simply is $st(a + b\varepsilon) = a$. To represent *passive states*, i. e. states with an infinite transition time, the hyperreal number $\omega := 1/\varepsilon$, represented by the floating point value “infinity”, can be used. ω is *unlimited*, i. e. it is larger than any real number. In the following we are mainly interested in the subset of positive finite hyperreals ${}^*\mathbb{R}_{fin}^{>0}$.

One can now formally define an *atomic NSA-DEVS* as a 7-tuple $\langle X, S, Y, \tau, ta, \delta, \lambda \rangle$ in the following way:

X	set of input ports and values,
S	set of states,
Y	set of output ports and values,
$\tau \in {}^*\mathbb{R}_{fin}^{>0}$	input delay time,
$ta : S \rightarrow {}^*\mathbb{R}_{fin}^{>0} \cup \{\omega\}$	time advance function,
$\delta : Q \times X^+ \rightarrow S$	transition function,
$\lambda : Q \times X^+ \rightarrow Y^+$	output function.

where the sets X, Y are defined as in section 1, but Q is changed slightly to $Q = \{(s, e) | s \in S, 0 \leq e \leq ta(s)\}$.

The main difference to the PDEVs formalism described above is the restriction to only one transition function and the extension of the output function, which is now called at all three kinds of events. This is identical to the RPDEVs definition in [6] and allows for a direct formulation of Mealy-type components. The formal difference to RPDEVs is small: All time values and intervals are now meant as subsets of the hyperreals ${}^*\mathbb{R}$ and ta is always > 0 . But the semantics are slightly different: When an external event, i.e. a set of

inputs $x \in X^+$, occurs at time t , the output function λ is called at $t + \tau$, followed by an immediate call of δ . An internal event, i.e. a state change after a waiting time $ta(s)$, leads to a direct (undelayed) call of λ and δ . A concurrent incidence of a (delayed) external event and an internal event can be detected by both functions directly and doesn't need a special mechanism.

A *coupled NSA-DEVS* is defined just like in RPDEVS and PDEVS, outputs are transported as usual and a coupled component has no additional input delays. For the usual confirmation of closure under coupling, i. e. the formulation of a coupled system as an atomic component, one simply uses the smallest delay of all internal components that are connected to external inputs, and adds additional delays where necessary.

3 The abstract NSA-DEVS simulator

The general concept of the abstract simulator is the same as for PDEVS, it uses the hierarchical structure, the message system and the three modules that have been introduced in section 1. The algorithms of the coordinator and the root coordinator for NSA-DEVS are identical to the PDEVS versions described in [11, p.205] and [11, p.352-353], the only difference are the type of the current simulation time and the event times, which are now hyperreals instead of real numbers.

The basic difference lies in the algorithm of the simulator module: Though it looks similar to the PDEVS simulator in [11, p.351], it implements the NSA-DEVS scheme, which directly supports Mealy-like behaviour using infinitesimal input delays. Its simplified pseudocode description is presented in Listing 1.

Listing 1: NSA-DEVS Simulator.

```

1 properties:
2   parent
3   tl
4   tn
5   model (NSA-DEVS incl.  $\tau$  and
           total state (s,e))
6   y
7   x*
8
9 when receive i-message(i,t) at time t
10  tl = t - e
11  tn = tl + ta(s)
12
13 when receive *-message(*,t) at time t
14  e = t - tl
    
```

```

15  y =  $\lambda(s, e, x^*)$ 
16
17  send y-message(y,t) to parent
    coordinator
18
19 when receive x-message(x,t) at time t
20  if x ==  $\emptyset$ 
21    e = t - tl
22    s =  $\delta(s, e, x^*)$ 
23    x* =  $\emptyset$ 
24    if ta(s) ==  $\omega$ 
25      tn =  $\omega$ 
26    else if st(ta(s)) == 0
27      tn = t + ta(s)
28    else
29      tn = st(t + ta(s))
30    tl = t;
31  else
32    if not (x* ==  $\emptyset$ )
33      add events from x to x*
34    else
35      x* = x
36    tn = t +  $\tau$ 
    
```

Lines 1–7 list the variables used by the simulator. The first five are the same as for PDEVS: the parent coordinator, the times of the last and the next event, the attached model – with the atomic NSA-DEVS structure and its complete state – and the output values. Since the input delay is realized inside the simulator, input values must be stored temporarily, using the variable x^* . In lines 9–11 the i-message is handled, which just computes the times of the last and the next events. The *-message is processed in lines 13–18, where the elapsed time is calculated, the λ function is executed and the y-message is sent to the parent coordinator. In contrast to the PDEVS algorithm λ is now a function of the total state (s,e) and the input value x^* that has been stored before.

The new part - compared to PDEVS - is the way the x-message is processed, which is shown in lines 19–36. It discriminates between an internal event (lines 21–30, $x == \emptyset$) and an external event (lines 32–36). In the latter case it stores all incoming values in x^* and schedules a new internal event at the delayed time $t + \tau$. All internal events are handled by calling the transition function δ and computing the time of the next event using ta in lines 24–29. This calculation deserves special attention: It guarantees that “real” time steps (i. e. non-infinitesimal ones) lead to real valued time values in order to implement a correct timing and to prohibit an accumulation of infinitesimal delays.

To better understand the operation of the abstract simulator, especially how it creates a proper Mealy be-

haviour, we introduce a simple example model N: It consists of a generator G, which outputs a value $t/10$ at times $t = 1, 2, 3 \dots$, a multiplication block M, which multiplies its input by a factor 3, and a terminator component T, which acts as a sink for the incoming values (cf. Fig. 2).

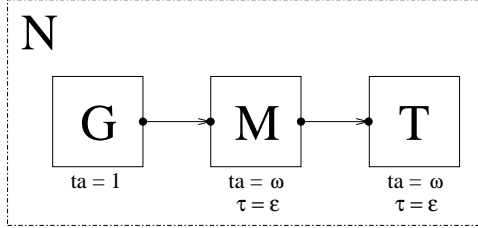


Figure 2: Simple example model.

The messages that are sent between the root coordinator RC, the coordinator C_N of coupled model N and the three simulators S_G , S_M and S_T with the associated models G, M and T are shown as a sequence diagram in Fig. 3. Downwards messages are denoted as *(msg type, current time)*, for the x-messages the input value is added. Upwards messages are shown as *(msg type, result)*.

In the initialisation step at $t = 0$, an i-message is sent and distributed to the simulators, returning the time $t = 1$ of the first internal event to RC. This is followed at $t = 1$ by a *-message sent to the simulator of the only imminent component G, which generates an output event and sends a y-message with its output 0.1 back to C_N . The coordinator now sends an empty x-message to S_G , which returns the time $t = 2$ of its next internal event. Moreover, C_N sends a non-empty x-message to S_M , which stores the value internally and schedules a new internal event according to the input delay time.

The next *-message at $t = 1 + \epsilon$ arrives at the simulator of the imminent component M, which calls its λ function and sends the output value 0.3 as a y-message to its coordinator. C_N now sends an empty x-message to S_M , which calls its δ and ta functions and returns $t = \omega$ to C_N . M is now in a passive state. The rest of the diagram shows how the output value propagates to S_T , which just terminates the incoming events. Since the coordinator stores all future event times of its children, it finally returns $t = 2$ (originally coming from S_G) as the time of the next event, which will repeat the whole cycle.

This example shows precisely, how the two parts of

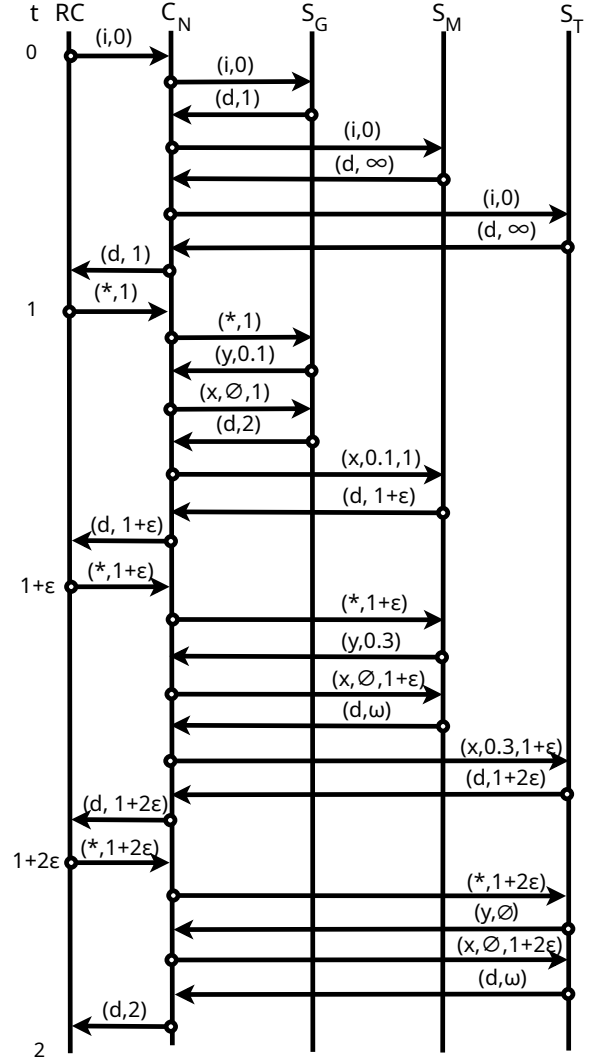


Figure 3: Example of message flow.

the x-message algorithm in the simulator module work together to implement the time delay and the Mealy behaviour of a simple Mealy block such as a multiplication function.

4 Implementation of the NSA-DEVS simulator

The “infinitesimal cloud” of hyperreal numbers around each real number has a complex structure with lots of different layers of smallness, e. g. using ϵ^2 or $\sqrt{\epsilon}$. The purpose of using ${}^*\mathbb{R}$ in the formulation of NSA-DEVS is the possibility to introduce short time intervals with-

out defining their size explicitly, but still being able to order them. Therefore times of the form $a + b\epsilon$ are sufficient here, they are stored as two-element vectors. The implementation of time comparisons and sorting has to be adapted accordingly.

Since the goal of NSA-DEVS is to provide a good basis for the concrete modeling of discrete event systems, an implementation should free the modeler from the tedious task of defining lots of additional infinitesimal parameters. Therefore the concrete simulator contains a variable $\tau_{def} = r\epsilon$ (usually $r = 1$) that is used as a default value of τ for all atomic components. Furthermore the user can still define transitory states with a transition time $ta(s) = 0$, which is replaced automatically by setting $ta(s) = r\epsilon$.

For debugging purposes it would be useful to make the infinitesimal delays explicitly visible. To this end the simulator contains a real (i. e. floating point) parameter μ , which is 0 normally, but can be set to a value larger than zero for debugging. In this case the infinitesimal ϵ is replaced by μ and all times are real values computed as

$$t' = \begin{cases} (t(1), t(2)) & \text{if } \mu = 0, \\ (t(1) + \mu t(2), 0) & \text{if } \mu > 0. \end{cases}$$

In complex models the value of μ has to be chosen carefully: It should be large enough to make the infinitesimal internal processes visible, but small enough to not induce any changes into the behaviour of the model. As a result of this extension, the implementation of the x-message gets more complicated, as can be seen in Listing 2.

Listing 2: Implementation of the x-message algorithm.

```

1 when receive x-message(x,t) at time t
2   if x == ∅
3     e = [t(1) - t1(1), t(2) - t1(2)]
4     s = δ(s, e, x*)
5     x* = ∅
6
7     tb = ta(s)
8     if tb == [0,0]
9       tb = [0, r]
10    if tb(1) == 0
11      if μ == 0
12        tn = [t(1), t(2) + tb(2)]
13      else
14        tn = [t(1) + μ*tb(2), 0]
15    else
16      tn = [t(1) + tb(1), 0]
17    t1 = t;
```

```

18 else
19   if not (x* == ∅)
20     add events from x to x*
21   else
22     x* = x
23   if μ == 0
24     tn = [t(1) + tau(1), t(2) + tau(2)]
25   else
26     tn = [t(1) + tau(1) + μ*tau(2), 0]
```

The special case of passive states in Listing 1 (l. 24f) is done automatically in line 16 due to the handling of the value “infinity” in floating-point arithmetic.

5 Case study: A simple queue-server system

To test the operation of the complete NSA-DEVS formalism – model specification and simulator –, a prototype has been created in Matlab, which is named NSA-DEVSforMATLAB. It contains all features introduced in section 4 and is implemented in an object-oriented way. To show its functionality, the singleserver example from [8] has been chosen as an example for this article; it is shown as a block diagram in Figure 4. The

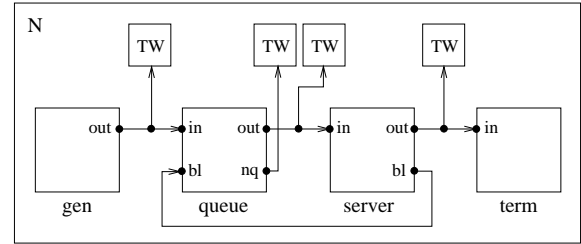


Figure 4: Example model singleserver.

following atomic models are used for the example:

- Generator: produces entities with an interval of one second,
- Queue: infinite queue,
- Server: service time 1.5 s,
- Terminator: terminates the entities,
- ToWorkspace (TW): logging data.

The special feature of this model is that the queue should only send entities to the server when it is not busy. The server announces this information via port blocked (bl), which is sent to port bl of the queue.

For functionality, the input delay of the queue must be greater than the input delay of the server. The input delays of the generator and terminator do not matter. A special role is played by the four ToWorkspace models, which are connected to the output ports *out* and *nq* according to Fig. 4 and store the output values. They too – like every atomic model – have an input delay.

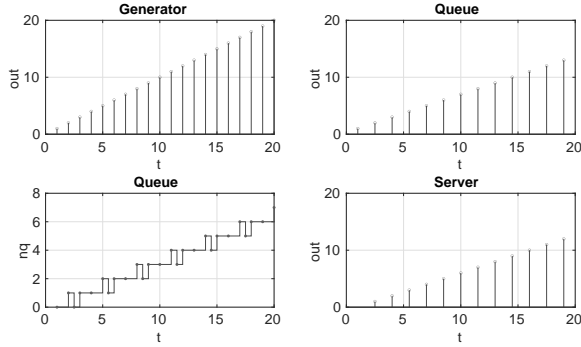


Figure 5: Simulation results with high input delay at ToWorkspace.

For the first simulation run, a large infinitesimal input delay was chosen for the ToWorkspace models. The result can be seen in Figure 5, which displays generator output (top left), queue output (top right), queue load (bottom left) and server output (bottom right).

The high input delay has the effect that output changes, which happen during a series of infinitesimal time intervals, are discarded inside a ToWorkspace block and only the final value before a finite time step is shown. This can be seen, for example, at time 10: The server has finished processing and is idle. Therefore the queue sends an entity to the server. At the same time, the generator also outputs an entity and sends it to the queue. In total, the load of the queue does not change.

However, if one uses a low input delay for the ToWorkspace models, one sees that the queue load at time 10 has the values 3 and 4 simultaneously. This means that the new entity enters the queue first and then an entity is sent to the server. This behavior is shown in Figure 6. One could use the debug mode, i. e. set the parameter μ to a finite value, to dissolve the “spike” at $t=10$ into a small step, thereby clearly showing the internal ordering of the events.

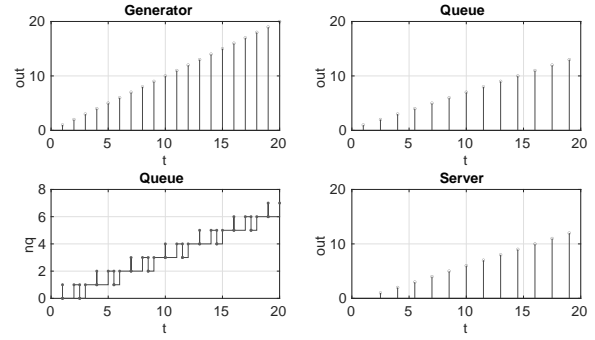


Figure 6: Simulation results with low input delay at ToWorkspace.

6 Conclusion

With the specification of an abstract simulator, which defines the behaviour of a model consisting of atomic and coupled components, the description of the NSA-DEVS formalism is now formally complete. We have shown that NSA-DEVS is able to directly describe Mealy-like models in the same way as RPDEVS, but with a much simpler simulator algorithm similar to the original PDEVS version. In this way NSA-DEVS combines the best of the two preceding formalisms.

Its principle usability has been demonstrated by the implementation of the simulator and a non-trivial example model in Matlab. The notoriously difficult modeling of concurrent events has been substituted by a clear definition of an ordering based on infinitesimal delays. The inclusion of a debugging mode further helps to understand the corresponding difficulties. An interesting side effect is the possibility to easily model systems with finite time delays.

At first sight, the NSA-DEVS approach seems to be very similar to the concept of superdense time [10], where a real time value is augmented by a natural number to order concurrent events. But the much richer structure of $^*\mathbb{R}$ – even of the small part that is used in the implementation – has profound consequences: On the practical side, one can use real infinitesimal delays to squeeze an event between existing ones, without the need to reorder the complete sequence. The conceptual difference, however, is the dynamic structure of NSA-DEVS: The order of concurrent events is defined by the infinitesimal delays in the complete model, which add up in a “realistic” way. While the fixed ordering of superdense time is similar to the *Select* function in Classical DEVS [11, p.104], NSA-DEVS – like PDEVS –

allows concurrent events on the infinitesimal scale and parallelism.

To further examine the practical usefulness of the NSA-DEVS formalism, one should next study a set of standard examples with complex event cascades and real-world case studies. This could help answering the crucial question, whether an abundance of new parameters is necessary in real models or if the use of a default delay is sufficient in many cases. Another interesting question is, whether one delay for an atomic model suffices, or if one needs port specific delay times.

Finally one should address the practical usefulness of the simulator and its implementation: How does it perform in comparison to existing PDEVS or RPDEVS simulators? Suitable benchmarks would address simulation times as well as the number of internal messages used inside a simulator. Although the definition of the simulator is a large step forward, much remains to be done before NSA-DEVS can be considered a solid approach for practical discrete event modeling and simulation.

References

- [1] Zeigler BP. *Theory of Modeling and Simulation*. New York: Wiley-Interscience, 1st ed. 1976.
- [2] Chow ACH. Parallel DEVS: A Parallel, Hierarchical, Modular Modeling Formalism and its Distributed Simulators. *Transactions of The Society for Computer Simulation International*. 1996;13(2):55–67.
- [3] Franceschini R, Bisgambiglia PA, Touraille L, Bisgambiglia P, Hill D. A survey of modelling and simulation software frameworks using Discrete Event System Specification. In: *Proc. of 2014 Imperial College Computing Student Workshop*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2014; pp. 40–49.
- [4] Goldstein R, Breslav S, Khan A. Informal DEVS conventions motivated by practical considerations. In: *Proc. of Symposium on Theory of Modeling & Simulation – DEVS Integrative M&S Symposium*. 2013; pp. 10:1–10:6.
- [5] Preyser FJ, Heinzl B, Raich P, Kastner W. Towards Extending the Parallel-DEVS Formalism to Improve Component Modularity. In: *Proc. of ASIM-Workshop STS/GMMS*. Lippstadt. 2016; pp. 83–89.
- [6] Preyser FJ, Heinzl B, Kastner W. RPDEVS: Revising the Parallel Discrete Event System Specification. In: *9th Vienna Int. Conf. Mathematical Modelling*. Wien. 2018; pp. 242–247.
- [7] Preyser FJ, Heinzl B, Kastner W. RPDEVS Abstract Simulator. *SNE Simulation News Europe*. 2019; 29(2):79–84. doi: 10.11128/sne.29.tn.10473.
- [8] Junglas P. NSA-DEVS: Combining Mealy Behaviour and Causality. *SNE Simulation News Europe*. 2021; 31(2):73–80. doi: 10.11128/sne.31.tn.10564.
- [9] Goldblatt R. *Lectures on the Hyperreals*. New York: Springer. 1998.
- [10] Sarjoughian HS, Sundaramoorthi S. Superdense time trajectories for DEVS simulation models. In: *SpringSim (TMS-DEVS)*. 2015; pp. 249–256.
- [11] Zeigler BP, Muzy A, Kofman E. *Theory of Modeling and Simulation*. San Diego: Academic Press, 3rd ed. 2019.

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

Dávid Bodnár^{1*}, Claudia Krull¹

¹Institut für Simulation und Graphik, Otto-von-Guericke-Universität Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany; *david.bodnar.ovgu@gmail.com

Abstract. Virtual Stochastic Sensors (VSSs) [1] aim to provide insight into stochastic processes by producing statistically relevant estimates of non-measurable system properties. During behavior reconstruction of these discrete stochastic systems the internal system state changes are often described as time-homogeneous distribution functions, as in Conversive Hidden non-Markovian Models (CHnMMs). However, the system behavior might change over time or the sample, used for the model creation, might not describe the system accurately.

In [2] and [3] we have shown that detecting these changes is possible, yet the resource consumption for the re-estimation of the model was a clear problem. In this paper we present a solution to that problem by replacing the used statistical tests with Kernel Density Estimation (KDE) and by integrating the hidden model description into the proxel-based state space simulation method.

By using the Change Adaptation Algorithm (CAA) this paper shows that adapting to runtime changes is possible, while preserving parameters on transitions where no change occurs. The algorithm was tested with 5 different types of Probability Density Functions (PDFs) which showed accurate results. By using the CAA one is able to construct adaptive models for behavior reconstruction without the need to fully parametrize the model. In this way, loss of modeling accuracy in the model construction process can be significantly decreased.

Keywords: Virtual Stochastic Sensor, Conversive Hidden non-Markovian Model, Drift, Model Change, Kernel Density Estimation, Proxel-based Simulation

Introduction

VSSs were introduced in [4]. They are tools to reconstruct the behavior of partially observable processes in discrete stochastic systems. Constructing VSSs relies in practice heavily on manually provided knowledge about the system. But what happens if that information becomes outdated or inaccurate during the model construction or over time? What happens if there are flaws in the model construction of the VSS?

To overcome this limitation this paper introduces the CAA, which describes how a proxel-based analysis of

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

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

1 Related Work

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

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

1.1 Conversive Hidden non-Markovian Model

The Hidden Markov Model (HMM) [7] is a well researched technique to analyze directly not observable models through probabilistic symbol emissions. The HMMs assume memoryless model state changes, which is why the concept of Hidden non-Markovian Model

(HnMM) was introduced in [8] to extend the HMMs with state changes governed by arbitrary continuous distribution functions. Using HnMMs one is able to define time dependence between different system states.

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

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

1.2 Proxels-Based Analysis

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

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

The analysis uses discrete timesteps to track the state changes during the process. The connection between a parent proxel and its children in the next time step is characterized by the Hazard Rate Function (HRF) in Equation 2 which describes the rate of probability that a specific state change will happen in the next timestep if it has not happened yet.

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

The network of parent and child proxels in the simulation domain construct the so-called proxel tree, which tracks all possible system states in discrete time steps. To prevent state-space-explosion and to provide accept-

able simulation times, impossible or very unlikely proxels are pruned.

1.3 Kernel Density Estimation

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

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

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

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

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

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

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

1.4 Energy distance

The energy distance [16] (Equation 5) describes a statistical distance based on Newton’s potential energy between two independent random samples (X and Y) described by the Cumulative Distribution Functions (CDFs) F and G . The resulting distance is 0 if and only if $F = G$.

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

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

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

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

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

2 Change Adaptation Algorithm

The CAA utilizes the KDE ideas described in Section 1.3 to construct the transition distributions in every time step. This extension has the additional positive effect that the state transitions are not directly linked to any elementary distribution function and its preliminary properties.

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

By using KDE, the CAA extends the content of a single time step. Before, the time step t_i contained n probable system states from which the most probable one(s) will survive. With CAA the number of proxels in a given time step represent $m \leq n$ different models with n different system states from which the most probable one(s) will survive.

Upon a model drift on a specific transition, the new samples will be automatically assigned to the history of the most probable transition as a result of Equation 2. By doing so, the new samples push out the old

model samples from the history with every new time step and this automatically results in a reconfiguration of the CHnMM model through the KDE. As a result, the model is able to adjust itself to model changes with a given delay defined by the length of the history vector and the mean of the new samples.

2.1 Implementation

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

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

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

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

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

KDE is a resource intensive computation. We used a look-up-table to speed up the algorithm with a residual error in the PDF of 10^{-5} . This error does not cause a significant difference in the end results unlike other computations in the background, like the Kingsbury-Rayner formula [7], that have a much higher computational error. This discretization has the negative drawback that very strong changes in the model result in 0 probability instead of a very small one which can result in a died out proxel tree. This means, that in practice the algorithm will need a roughly usable preseed at the beginning.

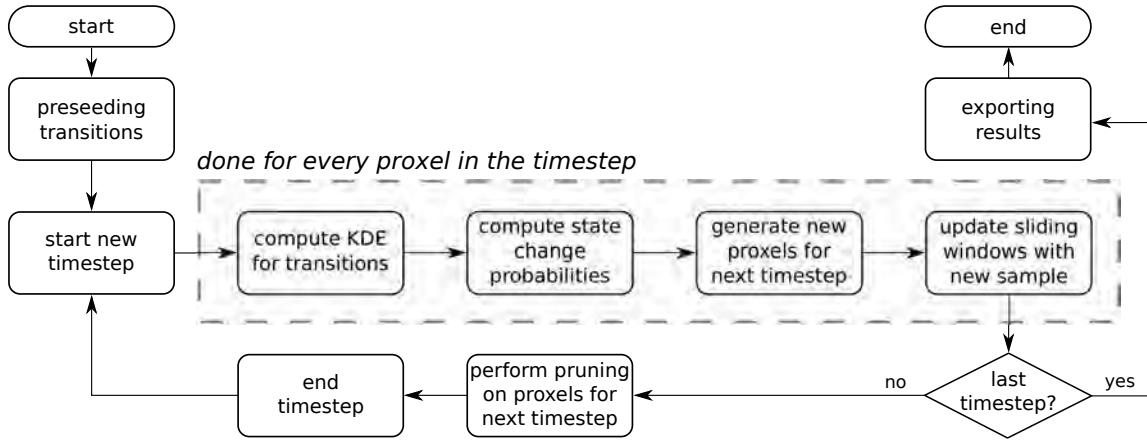


Figure 1: Flowchart of the Change Adaptation Algorithm

2.2 Parameters

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

2.2.1 Window size

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

2.2.2 Pruning

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

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

The pruning threshold r , as it can be seen in the

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

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

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

3 Experiments

3.1 Experiment setup

To validate the CAA, a basic academic example was used, originally introduced in [6]. In this model two production lines get merged before a common quality

tester. Both lines produce faulty products with a given probability.

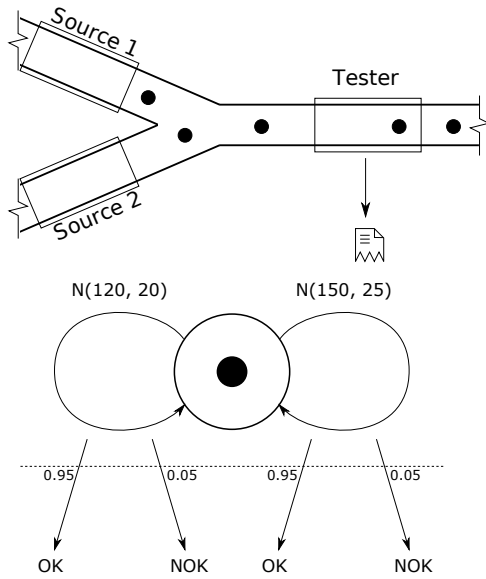


Figure 2: Quality tester example [6] and its ASPN [1]

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

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

The input data for the experiment was generated in a way that Source 2 received ground truth information, while for Source 1 samples from a changed distribution were generated. For easier comparison of the results, Scipy's [22] location-scale(-shape) parametrization was used. Please, be aware that this parametrization differs from the standard academic notation. Five distribution function types were analyzed: normal, uniform, exponential, lognormal and Weibull distribution. The

change consisted of location parameters on a grid of $[60, 150]$ with a step size of 10, while the scale parameter was tested between $[5, 30]$ with a step size of 5. For PDF types lognormal and Weibull an additional shape parameter was used to test a wider range of possible changes. For every parameter combinations, 20 random experiments were executed.

3.2 Parameter selection

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

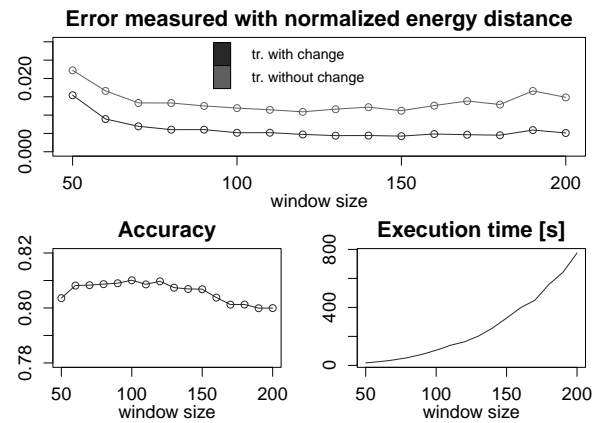


Figure 3: Impact of window size. The error is measured between the ground truth and the model distributions reached at t_{end} . The accuracy plot shows the rate of correctly classified symbols.

In Figure 3 one can see that by increasing window size the execution time increases drastically, as the KDE computation has $O(n^2)$ [13] computational complexity. However, we can also see that choosing a greater window size does not automatically lead to better performance as described in Section 2.2.1. Similarly, if we choose the window size too small then the outliers have a negative impact on the results. The optimal window size lies in this case between 120 – 130 so we chose 125 for the experiments.

In Figure 4 one can see that with increasing pruning threshold the model and the symbol classification become more inaccurate. This is a direct impact of reduced diversity in a time step. The optimal pruning

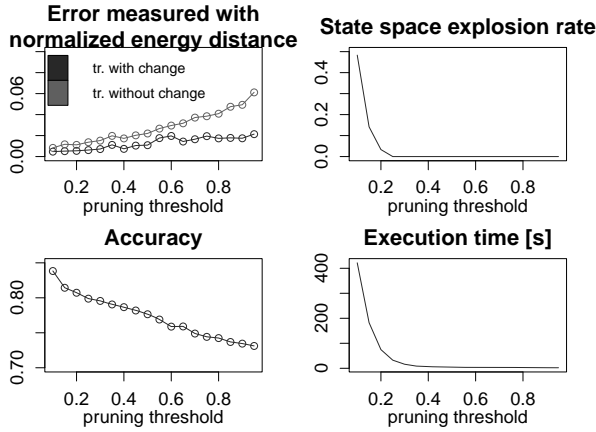


Figure 4: Impact of pruning threshold. The error is measured between the ground truth and the model distributions reached at t_{end} . The plot with the state space explosion shows the rate of failed experiment executions due to out of RAM. The accuracy plot shows the rate of correctly classified symbols.

threshold can be found around the value 0.2, however, there is a significant risk of experiencing state space explosion.

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

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

3.3 Experiment results

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

Surprisingly, even in case of the exponential distribution, we get accurate results. Without the CAA, as written in [2] and [3], concurrent exponential transitions were always a problem for the proxel-based simu-

lations as the HRF is in this case constant and all symbol emissions get assigned to the transition with the higher HRF. KDE cannot fully reproduce the exponential distribution in the background, in this way, by violating the theory of the exponential distribution we get practically usable results.

Table 2 shows the model errors at the end of the experiment. There is an expected amount of faulty results in the experiment which was basically predefined by the setup. The defined parameters in Subsection 3.1 already show that there are cases when the two transitions will (with a very high probability) merge. These expectations were documented in the „Tr. merge” column of the table with ranges in square brackets. One can see that these expectations were always fulfilled, but the values were in the lower range. This means that the CAA was able to distinguish in some cases between distributions which were very close to each other, however, this is most probably pure luck.

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

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

One can also see that the lognormal distribution results are by far the worst for the CAA. However, theoretically there should be no significant limitation for this distribution type. A high amount of bad models can be traced back to transition flips. After a deeper

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

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

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

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

analysis, it turned out that 90% of the models classified to be bad can be traced back to heavily tailed lognormal distributions. This is most probably a result of inaccurate KDEs and a too small window size. We assume that by increasing the window size these model problems would disappear.

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

4 Conclusion

The experiments show that the CAA is efficiently able to track and to adapt to model changes in CHnMMs positively affecting the symbol classification accuracy. Doing so does not affect the capability of withstanding changes on transitions where no change occurred.

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

independent sample upon start.

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

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

4.1 Applications and Future Research

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

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

eral pruning algorithm for VSSs.

Acknowledgement

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

References

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

A Hybrid User Model for Virtual Stochastic Sensors

Claudia Krull^{1*}

¹Otto-von-Guericke-University,
Magdeburg, 39106 Germany ; *claudia.krull@ovgu.de

Abstract. Virtual stochastic sensors (VSS) enable the reconstruction of unobserved behavior of discrete or hybrid stochastic systems from observable output. Augmented stochastic Petri nets (ASPN) are user models for VSS and describe discrete stochastic models that produce discrete output symbols based on the transitions or system states. Hybrid ASPN (H-ASPN) can describe hybrid stochastic systems with continuous quantities that are influenced by and interact with the discrete system parts, producing samples of these continuous system quantities as observable output. Real-world systems often contain both of these types of observable output. In household energy models, the total consumption is a continuous quantity that can be sampled regularly. Additionally, the usage behavior of some appliances might be known in advance or can be monitored easily, resulting in observable discrete symbols. Being able to utilize both of these for behavior reconstruction, promises better results than using either one alone.

In this paper, we describe an extended H-ASPN paradigm for modeling symbol output as well as sample measurement for partially observable hybrid stochastic systems. We demonstrate the paradigm on a small non-intrusive appliance load monitoring (NIALM) example and test the behavior reconstruction. The extended H-ASPN modeling paradigm enables faster and more reliable behavior reconstruction results, when using both observable symbols and samples. The experimental results indicate that extended H-ASPN could lead the way to practically feasible VSS for hybrid systems.

Introduction

Simulation and modeling are used in a forward manner to observe existing systems, build abstract representations and experiment with these in order to draw conclusions on the original systems behavior. Many systems are however not directly observable, but only through their output or interaction with the environment. To analyze these partially observable systems, a backward approach of behavior reconstruction based on the observed output becomes necessary.

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

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

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

1 State of the Art

The concept of virtual stochastic sensors (VSS) was introduced in [2] and has since evolved into a framework to describe and solve backward problems in modeling and simulation ([1]). VSS enable the behavior reconstruction of a broad range of partially observable stochastic systems based on the observable output. Originally, hidden non-Markovian models (HnMM) were developed as computational models for VSS ([3]). Increasing feasibility, converse hidden non-Markovian models restrict the modeling power to enable a much more efficient solution ([4]). Hybrid hidden non-Markovian models (HHnMM) expand the paradigm to include continuous measures ([5]). The Proxel method ([6]) is used for the actual behavior reconstruction task. It is a state space-based simulation method, which enables a deterministic analysis of stochastic models with arbitrarily distributed process durations employing the method of supplementary variables. Behavior reconstruction for discrete systems of realistic size is currently feasible. However, the extension to hybrid systems increases the computational complexity drastically, making hybrid VSS only applicable to small scale academic models so far. The user models defined for VSS are augmented stochastic Petri-nets (ASPN), which were first introduced in [4]. ASPN are based on well known stochastic Petri net paradigms ([7, 8]) and expand these using ideas from hidden Markov models (HMM) ([9]) by emissions of discrete symbols. These emissions can either depend on the current system state or can be triggered by the firing of a transition. In both cases, the emission time is recorded in a protocol along with the emitted symbol. The hybrid components in H-ASPN are modeled using ideas from stochastic reward nets (SRN) ([10]) and fluid stochastic Petri nets (FSPN) ([11, 12]). The observable output of a hybrid ASPN is generated by an independent sampling process, that records the values of one or

more of the continuous quantities in a protocol along with the sampling time stamp. [1]

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

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

2 Hybrid Augmented Stochastic Petri Nets

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

Formally, an H-ASPN is a tuple

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

with the following elements:

- $P = \{p_1 \dots p_{|P|}\}$ is a set of *Places* representing physical locations or system states. The marking of the H-ASPN is given by the distribution of tokens in the places of the net $M \in \hat{M} = \mathbb{N}^{|P|}$.

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

If a state can produce output symbols, then the sum of the probabilities of all output symbols of the state must be 1: $\exists b(v_i, p_j) > 0 \Rightarrow \sum_{k=1 \dots |V|} b(v_k, p_j) = 1$.

- If the output is generated depending on the ASPN transitions, then $b : V \times T \rightarrow [0, 1]$. If a transition can produce output symbols, then the sum of the probabilities of all output symbols of that particular transition must be 1: $\exists b(v_i, t_j) > 0 \Rightarrow \sum_{k=1 \dots |V|} b(v_k, t_j) = 1$.

- $W : \{w_1 \dots w_{|W|}\}$ is a set of variables, representing continuous system quantities. The current values of these quantities are given in $w_{|W|} \in \hat{W} = \mathbb{R}^{|W|}$.
- $w_0 = (w_1 \dots w_{|W|})$ contains the initial values of the continuous system quantities.
- $GF : T \times \mathbb{N}^{|P|} \times \mathbb{R}^{|W|} \rightarrow 0, 1$ describes the marking dependent guard of each transition. If the Boolean expression evaluates to 1, the transition is enabled, and it is disabled otherwise.
- $ir : T \times \hat{M} \times \hat{W} \rightarrow \mathbb{R}$ (or $\mathbb{R}^{|W|}$) describes a type of impulse reward, which can change the value of a continuous quantity immediately. The change can be dependent on the current marking of the ASPN.
- $rr : P \times \hat{M} \times \hat{W} \rightarrow \mathbb{R}$ (or $\mathbb{R}^{|W|}$) describes a type of rate reward, which can change the value of a continuous quantity continuously. The rate of change can be dependent on the current marking of the net. The rate reward can be associated to a specific place or can be completely independent of the discrete system state and be represented by an ODE.

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

Elements V and b hold the augmentation information. The specific output semantics are the following: when a transition with associated output symbols fires, one of these symbols is emitted, sampled according to their probabilities. When symbols are associated with places or specific system states, observations are made through an independent process. The symbol is sampled depending on the current system state and associated output symbols. Observed outputs are collected in

a protocol with their respective emission or sampling time stamps. Samples of the continuous measures can be taken at any time, and will be collected in a protocol associated with the time when they were drawn. The protocol can contain values of different continuous quantities, but not every continuous system quantity must be represented.

2.1 Modifications to Computational Model and Solution Algorithm

In order to enable behavior reconstruction for this modified H-ASPN model type, we also need to adapt the computational model and solution algorithm. Analogously to the combination of the user models, the computational model was created by combining the output processes of hidden non-Markovian models (HnMM) and hybrid hidden non-Markovian models (HHnMM) [1] to incorporate discrete symbol emissions as well as samples of continuous model variables. As solution method, we extended the Proxel-based analysis algorithm for hybrid models (see [1, 5]) to perform behavior reconstruction for extended H-ASPN. The trace can now contain samples of the continuous model quantities as well as discrete output symbols, and both will be handled accordingly. As described in [5], the parametrization process for the Proxel method for hybrid systems is tedious, since the inclusion of continuous variables considerably increases the complexity of the models. Countering the inherent problem of state-space explosion of the Proxel method becomes more difficult with more method parameters. Instead of having a binary decision on the validity of a Proxel, as with discrete outputs, the estimate of the continuous measure in the Proxel can be within ε of the observed sample. The size of ε and the cutoff probability or number for pruning the Proxel tree both need to be balanced, in order to achieve a useful reconstruction result in a computationally feasible manner. In the following experiment section, we will show how the behavior reconstruction works for the extended paradigm.

3 Example and Experiments

For our proof of concept, we will use a small example constructed from the UMass Smart* Data Set for Sustainability [15] (2013 release). The data was gathered from several private homes over the period of several months, and includes aggregate power readings, circuit

and appliance level consumption as well as environmental data. The goal of non-intrusive appliance load monitoring (NIALM) is to dis-aggregate a cumulative household energy consumption into the contributions of individual appliances or circuits. For demonstration purposes, we have chosen a subset of three circuits with different characteristics from House A, namely the *Dryer*, *MasterLights* and *CounterOutlets1*. For a detailed description of the model construction, please refer to [16]. The data set uses UNIX time stamps, and thus seconds as basic time unit. The goal for the experiments is to determine the unobserved system behavior in terms of appliance state changes from the cumulative power consumption, with and without information on individual appliances state changes.

3.1 NIALM Example System

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

In the formal description of the H-ASPN, H , ir and GF are omitted, since the system does not include inhibitor arcs, impulse rewards or guard functions.

$$\begin{aligned}
 \text{Circuits} &= (P, T, I, O, M_0, V, b, W, \vec{v}, rr) \\
 P &= \{Dr1, Dr2, Dr3, Dr1_{Done}, Dr2_{Done}, \\
 &\quad ML1, ML2, CO1, CO2\} \\
 T_I &= \{Dr1to2, Dr1to3, Dr2to1, Dr2to3\} \\
 T_T &= \{Dr1_T, Dr2_T, Dr3_T, ML1_T, ML2_T, \\
 &\quad CO1_T, CO2_T\}
 \end{aligned}$$

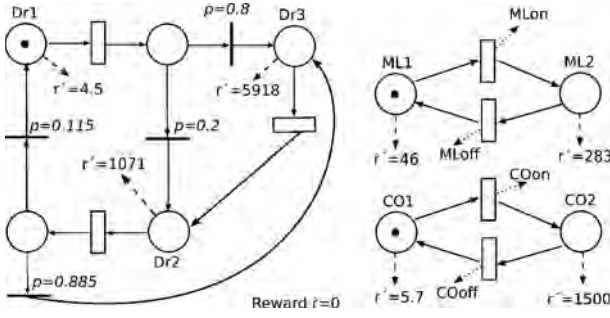


Figure 1: H-ASPn of the NIALM Example Process

$$I(p, t) = \begin{cases} 1: & (p = Dr1 \wedge t = Dr1_T) \\ & \vee (p = Dr3 \wedge t = Dr2_T) \\ & \vee (p = Dr2 \wedge t = Dr3_T) \\ & \vee (p = Dr1_{Done} \wedge (t = Dr1to2 \vee t = Dr1to3)) \\ & \vee (p = Dr2_{Done} \wedge (t = Dr2to1 \vee t = Dr2to3)) \\ & \vee (p = ML1 \wedge t = ML1_T) \\ & \vee (p = ML2 \wedge t = ML2_T) \\ & \vee (p = CO1 \wedge t = CO1_T) \\ & \vee (p = CO2 \wedge t = CO2_T) \\ 0: & \text{else} \end{cases}$$

$$O(p, t) = \begin{cases} 1: & p = Dr1_{Done} \wedge t = Dr1_T \\ & \vee p = Dr2_{Done} \wedge t = Dr2_T \\ & \vee p = Dr1 \wedge t = Dr2to1 \\ & \vee p = Dr2 \wedge t = Dr1to2 \\ & \vee p = Dr2 \wedge t = Dr3to2 \\ & \vee p = Dr3 \wedge t = Dr1to3 \\ & \vee p = Dr3 \wedge t = Dr2to3 \\ & \vee p = ML1 \wedge t = ML2_T \\ & \vee p = ML2 \wedge t = ML1_T \\ & \vee p = CO1 \wedge t = CO2_T \\ & \vee p = CO2 \wedge t = CO1_T \\ 0: & \text{else} \end{cases}$$

$$M_0 = (1, 0, 0, 0, 0, 1, 0, 1, 0)$$

$$V = \{MLon, MLoFF, COon, COoff\}$$

$$b(MLon, ML1_T) = 1$$

$$b(MLoFF, ML2_T) = 1$$

$$b(COon, CO1_T) = 1$$

$$b(COoff, CO2_T) = 1$$

$$\text{else } b(v, t) = 0$$

$$W = r$$

$$\vec{w}_0 = (0)$$

$$rr(Dr1) = 4.5$$

$$rr(Dr2) = 1071$$

$$rr(Dr3) = 5918$$

$$rr(ML1) = 46$$

$$rr(ML2) = 283$$

$$rr(CO1) = 5.7$$

$$rr(CO2) = 1500$$

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

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

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

Table 1: Excerpt of the Cumulative Consumption Trace with Symbols

In the experiments, we want to test, how including some symbols in the trace affects the accuracy of

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

Table 2: Ground Truth for Appliance State Changes

the reconstruction, compared to a reconstruction based only on samples. Therefore, we constructed the model in different variations: (0) no symbol emissions at all, (1) state changes of the *MasterLights* and *CounterOutlets* are detectable, however cannot be distinguished through the symbol emitted, (2) the state changes of *MasterLights* and *CounterOutlets* are detectable and can be distinguished from their symbols, which corresponds to the model formalized in the previous section. In case (0), the trace only contains samples of the continuous quantity, and the times when they were measured. In cases (1) and (2) the trace also contains symbols with the time stamps of when they were emitted.

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

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

Outlet switches, but at least matching the switch timings. Considering the ambiguity and large time step size of the reconstruction, this is a mediocre result.

Reconstructed Path	
Time Stamp	State Change
6120	<i>CO1to2</i>
6240	<i>CO2to1</i>
7620	<i>CO1to2</i>
7920	<i>CO2to1</i>

Table 3: Reconstructed Path Without Detectable State Changes

As a next step, we used model variant (2), where *MasterLight* and *CounterOutlets1* state changes are detectable and can be distinguished by the symbols emitted. Unfortunately, the paths reconstructed using the model with Weibull distributions again contained one state change in almost every time step, and were therefore not useful here. The most likely path reconstructed using the model with Exponential distributions is shown in Table 4. The path resembles the ground truth very closely, only the *Dryer* state change from *Dr1* to *Dr2* at 6120 and back again 2 minutes later does not correspond to the actual system behavior. This shows that, even if only some state changes are detectable, the reconstruction resembles the ground truth much better than without detectable state changes.

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

Table 4: Reconstructed Path With Distinguishable State Changes

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

and *CounterOutlets1* state changes.

We also want to compare the algorithm performance when including or excluding symbols. We will only use the two models using Exponential distributions, since only these resulted in useful reconstructions. When comparing the runtime for both algorithms with the same ϵ and cutoff threshold parameters, the analysis of the model without symbol emissions needs 20 seconds, whereas the analysis when including symbols needs only about 4 seconds.

Figure 2 shows the development of the number of valid paths over time. The upper graph shows the number oscillating around 1500 but staying above 500 after an initial period. The lower graph shows more pronounced drops in this number, some of which can be associated to observed symbols. This is due to the higher degree of observability of the model including detectable state changes, and thus a larger number of paths becoming invalid throughout the analysis.

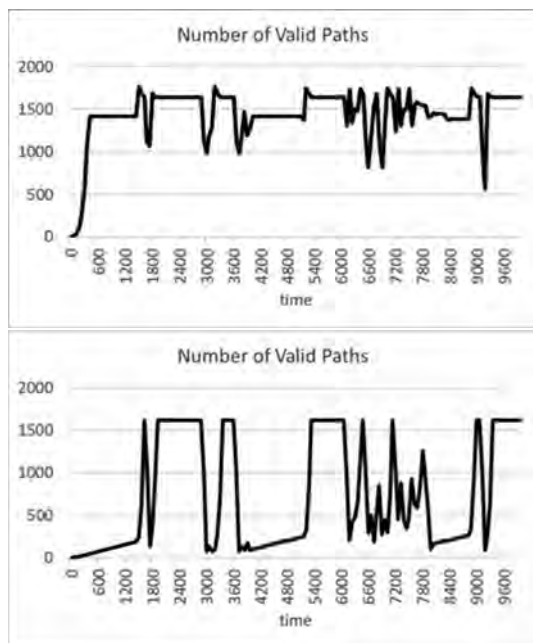


Figure 2: Valid Paths Over Simulation Time Without (Top) And With (Bottom) Detectable State Changes

Figure 3 shows the development of the trace probability in log scaling over time. In both cases, the probability decreases gradually, until the first detectable state change, also in the model without state change detection. In the lower graph, each detectable state change leads to a more noticeable drop in trace probability,

since invalidating paths decrease the remaining probability in the analysis.

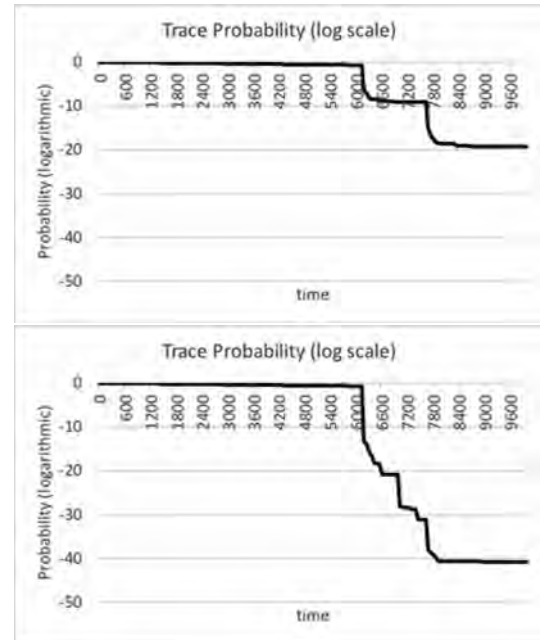


Figure 3: Trace Probability Over Simulation Time Without (Top) And With (Bottom) Detectable State Changes

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

4 Conclusion and Outlook

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

References

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

Simulation Technischer Systeme

Simulation of Technical Systems

Manöverbasierte Validierung einer Längsdynamiksimulation anhand generisch erzeugter Reallastverläufe (L)

S. Cortès, C. Dettmann, L. Heinzelmann, O. Haidar

Robustheitsanalyse einer Längsdynamiksimulation auf Basis verschiedener Qualitätskriterien (L)

S. Cortès, C. Dettmann, J. Heib, F. Millberg

Modellierung des Betriebsverhaltens eines Synchrongenerators als Stromerzeuger (L)

B. H. Herrmann, M. Ringel, D. Jörss, C. Fink

Numerische Simulation einer asymmetrischen Drallströmung - Vergleich der numerischen Turbulenzmodellierung mit experimentellen Messungen (K)

K. Zacharias, D. Welsch, W. Schlüter, A. Buchele

Systematischer modellbasierter Entwurf einer Reinforcement Learning-basierten neuronalen adaptiven Geschwindigkeitsregelung (L)

X. Liu-Henke, S. Jacobitz, Or Aviv Yarom, J. Fritz

Model-based Development of an Automated and Remotely-Controlled Demolition Excavator (L)

C. Richter, M. Shakoorianfard, V. Waurich, F. Will

Efficient state estimation for gas pipeline networks via low-rank approximations (L)

N. Stahl, N. Marheineke

Robotics simulation – A comparison of two state-of-the-art solutions (L)

M. Zwingel, C. May, M. Kalenberg, J. Franke

Parameter tables for PID controllers for time delayed systems optimized with a learning method (L)

R. Büchi

Interacting-Multiple-Model Partikelfilter zur Fehleridentifikation (L)

A. Lepp, D. Weidemann

Modelica simulation of pendulums and crane crab: PyMbs vs. OpenModelica (K)

M. Rößler, M. Ehrmann, F. Breitenecker

Die Kurzbeiträge (K) sind zu finden im **ARGESIM Report 19**,

Proceedings Kurzbeiträge ASIM SST 2022

ISBN ebook 978-3-901608-96-4, DOI: 10.11128/arep.19, ASIM Mitteilung AM 179

Manöverbasierte Validierung einer Längsdynamiksimulation anhand generisch erzeugter Reallastverläufe

M. Eng. Sven Cortès^{1*}, Prof. Dr.-Ing. Christian Dettmann¹,

B. Eng. Omar Haidar¹, B. Eng. Linus Heinzelmann¹

¹Technische Hochschule Ulm, Prittwitzstraße 10, 89075 Ulm, Germany; *sven.cortes@thu.de

Abstract. Werden reale Lastzustände von Automobilen auf einem Rollenprüfstand simuliert, so stehen neben der klassischen 1D-Lastsimulation weitere Prüfstandsregelarten zur Verfügung. So kann beispielsweise der Betriebszustand des Prüflings durch die Kombination von Fahrzeuggeschwindigkeit und eine den Fahrerwunsch beschreibende Größe ebenfalls eindeutig bestimmt werden.

Dieses Paper zeigt eine Methode auf, die es ermöglicht, ein Längsdynamikmodell eines Fahrzeuges anhand von Prüfstandsmessungen zu parametrisieren und manöverbasiert zu validieren. Die Validierungsumgebung besteht aus einem Rollenprüfstand und einem automatisierten Fahrroboter mit elektronischem Fahrpedal und wird innerhalb des IPEK X-in-the-Loop-Frameworks eingeordnet. Anhand der Längsdynamiksimulation können nun generische Streckenprofile erzeugt werden. Der Übertrag dieser Lastzustände auf die Validierungsumgebung erfolgt über eine definierte Schnittstelle, welche die unterschiedlichen Prüfstandsregelarten berücksichtigt.

Einleitung

Mit der Einführung des Real Driving Emissions (RDE)-Verfahrens im September 2017 fand ein Wandel innerhalb der Fahrzeugentwicklung von der Abgaszertifizierung und emissionsrelevanten Tests unter Laborbedingungen hin zu Fahrten im realen Straßenverkehr statt. Aufgrund der Fülle an emissionsrelevanten Einflussfaktoren bei Realfahrten sind diese nicht vollständig reproduzierbar. Insbesondere die Verkehrssituation und die Witterungsbedingungen sind dabei problematisch. Bei Fragestellungen, wie zum Beispiel der Steuergeräteapplikation, ist je-

doch die Reproduzierbarkeit der Testfälle besonders relevant. Aus diesem Grund stellt die Entwicklung geeigneter Validierungsumgebungen und Testfälle eine Notwendigkeit dar, wobei häufig von einem sogenannten „Road to Rig“ (R2R)-Konzept gesprochen wird. [1,2]

Der IPEK X-in-the-Loop (XiL)-Ansatz, welcher am Institut für Produktentwicklung (IPEK) des Karlsruher Instituts für Technologie (KIT) entwickelte wurde, bietet eine durchgängige Methodik zur Validierung von Systemen auf unterschiedlichen Systemebenen während des gesamten Entwicklungsprozesses. Das betrachtete System selbst und die mit ihm interagierenden Systeme können dabei in virtueller oder physischer Form und mit unterschiedlichem Detaillierungsgrad vorliegen. [3] Bei der Anwendung der Methodik in der Fahrzeugentwicklung kann durch die Simulation der Umweltbedingungen und Lastzustände des Fahrzeugs der Übergang von Fahrversuchen zur Validierung am Rollenprüfstand erreicht werden. Der Fahrer als drittes relevantes System kann anhand von virtuellen Fahrermodellen, Fahrrobotern oder realen Fahrern umgesetzt werden. [4]

Die Simulation der Lastzustände erfolgt durch die Prüfstandsregelung. Dabei kommt im Rahmen der Straßenlastsimulation häufig ein 1D-Punktmassenmodell mit experimentell bestimmten Fahrwiderstandskoeffizienten zum Einsatz. [5] Alternative Regelarten bieten dagegen die Möglichkeit, die Betriebspunkte anhand der Fahrzeuggeschwindigkeit oder Raddrehzahl und der Fahrpedalstellung durch die Prüfstandsregelung und einen Fahrroboter einzustellen. Die ergebnisgleiche Abbildung der Lastzustände, verglichen mit der Straßenlastsimulation wird durch die Untersuchung von CORTÈS et al. [6] be-

stätigt. Es werden dabei Fahrmanöver aus einer Datenbasis realer RDE-Fahrten anhand gemessener Signale in der Validierungsumgebung reproduziert. Durch die Anwendung der alternativen Regelarten ergibt sich der Vorteil eines geringeren Aufwands zur Durchführung der Manöver, da sowohl die komplexe messtechnische Erfassung des Steigungsprofils als auch die aufwendige Justage der simulierten Fahrwiderstände am Rollenprüfstand bei der Lastanpassung umgangen wird. Bei der vorgestellten Methodik zur Anwendung der Regelarten werden jedoch Messdaten real durchgeführter Fahrten benötigt.[6]

Am Institut für Fahrzeugsystemtechnik (IFS) der Technischen Hochschule Ulm (THU) wird derzeit der Ansatz verfolgt, eine modellbasierte Fahrzeugsimulation als digitalen Prototyp zur Generierung von Daten zu verwenden, die den Betrieb eines Rollenprüfstands anhand alternativer Regelarten entsprechend [6] ermöglicht. Dadurch sollen Fahrzeugsysteme auf Basis generischer oder gemessener Geschwindigkeitsverläufe validiert werden können.

Der vorliegende Beitrag betrachtet in diesem Zusammenhang die Hypothese, dass bereits ein einfaches Längsdynamikmodell eines Fahrzeugs, welches auf Parametern wie stationären Kennfeldern basiert, Simulationsergebnisse von ausreichender Güte liefern kann, um sowohl einfache Fahrmanöver aus gesetzlichen Prüfzyklen als auch generische Reallastverläufe auf einen Rollenprüfstand übertragen zu können. Zur Überprüfung der Hypothese wird das parametrisierte Modell hinsichtlich der Abbildung des makroskopischen Fahrzeugverhaltens bei den relevanten Fahrmanövern gegenüber Prüfstandsmessungen validiert.

1 Methodik

Die Parametrisierung und Validierung des Längsdynamikmodells erfolgt auf Basis des erweiterten ZHO-Modells nach ALBERS [3]. Dieses beschreibt den Entwicklungsprozess als zyklischen Wechsel zwischen Kreation bzw. Synthese und Validierung, wobei diese sowohl der Überprüfung definierter Anforderungen und deren Umsetzung als auch der reinen Gewinnung von Erkenntnissen über das Systemverhalten dienen kann. [3] Die Vorgänge der Modellierung der Längsdynamik und der Parametrisierung des erzeugten Modells stellen somit separate Syntheseprozesse dar, während die Bestimmung der Parameter ebenso wie die Überprüfung der Anforderun-

gen aus der Forschungshypothese als Validierungsprozess verstanden wird, da sie einen Erkenntnisgewinn über das Fahrzeugverhalten voraussetzt. Beide Validierungsprozesse erfolgen manöverbasiert. Hierzu wird anhand der definierten Validierungsziele ein gemeinsamer Manöverkatalog erstellt, wobei jedes Manöver anhand des initialen Fahrzeugzustands, der Fahrereingaben und Randbedingungen beschrieben wird. Für die einzelnen Manöver oder auch Testfälle wird zudem eine Testumgebung definiert, die eine Konfiguration der Validierungsumgebung zur konkreten Umsetzung des jeweiligen Manövers darstellt. [7] Die Auswertung der Ergebnisse erfolgt schließlich innerhalb der Testinterpretation. [3]

1.1 Validierungsumgebung

Die verwendete Validierungsumgebung besteht aus einem Maha AIP Abgasrollenprüfstand in Kombination mit einem Fahrroboter des Herstellers AIP Automotive GmbH & Co. KG. Die Infrastruktur der Validierungsumgebung wird in vereinfachter Form in Abbildung 1 dargestellt. Der Fahrroboter erlaubt sowohl den Betrieb in Geschwindigkeitsregelung als auch die direkte Vorgabe von Sollwerten der Fahrpedalstellung α , die anhand eines E-Gas Moduls umgesetzt werden. [8] Da Verzögerungsphasen nicht anhand der Fahrpedalstellung umgesetzt werden können, wechselt der Fahrroboter hierzu in Geschwindigkeitsregelung, sobald α den Wert null annimmt. Die Regelung des Rollenprüfstands ermöglicht sowohl den Betrieb in Straßenlastsimulation als auch in Geschwindigkeitsregelung, wobei zur Umsetzung der alternativen Regelarten v - α und n - α sowohl die Geschwindigkeit v als auch die Raddrehzahl n als Sollwert vorgegeben werden kann. Bei der Lastsimulation erfolgt die Simulation der Last anhand von Gleichung (1). Sie steht im Gleichgewicht mit der Zugkraft F_Z des Fahrzeugs. Der Term F_W steht dabei für den Fahrwiderstand, welcher anhand der Straßenlastkoeffizienten A, B, C berechnet wird. Die Größen F_a und F_{St} bezeichnen die Beschleunigungs- und Steigungswiderstände. [5]

$$\begin{aligned} F_Z &= F_W + F_a + F_{St} \\ &= A + B \cdot v + C \cdot v^2 + F_a + F_{St} \end{aligned} \quad (1)$$

Durch das Einregeln der Sollgeschwindigkeit eines Fahrmanövers durch den Fahrroboter ist der Lastzustand des Fahrzeugs somit jederzeit eindeutig bestimmt. Bei den alternativen Regelarten wird dies durch die Geschwindigkeitsregelung des Prüfstands und das Antriebsmoment des Fahrzeugs entsprechend der vorgegebenen Fahrpedalstellung erreicht. [5]

Die Messung der Zugkraft, der Geschwindigkeit und der Beschleunigung erfolgt durch die Prüfstandssensoren mit einer Abtastrate von 1 kHz. Zur Messung der Fahrpedalstellung kommt ein AD-Wandler des Herstellers DEDITEC GmbH zum Einsatz. Er wird mit dem Potentiometer des elektronischen Fahrpedals verbunden und misst den zeitlichen Verlauf der Spannung mit einer Abtastrate von 500 Hz. Die Raddrehzahl wird anhand eines Drehgebers des Herstellers Kistler Instrumente AG mit einer Auflösung von 1000 Pulsen pro Umdrehung aufgezeichnet. [12] Informationen über den Betriebszustand des Fahrzeugs liefert die standardisierte OBD 2 Schnittstelle. Zur Aufzeichnung der Messdaten kommt ein autarkes Rechnersystem zum Einsatz.

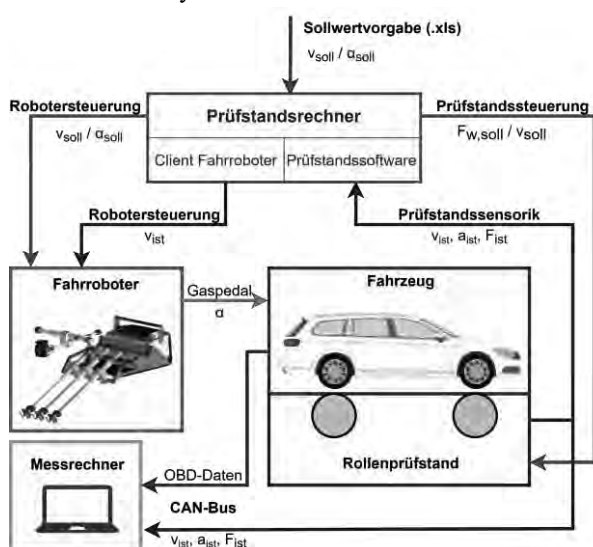


Abbildung 1: Infrastruktur der Validierungsumgebung am IFS der THU

Als Versuchsfahrzeug kommt ein Pkw der Mittelklasse mit einem 2,0 l Dieselmotor und Doppelkupplungsgetriebe zum Einsatz. Die wichtigsten Fahrzeugdaten sind in Tabelle 1 dargestellt.

Hersteller/Modell	VW Passat B8
Motor	Diesel, 4 Zyl, 1968 cm ³
Leistung	110 kW bei 3500 – 4000 1/min
Drehmoment	340 Nm bei 1750 – 4000 1/min
Antriebskonfiguration	Front-Quer
Getriebe	7-Gang Direktschaltgetriebe (DSG)
Gesamtmasse	1928 kg

Tabelle 1: Technische Daten des Versuchsfahrzeugs

Das Gesamtgewicht des Fahrzeugs setzt sich dabei aus dem Leergewicht sowie dem Gewicht der Messtechnik und zweier Insassen zusammen.

1.2 Modell der Längsdynamik

Zur Modellierung des Versuchsträgers werden zwei Längsdynamikmodelle erstellt. Hierbei wird der gesamte Antriebsstrang vom Verbrennungsmotor bis zum Rad modelliert. Zur Simulation von Prüfzyklen wird die Rückwärtssimulation gewählt. Zur Simulation von Real-lastverläufen bedarf es einer Erweiterung um einen virtu-ellen Fahrer, welcher die Geschwindigkeitsregelung durch einen Fahrer abbildet. Dadurch ergibt sich eine Vorwärtssimulation. Die Dynamik des Fahrermodells kann anhand spezieller Parameter eingestellt werden.

Grundsätzlich wird der Antriebsstrang bestehend aus Verbrennungsmotor, Getriebe, Differential und Räder für beide Modellvarianten identisch parametrisiert. Im Rahmen der Parameterbestimmung werden die für die Modellierung benötigten Kenngrößen und Kennfelder bestimmt. Für den Verbrennungsmotor wird das Massenträgheitsmoment und das Kennfeld des effektiven Drehmoments bestimmt, während das Getriebe auf Basis von Hoch- und Rückschaltkennlinien, den Übersetzungen der einzelnen Gangstufen und dem Massenträgheitsmoment modelliert wird. Für die Parametrisierung der Komponenten zwischen dem Getriebeausgang und der Räder wird ein Massenträgheitsmoment bestimmt und die geometrischen Daten des Fahrzeugs, wie Fahrwiderstandskoeffizienten und Fahrzeugmasse in die Modellierung eingesetzt. Für die Modellierungen werden unter Berücksichtigung des Verwendungszweckes verschiedene Vereinfachungen unternommen. Die Stabilität der Modellierung muss dabei stets berücksichtigt werden. Als Beispiel einer Vereinfachung kann die Modellierung des Doppelkupplungsgetriebes betrachtet werden. Das Doppelkupplungsgetriebe wird auf Basis eines Schaltgetriebes mit Kupplung modelliert. Der Anfahrregler wird dabei nicht vollständig abgebildet, wodurch der Modellierungsaufwand reduziert wird, jedoch Abweichungen bei Anfahrvorgängen entstehen.

1.3 Generische Reallastverläufe

Die um einen virtuellen Fahrer erweiterte Modellierung ermöglicht die Simulation generischer Reallastverläufe. Die Erzeugung der Verläufe basiert auf einem Algorithmus der Simulationssoftware, welcher einzelne Streckensegmente aus einer umfangreichen Datenbasis des

Softwareherstellers Gamma Technologies zu einer Gesamtstrecke mit zuvor festgelegten Eigenschaften anordnet. Die Eigenschaften können dabei unter Berücksichtigung der RDE-Konformität gewählt werden. So lassen sich u.a. die einzelnen Streckenanteile (Stadt, Überland, Autobahn), die Haltedauer bei Ampelphasen, sowie die Gesamtstrecke festlegen.

1.4 Parameterbestimmung

Zur Bestimmung der Modellparameter werden unter Berücksichtigung der messbaren Größen und der Parameter der Validierungsumgebung charakteristische Manöver definiert. Das Vorgehen wird im Folgenden anhand des Fahrmanövers 1 exemplarisch vorgestellt. Es bildet die wesentlichen Betriebszustände ab, die dem Motorkennfeld zugeordnet werden. Dieses stellt den Parameter mit dem höchsten Einfluss auf die Simulationsergebnisse dar.

Manöver 1. Das erste Fahrmanöver setzt sich aus Abschnitten konstanter Geschwindigkeit und variablen Motorlasten zur Durchführung einer stationären Vermessung des Verbrennungsmotors zusammen. Zur Definition der Lastpunkte werden etablierte Methoden der Rastervermessung angewendet, die anhand von Abbildung 2 verdeutlicht werden können. Es wird dabei zunächst der Variationsraum der fahrbaren Drehzahlen und Lasten mit Blick auf die thermische Belastung des Motors vermessen. Anschließend erfolgt eine Unterteilung des Variationsraums anhand eines äquidistanten Rasters, dessen Eckpunkte das Kandidatenset bilden. Die tatsächlichen Messpunkte setzen sich aus den Messpunktkandidaten zusammen, die sich innerhalb der Grenzen des Variationsraums befinden. [11] Die dadurch definierten 28 Lastpunkte werden durch aneinandergereihte Abschnitte konstanter Geschwindigkeiten, bei denen sich in der vierten Gangstufe Motordrehzahlen von 1200 1/min bis 3600 1/min in Abständen von 600 1/min einstellen, angefahren. Die Geschwindigkeitsregelung erfolgt dabei durch den Fahrroboter. Während der konstanten Abschnitte wird die Motorlast ausgehend von null schrittweise erhöht und jeweils für 180 s gehalten, um eine Stabilisierung zu gewährleisten. Die Einstellung der Last erfolgt anhand des simulierten Steigungswiderstands des Rollenprüfstands in Lastsimulation durch Variation der Steigung. Um reproduzierbare Ergebnisse zu erzielen, wird von einem betriebswarmen Motor als Randbedingung ausgegangen. Die Analyse Kriterien des Manövers

stellen die gemessene Zugkraft des Fahrzeugs F_Z und die Fahrpedalstellung α dar.

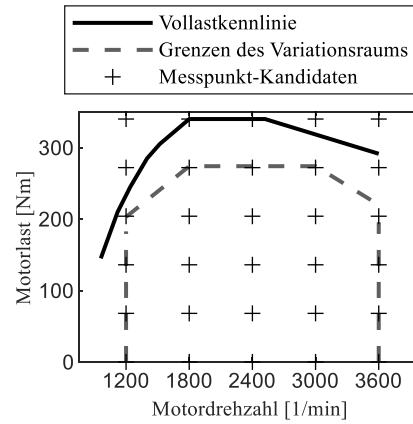


Abbildung 2: Kandidatenset und Versuchsraum der Rastervermessung

Berechnung der Modellparameter. Anhand der gewonnenen Messdaten und Erkenntnisse über das Fahrzeugverhalten erfolgt dessen Abbildung in Modellparametern. Im Falle des vorgestellten Fahrmanövers wird für jeden Lastpunkt, definiert durch die gemessenen Werte der Motordrehzahl n_M und der Fahrpedalstellung α die mittlere Zugkraft $\overline{F_Z}$ berechnet und mit Gleichung (2) anhand des dynamischen Radhalbmessers r_{dyn} , der Gesamtübersetzung in der vierten Gangstufe $i_{ges,4}$ und der im Vorfeld vermessenen Kraft durch die Reibung des Antriebsstrangs $F_{R,A}$ die Drehmomentwerte des Motors $M_M(n_M, \alpha)$ bzw. $M_M(n_M, 0) = M_{Schlepp,M}$ berechnet.

$$M_M = \frac{(\overline{F_Z} - F_{R,A}) \cdot r_{dyn}}{i_{ges,4}} \quad (2)$$

Zur Erzeugung des Rasterkennfelds werden die Drehmomentwerte anhand eines äquidistanten Gitters mit Netzweiten von $\Delta n_M = 100 \text{ 1/min}$ und $\Delta \alpha = 2 \%$ linear interpoliert.

1.5 Validierung des Simulationsmodells

Innerhalb der Modellvalidierung werden zunächst schrittweise die Größen Zugkraft, Fahrpedalstellung und Geschwindigkeit betrachtet, von denen jeweils zwei zur eindeutigen Beschreibung der Lastzustände des Fahrzeugs herangezogen werden können. Zur Generierung von Referenzdaten erfolgt der Betrieb des Rollenprüfstands in Lastsimulation, sodass die Zugkraft des Fahrzeugs sowohl für das Längsdynamikmodell als auch für

das Simulationsmodell die erste Größe zur Definition der Lastzustände darstellt. Durch Variation der zweiten Größe erfolgt dann die Validierung mit der sich aus dem Lastzustand ergebenden dritten Größe als Analyse Kriterium. Um anhand der so gewonnenen Validierungsergebnisse Aussagen über die Übertragbarkeit der Fahrmanöver unter Verwendung alternativer Regelarten tätigen zu können erfolgt zuletzt der Vergleich der Ergebnisse bei der Durchführung mit unterschiedlichen Regelarten.

Für den Validierungsprozess werden einerseits Fahrmanöver der Parameterbestimmung zur direkten Validierung der gewonnenen Modellparametrierung verwendet. Andererseits werden Manöver aus gesetzlichen Prüfzyklen und generischen Streckenverläufen herangezogen, um die Abbildung des Fahrzeugverhaltens in einem breiten Umfang von Last- und Fahrzuständen zu validieren. Zur Reduzierung potenzieller Fehlerquellen, müssen dabei bestimmte Randbedingungen, wie die Umgebungs- und Kühlmitteltemperatur übereinstimmen. Im Folgenden werden beispielhaft drei Manöver des erstellten Manöverkatalogs vorgestellt.

Manöver 2. Das Fahrmanöver 2 beschreibt einen außerstädtischen Abschnitt des NEFZ-Zyklus. Der Fokus liegt hierbei auf der geringen Dynamik und der ausgeprägten Konstantfahrphasen, was die Untersuchung der Zugkraft als Basis der weiteren Validierungsschritte ermöglicht. Als Eingangsgröße des Simulationsmodells dient dabei der gemessene Geschwindigkeitsverlauf in Abbildung 3 a). Dieser wird durch eine Prüfstands-fahrt unter Verwendung der Längsdynamikregelung des Fahrroboters erzeugt.

Manöver 3. Dieses Fahrmanöver wird aus jenem zur Bestimmung der Hochschaltkennlinien des Getriebes abgeleitet. Es basiert auf vier Beschleunigungsvorgängen, wobei dem Fahrroboter konstante Werte der Fahrpedalstellung von 20 % bis 50 % vorgegeben werden. Der Rollenprüfstand wird dabei in Lastsimulation betrieben. Die gemessene Geschwindigkeit, entsprechend Abbildung 3, wird als Eingangsgröße des parametrisierten Modells verwendet. Das Validierungskriterium des Manövers ist die Abbildung motorischer Betriebspunkte auf Basis einer Geschwindigkeitsvorgabe, die sich anhand der berechneten Fahrpedalstellung analysieren lässt.

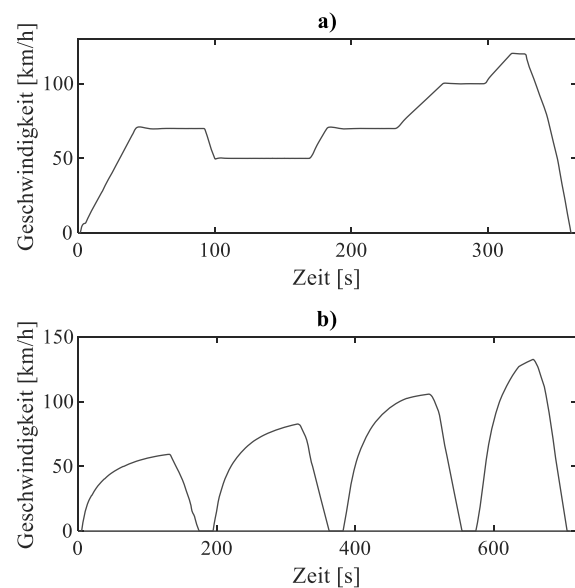


Abbildung 3: Gemessene Geschwindigkeitsverläufe als Eingangsgrößen des Simulationsmodells.
a) Manöver 2, b) Manöver 3

Manöver 4. Das vierte Manöver beschreibt einen Abschnitt eines generischen Geschwindigkeitsverlaufs. Hierzu gehören ein innerstädtischer-, ein außerstädtischer- sowie ein Autobahnabschnitt. Der Verlauf zeichnet sich durch eine erhöhte Dynamik und höhere Spitzengeschwindigkeiten aus. Das Geschwindigkeitsprofil wird innerhalb des erweiterten Simulationsmodells als Eingangsgröße des Fahrermodells verwendet. Der durch die Simulation berechnete Fahrpedalverlauf entsprechend Abbildung 4, dient wiederum als Sollwertverlauf des Fahrroboters. Dabei wird der Rollenprüfstand im ersten Schritt in Lastsimulation betrieben. In einem weiteren Schritt erfolgt dann der Übertrag unter Verwendung der v- α Regelung durch Vorgabe der Geschwindigkeit aus den Simulationsdaten. Das Analyse Kriterium stellt im ersten Schritt die gemessene Fahrzeuggeschwindigkeit hinsichtlich ihrer Abweichungen gegenüber dem durch die Simulation berechneten Verlauf dar. Im zweiten Schritt wird zusätzlich der Verlauf der Antriebsarbeit als resultierende Größe betrachtet.

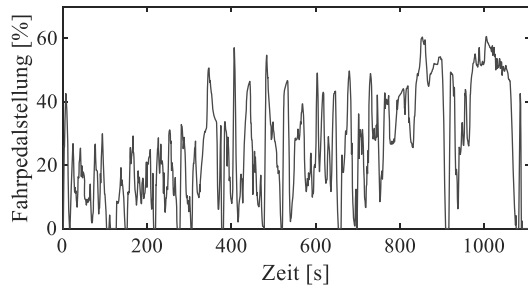


Abbildung 4: Fahrpedalverlauf als Eingangsgröße der Validierungsumgebung, Manöver 4.

2 Ergebnisse

In den folgenden Abschnitten werden die wichtigsten Ergebnisse der Validierungsprozesse zur Parameterbestimmung und Modellvalidierung anhand der vorgestellten Fahrmanöver präsentiert. Es wird dabei zunächst auf das Manöver der Parameterbestimmung und anschließend auf die Validierung des Simulationsmodells eingegangen.

2.1 Modellparameter

Abbildung 5 zeigt das anhand des Fahrmanövers 1 bestimmte Drehmomentkennfeld. Die Drehmomentwerte spiegeln dabei das effektive Drehmoment an der Kurbelwelle des Motors wider. Der Anstieg des Nullniveaus des Motordrehmoments bei höheren Drehzahlen reflektiert einerseits die quadratische Abhängigkeit des Motorschleppmoments von der Drehzahl in Folge von viskoser Reibung. Andererseits wirken sich im Bereich geringer Lasten auch die Charakteristik der Geschwindigkeitsregelung des Fahrroboters und der Kennlinie des Fahrpedals aus.

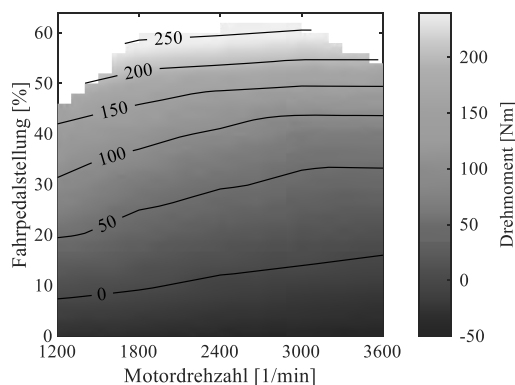


Abbildung 5: Rasterkennfeld des Motordrehmoments in Abhängigkeit der Motordrehzahl und der Fahrpedalstellung

2.2 Ergebnisse der Modellvalidierung

In diesem Abschnitt werden die Ergebnisse der Validierung des Simulationsmodells anhand der Testinterpretation der Fahrmanöver aus Abschnitt 1.5 vorgestellt. Die Charakterisierung von Abweichungen zwischen den Simulations- und Referenzdaten erfolgt anhand einer Regressionsgerade zur Beschreibung des Zusammenhangs zwischen den Datensätzen. Diesen liegt dabei ein gemeinsamer Zeitvektor als Bezugsgröße zugrunde. Die Bewertung der quantitativen Abweichungen erfolgt anhand von Streudiagrammen.

Die Testinterpretation von Fahrmanöver 2 hinsichtlich der Abbildung der Zugkraft zeigt Abbildung 6. Dabei zeigen sich bereits beim Vergleich der Verläufe große Abweichungen bei negativen Werten der Zugkraft in Folge der stark vereinfachten Modellierung des Bremssystems. Da die Verzögerung des Fahrzeugs bei verbrennungsmotorischen Antrieben jedoch nicht durch das Fahrpedal realisiert wird, sind diese Bereiche für die vorliegende Untersuchung nicht relevant und werden daher nicht berücksichtigt.

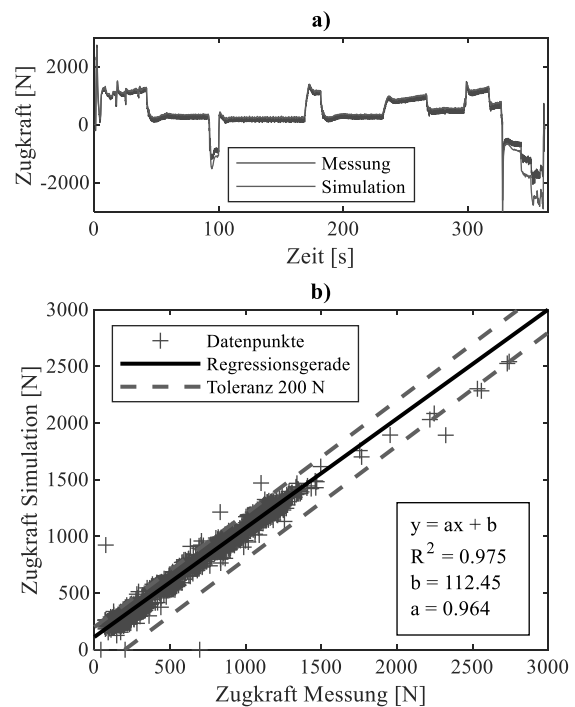


Abbildung 6: Testinterpretation Manöver 2.

- a) Vergleich der Zugkraftverläufe,
b) Streudiagramm

In den relevanten Bereichen zeigt sich dagegen ein stark linearer Zusammenhang zwischen den Datensätzen. Dies spiegelt sich sowohl in der Verteilung der Datenpunkte

als auch in der Steigung der Regressionsgeraden wider. Es treten jedoch konstante Abweichungen auf, die mit Toleranzen bei der Lastanpassung zur Justage der simulierten Fahrwiderstände am Rollenprüfstand zusammenhängen. Trotz der großen Spannweite der Schwankungen des Messsignals von 100 N, befindet sich nur ein sehr geringer Anteil der Datenpunkte außerhalb einer Toleranz von 200 N. Dies kommt auch durch den hohen Wert des Bestimmtheitsmaß der Regressionsgerade zum Ausdruck.

In Bezug auf die Fahrpedalstellung treten die größten Abweichungen der Simulation gegenüber dem realen Fahrzeugverhalten im Anfahrvorgang auf. Dies zeigt sich innerhalb der Testinterpretation des dritten Fahrmanövers entsprechend Abbildung 7 insbesondere bei der Betrachtung des Streudiagramms im Bereich von $\alpha < 10\%$.

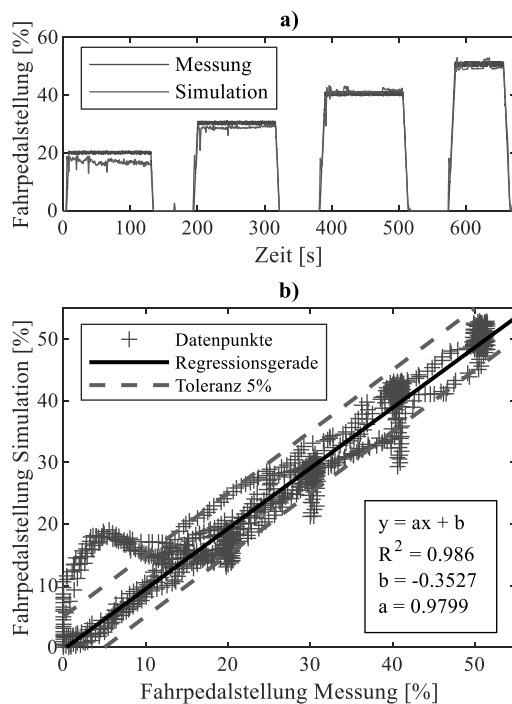


Abbildung 7: Testinterpretation Manöver 3.

- a) Verläufe der Fahrpedalstellung,
b) Streudiagramm der Fahrpedalstellung

Die Abweichungen in diesem Bereich sind auf den stark vereinfacht modellierten Anfahrvorgang zurückzuführen. Dieser Fahrzustand wird beim realen Fahrzeug durch ein elektronisches System gesteuert, sodass direkt nach dem initialen Lösen der Bremse ein Drehmoment am Rad erzeugt wird. Innerhalb des Simulationsmodells wird der Zustand durch die Fahrpedalstellung eingestellt. In den

Bereichen oberhalb von $\alpha = 10\%$ werden dagegen Toleranzen von 5 % größtenteils eingehalten. Die außerhalb liegenden Datenpunkte bei den definierten Sollwerten des Fahrmanövers sind auf geringe Unterschiede des dynamischen Verhaltens bei der Änderung des Fahrpedals zurückzuführen, die aufgrund der hohen Gesamtdynamik zu großen absoluten Abweichungen führen.

Bei der Durchführung des Fahrmanövers 4 in Lastsimulation wirken sich die zuvor identifizierten Ungenauigkeiten bei der Abbildung des Fahrzeugverhaltens aus, sodass das durch die Simulation berechnete Geschwindigkeitsprofil entsprechend Abbildung 8 nicht exakt reproduziert wird. Trotz der hohen Geschwindigkeiten und der hohen Dynamik, die bei dem Fahrmanöver auftreten, beträgt die maximale Abweichung 6,5 km/h bei einer Fahrzeuggeschwindigkeit von 80 km/h.

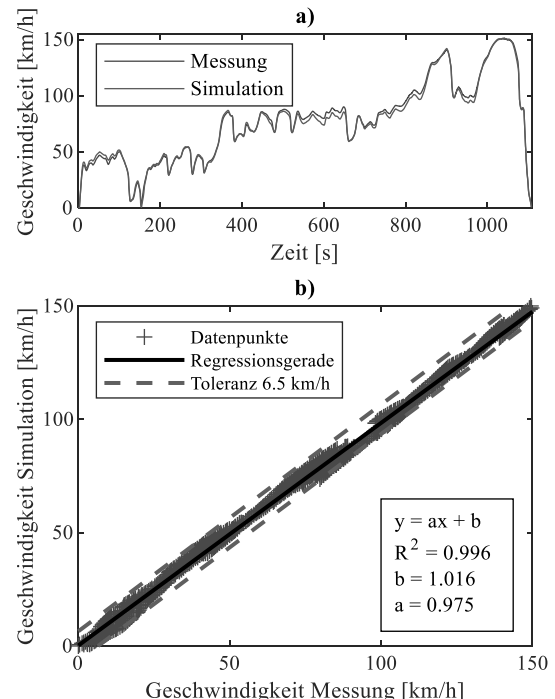


Abbildung 8: Ergebnisse Testinterpretation Manöver 4.

- a) Vergleich der Geschwindigkeitsverläufe,
b) Streudiagramm der Geschwindigkeit

Zum Nachweis der Übertragbarkeit der gewonnenen Erkenntnisse auf die Anwendung alternativer Regelarten wird jeweils die verrichtete Antriebsarbeit bei Anwendung der Lastsimulation (LS) und der v- α Regelung zur Durchführung des vierten Fahrmanövers betrachtet. Die Ergebnisse sind in Abbildung 9 dargestellt. Es zeigen sich dabei quantitative Unterschiede in den Verläufen, die auf eine höhere abgegebene Leistung in Lastsimula-

tion hindeuten. Dies ist eine Folge unterschiedlicher Betriebspunkte des Motors in Folge von unterschiedlichen Schaltvorgängen. Insgesamt nimmt jedoch die relative Abweichung der verrichteten Antriebsarbeit mit 0,13 % einen sehr kleinen Wert an.

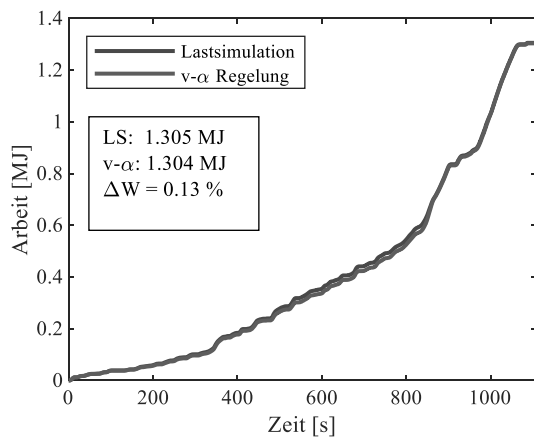


Abbildung 9: Ergebnisse Testinterpretation Manöver 4: Verrichtete Antriebsarbeit in Lastsimulation (LS) und v- α Regelung

3 Fazit und Ausblick

Anhand dieses Beitrags soll die Hypothese überprüft werden, dass bereits Simulationsdaten eines einfachen Längsdynamikmodells ausreichend sind, um einen Rollenprüfstand mit alternativen Regelarten zu betreiben. Das Ziel dabei ist, generische Fahrmanöver auf den Rollenprüfstand übertragen zu können, ohne die Notwendigkeit der Durchführung einer aufwändigen Justage der durch den Rollenprüfstand simulierten Fahrwiderstände.

Das verwendete Fahrzeugmodell in Kombination mit den ermittelten Parametern kann prinzipiell zur Abbildung des makroskopischen Fahrzeugverhaltens bei umfangreichen Fahrmanövern mit hoher Dynamik eingesetzt werden. Es ist jedoch aufgrund der simplen Struktur und der groben Parametrisierung mit entsprechenden Ungenauigkeiten zu rechnen. Beispielsweise erfordert die exakte Abbildung des Anfahrvorgangs eine höhere Modellierungstiefe bezüglich des betreffenden Regelsystems sowie des Bremssystems. Ähnliches gilt hinsichtlich der Modellierung des Motors und Getriebes zur Gewährleistung einer exakten Abbildung der Antriebskräfte in allen relevanten Geschwindigkeitsbereichen. Wird diese Genauigkeit jedoch nicht explizit gefordert, sind die Simulationsergebnisse durchaus ausreichend zum Übertrag generischer Fahrmanöver in der v- α Regelung. Da dies die einzige untersuchte Regelart darstellt, kann

die Hypothese nur in diesem Zusammenhang bestätigt werden. Für weitere Regelarten wie n- α sind daher separate Untersuchungen notwendig.

Acknowledgement

Das dieser Veröffentlichung zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung und den Ländern im Rahmen des Programms „Innovative Hochschule“ (IHS) unter dem Förderkennzeichen 03IHS024B gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor. Die Autoren bedanken sich bei dem Team der AIP GmbH & Co. KG für ihren Beitrag zu der im Rahmen dieser Papers veröffentlichten Arbeit.

Literatur

- [1] Tschöke, H. Hintergrund und Motivation In: Tschöke H (Hsg.). *Real Driving Emissions (RDE)*. Wiesbaden: Springer Vieweg; 2019. S 1-15.
- [2] Roberts P J, Mumby R, Mason A, Redford-Knight L, Kaur P. RDE Plus - The Development of a Road, Rig and Engine-in-the-Loop Test Methodology for Real Driving Emissions Compliance. *SAE Technical Paper*. SAE International; 2019. 12 S. doi: 10.4271/2019-01-0756
- [3] Albers A, Behrendt M, Klingler S, Matros K. Verifikation und Validierung im Produktentstehungsprozess. In: Lindemann U (Hsg.). *Handbuch Produktentwicklung*. München: Hanser; 2016. S. 541-569
- [4] Düser T. *X-in-the-Loop – ein durchgängiges Validierungsframework für die Fahrzeugentwicklung am Beispiel von Antriebsstrangfunktionen und Fahrerassistenzsystemen* [Dissertation]. Karlsruher Institut für Technologie (KIT); 2010.
- [5] Paulweber M, Lebert K. *Mess- und Prüfstandstechnik*. Wiesbaden: Springer Fachmedien Wiesbaden; 2014. 430 S.
- [6] Cortès S, Dettmann C, Heubach A, Schneider N, Boche P. Impact of control types of a chassis dynamometer on the reproduction of real world driving scenarios. *FISITA World 2021 Congress*; 2021 Sept; Prag.
- [7] Matros K, Schille F, Behrendt M, Holzer H. Manöverbasierte Validierung von Hybridantrieben. *ATZ*. 2015; 117: 64-71. doi: 10.1007/s35148-015-0384-z
- [8] Kraft CP, Staffen M. Fahrroboter ohne Anlernphase für den Rollenprüfstand. *ATZ Extra*. 2011; 16: 42-46. doi: 10.1365/s35778-011-0570-x
- [9] Maha-AIP GmbH & Co. KG. *The Road in Your Lab...*

Zukunftsweisende Prüfsysteme für reproduzierbare Straßenfahrt-Simulationen. Haldenwang: 2016. 52 S.

- [10] DEDITEC GmbH. *Datenblatt RO-CAN-Serie.* 2021
- [11] Kötter H, Sequenz H. Stationäre Motorvermessung mit verschiedenen Methoden und Modellen. In: Isermann R (Hsg.). *Elektronisches Management motorischer Fahrzeugantriebe.* 1. Auflage. Wiesbaden: Vieweg + Teubner; 2010. S. 130 – 162.
- [12] Kistler Group. *WPT Sensoren – Radimpulsgeber.* 2017.

Robustheitsanalyse einer Längsdynamiksimulation auf Basis verschiedener Qualitätskriterien

M. Eng. Sven Cortès ^{1*}, Prof. Dr.-Ing. Christian Dettmann ¹,

B. Eng. Julian Heib ¹, B. Eng. Felix Millberg ¹

¹Technische Hochschule Ulm, Prittwitzstraße 10, 89075 Ulm, Germany; *Sven.Cortes@thu.de

Abstract. Der Einsatz von simulationsbasierten Entwicklungsansätzen hat durch den rasanten Anstieg gesetzlicher Vorgaben und Fahrzeugfunktionen stark zugenommen. Um das Potential der virtuellen Antriebsstrangentwicklung zu demonstrieren, wird eine Längsdynamiksimulation am Beispiel des vollelektrischen BMW i3 modelliert. Zur Validierung der Fahrzeugmodellierung wird ein Manöverkatalog entworfen, der als Grundlage für diverse Referenzfahrten auf einem Rollenprüfstand dient. Die geschwindigkeitsabhängige Nullmomenten-Fahrpedalstellung ist der Kern der Einpedalstrategie („One-Pedal-Driving“) und die zu analysierende Trennlinie zwischen Zug- und Schubbetrieb des Versuchsfahrzeuges. Nach der Validierung wird das bestehende Modell einer numerischen Robustheitsbewertung, in Form von verschiedenen Beladungen, unterzogen. Des Weiteren wird die Genauigkeit auf Grundlage einer statistischen Bewertung, wie auch Grenzen der Simulation, untersucht. Zum Abschluss wird ein Ausblick auf eine mögliche Erweiterung der Modellierung geworfen.

Schlüsselwörter: Längsdynamiksimulation, Manöverkatalog, One-Pedal-Driving, Rollenprüfstand, Robustheitsanalyse

Einleitung

Das vom Europäischen Parlament verabschiedete Emissionsziel sieht vor, bis 2030 eine Absenkung der CO₂-Emissionen für Neufahrzeuge um 37,5 % gegenüber 2021 zu erreichen. [1, S. 22] Diese Situation hat die Entwicklung von alternativen Antrieben und die Elektrifizierung vorangetrieben.

Zugleich steht die elektrische Antriebsstrangentwicklung in der Automobilindustrie vor großen Herausforderungen. Hervorgerufen werden diese durch

zunehmend kürzere Entwicklungszyklen und eine gestiegene Derivateanzahl. Parallel dazu erschweren die, mit der Abgasnorm Euro 6c eingeführten, Real-Driving-Emissions-Tests (RDE) den Entwicklungsprozess zusätzlich. [2, S. 13]

RDE erfordert eine grundlegend neue Form der Entwicklung. Es wird davon ausgegangen, dass es einen Wechsel von einer Kennfeldbereichs- hin zu einer Eventoptimierung gibt. Grund dafür ist nicht allein der erweiterte Kennfeldbereich, sondern besonders auch die transienten Events, wie Lastsprünge, Motorstarts oder aktivierte Nebenverbraucher. Um sich speziell auf solche Events in der Entwicklung vorzubereiten, muss ein Umdenken von einer zyklusbasierten hin zu einer manöverbasierten Vorgehensweise passieren. [2, S. 5] Speziell das manöverbasierte Testen führt zu frühzeitigen und über den Prozess hinweg konsistenten Aussagen über das Verbrauchs- und Reichweitenverhalten eines elektrischen Antriebsstrangs unter Realfahrbedingungen. [2, S. 15]

Um die Validierung und Absicherung von Entwicklungszielen in einem solch komplexen Umfeld zu gewährleisten, erfordert es den Einsatz simulationsgestützter und automatisierbarer Verfahren zur gezielten Variation der Testrandbedingungen. Die Verknüpfung eines virtuellen Fahrzeugmodells mit einer Prüfstandsregelung ist im Bereich der Antriebsstrangentwicklung am Institut für Produktentwicklung (IPEK) am Karlsruher Institut für Technologie (KIT) bereits etabliert. Der dort definierte X-in-the-Loop (XiL) Ansatz, der sich auf Antriebssysteme spezialisiert, beschreibt eine durchgängige Vorgehensweise der Entwicklung auf verschiedenen Systemebenen. [3] Dieses Vorgehen erlaubt eine Teilvalidierung des Gesamtsystems zu einem sehr frühen Zeitpunkt im

Produktentstehungsprozess. Es ist hierbei auch vom Pull-Prinzip der Validierung die Rede, bei der eine Parallelisierung der primären und sekundären Validierungsaktivitäten stattfindet. [4, S. 556]

Voraussetzung ist eine genaue Modellbildung der Gesamt- und Teilsystemen. Eine Verknüpfung mit Referenzmanövern und Umweltbedingungen ist durch den XiL-Ansatz gegeben. Mittels eines Validierungsziels, welches mit einem Validierungsbedarf einhergeht, werden entsprechende Kriterien und die dazugehörigen Manöver gebildet. [5] Durch manöverbasiertes Validieren der einzelnen Komponenten bzw. Teilsysteme lässt sich ein Minimum an Ressourcen- und Zeitaufwand erzielen.[6, S. 537-544]

1 Methodik

Im Rahmen dieser Arbeit am Automotive Center des Instituts für Fahrzeugsystemtechnik (IFS) an der Technischen Hochschule Ulm (THU) wird eine Längsdynamiksimulation eines elektrischen Antriebsstranges aufgebaut. Als Versuchsträger für die spätere Modellierung dient der vollelektrische BMW i3 (2014). Mit Hilfe dieses Fahrzeugs wird das Simulationsmodell validiert, indem Referenzfahrten auf einem Rollenprüfstand durchgeführt werden.

1.1 Modellierung des elektrischen Antriebsstranges

Bei dem im IFS bereitgestellten Versuchsfahrzeug, handelt es sich um ein Vorserienfahrzeug der ersten Modellreihe des BMW i3. In der Abbildung 1 ist der schematische Aufbau des elektrischen Antriebsstranges dargestellt.

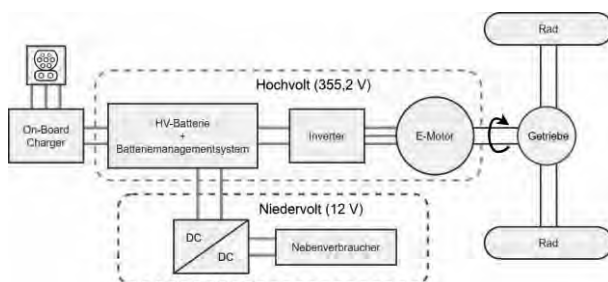


Abbildung 1: Schematische Darstellung eines elektrischen Antriebsstranges

Das Fahrzeug wird von einem Hybrid-Synchronmotor angetrieben, der über eine lang-

anhaltende Leistungskurve im Feldschwächebereich verfügt. Außerdem kann die elektrische Maschine als Generator fungieren. [7, S. 17] Die Hochvoltbatterie (HV-Batterie) ist aus insgesamt 96 Lithium-Ionen-Basiszellen aufgebaut, welche alle in Reihe geschaltet sind. Das integrierte Batteriemanagementsystem (BMS) regelt die Lade- und Entladeprozesse, wie auch die Betriebstemperatur der Zellen. [7, S. 19] Der Inverter, welcher ein Teil der Leistungselektronik darstellt, wandelt die Gleichspannung der HV-Batterie in eine Wechselspannung variabler Frequenz, die für den Elektromotor benötigt wird. Daneben ist das System auch in der Lage, die durch die Rekuperation gewonnene Energie zurück in die HV-Batterie zu speisen. Tabelle 1 fasst die technischen Daten des abzubildenden Fahrzeugs zusammen.

Versuchsfahrzeug BMW i3 (2014)	
Elektromotor	Hybrid-Synchronmotor
Nennleistung	75 kW bei 4800 U/min
Drehmoment	250 Nm bei 0 bis 4475 U/min
Getriebe	1-Gang Getriebe (9,665:1)
Batterietyp	Lithium-Ionen
Aufbau	8 Module mit je 12 Zellen (in Reihe) → 1p96s
Spannung	355,2 V
Kapazität	60 Ah
Reifen	Bridgestone Ecopia 155/70 R19
Fahrzeugmasse	1195 kg (Leergewicht)
Ausrollkoeffizienten	$F0 = 123,35 \text{ N}; F1 = 1,56 \frac{\text{N}}{\frac{\text{km}}{\text{h}}}$ $F2 = 0,015 \frac{\text{N}}{\frac{\text{km}^2}{\text{h}^2}}$

TABELLE 1: TECHNISCHE DATEN VERSUCHSFAHRZEUG [7, S. 43] [8]

1.2 Längsdynamiksimulation (Simulationsumgebung)

In der Simulationssoftware *GT-Suite* von Gamma Technologies wird ein Längsdynamikmodell aufgebaut. Das Modell (zu sehen in Abbildung 2) beinhaltet einen elektrischen Antrieb bestehend aus einer Elektrischen Maschine und der dazugehörigen Leistungselektronik. Zusammen mit der HV-Batterie und dem BMS bilden diese die elektrischen Triebstrangkomponenten. Zu den mechanischen Komponenten des Antriebsstranges zählt

die feste Übersetzung, die in Form eines eingängigen Getriebes ausgeführt ist, sowie das Differential. Abschließend finden sich die Räder mit der mechanischen Bremsanlage.

Es wird außerdem ein virtueller Fahrer in die Modellierung implementiert, dem ein beliebiger Geschwindigkeitsverlauf als Vorgabe hinterlegt wird. Der Baustein liefert ein Gaspedal- und Bremspedalwert als Ausgabe, mit denen das Fahrzeug angesteuert wird.

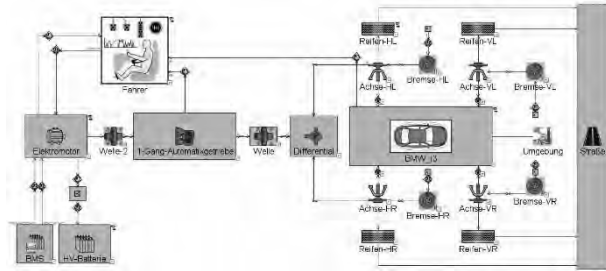


Abbildung 2: Topologie des Längsdynamikmodells in GT-Suite

Grundlage für die Parametrisierung des Längsdynamikmodells ist die Generierung von fahrzeugspezifischen Daten. Dies geschieht über das Programm *CANape*, mit dem die Datenbus-Größen aufgezeichnet werden.

1.3 Validierungsmethode

Zur Prüfung ob das Simulationsmodell auch bei der Entwicklung nicht spezifizierte Anforderungen erfüllt, wird eine manöverbasierte Validierungsmethode genutzt. Die Validierung beginnt aufgrund einer Systemanforderung (hier die Bewertung der Robustheit), wodurch ein Validierungsbedarf entsteht.

Mit der Validierung geht auch eine Validierungsumgebung einher. Diese fasst alle physischen und virtuellen Modelle zusammen, die innerhalb eines Unternehmens oder einer Abteilung zur Verfügung stehen. In Verbindung mit dem Validierungsbedarf werden ein oder mehrere Validierungsziele definiert. Anschließend wird für jedes Validierungsziel ein Test erstellt, der sich aus ein oder mehreren Testfällen, der Testumgebung und der Testinterpretation zusammensetzt. Die Aufgabe des Tests ist es Systemeigenschaften zu ermitteln und Erkenntnisse darüber zu liefern, ob das System die zuvor definierten Validierungsziele und objektiven Bewertungskriterien ganz, teilweise oder nicht erfüllt.

Die für die manöverbasierte Validierung notwendige

Testumgebung (Abbildung 3) fasst alle Mittel der Validierungsumgebung zusammen, die essenziell sind, um die Testfälle (Manöver) durchzuführen und das Systemverhalten zu erfassen. In diesem Fall beinhaltet die Testumgebung neben der virtuellen Längsdynamiksimulation einen zweiachsanangetriebenen Rollenprüfstand (RPS). Um die Fahrmanöver möglichst präzise und reproduzierbar umzusetzen, wird ein Fahrrobotersystem verwendet. In dem Simulationsprogramm *Driver's Aid* der Prüfstandsumgebung ist das jeweilige Fahrmanöver hinterlegt. Dieses liefert die Sollvorgabe der Geschwindigkeit an den Fahrroboter und an den *DynoServer*, welcher Signale mit der Rollenprüfstandsteuerung austauscht.

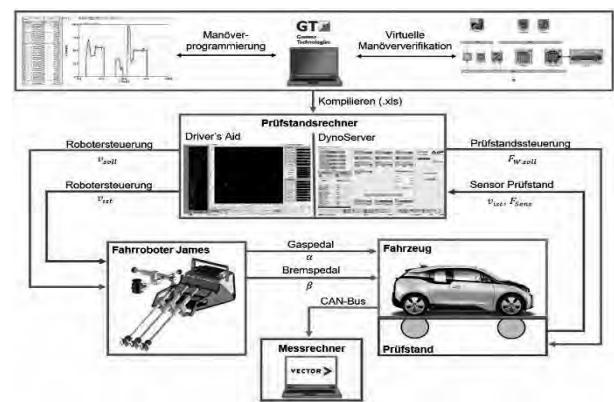


Abbildung 3: Testumgebung

Für die Durchführung der Referenzfahrten auf dem RPS stehen die Regelarten Lastsimulation und $v-\alpha$ zur Verfügung. Basis für die Lastsimulation stellt ein 1D-Punktmassmodell dar, welches innerhalb der Prüfstandsautomatisierung, passend zu jeder Geschwindigkeit, die entsprechenden Fahrwiderstände berechnet.

Diese umfassen Steigung-, Roll-, Luft- und Beschleunigungswiderstand und können approximiert mit einem Polynom 2. Grades beschrieben werden, wie in Gleichung (1) erkennbar. Die Koeffizienten F_0 , F_1 und F_2 beschreiben dabei die statischen, linearen und quadratischen Anteile der einzelnen Fahrwiderstände. Die Größen F_a und F_{St} bezeichnen die Beschleunigungs- und Steigungswiderstände.

$$F_Z = F_0 + F_1 \cdot v + F_2 \cdot v^2 + F_a + F_{St} \quad (1)$$

Die $v-\alpha$ -Regelung hingegen zeichnet sich durch zwei aktive Regelungen aus: in der Rolle, welche auf die Geschwindigkeit v regelt sowie die Vorgabe der Fahrpedalstellung α im Fahrzeug mittels Fahrroboter. Es stellt sich die entsprechende Zugkraft F_Z am Rad ein.

1.4 Manöverbasierte Validierung des Simulationsmodells

Nach ALBERS wird unter dem Begriff Validierung eine Überprüfung des Systems hinsichtlich dessen Eignung für seinen geplanten Einsatz verstanden. [4, S. 542] Die bei der manöverbasierten Validierung angewendeten Manöver bilden eine Referenzanwendung für die entsprechenden Validierungsziele ab.

Im Laufe einer Produktentwicklung werden neue Modellvarianten entwickelt. Eine neue Modellvariante basiert hierbei immer auf einem Referenzprodukt. In diesem Fall wird auf Basis vorliegender Daten eine Referenzmodellierung erstellt. Diese Referenzmodellierung befindet sich zunächst im Objektsystem. Das Zielsystem beschreibt den angestrebten Zielzustand, in diesem Fall entspricht es der Modellierung des Versuchsfahrzeuges in der Simulationsumgebung unter Berücksichtigung bestimmter Kriterien. Beim Durchlauf der Validierung wird eine Modellvariante analysiert und auf Abweichungen zum Zielsystem geprüft. Eine Analyse endet in der Erkenntnis über die Eigenschaften der aktuellen Modellvariante. Umgekehrt beschreibt die Synthese die Erstellung einer verbesserten Modellvariante auf Basis der zuvor gewonnenen Erkenntnisse. Im Handlungssystem befinden sich die zur Verfügung stehenden Ressourcen, Aktivitäten und Methoden. Der Zyklus beginnt nun von vorn auf Basis der modifizierten Modellierung.

1.5 Manöverkatalog

Für jedes Validierungsziel wird ein Test erstellt, der sich aus ein oder mehreren Testfällen (Manöver), zusammensetzt. Der Testfall stellt eine Referenzanwendung der zu validierenden Funktion dar. In einem sog. Manöverkatalog werden die Manöver sorgfältig definiert (siehe Tabelle 2). Ein Manöver ist dann vollständig beschrieben, wenn darin die Systemzustände, Systemeingaben und Umweltbedingungen festgelegt sind. [3, S. 69]

Aufgrund der exakten Definition der Manöver hinsichtlich Fahrereingaben und Randbedingungen ist eine Reproduzierbarkeit der Messungen gegeben.

1.6 One-Pedal-Driving

Der Antrieb ist mit dem als One-Pedal-Driving (OPD) bezeichneten Fahrerlebnis ausgestattet. Die Elektrische Maschine wechselt automatisch in den Rekuperationsmodus, wenn der Fahrpedalwinkel unter eine gewisse Schwelle fällt. Dabei generiert die elektrische Maschine ein negatives Bremsmoment, welches Strom zurück in die HV-Batterie speist. Die Fahrpedalprogression des OPD ist in folgende drei Stufen aufgeteilt [9]:

- **Verzögern:** Rekuperation bis zum Stillstand
- **Segeln:** Momentenneutrales Rollen
- **Beschleunigen/Konstantfahrt:** Momentanforderung an den Elektromotor.

Die beschriebene Fahrpedalprogression mit den drei Stufen ist in Abbildung 4 grafisch dargestellt.

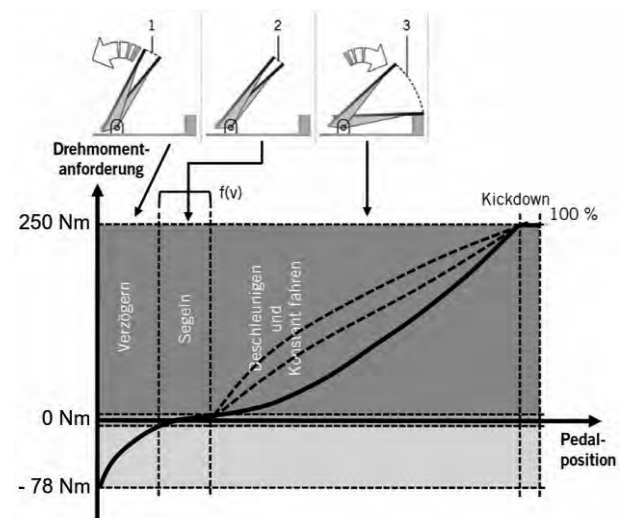


Abbildung 4: Fahrpedalprogression des One-Pedal-Driving mit den drei Stufen [9]

Technische Funktion	Ziel der Validierung	Beschreibung Manöver	Eingabe Fahrer	Bedingungen für Fahrzeug und Umwelt	Analysekriterien	Bewertungskriterien
One-Pedal-Driving	Abbildung geschwindigkeitsabhängige Nullmomentenlage	Konstantfahrt mit verschiedenen Geschwindigkeiten $v = 10:10:100 \text{ km/h}$	Fahrpedalstellungen $\alpha = 20:1:35 \%$	HV-Batterie $\text{SOC} > 80 \%$ und $T_{\text{Umgebung}} = 20 \text{ }^{\circ}\text{C}$	Mittlere Drehmomentenabweichung der elektrischen Maschine $-0,5 \leq \bar{m}_M \leq 0,5 \text{ [Nm]}$	Momentenverlauf um das Nullmoment

TABELLE 2: MANÖVERKATALOG FÜR DAS VALIDIERUNGSZIEL ONE-PEDAL-DRIVING

Der in Abbildung 4 gezeigte Momentenverlauf gilt nur für eine feste Geschwindigkeit. Das Fahrpedal-Drehmoment-Drehzahl Kennfeld berücksichtigt die Abhängigkeit der Geschwindigkeit auf die Kurve. Demnach leitet sich aus der geschwindigkeitsabhängigen Nullmomentenlinie der Validierungsbedarf und eine Hypothese mit der Fragestellung, ob mit einer höheren Fahrgeschwindigkeit ein höherer Nullmomenten-Fahrpedalwert α_{M0} korreliert.

1.7 Validierung am Beispiel One-Pedal-Driving

Bei der nachfolgenden Analyse und Validierung des OPD wird ein besonderer Fokus auf den Segelbereich und die charakteristischen Nullmomentenlinien gesetzt. Das dafür genutzte Manöver ist im Manöverkatalog (Tabelle 2) definiert. Untersucht wird der Geschwindigkeitsbereich von 10 km/h bis 100 km/h mit einer Schrittweite von 10 km/h. Zur Ermittlung des, für die jeweilige konstante Geschwindigkeit passende, Nullmomentenpunktes α_{M0} dient die v - α -Regelart. Eine konstante Geschwindigkeitsvorgabe durch den RPS wird mit einem gerasterten Fahrpedalwerte kombiniert.

Voruntersuchungen grenzen den Bereich für die Nullmomente zwischen Fahrpedalstellungen von 20 % und 35 % ein. Abbildung 5 zeigt beispielhaft die Geschwindigkeits- und Fahrpedalvorgabe für 50 km/h. Alle Geschwindigkeitsplateaus nutzen die gleiche dargestellte Fahrpedalvorgabe.

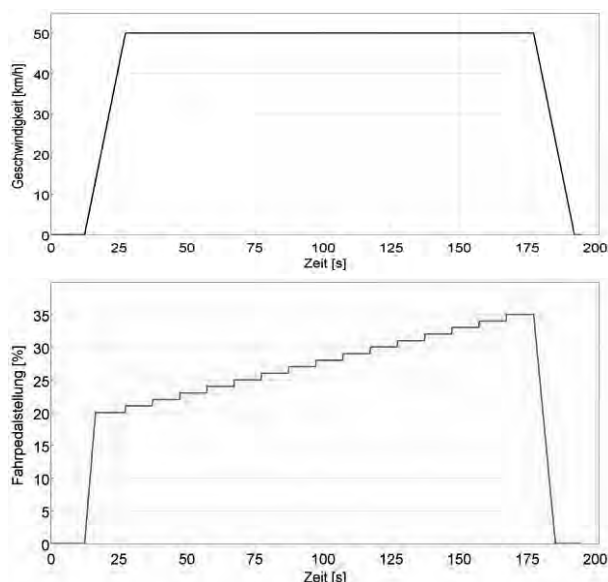


Abbildung 5: Geschwindigkeitsvorgabe für 50 km/h [oben] und Fahrpedalvorgabe [unten]

1.8 Testfälle

Das Versuchsfahrzeug unter Normalbedingungen, also ohne Zuladung und mit geladener HV-Batterie (SOC > 80 %), fungiert als Referenz. Auf dieser Grundlage wird der Segelbereich des OPD für den betrachteten Geschwindigkeitsbereich von 100 km/h bis 10 km/h validiert. Die Robustheit der Längsdynamiksimulation wird durch eine Zuladung in den Grenzen der Fahrzeugzulassung analysiert. Daraus folgen:

- Testfall 1: Referenz mit Leergewicht
- Testfall 2: Leergewicht plus 300 kg Beladung

Die Testfälle umfassen ein eigenes Manöver, (Regelart RPS: Lastsimulation), mit dem Zweck momentenneutral auszurollen. Es besteht aus einer Beschleunigungsphase auf 100 km/h mit anschließender Segelphase (Nullmoment). Dabei wird der Fahrpedalwert für die Nullmomentenlinie α_{M0} über der Geschwindigkeit nachgeregelt. In Abbildung 6 liegt α_{M0} zwischen den ermittelten 30 % und 25 %. Als Ausgangsgrößen ergibt sich das an der elektrischen Maschine wirkende Motormoment, sowie der Geschwindigkeitsverlauf des Segelvorgangs. Betrachtet wird die sich ergebende geschwindigkeitsabhängige Nullmomentenlinie und dessen Einfluss auf die verschiedenen Testfälle.

Der in Abbildung 6 dargestellte Inputverlauf gilt für beide Testfälle.

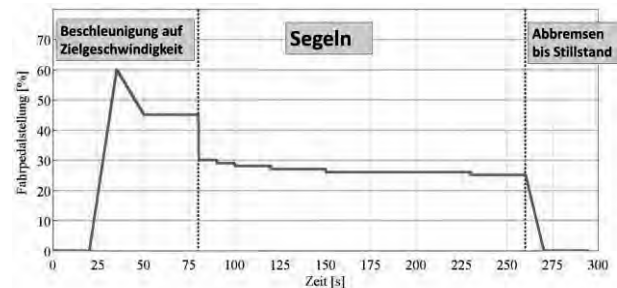


Abbildung 6: Fahrpedalvorgabe für Testfall 1 und 2

Die Daten aus den Manövern füllen das Fahrpedal-Drehmoment-Drehzahl-Kennfeld der Elektrischen Maschine in der Längsdynamiksimulation. Im Rahmen des Handlungssystems nähert sich das Objektsystem, die Längsdynamiksimulation, dem Zielsystem an. Auf Basis dieses Kennfeldes werden die Manöver, die zuvor auf dem RPS zur Generierung der Messdaten genutzt wurden, in der Simulation nachgefahren. Als Vorgabe dient das in Abbildung 5 dargestellte Manöver.

2 Ergebnisse

Der folgende Abschnitt befasst sich mit den Ergebnissen der manöverbasierten Validierung. Nachfolgend wird sich im Detail dem Validierungsziel des OPD gewidmet. Die Ergebnisse aus der Simulation werden im ersten Schritt qualitativ und danach quantitativ ausgewertet.

Abbildung 7 zeigt für den Geschwindigkeitsbereich von 10 km/h bis 100 km/h die detaillierten Nullmomenten-Fahrpedalwerte α_{M0} . Der Momentenverlauf für unterschiedliche Gaspedalstellungen bei einer Drehzahl von 2000 U/min (gesamter Drehmomentenbereich abgebildet) in Abbildung 7 unten dargestellt. Es ist zu erkennen, dass die Nullmomenten-Fahrpedalwerte α_{M0} zum einen mit höheren Geschwindigkeiten ansteigen. Zum anderen ist der Segelbereich bei höheren Geschwindigkeiten ausgeprägter. Damit stützt Abbildung 7 die zuvor aufgestellte Hypothese, dass der Nullmomenten-Fahrpedalwert mit der Geschwindigkeit steigt. Eine neue Erkenntnis ist jedoch, dass sich gleichzeitig auch der Segelbereich kontinuierlich vergrößert, was anhand der kegelartigen Hilfslinien verdeutlicht wird.

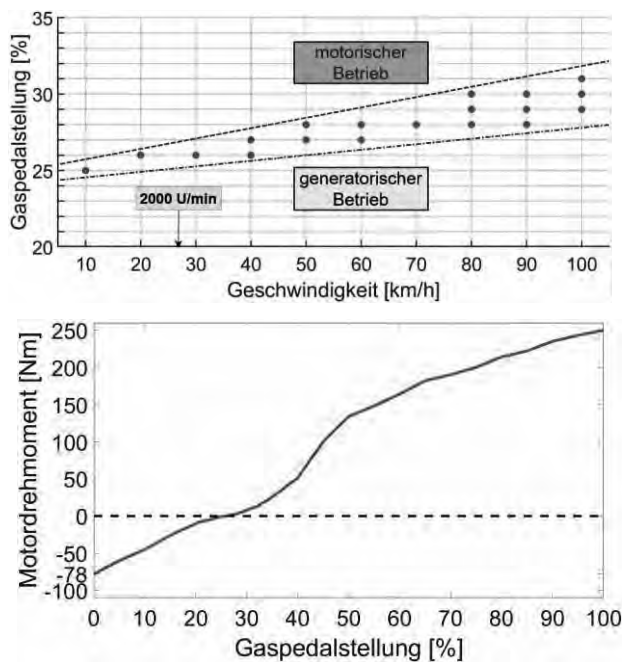


Abbildung 7: Nullmomenten-Fahrpedalkennlinie $\alpha_{M0}(v)$ [oben] und Momentenverlauf für 2000 U/min [unten]

Die beiden Drehmomentenverläufe aus der Simulation und den real gefahrenen Versuchen auf dem

RPS, für das in Abbildung 5 gezeigte Manöver (50 km/h), sind in Abbildung 8 zu sehen.

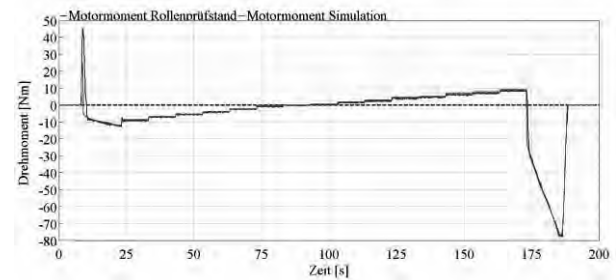


Abbildung 8: Motormoment um die Nullmomenten-Linie [50 km/h, siehe Manöverkatalog]

Der aus den Voruntersuchungen eingegrenzte Fahrpedalbereich bestätigt sich. Nach 83 Sekunden segelt das Referenzfahrzeug 20 Sekunden lang. Dem Fenster in Abbildung 7 sind die beiden zugehörigen α_{M0} -Werte bei 27 % und 28 % entnehmbar. Des Weiteren beweisen die annähernd übereinander liegenden Verläufe vor und nach dem Geschwindigkeitsplateau ($23 \text{ s} < t_{v=\text{konst}} < 173 \text{ s}$) die Genauigkeit des Kennfeldes, welches aus den Voruntersuchungen entstanden ist.

2.1 Statistische Untersuchung der Ergebnisse

Eine höhere Aussagekraft über den Modellierungsstand erzielen verschiedene statistische Methoden. Diese prüfen, ob das Validierungsziel des OPD erreicht ist.

Als Grundlage für die statistischen Methoden dient die Differenzfunktion $d(t)$ zwischen den Rollenprüfstandsverläufen (RPS) und Simulationsverläufen (SIM).

$$d(t) = x_{RPS}(t) - x_{SIM}(t) \quad (2)$$

Mit Gleichung (3) lässt sich der Mittelwert der zuvor bestimmten Differenzfunktion berechnen. Hierbei steht N für die Anzahl der Messdaten.

$$\bar{m} = \frac{1}{N} * \sum_{t=1}^N d(t) \quad (3)$$

Auswerten lässt sich der Bereich mit konstanter Geschwindigkeit aus Abbildung 8. Es wird nur das Geschwindigkeitsplateau betrachtet und statistisch ausgewertet. Abbildung 9 zeigt den zugehörigen Drehmomentenverlauf. Jeder Fahrpedalsprung resultiert in einem Sprung der Drehmomentenanforderung (für eine gleichbleibende Geschwindigkeit).

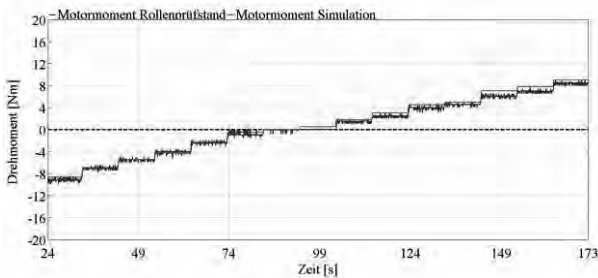


Abbildung 9: Drehmoment für Zeitbereich mit konst. Geschwindigkeit [50 km/h; Ausschnitt]

v [km/h]	\bar{m}_α [%]	\bar{m}_M [Nm]	α_{M0} [%]
10	0,0536	-0,0480	25
20	0,0607	-0,0297	26
30	0,0466	-0,4000	26
40	0,0413	-0,0992	26; 27
50	0,0215	-0,3378	27; 28
60	0,0439	0,1058	27; 28
70	0,0401	-0,0316	28; 29
80	0,0385	0,0618	28; 29; 30
90	0,0155	-0,0760	28; 29; 30
100	0,0329	0,0338	29; 30; 31

TABELLE 3: STATISTISCHE AUSWERTUNG DES VALIDIERUNGSZIELS

Die Tabelle 3 fasst die statistischen Ergebnisse aus den in Abschnitt 1.7 beschriebenen Manöver zusammen, wobei der Fokus der Auswertung auf der Nullmomentenlinie liegt.

Die zweite Spalte der Tabelle 3 zeigt die mittlere Abweichung des Fahrpedals (\bar{m}_α), welches als Inputgröße dient. Der Fahrroboter setzt über den gesamten Manöverzeitraum einen exakten Fahrpedalwert um, was die minimale Abweichung beweist.

Die Spalte \bar{m}_M zeigt ausgehend von der vorgegebenen Geschwindigkeit und dem Fahrpedalwert, die daraus resultierende mittlere Abweichung des Motormoments.

2.2 Testfall 1

Die Ergebnisse der Referenzfahrt, also für einen SOC > 80 % und ohne Zusatzbelastung, sind in Abbildung 10 für den Zeitbereich, in welchem das Fahrzeug segelt, aufgezeigt.

Wie beschrieben, zeichnet sich der Momentenverlauf der Simulation durch eine nachgeregeltere Fahrpedalstellung aus, mit dem Ziel über den betrachteten Zeitbereich zu segeln. Der Geschwindigkeitsverlauf zeigt das momentenneutrale Ausrollen von 100 km/h bis 10 km/h. Oben dargestellt sind die zeitabhängigen Fahrpedalstellungen und die 1%-Fahrpedalsprünge.

Der erste und der dritte Fahrpedalsprung äußern sich in einem Abfallen der Drehmomentenanforderung der Elektrischen Maschine. Der Zusammenhang zwischen negativem Fahrpedalsprung und geschwindigkeitsabhängiger negativer Momentenanforderung ist zu beobachten. Die zeit- und wertediskreten Drehmomente belegen die Auflösung des Fahrzeugbuses von 0,5 Nm. Der zweite und vierte Fahrpedalsprung hingegen sind nicht messbar.

Die Abweichungen zwischen RPS und Simulation von -2,0 Nm bis +2,5 Nm unterbrechen das Ausrollen nicht. Es tritt keine Konstantfahrt oder Beschleunigung auf. Erst im letzten Drittel steigt die mittlere Abweichung merklich an. Dieser Abschnitt startet mit einer generatorischen Anforderung, welche ab 171 s in eine steigende motorische Anforderung umschwingt. Bis zum Ende bleibt die Fahrpedalstellung bei einem Winkel von 26 %. Die untere Grenzggeschwindigkeit von 10 km/h wird bei 227,2 s unterschritten. Generell gilt, dass bei niedrigeren Geschwindigkeit kleinere Fahrpedalschwankungen mit höheren Drehmomentsprüngen verbunden sind um bei der Anfahrt dem hohen Trägheitsmoment entgegen zu wirken.

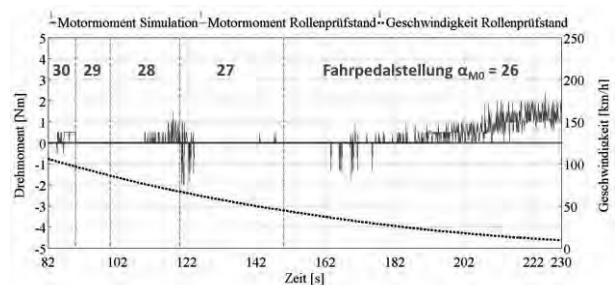


Abbildung 10: Momentenverlauf für Testfall 1 [Ausschnitt]

2.3 Testfall 2

Der zweite betrachtete Testfall dauert länger an, aufgrund der gestiegenen Trägheit des Referenzfahrzeuges. Von über 100 km/h bis unter 10 km/h vergehen 171,8 s. Im Vergleich dazu ist Testfall 1 deutlich kürzer mit 140,3 s. Dies hat zur Folge, dass das Diagramm den Fahrpedalsprung von 26 % auf 25 % bei 230 s beinhaltet.

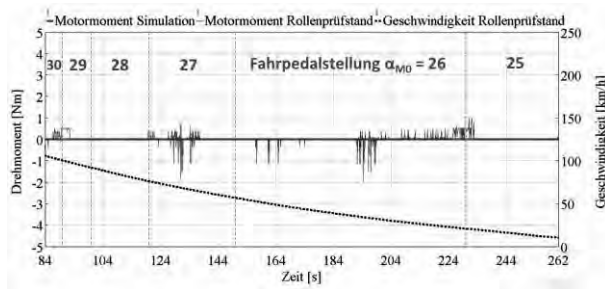


Abbildung 11: Momentenverlauf für Testfall 2
[Ausschnitt]

2.4 Auswertung der Testfälle

Die Ergebnisse der Ausschnitte beider Testfälle sind in Tabelle 4 aufgefasset. Neben der mittleren Abweichung des Motormoments ist auch die Standardabweichung aufgeführt.

Parameter	Testfall 1	Testfall 2
\bar{m}_M [Nm]	0,2544	0,1812
Standardabweichung s	0,4794	0,335

TABELLE 4: STATISTISCHE AUSWERTUNG DER TESTFÄLLE

3 Diskussion und Fazit

Die aufgestellte Hypothese, dass eine höhere Fahrgeschwindigkeit in einen höheren Nullmomenten-Fahrpedalwert α_{M0} resultiert, hat sich bestätigt.

Eine weitere Schlussfolgerung aus Abbildung 7 ist die Scharbildung an α_{M0} mit steigender Geschwindigkeit, welche eine geschwindigkeitsabhängige Hysterese darstellt. Es ist zu erwarten, dass bis zur maximalen Geschwindigkeit von 150 km/h wachsende Bereiche mit Nullmomenten das Segeln erheblich erleichtern. Dies liegt zum einen an dem, für Elektromotoren üblichen, Feldschwächebereich, in welchem das maximale Motordrehmoment bei Höchstdrehzahlen auf ein Drittel des Maximalwertes absinkt. Zum anderen ist das OPD auch eine Komfortfunktion, welche mit anderen

Funktionen, wie dem Fahrbarkeitsfilter, wechselwirkt.

Das Erfüllen des Analyse Kriterium aus dem Manöverkatalog beweist die Validierung des OPD auf Grundlage der statistischen Auswertung der Geschwindigkeiten (Tabelle 3). Nach ALBERS ist damit das Validierungsziel erfüllt. [4, S. 542] Die Längsdynamiksimulation ist damit geeignet, das OPD Verhalten im betrachteten Geschwindigkeitsbereich ausreichend abzubilden.

Die zwei betrachteten Testfälle beweisen die Robustheit der Simulation in Bezug auf Gewichtsva-riation. Der Rahmen der zulässigen Beladung des Testfahrzeuges wurde nicht verlassen. Damit zeigt sich, dass das Objektsystem unter Entwicklung von ZHO und manöverbasierter Validierung in den Grenzen der Fahrzeugzulassung robust ist.

Durch die manöverbasierte Validierung der Hypothese haben sich weitere Wissenslücken ergeben. Aus den Voruntersuchungen ist die höchste Steilheit Ausgleichsgerade der Nullmomenten-Punkte bei kleineren Geschwindigkeiten beobachtet worden. Unter Einfluss vom Anfahrregler und dem OPD-Kennfeld bildet sich ein weiterer Validierungsbedarf für das Modell.

Literatur

- [1] H. Zlojo, A. Kuznik, T. Steinhaus und C. Beidl, „Modellbasierte Hybridentwicklung - Verschmelzung von virtuellem und realem Test“, MTZextra, S. 22–25, 2020.
- [2] J. Liebl und G. Rainer, Hg., VPC.plus 2014: Simulation und Test für die Antriebsentwicklung. Wiesbaden: Springer Fachmedien Wiesbaden, 2018.
- [3] K. Matros, F. Schille, M. Behrendt und H. Holzer, „Manöverbasierte Validierung von Hybridantrieben“, ATZ - Automobiltechnische Zeitschrift, Jg. 117, S. 64–71, 2015.
- [4] A. Albers, M. Behrendt, S. Klingler und K. Matros, „Verifikation und Validierung im Produktentstehungsprozess“ in *Handbuch Produktentwicklung*, U. Lindemann, Hg., München: Carl Hanser Verlag GmbH & Co. KG, 2016, S. 541–569.
- [5] M. Geier, C. Stier, T. Düser, M. Behrendt, S. Ott und A. Albers, „Simulationsgestützte Methoden IDE und XiL zur Entwicklung von Antriebsstrangkomponenten“, ATZ - Automobiltechnische Zeitschrift, S. 48–53, 2009.

- [6] M. Bargende, H.-C. Reuss und A. Wagner, Hg.,
21. Internationales Stuttgarter Symposium:
Automobil- und Motorentechnik. Wiesbaden:
Springer Fachmedien Wiesbaden, 2021.
- [7] BMW PressClub, Der BMW i3, 2013. [Online].
Verfügbar unter:
[https://www.press.bmwgroup.com/switzerland/art
icle/detail/T0148286DE/der-bmw-
i3?language=de](https://www.press.bmwgroup.com/switzerland/article/detail/T0148286DE/der-bmw-i3?language=de)
- [8] Idaho National Laboratory, „2014 BMW i3:
Advanced Vehicle Testing“, 5. Okt. 2016.
[Online]. Verfügbar unter:
[https://avt.inl.gov/sites/default/files/pdf/fsev/fact2
014bmwi3ev.pdf](https://avt.inl.gov/sites/default/files/pdf/fsev/fact2014bmwi3ev.pdf). Zugriff am: 8. Juni 2021.
- [9] M. Jung, F. Kessler, P. Müller und S. Wahl,
„Fahrzeugintegration und Fahrverhalten des
BMW Active E“, ATZ - Automobiltechnische
Zeitschrift, S. 808–812, 2012.

Modellierung des Betriebsverhaltens eines Synchrongenerators als Stromerzeuger

Bert Henrik Herrmann¹, Maximilian Ringel¹, Daniel Jörss¹, Christian Fink^{1*}

¹. Fakultät für Maschinenbau, Hochschule Wismar, Philipp-Müller-Str. 14, 23966 Wismar, Germany

*christian.fink@hs-wismar.de

Abstract. Im Rahmen des BMBF-finanzierten Forschungsprojektes SIDYN soll das Betriebsverhalten eines Stromerzeugers in Inselnetzen mit dynamischen Netzlasten simulativ untersucht werden. Der präsentierte Entwicklungsstand beschreibt das mechanisch-elektrische Generator-Netz-Modell. Es werden die Herausforderungen bei der Modellierung und Parametrierung eines solchen Synchrongenerators beschrieben. Zur Validierung wurde ein Synchrongenerator beschafft, der zukünftig im Zusammenspiel mit einem Deutz TD 2.9 -Turbodieselmotor als Referenzprüfstand dienen soll.

Ziel der Entwicklung ist ein Simulationsmodell, das keine symmetrische Belastung voraussetzt und somit für Inselnetze typische Belastungszustände sicher simulieren kann. Das Modell soll möglichst flexibel erweiterbar sein, so dass es im Verlauf der Entwicklung angepasst werden kann. Entsprechende Anpassungen werden im Rahmen der Prüfstandsmessungen erwartet. Zur Gewährleistung einer bestmöglichen Anpassbarkeit wurde nicht auf bestehende physikalische Modelle der Simscape®-Umgebung aufgebaut, sondern es erfolgte eine komplette Neuentwicklung eines signalfussorientierten MATLAB Simulink®-Modells. Der Entwicklungsstand entspricht in etwa dem SI-Modell aus der MATLAB Simscape® Electrical Bibliothek. Demgemäß wurde dieses aufgrund fehlender Prüfstandsergebnisse für eine Model-by-Model-Validierung benutzt, deren Ergebnisse ebenfalls präsentiert werden.

Einleitung

Im Rahmen des BMBF-finanzierten Forschungsvorhabens SIDYN soll das Betriebsverhalten eines Stromerzeugers in Inselnetzen mit dynamischen Netzlasten auf Grundlage eines modularen Simulationsmodells abgebildet werden. Das soll die Möglichkeit zur Variation unterschiedlicher Motor- und Netzkonfigurationen sowie verschiedener Lastszenarien bieten.

Ein Inselnetz ist ein vom öffentlichen Stromnetz unabhängiges System aus Stromerzeugern (meist Synchrongeneratoren und Verbrauchern).

Zur Erweiterung des bereits vorgestellten Simulationsmodells eines Industrienotors [1] sollen hier die elektromechanischen und elektrischen Modellansätze beschrieben werden. Es geht vor allem darum, ein tieferes Verständnis zum dynamischen Verhalten eines Synchrongenerators mit angeschlossener Last aufzubauen und ein Systemmodell aufzustellen, das flexibel anpassbar ist. Der Aufbau und eine Anwendung eines solchen Modells setzt ein tiefgreifendes Verständnis über die physikalischen Zusammenhänge voraus. Verfügbare Modelle in MATLAB Simscape® zeigen eine plausible Funktion, jedoch sind die Modellbeschreibungen nicht vollumfänglich nachvollziehbar. Aus diesem Grund sowie der Forderung nach einer anpassbaren Modelltiefe für eine spätere Anwendung wurde der Aufbau eines MATLAB Simulink®-Modells favorisiert, welches das Generator- und Regelverhalten realitätsnah abbildet.

Generatorsysteme werden meist aus Rechenvorschriften von Motormodellen abgeleitet. Diese Modelle verhalten sich ähnlich, erfüllen aber nicht die Anforderung zur Berechnung eines benötigten Drehmoments aus einer bekannten elektrischen Last am Generator bei einer bestimmten Drehzahl unter Berücksichtigung einer geregelten Erregung.

Bezogen auf die Anwendung, soll ein Synchrongene-

rator in verschiedenen Belastungsszenarien simuliert werden, der direkt mit einem Verbrennungsmotor gekoppelt ist. Daraus folgen die Bedingungen, dass eine Drehzahl des Verbrennungsmotors unter Wirkung einer drehelastischen Kupplung zu jedem inkrementalen Zeitpunkt vorgegeben ist. Aus der elektrischen Last folgt im Generator ein notwendiges Drehmoment, das als Lastanforderung zum Verbrennungsmotor zurückgeführt wird.

Zur Validierung der entwickelten Modelle erfolgt ein Vergleich mit Simulationsergebnissen äquivalenter Modelle aus MATLAB Simscape® sowie im späteren Projektverlauf mit Prüfstandsmessungen.

1 Recherche nach Modellen

Vor der Entwicklung und Implementierung eines eigenen Modells erfolgte eine Recherche hinsichtlich verfügbarer Modelle in der zu benutzenden Modellierungs- und Simulationsumgebung MATLAB Simulink®. Der Fokus lag auf bestehenden Modellen, die die Möglichkeit bieten, bei definierter Winkelgeschwindigkeit und Sollspannung einen lastfallabhängigen Momentenverlauf zu bestimmen. In der MATLAB Simscape® Toolbox Electrical [2] sind solche Modelle vorhanden und beschrieben. Jedoch ist die Parametrierung nicht trivial und bedarf einiger Einarbeitung. Es ist die Wahl zwischen einem auf SI-Einheiten bzw. auf PU-Einheiten basierten Modell zu treffen. Zur Verifizierung des selbst erstellten Modells wurde sich für das SI-Einheitenmodell entschieden. Im Simscape®-Modell fehlen jedoch grundlegende Annahmen zu Signalarückkopplungen sowie die Einbeziehung der elektrischen Lasten. Demgemäß wurde eine eigene Modellentwicklung angestrebt, die eine flexible Anpassung des Modells im Rahmen des Projektverlaufs erlaubt.

Für erste Modellvalidierungen kann das SI-Einheitenmodell in Simscape® herangezogen werden. Als Anwendungsfall dient die Standardkonfiguration eines Generators mit 42,5 kVA.

2 Modellbeschreibung und Implementierung

Leitfäden zur Modellierung von Synchrongeneratoren können in der Literatur u.a. unter [2], [3], [5], [6], [9] und [10] gefunden werden. Es sei erwähnt, dass die Bezeichnung von Modellgrößen stark divergiert, weshalb eine Normierung wünschenswert wäre. Die hier beschriebene Modellierung basiert auf dem vom IEEE (Institute of

Electrical and Electronic Engineers) veröffentlichten Manuskript [3] sowie [2]. Hieraus geht hervor, dass unter Berücksichtigung einer Ersatzschaltung im d-q-Koordinatensystem die folgenden grundlegenden Gleichungen 1 und 2 für die Flussverkettung im rotorbezogenen Modell aufzustellen sind:

$$u_d = -i_d R_a - p \omega_{mech} \psi_q + \frac{d\psi_d}{dt} \quad (\text{Gl. 1})$$

$$u_q = -i_q R_a + p \omega_{mech} \psi_d + \frac{d\psi_q}{dt} \quad (\text{Gl. 2})$$

Für das elektromagnetische Moment gilt Gleichung 3. Somit kann das Moment T , welches der Generator im eingestellten Lastzustand zum Antrieb benötigt, ermittelt werden.

$$T = p(\psi_d i_q - \psi_q i_d) \quad (\text{Gl. 3})$$

Die grundlegenden Funktionen und Zusammenhänge sind in Abbildung 1 dargestellt. Wobei ausschließlich der innere Block der fremderregten Synchronmaschine in den Quellen ausführlich beschrieben ist.

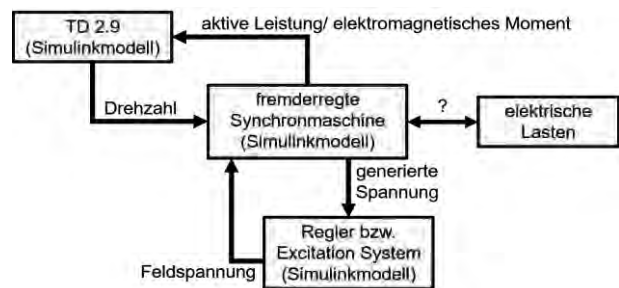


Abbildung 1: Darstellung der Subsysteme und Wechselwirkungen

Fraglich bleibt, welche Spannungen u_d und u_q bei gegebener Drehzahl und Feldspannung generiert werden und welche Ströme i_d und i_q sich daraus ergeben. Die Frage nach der Funktionsweise des Modells klärt sich erst mit der Einbindung einer elektrischen Last in das Modell, welche den Spannungsausgang des Generators mit einem Strom belastet und somit eine Leistung vom Generatorsystem fordert. Da die Lastauswirkung einfacher in zeitdiskreten Ursprungskoordinaten zu bewerten ist, soll die Lastrechnung nach der Rücktransformation aus dem d-q-Koordinatensystem erfolgen. Somit ergibt sich ein weiterer Rückkoppelzweig über die Last (siehe Abbildung 2).

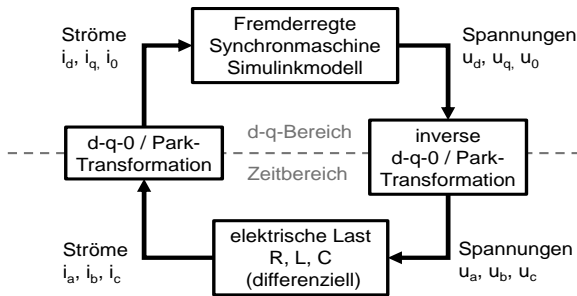


Abbildung 2: Darstellung der Lasteinbindung mit Hin- und Rücktransformation

Im Wesentlichen entspricht die Einbindung eines geeigneten Lastmodells der Modellierung des Betriebsverhaltens des Generatorsystems.

Das Hauptaugenmerk liegt auf der Betrachtung der Phasenverschiebungen der einzelnen Ströme i_a , i_b und i_c zu den jeweiligen Spannungen u_a , u_b und u_c . Zudem soll das Modell eine asymmetrische Belastung einer rein ohmschen bis hin zu induktiven und kapazitiven Lasten ermöglichen.

Ein geeigneter Ansatz besteht in der Eingabe der entsprechenden Leistungen, welche in Wirkleistungskomponenten (ohmscher Lastanteil) und Blindleistungskomponenten (kapazitive und induktive Lastanteile) unterteilt werden können. Die eingegebenen Leistungswerte werden vorab in äquivalente Widerstands-, Kapazitäts- bzw. Induktivitätswerte umgerechnet werden. Ein Ersatzschaltbild beschreibt diesen Ansatz in Abbildung 3 in Form eines Schwingkreises. Demnach stehen x_F , x_R und x_L für die äquivalenten Widerstandswerte der Feldleistung, der ohmschen Leistung sowie dem ohmschen Widerstand der induktiven Last. Der Parameter L steht für die Induktivität und C gibt die Kapazität wieder. Jeder Phase wird ein solcher Schwingkreis zugeordnet. $U_c(t)$ steht dabei für die entsprechenden Spannungsverläufe der 3 Phasen u_a , u_b und u_c .

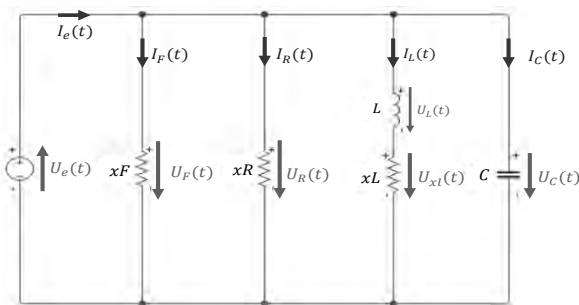


Abbildung 3: Last als Schaltungsäquivalent für einen RLC-Schaltkreis

Demzufolge wurde die Differenzialgleichung 4 entwickelt, die das dynamische Verhalten des oben abgebildeten RLC-Schwingkreises beschreibt und bei den eingestellten Lasten für jede Phase den zum Spannungsverlauf zugehörigen Stromverlauf angibt. Gleichung 4 wurde anschließend in ein MATLAB Simulink® Subsystem überführt.

$$I(t) = U(t) \left(\frac{1}{x_F} + \frac{1}{x_R} - \frac{CxL}{L} \right) + \frac{1}{L} \left(1 + \frac{xL}{x_F} + \frac{xL}{xR} \right) U(t) dt - \frac{xL}{L} \int I(t) dt + C \frac{dU(t)}{dt} \quad (\text{Gl. 4})$$

Soll eine Leistungsvorgabe erfolgen, müssen Funktionen zur Umrechnung der Parameter aus den Leistungswerten in äquivalente Widerstandswerte implementiert werden. Abbildung 4 zeigt diese Kopplung mit der notwendigen Last, um aus den Spannungsäquivalenten u_d , u_q und u_0 auf die resultierenden Ströme i_d , i_q und i_0 zu schließen. Mittels der inversen dq0-Transformation werden die Spannungen zuerst in zeitvariante Sinusspannungen u_a , u_b und u_c transformiert. Dann werden die Lastrechnungen im Statorsystem mittels der Differenzialgleichung 4 für jede Phase bestimmt, um daraufhin die resultierenden Ströme mit der dq0-Transformation wieder in das rotorbasierte Modellsystem zurückzuführen. Dieses Vorgehen ist notwendig, um sämtliche Phaseninformationen dem elektromechanischen Grundmodell zurück koppeln zu können.

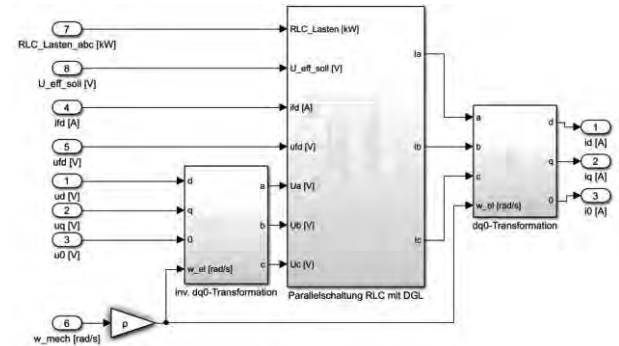


Abbildung 4: Umsetzung der Lasteinbindung in MATLAB Simulink®

In Abbildung 5 ist die gesamte Verknüpfung des Gleichungsansatzes zur Modellierung des Generatorsystems inklusive der elektrischen Last auf Basis des Strukturiagramms aus Abbildung 2 dargestellt. Zu erkennen ist, dass die Matrizendarstellung im dq0-System Parametrierungen voraussetzt, die im aktuellen Schleifendurchlauf berechnet werden müssen. Dadurch ergeben sich bei direkter Umsetzung in MATLAB Simulink® diverse algebraische Schleifen, die durch Feedback-Loops bzw. durch geeignete Wahl von Startwerten aufzulösen

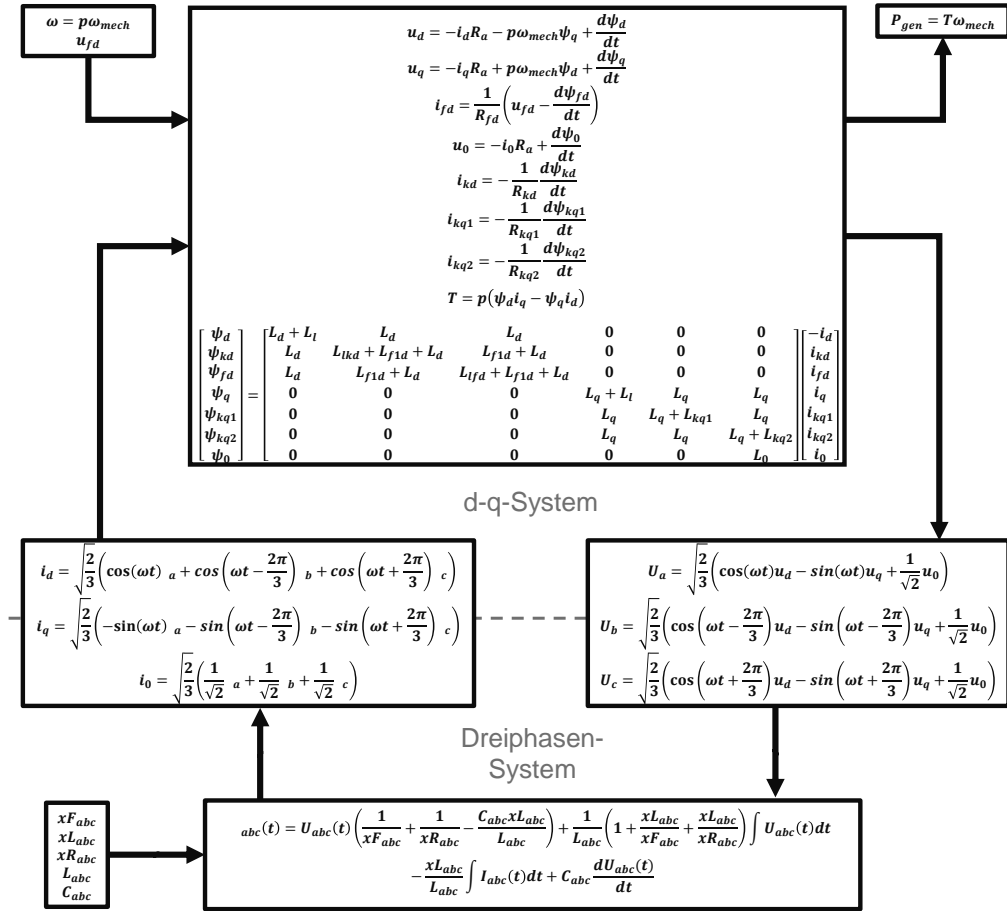


Abbildung 5: Zusammenstellung und Verknüpfung aller verwendeten Gleichungen

sind. Der bestmögliche Umgang mit dieser Anforderung ist Thema aktueller wissenschaftlicher Arbeiten, um ein stabil laufendes Simulationsmodell gewährleisten zu können. Neben den Gleichungen für die Ströme i_d und i_q , beinhaltet Abbildung 5 weitere Gleichungen zur Bestimmung des Erregerfeldstromes i_{fd} sowie der Ströme in den d- und q-Dämpferwicklungen i_{kd} , i_{kq1} und i_{kq2} . Weiterhin ist die Matrix für alle im Modell verwendeten Flussverkettungen dargestellt. Die Indizes abc im Dreiphasensystem, sollen darauf verweisen, dass diese Berechnungen für jede Phase separat durchgeführt werden.

Im Zusammenspiel mit einem geeigneten Intervall folgt ein numerisches Modell, welches Simulationen auf Basis des mathematischen Modells und eingestellten Lasten ermöglicht. Das Modell beschreibt in dieser Form die Modellvariante 2.2 aus dem IEEE 1110 – Design Guide [3] (siehe Abbildung 6).

Zur Vereinfachung auf das Model 2.1 nach [3], mit

nur einem Dämpferkreis, kann in der q-Achsen-Ersatzschaltung (unteres Schaltbild in Abbildung 6) der Zweig aus L_{kq2} und R_{kq2} weggelassen werden. Diese Anord-

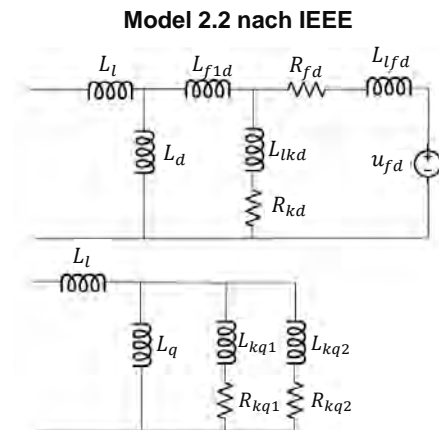


Abbildung 6: Darstellung der Modellvariante nach IEEE; Formelzeichen angepasst [3]

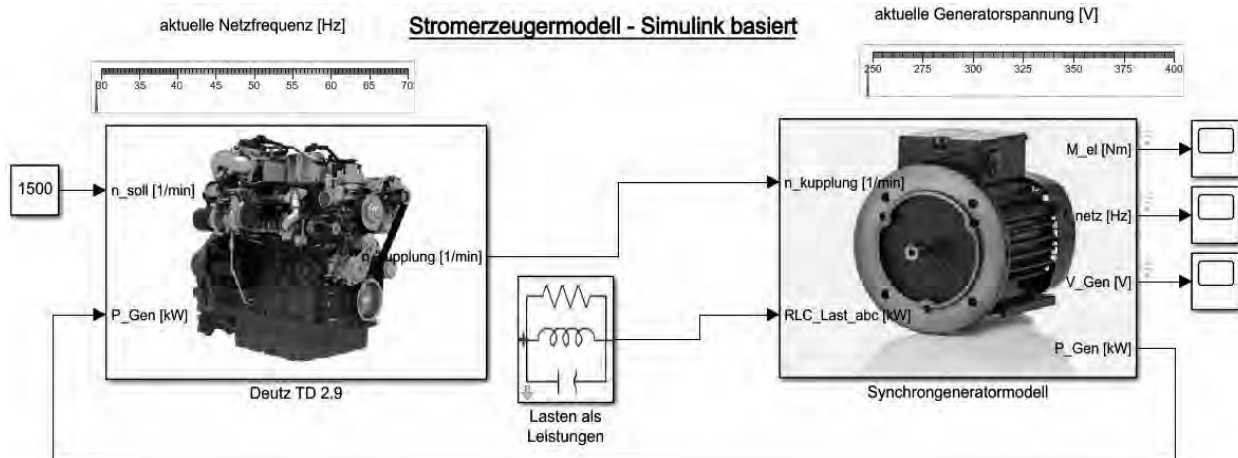


Abbildung 7: Oberste Modellebene und Benutzeroberfläche zur Eingabe von Lasten sowie Darstellung der wichtigsten Prozessparameter; Bildelemente des Simulink®-Modells (Bildelemente aus [11] und [12])

nung entspricht dann dem zu modellierenden Synchron-generator im Prüfaufbau, jedoch immer noch mit vereinfachter Darstellung des Erregerkreises.

Die Implementierung des signalflußorientierten Modells in MATLAB Simulink® ist in Abbildung 7 in der oberste Modellebene des gesamten Simulink®-Modells (Synchrongenerator und Verbrennungsmotor nach [1]) dargestellt. Die Anforderung aus dem Projekt nach einer einfachen Bedienoberfläche, die das Zusammenspiel aus Generator und Motor abbildet, ist somit erfüllt. Es können Lasten als ohmsche, induktive oder kapazitive Leistungen getrennt für jede einzelne Phase definiert und mit Lastsprüngen versehen werden.

Die resultierenden Netzwankungen werden in Frequenzstabilität und Spannungsstabilität angezeigt. Wichtige Modellparameter des Verbrennungsmotors und des Synchrongenerators können über Anklicken der entsprechenden Subsystemblöcke eingegeben werden. Somit wurde ein vollständig in MATLAB-Simulink® erstelltes Modell entwickelt, das mit einer reduzierten Parametrierung arbeitet und noch ohne die Berücksichtigung des transienten und subtransienten Verhaltens rechnet. Ebenfalls unberücksichtigt bleiben vorerst der Temperatureinfluss sowie die Sättigungseffekte. Die konkrete Weiterführung der Modellentwicklung wird abhängig von den Messergebnissen am geplanten Prüfstand entschieden.

Für eine erste Validierung dient ein MATLAB Simscape® Modell, welches laut [2] ebenfalls die Eigenschaften des vorgestellten Modells beinhaltet. In dem Simscape®-Modell für einen Synchrongenerator wird die Erregerleistung aus einer externen Spannungsquelle bezogen. Um die Vergleichbarkeit der beiden Modelle

zu gewährleisten, wird auch in dem Simulink®-Modell die Erregerleistung von außen bezogen. Es besteht aber nach wie vor die Option die Erregerleistung aus der generierten Leistung zu entnehmen, diese Option wurde in dem Simscape®-Modell nicht realisiert. Auch die Berücksichtigung von Sättigungseffekten wird im Simscape®-Modell deaktiviert. Zudem sei hier erwähnt, dass die Modellparametrierungen sich in der Wahl geeigneter Startwerte und der Regelparame-ter des Excitation Systems unterscheiden können, da hierzu nicht alle Informationen aus dem Simscape®-Modell herausgelesen werden können. Der Modellcode ist gesperrt und Parameter können ausschließlich nachvollzogen werden, wenn diese editierbar sind.

3 Modellvalidierung

Zur Validierung des Simulink®-Modells werden die Berechnungsergebnisse mit Ergebnissen des Simscape®-Modells für einen Synchrongenerator mit 42,5 kVA verglichen. Beide Synchrongeneratormodelle sind mit einem Verbrennungsmotormodell eines Deutz TD 2.9 4-Zylinder Turbodiesel mit 49 kW gekoppelt. Für eine ausführliche Beschreibung des Verbrennungsmotormodells wird auf [1] verwiesen. Verbrennungsmotor und Synchrongenerator sind über eine elastische Kupplung miteinander verbunden. Somit ist die Ausgangsgröße des Verbrennungsmotormodells der zeitliche Drehzahlverlauf hinter der elastischen Kupplung. Dieser Drehzahlverlauf bildet anschließend die Eingangsgröße für das jeweils verwendete Synchrongeneratormodell. Über den Parameter n_{soll} wird die gewünschte Solldrehzahl des Verbrennungsmotors vorgegeben, ein Drehzahlregler hält die Motordrehzahl

auf einem vorgegebenen Wert, im Validierungsfall 1500 min^{-1} . Im untersuchten Szenario soll ein ohmscher Lastsprung von 9 kW auf 30 kW bei einem Zeitpunkt von 10 s simuliert werden. Anschließend kommt es zum Zeitpunkt von 20 s zu einer induktiven Lastaufschaltung von 12 kVar. Dabei werden alle drei Phasen symmetrisch belastet. Die Generatorausgangsspannung wird über einen PID-Regler auf einem Effektivwert von 325 V, gemittelt über alle drei Phasen, gehalten. Eine kapazitive Lastaufschaltung wird nicht untersucht, da diese in der Praxis sehr selten auftritt. Der gesamte Simulationszeitraum beträgt 25 s. Es werden die Verläufe der Netzfrequenz, des elektromagnetischen Momentes und der Generatorausgangsspannung zwischen dem Simulink®- und dem Simscape®-Modell verglichen. Die für beide Modelle verwendeten Parameter sind in der untenstehenden Tabelle aufgelistet¹:

Parameter	Parameterwert
R_a	0,1810 Ω
p	2
R_{fd}	0,1002 Ω
R_{kd}	0,0664 Ω
R_{kq1}	0,0292 Ω
R_{kq2}	0,0079 Ω
L_d	0,02529 H
L_l	0,00095 H
L_{lkd}	0,00139 H
L_{fld}	0,00132 H
L_{lfd}	0,00190 H
L_q	0,01114 H
L_{kq1}	0,00069 H
L_{kq2}	0,00248 H
L_0	0,00009 H
Proportional Gain	1
Integral Gain	1
Differential Gain	0
$U_{\text{eff_soll}}$	325 V
N_{soll}	1500 min^{-1}
xL	0,45 Ω

In Abbildung 8 ist der Verlauf der Netzfrequenz, wel-

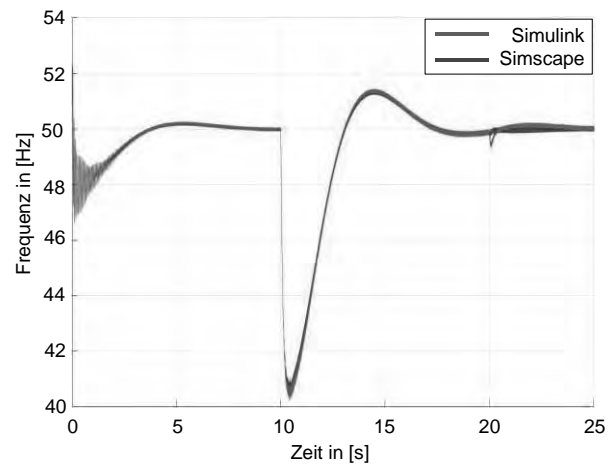


Abbildung 8: Vergleich der Netzfrequenz bei ohmscher und induktiver Lastzuschaltung

che proportional zur Motordrehzahl ist, für den beschriebenen Lastfall dargestellt. Durch die Vorgabe der Startwerte kommt es in den ersten Sekunden zu einem Einschwingverhalten. Ab ca. acht Sekunden wurden quasi stationäre Zustände erreicht. Verbrennungsmotor und Generator laufen mit einer Leistung von rund 9 kW bei 1500 min^{-1} , was aufgrund der Polpaarzahl einer Netzfrequenz von 50 Hz entspricht. Durch den ohmschen Lastanstieg auf 30 kW kommt es zu einem Einbruch der Netzfrequenz auf rund 40 Hz. Durch die Drehzahlregelung des Verbrennungsmotors schwingt die Netzfrequenz in der Folge leicht über und erreicht anschließend ihren Sollwert

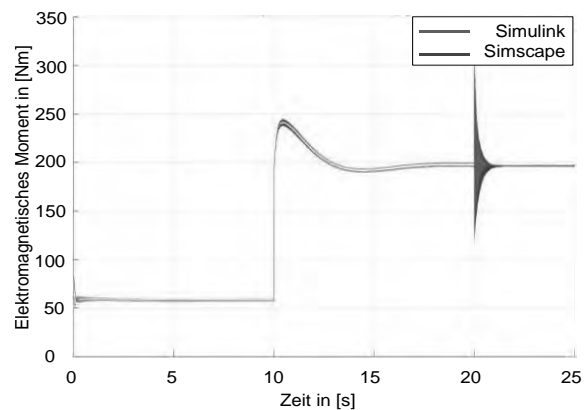


Abbildung 9: Vergleich des elektromagnetischen Momentes bei ohmscher und induktiver Lastzuschaltung

¹ Die Ermittlung der Parameterwerte geht nicht vollständig aus Datenblättern oder Messungen hervor, sondern muss teilweise aus Funktionsbeschreibungen oder Materialbedingungen ermittelt werden [4]. In diesem Fall wurden sie dem Beispielmodell aus MATLAB® Simscape entnommen.

von 50 Hz. Man kann erkennen, dass der Verlauf des Simulink®- Modells mit dem des Simscape®- Modells größtenteils übereinstimmt und ähnliche Verhaltensweisen zu beobachten sind. Abweichungen sind ausschließlich bei der Zuschaltung der induktiven Last zu beobachten.

Die Verläufe des elektromagnetischen Momentes sind in Abbildung 9 dargestellt. Dabei ist eine geringe Abweichung zwischen den zwei Modellen zu beobachten. Durch den ohmschen Lastsprung von 9 kW auf 30 kW kommt es zu einer Erhöhung des elektromagnetischen Momentes. Auch hier ist durch das Regelverhalten der beiden Systeme ein Überspringen zu verzeichnen. Gleichzeitig erklärt der Momentanstieg den Drehzahl- und somit auch den Frequenzeinbruch. Durch die sprunghafte Erhöhung der Generatorlast, steigt auch die Lastanforderung an den Verbrennungsmotor. Durch die Trägheiten des Systems kann erst mit einer gewissen Verzögerung auf den Lastanstieg reagiert werden, die Drehzahl bricht dementsprechend im ersten Moment ein. Nach dem Einschwingvorgang erreicht das elektromagnetische Moment konstante Werte.

Die induktive Lastzuschaltung führt zu einer hochfrequenten gedämpften Schwingung des Momentes, welche nach einer kurzen Zeit abklingt, sodass das Moment wieder konstante Werte erreicht. Auch hier ist zu erkennen, dass die Verläufe der untersuchten Modelle eine sehr gute Übereinstimmung aufweisen. Die Unterschiede während der induktiven Lastaufschaltung können gegebenenfalls durch eine Anpassung des ohmschen Widerstandes der induktiven Last verringert werden. Dieser konnte aus dem Simscape®- Lastenmodell nicht entnommen werden.

Abbildung 10 bildet die Spannungsverläufe ab. Die Generatorsollspannung wird nach dem anfänglichen Einschwingverhalten schnell erreicht. Durch die ohmsche Lastzuschaltung bricht die Spannung ein. Eine Erhöhung der Erregerspannung durch das Excitation- System führt dazu, dass die Sollgröße anschließend schnell erreicht wird. Auch die induktive Lastzuschaltung führt zu einer Störung des Generatorspannungsverlaufes. Die Generatorspannung übersteigt kurzzeitig den Sollwert. Beide Modelle reagieren annähernd gleich auf den untersuchten Lastfall. Die größten Unterschiede sind während der induktiven Lastzuschaltung zu verzeichnen.

4 Zusammenfassung

Mit dem hier vorgestellten Modell wurde eine wich-

tige Grundlage zur Erreichung des Projektziels geschaffen. Es wurden erfolgreich ein physikalisch-mathematisches Modell in MATLAB® Simulink® entwickelt und die Zusammenhänge der Signalkückkopplung über die elektrischen Lasten beschrieben. Ein Vergleich von Berechnungsergebnissen mit einem Simscape®-Modell zeigt nur geringe Abweichungen, die besonders während

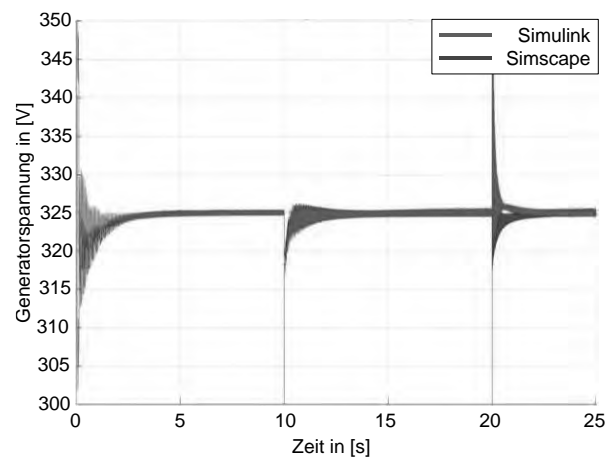


Abbildung 10: Vergleich der Generatorspannungsamplitude bei ohmscher und induktiver Lastzuschaltung

der induktiven Lastzuschaltung deutlicher werden. Der erreichte Stand bietet eine gute Basis für Modellanpassungen und Modellerweiterungen. Diese richten sich nach den Messergebnissen, welche aus dem Referenzprüfaufbau gewonnen werden, der sich aktuell an der Hochschule Wismar im Aufbau befindet und im kommenden Jahr den Messbetrieb aufnimmt.

Danksagung

Die hier vorgestellten Ergebnisse wurden im Rahmen des öffentlich finanzierten Projektes SIDYN (FKZ: 13FH043PX8) erarbeitet. Die Autoren möchten sich beim BMBF für die Unterstützung des Vorhabens bedanken.

Nomenklatur

Parameter	Bedeutung	Einheit
C	äquivalente Kapazität der kapazitiven Last	[F]
Differential Gain	Differentialanteil PID-Regler	[-]
i_a, i_b, i_c	Strom (Phase ABC/UVW)	[A]
i_d	Strom d-Achse	[A]
i_{fd}	Erregerfeldstrom	[A]
i_{kd}	d-Dämpferkreisstrom	[A]

Parameter	Bedeutung	Einheit
i_{kq}	q1-Dämpferkreisstrom	[A]
i_{kq}	q2-Dämpferkreisstrom	[A]
Integral Gain	Integralanteil PID-Regler	[-]
i_q	Strom q-Achse	[A]
i_0	Strom 0-System	[A]
L_d	Induktivität d-Achse	[H]
L	äquivalente Induktivität der induktiven Last	[H]
L_{fld}	Gegeninduktivität Feldwicklung zu d-Dämpferkreis	[H]
L_l	Statorstreueinduktivität	[H]
L_{lfd}	Induktivität Feldwicklung	[H]
L_{lkd}	Induktivität d-Dämpferkreis	[H]
L_{kq1}	Induktivität q1-Dämpferkreis	[H]
L_{kq2}	Induktivität q2-Dämpferkreis	[H]
L_q	Induktivität q-Achse	[H]
L_0	Induktivität 0-System	[H]
n_{soll}	Solldrehzahl	[U/min]
P_{gen}	Generatorleistung	[W]
Proportional Gain	Proportionalanteil PID-Regler	[-]
R_a	Widerstand Ständerkreis	[Ω]
RLC_Lasten_abc	Lastvorgaben (induktiv, kapazitiv und ohmsch)	[kW]
T	elektromagnetisches Moment	[Nm]
u_a, u_b, u_c	Spannung (Phase ABC/UVW)	[V]
u_d	Spannung d-Achse	[V]
U_{eff_soll}	effektiver Generatorspannungssollwert	[V]
u_{fd}	Erregerfeldspannung	[V]
u_q	Spannung q-Achse	[V]
u_0	Spannung 0-System	[V]
x_F	äquivalenter Widerstand der Erregerfeldleistung	[Ω]
x_L	ohmscher Widerstand der induktiven Last	[Ω]
x_R	äquivalenter Widerstand der ohmschen Last	[Ω]
ψ_d	magnetischer Fluss d-Achse	[Wb]
ψ_{fd}	magn. Fluss Feldwicklung	[Wb]
ψ_{kd}	magn. Fluss d-Dämpferkreis	[Wb]
ψ_{kq}	magn. Fluss q1-Dämpferkreis	[Wb]
ψ_{kq}	magn. Fluss q2-Dämpferkreis	[Wb]
ψ_q	magn. Fluss q-Achse	[Wb]

Parameter	Bedeutung	Einheit
ψ_0	magn. Fluss 0-Achse	[Wb]
ω_{mech}	Winkelgeschwindigkeit Verbrennungsmotor	[rad/s]
ω_e	Winkelgeschwindigkeit des elektrischen Drehfeldes	[rad/s]

Referenzen

- [1] Jörss D., Herrmann B., Fink C., „Modellierung des Betriebsverhaltens eines Diesel-Industriemotors als Stromerzeuger“, Virtueller ASIM Workshop der Fachgruppen STS/GMMS & EDU, <https://www.cea-wismar.de/asim2021/tagungsband/>, 2021
- [2] The MathWorks, Inc., <https://de.mathworks.com/help/phymod/sps/power-sys/ref/synchronousmachinesifundamental.html>, letzter Zugriff: 27.10.2021
- [3] Electrical Machinery Committee of the IEEE Power and Energy Society, *IEEE Guide for Synchronous Generator Modelling Practice and Parameter Verification with Applications in Power Systems Stability Analysis*, IEEE Std 1110™-2019: New York, 2020
- [4] Electrical Machinery Committee of the IEEE Power and Energy Society, *IEEE Guide for Test Procedures for Synchronous Machines Including Acceptance and Performance Testing and Parameter Determination for Dynamic Analysis*, IEEE Std 115™-2019: New York, 2020
- [5] Kundur P., *Power System Stability and Control*, New York: McGraw-Hill, Inc., 1993: 45 ff.
- [6] Ternes B., Feldkamp J., „Modellbasierte Implementierung einer Vektorregelung für Synchronmaschinen“, Hochschule Bochum, 2015
- [7] Mrugowsky H., *Drehstrommaschinen im Inselbetrieb*, Wiesbaden: Springer Fachmedien, 2015
- [8] Binder A., *Elektrische Maschinen und Antriebe*, Berlin: Springer-Verlag, 2017
- [9] Beckert U., „Modellierung und Regelung der permanent-erregten Synchronmaschine - Skriptum für Nichtelektrotechniker“, TU Bergakademie Freiberg, 2011
- [10] Spoljaric et al., „Synchronous Generator Modeling Using Matlab“, University of Osijek, Croatia
- [11] Deutz AG, https://www.deutz.com/produkte/motoren?tx_deutzenginde_pi1%5Baction%5D=list&tx_deutzengines_pi1%5BconBcontro%5D=Engine&tx_deutzengines_pi1%5Ben-gine%5D=133&cHash=b7163a33035a85badaeb40b605e06e13, letzter Zugriff: 27.10.2021
- [12] Johannes Hübner Fabrik elektrischer Maschinen GmbH, <https://www.huebner-giessen.com/produkte/motoren-generatoren/detail/synchronegenerator/>, letzter Zugriff: 27.10.2021

Systematischer modellbasierter Entwurf einer Reinforcement Learning-basierten neuronalen adaptiven Geschwindigkeitsregelung

Xiaobo Liu-Henke*, Sven Jacobitz, Or Aviv Yarom, Jannis Fritz

Institut für Mechatronik, Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel, Deutschland; *x.liu-henke@ostfalia.de

Abstract. In diesem Beitrag wird der systematische modellbasierte Entwurf einer Reinforcement Learning-basierten neuronalen adaptiven Geschwindigkeitsregelung beschrieben. Ausgehend von einer Einleitung und der Zusammenfassung aktueller Grundlagen, werden Entwurfsmethoden für intelligente Fahrfunktionen vorgestellt. Der Fokus liegt auf der erstmaligen Vorstellung einer neuartigen Entwurfsmethodik für Künstliche Neuronale Netze in der Regelungstechnik. Diese wird anschließend am Beispiel einer adaptiven Geschwindigkeitsregelung vollständig angewendet und validiert.

Einleitung

Der vom Europäischen Fonds für regionale Entwicklung (EFRE) geförderte Innovationsverbund *autoMoVe (Dynamisch konfigurierbare Fahrzeugkonzepte für den nutzungsspezifischen autonomen Fahrbetrieb)*, hat die Entwicklung eines autonomen, modularen und elektrischen Fahrzeugkonzeptes zum Ziel. Durch den Austausch anwendungsspezifischer Module zur Laufzeit soll eine Vielzahl von Anwendungen vom innerbetrieblichen Gütertransport bis zur Personenbeförderung im Straßenverkehr autonom realisiert werden. Im Rahmen dieses Forschungsvorhabens fokussiert das Teilprojekt der Ostfalia *autoEVM (Ganzheitliches elektronisches Fahrzeugmanagement für autonome Elektrofahrzeuge)* die modellbasierte Entwicklung innovativer intelligenter Algorithmen und Funktionen für den autonomen Fahrbetrieb.

Mit einer höheren Automatisierung des Fahrbetriebs geht auch ein Anstieg der Anforderungen an das Fahrzeug bzw. die automatisierten Fahrfunktionen einher. Aktuelle Funktionen und Algorithmen, die auf Methoden der Regelungstheorie oder der klassischen Informationsverarbeitung basieren, können diesen nicht mehr in vollem Umfang gerecht werden [1]. Deshalb stellt Künstliche Intelligenz (KI) in diesem Vorhaben bzw. für viele Domänen, die an der Entwicklung und Nutzung intelligenter, automatisierter Fahrzeuge beteiligt sind eine

Schlüsseltechnologie dar [2].

Unabhängig von der Art der Informationsverarbeitung sind moderne Fahrzeuge und Fahrfunktionen komplexe mechatronische Systeme mit einem hohen internen und externen Vernetzungsgrad. Zur Beherrschung dieser Komplexität im Entwicklungs- und Absicherungsprozess ist ein systematisches Entwurfsvorgehen unabdingbar. Dabei wird zum einen auf eine in der Mechatronikforschung etablierte verifikationsorientierte und simulationsgestützte Entwurfsmethodik zurückgegriffen. [3] KI-Algorithmen, insbesondere die hier eingesetzten Künstliche Neuronale Netze (KNN), unterscheiden sich in ihrer Funktionsweise und ihrem Auslegungsprozess stark von regelungstheoretischen Ansätzen. Daher wird in dieser Veröffentlichung zum anderen eine neuartige Entwurfsmethodik für KNN angewandt und erstmalig vorgestellt.

Als Teilziel des oben genannten Forschungsvorhabens wird in diesem Beitrag die erwähnte Entwurfsmethodik für KNN auf ein System zur automatisierten Längsführung im Sinne einer intelligenten adaptiven Geschwindigkeitsregelung angewendet. Intelligent bezieht sich zum einen auf den Einsatz der, an der Funktionsweise des menschlichen Gehirns angelehnten, KNN und zum anderen auf den selbstlernenden bzw. erfahrungs-basierten Trainingsansatz des Reinforcement Learning.

1 Stand der Technik

1.1 Intelligente Fahrfunktionen zur automatisierten Längsführung

Mit steigender Anzahl und Vernetzung von Fahrerassistenzsystemen (FAS) delegiert der menschliche Fahrer sukzessive Fahraufgaben an das Fahrzeug, bis er beim autonomen Fahrbetrieb schließlich zum Passagier wird. Dann spricht man nicht mehr von FAS, sondern von (automatisierten) Fahrfunktionen.

Ein Beispiel für eine solche automatisierte Fahrfunktion ist die hier betrachtete adaptive Geschwindigkeitsregelung (*engl. Adaptive Cruise Control (ACC)*) als Weiterentwicklung der Geschwindigkeitsregelung. Im Gegensatz zur einfachen Geschwindigkeitsregelung, bei der lediglich ein Abgleich zwischen der Vorgabegeschwindigkeit des Fahrers und der Ist-Geschwindigkeit des Fahrzeugs stattfindet, kann die ACC durch Auswertung von Umfeldsensorik die Vorgabegeschwindigkeit nach unten hin überschreiben. Durch anschließende Vorgabe von Sollwerten an unterlagerte Systeme der Längsdynamik (Antrieb, Bremse, Getriebe), hält das Fahrzeug einen geschwindigkeitsabhängigen Sicherheitsabstand zu vorausfahrenden Fahrzeugen und Objekten ein. [4]

Für die Erkennung vorausfahrender Hindernisse sowie die anschließende Bestimmung der relativen Geschwindigkeit und des Abstands kommen typischerweise Radarsensoren zum Einsatz [5]. Moderne Systeme nutzen häufig zusätzlich Kamera- und Lidarsensoren zur Objekterkennung. Ein weiterer Ansatz ist der Einsatz drahtloser Vehicle-to-everything (V2X)-Kommunikation zur Realisierung eines kooperativen Fahrbetriebs [8].

Im regelungstechnischen Sinne ist die ACC ein kaskadiertes System, bei dem die innere Kaskade eine Geschwindigkeit einregelt, die von der äußeren Kaskade vorgegeben wird, um den Sicherheitsabstand einzuregeln. In der Praxis existieren jedoch einige Fahrfunktionen, die zwischen dem Abstandhalten und der reinen Geschwindigkeitsregelung umschalten. Für einen hochautomatisierten Fahrbetrieb sind höherwertige und ganzheitliche Funktionen jedoch von Vorteil. [5]

In der aktuellen Literatur sind viele Ansätze, wie eine klassische PI- [6], eine nichtlineare modellprädiktive [7] oder auch eine neuro-fuzzy Regelung [4] für eine ACC mit jeweils spezifischen Vor- und Nachteilen zu finden. In diesem Beitrag soll ein neuartiger ganzheitlicher Ansatz für eine ACC mit KNN und Reinforcement Learning untersucht werden.

1.2 Grundlagen Künstlicher Neuronaler Netze und Maschinellen Lernens

Der Begriff KI umfasst eine Vielzahl unterschiedlichster Methoden und Algorithmen, die sich mit der selbstständigen und automatisierten Lösung von Problemen befassen [2]. KNN und ML bilden ein Teilgebiet der KI, das sich in zahlreichen Problemen verschiedenster Gebiete, auch beim autonomen Fahrbetrieb, als geeignet erwiesen hat. Deshalb fokussiert sich dieser Beitrag auf dieses Teilgebiet. Die zahlreichen positiven Eigenschaften von

KNN und ML, wie Anpassungsfähigkeit, Fehlerresistenz, Vielseitigkeit und vor allem Lernfähigkeit sind auf ihre Anlehnung an die Struktur und Funktionsweise des menschlichen Gehirns zurückzuführen.

Analog zur Biologie sind (künstliche) Neuronen Verarbeitungseinheiten, die Eingangsreize über gewichtete Verbindungen kumulieren und mit Hilfe einer Aktivierungsfunktion eine Ausgabe berechnen. Durch die Zusammenschaltung mehrerer Neuronen in mindestens zwei Schichten entsteht das KNN. Üblich sind Kombinationen von bis zu einigen Hundert Neuronen in bis zu über Einhundert Schichten. Dabei sind nicht nur beliebige vorwärtsgerichtete, sondern auch zeitlich rückgekoppelte Verbindungen im KNN möglich. Die optimale Architektur eines KNN lässt sich bisher nicht analytisch bestimmen [9]. Deshalb ist neben Erfahrung und Versuchsreihen auch eine systematische Entwurfsmethodik notwendig, um eine geeignete Architektur im Zielkonflikt zwischen Rechenaufwand und Leistungsfähigkeit zu finden. Die Anzahl, Verschaltung und Gewichtung der Verbindungen charakterisiert die „Intelligenz“ eines KNN. Allgemein gesagt, bedeuten mehr Neuronen und Verbindungen eine höhere Leistungsfähigkeit des KNN, bei gleichzeitig steigendem Rechenaufwand.

Genau wie ein menschliches Gehirn, muss das KNN eine Aufgabe zunächst lernen bzw. trainieren. Diese Begriffe bezeichnen die Anpassung der Verbindungs-gewichte. Im Umfeld des autonomen Fahrens sind dafür das Supervised Learning (SL) und das Reinforcement Learning (RL) relevant. Beim SL werden dem KNN Eingangsdaten und die zugehörige Ausgabe präsentiert. Das KNN lernt iterativ den Zusammenhang zwischen den beiden Größen [10]. Dieses Lernverfahren eignet sich z.B. besonders für bildbasierte Objekterkennung [11]. Beim RL lernt das KNN sukzessive aus der Erfahrung vergangener Sequenzen die optimale Strategie im Sinne einer vorgegeben Belohnungsfunktion [10]. Dieses Verfahren wird eingesetzt, wenn keine Trainingsdaten verfügbar sind, z.B. bei der automatisierten Fahrzeugführung [12]. SL und RL sind Oberkategorien von Lernverfahren, mit diversen Trainingsalgorithmen. Genau wie die KNN Architektur lassen sich die optimalen Trainingsalgorithmen bzw. deren Parameter nicht analytisch bestimmen. So sind auch hier Erfahrung und Versuche erforderlich.

2 Entwurfssystematik

2.1 Modellbasierte Reglerauslegung

Die Komplexität moderner Fahrzeuge steigt durch den höheren internen und externen Vernetzungsgrad sowie die wachsende Anzahl intelligenter und leistungsfähiger Hard- und Softwarekomponenten stetig an. Zur Beherrschung der Systemkomplexität und frühzeitigen Fehlervermeidung bei der Auslegung der Informationsverarbeitung, ist eine ganzheitliche Entwurfsmethodik unverzichtbar. Daher hat sich die durchgängige, verifikationsorientierte, modellbasierte Entwurfsmethodik, basierend auf dem Rapid Control Prototyping (RCP) und Model-in-the-Loop-(MiL), Software-in-the-Loop-(SiL) und Hardware-in-the-Loop-(HiL) Simulationen etabliert. [13]

Die Methodik baut auf funktionsorientierten physikalischen Modellen einer Regelstrecke auf. Die Regelungsfunktion wird anschließend in Abhängigkeit des Systemverhaltens simulativ ausgelegt und frühzeitig in MiL-Simulationen validiert. Um manuelle Programmierung zu umgehen, werden Modell und Regelungsfunktion in blockschaltbildbasierten Programmiersprachen entwickelt. Der anschließend automatisch generierte Funktionscode wird in SiL-Simulationen erneut gegen das Streckenmodell getestet. HiL-Simulationen dienen zur weiteren Validierung und Optimierung der Informationsverarbeitung mit echtzeitfähigen Simulationsmodellen und realen Teilkomponenten des zu regelnden Systems.

Der verifikationsorientierte und iterative Ansatz dieser Methodik unterstützt den Entwicklungsprozess auch in der herausfordernden Aufgabe der Absicherung. Die Methodik adressiert die Schwächen der klassischen auf physischen Prototypen basierenden Validierung, wie einem hohem Ressourcenaufwand oder Sicherheitsrisiken für Mensch, Maschine und Umwelt. Durch ihren virtuellen Charakter sparen MiL-, SiL- und HiL-Simulationen Zeit und Kosten [14]. Sie ermöglichen jederzeit durchführbare und reproduzierbare Tests ohne direkte Abhängigkeit von physischen Prototypen, Tageszeiten oder menschlichen Experten. So lassen sich Simulationsdurchläufe, für verschiedene Funktionsvarianten oder Szenarien automatisiert durchführen. Damit eignet sich diese Methodik auch besonders für das Training von KNN. Denn das sogenannte maschinelle Lernen verläuft, bis auf seltene Ausnahmen, immer iterativ.

Virtuelle Entwurfsmethoden wie diese bilden die Grundlage für viele intelligente Systeme, wie hochautomatisierte Fahrzeuge. Mit prototypenbasierten Tests sind

die benötigten Hunderttausenden Testkilometer nicht mit vertretbarem Zeit- und Kostenaufwand zu leisten. [14]

2.2 Systematische Auslegung von KNN

Die im vorherigen Abschnitt 2.1 vorgestellte Entwurfssystematik zum modellbasierten Reglerentwurf ist eine übergeordnete Methodik. Die Auslegung eines KNN kann analog zu einer Reglerauslegung mit dynamischer Kompensation als ein Teil dieser Methodik betrachtet werden. KNN Auslegung ist ein komplexer Prozess, der aus vielen Entwurfsentscheidungen besteht, welche einzeln oder in Kombination einen großen Einfluss auf das Entwicklungsergebnis haben. Diese Entscheidungen sind über den gesamten Entstehungsprozess der KNN verteilt und betreffen z.B. die Systemstruktur (End-2-End-Funktion / Kombination von Sub-Funktionen), die Lernart (SL / RL), die Netzarchitektur, die Parametrierung des Trainingsalgorithmus oder auch die Bewertung des KNN.

Die Besonderheit hierbei liegt darin, dass im Gegensatz zu den sehr systematisch-analytischen Auslegungsverfahren der klassischen und modernen Regelungstechnik, die Entscheidungen bzw. deren Auswirkungen bei der KNN Auslegung für den menschlichen Verstand meist nicht mathematisch-logisch durchdringbar sind. Infolgedessen existiert bisher kein systematisches Auslegungsverfahren für KNN [9]. Obwohl man in aktueller Literatur viele Anwendungen von KNN finden kann, sind Informationen zu den jeweiligen Auslegungsprozessen rar. Wenn Angaben dazu gemacht werden, verfolgen die Autoren oft einen ergebnisorientierten auf Erfahrungen basierenden empirischen Ansatz mit wenig Systematik.

Das in Abbildung 1 dargestellte Vorgehensmodell stellt einen ersten Lösungsansatz vor, um den Auslegungsprozess von KNN zu systematisieren. Der Ausgangspunkt ist die regelungstechnische Problemspezifikation. Sie unterscheidet sich kaum von der klassischen Regelungstheorie. Beispielsweise wird auch hier - vor allem beim RL - typischerweise modellbasiert gearbeitet. Die Besonderheit liegt in der Wahl der Systemstruktur und Schnittstellen. Während sich diese bei klassischen Reglern oft aufgrund physikalischer Gegebenheiten ergeben, können sie bei KNN theoretisch frei gewählt werden. Anschließend muss eine für das jeweilige Problem geeignete Lernart bzw. ein konkreter Lernalgorithmus gewählt werden. Es existieren viele Lernalgorithmen, mit spezifischen Vor- und Nachteilen, die für jede Anwendung neu ausgewählt werden müssen. Beim RL unterscheidet man grundsätzlich zwischen gradienten-basierenden und gradientenfreien Algorithmen.

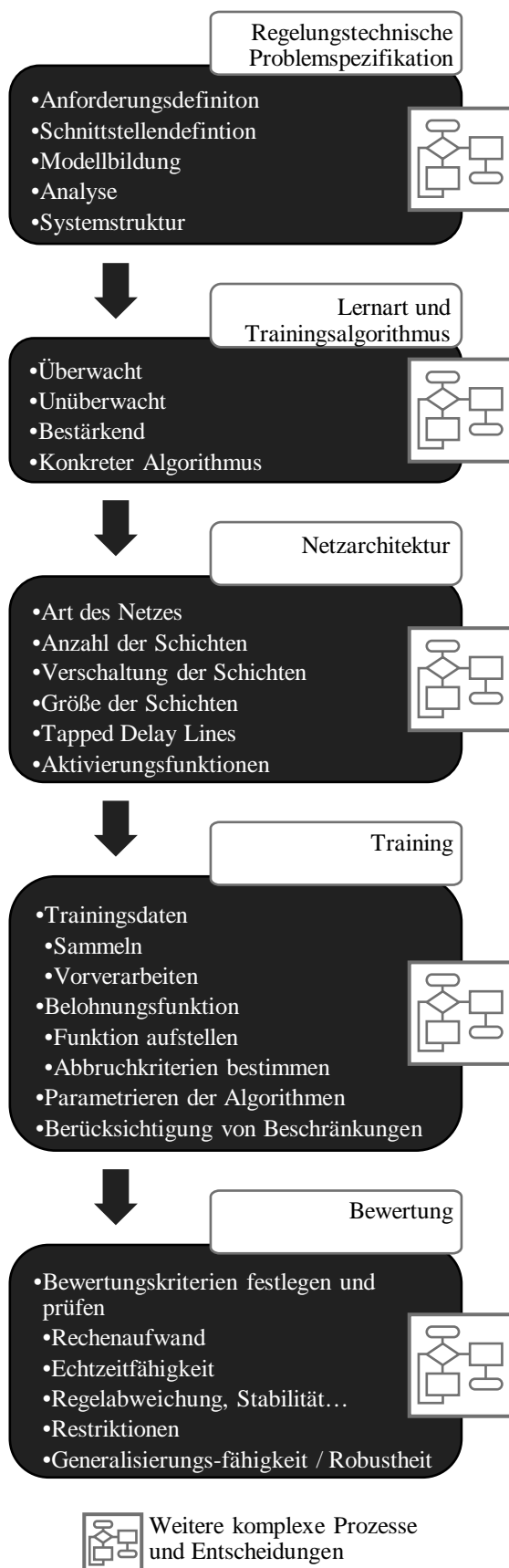


Abbildung 1: Vorgehensmodell zur systematischen Auslegung von KNN

Die Auswahl der Netzarchitektur ist der aufwendigste Teil der KNN Auslegung und hat sich zu einem eigenen modernen Forschungsgebiet entwickelt. Dabei versucht man die Architekturauswahl als Suchproblem zu formulieren und durch ein überlagertes Optimierungsverfahren automatisiert zu lösen. Man spricht von der Neural Architecture Search (NAS). [15] Nachdem die Netzarchitektur feststeht, kann mit dem Training, dem eigentlich Kern der Auslegung, fortgefahren werden. Dabei wird das Verhalten des KNN hinsichtlich der ursprünglichen Aufgabe optimiert. In diesem Schritt müssen vor allem die Hyperparameter der Trainingsalgorithmen eingestellt werden. Der letzte Schritt ist die Bewertung des KNN hinsichtlich der ursprünglichen Anforderungen und unter Berücksichtigung seines Einsatzzwecks, z.B. auf einem Echtzeitsystem.

Wie bereits bei der NAS angedeutet, bildet jeder Block im Vorgehensmodell aus Abbildung 1 einen übergeordneten Prozessschritt und beinhaltet jeweils weitere komplexe Prozesse und Entscheidungen. Da sich dieser Beitrag auf die Anwendung eines KNN als ACC fokussiert, werden nur einige davon später in diesem Beitrag konkreter aufgegriffen.

3 Systematischer modellbasierter Entwurf der neuronalen adaptiven Geschwindigkeitsregelung

Ziel des Beitrags ist der systematische modellbasierte Entwurf einer Reinforcement Learning-basierten neuronalen adaptiven Geschwindigkeitsregelung. Dieser Prozess wird nun nach der vorgestellten Entwurfsmethodik aus Abbildung 1 vorgestellt.

3.1 Regelungstechnische Problemspezifikation

Anforderungsdefinition

Die hier vorgestellte und entwickelte Funktion zur automatisierten Längsführung soll die folgenden Anforderungen erfüllen:

1. Eine vom Fahrer vorgegebene Geschwindigkeit einhalten und keinesfalls überschreiten
2. Einen 2-Sekunden-Sicherheitsabstand zu einem möglicherweise vorausfahrenden Fahrzeug einhalten
3. Den Sicherheitsabstand nicht bzw. nur wenig und kurz unterschreiten

4. Beliebigen Geschwindigkeitsprofilen folgen, auch beim Einhalten des Sicherheitsabstands
5. Geschwindigkeit und Abstand möglichst ohne Schwingungen einhalten
6. Das Ego Fahrzeug soll Beschleunigungen im Bereich $-3 \frac{m}{s^2}$ bis $2 \frac{m}{s^2}$ realisieren
7. Auf KNN und RL basieren

Systemstruktur und Schnittstellendefinition

Abbildung 2 zeigt die Systemstruktur, der hier entworfenen neuronalen adaptiven Geschwindigkeitsregelung inklusive Systemumgebung schematisch. Ausgangspunkt ist ein Führungsfahrzeug (engl. *lead car*), das sich mit einer beliebigen Geschwindigkeit v_{lead} bewegt und damit seine Position x_{lead} verändert. Das eigentliche Zentrum der Struktur ist aber das Ego-Fahrzeugmodell, das basierend auf Beschleunigungs- F_A und Bremskräften F_B die Geschwindigkeit v_{ego} und Position x_{ego} des Ego-Fahrzeugs berechnet. Diese Positionen und Geschwindigkeiten des Ego- und Lead Car, dienen dem Radarsensor mit „eingebautem“ Preprocessing, um innerhalb seiner Reichweite von 150 m die Eingangsgrößen für das KNN zu bestimmen. Diese sind:

- der Geschwindigkeitsfehler v_{err} , also die Differenz von der vom Fahrer vorgegebenen Geschwindigkeit

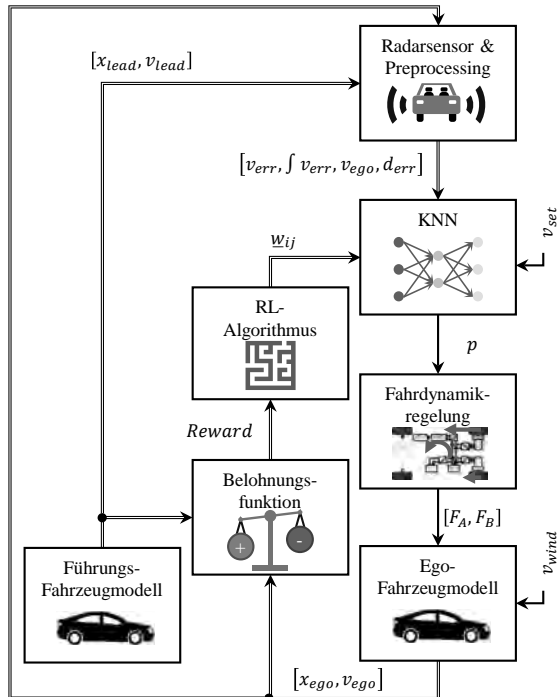


Abbildung 2: Funktionsstruktur der Reinforcement Learning-basierten neuronalen adaptiven Geschwindigkeitsregelung

v_{set} und der tatsächlichen Ego-Geschwindigkeit

- das Integral des Geschwindigkeitsfehlers über die Zeit $\int v_{err}$
- die Ego-Geschwindigkeit v_{ego}
- der Abstandsfehler d_{err} , also die Differenz zwischen dem sicheren Abstand d_{safe} und dem tatsächlichen Abstand zwischen den Fahrzeugen d .

Das KNN berechnet dann mithilfe dieser Eingangsgrößen eine Vorgabe für die Fahr- und Bremspedalstellung $p \in \{-1 \leq p \leq 1 \mid \mathbb{R}\}$, welche von einer unterlagerten Fahrdynamikregelung in die oben genannten Längskräfte umgesetzt wird.

Dies sind alle Komponenten, die das System in der Anwendungsphase, also nach dem Training verwendet. Während des Trainings, werden zusätzlich zwei weitere Komponenten, benötigt. Die vom Entwickler vorgegebene bzw. auszuarbeitende Belohnungsfunktion, welche basierend auf Daten der Systemkomponenten einen sogenannten *Reward* berechnet, um das KNN im Training zu bewerten. Der RL-Algorithmus ist ein Optimierungsverfahren, das die Gewichte $w_{i,j}$ des KNN so anpasst, dass der erwartete *Reward* in der kommenden Episode (ein Simulationsdurchlauf), maximiert wird.

Modellbildung

Das Gesamtsystem aus Abbildung 2 wird in der Simulationsumgebung für die automatisierte Modellkonfiguration zur Auslegung und Absicherung KI-basierter Fahrfunktionen aus [16] modelliert, simuliert und trainiert.

Um das Verhalten der Ego und Lead Fahrzeuge abzubilden, dient ein klassisches Längsdynamikmodell:

$$m \cdot \ddot{x} = F_A - \underbrace{c_w \cdot A \cdot \frac{\rho}{2} \cdot \dot{x}^2}_{\text{Luftwid.}} - \underbrace{\frac{m \cdot \sin(\alpha)}{\text{Steigungswid.}}}_{\text{Rollwid.}} - \underbrace{m \cdot \cos(\alpha) \cdot f_R}_{\text{Rollwid.}} - F_B. \quad (1)$$

Die Variable für die Fahr- und Bremspedalstellung p wird auf die entsprechenden Kräfte F_A und F_B skaliert. Um einen Stillstand und eine Rückwärtsfahrt simulieren zu können, wurden in der Simulationsumgebung weitere Mechanismen implementiert, auf die in diesem Beitrag nicht eingegangen wird.

Der Radarsensor mit Preprocessing wird aktiviert, sobald die Distanz zwischen den Fahrzeugen die Reichweite des Sensors unterschreitet. Hier werden zum einen situationsbedingte Sollwerte für Abstand und Geschwindigkeit berechnet. Beispielsweise gilt für die Sollgeschwindigkeit des Ego Car $\min(v_{set}, v_{ego})$, wenn sich das Ego Car im Bereich von d_{safe} zum Lead Car oder

darunter befindet. Zum anderen werden die berechneten Sollgrößen auch mit den Ist-Größen abgeglichen und Abweichungen für die Distanz und die Ego Geschwindigkeit berechnet. So liefert dieser Funktionsbestandteil nicht nur die Eingangsgrößen für das KNN, sondern auch Basisgrößen für die Belohnungsfunktion.

3.2 Lernart und Trainingsalgorithmus

Prinzipbedingt kommen für diese Anwendung nur Algorithmen des RL infrage. Um einerseits eine optimale ACC Funktion zu erzeugen und andererseits weitere Erfahrung für den systematischen Entwurf von KNN zu sammeln, sollen hier verschiedene Trainingsalgorithmen verwendet und verglichen werden. In Vorversuchen mit niedrigeren Anforderungen kamen die folgenden Trainingsalgorithmen zum Einsatz:

- Genetischer Algorithmus (GA)
- Particle Swarm Optimization (PSO)
- Deep Deterministic Policy Gradients (DDPG).

Bei den beiden ersten Verfahren handelt es sich um gradientenfreie, evolutionsbasierte Algorithmen. Diese Klasse von Algorithmen entwickelt sich in der Regel stetig in Richtung einer höheren Belohnungsfunktion, konvergiert aber nicht immer in ein echtes Optimum. Ihr Vorteil ist, dass Sie in der Regel recht schnell zu einem guten Ergebnis gelangen. Eine Besonderheit des GA ist, dass dieser recht zufallsbehaftet ist und bei richtiger Parametrierung eher dazu neigt, das globale Optimum zu finden, als andere Algorithmen. Der DDPG hingegen, ist ein gradientenbasierter Algorithmus, der ziemlich sicher in ein lokales Optimum konvergiert. Dafür benötigt er jedoch meist mehr Zeit. Außerdem wird nur selten das globale Optimum erreicht.

Die Versuchsreihe zeigte, dass der DPPG mit hoher Wahrscheinlichkeit die besten Ergebnisse erzielt. Um die Ergebnisse vergleichbar zu machen, wurde für alle Algorithmen die gleiche rudimentäre Belohnungsfunktion verwendet. Während diese Versuchsreihe bereits dafür genutzt wurde die Hyperparameter der Algorithmen zu optimieren, wurde die Belohnungsfunktion an späterer Stelle detailliert entworfen.

3.3 Netzarchitektur

Durch die Wahl des DDPG als Trainingsalgorithmus müssen zwei KNN Architekturen gewählt werden. Denn hier werden ein *actor* und ein *critic* Netzwerk benötigt. Das *actor* Netz ist das KNN, welches die spätere Regelungsaufgabe übernimmt. Das *critic* Netz dient als Über-

setzer zwischen der Belohnungsfunktion und dem Algorithmus zur Aktualisierung der Verbindungsgewichte. Es wird nur während der Trainingsphase benötigt.

Das *critic* Netzwerk kann meist als einfaches vorwärtsgerichtetes Netz mit Gleichrichterfunktionen als Aktivierung gewählt werden. Daher wurde hier ein solches KNN mit fünf Schichten als *critic* Netzwerk festgelegt. Die Eingangsgrößen sind neben denen des *actor* Netzes auch dessen Ausgang.

Die Architektur des *actor* Netzwerks wurde mittels einer NAS, die auf einem GA basiert automatisiert optimiert. Das Ergebnis ist ein vorwärtsgerichtetes KNN mit vier Schichten. In den ersten drei Schichten befinden sich jeweils 48 verdeckte Neuronen sowie Gleichrichterfunktionen zur Aktivierung. In der letzten Schicht befindet sich nur noch ein Neuron mit einer *tanh*-Aktivierung.

3.4 Training

Nachdem die Netzarchitekturen festgelegt wurden, muss nun die Belohnungsfunktion entworfen werden, die das letztendliche Verhalten des KNN bzw. des Ego Car bestimmt. Diese besteht aus vier Termen:

- $-0,1 \cdot (v_{err})^2$: Bestrafung von Geschwindigkeitsfehlern, damit die Sollgeschwindigkeit eingehalten wird
- $-(a)^2$: Bestrafung von Beschleunigungen, damit Schwingungen und abrupte Geschwindigkeitsänderungen vermieden werden
- $-0,1 \cdot (d_{err})^2$: Bestrafung von Abstandsfehlern, damit der Sollabstand eingehalten wird
- $+v_{err}$ für $v_{err} \leq 0,25 \frac{m}{s}$: Belohnung von kleinen Geschwindigkeitsfehlern, damit die Sollgeschwindigkeit eingehalten wird.

In einem iterativen Prozess durchläuft das Ego Car nun mehrere Episoden in denen das *actor* Netz immer wieder als Längsdynamikregler eingesetzt wird und für das Verhalten einen entsprechenden Reward erhält. Abbildung 3 zeigt den Verlauf der Rewards über die Episoden sowie den durchschnittlichen Reward. Der Reward ist zu Beginn des Trainings sehr negativ, da das KNN sich ohne Vorwissen noch nicht gut verhält. Der Reward verbessert sich schnell in den Bereich um ca. -600 und steigert sich dann langsam bis das KNN in der besten Episode am Ende ca. -91 Reward-Punkte erreicht. Zwischendurch gibt es vereinzelte Einbrüche, was auf eine falsche Gewichtsveränderung hindeutet, aber vom Algorithmus wieder ausgeglichen werden kann. Dass der beste Reward bei -91 liegt, sagt nicht, dass das Verhalten des KNN schlecht ist. Dies ist lediglich auf das Design der

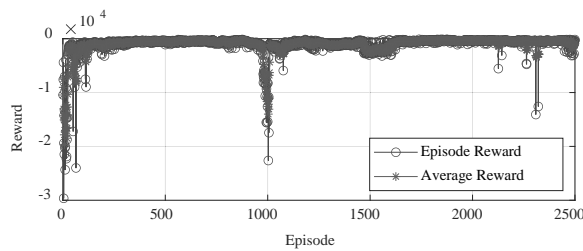


Abbildung 3: Verlauf des Reward beim Training

Belohnungsfunktion zurückzuführen. Der Verlauf der Rewards ist also entscheidender als sein absoluter Wert. Daher ist die Bewertung nach dem Training essentiell.

3.5 Bewertung

Die Bewertung ist der letzte Schritt bei der Auslegung des KNN für die neuronale adaptive Geschwindigkeitsregelung. Ein besonderer Aspekt, der bei der Bewertung berücksichtigt werden muss, ist die sogenannte Generalisierungsfähigkeit. Das bedeutet, dass das KNN das erlernte Wissen auf neue, bisher unbekannte Fragestellungen anwenden soll. Um die Generalisierungsfähigkeit beurteilen zu können, muss sich das Testszenario zur Bewertung also von der Trainingssituation unterscheiden.

Während des Trainings lag v_{set} bei $100 \frac{km}{h}$ und das Lead Car fuhr mit einem Beschleunigungssinus vorneweg. In der Testsituation wird v_{set} auf $108 \frac{km}{h}$ erhöht und das Fahrdynamikmodell des Lead Car durch ein „hartes“ Geschwindigkeitsprofil ersetzt, um die Situation für das KNN weiter zu erschweren. Das Geschwindigkeitsprofil des Lead Car ist im oberen Teil der Abbildung 4 in rot zu sehen, v_{set} in blau. Die gelbe Kurve zeigt die Geschwindigkeit des Ego Car. Im unteren Teil der Abbildung sind die Abstände d_{safe} und d über der Zeit aufgetragen.

Zu Beginn wurde das Ego Car bewusst sehr nah ($d < d_{safe}$) hinter dem Lead Car platziert, sodass es zunächst seine Geschwindigkeit halten muss, um den Sicherheitsabstand einzuhalten. Anschließend folgt das Ego Car dem Geschwindigkeitsverlauf des Lead Car sehr gut und stabil, bis das Lead Car bei ca. 24 s auf $120 \frac{km}{h}$, also oberhalb von v_{set} beschleunigt. Das Ego Car überschreitet, wie gewünscht, v_{set} nicht und der Abstand wird größer. Anschließend bremst das Lead Car abrupt ab und das Ego Car bremst ebenfalls stark, um den Sicherheitsabstand einzuhalten, ohne die Beschleunigungsanforderung (Abschnitt 3.1) zu verletzen. Zum Schluss beschleunigt das Lead Car erneut. Das Ego Car regelt v_{set} stabil ein.

Das bedeutet, dass das KNN die adaptive Geschwindigkeitsregelung generalisierungsfähig erlernt hat und alle Anforderungen zufriedenstellend und erwartungsgemäß erfüllt. Somit kann auch die Entwurfsmethodik als validiert betrachtet werden.

4 Zusammenfassung und Ausblick

In diesem Beitrag wurde der systematische modellbasierte Entwurf einer Reinforcement Learning-basierten neuronalen adaptiven Geschwindigkeitsregelung vorgestellt. Ausgehend von einer Einleitung wurde der Stand der Technik und die zugrundeliegende Entwurfsmethodik zusammengefasst. Diese bezieht sich auf die modellbasierte Reglerauslegung und die systematische Auslegung von KNN. Letztere wurde in diesem Beitrag erstmalig als Neuheit vorgestellt.

Anschließend erfolgte die Anwendung und Validierung dieser Methodik am Beispiel einer neuronalen adaptiven Geschwindigkeitsregelung. Der Auslegungsprozess wurde detailliert beschrieben und die entstandene Funktion intensiv ausgewertet und anhand der Anforderungen validiert.

Zukünftige Arbeitsschritte umfassen die weitere Validierung der Entwurfsmethodik sowie die Auslegung weiterer automatisierter Fahrfunktionen.

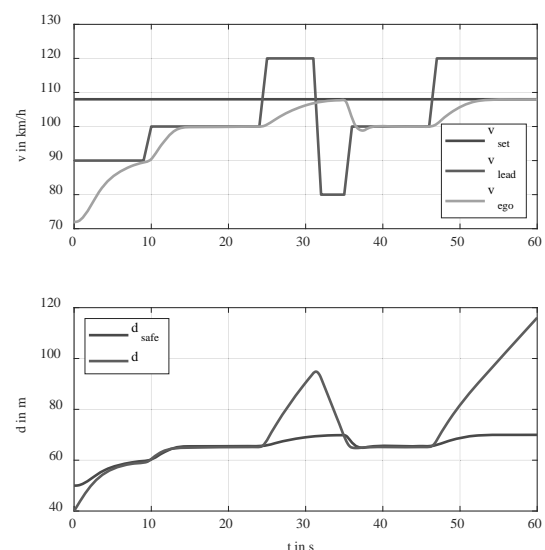


Abbildung 4: Simulationsergebnis der neuronalen adaptiven Geschwindigkeitsregelung

Danksagung

Diese Veröffentlichung entstand aus dem Teilprojekt "autoEVM" (*Ganzheitliches elektronisches Fahrzeugmanagement für autonome Elektrofahrzeuge*) (ZW 6-85030889), im Rahmen des Innovationsverbundes "auto-MoVe" (*Dynamisch konfigurierbare Fahrzeugkonzepte für den nutzungsspezifischen autonomen Fahrbetrieb*), welcher vom Europäischen Fonds für regionale Entwicklung (EFRE) gefördert und vom Projektträger NBank verwaltet wird.



EUROPÄISCHE UNION



Literatur

- [1] Milz S., Schrepfer J. Is artificial intelligence the solution to all our problems? Exploring the applications of AI for automated driving. In: Bertram T. (eds) *Automatisiertes Fahren 2019*. Springer Vieweg, Wiesbaden, 2020.
- [2] Schiekofe, P. et al. Maschinelles Lernen für das automatisierte Fahren. *ATZ Automobiltech Z*, vol. 121, 2019.
- [3] Liu-Henke X., Jacobitz S., Scherler S., Göllner M., Yarom O., Zhang J. A Holistic Methodology for Model-Based Design of Mechatronic Systems in Digitized and Connected System Environments. *16th International Conference on Software Technologies (ICSOT)*, online, 2021.
- [4] Lin Y.-C., Nguyen H.-L. T., Wang C.-H. Adaptive neuro-fuzzy predictive control for design of adaptive cruise control system. *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*, Calabira, Italy, 2017.
- [5] Abdullahi A., Akkaya S. Adaptive Cruise Control: A Model Reference Adaptive Control Approach. *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*, Sinaia, Romania, 2020.
- [6] Kiencke U., Nielsen L. *Automotive control systems: for engine driveline and vehicle*. Springer, Berlin, 2005.
- [7] Shakouri P., Ordys A. Nonlinear model predictive control approach in design of adaptive cruise control with automated switching to cruise control. *Control Engineering Practice*, vol. 26, 2014.
- [8] Anayor C., Gao W., Odekunle A. Cooperative Adaptive Cruise Control of A Mixture of Human-driven and Autonomous Vehicles. *SoutheastCon 2018*, St. Petersburg, FL, USA, 2018.
- [9] Tirumala S.S. Evolving deep neural networks using co-evolutionary algorithms with multi-population strategy. In: *Neural Comput & Applic*, vol. 32, 2020.
- [10] Duriez T., Brunton S., Noack B. R. *Machine Learning Control*. Springer International Publishing, Cham, Switzerland, 2017.
- [11] Lyu H. et. al. Esnet: Edge-Based Segmentation Network for Real-Time Semantic Segmentation in Traffic Scenes. *2019 IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, 2019.
- [12] Huang Z. et. al. Parameterized batch reinforcement learning for longitudinal control of autonomous land vehicles. *IEEE Trans. Syst., Man, Cybern, Syst.*, vol. 49, no. 4, 2019.
- [13] Liu-Henke X. et. al. Holistic development of a full-active electric vehicle by means of a model-based systems engineering. *2016 IEEE International Symposium on Systems Engineering (ISSE)*, Edinburgh, UK, 2016.
- [14] Yarom O. A. et. al. Artificial Neural Networks and Reinforcement Learning for model-based design of an automated vehicle guidance system. *12th International Conference on Agents and Artificial Intelligence (ICAART)*, Valletta, Malta, 2020.
- [15] Rock J., Roth W., Toth M., Meissner P., Pernkopf F. Resource-Efficient Deep Neural Networks for Automotive Radar Interference Mitigation. in *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, 2021.
- [16] Yarom O., Liu-Henke X. Development of a Simulation Environment for Automated Model Configuration for the Design and Validation of AI-Based Driving Functions. *11th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)*, online, 2021.

Model-based Development of an Automated and Remotely-Controlled Demolition Excavator

Christian Richter¹, Masoud Shakoorianfard¹, Volker Waurich¹, Frank Will¹

¹Professur für Baumaschinen, TU Dresden, Münchner Platz 3, 01069 Dresden, Germany; christian.richter1*, masoud.shakoorianfard*, volker.waurich*, frank.will@tu-dresden.de

Abstract. This paper describes the application of a 3D-simulation model to develop the control system of a demolition excavator. In order to simplify the system development and provide means for early prototype validation, a simulation-based development workflow is presented. A 3D model of the excavator, a demolishable wall, a virtual LiDAR sensor as well as CAN and Ethernet communication interfaces are used to support the development and testing of different software components as well as a test setup for automation strategies.

Introduction

In the case of nearby settlements and infrastructure, the demolition of natural draft cooling towers is restricted to non-explosive demolition techniques. One innovative approach is the usage of a modified excavator that sits on the wall structure and demolishes the wall piece by piece with a pulverizer. Figure 1 shows the excavator during operation. The excavator drives around the wall to demolish a 2 m high ring segment during each turn and reduce the height of the cooling tower up to a certain height. After this, the cooling tower can be collapsed in a controlled fashion by weakening the structure.



Figure 1: Demolition of the cooling tower in Mühlheim-Kährlich with the demolition excavator.

The excavator is currently operated from a platform on top of the cooling tower with a handheld remote control by sight, as depicted in Figure 2. Former attempts of controlling the excavator by video streams have been neglected because of image latency and poor depth perception. Due to these inefficient and dangerous operation conditions, an improved approach for a tele-remote workplace must be designed and implemented. In order to further facilitate the operation, suitable sub-tasks (e.g., the demolition of a wall segment) must be automated. The authors developed a simulation model of the machine and the process in order to support the product development even in the early stages.



Figure 2: Conventional operation of the excavator with a handheld remote control from a platform on top of the cooling tower.

This chapter gives an introduction to the basic machine setup and the corresponding simulation model. Chapter 1 describes the virtual LiDAR sensor and the point cloud processing. Chapter 2 presents the abilities to use the simulation model to validate CAN-communication and to act as a Hardware-in-the-Loop (HiL)-simulation. Chapter 3 outlines the automation strategy and how

this can be tested with the simulation model. Chapter 4 describes the developed approach for digital twin-based teleoperation. Chapter 5 closes the paper with a conclusion.

1 Machine and Simulation Setup

1.1 Equipment and Sensor Setup

The excavator consists of a frame structure including two driven wheel flanges that drive the whole machine on the edge of the cooling tower. On the frame, the working attachment of a conventional excavator is installed. During demolition work, the frame is clamped on the wall by two runners on both sides. The excavator has an adjustable boom and has four cylinders to adjust the height and range of the tool. The working attachment can be rotated by a slewing cylinder. The demolition tool is a pulverizer e.g., a massive gripper that cuts out a segment of the reinforced concrete. The tool itself can be rotated, opened, and closed.

In order to sense the working attachment, the main boom, the adjustable boom, the arm, and the tool coupler are equipped with 1D inclination sensors in order to compute the joint angles. The length of the slewing cylinder is sensed with a draw-wire sensor. An encoder inside the rotary feedthrough of the pulverizer captures the rotation angle. Pressure sensors determine the opening and closing state of the tool. The pitch and roll angle of the machine is captured by a 2D-inclination sensor. With this sensor setup, a complete kinematic representation is possible by calculating the forward kinematics.

A 3D-LiDAR sensor is used to capture the working process, i.e. the wall during demolition. A LiDAR is a laser-based distance sensor that is able to record a 3D point cloud of the surrounding. The LiDAR is attached to the frame and faces the inner face of the nearby wall that is in reach of the excavator. Figure 3 illustrates the sensors to sense the working attachment and the wall surface.

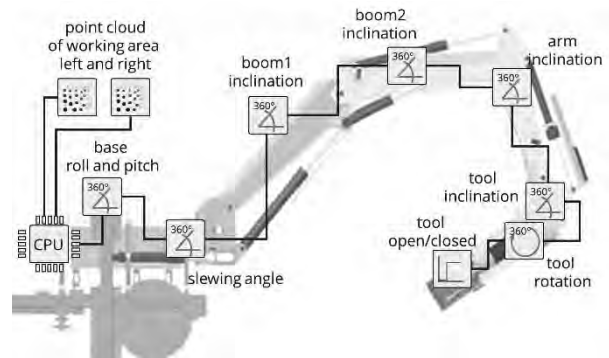


Figure 3 Sensor Setup of the Demolition Excavator

1.2 Simulation Setup

At the beginning of the project, the excavator was modeled as a CAD-Model. The excavator working attachment itself had to be reverse-engineered whereas the manufacturer has provided a CAD-Model of the frame. Based on the geometric information, a simulation model has been developed that serves the following purposes:

- Kinematic simulation to test reachability
- 3D-collision detection
- Simulation of a destructible wall volume
- Emulation of LiDAR sensors to retrieve virtual point cloud data of the machine and the variable wall geometry
- Software and hardware interfaces to enable HiL, SiL, and Co-simulation
- An appealing 3D visualization to reuse the model for a tele-remote operation
- Realtime Feedback
- Scriptable simulations for automated testing

As a simulation framework, the Unity Engine [1] was chosen since it provides an efficient 3D environment with a built-in physics engine, which is optimized for real-time applications. Compared to domain-specific simulation environments, it is relatively easy to implement special-purpose features such as specific virtual sensors or variable 3D geometries.

The serial kinematic of the machine is modeled as a kinematic tree that is visualized with reduced CAD models, as can be seen in Figure 4. The inputs to drive the model are the angles of the revolute joints that connect the rigid bodies of the working attachment. The visualization of hydraulic cylinders and rods moves accordingly. A kinetic multi-body simulation or hydraulic system simulation is not implemented as this is unnecessary for the application. A Co-Simulation to e.g., a Modelica-

Based Model via an FMU or interprocess communication would be possible as done by the authors in previous projects [2].

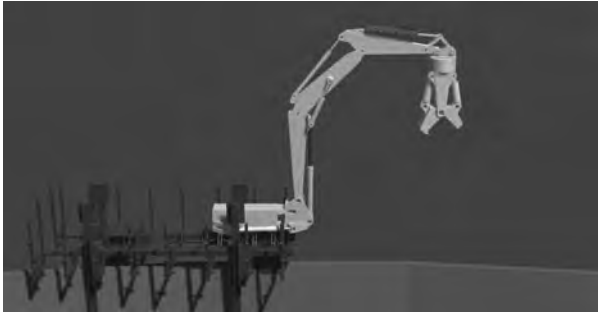


Figure 4 Visualization of the kinematic excavator model

2 Virtual LiDAR and Wall Sensing

To detect the wall segment in front of the excavator, a Livox Mid-70 LiDAR is used [4]. In general, a LiDAR is an array of laser-based distance measurement sensors that either rotate around an axis (rotary LiDAR) or are arranged in a matrix (flash LiDAR). The used LiDAR sensor is somehow unique in its scanning pattern. It features a non-repetitive scanning technology in the shape of a hypotrochoid (i.e., a small circle rolls inside a bigger one). Therefore, the coverage inside the field of view (FOV) increases over time.

In the simulated sensor model, the FOV, rotation speed, number of rays per shot, ratio of the circle radii are adjustable. In addition to the simulated LiDAR, a point cloud accumulator is implemented, that aggregates the measured point clouds over time. Figure 5 depicts the scanning patterns of the rotating hypotrochoid in the simulation and the accumulated point clouds over time.

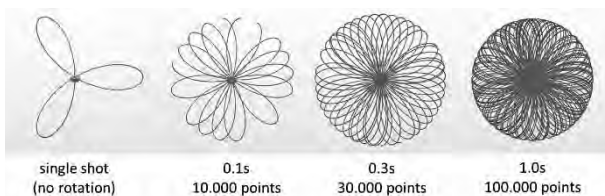


Figure 5 Scanning Pattern and accumulated point cloud over time.

The simulated sensor is able to provide the same point rate of 100.000 points/s as the real sensor on an i7-3930K @3.20GHz.

The simulation of the LiDAR utilizes the built-in

physics engine in the Unity software. Each distance measurement is done by calling the built-in function:

```
bool UnityEngine.Physics.Raycast(
    Vector3 origin,
    Vector3 direction,
    out RaycastHit hit,
    float maxDistance);
```

A Raycast outputs a RaycastHit object if the ray hits a so-called GameObject that has an instance of a collider-class. Therefore, every object that must be detected by the LiDAR needs to provide this ability. There are different collider classes available, both for geometric primitives (cubes, spheres, to name but two) and mesh-based colliders. For the detection of the deforming wall, an adaptive mesh-collider is needed. For the detection of the working attachment of the excavator, geometric primitives as bounding boxes of the complex geometry are sufficient. The base machine and the cooling tower do not feature a Collider object because these areas will be cropped out of the point clouds. Moreover, considering the interaction between the wall and the tool as the most important part of the simulation, it is required to differentiate the points that corresponded with the wall and tool from the other objects in the scene. This step, which is called Segmentation, is done by taking advantage of the kinematic information provided by sensors and accordingly moving colliders of the working attachment of the excavator, resulting in removing the points that corresponded with the machine objects except for the tool. Figure 6 shows the simulated point cloud on the wall segment and the tool.

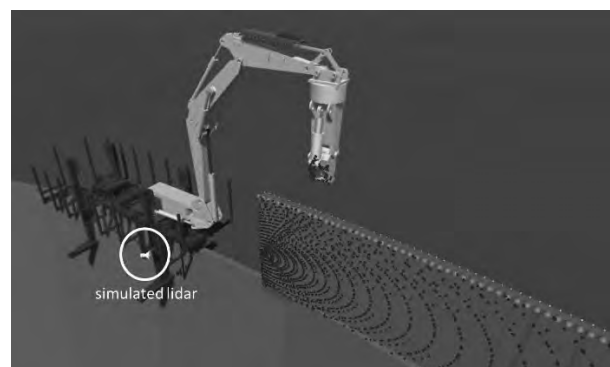


Figure 6 Simulated LiDAR with a wall segment that implements a mesh collider.

In order to simulate the demolition process of the wall, parts of the wall have to be removed. Therefore, a procedural and time-varying mesh of the wall segment is

implemented. The same mesh is used, both for visualization and to serve as a mesh collider for the simulated LiDAR.

The procedural wall model is initialized with a fixed height and depth at a specified position. At the position of the excavator tool, a cubic collider is attached, that has the size of the pulverizer chamber. If a wall vertex is hit by the tool collider, the top vertices of the mesh are shifted vertically to the bounding box of the collider. Figure 7 illustrates the deformation of the wall mesh by shifting the top vertices.

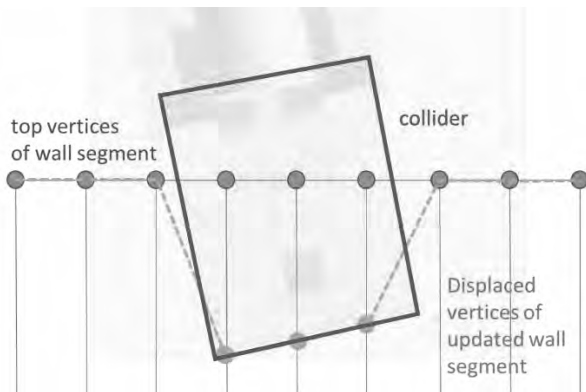


Figure 7 Wall deformation by the excavator tool

The result of a deformed wall segment reflecting the raycasts by the emulated LiDAR can be seen in Figure 8.

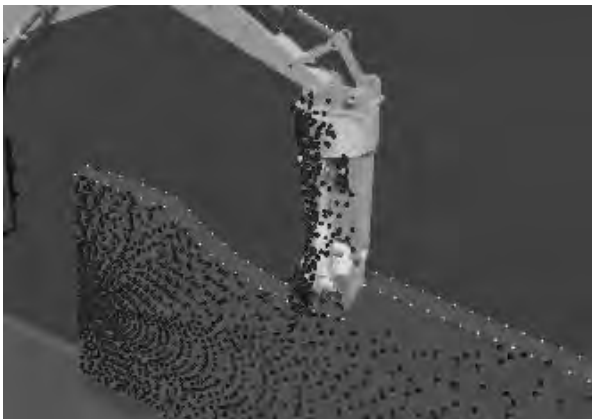


Figure 8 Deformed wall segment with updated mesh collider to reflect the raycasts of the LiDAR

The resulting point cloud data of the emulated LiDAR sensor is provided as a simple array of Cartesian coordinates referenced in the sensor coordinate system to be used for interprocess communication via shared memory. Additionally, a network interface is provided as well, fea-

turing the basic parts of the official Livox-communication protocol, resulting in the possibility to establish an emulated network stream to communicate with the commonly used tools officially provided by the manufacturer, including the official Livox-SDK, livox-ros-driver and livox-ros2-driver ROS packages, and the Livox Viewer software.

The point cloud data provided by the simulation must be processed as if it is provided by the real sensor installed on the machine. To do so, an industrial PC (IPC) and the ROS2 framework are utilized as the processing unit. This unit provides the required data for the visualization unit in the tele-remote-operation mode and implements the overall automation strategy in automation mode, both with the help of the processed data.

In order to have a more realistic data communication in the simulation of the process, the simulation PC and the IPC are networked, and the point cloud data is sent and received over the implemented emulated network stream. An instance of the simulation in Unity, ROS2, and Livox Viewer can be seen in Figure 9, which shows that the point cloud data is successfully sent from the Unity (simulation PC), received by ROS2 (IPC), and optionally visualized by Livox Viewer (IPC) which is the official software provided by the manufacturer to use with real sensors.

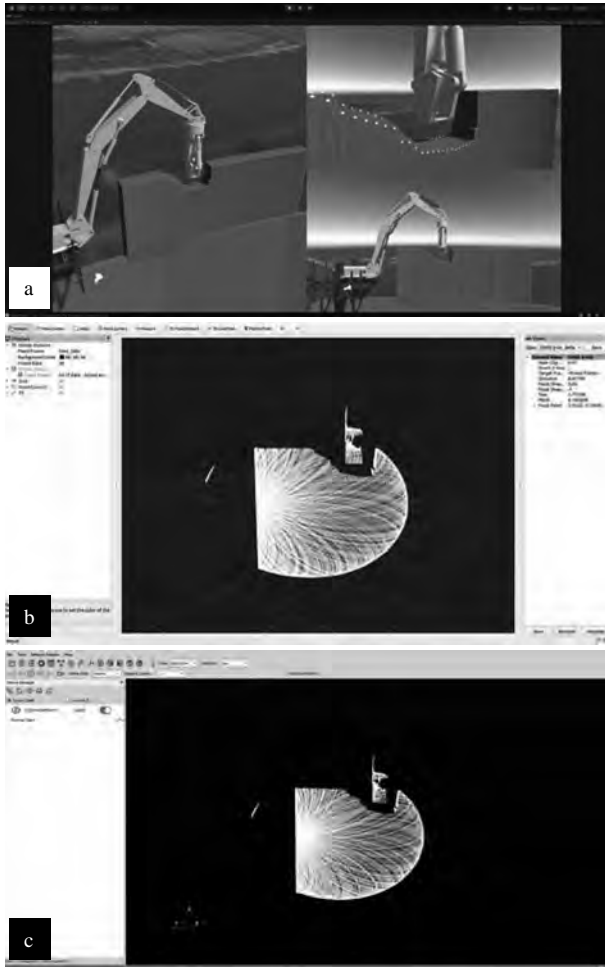


Figure 9 An instance of the simulation environment in: a) Unity software on the simulation PC, b) ROS2 framework on IPC, c) Livox Viewer on IPC. The point cloud data is accumulated for 3 seconds.

3 CAN-Communciation

The real machine is equipped with two CAN bus systems. The first CAN bus features the sensor data of the angle sensors. All the sensors attached to the working equipment are CANOpen based [5]. With the help of a USB-to-CAN interface and the corresponding API, the simulation model feeds its sensor data with a real CAN bus. A subset of the CANOpen-slave functionality is implemented in order to provide the same CAN-bus protocol as the real sensors in the expected operation mode. The second CAN bus serves as an interface between the ECU to control the hydraulic system and the IPC that processes the point-cloud data and implements the overall automation strategy. A custom CAN protocol has been developed to send the data of the desired trajectory from

the IPC to the ECU and to transmit status and control messages between the automation authority and the control ECU.

In order to test the CAN communication on the sensor bus as well as to test and modify the custom CAN protocol, the simulation model which implements the emulated CAN interface is a helpful tool. The emulation of the CAN communication which is fed by a comprehensive system simulation provides several advantages:

- Software development and interface specification can be done without the need of the real system.
- The simulation can be distributed several times whereas the real machine exists only once.
- Simulation-based but realistic data is a feasible way to provide a vast amount of test-cases for test-driven development.
- A simulator can facilitate the reconstruction of dangerous or complex situations.

4 Tele-Remote Operation

The machine is supposed to be controlled remotely. In previous applications, the machine was controlled with a handheld device with a direct line of sight. A camera-based solution to control the excavator from a control cabin has been rejected by the operators because of the high delay of the video streaming which makes the tele-operation inefficient. Besides the latency issue, camera-based tele-remote control solutions have several other drawbacks:

- The 2D image of a video hinders the depth perception of the operation which is important particularly when moving the working attachment of an excavator
- The camera position is fixed and does not allow a moving point of view.
- Optical cameras only work in good light conditions and in the absence of fog and dust.
- Compression and Decompression of digital video signals introduces additional delay and requires dedicated hardware
- Data throughput increases with resolution and number of cameras

In order to provide an efficient solution for tele-remote operation, a digital twin approach is implemented. All sensor signals are used to visualize a 3D model of the machine. The amount of data to represent all degrees of freedom of the excavator is comparatively low ($9 \times \text{Int16}$

at 50Hz). As the raw point cloud data stream requires a performant wireless link, a post-processing algorithm is implemented on the machine's IPC. This reduces the throughput to transmit the point data drastically ($300 \times \text{Int16}$ at 10Hz).

The basic idea to reduce the point cloud data is to extract only the relevant information to control the demolishing process, i.e. the top edge of the remaining wall. Different filter algorithms are applied to the point cloud to compute an array of points in a discrete spacing to indicate the top-right edge of the wall. Figure 10 shows the result of the edge detection algorithm based on the simulated LiDAR data.

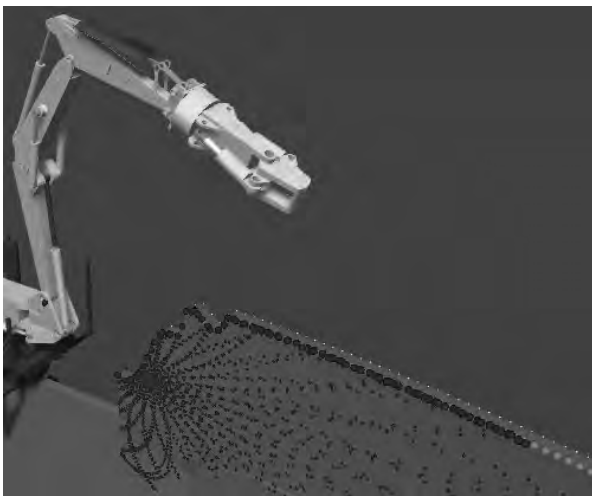


Figure 10 Edge detection for the simulated wall under demolition. The red spheres indicate the detected points of the top-right wall edge.

To provide a real-time tele-remote visualization, the 3D visualization of the simulator is reused to visualize the machine and point cloud data. Based on the detected points on the top-right edge of the wall, a mesh is created that resembles the real wall. Considering the fixed wall thickness and using radial extrapolation of the given points, a textured volume can be constructed to visualize the wall segment within reach of the excavator.

This approach to visualize this particular machine-process interaction has several advantages:

- The overall amount of data to visualize all relevant information is relatively low compared to multiple high-definition video streams. This reduces delay.
- The virtual camera that is used to render the scene can be moved by the user during operation or several virtual cameras can be used to visualize the scene from

different locations.

- Additional information can be computed and displayed in the scene, of which position coordinates of the end effector, limits of the working area, recommendations to align the tool efficiently, to name but a few, are of great importance.
- To enhance depth perception of the scene, the visualization could be rendered for a VR-Headset. As this is not feasible for long operations, a head tracking-based camera movement has been tested to align the view with the head movement of the operator and to create motion parallax.

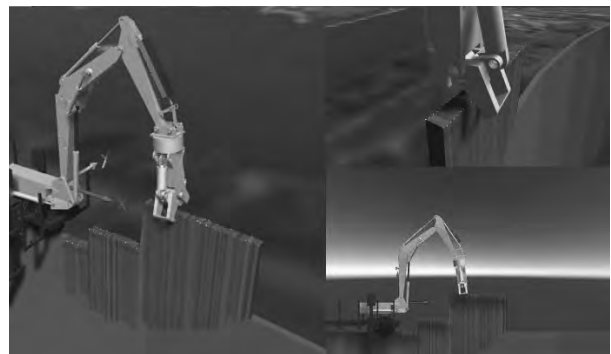


Figure 11 Visualization of real machine data and real sensor data for tele-remote operation with 3 different camera positions. The left bottom image shows a side view with an orthographic projection

5 Conclusion

This paper describes the components of a 3D-simulation environment for a demolition excavator, which can also serve as a visualization tool for a real-time tele-remote operation. The simulation comprises a 3D kinematic model for the excavator, an emulated Livox LiDAR, CAN bus-based interfaces for Hardware-in-the-loop applications, and a deformable wall segment that resembles a demolishable wall. The simulation is a helpful tool to develop and test the communication protocol, the software for automation and machine control as well as the point cloud processing. The majority of the simulation environment has been reused to serve as a real-time visualization tool for a tele-remote operation that has several advantages compared to a conventional camera-based view of the process.

In the ongoing project, a partly automation of the demolition process has to be developed. Therefore, a tra-

jectory planner and a process strategy have to be implemented and tested. As the real demolition of a wall is a laborious and costly procedure, the simulation-based development of the process routine is an invaluable method.

Acknowledgement

This Project is supported by the Federal Ministry for Economic Affairs and Energy (BMWi) on the basis of a decision by the German Bundestag.

References

- [1] Unity Technologies, 14 10 2021. [Online]. Available: www.unity.com.
- [2] V. Waurich und J. Weber, „Interactive FMU-Based Visualization for an Early Design Experience,“ in *Proceedings of the 12th international Modelica Conference*, Prague, 2017.
- [3] Livox Technology, [Online]. Available: <https://www.livoxtech.com/de/mid-70>. [Zugriff am 15 10 2021].
- [4] CiA, „www.can-cia.org,“ [Online]. Available: <https://www.can-cia.org/canopen/>. [Zugriff am 26 10 2021].
- [5] V. Waurich, „Simulation der automatisierten Schaufelbefüllung eines Radladers mit Modelica,“ in *Tagungsband ASim Workshop STS/GMMS*, Vienna, 2018.

Efficient state estimation for gas pipeline networks via low-rank approximations

Nadine Stahl^{1*}, Nicole Marheineke¹,

¹Chair of Modeling and Numerics, FB IV – Mathematics, Trier University, D-54286 Trier, Germany

*nadine.stahl@uni-trier.de

Abstract. In this paper we investigate the performance of projection-based low-rank approximations in Kalman filtering. For large-scale gas pipeline networks structure-preserving model order reduction has turned out to be an advantageous way to compute accurate solutions with much less computational effort. For state estimation we propose to combine these low-rank models with Kalman filtering and show the advantages of this procedure to established low-rank Kalman filters in terms of efficiency and quality of the estimate.

Introduction

Efficient state estimation plays an important role for model-predictive control of gas pipeline networks, [1, 8, 18]. Gas network models are large-scale systems of high complexity. They consist of nonlinear partial differential equations coupled with algebraic constraints, whose simulation is computationally very expensive. To reduce the effort, model hierarchies have been established [3] and, recently, structure-preserving model order reduction has been developed [6, 14]. The reduced (low-rank) models are in particular mass-conserving and energy-dissipating.

In state estimation Kalman filtering is popular. The Kalman Filter, developed in the 1960s [10], is still one of the most prominent filtering algorithms because of its optimality property for linear systems. Since it is not suitable for large-scale systems due to computational reasons, low-rank Kalman filters have been developed, see [16] and references within. They reduce the computational effort by simplifying the costly and memory-demanding calculations of the error covariances in the filtering process. For example, the Compressed State Kalman Filter [12] reduces the covariance matrices via projection techniques and the Spectral Kalman Filter [7] by help of a Taylor expansion, whereas the Ensemble Kalman Filter [9] considers a sample covariance by

collecting an ensemble of possible states with a Monte-Carlo method. Recent work addresses the use of a reduced-order model in the filtering process for linear and nonlinear parabolic systems [5, 17].

In this paper we investigate the use of a structure-preserving reduced (low-rank) model in Kalman filtering for gas networks. Since a reduced nonlinear model has proven to be still too computationally expensive in the application [19], we consider a linearized model variant. We deduce our low-rank approximation by projection-based model order reduction. Our approach has hence similarities with the Compressed State Kalman Filter being a projection-based low-rank filter. However, we show that our approach is significantly superior to established low-rank Kalman filters in terms of computational speed and memory demands, while the obtained state estimates are of competitive accuracy. We also study the combination of a low-rank model with a low-rank filter.

The paper is structured as follows: After introducing a model hierarchy for gas pipeline networks and the filtering in Sec. 1, we present our Kalman filtering approach with a reduced model and discuss its similarities and differences to the Compressed State Kalman Filter in Sec. 2. A numerical performance study for the low-rank approximations – in terms of quality and efficiency – is carried out in Sec. 3, using an academic benchmark example as well as a real gas pipeline network from western Germany.

1 Problem setting

Modeling a gas pipeline network, we proceed from the nonlinear isothermal Euler equations and establish a model hierarchy by help of linearization and structure-preserving reduction.

1.1 Hierarchical gas network models

The gas pipeline network is described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the pipes are represented by the edges \mathcal{E} and the junctions \mathcal{V}_I and inlets/outlets \mathcal{V}_B by the nodes \mathcal{V} , in particular $\mathcal{V} = \mathcal{V}_I \cup \mathcal{V}_B$, $\mathcal{V}_I \cap \mathcal{V}_B = \emptyset$. The gas dynamics in a pipe e of length l^e is modeled by nonlinear isothermal Euler-type equations for pressure p^e and mass flux q^e , i.e., for $e \in \mathcal{E}$, $(x, t) \in [0, l^e] \times [0, T]$,

$$a^e \partial_t p^e = -\partial_x q^e, \quad b^e \partial_t q^e = -\partial_x p^e - d^e \frac{|q^e|}{p^e} q^e, \quad (1)$$

with constant pipe parameters a^e, b^e, d^e and coupled via Kirchhoff's conditions in every junction $v \in \mathcal{V}_I$

$$\sum_{e \in \delta_v^-} q^e(l^e, t) = \sum_{e \in \delta_v^+} q^e(0, t),$$

$$p^e(l^e, t) = p^v(t), e \in \delta_v^+, \quad p^e(0, t) = p^v(t), e \in \delta_v^-,$$

where δ_v^+, δ_v^- denote the sets of all topologically ingoing and outgoing edges to v . At inlets/outlets $v \in \mathcal{V}_B$ we prescribe the pressure p^v as

$$p^v(t) = u^v(t) = u_D^v(t) + u_S^v(t),$$

$$du_S^v = \kappa^v(\mu^v - u_S^v) dt + \sigma^v dW_t^v.$$

The time-varying input u^v consists of a deterministic part u_D^v and a stochastic one u_S^v , where we particularly model u_S^v by an Ornstein-Uhlenbeck process with constant parameters $\kappa^v, \mu^v, \sigma^v$ and a standard Wiener process W_t^v . The stochastic boundary data allows the incorporation of, e.g., market strategies or the concept of Power-to-Gas [4]. Certainly, also other boundary types can be considered, see e.g., [14]. We initialize the transient model with the stationary solution associated to the unperturbed input $u^v = u_D^v(0)$, $v \in \mathcal{V}_B$. A linear model variant is obtained when considering a simplified linearized friction term in (1) with respective constant d_l^e ,

$$a^e \partial_t p^e = -\partial_x q^e, \quad b^e \partial_t q^e = -\partial_x p^e - d_l^e q^e. \quad (2)$$

For an overview on further hierarchical models see, e.g., [3].

1.2 Structure-preserving reduced models

The unperturbed network models can be embedded in a port-Hamiltonian framework and allow for structure-

preserving and robust spatial Galerkin approximations. The Galerkin approximation is applicable to finite element discretization and projection-based model order reduction. Under mild assumptions on the ansatz spaces, mass conservation and energy dissipation are ensured [6, 14, 15].

As in [6], we use mixed finite elements and the moment matching method proposed there for the network model (2), yielding linear time-invariant descriptor systems of the form

$$\mathbf{E} \dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B}(\mathbf{u}_D + \mathbf{u}_S), \quad (3a)$$

$$\mathbf{y} = \mathbf{C} \mathbf{x}, \quad (3b)$$

$$d\mathbf{u}_S = \mathbf{R}(\boldsymbol{\mu} - \mathbf{u}_S) dt + \boldsymbol{\Sigma} d\mathbf{W}_t. \quad (3c)$$

The input functions u_D, u_S account for the boundary conditions with diagonal matrices $\mathbf{R}, \boldsymbol{\Sigma} \in \mathbb{R}^{|\mathcal{V}_B| \times |\mathcal{V}_B|}$ containing the entries κ^v and σ^v , respectively, and $\boldsymbol{\mu} = (\mu^v)_{v \in \mathcal{V}_B}$. The output $\mathbf{y}(t) \in \mathbb{R}^R$ with certain pressure or flux values corresponds to measurement data in the context of filtering, $t \in [0, T]$. In case of the full order model the state $\mathbf{x}(t) \in \mathbb{R}^N$ represents the space-discrete pressure and flux, its size is determined by the number of pipes and finite elements for each pipe. In the model reduction the system structure is kept. We introduce an orthonormal projection matrix $\mathbf{V} \in \mathbb{R}^{N \times n}$, $n \ll N$ with the property $\mathbf{x} \approx \mathbf{V} \hat{\mathbf{x}}$, i.e., $\mathbf{V}^T \mathbf{V} = \mathbf{I}$, \mathbf{I} identity. The reduced quantities are indicated by $\hat{\cdot}$, the system matrices particularly read

$$\hat{\mathbf{E}} = \mathbf{V}^T \mathbf{E} \mathbf{V}, \quad \hat{\mathbf{A}} = \mathbf{V}^T \mathbf{A} \mathbf{V}, \quad \hat{\mathbf{B}} = \mathbf{V}^T \mathbf{B}, \quad \hat{\mathbf{C}} = \mathbf{C} \mathbf{V}. \quad (4)$$

Note that we preserve a block structure by setting up the reduced spaces for pressure V_p and flux V_q separately. Imposing a compatibility condition on the spaces, $\partial_x V_p = V_q$, ensures stable reduced models [6, 15].

1.3 Stochastically forced model for filtering

The state estimation is based on a filtering model, for which we use the linear spatially discretized network model in the full or a reduced version. Uncertainties are incorporated via a system noise being modeled as (driving) uncorrelated Gaussian process in (3a).

Let $\mathbf{x}(t) = (\mathbf{x}^T, \mathbf{u}_S^T)^T(t) \in \mathbb{R}^{N+|\mathcal{V}_B|}$, $t \in [0, T]$, we consider equidistant time points $t_k = k\tau$, $\tau = T/K$, $k = 0, \dots, K$, then our filtering model in time-discrete

form reads

$$\begin{aligned} \mathbf{x}_{k+1} &= \Phi \mathbf{x}_k + \Psi \mathbf{u}_k + \mathbf{w}_k \\ \Phi &= \begin{pmatrix} \mathbf{A}_\tau^{-1}(\mathbf{E} + \tau(1-\theta)\mathbf{A}) & \tau\mathbf{A}_\tau^{-1}\mathbf{B} \\ (\mathbf{I} + \tau\mathbf{K})^{-1} \end{pmatrix} \\ \Psi &= \begin{pmatrix} \tau\mathbf{A}_\tau^{-1}\mathbf{B} \\ \tau(\mathbf{I} + \tau\mathbf{K})^{-1}\mathbf{K}\mu \end{pmatrix}. \end{aligned} \quad (5)$$

with $\mathbf{A}_\tau = \mathbf{E} - \tau\theta\mathbf{A}$. Here, \mathbf{x}_k denotes the approximation $\mathbf{x}_k \approx \mathbf{x}(t_k)$, and $\mathbf{u}_k = ((\theta\mathbf{u}_D(t_{k+1}) + (1-\theta)\mathbf{u}_D(t_k))^T, \mathbf{1}^T)^T$ the respective input at time t_k with vector of ones $\mathbf{1} \in \mathbb{R}^{|\mathcal{I}_B|}$. The underlying time-integration is based on a θ -scheme for (3a) and an Euler-(Maruyama)-scheme for (3c). Our different treatment is motivated by the observation that the Euler-scheme is sufficient for the computation of the boundary data, whereas the more sophisticated θ -scheme allows for a better capturing of the dynamic behavior of the flow quantities. The state noise $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ is a normal distributed centered random variable with constant diagonal covariance matrix \mathbf{Q} for every time t_k , in particular

$$\mathbf{Q} = \tau \begin{pmatrix} \mathbf{Z}\mathbf{Z}^T & \\ & \Sigma\Sigma^T \end{pmatrix}.$$

It results from the system noise with amplitude $\mathbf{Z} \in \mathbb{R}^{N \times N}$ (diagonal matrix) added to (3a) and the scaled Wiener process in (3c).

The output is assumed to be measurable with suitable devices. To account for measurement errors, noise is added to the output equation (3b), i.e.,

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (6)$$

with \mathbf{y}_k output at time t_k and $\mathbf{H} = (\mathbf{C}, \mathbf{0})$. The discrete time noise $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$, with constant covariance matrix $\mathbf{R} \in \mathbb{R}^{R \times R}$, is assumed to be uncorrelated in time and to be component-wisely independent of \mathbf{w}_k . In this study the measurement data $\{\mathbf{y}_k\}$ for the filtering is provided by the outputs of the nonlinear network model (1).

Using the reduced-order model for filtering, the system matrices are replaced by their reduced counter parts, yielding $\hat{\Phi}$, $\hat{\Psi}$ and $\hat{\mathbf{H}}$ in (5) and (6). Since the input process is not affected by the model order reduction, the reduced covariance of the state noise is given by $\hat{\mathbf{Q}} = \mathbf{V}_x^T \mathbf{Q} \mathbf{V}_x$ with

$$\mathbf{V}_x = \begin{pmatrix} \mathbf{V} \\ \mathbf{I} \end{pmatrix}. \quad (7)$$

2 Low-rank approximations

The focus of this work is on Kalman filtering algorithms, since our filtering model is linear. The Kalman Filter (KF) is known to be optimal for linear systems with white noise as state and measurement noise in the sense that it is unbiased and minimizes the error variance [10]. However, it is not applicable to large-scale systems for computational reasons. We investigate a filtering approach based on projection-based model order reduction involving low-rank models and discuss its relation to the established Compressed State Kalman Filter [12, 13] being a projection-based low-rank Kalman filter.

Kalman filtering consists of a prediction and a correction step. In each of these steps a state estimate and its error covariance are computed. At time point t_k , the state and the error covariance matrix of the prediction step are denoted by $\mathbf{x}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$, where the first index indicates the current time point t_k and the second index stands for the time up to which measurement data is considered, i.e., t_{k-1} . In the correction step additional measurement data of the current time point is taken into account, yielding the state estimate $\mathbf{x}_{k|k}$ and the corrected error covariance matrix $\mathbf{P}_{k|k}$. The correction ratio between prediction and correction is given by the so-called Kalman gain \mathbf{K}_k , see Algorithm 1.

2.1 Kalman Filter using reduced-order models

In the Kalman Filter the computational effort comes from the determination of the error covariances (and the Kalman gains) being fully occupied matrices of the system's size. The matrix evaluations become memory-intensive and expensive for large-scale systems. Establishing a model hierarchy, we derive a low-rank approximation from projection-based model order reduction. Using a low-rank model to estimate the state and to compute the error covariances significantly reduces the effort and makes the Kalman Filter applicable. Prolongating the resulting low-rank estimate to the high-dimensional space yields then the desired state estimate for the large-scale system.

The Kalman Filter applied to the reduced-order model (RKf) gives an estimate for the reduced state $\hat{\mathbf{x}}_{k|k}$ and its associated error covariance matrix $\hat{\mathbf{P}}_{k|k}$ of small size. To get the quantities $\mathbf{x}_{k|k}$ and $\mathbf{P}_{k|k}$ of the full order (large-scale) system, we exploit the linearity of the expectation value and the approximation property of the reduced state $\mathbf{x} \approx \mathbf{V}_x \hat{\mathbf{x}}$. This leads to $\mathbf{x}_{k|k} \approx \mathbf{V}_x \hat{\mathbf{x}}_{k|k}$ and

Algorithm 1: Kalman Filter

```

// initialization
1  $\mathbf{x}_{0|0} = \mathbf{x}_0, \mathbf{P}_{0|0} = \mathbf{P}_0;$ 
2 for  $k = 0, \dots, K-1$  do
    // prediction
3    $\mathbf{x}_{k+1|k} = \Phi \mathbf{x}_{k|k} + \Psi \mathbf{u}_k;$ 
4    $\mathbf{P}_{k+1|k} = \Phi \mathbf{P}_{k|k} \Phi^T + \mathbf{Q};$ 

    // correction
5    $\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1|k} \mathbf{H}^T + \mathbf{R})^{-1};$ 
6    $\mathbf{x}_{k+1|k+1} = \mathbf{x}_{k+1|k} + \mathbf{K}_{k+1} (\mathbf{y}_{k+1} - \mathbf{H} \mathbf{x}_{k+1|k});$ 
7    $\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \mathbf{P}_{k+1|k};$ 
8 end
    
```

$$\mathbf{P}_{k|k} \approx \mathbf{V}_x \hat{\mathbf{P}}_{k|k} \mathbf{V}_x^T.$$

2.2 Compressed State Kalman Filter

The idea of the Compressed State Kalman Filter (CSKF) to overcome the computational complexity of the (classical) Kalman Filter is to replace the error covariance matrix with a low-rank approximation. Thus, it is assumed that there exists a constant orthonormal projection matrix \mathbf{V}_P , $\mathbf{V}_P^T \mathbf{V}_P = \mathbf{I}$, such that $\mathbf{P}_{k|l} \approx \mathbf{V}_P \tilde{\mathbf{P}}_{k|l} \mathbf{V}_P^T$ holds for both the predicted ($l = k-1$) and the corrected error covariance matrix ($l = k$) with $\tilde{\mathbf{P}}_{k|l}$ being of much smaller size. For the filter matrices, low-rank versions are precomputed offline, i.e., $\tilde{\Phi} = \mathbf{V}_P^T \Phi \mathbf{V}_P$, $\tilde{\mathbf{H}} = \mathbf{H} \mathbf{V}_P$ and $\tilde{\mathbf{Q}} = \mathbf{V}_P^T \mathbf{Q} \mathbf{V}_P$, that are then used for the determination of the low-rank error covariance $\tilde{\mathbf{P}}_{k|k}$ and the respective low-rank Kalman gain $\tilde{\mathbf{K}}_k$ in each time step (cf., Algorithm 1, lines 4, 5 and 7). The calculations for the state estimate are performed with the full-rank filter matrices. They stay the same as in the (classical) Kalman Filter, except for prolongating the Kalman gain to the high-dimensional space ($\mathbf{K}_k \approx \mathbf{V}_P \tilde{\mathbf{K}}_k$).

2.3 Relation between projection-based filtering variants

The Kalman Filter on a reduced-order model and the Compressed State Kalman Filter seem to be very similar at first glance, since they both deal with projection-based low-rank approximations. However, there exist severe differences.

CSKF still uses the large-scale system for the computation of the state estimates, while the reduction

only affects the error covariance matrices and thus the Kalman gain. In contrast, RKF uses a low-rank model for the entire filtering process, including the state computation. This is expected to further improve the computational speed, but probably at the cost of accuracy compared to CSKF. For a numerical study on this point see Sec. 3.

As far as the projection is concerned, there exist multiple techniques in the field of model order reduction on how to construct the projection matrix \mathbf{V}_x for a low-rank model, e.g., with Moment Matching, Balanced Truncation or Proper Orthogonal Decomposition (see, e.g., [2] for an overview). Usually, the special structure of the underlying large-scale system (differential, algebraic or additional input equations) is taken into account to achieve the best possible approximation quality. In CSKF, the choice of \mathbf{V}_P is not obvious, as it exclusively acts on a statistical quantity (error covariance matrix). In the following investigations we use $\mathbf{V}_x = \mathbf{V}_P$.

Applying the same projection matrix, one might expect the same low-rank approximations for the error covariances in both filtering variants when starting with the same initialization. But this is in general not the case, as a closer look at their computation in the prediction step reveals. We find

$$\begin{aligned} \hat{\mathbf{P}}_{k+1|k} &= \hat{\Phi} \hat{\mathbf{P}}_{k|k} \hat{\Phi}^T + \hat{\mathbf{Q}} \quad \text{for RKF,} \\ \tilde{\mathbf{P}}_{k+1|k} &= \tilde{\Phi} \tilde{\mathbf{P}}_{k|k} \tilde{\Phi}^T + \tilde{\mathbf{Q}} \quad \text{for CSKF.} \end{aligned}$$

While the state noise covariances satisfy $\tilde{\mathbf{Q}} = \hat{\mathbf{Q}}$, the filter state matrices differ, $\hat{\Phi} \neq \tilde{\Phi}$, in case of an underlying implicit time-discretization. See, e.g., the first diagonal block matrix for our pipe network model and the projection matrix (7),

$$\begin{aligned} \hat{\Phi}_{11} &= (\mathbf{V}^T (\mathbf{E} - \tau \theta \mathbf{A}) \mathbf{V})^{-1} \mathbf{V}^T (\mathbf{E} + \tau (1 - \theta) \mathbf{A}) \mathbf{V} \\ &\neq \mathbf{V}^T (\mathbf{E} - \tau \theta \mathbf{A})^{-1} (\mathbf{E} + \tau (1 - \theta) \mathbf{A}) \mathbf{V} = \tilde{\Phi}_{11}. \end{aligned}$$

In this context, the projection might be interpreted as a model reduction before or after time-discretization for RKF and CSKF, respectively.

2.4 Reduced Ensemble Kalman Filter

Additionally to the projection-based approaches, we consider the Ensemble Kalman Filter on a reduced-order model (REnKF) for comparison reasons in Sec. 3. The Ensemble Kalman Filter is probably the most used

Kalman filter variant for large-scale systems, [9]. This filtering algorithm avoids the costly evaluation of the algebraic expression of the error covariance matrix in the prediction step (matrix-matrix-matrix product of full occupied matrices of system's size, cf. Algorithm 1, line 4). Applying a Monte-Carlo method, the error covariance is numerically approximated on basis of an ensemble of samples of the stochastic filter model. The calculation of Kalman gain and corrected error covariance follows the procedure of the classical Kalman Filter. The state estimate is given as ensemble average.

This procedure usually requires only a very small amount of memory, as not the whole covariance matrix, but only a few samples, whose numbers are usually much less compared to the state dimension, have to be stored. Moreover, there exist observation matrix-free implementations. As for computational time, there is not much gain to be expected from parallelizing the ensemble filter, since collecting all samples after each time step involves a massive overhead. The Ensemble Kalman Filter can be combined straightforward with a reduced-order (low-rank) model, as filter model and filter technique are independent of each other.

3 Numerical investigations

We investigate the performance of the low-rank filtering techniques in terms of accuracy of the state estimates and computational effort. As gas pipeline networks we consider an academic example of small size (diamond topology) for which the classical Kalman Filter is still applicable as well as a large-scale real network from western Germany. As measurement data we take the mass flux values at the boundary nodes obtained from the nonlinear network model which also serves as reference. For REnKF we always use 100 samples in accordance to the recommendations in [9]. All computations are performed in MatlabR2017b on an Intel Xeon with 2.2GHz using 12 Cores.

3.1 Academic diamond network

The small diamond network of 7 edges and 6 nodes serves as academic benchmark setting, see Fig. 1. The pipe parameters are taken from [4]. For the linear model we use $d_l^e = d^e |q_{av}^e| / p_{av}^e$, where p_{av}^e and q_{av}^e are the average pressure and mass flux on edge e of the stationary solution associated to the input $u^v = u_D^v(0)$, $v \in \mathcal{V}_B$. In the subsequent example each pipe is discretized with

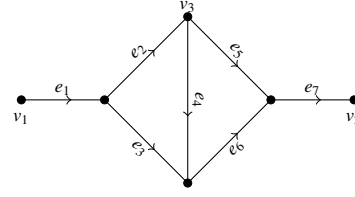


Figure 1: Academic example: diamond network topology.

250 equidistant finite elements, yielding a full order system with $N = 3511$ degrees of freedom. The reduced model is of size $n = 29$, which implies a low-rank filter model of size $n + |\mathcal{V}_B| = 31$ when adding the two equations for the boundary data. The reduction error is of order $\mathcal{O}(10^{-4})$. Concerning the filter model, the system noise is exclusively added to the dynamic equations, it is set up with respect to the maximal deviation of the stationary solution from its average values (pressure, mass flux) on each pipe. The algebraic constraints are not perturbed, as they describe mass conservation across junctions. Note that they are preserved under the applied model order reduction. The measurement noise accounts for 1% deviation from the maximal measured flux values. As input functions we choose

$$u_D^{v_1}(t) = \begin{cases} 2+t, & 0 \leq t < 1, \\ 3, & 1 < t < 5, \\ 1.5 - 0.1t, & 5 \leq t < 10, \\ 2, & t \geq 10, \end{cases} \quad u^{v_2}(t) \equiv 2,$$

$$(\mu, \kappa, \sigma)^{v_1} = (0, 3, 0.2)$$

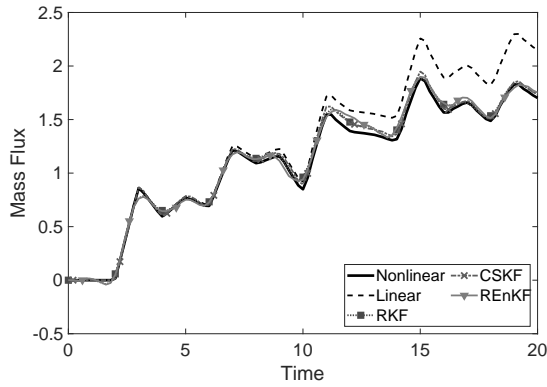
for $t \in [0, 20]$. We apply 1000 time steps and set $\theta = 0.51$.

Figure 2 illustrates the temporal evolution of the mass flux at the inner node v_3 on edge e_2 (cf., Fig. 1) that is obtained from the nonlinear network model (1) (reference solution) and from the linear model (2) as well as is estimated by the different low-rank filtering techniques. Although the estimates are computed on basis of the linear model whose solution differs strongly from the nonlinear one, they capture the nonlinear behavior being included via the measurements very well. Regarding the estimation errors (i.e., temporal mean of the relative spatial L^2 -errors wrt. reference) in Table 1, the (classical) Kalman Filter shows the best approximation properties. CSKF achieves a comparable error, whereas the errors of the other low-rank filters (Kalman and Ensemble Filter on reduced-order model)

Table 1: Errors and CPU time for diamond network

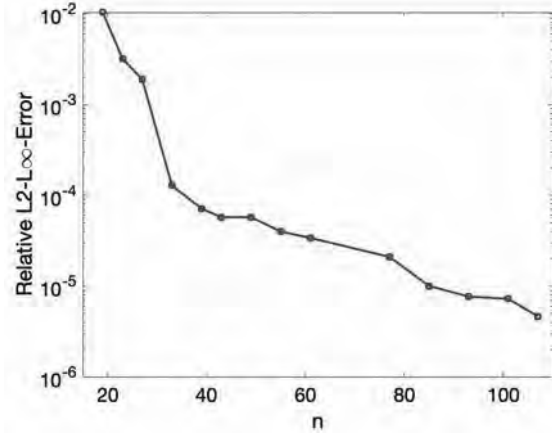
Filter	$\text{mean}_j \frac{\ \mathbb{E}[\mathbf{x}_j - \hat{\mathbf{x}}_{j,j}]\ _{L^2}}{\ \mathbb{E}[\mathbf{x}_j]\ _{L^2}}$	Offline [s]	Online [s]
KF	$3.7 \cdot 10^{-2}$	$8.4 \cdot 10^2$	$5.2 \cdot 10^1$
RKF*	$6.0 \cdot 10^{-2}$	$0.2 \cdot 10^0$	$7.4 \cdot 10^{-2}$
CSKF	$3.8 \cdot 10^{-2}$	$2.1 \cdot 10^0$	$1.5 \cdot 10^0$
REnKF* \diamond	$9.8 \cdot 10^{-2}$	$1.2 \cdot 10^{-1}$	$3.2 \cdot 10^{-1}$

* additional post-processing for prolongation is required

 \diamond offline time refers to sampling of random variables**Figure 2:** Simulation and estimation results for mass flux $q^{e2}(v_3)$, cf., Fig. 1.

are of same order but slightly worse. The better accuracy of CSKF towards the other low-rank filters might be expected as it estimates the states on basis of the large-scale model. However, all results are astonishingly good regarding the underlying model order reduction error (cf., Fig. 3).

Concerning the computational time, RKF – but also REnKF – clearly outperforms the filters operating on the large-scale model by several orders of magnitude, Table 1. We distinguish between offline (pre-computation) and online (actual runtime) phases. For the (classical) Kalman Filter as well as its projection-based low-rank variants, the error covariance matrices and the Kalman gains are precomputed offline. So, the runtime only consists of the state estimation taking into account the actual inputs and measurements. In the ensemble filtering this pre-computation of covariances and Kalman gains is not possible, since all quantities depend on the chosen samples. Changing input or measurement data hence requires a completely new filtering process. The effort of REnKF to approximate the sam-

**Figure 3:** Model order reduction error for p and q (L^2 in space, L^∞ in time) over system size.

ple covariance is slightly higher as running the online phase of RKF. REnKF scales here with the number of samples. As a low-rank filtering method operating on a low-rank model, better performance could possibly be achieved by tuning. For sophisticated tuning of the filter, i.e., reducing the required sample size for a comparatively good approximation and achieving better run-times, we refer to [9, 11]. The offline time of REnKF listed in Table 1 refers to the sampling of the random variables which can become time-consuming for large vectors in Matlab. Note that filtering on a reduced-order model requires an additional post-processing step, i.e., prolongation of the reduced variables, to obtain the estimate for the high-dimensional state and its error covariance.

3.2 Large-scale real pipeline network

The partDE network from western Germany consists of 636 pipes connected in 487 junctions with 47 in-/outlets, see Fig. 4. The pipes have a maximal length of about 120 km, the specific pipe parameters can be found in [20]. In the subsequent example each pipe is discretized equidistantly with a maximal element length of 100 m, yielding a full order model of dimension $N = 86160$. The reduced model is of size $n = 1356$, the reduction error is of order $\mathcal{O}(10^{-4})$. As input we prescribe the pressure in bar, particularly we impose

$$\begin{aligned} u_D^{v_1}(t) &= 60 + 5 \sin(0.03t), \\ u_D^{v_2}(t) &= 70 + 7 \cos(0.1t), \end{aligned}$$



Figure 4: Gas pipeline network from western Germany "partDE" [20].

$$\begin{aligned} u_D^{v3}(t) &= 65 + 2 \sin(0.05t), \\ u_D^{v4}(t) &= 80 + 4 \sin(0.008t), \\ u_D^{v5}(t) &= 55 + 5 \sin(0.017t), \end{aligned}$$

$(\mu, \kappa, \sigma)^{v_i} = (0, 3, 0.2)$, $i = 1, \dots, 5$, at the boundary nodes marked in Fig. 4 and $u^{v_j} \equiv 60$ at all others, $j = 6, \dots, 47$. A time horizon of 12 hours is covered by 720 time steps of 1 minute length. The other model parameters are set and the computations are performed in the same way as in the academic example.

This real large-scale network makes it necessary for computational reasons to use low-rank approximations for state estimation. The performance of the filters is as in the academic example, see Table 2. Compared to filtering with the reduced-order (low-rank) model, the approximation quality of CSKF is slightly better (by less than one order) since the state is estimated on the large-scale model. The price to pay, however, is a significantly longer computing time. An additional drawback, getting more importance the larger the full order model becomes, is the amount of memory needed. This results from the Kalman gains being matrices in dimension of the large-scale model, which are saved in the pre-computation. The other two methods do not have this disadvantage, since they operate exclusively on the

Table 2: Errors and CPU time for partDE network

Filter	mean _j $\frac{\ \mathbb{E}[\mathbf{x}_j - \hat{\mathbf{x}}_{j j}]\ _{L^2}}{\ \mathbb{E}[\mathbf{x}_j]\ _{L^2}}$	Offline [s]	Online [s]
RKF *	$2.2 \cdot 10^{-1}$	$2.5 \cdot 10^2$	$3.4 \cdot 10^1$
CSKF	$6.3 \cdot 10^{-2}$	$2.6 \cdot 10^4$	$2.5 \cdot 10^4$
REnKF * \diamond	$3.5 \cdot 10^{-1}$	$7.1 \cdot 10^1$	$1.4 \cdot 10^2$

* additional post-processing for prolongation is required

\diamond offline time refers to sampling of random variables

low dimension and the Kalman gains are hence of small size. The time for prolongation of state and error covariance is not included in the listed CPU time as it can be performed for selected time points in a post-processing step.

RKF and REnKF yield comparable results in terms of accuracy and efficiency. For ensemble filtering, one might expect that the computational costs could be further reduced by tuning (sample selection), but this is left to future research. However, RKF – with its separation in offline and online phases – is clearly preferable, when input or measurement data changes, as is the case for control of gas pipeline networks.

4 Conclusion

In this work we presented an efficient approach for state estimation in gas networks. Using a low-rank model derived by linearization and projection-based model order reduction, the Kalman Filter, which is unfeasible for the original large-scale system, becomes applicable. We compared our approach to the Compressed State Kalman Filter whose low-rank also comes from projection-based reduction. Although the underlying approximation idea is very similar, our approach yielding estimates of comparable accuracy proved itself to be much more efficient and less memory-intensive in a numerical study on a German pipeline network. Moreover, the structure-preserving model order reduction can be straightforward combined with other low-rank Kalman filter variants, such as the Ensemble Kalman Filter.

Acknowledgement

This work is supported by the German Federal Ministry for Economic Affairs and Energy, project *MathEnergy – Mathematical Key Technologies for Evolving Energy Grids*.

References

- [1] Hesam Ahmadian Behrooz and R. Bozorgmehry Boozarjomehry. Modeling and state estimation for gas transmission networks. *Journal of Natural Gas Science and Engineering*, 22:551–570, 2015.
- [2] Peter Benner, Albert Cohen, Mario Ohlberger, and Karen Willcox, editors. *Model Reduction and Approximation: Theory and Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, 2017.
- [3] Jens Brouwer, Ingenuin Gasser, and Michael Herty. Gas pipeline models revisited: Model hierarchies, non-isothermal models, and simulations of networks. *Multi-scale Modeling & Simulation*, 9(2):601–623, 2011.
- [4] T. Clees, A. Baldin, P. Benner, S. Grundel, C. Himpe, B. Klaassen, F. Küsters, N. Marheineke, L. Nikitina, I. Nikitin, J. Pade, N. Stahl, C. Strohm, C. Tischendorf, and A. Wirsén. MathEnergy – Mathematical key technologies for evolving energy systems. In *Mathematical Modeling, Simulation and Optimization for Power Engineering and Management*, pages 233–262. Springer, 2021.
- [5] Markus Dihlmann and Bernard Haasdonk. A reduced basis Kalman filter for parametrized partial differential equations. *ESAIM - Control, Optimisation and Calculus of Variations*, 22(3):625–669, 2016.
- [6] Herbert Egger, Thomas Kugler, Björn Liljegren-Sailer, Nicole Marheineke, and Volker Mehrmann. On structure-preserving model reduction for damped wave propagation in transport networks. *SIAM Journal on Scientific Computing*, 40(1):A331–A365, 2018.
- [7] Hojat Ghorbanidehno, Amalia Kokkinaki, Judith Li, Eric Darve, and Peter K. Kitanidis. Real-time data assimilation for large-scale systems: The spectral Kalman filter. *Advances in Water Resources*, 86:260–272, 2015.
- [8] Martin Gugat and Rüdiger Schultz. Boundary feedback stabilization of the isothermal Euler equations with uncertain boundary data. *SIAM Journal on Control and Optimization*, 56(2):1491–1507, 2018.
- [9] P. L. Houtekamer and Herschel L. Mitchell. Ensemble Kalman filtering. *Quarterly Journal of the Royal Meteorological Society*, 131:3269–3289, 2005.
- [10] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.
- [11] Matthias Katzfuss, Jonathan R. Stroud, and Christopher K. Wikle. Understanding the Ensemble Kalman Filter. *American Statistician*, 70(4):350–357, 2016.
- [12] Peter K. Kitanidis. Compressed state Kalman filter for large systems. *Advances in Water Resources*, 76:120–126, 2015.
- [13] Judith Yue Li, Amalia Kokkinaki, Hojat Ghorbanidehno, Eric F. Darve, and Peter K. Kitanidis. The compressed state Kalman filter for nonlinear state estimation: Application to large-scale reservoir monitoring. *Water Resources Research*, 51(12):9942–9963, 2015.
- [14] Björn Liljegren-Sailer and Nicole Marheineke. Port-Hamiltonian approximation of a nonlinear flow problem on networks. *SIAM Journal on Scientific Computing*, 2020. accepted.
- [15] Björn Liljegren-Sailer and Nicole Marheineke. On snapshot-based model reduction under compatibility conditions for a nonlinear flow problem on networks. 2021. Preprint, arXiv:2110.04777.
- [16] Lars Nerger, Wolfgang Hiller, and Jens Schröter. A comparison of error subspace Kalman filters. *Tellus A: Dynamic Meteorology and Oceanography*, 57(5):715–735, 2005.
- [17] Stefano Pagani, Andrea Manzoni, and Alfio Quarteroni. Efficient state/parameter estimation in nonlinear unsteady PDEs by a Reduced Basis Ensemble Kalman filter. *SIAM-ASA Journal on Uncertainty Quantification*, 5(1):890–921, 2017.
- [18] H. Prashanth Reddy, Shankar Narasimhan, and S. Murty Bhallamudi. Simulation and state estimation of transient flow in gas pipeline networks using a transfer function model. *Industrial & Engineering Chemistry Research*, 45(11):3853–3863, 2006.
- [19] Nadine Stahl and Nicole Marheineke. A low-rank Extended Kalman filter for gas pipeline networks. In *Progress in Industrial Mathematics at ECMI 2021*. Springer, 2021.
- [20] TU Berlin (TUB-ER). Gas network – partDE data set, 2019. <https://www.er.tu-berlin.de/menue/home/parameter/en/>.

Robotics simulation – A comparison of two state-of-the-art solutions

Maximilian Zwingel^{1*}, Christopher May¹, Matthias Kalenberg¹, Jörg Franke¹

¹Institute for Factory Automation and Production Systems (FAPS), Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany; *maximilian.zwingel@faps.fau.de

Abstract. Simulation plays a significant role in development and deployment of robotics systems. Depending on the planned tasks, different functionalities shift into focus. For mobile robotics systems using the Robot Operating System (ROS), Gazebo has been the goto solution for simulating behavior, sensors, and interaction with environments for many years. With the availability of numerous novel solutions, a comparison of the different strengths from a use-case perspective becomes necessary.

In our paper, we compare Gazebo and NVIDIA's Isaac Sim in the context of ROS-based autonomous mobile robots. Special attention is given to the integration of the simulators in the ROS-environment and their ability in regards to sensor simulation. This paper aims to provide a comparison between different features in Gazebo and NVIDIA Isaac Sim for simulating mobile robots and sensor systems.

Introduction

Robotic systems play a significant role in addressing the challenges of future production. Tasks like handling, separation, identification and transportation need to be automated to enable sustainable productivity in globalized markets. Simulation plays a significant role in development and deployment of robotics systems to reduce the risks of misfitting and delays in ramp up. In light of these developments, different companies from tech business sectors have recently begun to focus on robotics as well as interlocked technologies to challenge existing solutions in the field and to open up new markets. Especially in the area of simulation of mobile robotics new frameworks with a focus on photo-realistic presentation emerge.

NVIDIA is well known for its graphics processors, but is also pushing towards AI as well as autonomous vehicles and robots in recent years. With Isaac Software Development Kit (SDK) and Isaac Sim, they develop a new solution for simulation addressing their Jetson platform. Since 2018, NVIDIA has further developed

NVIDIA Isaac Sim (NIS) to support robots based on the Robot Operating System (ROS) and to use NVIDIA Omniverse as its platform.

The current go-to solution for simulating ROS-based robots is Gazebo, an open-source solution being under development since 2001. [1] Gazebo – developed by the Open Source Robotics Foundation (OSRF) - has reached its last major release in 2020 and is currently transferred to a new software foundation called “ignition”. Gazebo's main focus is set on realistic physics simulation as well as digitally replicating sensor systems. Gazebo's role as a go-to solution in science and software development is reflected both in the number of publications found using the search term "Gazebo ROS" in the IEEE database and in Google search engine trends. Alternatives, such as RoboDK, SimSpark or Coppelia Sim do not reach the popularity of Gazebo. Nvidia Isaac Sim has yet to be focussed in a scientific publication, however Google search trends show a significant increase in interest since January, overcoming most of the other simulation frameworks.

With novel solutions for robotics simulation slowly reaching maturity, a comparison between established software and newcomers is necessary to aid future developers. In our paper we compare Gazebo and NIS in the context of mobile, ROS-based robotics to achieve a use-case comparability. In chapter 1, the two simulation frameworks are detailed and a general comparison of features is done. In chapter 2, the applicability and features of both systems to use cases in mobile robotics is examined. In chapter 3, the real-world accuracy of both simulation environments is tested with different tasks representative for mobile robotics systems. A discussion of strengths and weaknesses is presented in chapter 4 before chapter 5 finishes with an outlook upon further improvements.

Disclaimer: All statements refer to Gazebo 9.16 and NIS version 2021.2, which are the latest version at the time of writing, and might not apply in later versions.

1 Comparison of features

Isaac Sim is a simulation suite with focus on photo-realistic presentation of simulated environments in conjunction with NVIDIAs hardware systems. Comparable with other visually focused frameworks like Unity [2], its strength lays in graphic representation rather than physics and behavior simulation. [3] Gazebo is a three-dimensional multi body simulation with a strong focus on the physical properties of materials and the modeling of physical effects.

1.1 Hardware requirements

The hardware requirements for Gazebo are low for today's standards. The OSRF specifies an Intel i5 (no specified generation) or equivalent CPU, a dedicated GPU and at least 500 MB free disk space. [4]

NIS has significantly higher requirements: As a minimum, an Intel i7 7th gen. or equivalent CPU, 32 GB of RAM, SSD Storage and a GeForce RTX 2070 GPU are required. [5] Without an RTXTM GPU by NVIDIA, NIS respectively the RTXTM renderer will not run. NIS therefore poses a much higher entrance barrier on the hardware side than Gazebo.

1.2 Degree of physics simulation

Gazebo offers four different physics engines: Open Dynamics Engine (ODE), Bullet, Dynamic Animation and Robotics Toolkit (DART) and Simbody. [6] NIS uses Nvidias PhysX engine, which originally was developed for computer games.

A 2015 paper found PhysX to be fast, but sometimes lacking in real world accuracy compared to Bullet and ODE. [7] Since then, NVIDIA has continuously worked on their physics engine, including an announcement to incorporate the needs of robotics simulation in 2018. [8] To date there is no specific research in performance and truth to reality of the different engines.

One major task of simulation is description of multi-body systems. In the following we will compare the joint types available in Gazebo and the joint types available in NIS. Both programmes provide essential standard types of joints by their file formats. Those are rotational joints in one or two degrees of freedom (dof), geared rotational joints, linear and fixed joints. Both simulators allow adding limits to those, although to a different extent. While Gazebo requires adding a different joint type to implement limits, NIS enables the users to add limits in the

properties of the joint, which is more intuitive to the user.

In addition to those, each software has more special types. Gazebo additionally offers the “revolute2” joint – two serially coupled rotational joints – and a “screw” joint, which couples linear and rotational movement in one axis. NIS additionally offers three joints: Firstly, a 6-dof joint, which allows movement in all three linear and rotational axes, which can be limited. Secondly, a “distance” joint, which is only limited by the distance between the two connected bodies. Thirdly, a “rack-and-pinion” joint, which translates between rotational and linear movement, including a gear-ratio. Table 1 provides an overview over the mentioned joint types in each software. [10,9]

Joint type	Gazebo	NIS
Rotational (1-dof)	Yes	Yes
Rotational (2-dof)	Yes	Yes
Rotational (geared)	Yes	Yes
Rotational (serial)	Yes	No
Linear (1-dof)	Yes	Yes
Fixed	Yes	Yes
Screw	Yes	No
6-dof	No	Yes
Distance	No	Yes
Rack-and-Pinion	No	Yes

Table 1: Comparison of available joint-types in Gazebo and NIS

In NIS, non-geared joints can also be transferred into “articulations”, which NVIDIA claims to behave more accurately than traditional simulated joints. [10]

Gazebo's physics library supports aerodynamics and hydrodynamics which allows third party plugins to extensively simulate unmanned aerial vehicles (UAVs) and unmanned underwater vehicles (UUVs). [12,11] NIS does not support this and is therefore currently no alternative for those domains of robotics.

1.3 Graphics simulation

A field where NIS outdoes Gazebo literally on first look is the graphics simulation. Graphics is where Gazebo/OGRE start showing their age, while NIS shows its roots at NVIDIA and their profound experience in computer graphics.

Gazebo utilizes “OGRE” – the “Object-Oriented Graphics Rendering Engine”. [6,13] NIS uses the

NVIDIA Omniverse RTX™ Renderer which allows for high-fidelity rendering using raytracing. NVIDIA's renderer uses their "Material Definition Language" (MDL) which is layer based and supposed to interact realistically with virtual light sources. [14] NIS comes with varied materials provided. Figure 1 illustrates the differences in graphics simulation between both programs.



comparability, we define a use case relevant for research in mobile robotics to test both simulation toolkits. Subsequently, we are comparing the capabilities of Gazebo and NIS in regards to use-case specific features, mainly sensor simulation addressing mobile robots. We also address the potential change and interoperability from Gazebo to NIS, based on the broad application of Gazebo today.

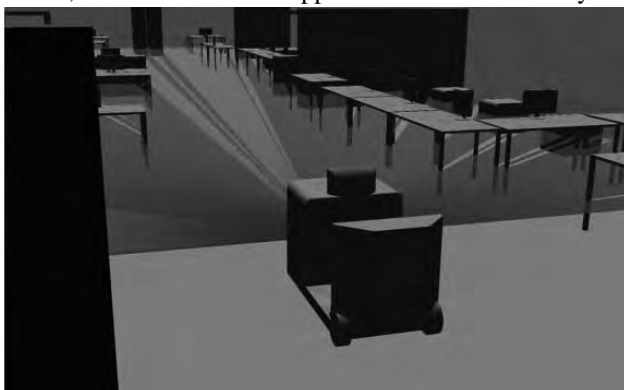


Figure 1: Screenshot from NIS (left) and Gazebo (right) showing a representative application scenario

Albeit in different ways, both Gazebo and NIS allow their users to add and modify virtual cameras. The parameters are comparable and allow the editing of intrinsic camera parameters like focal length or the projection type. NIS additionally allows editing the aperture and f-stop to simulate the depth of field of real cameras. [16,15]

1.4 Connection to other Frameworks

While Gazebo and NIS can be used as a standalone simulator, for mobile robotics they are often used in conjunction with ROS. Both simulators can initialize a ROS-node which acts as the translator between ROS and the simulator.

In Gazebo, this can be achieved by installing the "gazebo_ros_pkgs" metapackage in ROS, which interfaces ROS with Gazebo. Other open-source robotics frameworks cannot be integrated natively with Gazebo. [4]

NIS is comparable in this regard, but additionally can be integrated with the NVIDIA Isaac Toolkit, aimed towards the development of AI-powered robots on NVIDIA's platforms. [17] For NIS there is no need to install a ROS-package, as the ROS-bridge is included as an extension.

2 Use-case specific comparison

The usefulness of different simulation environments is highly dependent on the corresponding use case. For

2.1 Mobile robotics as use-case

Mobile robot systems are often used in intralogistics applications for warehousing and commissioning. Perception is especially important for autonomous mobile robots, since they need to be able to react safely in dynamic environments. To guarantee a safe and predictive behaviour, simulation of systems and intralogistics environments is necessary, especially in regards to capabilities of used sensor systems, like radio detection and ranging (RADAR), cameras, light detection and ranging (LiDAR) and other sensor principles. Additionally, testing mobile robots in real environments is challenging, because they might pose a threat to bystanders and hinder operations in work environments. Thus, we chose a mobile robot to compare Gazebo with NIS. As real-world-reference, we use an autonomous tow-truck, which is described in [18]. The robot and its testing environment have been recreated in both simulators.

2.2 Interfaces between ROS, Gazebo and Isaac Sim

The first step in simulating mobile robotics is the integration of robot models and worlds into the corresponding environment. Gazebo uses .sdf-files to describe robots and worlds, NIS uses the .usd-format. Right now, both formats are not directly interchangeable and to different degrees supported by ROS. To facilitate the use of models in other formats, NIS offers a variety of "connectors",

which allow easy switching back and forth with software by other companies like Autodesk.

For .urdf-files, which are used in ROS to describe robots and are also compatible with Gazebo's .sdf-Format, NIS provides a simple import function. However, the importer does not deliver reliable and deterministic results. Especially the type of joints and their orientation are not converted reliably for different models. For more complex .sdf-files, currently no way to import them into NIS exists. This results in the need to completely rebuild the simulated environments and also the robots in most cases. For some common commercially available robots, NIS provides pre-built models. If worlds or objects are available in a format usable by the free 3D-modeling software blender, they can be exported as .usd-files from there. To sum up, migrating extensive worlds and models between Gazebo and NIS is a laborious task. It is advantageous to decide early in a project, which software to use.

Otherwise, using ROS with NIS is pretty straightforward. To use the ROS-bridge in NIS, users have to add so called "ROS-Components" to their model, which is done intuitively via the GUI. Available are: ROS-clock, Camera, Joint State, Lidar, Pose Tree, Teleport, Surface Gripper and Differential Base. The components can be configured in various ways and their corresponding topics can be modified to work with the topics of an existing ROS project. Therefore, if all necessary components exist in NIS, the integration of ROS on the software side is just a minor inconvenience.

However, not all necessary components for the mentioned use-case are available in NIS. The considered tow-truck doesn't use a differential drive model which is common for robots, but a model which is comparable to Ackermann steering for vehicles. For this reason, some modifications are necessary, which translate the standard message type of ROS to a rotation angle of the steering and velocity of the drive wheel. Those topics can then be handled by joint state components in NIS. Additionally, not all necessary sensors are available, which is described in the following subsection.

2.3 Sensor simulation

Sensor systems are a core element of the simulation of autonomous robots. For this reason, special attention is given to the capabilities of both simulators in this area of operation. The real use-case robot utilizes a LiDAR-scanner, two RGB-D cameras with integrated Inertial

Measurement Units (IMUs), a RADAR sensor, ultrasonic sensors and different local and global multilateration systems like GPS.

Natively, Gazebo offers mainly the following virtual sensors: Different cameras (including depth-cameras), beam and wave-based range sensors, which natively allow the simulation of 1D to 3D laser sensors, inertial measurement units (IMUs) and a bumper or contact sensor. Further sensors can easily be added via third-person plugins, exemplary a GPS receiver. [4]

NIS offers the same sensors that Gazebo offers natively, additionally the range sensor can be implemented as an ultrasonic sensor. However, the ROS-Bridge of NIS only allows the use of cameras and LiDARS – support for other sensors currently has to be developed by the users. Ultrasonic and contact sensors as well as IMUs can therefore not be used with ROS out of the box. Additionally, the documentation for implementation and modification those sensors is extremely scarce. NIS does allow user-made extensions and provides a tutorial on how to integrate the contact sensor with ROS. Unfortunately, at the time of writing only very few of those extensions are publicly accessible. Thus, not all sensors necessary for mobile robotics systems are available. The following Table 2 provides an overview on the availability of different sensor types in both simulators out of the box.

Sensor type	Gazebo	NIS
RGB camera	Yes, natively	Yes
Depth camera	Yes, natively	Yes
LiDAR	Yes, natively	Yes
IMU	Yes, natively	Limited, see text
Contact	Yes, natively	Limited, see text
Ultrasonic	Yes, plugin	Limited, see text
GPS	Yes, plugin	No

Table 2: Availability of relevant sensors in Gazebo and NVIDIA Isaac Sim

When using simulations, one always has to keep in mind the deviations between the simulation and reality, known as "Sim2Real Gap". NVIDIA claims to close this gap with NIS. [19] Upon inspection, the statement not yet holds.

Comparable to Gazebo [1] NIS calculates the measurements of range sensors using the collision geometry of objects. That approach neglects the considerable effects of material properties like transparency and

reflections on optical time-of-flight sensors. NVIDIA is planning on supporting RTX™ and with it raytracing for sensors in NIS in the future, starting with the version released in spring 2022. [20]

An even more relevant influence on real sensor measurements is noise, which is existent in all sensor systems. Some sensors and environmental influences have been studied extensively, for example LiDAR systems, especially in the context of safety systems. [22,21] To simulate real working conditions, it is necessary to also model noise in simulated sensors. Gazebo supports adding Gaussian noise to sensor readings, which is usually sufficient to approximate sensor noise. [4,22,21] NIS does not support adding noise to any sensors, which is a major drawback for realistic simulation tasks. Modifications by adding sensor noise in ROS are possible but cannot substitute an integration in the simulation environment. Support for sensor noise is also announced for the next release of NIS. [23]

2.4 Simulation of moving objects and persons

In recent years, increasingly many robots are used in mixed areas with human workers. To simulate operation in those spaces, especially for mobile robotics, a realistic simulation of moving persons is necessary. Gazebo handles this task using “actors”. Actors are animated models, on which a skeleton and a trajectory animation can be applied. Their trajectory can also be influenced by the environment. [4] NIS does not have those capabilities. While NIS does provide animated and realistic looking models, eg. of a worker, the animation in NIS only influences the visual representation of a model. If a collision model is added, it stays in a stationary “T-Pose”. Also, there is no way to let those movements be influenced by the environment. At the time of writing, there are no noticeable intentions by NVIDIA to change this.

3 Comparison of simulation and a real system / Sim to Real

Based on the defined use case, a comparison of both simulations and a real system is conducted to evaluate the degree of concordance. In mobile robotics – like the described autonomous tow-truck - occupancy grid maps (OGMs) are commonly used for mapping and localization purposes and heavily depend on the sensors and their perception of the working environment. With many algorithms allowing for extensive parametrization,

simulation plays a major role in optimization and reducing field testing.

The evaluation is conducted by comparing different mapping algorithms in three representations of the same environment. We apply the particle-filter-based “hector_mapping”, which only uses LiDAR-data, and the two graph-based approaches “slam_toolbox” and “RTAB-map”. [24,26,25] The former utilizes LiDAR- and odometry-data, the latter has been configured to use RGB-D and odometry-data.

Each simulation environment is parameterized and hand-built to replicate the real environment as realistically as possible. The driven trajectory of the mobile robot is comparable for all mapping processes and consists of a closed loop through the environment. Mapping in the real environment has been conducted with an Omron OS32C-SP1-4M laser scanner and an Intel D435 RGB-D camera. Both sensors are recreated in the simulators as precisely as possible. Noise is added to the laser scanner readings natively in Gazebo and using a self-written script with comparable functionality in NIS. The odometry-data of the real system is generated using the sensor data fusion approach described in [18]. For the simulated systems, no modification of odometry data is possible. This is suboptimal, since the quality of odometry data significantly influences the SLAM-problem. [27]

The following maps in Figure 2 are generated using the LiDAR-based hector_mapping in the three environments. It can be seen as a reference for Extended Kalman Filter – SLAM algorithms (EKF-SLAM).

We can see, that hector_mapping delivers good results, both in reality and simulation. The influence of noise for detectable surfaces can be sufficiently approximated using gaussian noise. In addition, however, it also becomes apparent that critical interfering factors for laser systems, such as reflections or materials with different optical properties, cannot be simulated correctly.

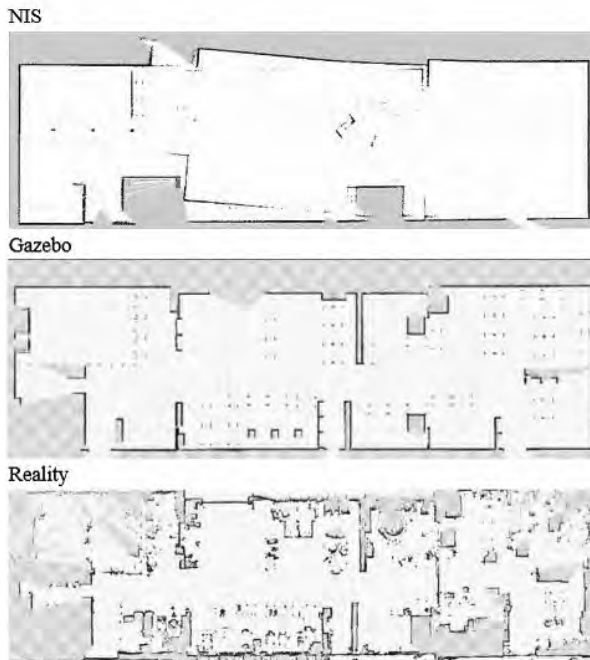


Figure 2: Occupancy Maps generated using hector_mapping in NIS, Gazebo and using a real system; The two simulated environments were built to mirror the real environments in dimension and interior.

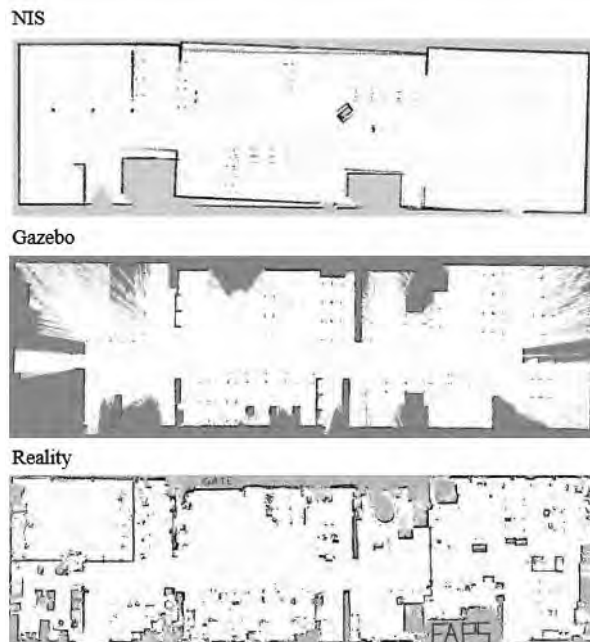


Figure 3: Occupancy Maps generated using SLAM Toolbox with added noise in NIS ($\sigma = 0.5\%$)

The maps shown in Figure 3 are generated using the “slam_toolbox” ROS-package to show the capabilities of the different environments for development and testing of graph-based SLAM. Again, both simulators deliver usable results. However, the Sim2Real gap is wider here,

likely because of the use of perfect odometry data in simulation.

The previously described approaches did not deliver significantly distinguishable results between both simulators. This meets our expectations, since like described in section 2.3, NIS does not yet provide improvements in simulation of LiDAR-scanners. Next, we compare the results from the “RTAB-Map” package, which uses data from a RGB-D camera and odometry data.

From Figure 4 it can be seen that Gazebo is not suitable for photorealistic rendering of depth images. Due to the low level of detail, feature-dependent algorithms such as “RTAB-Map” can only be used and tested to a limited extent. NIS enables the robust use of the same algorithms, even though a difference in results between the real system with changing lighting situations and perturbations and the simulation can still be seen.

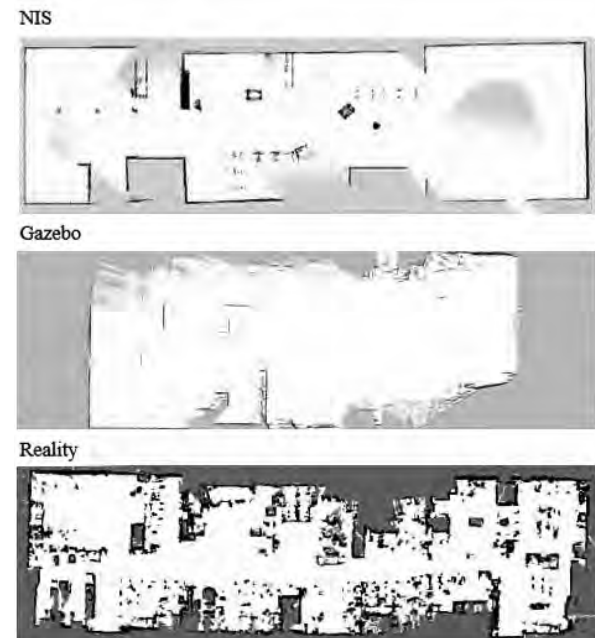


Figure 4: Occupancy Map generated using RTAB-Map with RGB-D and Odometry data for the different environments

4 Discussion of performance

In this chapter we summarize our findings into recommendations for a ROS-simulator for different demands. The results are presented as tables. Table 3 compares NIS and Gazebo in different Criteria which are based on performance described in the previous sections. Both simulators are rated either with ++ (very positive), + (positive), 0 (neutral), - (negative), -- (very negative) as

shown in [28–30]).

Criteria	Gazebo	NVIDIA Isaac Sim
Hardware requirements	+	--
Software requirements	+	0
Physics simulation	+	+
Graphics simulation	-	++
Sensor simulation	+	-
Universality	++	0

Table 3: Rating of NVIDIA Isaac Sim and Gazebo in different Categories

Table 4 lists different possible demands in a project and suggests the better suited simulator. Both tables are meant to provide guidance when choosing a software to simulate mobile ROS-based robots during development. The criteria and ratings are explained in detail in the previous sections.

Demand	Better suitable simulator
Physics simulation	Gazebo
Evaluation of vision-based algorithms	NVIDIA Isaac Sim
Evaluation of optical-/laser-based algorithms	Gazebo (until NIS utilizes raytracing for laser simulation)
Project duration	Short- & medium-term: Gazebo Long-term: NVIDIA Isaac Sim
Simulation of UAVs or UUVs	Gazebo
Availability of moving objects and persons	Gazebo

Table 4: Simulator recommendations for different demands

5 Outlook on further developments

As described above, development cycles are becoming shorter and shorter and new players in the field of robotics simulation are entering the market. Based on our expertise, we venture a look into the crystal ball and attempt a forecast for developments in the near future.

5.1 Expected evolution of Isaac Sim

As shown before, NIS is built on a more modern platform than Gazebo. However, NIS is not yet as mature as Gazebo and lacks features in some areas crucial for mobile robotics. In this subsection we summarize what to expect from NIS for the future based on press releases and forum discussions. In general, NVIDIA’s staff is very helpful in the forum and happy to add requested features or reported bugs to their list. It is therefore expected that frequently demanded features will make their way into NIS sooner or later. Unfortunately, there is no official roadmap available, which complicates planning in projects. The OSRF and NVIDIA have announced interoperability between NIS and the new Ignition Gazebo. Prospectively, both simulators are supposed to feature converters for .usd respective .sdf files and also allow direct interaction. Ignitions roadmap aims to release these features in September 2022. [31]

5.2 Conclusion

From our research, we conclude that NIS is not yet ready to supersede Gazebo as the go-to simulation tool for mobile robots using ROS. At the time of writing, NIS imposes many hurdles to overcome when using the software. Many restraints of NIS can be overcome by developing own workarounds. However, those are seldom optimal. Gazebo is also still on par with NIS regarding physics simulation tasks. While NVIDIA has some lighthouse projects together with major corporations like BMW or recently Amazon, it’s scientific user base seems to be small. This is a chicken-and-egg-problem which requires some pioneering work to solve itself. Nevertheless, in the time of our usage, we noticed significant advances in NIS. We therefore see a high potential for NIS. Coming features could lift NIS on a new level, eg. factoring in reflection properties of surfaces for ToF-sensors. If one has access to a system capable of running NIS, trying it out might be worthwhile depending on the type of project – see Table 4. If one can’t access such a system, Gazebo still is a very powerful tool and might have a comparable successor for ROS 2 with Ignition Gazebo. Further research should be conducted to evaluate the current state of the PhysX-engine and comparing performance and truth to reality of the mentioned engines. New releases of NIS might generate additional needs for research, especially on the simulation of reflective properties for ToF-sensors and the interoperation between NIS and Ignition Gazebo.

References

- [1] Koenig, N., Howard, A., 2004. Design and use paradigms for gazebo, an open-source multi-robot simulator, in: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566). 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Sendai, Japan. 28 Sept.-2 Oct., 2004. IEEE, pp. 2149–2154.
- [2] Hussein, A., Garcia, F., Olaverri-Monreal, C., 2018. ROS and Unity Based Framework for Intelligent Vehicles Control and Simulation, in: 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES). 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Madrid. 12.09.2018 - 14.09.2018. IEEE, pp. 1–6.
- [3] Santos Pessoa de Melo, M., Da Gomes Silva Neto, J., Da Jorge Lima Silva, P., Natario Teixeira, J.M.X., Teichrieb, V., 2019. Analysis and Comparison of Robotics 3D Simulators, in: 2019 21st Symposium on Virtual and Augmented Reality (SVR). 2019 21st Symposium on Virtual and Augmented Reality (SVR), Rio de Janeiro, Brazil. 28.10.2019 - 31.10.2019. IEEE, pp. 242–251.
- [4] Osrf, 2022. Gazebo : Tutorials. <http://gazebo.org/tutorials>. Accessed 22 March 2022.
- [5] Isaac Sim Requirements — Omniverse Robotics documentation, 2022. https://docs.omniverse.nvidia.com/app_isaac-sim/app_isaacsim/requirements.html. Accessed 15 February 2022.
- [6] Osrf, 2022. Gazebo. <http://gazebo.org/>. Accessed 18 February 2022.
- [7] Erez, T., Tassa, Y., Todorov, E. Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX, in: 2015 IEEE International Conference, pp. 4397–4404.
- [8] Lebedev, R., 2018. NVIDIA Extends PhysX for High-Fidelity Simulations, Goes Open Source. NVIDIA. <https://blogs.nvidia.com/blog/2018/12/03/physx-high-fidelity-open-source/>. Accessed 15 February 2022.
- [9] Osrf, 2022. SDFFormat Specification. <http://sdformat.org/spec?ver=1.9&elem=joint>. Accessed 28 February 2022.
- [10] Physics Core — Omniverse Create documentation, 2022. https://docs.omniverse.nvidia.com/app_create/prod_extensions/ext_physics.html. Accessed 28 February 2022.
- [11] Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., Stryk, O. von, 2012. Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo, in: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Noda, I., Ando, N., Bruggali, D., Kuffner, J.J. (Eds.), Simulation, Modeling, and Programming for Autonomous Robots, vol. 7628. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 400–411.
- [12] Manhaes, M.M.M., Scherer, S.A., Voss, M., Douat, L.R., Rauschenbach, T., 2016. UUV Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation, in: OCEANS 2016 MTS/IEEE Monterey. OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA. 19.09.2016 - 23.09.2016. IEEE, pp. 1–8.
- [13] OGRE - Open Source 3D Graphics Engine | Home of a marvelous rendering engine, 2022. <https://www.ogre3d.org/>. Accessed 18 February 2022.
- [14] NVIDIA, 2022. Material Definition Language from NVIDIA. <https://www.nvidia.com/en-us/design-visualization/technologies/material-definition-language/>. Accessed 18 February 2022.
- [15] Osrf, 2022. SDFFormat Specification. <http://sdformat.org/spec?ver=1.9&elem=sensor>. Accessed 21 February 2022.
- [16] Cameras — Omniverse Create documentation, 2022. https://docs.omniverse.nvidia.com/app_create/prod_materials-and-rendering/cameras.html#. Accessed 21 February 2022.
- [17] NVIDIA Developer, 2019. NVIDIA Isaac SDK. <https://developer.nvidia.com/isaac-sdk>. Accessed 7 March 2022.
- [18] Herbert, M., Zwingel, M., Czapka, C., Franke, J., 2022. A Multi-source Localization System for Driverless Material Transport in Mixed Indoor and Outdoor Areas, in: Production at the Leading Edge of Technology. Springer International Publishing, Cham, pp. 421–429.
- [19] NVIDIA Technical Blog, 2021. Training in NVIDIA Isaac Sim Closes the Sim2Real Gap | NVIDIA Technical Blog. <https://developer.nvidia.com/blog/training-in-nvidia-isaac-closes-the-sim2real-gap/>. Accessed 21 February 2022.
- [20] NVIDIA Developer Forums, 2021. Influence of materials on LiDAR and (depth-)cameras - Isaac / Omniverse Isaac Sim - NVIDIA Developer Forums. <https://forums.developer.nvidia.com/t/influence-of-materials-on-lidar-and-depth-cameras/197372>. Accessed 21 February 2022.
- [21] Ye, C., Borenstein, J., 2002. Characterization of a 2D laser scanner for mobile robot obstacle negotiation, in: Proceedings of the 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292). 2002 IEEE International Conference on Robotics and Automation, Washington, DC, USA. 11-15 May 2002. IEEE, pp. 2512–2518.
- [22] Reina, A., Gonzales, J., 1997. Characterization of a radial laser scanner for mobile robot navigation, in: Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97. 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97, Grenoble, France. 11.09.1997 - 11.09.1997. IEEE, pp. 579–585.
- [23] NVIDIA Developer Forums, 2021. Adding noise to sensor readings - Isaac / Omniverse Isaac Sim - NVIDIA Developer Forums. <https://forums.developer.nvidia.com/t/adding-noise-to-sensor-readings/186141>. Accessed 21 February 2022.
- [24] Kohlbrecher, S., Stryk, O. von, Meyer, J., Klingauf, U., 2011. A flexible and scalable SLAM system with full 3D motion estimation, in: 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics. 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Kyoto, Japan. 01.11.2011 - 05.11.2011. IEEE, pp. 155–160.
- [25] Macenski, S., Jambrecic, I., 2021. SLAM Toolbox: SLAM for the dynamic world. Journal of Open Source Software 6 (61), 2783.
- [26] Labbé, M., Michaud, F., 2019. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. J Field Robotics 36 (2), 416–446.
- [27] Thrun, S., Burgard, W., Fox, D., 2005. Probabilistic robotics. MIT Press, Cambridge, Mass., London, 647 pp.
- [28] Aaron Staranowicz, Gian Luca Mariottini, 2011. A Survey and Comparison of Commercial and Open-Source Robotic Simulator Software. ACM, New York, NY, 401 pp.
- [29] Michal, D.S., Etzkorn, L. A Comparison of Player / Stage / Gazebo and Microsoft Robotics Developer Studio. ACM Conferences. Association for Computing Machinery, New York, NY, 399 pp. <http://dl.acm.org/citation.cfm?id=2016039>.
- [30] Pitonakova, L., Giuliani, M., Pipe, A., Winfield, A., 2018. Feature and Performance Comparison of the V-REP, Gazebo and ARGoS Robot Simulators, in: Towards Autonomous Robotic Systems. Lecture Notes in Artificial Intelligence. Towards Autonomous Robotic Systems (TAROS) Conference, Bristol. July 25-27, 2018. Springer International Publishing, Cham, pp. 357–368.
- [31] Ignition - Docs: Roadmap, 2021. <https://ignitionrobotics.org/docs/all/roadmap>. Accessed 21 February 2022.

Parameter tables for PID controllers for time delayed systems optimized with a learning method

Roland Büchi^{1*}

¹Zurich University of Applied Sciences, School of Engineering, Technikumstrasse 9, CH 8401 Winterthur, Switzerland;

*roland.buechi@zhaw.ch

Abstract. This publication provides useful parameter sets in tabular form for PID controllers for various rise and dead times of step responses of asymptotically stable control systems, which minimize the common quality criteria in the time domain, integral of absolute error IAE, integral of time-multiplied absolute value of error ITAE and integral of squared error ISE. Since the determination of the parameter sets is very computationally intensive, an approach from the field of artificial intelligence was chosen.

The application of the parameter sets found is verified with examples. The parameter sets also take into account the controller output limitations that are relevant in practice and can basically be used for all PID controllers of controlled systems with a time delay.

1 Introduction and related work

PID controllers are still by far the most frequently used controller structures for single-in, single-out (SISO) systems. The control of controlled systems with dead time is challenging. The parameters found with heuristic methods lead to asymptotically stable systems. The most famous of them are those from Ziegler Nichols, Latzel, or Chien Hrones and Reswick. However, all parameters found with these methods still have to be readjusted in the practical system so that sensible transient behavior results.

The time-delayed systems are very common in practice. They require special demands, because their control is challenging. In practice, however, they are very common, especially in process engineering or in thermal systems, since the sensor often can not be placed directly next to the actuator.

There are different approaches known for finding PID controller parameters from step responses of time-delayed systems. All of them result in stable control systems. Especially, as the dead time increases, it becomes difficult to find suitable PID parameter sets. There are some heuristic methods for this, which can be used in the time and frequency domain. However, these parameters

must be further optimized afterwards. The first approach was the parameter set from Ziegler Nichols [1]. There are also several others existing, for example Chien, Hrones and Reswick [2]. For further optimization, there are several methods used, also some from the field of artificial intelligence, for example particle swarm optimization, PSO [3].

In this paper, the method hill climbing [4] is used. It is another stochastic method for optimizing controllers, but it is related to PSO. For optimizing controller parameters, there also other approaches known [5]–[9], [13]–[15].

In order to be able to handle time-delayed systems in terms of simulation at all, the turning point tangent method is often used. A PTn system is identified with n PT1 (1st order) elements connected in series, which have identical time constants. They are dealt with in the literature [10], [11], [12]. The PID parameter tables, which are described and used in the next chapters, however, refer to these PTn systems with identical time constants. Such systems are very common and can be found in all engineering disciplines. The series connection of such PTn systems according to formula (1) leads to step responses which are delayed. In particular, the dead times can be approximated with linear models in this way. Here, K_s is the static gain, n is the system order and T_1 is the time constant of the n identical PT1 elements.

$$\frac{K_s}{(s \cdot T_1 + 1)^n} \quad (1)$$

2 Identification of PTn

The turning point tangent should be used here as a reference for identification. In many cases, one can simply measure the delay time T_u and the rise time T_g according to figure 1 by placing a tangent at the point of inflection. From this one can identify the number n of PT1 elements connected in series and their identical time constants T_1 .

The measurement of the step response of a controlled system can then be dealt with using table 1, which is well known from literature [10].

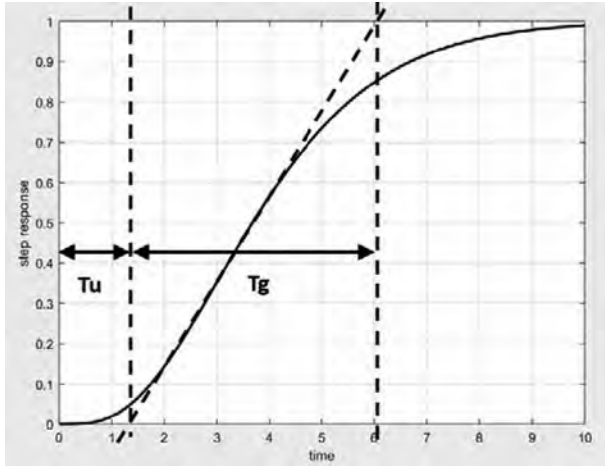


Figure 1: Step response of a PTn- Systems, turning point tangent and subdivision into Tu and Tg.

n, PTn	Tg/T1	Tu/T1	Tg/Tu
2, PT2	2.72	0.28	9.65
3, PT3	3.69	0.81	4.59
4, PT4	4.46	1.43	3.13
5, PT5	5.12	2.10	2.44
6, PT6	5.70	2.81	2.03

Table 1: Calculation of Tg, Tu, T1 and PTn

The parameters in table 1 can be calculated using formulas 2 to 4, for different system order n.

$$\frac{T_g}{T_1} = \frac{(n-1)!}{(n-1)^{n-1}} \cdot e^{n-1} \quad (2)$$

$$\frac{T_u}{T_1} = n - 1 - \frac{(n-1)!}{(n-1)^{n-1}} \cdot \left[e^{n-1} - \sum_{m=0}^{n-1} \frac{(n-1)^m}{m!} \right] \quad (3)$$

$$\frac{T_g}{T_u} = \frac{\frac{(n-1)!}{(n-1)^{n-1}} \cdot e^{n-1}}{n - 1 - \frac{(n-1)!}{(n-1)^{n-1}} \cdot \left[e^{n-1} - \sum_{m=0}^{n-1} \frac{(n-1)^m}{m!} \right]} \quad (4)$$

3 ITAE, IAE and ISE criteria

The block diagram of the controlled system is shown in figure 2. The parameters found for the PTn system are Ks, T1 and n.

Among others, the criteria IAE, ITAE and ISE are used for optimizing, which describe the error area of a step response of the controlled system. These error areas are shown in figure 3.

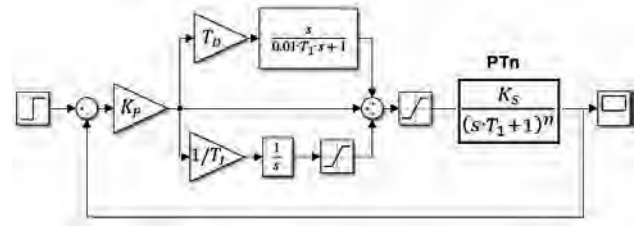


Figure 2: Block diagram of a PTn- systems, which is controlled by a PID controller.

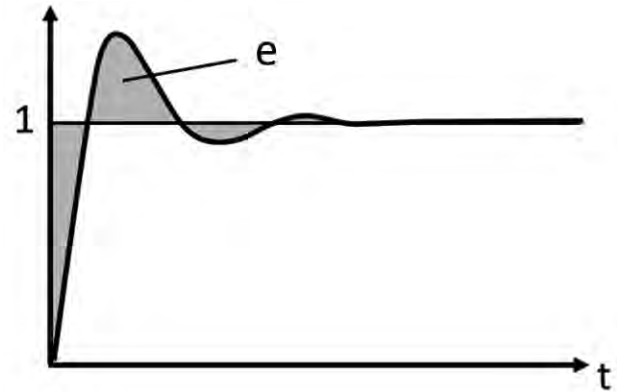


Figure 3: Error area in the transient response of the closed-loop system according to figure 2, for calculating the IAE, ITAE and ISE criteria.

IAE means integral of absolute error, ITAE means integral of time-multiplied absolute value of error and ISE means integral of squared error. It can be seen from this that the IAE criterion calculates the amount of the error area. The ITAE criterion is an extension of the IAE criterion and also takes time into account. Thus, the error area is weighted more heavily as time progresses. The two criteria IAE and ITAE are also called the L1 criterion. The ISE criterion does not calculate the error area, but its square. This means that there it is no need to calculate the

absolute values of the error area, since negative signs cancel each other out when squared. The ISE criterion is also called the L2 criterion.

$$IAE: \int_0^{\infty} |e(t) - e(\infty)| dt \quad (5)$$

$$IITAE: \int_0^{\infty} |e(t) - e(\infty)| \cdot t \cdot dt \quad (6)$$

$$ISE: \int_0^{\infty} [e(t) - e(\infty)]^2 dt \quad (7)$$

4 Calculation of PID parameters with the hill climbing learning method

Now, since in chapter 1 the time-delayed systems were approximated with the turning point tangent method as PTn elements, the quality criteria for step responses calculated for all orders n and also all PID controller parameters Kp, Ti and Td. The parameters which correspond to the minimum of the criteria can then be displayed as table values. The problem is that this has to be done for a multidimensional space (order n, Kp, Ti, Td, quality criteria). So it would take far too long with the computing power available today. Therefore, ‘hill climbing’, a method from the field of artificial intelligence was chosen [4]. With this method, heuristic functions are added to some of the parameters, in this case the parameters of the PID controller, and then it is calculated whether the quality criteria IAE, ITAE and ISE have become smaller. If there is, the new parameters will be used as a reference. If not, the old ones stay. In this way and after many iterations, the final values of the parameters remain at local minima of the quality criteria. The method requires much less computing time than a complete calculation in multidimensional space, for example with nested loops of all parameters. With the 5 parameters, order n, Kp, Ti, Td, quality criteria, this would have the time complexity $f(n) = O(n^5)$.

However, since it only finds local minima, several different random tuples of starting values for the control parameters are used. Since many of the results of the converged parameters for the minimal quality criteria then

agree with one another, it can be assumed with reasonably good certainty that the parameters found are actually PID parameters Kp, Ti and Td, which either correspond to the absolute minima of the criteria, or which come very close to these at least.

The search for optimal parameters in multi-dimensional space, as with this specific problem in control engineering, is also one of the good arguments for using an artificial intelligence method here as well. Often, complete calculations cannot be carried out in the entire parameter space due to the computing power available. Since only part of the parameter space is calculated with such methods, the computing time is significantly reduced and the results are parameter sets for excellent transient behavior.

5 PID controller parameters after the minimization of the quality criteria IAE, ITAE and ISE

This chapter is the essence of the publication. The table below describes the PID controller parameters calculated with Matlab / Simulink and the ‘hill climbing’ method according to the minimized quality criteria IAE, ITAE and ISE. The block diagram in figure 2 serves as the basis for this. It is particularly noteworthy that the static gain Ks and in particular the time constant T1 of the n identical PT1 elements are included in the table. This makes them usable and scalable for all PTn systems. The values up to n = 6 are shown here. The controller output limitation is implemented on the one hand after the controller and on the other hand also after the integrator (anti windup) and is assumed to be +/- 2, +/- 3, +/- 5, +/- 10. In the calculations the anti windup is never active, but it is inserted anyway, because in practice it can happen for various reasons that the controlled variable does not reach the desired variable in the static end value.

The controller output limitation is calculated as (maximum possible controller output - controller output before the step) divided by (controller output for the stationary end value - controller output before the step). In many cases, the controller output before the step is equal to 0, thus the controller output limitation is calculated as maximum possible controller output divided by controller output for the stationary end value. In the table, the maximum parameter value is limited to 10.

PT1	+/- 2	+/- 3	+/- 5	+/- 10
Tg/Tu: 9.65				
IAE	Kp·Ks = 10 Ti = 3.1·T1 Td = 0 (PI)	Kp·Ks = 10 Ti = 2·T1 Td = 0 (PI)	Kp·Ks = 10 Ti = 1.3·T1 Td = 0 (PI)	Kp·Ks = 10 Ti = 1·T1 Td = 0 (PI)
ITAE	Kp·Ks = 9.3 Ti = 2.9·T1 Td = 0 (PI)	Kp·Ks = 9.5 Ti = 1.9·T1 Td = 0 (PI)	Kp·Ks = 9.1 Ti = 1.2·T1 Td = 0 (PI)	Kp·Ks = 10 Ti = 1·T1 Td = 0 (PI)
ISE	Kp·Ks = 10 Ti = 2.7·T1 Td = 0 (PI)	Kp·Ks = 10 Ti = 1.6·T1 Td = 0 (PI)	Kp·Ks = 9.8 Ti = 1.5·T1 Td = 0 (PI)	Kp·Ks = 10 Ti = 0.2·T1 Td = 0 (PI)
PT2	+/- 2	+/- 3	+/- 5	+/- 10
Tg/Tu: 9.65				
IAE	Kp·Ks = 10 Ti = 9.6·T1 Td = 0.3·T1	Kp·Ks = 10 Ti = 7.3·T1 Td = 0.3·T1	Kp·Ks = 10 Ti = 5.6·T1 Td = 0.3·T1	Kp·Ks = 10 Ti = 3.7·T1 Td = 0.2·T1
ITAE	Kp·Ks = 10 Ti = 9.6·T1 Td = 0.3·T1	Kp·Ks = 10 Ti = 7.3·T1 Td = 0.3·T1	Kp·Ks = 9.6 Ti = 5.4·T1 Td = 0.3·T1	Kp·Ks = 9.8 Ti = 4.7·T1 Td = 0.3·T1
ISE	Kp·Ks = 10 Ti = 9.7·T1 Td = 0.2·T1	Kp·Ks = 10 Ti = 7.3·T1 Td = 0.2·T1	Kp·Ks = 10 Ti = 5.1·T1 Td = 0.2·T1	Kp·Ks = 10 Ti = 4.6·T1 Td = 0.1·T1
PT3	+/- 2	+/- 3	+/- 5	+/- 10
Tg/Tu: 4.59				
IAE	Kp·Ks = 5.4 Ti = 9.4·T1 Td = 0.7·T1	Kp·Ks = 7 Ti = 10·T1 Td = 0.7·T1	Kp·Ks = 8.4 Ti = 9.8·T1 Td = 0.7·T1	Kp·Ks = 10 Ti = 9.7·T1 Td = 0.7·T1
ITAE	Kp·Ks = 5.4 Ti = 9.4·T1 Td = 0.7·T1	Kp·Ks = 7 Ti = 10·T1 Td = 0.7·T1	Kp·Ks = 8.2 Ti = 9.6·T1 Td = 0.7·T1	Kp·Ks = 10 Ti = 9.7·T1 Td = 0.7·T1
ISE	Kp·Ks = 6.1 Ti = 10·T1 Td = 0.6·T1	Kp·Ks = 8.1 Ti = 9.8·T1 Td = 0.6·T1	Kp·Ks = 10 Ti = 10·T1 Td = 0.6·T1	Kp·Ks = 10 Ti = 7.8·T1 Td = 0.6·T1
PT4	+/- 2	+/- 3	+/- 5	+/- 10
Tg/Tu: 3.13				
IAE	Kp·Ks = 2 Ti = 5.2·T1 Td = 1.1·T1	Kp·Ks = 2.9 Ti = 6.5·T1 Td = 1.2·T1	Kp·Ks = 3.3 Ti = 7.1·T1 Td = 1.3·T1	Kp·Ks = 3.3 Ti = 6.9·T1 Td = 1.3·T1
ITAE	Kp·Ks = 1.9 Ti = 5·T1 Td = 1.1·T1	Kp·Ks = 2.4 Ti = 5.9·T1 Td = 1.2·T1	Kp·Ks = 2.3 Ti = 5.7·T1 Td = 1.2·T1	Kp·Ks = 2.1 Ti = 5·T1 Td = 1.1·T1
ISE	Kp·Ks = 2.8 Ti = 6.6·T1 Td = 1.2·T1	Kp·Ks = 3.6 Ti = 7·T1 Td = 1.2·T1	Kp·Ks = 4.9 Ti = 7.1·T1 Td = 1.4·T1	Kp·Ks = 5.2 Ti = 7·T1 Td = 1.4·T1

PT5	+/- 2	+/- 3	+/- 5	+/- 10
Tg/Tu: 2.44				
IAE	Kp·Ks = 1.7 Ti = 5.8·T1 Td = 1.6·T1	Kp·Ks = 1.8 Ti = 5.9·T1 Td = 1.6·T1	Kp·Ks = 1.8 Ti = 5.8·T1 Td = 1.6·T1	Kp·Ks = 1.7 Ti = 5.5·T1 Td = 1.6·T1
ITAE	Kp·Ks = 1.4 Ti = 5.3·T1 Td = 1.4·T1	Kp·Ks = 1.4 Ti = 5.2·T1 Td = 1.4·T1	Kp·Ks = 1.4 Ti = 5.2·T1 Td = 1.4·T1	Kp·Ks = 1.4 Ti = 5.0·T1 Td = 1.4·T1
ISE	Kp·Ks = 1.9 Ti = 5.9·T1 Td = 1.7·T1	Kp·Ks = 2.6 Ti = 6.5·T1 Td = 1.8·T1	Kp·Ks = 2.5 Ti = 6.3·T1 Td = 1.8·T1	Kp·Ks = 2.5 Ti = 6.1·T1 Td = 1.8·T1
PT6	+/- 2	+/- 3	+/- 5	+/- 10
Tg/Tu: 2.03				
IAE	Kp·Ks = 1.3 Ti = 5.9·T1 Td = 1.9·T1	Kp·Ks = 1.3 Ti = 5.8·T1 Td = 1.9·T1	Kp·Ks = 1.3 Ti = 5.8·T1 Td = 1.9·T1	Kp·Ks = 1.3 Ti = 5.6·T1 Td = 1.9·T1
ITAE	Kp·Ks = 1.1 Ti = 5.5·T1 Td = 1.7·T1	Kp·Ks = 1.1 Ti = 5.5·T1 Td = 1.7·T1	Kp·Ks = 1.1 Ti = 5.4·T1 Td = 1.7·T1	Kp·Ks = 1.1 Ti = 5.3·T1 Td = 1.7·T1
ISE	Kp·Ks = 1.8 Ti = 6.8·T1 Td = 2.1·T1	Kp·Ks = 1.8 Ti = 6.5·T1 Td = 2.1·T1	Kp·Ks = 1.8 Ti = 6.5·T1 Td = 2.1·T1	Kp·Ks = 1.8 Ti = 6.3·T1 Td = 2.1·T1

Table 2: Table values of the PID parameters for the minimum IAE, ITAE and ISE criterions of controlled PTn or time delayed systems.

It is noteworthy that the table scales with T1 and Ks. The results are therefore very widely applicable.

6 Applications of table values

6.1 Control of a PT3 system

In the first application example, a didactic example is used to show the general usability of the parameter table. The response of a time-delayed system to a unit jump shows a static end value of 1, a delay time Tu of 0.81 seconds and a rise time Tg of 3.69 seconds. This results in Tg / Tu = 4.59 and this results in a PT3 behavior with Ks = 1 and T1 = 1 second.

For the ITAE criterion, the table values of the PID parameters for the PT3 system are read off. Since Ks = 1 and T1 = 1s, the table values are multiplied by 1 and therefore correspond to those for the controller parameters Kp, Ti and Td. The simulation of the step responses of the closed loop system according to figure 2 is shown in figure 4. It shows a very nice transient response. The

different dynamics or rise times can be explained with the different controller output limitations. This also shows very well that these must be included into the controller design.

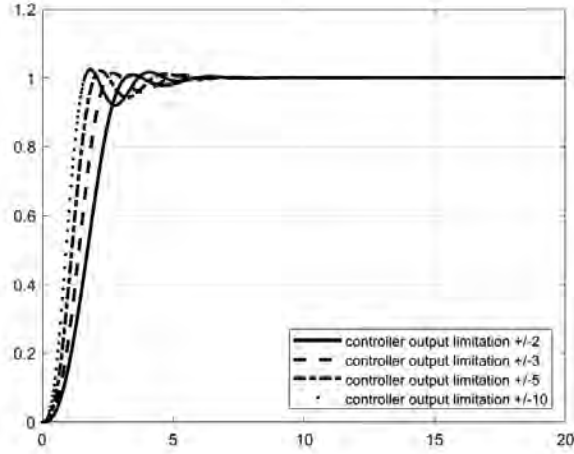


Figure 4: Step response of the closed loop system according to figure 2 for a PT3 system with the PID table values for the ITAE criterion.

6.2 Control of a PT2 system, comparison with Ziegler Nichols and Chien, Hrones and Reswick

In the following, the controller parameters found are compared with those of Ziegler-Nichols and Chien, Hrones and Reswick, using an example of second order. The used system has an order $n = 2$ and a time constant T_1 of 8s. As a comparison to practical systems this could be a thermal system, where the heating coil and the temperature sensor are not exactly located at the same place.

$$\frac{K_s}{(s \cdot T_1 + 1)^n} = \frac{1}{(s \cdot 8 + 1)^2} \quad (8)$$

The controller output signal for the stationary end value of the controlled system is 1 and the controller output signal is limited to ± 2 , which results in a controller output limitation factor ± 2 . Using the table 1, it results for PT2 a $T_g = 21.76$ seconds and $T_u = 2.24$ seconds. According to the Ziegler-Nichols step response method, controlled systems with dead time and a PT1 are treated. In this case, T_u is assumed to be the dead time and T_g as the time constant. This results in the controller parameters:

$$K_p = \frac{1.2 \cdot T_g}{K_s \cdot T_u} = 11.65, \quad T_i = 2 \cdot T_u = 4.48s$$

$$T_d = 0.5 \cdot T_u = 1.12s$$

According to Chien, Hrones and Reswick with the parameters for 'aperiodic', the result is:

$$K_p = \frac{0.6 \cdot T_g}{K_s \cdot T_u} = 5.83, \quad T_i = 1.0 \cdot T_g = 21.76s$$

$$T_d = 0.5 \cdot T_u = 1.12s$$

The method calculated above with the parameters according to the minimal ITAE criterion provides a T_1 of 8s and $n = 2$ according to Table 1, i.e. a PT2 behavior. This results in the following parameters from table 2:

$$K_p = \frac{10}{K_s} = 10, \quad T_i = 9.6 \cdot T_1 = 76.8s,$$

$$T_d = 0.3 \cdot T_1 = 2.4s$$

The simulation according to the block diagram according to figure 1 (PT2 with PID) shows the results according to figure 5 for the three parameter sets.

The rise time is similar for all three parameter sets, because all systems run into the controller output limitation in this phase. It shows very nicely that the calculated values with the minimum ITAE criterion according to Table 2 show an excellent transient behavior.

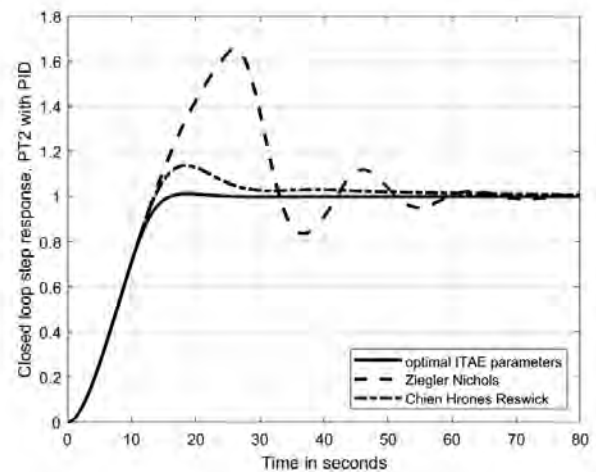


Figure 5: Comparison of the step responses with the control parameters for a PT2 plant, with a control output limitation factor ± 2 .

6.3 Control of a general PT4 system, comparison with $K_s \neq 1$ and $T_1 \neq 1$

In the next example, a general problem is dealt with in order to also show the scalability of the presented parameter table. There was measured the response to a unit step and it is shown in figure 6.

With the application of the turning point tangent method according to figure 1 and table 1, the step response leads to the following transfer function, with $K_s = 0.4$ and $T_1 = 0.5s$:

$$G = \frac{0.4}{(s \cdot 0.5 + 1)^4} \quad (9)$$

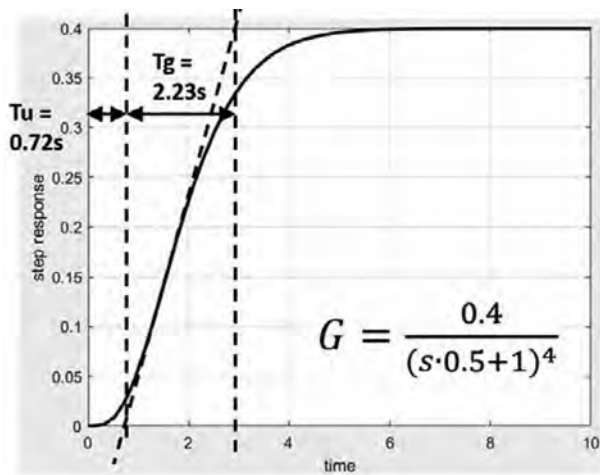


Figure 6: Step response of a PT4 system.

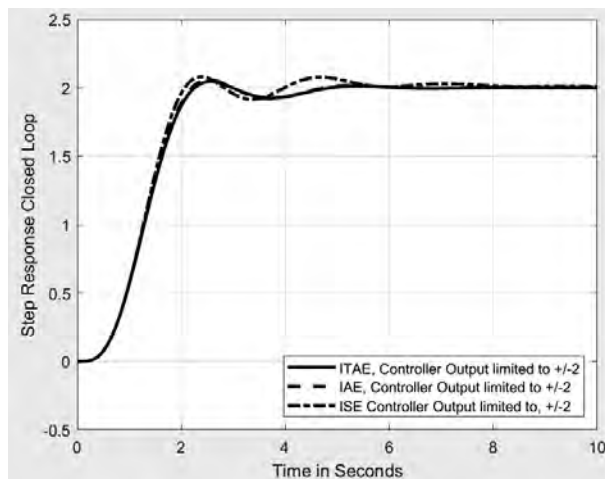


Figure 7: Response of a closed loop, PID with PT4 for a step from 0 to 2

One would like to design the system with a PID controller

according to figure 2 and execute a setpoint jump from 0 to 2. Since the static gain $K_s = 0.4$, the controller output for the stationary end value is then $2 / K_s$, or $2 / 0.4 = 5$. Assuming that the controller output is limited to ± 10 , the result is a controller output limitation factor $\pm 10/5 = \pm 2$.

The controller parameters are to be calculated for the IAE criterion as an example.

$$K_p = \frac{2}{K_s} = 5 \quad T_i = 5.2 \cdot T_1 = 2.6s$$

$$T_d = 1.1 \cdot T_1 = 0.55s$$

For minimizing the criteria IAE, ITAE and ISE, this results in the closed loop behavior according to figure 7 for a setpoint jump from 0 to 2.

7 Discussion and outlook

Good transient behavior can be seen for all parameter sets in the table. Compared with heuristic methods, these parameters are hard-calculated values that minimize the quality criteria. It is also up to the discussion what would happen if one would perform different jumps and therefore had to choose the parameters according to different factors of the controller output limitation. The parameters are very similar, however, and values for jumps should be selected which are most likely to occur in the specific system. Even for the general setpoint jump by any value, the parameters still give very good transient behavior. An exciting finding emerges from the discussion of the question of how the parameters K_p , T_i and T_d develop for changing ratios T_g / T_u (i.e. rise time in relation to the delay time).

The smaller the ratio, the greater the delay time in relation to the rise time, the smaller K_p on the one hand and the greater T_d on the other hand.

The effect of a small K_p means that the system can only be regulated slowly. In the literature [10] this is also described in such a way that the controllability for systems with longer dead times is reduced. If one were to also plot the manipulated variable, one would see that this is also only relatively small. Therefore it is of no use in these systems if an additional regulator reserve is made available through amplifiers, because this cannot be used at all due to the time delay of the system.

If you follow the development of the value of K_p in the

tables, then with smaller ratios T_g / T_u (or larger orders n of the PTn systems and thus larger dead times) and smaller K_p , greater system dynamics are no longer achieved. On the one hand, the controlled variable shows a nice transient response according to the minimized quality criteria and, in particular, also reaches the setpoint in the stationary end value, which is often sufficient in practice.

When looking at the differential component of the controller T_d , it becomes apparent that a differential component for optimizing the quality criteria is missing when regulating frequently occurring PT1 elements, i.e. a pure PI controller is already optimal. With an increasing system order, i.e. a decreasing ratio T_g / T_u or a larger delay, the required D component (T_d) becomes larger and larger.

It turns out that the PTn systems that occur very frequently in practice can be regulated very well with the table values available according to the minimized IAE, ITAE and ISE criteria. In practice, you can often do without a simulation and only measure the step response of the system. Then the order n and the associated parameters for the PID controller can be read from the table, also using K_s and T_I and implement the controller directly on the system.

References

- [1] Ziegler, J. B., Nichols N. B., Optimum settings for automatic controllers, ASME Transactions, v64 (1942), pp. 759-768
- [2] Kun Li Chien Kun Li, Hrones, J. A., Reswick, J. B., On the Automatic Control of Generalized Passive Systems. In: Transactions of the American Society of Mechanical Engineers., Bd. 74, Cambridge (Mass.), USA, Feb. 1952, S. 175–185
- [3] Qi, Zhi, Qian Shi, and Hui Zhang. "Tuning of digital PID controllers using particle swarm optimization algorithm for a CAN-based DC motor subject to stochastic delays." IEEE Transactions on Industrial Electronics 67.7 (2019): 5637-5646.
- [4] Russel, Stuart J, Norvig, Peter, Artificial Intelligence: A Modern Approach (2nd edition), Upper Saddle River, New Jersey: Prentice Hall, pp 111- 114, ISBN 0-13-790395-2, 2003
- [5] Joseph, E. A., Olaiya, O. O., Cohen- Coon PID Tuning Method, A Better Option to Ziegler Nichols- PID Tuning Method, Computer Engineering and Intelligent Systems, ISSN 2222-1719, Vol. 9, No. 5, 2018
- [6] Ozana Stephan, Docekal, Tomas., PID Controller Design Based on Global Optimization Technique with Additional Constraints, Journal of Electrical Engineering, Vol. 67, No3, pp 160 – 168, 2016
- [7] K. M. Hussain et al, Comparison of PID Controller Tuning Methods with Genetic Algorithm for FOPTD System, Int. Journal of Engineering Research and Applications, ISSN: 2248-9622, Vol. 4, Issue 2, pp. 308-314, February 2014
- [8] Büchi, Roland. State space control, LQR and observer: step by step introduction with Matlab examples. Nordersedt Books on Demand, 2010.
- [9] da Silva LR, Flesch RC, Normey-Rico JE. Controlling industrial dead-time systems: When to use a PID or an advanced controller. ISA transactions. 2020 Apr 1;99:339-50.
- [10] Unbehauen H. Regelungstechnik. Braunschweig: Vieweg; 1992.
- [11] S. Zacher, M. Reuter: Regelungstechnik für Ingenieure. 15.Auflage, Springer Vieweg Verlag, 2017
- [12] G. Schwarze: Bestimmung der regelungstechnischen Kennwerte von P-Gliedern aus der Übergangsfunktion ohne Wendetangentenkonstruktion, In: - messen-steuern-regeln Heft 5, S. 447-449, 1962
- [13] Martins, Fernando G., Tuning PID Controllers using the ITAE Criterion, Int. Journal Engineering, Edition. Vol. 21, No. 5. Pp. 867-873, 2005
- [14] Büchi, Roland. "PID Controller Parameter Tables for Time-Delayed Systems Optimized Using Hill-Climbing." Signals 3.1 (2022): 146-156.
- [15] Guillermo J. Silva, Aniruddha Datta. S. P. Bhattacharyya. "PID Controllers for Time-Delay Systems". Boston. ISBN 0-8176-4266- 8.2005.

Interacting-Multiple-Model Partikelfilter zur Fehleridentifikation

Adrian Lepp¹, Dirk Weidemann¹

¹Institut für Systemdynamik und Mechatronik, FH Bielefeld, Interaktion 1, 33619 Bielefeld;
{adrian.lepp,dirk.weidemann}@fh-bielefeld.de

Kurzfassung. Aktive fehlertolerante Regelungen erfordern nicht nur die Fehlererkennung, sondern auch die Lokalisierung des Fehlers und die Schätzung der Fehlergröße, was als Fehleridentifikation bezeichnet wird. Um die Fehlerlokalisierung und -identifikation für Aktor-, Sensor- und Prozessfehler zu ermöglichen, eignen sich Ansätze, die auf der gleichzeitigen Verwendung mehrerer Modelle basieren, sodass die Fehlerlokalisierung und -identifikation als Mehrmodellschätzung, auch bekannt als hybride Schätzung, durchgeführt werden können. Typische hybride Schätzer wie Generalized Pseudo Bayesian (GPB)-Ansätze oder der Interacting Multiple Model (IMM)-Ansatz verwenden häufig erweiterte Kalman- oder unscented Kalman-Filter. Bei nicht normalverteilten Störungen und/oder stark ausgeprägten Nichtlinearitäten führen diese Filter oft zu unbefriedigenden Schätzungen. In derartigen Fällen bieten sich Partikelfilter an. Die direkte Verwendung von Partikelfiltern in Kombination mit hybriden Partikeln kann jedoch zu einer Degeneration der Partikel führen. Das IMM Partikelfilter (IMMPF), das aus dem exakten Bayes-Filter für den hybriden Zustand abgeleitet werden kann, vermeidet dieses Problem, indem es die Dichte jedes Modus mit einer konstanten Anzahl von Partikeln approximiert. Am Beispiel eines Dreitank-Systems wird die Fehleridentifikation mit Hilfe des IMMPF-Ansatzes erläutert und die Vor- und Nachteile des zugrundeliegenden Modellierungsansatzes, der eine Quantisierung der Fehlergröße verwendet, werden diskutiert.

Einleitung

Bei allen technischen Systemen und industriellen Prozessen können während des Betriebs Fehler auftreten, welche die Nutzung des Systems bzw. den Betrieb des Prozesses einschränken oder gar zu lebensbedrohlichen Szenarien für den Nutzer bzw. Betreiber führen. Während redundante Hardware zum Einsatz kommt, um den zweiten Fall, d.h. lebensbedrohliche Situationen zu vermeiden, verwendet man im erstgenannten Fall eine analytische Redundanz. Hierbei finden softwaretechnisch realisierte Diagnosefunktionen Verwendung, die vielfach mit aktiven fehlertoleranten Regelungsstrategien kombiniert werden.

Entsprechend [1] erfordern aktive fehlertolerante

Regelungen normalerweise nicht nur die Lokalisierung des Fehlers, sondern auch die Bestimmung der Fehleramplitude, was als Fehleridentifikation bezeichnet wird. Während für die reine Fehlerdetektion als auch für die Fehlerlokalisierung bereits zahlreiche modellbasierte Ansätze existieren (vgl. [1],[4]), gibt es nur eine geringe Anzahl an Ansätzen für die Fehleridentifikation. Ferner sind diese typischerweise entweder für die Identifikation von Aktor- bzw. Sensorfehlern oder Prozessfehlern geeignet.

Einen Zugang zur Identifikation von Sensor-, Aktor- und Prozessfehlern bieten auf Multi-Modell Verfahren basierende Zustandsschätzer. Derartige Ansätze sind aus dem Bereich der Objektverfolgung wohlbekannt (s. [9]) und können darüber hinaus wie in [12] auch sinnvoll für die Fehlerdiagnose eingesetzt werden, wenn zusätzlich zu einem Modell des fehlerfreien Systems für jeden zu berücksichtigenden Fehlerfall ein weiteres Modell verwendet wird. Folglich modelliert eine endliche Menge von Modellen \mathcal{M} das Verhalten des Systems, wobei $q \in \mathbb{N}$ die Mächtigkeit von \mathcal{M} angibt. Die diskrete Dynamik, d.h. der Wechsel zwischen diesen Modellen, die auch als Moden bezeichnet werden, wird in der Regel durch eine Markov-Kette modelliert, während die kontinuierliche Dynamik jedes Modells aus \mathcal{M} entweder durch einen Satz Differential- oder wie in [12] Differential-Algebraischer-Gleichungen gegeben ist. Das Ziel ist, sowohl den diskreten als auch den kontinuierlichen Zustand, d.h. einen hybriden Zustand zu schätzen. Daher gehören diese Verfahren der Zustandsschätzung zur Klasse der hybriden Zustandsschätzer, s. [8].

Zur Schätzung des hybriden Zustands wird bei Multi-Modell Zustandsschätzern ein stochastisches Filter für jedes Modell genutzt. Hierbei finden bei nichtlinearen Systemen vielfach erweiterte Kalman- oder unscented Kalman-Filter Verwendung. Insbesondere wenn aufgrund nicht normalverteilter Störungen und/oder stark ausgeprägter Nichtlinearitäten diese Fil-

tervarianten nicht mehr zu zufriedenstellenden Schätzungen führen, bieten sich Partikelfilter an. Ein Partikelfilter ist ein sequentielles Monte-Carlo-Verfahren zur Schätzung der Zustände eines stochastisch gestörten dynamischen Systems auf der Grundlage verrauschter Messungen, vgl. [5].

Ein naheliegender Ansatz zur hybriden Zustandsschätzung mittels Partikelfilter basiert auf der Nutzung bereits bekannter Partikelfilteralgorithmen in Kombination mit hybriden Partikeln, vgl. [10]. Nachteilig erweist sich bei einem derartigen Ansatz, dass die Anzahl der Partikel in einem Mode proportional zur Modewahrscheinlichkeit ist. Folglich liegt bei Moden mit geringer Wahrscheinlichkeit eine schlechte Approximationsgüte vor.

Diesen Nachteil vermeiden die in [3],[6] präsentierten Ansätze durch Gliederung der hybriden Zustandsschätzung in eine Schätzung der modespezifischen Wahrscheinlichkeitsdichtefunktionen und der Modewahrscheinlichkeiten. Die Dichte des hybriden Zustands ergibt sich gemäß der Bayes-Regel aus der Multiplikation der Modewahrscheinlichkeit mit der zugehörigen modespezifischen Wahrscheinlichkeitsdichtefunktion. Hierbei erfolgt in beiden Ansätzen die Bestimmung der modespezifischen a-Posteriori Dichte mit Hilfe von Partikelfiltern.

Obgleich beide Ansätze die modespezifischen a-Priori Dichten nach dem Modewechsel entsprechend dem Interaktionsschritt des Interacting-Multiple-Modell Ansatzes [2] bestimmen und rein reellwertige Partikel verwenden, weisen die Ansätze Unterschiede auf. In [6] erfolgt die Bestimmung der Partikel, mit denen die modespezifischen a-Priori Dichten approximiert werden, in einem zweistufigen Verfahren. Im ersten Schritt wird auf der Grundlage der Mischwahrscheinlichkeiten zufällig eine der modespezifischen a-Posteriori Dichten gewählt. Im zweiten Schritt wird zufällig ein Partikel entsprechend der Partikelgewichte gewählt. Im Unterschied zu [6] werden in [3] die Modewahrscheinlichkeiten durch die Summe der Partikelgewichte dargestellt, wodurch jedes Partikel für jeden Mode eine eigene Mischwahrscheinlichkeit erhält und die modespezifischen a-Priori Dichten aus allen Partikeln der modespezifischen a-Posteriori Dichten gebildet werden. Folglich weisen die modespezifischen a-Priori Dichten eine q mal höhere Anzahl an Partikeln als die modespezifischen a-Posteriori Dichten auf, was einen Resampling Schritt erfordert, wenn die modespezifische Filterung stets mit gleichbleibender Anzahl an

Partikel durchgeführt werden soll.

Ein weiterer Unterschied ist, dass die Gewichtung der Partikel und das Update der Modewahrscheinlichkeiten in [3] in einen Kombinationsschritt erfolgt, während die Berechnung in [6] getrennt geschieht.

Der nachfolgende Ansatz zur Fehleridentifikation nutzt den in [3] vorgestellten IMM Partikelfilter (IMMPF), da dieser IMMPF-Ansatz auch für hybride Systeme anwendbar ist, bei denen die Modeübergangswahrscheinlichkeit sowohl vom aktuellen Mode als auch vom aktuellen wertkontinuierlichen Zustand abhängig ist. Der zugrunde liegende Modellierungsansatz wird in Abschnitt 1 dargestellt. In Abschnitt 2 werden zunächst die exakten rekursiven Gleichungen des Bayes-Filters hergeleitet, wobei sich zeigt, dass die Berechnung der modespezifischen a-Priori Dichte nach dem Modewechsel analog zu der des IMM-Ansatzes erfolgt. Anschließend werden die Dichten durch Partikel approximiert, um den IMMPF-Algorithmus zu erhalten. In Abschnitt IV wird anhand von Simulationsergebnissen die Effizienz des vorgeschlagenen Ansatzes veranschaulicht.

1 Modellierung

Um fehlerbehaftete Systeme zu modellieren, gilt es, zwischen den Orten, an denen Fehler auftreten können, und dem zeitlichen Verlauf ihrer Amplitude zu unterscheiden. Da Fehler sowohl Sensoren, Aktoren als auch den Prozess selbst betreffen können, werden diese Fehler üblicherweise als Sensor-, Aktor- und Prozessfehler bezeichnet. Obgleich die Fehleramplitude einen beliebigen zeitlichen Verlauf aufweisen kann, finden sich Fehler mit abrupt wechselnder Amplitude am häufigsten.

Hinsichtlich der Ermittlung der Amplitude eines abrupten Fehlers besteht ein möglicher Ansatz darin, die Fehlergröße als einen erweiterten Zustand zu modellieren. Allerdings führt eine derartige Vorgehensweise zu einem hybriden Modell mit heterogenen Zustandsräumen. Dieses Problem wird für den IMM-Ansatz in [7] und GPB-Ansätze in [11] behandelt, stellt jedoch im Zusammenhang mit dem IMMPF-Ansatz ein offenes Problem dar. Vor diesem Hintergrund wird die Fehlergröße quantisiert, sodass die Anzahl der Modelle pro Fehler der Anzahl der Quantisierungsstufen entspricht. Dies führt zu einem stochastischen hybriden Modell, bei dem die wertkontinuierliche Dynamik jedes Modes

durch eine Zustandsraumdarstellung der Form

$$x_k = f(m_k, x_{k-1}, w_{k-1}, u_{k-1}), \quad (1)$$

$$y_k = h(m_k, x_k, u_k, v_k) \quad (2)$$

gegeben ist, während die wertdiskrete Dynamik der diskreten Übergangsabbildung

$$m_k = c(m_{k-1}) \quad (3)$$

genügt. Hierbei kennzeichnen $x_k \in \mathbb{R}^n$ die Zustände der wertkontinuierlichen Dynamik, $y_k \in \mathbb{R}^r$ den Messausgang und $m_k \in \mathcal{M}$ die Zustände der wertdiskreten Dynamik. Letztere werden im Folgenden als Moden bezeichnet, wobei der hybride Zustand mit $(x_k, m_k) \in \mathbb{R}^n \times \mathcal{M}$ gegeben ist. Darüber hinaus sind w_k und v_k unabhängige und identisch verteilte Zufallsvariablen.

2 IMMPF

Entsprechend der Modellierung des fehlerbehafteten Systems als stochastisches hybrides System kann die Fehleridentifikation durch Schätzung der bedingten Dichte des hybriden Zustands $p(x_k, m_k^{(j)} | Y_k, U_k)$ für alle $j \in \{1, 2, \dots, q\}$ erfolgen, wobei $Y_k = (y_0, y_1, \dots, y_k)$ die Sequenz der Messungen und $U_k = (u_0, u_1, \dots, u_k)$ die der Stellgrößen repräsentieren. Unter der Annahme, dass $p(x_{k-1}, m_{k-1}^{(i)} | Y_{k-1}, U_{k-1})$ für alle $i \in \{1, 2, \dots, q\}$ bekannt ist, lässt sich die Schätzung rekursiv durchführen. In Anlehnung an [3] werden hierfür zunächst die rekursiven Gleichungen des Bayes-Filters exakt bestimmt. Hierbei zeigt sich, dass die Bestimmung der a-Priori Dichte des hybriden Zustands nach dem Modewechsel dem Interaktionsschritt des IMM-Ansatzes ähnelt, vgl. [2]. Abschließend werden die Dichten durch Partikel approximiert, was zum IMMPF-Algorithmus führt.

2.1 Exaktes rekursives Bayes-Filter

Modewechsel. Gemäß der Bayes-Regel ist die Dichte des hybriden Zustands nach dem Modewechsel durch

$$p(x_{k-1}, m_k^{(j)} | Y_{k-1}, U_{k-1}) = p(x_{k-1} | m_k^{(j)}, Y_{k-1}, U_{k-1}) P(m_k^{(j)} | Y_{k-1}, U_{k-1}) \quad (4)$$

gegeben, wobei $p(x_{k-1} | m_k^{(j)}, Y_{k-1}, U_{k-1})$ die modespezifische a-Priori Dichte und $P(m_k^{(j)} | Y_{k-1}, U_{k-1})$ die a-

Priori Modewahrscheinlichkeit darstellen. Durch Marginalisierung von (4) über x_{k-1} und Nutzung des Satzes der totalen Wahrscheinlichkeit folgt für die a-Priori Modewahrscheinlichkeit

$$\begin{aligned} P(m_k^{(j)} | Y_{k-1}, U_{k-1}) &= \int_{\mathbb{R}^n} p(x_{k-1}, m_k^{(j)} | Y_{k-1}, U_{k-1}) dx_{k-1} \\ &= \int_{\mathbb{R}^n} \sum_{i=1}^q p(x_{k-1}, m_k^{(j)}, m_{k-1}^{(i)} | Y_{k-1}, U_{k-1}) dx_{k-1} \quad (5) \\ &= \int_{\mathbb{R}^n} \sum_{i=1}^q p(m_k^{(j)} | m_{k-1}^{(i)}, x_{k-1}, Y_{k-1}, U_{k-1}) \\ &\quad p(x_{k-1}, m_{k-1}^{(i)} | Y_{k-1}, U_{k-1}) dx_{k-1}. \end{aligned}$$

Im Folgenden wird davon ausgegangen, dass $m_k = c(m_{k-1})$ durch eine Markov-Kette modelliert wird. Daher vereinfacht sich (5) zu

$$\begin{aligned} P(m_k^{(j)} | Y_{k-1}, U_{k-1}) &= \int_{\mathbb{R}^n} \sum_{i=1}^q \pi_{ij} p(x_{k-1}, m_{k-1}^{(i)} | Y_{k-1}, U_{k-1}) dx_{k-1}. \quad (6) \end{aligned}$$

Hierbei ist $\pi_{ij} = P(m_k^{(j)} | m_{k-1}^{(i)}) \forall i, j \in \{1, 2, \dots, q\}$ die zeitinvariante Modeübergangswahrscheinlichkeit. Nutzt man den Satz der totalen Wahrscheinlichkeit wie in (5), ergibt sich die modespezifische a-Priori Dichte nach dem Modewechsel zu

$$\begin{aligned} p(x_{k-1} | m_k^{(j)}, Y_{k-1}, U_{k-1}) &= \frac{\sum_{i=1}^q \pi_{ij} p(x_{k-1}, m_{k-1}^{(i)} | Y_{k-1}, U_{k-1})}{P(m_k^{(j)} | Y_{k-1}, U_{k-1})}. \quad (7) \end{aligned}$$

Bemerkung: Falls $P(m_k^{(j)} | Y_{k-1}, U_{k-1}) > 0$ nicht gilt, wird die Berechnung (7) nicht benötigt.

Prädiktion. Die zeitliche Entwicklung der wertkontinuierlichen Zustände ausgehend von $k-1$ zum nächsten Abtastzeitpunkt k wird durch die a-Priori Dichte

$$\begin{aligned} p(x_k, m_k^{(j)} | Y_{k-1}, U_{k-1}) &= \int_{\mathbb{R}^n} p(x_k, x_{k-1}, m_k^{(j)} | Y_{k-1}, U_{k-1}) dx_{k-1} \\ &= \int_{\mathbb{R}^n} p(x_k | x_{k-1}, m_k^{(j)} | Y_{k-1}, U_{k-1}) \\ &\quad p(x_{k-1}, m_k^{(j)} | Y_{k-1}, U_{k-1}) dx_{k-1} \quad (8) \end{aligned}$$

charakterisiert, welche sich aufgrund der Markov-Eigenschaft zu

$$\begin{aligned} p(x_k, m_k^{(j)} | Y_{k-1}, U_{k-1}) \\ = \int_{\mathbb{R}^n} p(x_k | x_{k-1}, m_k^{(j)}, u_{k-1}) \\ p(x_{k-1}, m_k^{(j)} | Y_{k-1}, U_{k-1}) dx_{k-1} \end{aligned} \quad (9)$$

vereinfacht.

Korrektur. Die a-Posteriori Dichte wird durch Korrektur der a-Priori Dichte gemäß

$$\begin{aligned} p(x_k, m_k^{(j)} | Y_k, U_k) \\ = p(y_k | x_k, m_k^{(j)}, u_k) p(x_k, m_k^{(j)} | Y_{k-1}, U_{k-1}) / c_k \end{aligned} \quad (10)$$

bestimmt, wobei $p(y_k | x_k, m_k^{(j)}, u_k)$ die modespezifische Messwahrscheinlichkeit und c_k die Normalisierungskonstante sind.

2.2 IMMPF-Algorithmus

Der IMMPF beginnt zum Abtastzeitpunkt $k - 1$ mit einer Menge von $q \cdot S$ gewichteten Partikeln $\{x_{k-1}^{(j,l)}, \mu_{k-1}^{(j,l)}; j \in \{1, \dots, q\}, l \in \{1, \dots, S\}\}$, welche die a-Posteriori Dichte

$$p(x_{k-1}, m_{k-1}^{(j)} | Y_{k-1}, U_{k-1}) \approx \sum_{l=1}^S \mu_{k-1}^{(j,l)} \delta(x - x_{k-1}^{(j,l)}) \quad (11)$$

approximieren. Nutzung der in (11) gegebenen Approximation in den einzelnen in Abschnitt 2.1 aufgeführten Dichten, führt zum vollständigen IMMPF-Algorithmus.

Modewechsel. Setzt man (11) in (6) ein, folgt

$$\begin{aligned} P(m_k^{(j)} | Y_{k-1}, U_{k-1}) \\ \approx \tilde{\gamma}_k^{(j)} = \int_{\mathbb{R}^n} \sum_{i=1}^q \left[\pi_{ij} \sum_{l=1}^S \mu_{k-1}^{(i,l)} \delta(x - x_{k-1}^{(i,l)}) \right] dx_{k-1} \\ = \sum_{i=1}^q \sum_{l=1}^S \pi_{ij} \mu_{k-1}^{(i,l)} \end{aligned} \quad (12)$$

für die a-Priori Modewahrscheinlichkeit, welche als Ausgangspunkt für die Bestimmung der a-Priori Partikelgewichte

$$\bar{\mu}_{k-1}^{(j,l)} = \tilde{\gamma}_k^{(j)} / S \quad (13)$$

dient. Setzt man im Weiteren die Approximation (11) in (7) ein, ergibt sich die modespezifische a-Priori Dichte

$$\begin{aligned} p(x_{k-1} | m_k^{(j)}, Y_{k-1}, U_{k-1}) \\ \approx \sum_{i=1}^q \left[\pi_{ij} \sum_{l=1}^S \mu_{k-1}^{(i,l)} \delta(x - x_{k-1}^{(i,l)}) \right] / \tilde{\gamma}_k^{(j)} \\ = \sum_{i=1}^q \sum_{l=1}^S \pi_{ij} \mu_{k-1}^{(i,l)} \delta(x - x_{k-1}^{(i,l)}) / \tilde{\gamma}_k^{(j)}. \end{aligned} \quad (14)$$

Die Approximation der Dichte $p(x_{k-1} | m_k^{(j)}, Y_{k-1}, U_{k-1})$ in (14) erfolgt mit $q \cdot S$ Partikeln, während $p(x_{k-1}, m_{k-1}^{(j)} | Y_{k-1}, U_{k-1})$ in (11) nur durch S Partikel approximiert wird, sodass ein Resampling durchzuführen ist, um die Anzahl der Partikel wieder auf S zu reduzieren. Hierbei werden für jeden Mode S Partikel entsprechend

$$\bar{x}_{k-1}^{(j,l)} \sim \sum_{i=1}^q \sum_{l=1}^S \pi_{ij} \mu_{k-1}^{(i,l)} \delta(x - x_{k-1}^{(i,l)}) / \tilde{\gamma}_k^{(j)} \quad (15)$$

gezogen. Die a-Priori Partikel und Gewichte $\{\bar{x}_{k-1}^{(j,l)}, \bar{\mu}_{k-1}^{(j,l)}\}$ approximieren die a-Priori Dichte des hybriden Zustands nach dem Modewechsel

$$p(x_{k-1}, m_{k-1}^{(j)} | Y_{k-1}, U_{k-1}) \approx \sum_{l=1}^S \delta(x - \bar{x}_{k-1}^{(j,l)}) \bar{\mu}_{k-1}^{(j,l)}. \quad (16)$$

Prädiktion. Die a-Priori Dichte des hybriden Zustands nach der Prädiktion in (8) wird mittels

$$p(x_k, m_k^{(j)} | Y_{k-1}, U_{k-1}) \approx \sum_{l=1}^S \bar{\mu}_{k-1}^{(j,l)} \delta(x - x_k^{(j,l)}) \quad (17)$$

approximiert. Die benötigten Partikel $x_k^{(j,l)}$ lassen sich durch Einsetzen der a-Priori Partikel $\bar{x}_{k-1}^{(j,l)}$ und der entsprechend

$$w_{k-1}^{(j,l)} \sim p(w_{k-1}) \quad (18)$$

gezogenen Rauschpartikel in (1) bestimmen, sodass sich

$$x_k^{(j,l)} = f(m_k^{(j)}, \bar{x}_{k-1}^{(j,l)}, w_{k-1}^{(j,l)}, u_{k-1}) \quad (19)$$

ergibt.

Korrektur. Die Substitution von $p(x_k, m_k^{(j)} | Y_{k-1}, U_{k-1})$ in (10) durch (17) ergibt die a-Posteriori Dichte des hy-

briden Zustands

$$p(x_k, m_k^{(j)} | Y_k, U_k) \approx p(y_k | x_k, m_k^{(j)}, u_k) \sum_{l=1}^S \bar{\mu}_{k-1}^{(j,l)} \delta(x - x_k^{(j,l)}) / c_k. \quad (20)$$

Hierfür bedarf es eines Updates

$$\mu_k^{(j,l)} = \bar{\mu}_{k-1}^{(j,l)} p(y_k | x_k^{(j,l)}, m_k^{(j)}, u_k) / c_k \quad (21)$$

der a-Priori Gewichte. Dazu wird die Messgleichung in (3) für jedes Partikel gelöst und eine Messwahrscheinlichkeit $p(y_k | x_k^{(j,l)}, u_k)$ berechnet, die angibt, wie nahe die berechnete Messung an der realen liegt.

Nach dem Gesetz der totalen Wahrscheinlichkeit ergibt sich aus der a-Posteriori Dichte des hybriden Zustands die a-Posteriori Modewahrscheinlichkeit

$$P(m_k^{(j)} | Y_k, U_k) = \int_{\mathbb{R}^n} p(x_k, m_k^{(j)} | Y_k, U_k) dx_k, \quad (22)$$

sodass man durch Substitution der Dichte des hybriden Zustands in (22) durch (21) die Approximation

$$P(m_k^{(j)} | Y_k, U_k) \approx \gamma_k^{(j)} = \int_{\mathbb{R}^n} \sum_{l=1}^S \mu_k^{(j,l)} \delta(x - x_k^{(j,l)}) dx_k = \sum_{l=1}^S \mu_k^{(j,l)} \quad (23)$$

erhält. Dividiert man im Weiteren (21) durch (23), ergibt sich für $P(m_k^{(j)} | Y_k, U_k) > 0$ die Approximation der modespezifischen a-Posteriori Dichte

$$p(x_k | m_k^{(j)}, Y_k) \approx \sum_{l=1}^S \mu_k^{(j,l)} \delta(x - x_k^{(j,l)}) / \gamma_k^{(j)}. \quad (24)$$

Abschließend lässt sich der approximierte Erwartungswert des wertkontinuierlichen Zustands durch

$$\hat{x}_k = \sum_{i=1}^q \gamma_k^{(i)} \hat{x}_k^{(i)}. \quad (25)$$

mit

$$\hat{x}_k^{(j)} = \sum_{l=1}^S (\mu_k^{(j,l)} x_k^{(j,l)}) / \gamma_k^{(j)} \quad (26)$$

bestimmen.

Der IMMPPF-Algorithmus ist im Folgenden als Pseudocode für einen Zeitschritt von $k-1$ zu k zusammengefasst.

Algorithmus 1 IMMPPF

$\{x_{k-1}^{(j,l)}, \mu_{k-1}^{(j,l)}; m^{(j)} \in \mathcal{M}; j = 1, \dots, q; l \in \{1, \dots, S\}\}$

• Modewechsel:

for $j = 1, \dots, q$ **do**

$$\tilde{\gamma}_k^{(j)} = \sum_{i=1}^q \pi_{ij} \gamma_{k-1}^{(i)}$$

$$\bar{\mu}_{k-1}^{(j,l)} = \tilde{\gamma}_k^{(j)} / S \quad l = 1, \dots, S$$

if $\tilde{\gamma}_k^{(j)} = 0$ **then**

$$\bar{x}_{k-1}^{(j,l)} = x_{k-1}^{(j,l)} \quad l = 1, \dots, S$$

else

$$\bar{x}_{k-1}^{(j,l)} \sim \sum_{i=1}^q \pi_{ij} \sum_{l=1}^S \mu_{k-1}^{(i,l)} / \tilde{\gamma}_k^{(j)} \quad l = 1, \dots, S$$

• Prädiktion:

for $j = 1, \dots, q$ **do**

$$w_{k-1}^{(j,l)} \sim p(w_{k-1}) \quad l = 1, \dots, S$$

$$x_k^{(j,l)} = f(m^{(j)}, \bar{x}_{k-1}^{(j,l)}, w_{k-1}^{(j,l)}, u_{k-1}) \quad l = 1, \dots, S$$

• Korrektur:

for $j = 1, \dots, q$ **do**

$$\mu_k^{(j,l)} = \bar{\mu}_{k-1}^{(j,l)} P(y_k | x_k^{(j,l)}, u_k) / c_k \quad l = 1, \dots, S$$

mit c_k so, dass $\sum_{i=1}^q \sum_{l=1}^S \mu_k^{(i,l)} = 1$

3 Simulationsergebnisse

Beispielhaft wird die hybride Zustandsschätzung mit dem IMMPPF auf ein Drei-Tank-System, wie in Abbildung 1 dargestellt, zur Fehlerdiagnose angewendet. Drei Tanks sind über die Ventile V_{13} und V_{32} miteinander verbunden, während das Ventil V_{2R} zusätzlich Tank 2 mit einem Reservoir verbindet. Aus dem Reservoir wird Wasser über eine Pumpe mit einem konstanten Volumenstrom Q_1 in Tank 1 gefördert.

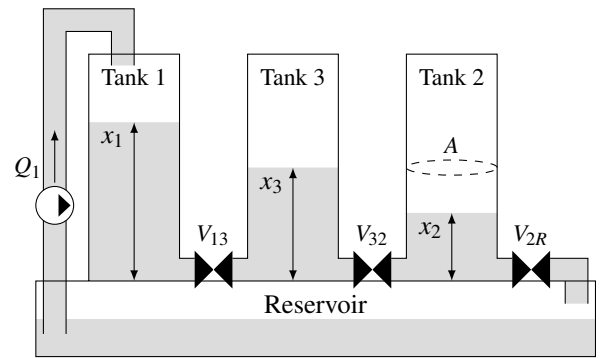


Abb. 1: Skizze des Drei-Tank-Systems

Als Systemzustände werden die Füllstandhöhen der drei Tanks x_1 , x_2 und x_3 gewählt und die Stellgröße ist der Volumenstrom $u = Q_1$. Die Systemdynamik kann

durch die Differentialgleichungen

$$\dot{x}_1 = \frac{1}{A}(u - c_{13} \cdot \text{sign}(x_1 - x_3) \sqrt{2g |x_1 - x_3|}), \quad (27)$$

$$\dot{x}_2 = \frac{1}{A}(c_{32} \cdot \text{sign}(x_3 - x_2) \sqrt{2g |x_3 - x_2|} - c_{2R} \sqrt{2gx_2}), \quad (28)$$

$$\dot{x}_3 = \frac{1}{A}(c_{13} \cdot \text{sign}(x_1 - x_3) \sqrt{2g |x_1 - x_3|} - c_{32} \cdot \text{sign}(x_3 - x_2) \sqrt{2g |x_3 - x_2|}) \quad (29)$$

beschrieben werden. Die für den Algorithmus benötigten zeitdiskreten Zustandsgleichungen

$$x_k = f(x_{k-1}, u_{k-1}) + [\sigma_x \quad \sigma_x \quad \sigma_x]^T w_{k-1} \quad (30)$$

werden durch eine Diskretisierung der Differentialgleichungen (27)-(29) mit dem Runge-Kutta Verfahren vierter Ordnung berechnet. Dabei ist σ_x die Varianz des Prozessrauschens. Die Ausgangsgleichung ist mit

$$y_k = I [x_{1,k} \quad x_{2,k} \quad x_{3,k}]^T + [\sigma_y \quad \sigma_y \quad \sigma_y]^T v_k \quad (31)$$

gegeben, wobei I die $n \times n$ Einheitsmatrix und σ_y die Varianz des Messrauschens ist. Die Zufallsvariablen w_k und v_k werden aus einer Standardnormalverteilung gezogen. Alle Tanks haben die gleiche Querschnittsfläche A und die Ventile werden durch die Parameter c_{13} , c_{32} und c_{2R} beschrieben. Die Systemparameter sind in Tabelle 1 aufgeführt. Es werden $q = 10$ Moden ange-

Beschreibung	Abk.	Wert
Querschnittsfläche	A	$0,0154 \text{ m}^2$
Volumenstrom	Q_1	$1,2371 \cdot 10^{-04} \frac{\text{m}^3}{\text{s}}$
Ventil Parameter V_{13}	c_{13}	$2,5046 \cdot 10^{-05} \text{ m}^2$
Ventil Parameter V_{32}	c_{32}	$2,5046 \cdot 10^{-05} \text{ m}^2$
Ventil Parameter V_{2R}	c_{2R}	$1,9988 \cdot 10^{-05} \text{ m}^2$
Gravitationskonstante	g	$9,81 \frac{\text{m}}{\text{s}^2}$
Prozessrauschen	σ_x	$1 \cdot 10^{-8} \text{ m}$
Messrauschen	σ_y	$5 \cdot 10^{-7} \text{ m}$

Tabelle 1: Systemparameter des Dreitanksystems

nommen, wobei $m^{(1)}$ für das Modell des fehlerfreien Systemverhaltens steht und $m^{(i)}$, $i \in \{2, \dots, q\}$ je einen Sensor-, Aktor- oder Prozessfehler mit jeweils drei Fehleramplituden modellieren. Als Aktorfehler wird ein abweichender Volumenstrom u der Pumpe angenommen, der Prozessfehler ist ein Fehler im Ventil V_{13} und für den Messfehler liegt eine fehlerhafte Messung von y_1 vor. Die Fehlermoden mit den zugehörigen Werten

der angenommenen Fehleramplituden sind in Tabelle 2 aufgelistet.

Aktorfehler	Prozessfehler	Sensorfehler
$m^{(2)}, m^{(3)}, m^{(4)}$	$m^{(5)}, m^{(6)}, m^{(7)}$	$m^{(8)}, m^{(9)}, m^{(10)}$
$u^{(2)} = 1.2u_0$	$c_{13}^{(5)} = 0.7c_{13,0}$	$y_1^{(8)} = 1.2y_{1,0}$
$u^{(3)} = 1.4u_0$	$c_{13}^{(6)} = 0.5c_{13,0}$	$y_1^{(9)} = 1.4y_{1,0}$
$u^{(5)} = 1.6u_0$	$c_{13}^{(7)} = 0.3c_{13,0}$	$y_1^{(10)} = 1.6y_{1,0}$

Tabelle 2: Fehleramplituden

Für jedes modespezifische Partikelfilter werden $S = 1000$ Partikel verwendet. Die Modeübergangswahrscheinlichkeiten werden mit $\pi_{ij} = 0,9875$ für $i = j$ und ansonsten mit $\pi_{ij} = 0,0015$ angenommen.

Im Folgenden werden drei unterschiedliche Fehlerzenarien betrachtet, um das Verhalten des IMMPF im Hinblick auf die Fehleridentifikation zu untersuchen. Die Modeschätzung der drei Szenarien wird in den Abbildungen 2, 4, 6 dargestellt und der RMS Fehler der Zustandsschätzungen ist in den Abbildungen 3, 5, 7 zu sehen.

Szenario 1. Zuerst entsprechen die aufgeprägten Fehleramplituden genau dem Wert, der durch einen der Moden modelliert wird. Die auftretenden Fehler sind

- ein Fehler der Stellgröße u für $50\text{s} < t \leq 200\text{s}$,
- ein Fehler im Ventil V_{13} für $250\text{s} < t \leq 300\text{s}$ und
- ein Messfehler für $450\text{s} < t \leq 600\text{s}$.

Für jeden der drei Fehler ändert sich die Fehleramplitude nach jeweils 50 Sekunden zu dem Wert des nächsten Modes. Der Mode und damit auch die Art des Fehlers und die Fehleramplitude werden richtig geschätzt, jedoch ungefähr eine Sekunde zu spät. Die Systemzustände werden sehr präzise geschätzt. Lediglich zum Zeitpunkt der Modewechsel hat der RMS Fehler einen kleinen Peak.

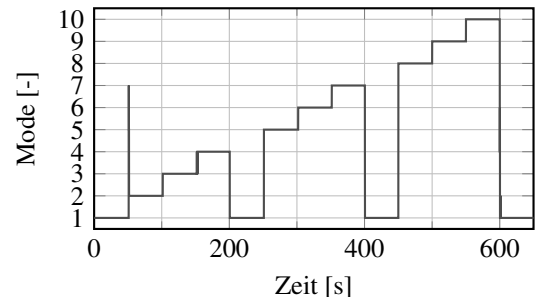


Abb. 2: Modeschätzung Szenario 1

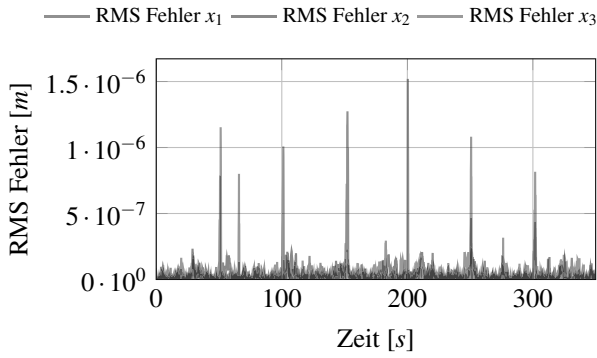


Abb. 3: RMS Fehler der Zustandsschätzung Szenario 1

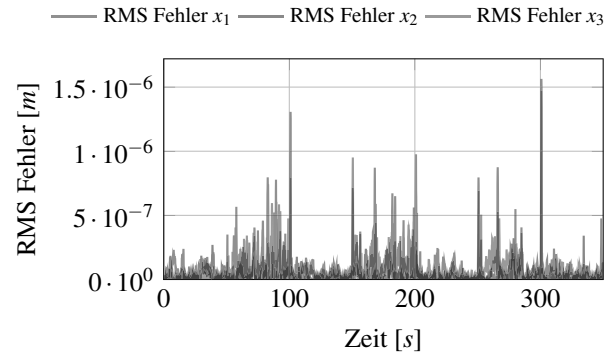


Abb. 5: RMS Fehler der Zustandsschätzung Szenario 2

Szenario 2. Als nächstes liegt die aufgeprägte Fehleramplitude jeweils zwischen zwei modespezifischen Fehlerwerten. Im Folgenden werden nur die Ergebnisse für den Prozessfehler im Ventil V_{13} mit einer Amplitude zwischen den Werten von $m^{(6)}$ und $m^{(7)}$ präsentiert, da das Verhalten für die anderen Fehler vergleichbar ist. Die aufgeprägten Fehleramplituden betragen

- $c_{13} = (c_{13}^{(6)} + c_{13}^{(7)})/2$ für $50s < t \leq 100s$,
- $c_{13} = (c_{13}^{(6)} + 3c_{13}^{(7)})/4$ für $150s < t \leq 200s$,
- $c_{13} = (c_{13}^{(6)} + 7c_{13}^{(7)})/8$ für $250s < t \leq 300s$, sonst
- $c_{13} = c_{13,0}$.

Die Modeschätzung wechselt zwischen den Moden, wo der Fehler liegt. Je näher die Fehleramplitude am Fehlerwert von $m^{(6)}$ liegt, desto öfter wird dieser Mode geschätzt. Die exakte Fehleramplitude kann jedoch nicht aus dem Verhältnis des Modewechsels bestimmt werden. Der RMS Fehler nimmt etwas zu, wenn ein Fehler auftritt, bleibt aber immer noch in der gleichen Größenordnung wie zuvor.

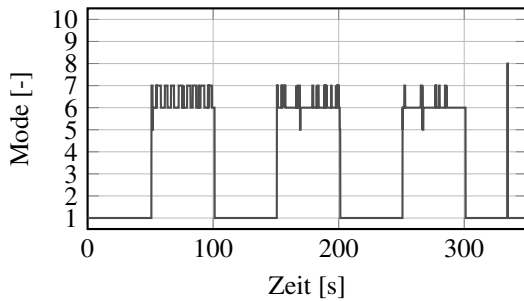


Abb. 4: Modeschätzung Szenario 2

Szenario 3. Im letzten Szenario werden Fehleramplituden aufgeprägt, die größer als die größte modellierte Amplitude sind. Wie zuvor werden nur die Ergebnisse für einen Fehler im Ventil V_{13} dargestellt. Die aufgeprägten Fehleramplituden betragen

- $c_{13} = 5/6c_{13}^{(7)}$ für $50s < t \leq 100s$,
- $c_{13} = 2/3c_{13}^{(7)}$ für $150s < t \leq 200s$,
- $c_{13} = 1/2c_{13}^{(7)}$ für $250s < t \leq 300s$, sonst
- $c_{13} = c_{13,0}$.

Solange die Fehleramplitude nah an den modellierten Werten liegt, kann der Mode immer noch richtig geschätzt werden. Bei einer größeren Amplitude wird die Modeschätzung jedoch immer ungenauer und ein Wechsel in den fehlerfreien Fall wird zu spät erkannt. Der RMS Fehler wird mit steigender Fehleramplitude größer und die Zustandsschätzung kann nicht mehr durchgeführt werden.

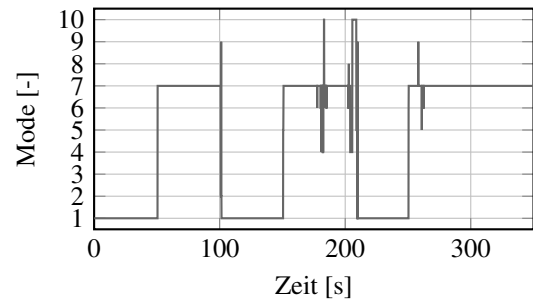


Abb. 6: Modeschätzung Szenario 3

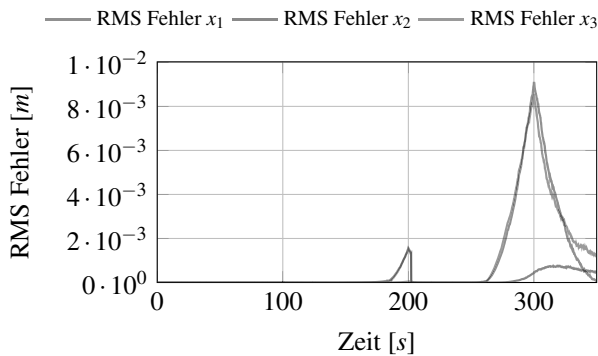


Abb. 7: RMS Fehler der Zustandsschätzung Szenario 3

4 Zusammenfassung

Der IMMPF-Ansatz wird zur Fehleridentifikation bei einem Drei-Tank-System angewandt, wobei die Fehleramplitude jedes betrachteten Fehlers geeignet quantisiert wird. Folglich kommen pro Fehler so viele Moden zum Einsatz, wie es der Anzahl der Quantisierungsstufen entspricht. Es zeigt sich, dass die Fehleramplitude genau geschätzt werden kann, solange der Wert der realen Fehleramplitude einem der in den Moden verwendeten Amplitudenwerte entspricht. Liegt die tatsächliche Fehleramplitude zwischen zwei Werten der modellierten Größen, so kann die Art des Fehlers immer noch genau geschätzt werden, aber nicht die Fehleramplitude. Die Systemzustände werden nach wie vor erfolgreich geschätzt. Ist die reale Fehleramplitude jedoch größer als die größte modellierte Fehleramplitude, kann die reellwertige Zustandsgröße nicht mehr geschätzt werden. Die Art des Fehlers wird für nicht zu große Fehleramplituden noch erkannt, während die Detektion des Modewechsels zu spät bis gar nicht gelingt.

Um einen auftretenden Fehler lokalisieren zu können und um die Systemzustände zu schätzen, ist es daher notwendig, die Moden mit ihren spezifischen Fehleramplituden sinnvoll zu wählen, sodass alle möglichen Fehler in ihrer Art und Amplitude abgedeckt werden. Wenn zudem die Fehleramplitude bestimmt werden soll, kann die Anzahl der Quantisierungsstufen und damit auch die der Moden erhöht werden. Das führt jedoch auch zu einer Erhöhung der Rechenzeit. Ein sinnvollerer Ansatz ist die Modellierung der Fehleramplituden als erweiterte Zustandsvariablen. Dies führt jedoch zu einer hybriden Zustandsschätzung mit heterogenen Zustandsräumen, was für den IMM-Ansatz in [7] und GPB-Ansätze in [11] behandelt wurde, aber im Kontext von Partikelfiltern ein offenes Problem ist.

Literatur

- [1] Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., *Diagnosis and fault-tolerant control*, Springer, 2006.
- [2] Blom, H.A.P., Bar-Shalom, Y., *The interacting multiple model algorithm for systems with Markovian switching coefficients*, IEEE Transactions on Automatic Control, Vol. 33, No. 8, S. 780–783, 1988.
- [3] Blom, H.A.P., Bloem, E.A., *Exact Bayesian and particle filtering of stochastic hybrid systems*, IEEE Transactions on Aerospace and Electronic Systems, Vol. 43, No. 1, S. 55–70, 2007.
- [4] Ding, S.X., *Model-based fault diagnosis techniques*, Springer, 2008.
- [5] Doucet, A., Godsill, S.J., Andrieu, C., *On sequential Monte Carlo sampling methods for Bayesian filtering*, Statistics and Computing, Vol. 10, No. 3, S. 197–208, 2000.
- [6] Driessen, H., Boers, Y., *An efficient particle filter for jump Markov nonlinear systems*, in IEE Target Tracking 2004: Algorithms and Applications, S. 19–22, 2004.
- [7] Granstroem, K., Willet, P., Bar-Shalom, Y., *Systematic approach to imm mixing for unequal dimension states*, IEEE Transactions on Aerospace and Electronic Systems, Vol. 51, No. 5, S. 2975–2986, 2015.
- [8] Hofbaur, M., *Hybrid estimation of complex systems*, Springer, 2006.
- [9] Li, X., Jilkov, V., *Survey of maneuvering target tracking - part v: multiple-model methods*, in IEEE Transactions on Aerospace and Electronic Systems, Vol. 41, No. 4, S. 1255–1320, 2005.
- [10] McGinnity, S., Irwin, G.W., *Multiple model bootstrap filter for maneuvering target tracking*, IEEE Transactions on Aerospace and Electronic Systems, Vol. 36, No. 3, S. 1006–1012, 2000.
- [11] Skeli, E., Weidemann, D., *Multiple-model estimation applied to unequal, heterogeneous state space models*, in Proc. of Int. Conf. on Computer aided System Theory, S. 12–19, 2020.
- [12] Weidemann, D., Alkov, I., *Fault diagnosis of nonlinear differential-algebraic systems using hybrid estimation*, in Variable-Structure Approaches - Analysis, Simulation, Robust Control and Estimation of uncertain dynamic Processes, Rauh, A., Senkel, L. (Editoren), S. 283–307, Springer, 2016.

Simulation Biologie, Medizin und Umwelt

Simulation in Biology, Medicine, and Environment

Ein Fitnesscoach für Menschen mit Querschnittslähmung (L)

K. Nickel, J. Bolz

Wirkstoffidentifikation anhand mikrobieller Proteinmuster (K)

C. Wahmkow, H. Hillenhagen, J. Bernhardt, A. Grüning

Die Flugrouten der Graukraniche – eine GIS-basierte Analyse zum Zusammenhang zwischen Routenwahl und Wetterbedingungen (K)

S. Schreiber, T. Christiansen, J. Wittmann

Modelling Blood Volume Oscillations in the Finger (K)

H. Kastinger, M. Bachler

SIR-type Epidemic Models for Education at Web (K)

A. Freißlinger, A. Körner, F. Breiteneker

Die Kurzbeiträge (K) sind zu finden im **ARGESIM Report 19,**
Proceedings Kurzbeiträge ASIM SST 2022

ISBN ebook 978-3-901608-96-4, DOI: 10.11128/arep.19, ASIM Mitteilung AM 179

Ein Fitnesscoach für Menschen mit Querschnittlähmung

Kilian Nickel¹, Janika Bolz²

¹Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS, Schloss Birlinghoven 1, 53757 Sankt Augustin, Deutschland; kilian.nickel@iais.fraunhofer.de

²Deutsche Sporthochschule Köln, Am Sportpark Müngersdorf 6, 50933 Köln, Deutschland; j.bolz@dshs-koeln.de

Abstract. Fitness-Apps versprechen ein individuelles, flexibles und kostengünstiges Training als Alternative zu klassischen Fitnessangeboten. Es gibt zahlreiche Varianten am Markt, jedoch derzeit keine geeigneten Angebote für Menschen mit Querschnittlähmung. Gerade für diese Zielgruppe ist sportliche Betätigung für die Förderung und den Erhalt der eigenen Mobilität wichtig. Diese Lücke versucht das Projekt ParaGym¹ zu schließen. Es zielt darauf ab, eine App zu entwickeln und im Zusammenspiel mit einem Fitness-Wearable (Sensorshirt) zu demonstrieren. Zwei zentrale Herausforderungen darin bestehen einerseits in der Ermittlung und Umsetzung der zielgruppenspezifischen sportlich-medizinischen Anforderungen und andererseits in der Integration eines Sensor-Wearables zur Steigerung der User Experience. Für letztere stellen Modelle des menschlichen Bewegungsapparates (*musculoskeletal models*) eine Möglichkeit dar, um Körperbewegungen zu analysieren. Dieser Artikel befasst sich mit den bisherigen Erkenntnissen zu diesen beiden Herausforderungen im Rahmen des Projektes.

Einleitung

Fitness-Apps sind eine beliebte Alternative zu klassischen Angeboten wie Fitnessstudios, die vor allem in Pandemie-Zeiten einen Aufschwung erlebten. Ihre Entwicklung geht einher mit dem Trend zum Tracking von Aktivitäten und Vitalwerten des eigenen Körpers mithilfe von Wearables.

Apps und Wearables bieten zahlreiche Ansätze für datengetriebene Services, z.B. individualisierte Trainingsprogramme. Mainstream-Produkte setzen jedoch in der Regel einen vollständig bewegungsfähigen Körper voraus.

Im Rahmen von ParaGym geht es speziell um die Zielgruppe der Querschnittgelähmten, zu der in Deutschland ca. 140.000 Menschen gehören. Für diese Gruppe gibt es keine dedizierten Angebote im Bereich Fitness-Apps. Jedes Jahr kommen ca. 2.350 neue Querschnitte hinzu, die durch Unfälle (55 %) oder Krankheiten (45 %) verursacht werden, bei denen eine Schädigung des Rückenmarks entsteht. Angeborene Querschnitte machen dabei nur einen geringen Anteil aus (weniger als 1 %)² [1].

Menschen, die im Laufe ihres Lebens eine Querschnittlähmung erwerben, durchlaufen eine Rehabilitation, um neue Bewegungsmuster zu erlernen, insbesondere im Zusammenspiel mit dem Rollstuhl, und ihre Mobilität so gut wie möglich zurückzugewinnen. Im Alltag ist langfristige körperliche Fitness entscheidend, um diese Mobilität beizubehalten oder auch zu steigern.

Das Projekt ParaGym basiert auf dem Functional-Fitness-Konzept der Kernwerk GmbH³, die dieses mit der Kernwerk®-App erfolgreich am Markt vertreibt. Diese App ist bislang ausgelegt für Menschen ohne Behinderung und umfasst ein Übungsrepertoire für sämtliche Muskelgruppen. Hierbei kommen verschiedene Geräte zum Einsatz, wie z.B. Lang- und Kurzhanteln, Kettlebells, Kisten, Klimmzugstange oder auch die Umgebung (freie Laufstrecke, stabile Wand). Der oder die Nutzer*in kann das eigene Workout jederzeit individualisieren, sowohl was die Verfügbarkeit der Geräte angeht als auch die bevorzugte Dauer und Tagesform. Grundsätzlich verfolgt das Trainingskonzept das Ziel einer Ganzkörperfitness, bei der alle Muskelgruppen berücksichtigt werden.

Dieses Prinzip soll durch ParaGym so erweitert werden, dass es auch für Menschen mit Querschnittlähmung zugänglich und sinnvoll nutzbar wird. Langfristig sollen

¹ <https://welcome.kernwerk.de/paragym>

² <https://www.der-querschnitt.de/archive/1461>

³ <https://welcome.kernwerk.de/>

sowohl Sportler mit als auch ohne Behinderung gemeinsam nach dem Kernwerk-Konzept trainieren und sich innerhalb derselben Community vernetzen können.

ParaGym wird unter dem Projektnamen FIT-IN3 vom Bundesministerium für Bildung und Forschung (BMBF) gefördert⁴.

1 Anforderungen aus sportlich-medizinischer Sicht

In diesem Abschnitt werden einige zentrale Anforderungen aus sportlich-medizinischer Sicht beschrieben, die sich aus der Projektarbeit ergeben haben. Sie basieren auf eigenen Umfragen und Recherchen [3] sowie auf der Zusammenarbeit mit der Zielgruppe im Rahmen von Workshops.

Je nachdem, an welcher Stelle im Rückenmark der Querschnitt vorliegt, besitzen Menschen eine unterschiedlich ausgeprägte Restfunktion in den Beinen und im Rumpf. Manche Paraplegiker*innen mit hoher Restfunktion sind durchaus in der Lage, kurze Strecken auf zwei Beinen zu gehen, aber sind im Alltag dennoch auf einen Rollstuhl angewiesen. Dies bedeutet, dass das Übungsrepertoire grundsätzlich auch Übungen für die Beinmuskeln enthalten soll. Für Menschen ohne Restfunktion in den Beinen sind diese Übungen jedoch nicht durchführbar und sollten vom Programm gar nicht erst angeboten werden. Die ParaGym-App sieht hierfür eine Reihe von Filtern vor, die bei der Initialisierung abgefragt werden und im Laufe der Benutzung verändert werden können.

Der Rollstuhl ist zentraler Bestandteil der meisten Übungen, wobei er teilweise auch selbst als Sportgerät dient. Beispiele hierfür sind Balancierübungen auf den Antriebsrädern sowie Kombinationen mit dem Theraband, das um verschiedene Partien des Rollstuhls gespannt werden kann. Außerhalb des Rollstuhls besteht die Möglichkeit, Übungen im Liegen durchzuführen. Hierfür ist die Fähigkeit des Transfers zwischen Rollstuhl und Boden bzw. Matte notwendig. Auch hier kann es große Unterschiede bei den Fähigkeiten einzelner Menschen geben, sodass die Transferfähigkeit zum zentralen Filterkriterium wird.

Ein Workout wird durch die App über einen Algorithmus tagesaktuell zusammengestellt und an die eigenen Fähigkeiten angepasst. Hierbei ist darauf zu achten, dass

die Substitution bestimmter Übungen nicht dazu führen darf, dass andere Muskelgruppen übermäßig belastet werden. Darüber hinaus muss beachtet werden, dass ein Workout keine zu häufigen Wechsel zwischen Bodenübungen und Rollstuhlübungen verlangt, da Transfers zwischen Rollstuhl und Boden für viele Trainierende kräftezehrend sein können.

Was die Bedienung der App angeht, so ist der bisherige Modus der, dass der oder die Nutzer*in manuell zur nächsten Übung durch einen Knopfdruck weiterschalten muss. Das bedingt, dass das Smartphone permanent in Reichweite liegen muss. Dies kann gerade beim Training im Rollstuhl und beim Wechsel auf den Boden zum Störfaktor werden. Um dies zu verbessern, sollte die App geeignete akustische Hinweise auf die nächste Übung geben sowie idealerweise eine kontaktlose Möglichkeit zur Navigation zur nächsten Übung anbieten. Hier bietet sich an, dies über eine bestimmte Bewegung oder Geste zu realisieren. Das für ParaGym entwickelte Wearable, ein sensorisches Sport-T-Shirt, kann diese Bewegungen registrieren.

Eine zentrale Anforderung bei der Gestaltung einer jeden Fitness-App besteht darin, die Motivation langfristig aufrecht zu erhalten. Das Ziel ist es, eine dauerhafte Verhaltensänderung herbeizuführen, sodass die Sportübungen zur Routine werden. Dazu gehört es auch, Frustration vorzubeugen. Letztere können leicht entstehen, wenn das geforderte Übungsspensum die körperlichen Fähigkeiten übersteigt. Zu den wirkungsvollsten Methoden aus der Verhaltensforschung zählen unter anderem das Setzen von Zielen und dessen Fortschrittsmonitoring, soziale Unterstützung (z. B. Online-Community, Wettbewerb), Feedback und Coaching [4].

Einige dieser Methoden (Zielsetzung, soziale Faktoren) werden in der Kernwerk®-App bereits umgesetzt. Für die ParaGym-Variante soll ein zusätzlicher Fokus auf Echtzeit-Feedback und virtuelles Coaching gelegt werden, das durch die Wearable-Verknüpfung ermöglicht wird. Die App soll jedoch auch ohne Wearable nutzbar sein.

⁴ <https://www.interaktive-technologien.de/projekte/fit-in3>

2 Sensor-Wearables und Körpermodelle

2.1 Grundlagen

Körpernahe Sensoren bieten die Möglichkeit, verschiedene Vitalparameter aufzuzeichnen: Herzschlag, Atemfrequenz, Körpertemperatur, Blutsauerstoffsättigung oder auch ein Elektrokardiogramm können von kleinen Sensoren gemessen werden. Eine besondere Rolle spielen Bewegungssensoren (Inertial Measurement Units, IMUs). Mit diesen können verschiedene Bewegungsmuster der Sporttreibenden registriert werden und auf vielseitige Weise ausgewertet werden. Die vielleicht verbreitetste Anwendung ist der Schrittzähler. IMUs werden flächendeckend in Smartwatches und Fitness-Armbändern verbaut.

Studien weisen darauf hin, dass das Aktivitätstracking über Wearables eine positive Wirkung auf die Motivation zur sportlichen Betätigung haben kann [5][6]. Für die Art und Weise, die Messwerte in das Trainingserlebnis zu integrieren, gibt es jedoch keine offensichtliche beste Lösung. Ein reines Tracking der gemessenen Vitaldaten ist nur begrenzt sinnvoll, wenn der/die Nutzer*in diese nicht zu interpretieren weiß. Was bedeutet ein Puls von 110 Schlägen pro Minute? Im Falle des Pulses verwendet beispielsweise Fitbit⁵ eine individuelle Einteilung von Herzfrequenzzonen, denen die Funktionen "Fettverbrennung", "Cardio" und "Höchstleistung" zugeordnet werden. Das System teilt mit, wie viele Minuten man sich in welcher Zone befunden hat. Die Vergabe von Punkten für bestimmte Aktivitäten ist ein verbreiteter Belohnungsansatz. Andere Hersteller von Fitness-Wearables verwenden wiederum eigene Bewertungsmechanismen.

2.2 Bewegungssensoren (IMUs)

IMUs messen die lineare Beschleunigung, die Winkelgeschwindigkeit und oftmals auch die Richtung des umgebenden Magnetfeldes in jeweils drei Raumdimensionen. Aus diesen insgesamt neun Signalen kann auf die räumliche Lage des Sensors im zeitlichen Verlauf geschlossen werden. Wenn diese Sensoren an festen Punkten am Körper angebracht sind, kann auf die ausgeführten Bewegungsmuster und Aktivitäten rückgeschlossen

werden. Das Klassifizieren von Aktivitäten ist ein typischer Anwendungsfall für Machine-Learning-Methoden, die z.B. in [7] systematisch verglichen werden. Hierbei können klassische Verfahren wie Logistische Regression oder Multilayer Perceptron verwendet werden [8], als auch tiefe neurale Netzwerke. Letztere enthalten typischerweise eine Kombination aus CNN⁶-Schichten (PerceptionNet [9]) und LSTM⁷-Schichten [10], wodurch in der Regel eine höhere Genauigkeit erzielt werden kann.

Grundsätzlich eignen sich IMU-Messungen für Bewegungen relativ zum körpereigenen Bezugssystem. Für absolute Messungen, z.B. die Berechnung von Schrittlängen, sind sie weniger geeignet, da sich numerische Fehler über die Zeit akkumulieren.

2.3 ParaGym-Sensorshirt

Für ParaGym wurde ein Sensorshirt konzipiert, das zum einen eine Pulsmessung durchführt und zum anderen Bewegungen über fünf am Oberkörper verteilte IMUs erfasst (siehe Abb. 1). Die zentrale IMU sitzt auf der Base Unit, die außerdem den Akku, die Bluetooth-Antenne und integrierte Schaltkreise beherbergt. Die anderen IMUs sind an den Außenseiten der Arme angebracht.

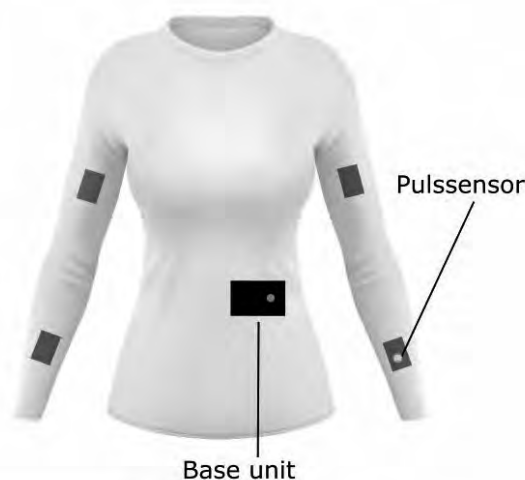


Abb. 1: Skizze des ParaGym-Sensorshirts mit fünf Modulen.

Mithilfe der fünf Bewegungssensoren kann die Bewegung der Arme und des Oberkörpers vermessen werden. Dies kann auf verschiedene Weisen dazu eingesetzt

⁵ Beispielprodukt: Fitbit Charge 4

⁶ Convolutional Neural Network

⁷ Long short-term memory

werden, dem/der Trainierenden ein Echtzeit-Feedback zu Bewegungsqualität und -umfang zu geben.

Die Herangehensweise für die Bewegungsanalyse wird im Folgenden genauer erläutert. Sie beruht darauf, die Bewegungen auf ein digitales Körpermodell zu übertragen. Von diesem Modell aus kann auf die Winkel zwischen beliebigen Gliedmaßen geschlossen werden. Für diesen Zweck bietet sich OpenSim [11] an, ein Open-Source-Paket zur Modellierung und Simulation von Gelenken, Muskeln und Sehnen. OpenSim stellt insbesondere auch eine Schnittstelle zur Interpretation von IMU-Daten bereit: OpenSense [12].

2.4 Orientierungsbestimmung

Bevor diese Tools eingesetzt werden können, muss eine Vorverarbeitung (Pre-processing) der IMU-Daten stattfinden. Jeder Sensor tastet zwar mit einer kontinuierlichen Frequenz ab, jedoch erreichen die einzelnen Messwerte den Empfänger in Form von zeitlich gestauchten Paketen. Diese müssen empfängerseitig wieder „entzerrt“, d.h. in äquidistante Zeitreihen überführt werden. Außerdem enthalten Sensordaten grundsätzlich Rauschen, sodass hier geeignete Filter angewendet werden müssen (z. B. rolling average). Schließlich muss ein Verfahren angewendet werden, um die Orientierung eines Sensors im Raum zu berechnen. Für die Darstellung der Orientierung werden typischerweise Quaternionen q der Länge Eins verwendet. Mit diesen lässt sich der Zusammenhang zur Winkelgeschwindigkeit durch folgende Differenzialgleichung ausdrücken:

$$\frac{dq}{dt} = \frac{1}{2} \omega q.$$

Hier ist $\omega = (0, \omega_x, \omega_y, \omega_z)$ die Quaternion-Darstellung der Winkelgeschwindigkeit $\vec{\omega}$, gemessen im körperfesten Bezugssystem des Sensors. Theoretisch kann $q(t)$ durch Integration gelöst werden, unter Kenntnis der Ausgangsorientierung $q(t = 0)$. In der realen Welt führen jedoch die unperfekten Sensormesswerte sowie numerische Ungenauigkeiten dazu, dass diese Lösung fehleranfällig ist und zu Drifts neigt.

Aus diesem Grund wird häufig die Linearbeschleunigung und auch die Richtung des Magnetfelds in das Lösungsverfahren miteinbezogen (Sensorfusion). Die Verfahren werden komplexer, aber weniger anfällig für Drift. Ein Beispiel ist der so genannte Mahony-Filter [13] [14]. Lösungen für diese Problemstellung werden allgemein als *Attitude and Heading Reference System* (AHRS)

bezeichnet. Das gleichnamige Python-Paket AHRS [15] stellt 17 verschiedene Algorithmen für diesen Zweck zur Verfügung.

2.5 Körpersimulation

Sind die Orientierungen der Sensoren bekannt, so können diese auf ein vorgegebenes Körpermodell übertragen werden. Es gibt eine Reihe von Modellen, die von der OpenSim-Community zur Verfügung gestellt werden. Für unsere Zwecke eignet sich das „Full Body Model“ [16]. Dieses umfasst insgesamt 37 Freiheitsgrade zwischen einzelnen Gliedmaßen. Beschränkt man diese nur auf den Oberkörper, so werden folgende Bewegungsspielräume modelliert:

- Vor-, Rück- und Seitwärtsneigung und Rotation der Wirbelsäule (3),
- Flexion, Adduktion und Rotation der Oberarme (6),
- Flexion und Rotation der Ellenbogen (4),
- Flexion und Deviation der Handgelenke (4).

Die Zahl in Klammern gibt die Freiheitsgrade wieder. Insgesamt wird der Oberkörper durch 17 Freiheitsgrade modelliert. Ein Großteil der Übungen aus dem ParaGym-Übungsrepertoire findet im Sitzen statt und kann durch das Modell abgebildet werden. Die Ausgangsposition ist in Abb. 2 gezeigt.



Abb. 2: Sitzende Ausgangsposition des Full Body Models in OpenSim

Wenn die Platzierung der IMU-Sensoren richtig konfiguriert ist, so kann das Tool die Bewegungen des Modells aus den Sensormessungen ableiten. Der Prozess nennt sich Inverse Kinematik (IK) und basiert auf einer Least-Squares-Minimierung zwischen Modellpose und Orientierungsdaten. Als Output werden die Winkelkoordinaten der einzelnen Gelenke berechnet. Dies ist auch in Echtzeit möglich [17]. Eine Validierung des Outputs ist notwendig und kann direkt grafisch erfolgen, indem die berechneten Winkelkoordinaten auf das Modell angewendet und in der grafischen Oberfläche von OpenSim visualisiert werden. Eine genauere Validierung der Winkel würde erfordern, dass eine Messreihe mit bestimmten Bewegungsabläufen durchgeführt wird, wobei die Testperson einzelne Schlüsselpositionen mit bekannten bzw. gemessenen Winkeln durchläuft. Die berechneten Winkel können dann mit den gemessenen Winkeln in den Schlüsselpositionen verglichen werden.

Die Gelenkwinkel sind der Ausgangspunkt, um den Trainierenden Feedback zu geben. Die meisten Übungen sind wiederholungsbasiert und geben einen bestimmten Bewegungsumfang vor, der sich über Winkel zwischen den Gliedmaßen quantifizieren lässt. Die App könnte also das Erreichen eines Soll-Winkels visualisieren, z. B. in Form einer Balken- oder Tachometer-Darstellung. Dieses kann kombiniert werden mit einer punktebasierten Auswertung über den Übungssatz. So könnte ein abgeschlossener Übungssatz mit einer Punktzahl abhängig von der Ausführung belegt werden. Ebenfalls können Grenzwerte für Fehler, wie Überstreckungen oder zu ruckartige Bewegungen, festgelegt werden und in Echtzeit von der App korrigiert werden. Eine weitere Möglichkeit besteht darin, die Links-Rechts-Symmetrie von bestimmten Übungen zu quantifizieren und die Nutzen auf möglich unsymmetrische Ausführung hinzuweisen.

3 Fazit

Die ParaGym-App verfolgt das Ziel, ein abwechslungsreiches und passgenaues Training für Menschen mit Querschnittslähmung zu entwickeln, das langfristig motiviert und letztendlich mehr Mobilität im Alltag verspricht. Hierbei kommt ein spezielles Sensor-T-Shirt zum Einsatz, um das Training mittels Echtzeit-Feedback interaktiver und motivierender zu gestalten. Dieser Artikel beleuchtet einige der Herausforderungen in der Entwicklungsarbeit.

Das Thema Bewegungsanalyse für Sport-, Rehabilitations- und Forschungszwecke mithilfe von kostengünstigen IMUs erfährt seit einigen Jahren viel Aufmerksamkeit. Mithilfe von Neuronalen Netzwerken können Bewegungen in verschiedenen Einsatzgebieten klassifiziert, gezählt und analysiert werden. Voraussetzung sind entsprechende Trainingsdaten. Der hier vorgestellte Weg über die Inverse-Kinematik-Funktion von OpenSim benötigt keine Trainingsdaten. Er bietet auch eine realistische und flexible Körpermodellierung (andere Lösungen nutzen stark vereinfachte Körpermodelle), die gegebenenfalls für andere sportmedizinische Faktoren (z.B. Erkennung von Überlastungssituationen) genutzt werden können.

Für den Einsatz der App und des Wearables in der Praxis sind allerdings Performance und Benutzerfreundlichkeit entscheidende Kriterien; diese werden in der bevorstehenden Entwicklungsarbeit im Projekt stärker in den Vordergrund rücken.

Bibliographie

- [1] Flieger, R. (2012). *Einführung zur Querschnittslähmung*, in: Haas, U. (Hrsg.): *Pflege von Menschen mit Querschnittslähmung. Probleme, Bedürfnisse, Ressourcen und Interventionen*, Bern
- [2] Hahn, N., Konrad, T. (2022). <https://www.der-querschnitt.de/archive/1461>
- [3] Bolz, Janika (2021). *Konzept zur Entwicklung einer Fitness-App für Menschen mit Paraplegie und zur Steigerung der Gesundheitskompetenz im Rahmen des FIT-IN3-Projekts*. Masterthesis, Deutsche Sporthochschule Köln.
- [4] Sullivan, A. N., Lachman, M. E. (2017). *Behaviour Change with Fitness Technology in Sedentary Adults: A Review of the Evidence for Increasing Physical Activity*. *Front. in Public Health* 4:289. (<https://doi.org/10.3389/fpubh.2016.00289>)
- [5] Brickwood K.J., Watson G., O'Brien J., Williams A.D. (2019). *Consumer-Based Wearable Activity Trackers Increase Physical Activity Participation: Systematic Review and Meta-Analysis*. *JMIR Mhealth Uhealth*. 2019;7(4):e11819. <https://doi.org/10.2196/11819>
- [6] Maher, C., Ryan, J., Ambrosi, C. et al. (2017). *Users' experiences of wearable activity trackers: a cross-sectional study*. *BMC Public Health* 17, 880. <https://doi.org/10.1186/s12889-017-4888-1>
- [7] Abdel-Salam, R., Mostafa, R., & Hadhood, M. (2021). *Human Activity Recognition using Wearable Sensors: Review, Challenges, Evaluation Benchmark*. *ArXiv*, abs/2101.01665.
- [8] Kwapisz, J.R., Weiss, G.M., & Moore, S. (2011). *Activity recognition using cell phone accelerometers*.

SIGKDD Explor., 12, 74-82.

- [9] Kasnesis, P., Patrikakis, C.Z., & Venieris, I.S. (2018). *PerceptionNet: A Deep Convolutional Neural Network for Late Sensor Fusion*. ArXiv, abs/1811.00170.
- [10] Xia, K., Huang, J., Wang, H. (2020). *LSTM-CNN Architecture for Human Activity Recognition*, in IEEE Access, vol. 8, pp. 56855-56866, 2020, doi: 10.1109/ACCESS.2020.2982225.
- [11] Seth, A., Hicks J.L., Uchida, T.K., Habib, A., Dembia, C.L., Dunne, J.J., Ong, C.F., DeMers, M.S., Rajagopal, A., Millard, M., Hamner, S.R., Arnold, E.M., Yong, J.R., Lakshmikanth, S.K., Sherman, M.A., Delp, S.L. (2018). *OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement*. Plos Computational Biology. 2018; 14(7)
- [12] Borno, M.A., O'Day, J., Ibarra, V., Dunne J., Seth, A., Habib, A., Ong, C., Hicks, J., Uhlich, S., Delp, S. (2021). *OpenSense: An open-source toolbox for Inertial-Measurement-Unit-based measurement of lower extremity kinematics over long durations*. bioRxiv 2021.07.01.450788; <https://doi.org/10.1101/2021.07.01.450788>
- [13] Mahony, R., Hamel, T., Pflimlin, J.-M. (2008). *Nonlinear complementary filters on the special orthogonal group*. Automatic Control, IEEE Transactions on, 53(5):1203 –1218, june 2008.
- [14] Euston, M., Coote, P., Mahony, R., Kim, J., Hamel, T. (2008). *A Complementary Filter for Attitude Estimation of a Fixed-Wing UAV*. 340 - 345. 10.1109/IROS.2008.4650766.
- [15] Mario Garcia, 2022. <https://pypi.org/project/AHRS/>
- [16] Rajagopal, A., Dembia, C.L., DeMers, M.S., Delp, D.D., Hicks, J.L., and Delp, S.L. (2016). "Full-Body Musculoskeletal Model for Muscle-Driven Simulation of Human Gait." IEEE Transactions on Biomedical Engineering 63.10 (2016): 2068-2079.
- [17] Slade, P., Habib, A., Hicks, J., Delp, S. (2021). *An Open-Source and Wearable System for Measuring 3D Human Motion in Real-Time*. IEEE Transactions on Biomedical Engineering, PP. 1-1. 10.1109/TBME.2021.3103201.

Simulationsmethoden

Methods in Simulation

A time domain approach for system identification using hill climbing (L)
R. Büchi

Geometrisches Mehrgitterverfahren für die gyrokinetische Poisson-Gleichung mit
Anwendung auf Kernfusion (K)
C. Schwarz, P. Leleux, M. J. Kühn, U. Rüde

Using the Proxel Method to build EHMM for behaviour reconstruction (L)
P. Krenckel, C. Krull

An Extension for the Specification and Automated Selection of System Variants Based
on the System Entity Structure Using a Problem from Process Industry (L)
H. Folkerts, T. Pawletta, U. Durak

Hestia jl: A Julia library for heat conduction modeling with boundary actuation (K)
S. Scholz, L. Berger

Analysis of Selected Literature on Co-simulation (L)
I. Hafner, S. Emrich, N. Popper

Coupling of Simulation Tools for Obtaining Local Fatigue in Combination with
Experimental Data (K)
N. Adamchuk, B. Rösch, W. Schlüter, F.-C. Reissner, J. Baumgartner

Hochaufgelöste Energieprofile durch hybride Simulation (K)
B. Woerrlein, S. Strassburger

Modular platform for route guidance in the cyber-physical laboratory test field (L)
M. Göllner, S. Jacobitz, T. Li, X. Liu-Henke

Untersuchung der anwendungsspezifischen Verifikations- und Validierungstechniken
unter Berücksichtigung einer veränderten Datenbasis (K)
K. Langenbach, A. A. Scheidler, M. Rabe

Service-oriented Architecture for Automatic Evaluation of Urban Logistical Concepts (L)
R. Pump, L. Bienzeisler, T. Lelke, V. Ahlers, A. Koschel

Die Kurzbeiträge (K) sind zu finden im **ARGESIM Report 19,**
Proceedings Kurzbeiträge ASIM SST 2022

ISBN ebook 978-3-901608-96-4, DOI: 10.11128/arep.19, ASIM Mitteilung AM 179

A time domain approach for system identification using hill climbing

Roland Büchi^{1*}

¹Zurich University of Applied Sciences, School of Engineering, Technikumstrasse 9, CH 8401 Winterthur, Switzerland; *roland.buechi@zhaw.ch

Abstract. A large number of methods are known for system identification, which are used both in the time domain and in the frequency domain. In particular, genetic algorithms are increasingly being used today in order to determine the parameters of a model on the basis of measurements. In this article, the related method 'hill climbing' is used together with the least square criterion in order to correctly identify models of small order on the basis of measured step responses in the time domain. It is shown that the algorithm converges well for many starting values and that this method can be applied very well and efficiently for the topic of system identification.

Introduction

System identification is a field of engineering that is particularly important for parameter identification and modeling. The identification in the time domain has been known in particular since the 20th century, for example in [1]. In the following years and in the course of the rapidly increasing computing power, identification methods with genetic algorithms or particle swarm optimization, PSO became known. In this paper, the 'hill climbing' method [2], [3] is used. This is also a stochastic method for the use in optimizing controllers, and it is related to PSO. The most important difference is that here, the particles are processed independently to each other. This shows a good probability for finding smaller local minima. Since, in applications in mechanical engineering, calculations are often not made in the frequency domain but in the time domain [4], this approach is used for the parameter identification of such a problem.

Figure 1 shows a torsional oscillator. A spring is clamped on one side and connected to a rotatably mounted disc on the other side. The system is fraught with friction. First of all, it should be assumed that there is no model of it and that one only wants to identify it on the basis of a measurement in the time domain. The system is equipped

with an encoder with 2000 steps per revolution. For the measurement, the disk is now turned one turn and then released. The resulting curve in the time domain is shown in figure 2.

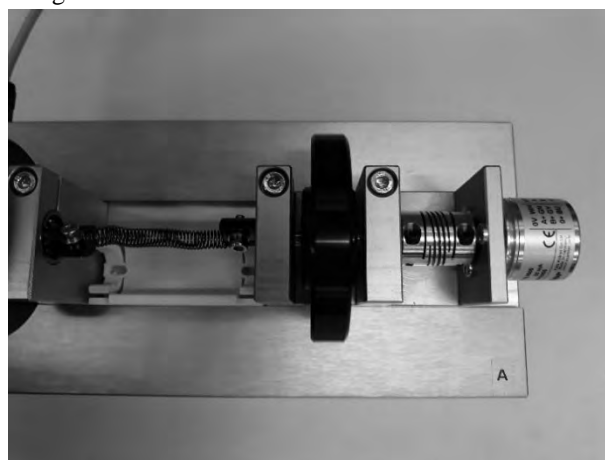


Figure 1: Torsional oscillator with spring, disk and encoder

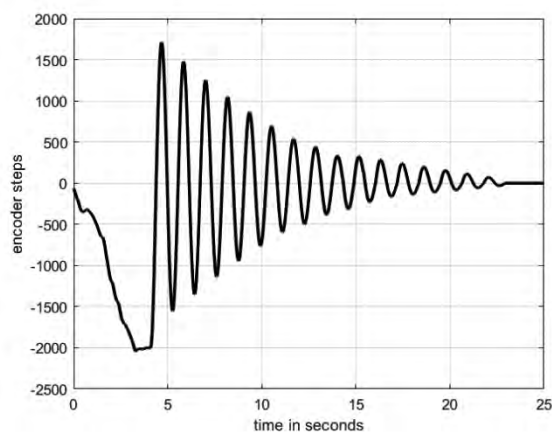


Figure 2: Torsional oscillator, measurement of step response

1 Identification

One now wants to determine a transfer function of the system $G(s) = \frac{\varphi(s)}{M(s)}$ using the criterion of the Least Square [5]. The output is the angle φ and the input is the torque M . The system has a torque $-M_0$ as an initial condition due to the rotation, and when it is released the torque changes suddenly to 0. This affects the system in this way as if there were a permanent change in the torque from 0 to $+M_0$ at time $t = t_0$.

In a general case, the system can be represented as a transfer function with the Laplace operator 's' as shown in formula 1. First of all, it is assumed that the system model is not known and that all parameters a_n and b_n are different from zero. A 3rd order system is assumed here at the beginning:

$$G(s) = \frac{b_3 \cdot s^3 + b_2 \cdot s^2 + b_1 \cdot s + b_0}{a_3 \cdot s^3 + a_2 \cdot s^2 + a_1 \cdot s + a_0} \quad (1)$$

This transfer function is the most compact form of system representation. Its step response can now be evaluated in the time domain using Matlab. The evaluation of the transfer function in the time domain with Matlab is also the basis of the following calculations and considerations. In the specific example it is of course clear that the transfer function to be identified in the time domain is of a smaller order, but it is hoped that the corresponding higher order coefficients in the following calculation are so small that the order can be reduced. In the following, it is here still assumed, that there is no physical model of the system. In principle, it would be possible to calculate the step responses of the transfer functions for all parameter combinations a_0 to a_3 and b_0 to b_3 and to compare them with the measurement using least square. However, if you consider that you have to calculate the parameters as example from 0 to 10 in steps of as example $1E-4$, the result is a huge number of calculations. If the parameter a_0 is normalized to 1, then the seven remaining parameters span an seven-dimensional space and with this number of steps there would be $1E5 * 1E7 = 1E12$ calculations. The number theory results in a polynomial computational effort with nested loops, but this large number would be far too computationally intensive even with the fastest today's computers. Remember that for each calculation step, the least square criterion would have to be

calculated and added up for each measurement and simulation value. The approach with the conventional method, with the calculation with all possible combinations, does not work here.

2 Hill Climbing algorithm with Least Square

The hill climbing algorithm is a method that has been known for a long time. Today it is assigned to the methods of artificial intelligence. In recent years, there were also similar methods used for parameter optimization in system identification and control theory [6] – [12]. The idea behind this hill climbing algorithm is simple: With each new calculation step, the smallest parameter change, multiplied by a random sign (+/- 1) or 0, is added to each of the parameters to be changed. Then the new least square is calculated and compared with the previous one. If the new least square is smaller, the new values are retained and assumed as new. If the new least square is larger, the old values are used again as the new. The hill climbing algorithm always works with a heuristic function that changes the parameters. The heuristic function for the shown problem is:

$$f_h = p + \text{randomsign} \cdot \Delta p \quad (2)$$

Where $p = [b_3, b_2, b_1, b_0, a_3, a_2]$ (there is $a_1 = 1$)

The hill climbing method actually only finds local minima, in this case local minima of the least square value. It turns out, however, that with different starting values after many calculations, always the same or very similar parameters are found. This suggests that there are essentially only monotonically falling or rising least square values. This fact would have to be examined more closely in further steps.

The convergence of hill climbing with the search for the least square is also relatively good. For small parameter changes, the author only carries out up to a few million calculation steps, depending on the simulation. It would be possible to define as a termination criterion when the values of the least square no longer change significantly or no longer change at all. Here, too, further calculations are likely to be carried out. However, in practice the parameters found are quite good.

3 Results

In a first step, the parameters are calculated for a higher order than assumed; in this example, with the boundary conditions for a parameter change of $1\text{E-}4$ per calculation step, the best result for the 3rd order system after 10 simulations with randomized starting values and 3'000'000 learning iterations each is:

$$G(s) = \frac{0.0018 \cdot s^3 + 0.0434 \cdot s^2 + 5.884 \cdot s + 6.32}{0.0322 \cdot s^3 + 0.0454 \cdot s^2 + 0.9457 \cdot s + 1} \quad (3)$$

In many representations of the transfer function, the parameter of the highest denominator order, here a_3 , is normalized to 1. When calculating the parameters with the hill climbing method, however, the author advises normalizing the parameter a_0 to 1, since this exists in all systems with a stationary end value. In systems with a free integrator without a stationary end value, this parameter is zero. Otherwise, it would make sense to normalize the parameter a_1 to 1.

The convergence of the parameters was also calculated over the iterations. It always behaves similarly over the 10 simulations carried out with different starting values. Figure 3 shows the course. Since all parameters are changed simultaneously using the hill climbing method, the method converges slowly.

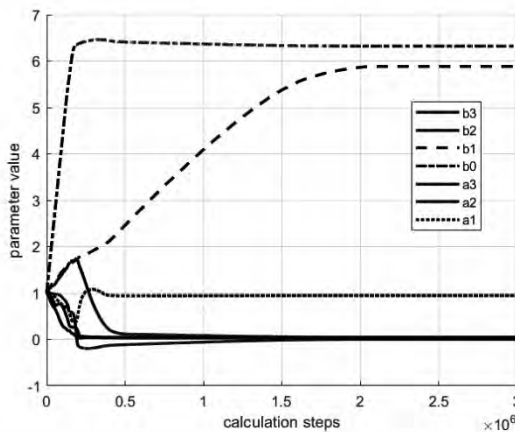


Figure 3: parameter convergence for the transfer function of formula 3

3.1 Reduction of system order

Now it turns out that the parameters for the higher orders a_3 and a_2 in the denominator and b_3 and b_2 in the numerator are much smaller than the other parameters. Another issue is also the reduction of order. It is also a field of science in itself. The literature on it is immense and there is only a very brief introduction given here. The transfer function from formula 3 can also be represented in the pole-zero notation [1]. With this notation according to the pole-zero notation, $G(s)$ can also be represented as in formula 4.

$$G(s) = k \cdot \frac{(s - \beta_3) \cdot (s - \beta_2) \cdot (s - \beta_1)}{(s - \alpha_3) \cdot (s - \alpha_2) \cdot (s - \alpha_1)} \quad (4)$$

For the system order reduction you can now also search for dominant zeros and poles. These are those that are sufficiently close to 0. The non-dominant poles and zeros are those that are sufficiently far away from 0; these can be neglected if necessary. One also has to take into account the sampling time of the systems, because the resulting aliasing also affects the system behavior.

In this example, a zero and a pole are almost identical. One solution of the numerator polynomial is $s = -1.0824$ and one of the denominator polynomial is $s = -1.0707$. Such solutions result in identical local minimums of the least square. One can cancel the zeroes. In this case there is an order reduction of the numerator and the denominator by one order each. Another calculation is now carried out in such a way that the corresponding coefficients of the third order, b_3 and a_3 , are set to zero, taking into account that order reduction. The best result after 10 simulations with randomized starting values and 1'500'000 learning iterations each is as follows.

$$G(s) = \frac{0.0033 \cdot s^2 + 0.0493 \cdot s + 6.317}{0.0346 \cdot s^2 + 0.0117 \cdot s + 1} \quad (5)$$

Here, too, it must be discussed whether a further reduction in order can be carried out. The two parameters b_2 and b_1 are small compared to b_0 . The modeling of the system that is available in this special case can also be used for discussion. This is shown below.

Assumed that all of this has been taken into account and one came to the conclusion that a_3 , b_3 , b_2 and b_1 can actually be neglected, one can do the calculation again. With the parameter change of 0.0001, the parameters of formula 6 result after 10 Simulations with 500'000 calculation steps. The number of calculation steps decreases, since there are less parameters to change. This results in a faster convergence.

$$G(s) = \frac{6.315}{0.0343 \cdot s^2 + 0.0119 \cdot s + 1} \quad (6)$$

Here, too, the parameters converge in all simulations. Since fewer different parameters are now changed via the hill climbing method, the convergence is much faster than with the formula 3 with more changed parameters.

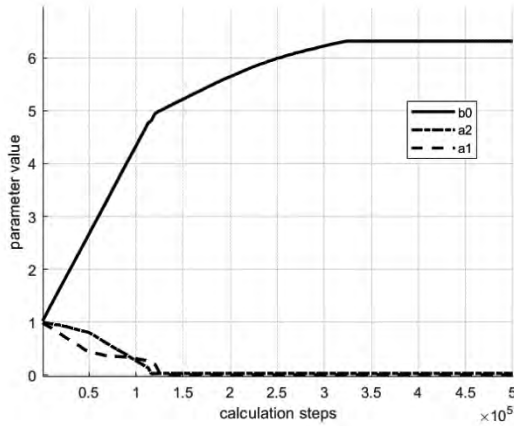


Figure 4: parameter convergence for the transfer function of formula 6

The comparison of simulation and measurement is shown in figure 5.

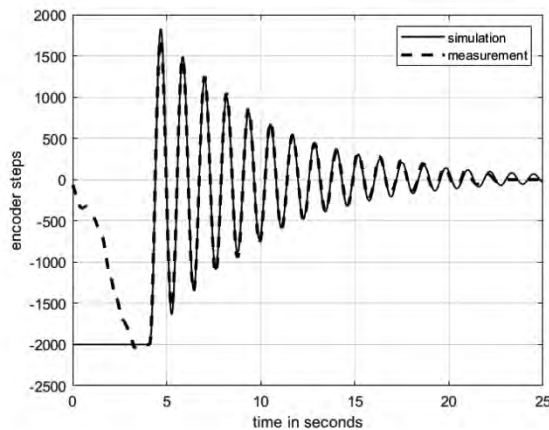


Figure 5: Comparison of simulation and measurement for the torsional oscillator

3.2 Reduction of system order using the physical model

The method presented here works in the general case without any knowledge of the model. This is how it was carried out up to here, up to the model reduction due to small parameter values or the shortening of poles and zeros. However, if a physical model can be derived, one can reduce the model to the relevant parameters right from the start. According to formula 7:

$$J \cdot \ddot{\varphi} = \sum M = M_0 - b \cdot \dot{\varphi} - c \cdot \varphi \quad (7)$$

Thus, the differential equation also results to:

$$J \cdot \ddot{\varphi} + b \cdot \dot{\varphi} + c \cdot \varphi = M_0 \quad (8)$$

And after the laplace transformation, the transfer function of formula 9 results:

$$\frac{\varphi(s)}{M(s)} = \frac{1}{J \cdot s^2 + b \cdot s + c} = \frac{1/c}{J/c \cdot s^2 + b/c \cdot s + 1} \quad (9)$$

This shows that the parameter or order reduction of formula 6 is correct.

If you take a closer look at the measured step response according to figure 2, it is also noticeable that the step response of the simulation no longer corresponds very well with the measurement, especially after a long time. This is due to the non-linear effect of static friction, which is responsible for the fact that the shaft simply stops after falling below a certain low speed. This small non-linearity is not taken into account in the model.

4 Discussion and outlook

It turns out that this is a very applicable and transparently explainable method to carry out the parameter identification in the time domain. The convergence is quite good in the discussed examples, although the orders are not very large. With many systems that occur in practice, however, measurements of sub-systems can also be carried out, which then often have an order that is small

enough that this method can be used. Often there are also several measurements with slightly different step responses. This can be caused by small measurement inaccuracies, for example. This can also be dealt with using this method. The least square method is then simply applied to the individual comparisons with the measurements, one after the other. In this case the least square results will be added up and compared with older added up least square results.

So that the reader can test out examples himself, a Matlab code of a simple example is also printed at the end. This can be copied into an editor. In order to simulate an existing measurement, a transfer function was set at the beginning and its step response was subsequently assumed as a measurement. The hill climbing method also converges well here and delivers the identified parameters as discussed above.

```
clear
ls = 1E20;
% didactic example with given Gmeas which
% replaces Measurement
Gmeas = tf([2 2],[3 2 1])
delta_t = 0.01; %% sample time
t = 0.01:delta_t:50; %% time vector
[xmeas,tmeas] = step(Gmeas,t);

b2 = 1; delta_b2 = 0.01;
%% hill climbing parameter
b1 = 1; delta_b1 = 0.01;
%% hill climbing parameter
b0 = 1; delta_b0 = 0.01;
%% hill climbing parameter
a2 = 1; delta_a2 = 0.01;
%% hill climbing parameter
a1 = 1; delta_a1 = 0.01;
%% hill climbing parameter
a0 = 1; delta_a0 = 0.0;
%% hill climbing parameter

for counter = 1:10000;
counter
sign_b2 = fix(5*rand(1))-2;
% [-2...2] heuristic function
b2 = b2 + sign_b2*delta_b2;
sign_b1 = fix(5*rand(1))-2; % [-2...2]
b1 = b1 + sign_b1*delta_b1;
sign_b0 = fix(5*rand(1))-2; % [-2...2]
b0 = b0 + sign_b0*delta_b0;
sign_a2 = fix(5*rand(1))-2; % [-2...2]
a2 = a2 + sign_a2*delta_a2;
sign_a1 = fix(5*rand(1))-2; % [-2...2]
a1 = a1 + sign_a1*delta_a1;
% a0 is kept as 1

G = tf([b2 b1 b0],[a2 a1 a0]);
```

```
[x,t1] = step(G,t);

lsnew = leastsquare(x,xmeas,delta_t);
if (lsnew < ls)
% Backpropagation
ls = lsnew
else
b2 = b2 - sign_b2*delta_b2;
b1 = b1 - sign_b1*delta_b1;
b0 = b0 - sign_b0*delta_b0;
a2 = a2 - sign_a2*delta_a2;
a1 = a1 - sign_a1*delta_a1;
end
end %counter

plot(t,x,tmeas,xmeas)

ls = leastsquare(x,xmeas,delta_t);

function y = leastsquare(a,b,deltat)
y = 0;
for k = 1:length(a);
y = y + (a(k)-b(k))^2*deltat;
end
end
```

References

- [1] Unbehauen, Heinz. Regelungstechnik. Braunschweig: Vieweg, 1992.
- [2] Skalak, David B. "Prototype and feature selection by sampling and random mutation hill climbing algorithms." Machine Learning Proceedings 1994. Morgan Kaufmann, 1994. 293-301.
- [3] Büchi, Roland. "Optimal ITAE Criterion PID Parameters for PTn Plants Found with a Machine Learning Approach." 2021 9th International Conference on Control, Mechatronics and Automation (ICCA). IEEE, 2021
- [4] Chatzis, Manolis N., Eleni N. Chatzi, and Andrew W. Smyth. "An experimental validation of time domain system identification methods with fusion of heterogeneous data." Earthquake Engineering & Structural Dynamics 44.4 (2015): 523-547.
- [5] Verhaegen, Michel, and Vincent Verdult. Filtering and system identification: a least squares approach. Cambridge university press, 2007.
- [6] Liu, Xiaoyong, Huajing Fang, and Zhaoxu Chen. "A novel cost function based on decomposing least-square support vector machine for Takagi-Sugeno fuzzy system identification." IET Control Theory & Applications 8.5 (2014): 338-347.
- [7] Wang, Grace S. "Application of hybrid genetic algorithm to system identification." Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures 16.2 (2009): 125-153.

- [8] Orszulik, Ryan R., and Jinjun Shan. "Active vibration control using genetic algorithm-based system identification and positive position feedback." *Smart Materials and Structures* 21.5 (2012): 055002.
- [9] Alfi, Alireza, and Hamidreza Modares. "System identification and control using adaptive particle swarm optimization." *Applied Mathematical Modelling* 35.3 (2011): 1210-1221.
- [10] Bian, Qi, et al. "System identification method for small unmanned helicopter based on improved particle swarm optimization." *Journal of Bionic Engineering* 13.3 (2016): 504-514
- [11] Büchi, Roland. *State space control, LQR and observer: step by step introduction with Matlab examples*. Norderstedt Books on Demand, 2010.
- [12] Razmjoo, Navid, and Mehdi Ramezani. "Training wavelet neural networks using hybrid particle swarm optimization and gravitational search algorithm for system identification." *International Journal of Mechatronics, Electrical and Computer Technology* 6.21 (2016): 2987-2997.

Using the Proxel Method to build EHMM for behaviour reconstruction

Pascal Krenckel*, Claudia Krull

Institut für Simulation und Graphik, Otto-von-Guericke-Universität Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Deutschland; *krenckel@ovgu.de

Abstract.

Die Proxel-Methode wurde an der Otto-von-Guericke-Universität Magdeburg entwickelt, und wird dazu verwendet partiell-beobachtbare Systeme mit diskretem Zustandsraum zu analysieren und das Systemverhalten zu rekonstruieren. Dabei können nur die Forward-Wahrscheinlichkeiten berechnet werden. In diesem Paper stellen wir ein Verfahren vor, dass mithilfe der Proxel-Methode ein Expanded-Hidden-Markov-Model erzeugt. So können nicht nur die Forward-Wahrscheinlichkeiten, sondern auch die Backward-Wahrscheinlichkeiten berechnet werden. Dies ermöglicht bessere Zustandsrekonstruktionen. Dabei konzentrieren wir uns nur auf das Decoding als Problemstellung. Unsere Experimente zeigen, dass diese Methode in Genauigkeit mit der Proxel-Methode mithalten kann.

Einführung

Virtuelle Stochastische Sensoren (VSS) können für verschiedene Probleme angewendet werden. So können sie zum Beispiel zur Aktivitäts-Erkennung im Bereich des Ambient Assisted Living verwendet werden [1]. Auch zur Gestenerkennung von 3D- oder Touch-Gesten können Virtuelle Stochastische Sensoren verwendet werden. [2, 3, 4] Die Proxel-Methode kann zur Verhaltensrekonstruktion partiell-beobachtbarer stochastischer Systeme verwendet werden, insbesondere, wenn diese nicht die Markov-Eigenschaft erfüllen. Dabei kann diese Methode nur die Forward-Wahrscheinlichkeiten berechnen. Doch auch die Backward-Wahrscheinlichkeiten können von Interesse sein. So kann z.B. mithilfe der Forward- und Backward-Wahrscheinlichkeiten berechnet werden, wie wahrscheinlich ein Zustand s zum Zeitpunkt t bei einer Beobachtungssequenz T ist. Dabei kann $|T|$ auch größer als t sein. Dies ist notwendig, wenn ein bestimmter früherer Systemzustand rekonstruiert

werden soll. Wir präsentieren hier eine Methode auf Basis der Proxel-Methode und der Expanded-Hidden-Markov-Modellen, die nicht nur die Forward-Wahrscheinlichkeiten, sondern auch die Backward-Wahrscheinlichkeiten berechnen kann.

1 Hintergrund

1.1 Virtuelle Stochastische Sensoren

VSS wurden 2011 von Claudia Krull eingeführt. [5] Ein VSS modelliert einen doppelt stochastischen Prozess, bei dem nicht nur der interne Systemzustand von einem stochastischen Prozess abhängt, sondern auch die Ausgabe der „physischen Sensoren“ kann von einem stochastischen Prozess abhängen. Die häufig verwendeten Hidden-Markov-Modelle sind auch Virtuelle Stochastische Sensoren. Es können aber auch Modelltypen mit gedächtnisbehaftetem Verhalten wie Augmented Stochastic Petri Nets verwendet werden. Ein VSS besteht immer aus einem Modell und einem Lösungsverfahren. Dabei gibt es drei Zielstellungen: Evaluierung, Decoding und Training. [5, 6]

Das Evaluierungsproblem versucht die Frage zu beantworten, wie wahrscheinlich bestimmte Ereignisse sind - z.B. eine gegebene Beobachtungssequenz. Dadurch kann entschieden werden, welches System wahrscheinlich eine bestimmte Beobachtungssequenz erzeugt hat. Das Ziel des Decodings ist es, bei einer gegebenen Sequenz von Beobachtungen (Trace/Spur) die Sequenz der internen Zustände (Path/Pfad) zu ermitteln. Dabei gibt es verschiedene Herangehensweisen. Die drei meist genutzten sind:

1. Der wahrscheinlichste Pfad
2. Die Sequenz der wahrscheinlichsten Zustände
3. Die Sequenz der wahrscheinlichsten Zustände, die immer noch ein Pfad ist.

Bei Hidden-Markov-Modellen heißen die entsprechenden Algorithmen die diese Sequenzen berechnen: Viterbi, Posterior und Posterior-Viterbi. Beim Training wird versucht das Parameterset eines Modells zu finden, das die höchste Wahrscheinlichkeit hat eine gegebene Menge von Beobachtungen zu erzeugen.

Für Modelle mit diskreten Zustandsräumen existieren bereits Lösungsverfahren die das Evaluierungsproblem und das Decoding-Problem lösen können. Die Proxel-Methode von Graham Horton (2002) und deren Erweiterungen ermöglichen das Lösen, ohne dass Differenzialgleichungen gelöst werden müssen. Dafür wird die Zeitdimension diskretisiert und in Zeitschritten gerechnet. [7, 8, 9, 10]

1.2 Proxel-Methode

Die Proxel-Methode dient zur Analyse von stochastischen Systemen mit diskreten Zuständen, wie Petri-Netze, Warteschlangensysteme oder Hidden-non-Markovian-Modellen auf Basis der zusätzlichen Variablen (supplementary variables) [7, 11]. Dabei werden die kontinuierlichen Systeme diskretisiert und der Zustandsraum exploriert. Die Systeme müssen nicht die Markov-Eigenschaft erfüllen.

Ein Proxel ist ein Tupel, das den Systemzustand und die Wahrscheinlichkeit speichert. Für jedes Proxel können die nachfolgenden Proxel berechnet werden. Dazu wird die Wahrscheinlichkeit berechnet, dass das System aus dem Zustand in den neuen Zustand innerhalb eines Zeitschrittes wechselt.

Ein interner Systemzustand stellt dabei einen Zustand des Systems dar. Dieser muss nicht beobachtbar sein, aber er ist diskret beschreibbar. Bei einem Warteschlangenproblem könnten dies z.B. die Anzahl der Personen in der Warteschlange sein. Liegt das Model in Form eines Petri-Netzes vor, so stellt das Marking einen internen Systemzustand dar.

Ein Erweiterter Systemzustand enthält dabei noch einen τ -Vektor. Dieser enthält für alle Transitionen, wie lange sie aktiv waren, ohne dass sie ausgelöst haben. Dies ist notwendig, um die Wahrscheinlichkeit zu berechnen, dass eine Transition in einem bestimmten Zeitschritt feuert.

Der erweiterte Zustand eines Proxels, kann dabei noch weitere Informationen enthalten, die zur Analyse

benötigt werden. In diesem Paper ist dies z.B. eine Pfad-ID, mit der der Pfad rekonstruiert werden kann. Ein Pfad ist dabei eine Sequenz von internen Zuständen.

Bisher ist es nur möglich vorwärts zu rechnen, und so von einem Eltern-Proxel die Kinderproxel zu bestimmen. Das entspricht dem Forward-Algorithmus. Für das Rückwärtsrechnen (Backward-Algorithmus) sind die erweiterten Endzustände nicht bekannt, da die Proxel-Methode diese on-the-fly berechnet. Auch das Rekonstruieren der Elternproxel aus den Kinderproxel ist nicht ohne weiteres möglich, da der ursprüngliche τ -Wert, wenn eine Transition gefeuert hat, nicht bekannt ist. Für das Decoding bedeutet dies, dass nur der Viterbi-Algorithmus (Wahrscheinlichste Pfad) und nicht der Posterior (Sequenz wahrscheinlichster Zustände) oder der Posterior-Viterbi-Algorithmus (Pfad wahrscheinlichster Zustände) verwendet werden kann. [8, 9, 7]

Bei Expanded-Hidden-Markov-Modellen existieren hingegen bereits Algorithmen zum Berechnen der Forward- und Backward-Wahrscheinlichkeiten.

1.3 Expanded-Hidden-Markov-Modell

2007 passte Claudia Krull die Hidden-Markov-Modelle an. [12] Im Gegensatz zu den HMM werden die Ausgaben bei Expanded-Hidden-Markov-Modellen (EHMM) nicht in den Systemzuständen, sondern in den Übergängen erzeugt. Da die resultierenden Berechnungen identisch sind, sind beide Modelle ineinander umwandelbar. Die von uns untersuchten Modelle erzeugen ihre Ausgaben in den Übergängen. Daher ist ein EHMM wesentlich kompakter als das entsprechende HMM. Mit EHMMs lässt sich sowohl der Viterbi-Pfad als auch der Posterior- und Posterior-Viterbi-Pfad berechnen, da diese die Forward- und die Backward-Wahrscheinlichkeiten berechnen können. [12]

Das Berechnen der Backward-Wahrscheinlichkeiten ermöglicht weitere Anwendungsfälle, wie die Rekonstruktion wahrscheinlicher Zustände. Dies ist für den Posterior und den Posterior-Viterbi Algorithmus notwendig.

Insbesondere zeigen frühere Arbeiten, dass für das Decoding-Problem je nach Modell der Posterior bzw. der Posterior-Viterbi-Algorithmus bessere Ergebnisse

erzielen, als der Viterbi-Algorithmus. [13]

Das Ergebnis der Decoding-Algorithmen ist ein einzelner rekonstruierter Pfad aus einer Beobachtungssequenz. Um die Güte dieser Rekonstruktion zu messen, verwenden wir Distanzmaße zwischen dem echten Pfad und dem rekonstruierten Pfad.

1.4 Distanzmaße für Sequenzen

Die in diesem Paper verwendeten Distanzmaße sind die Hamming-Distanz und die Damerau-Levenshtein-Distanz. Die Hamming-Distanz ist ein weit verbreitetes Distanzmaß. Diese entspricht der Anzahl der Unterschiede zwischen zwei Zeichenketten. Dabei werden nur Zeichen mit gleichem Index miteinander verglichen. Dies hat zur Folge, dass nur Zeichenketten mit gleicher Länge verglichen werden können. Vorteil der Hamming-Distanz ist, dass sie sehr einfach und ressourcensparend zu berechnen ist.

Die Damerau-Levenshtein-Distanz ist die Anzahl der Operationen, die es benötigt, um die Zeichenketten in einander umzuformen. Die Operationen sind: Einfügen, Ersetzen oder Löschen eines Zeichens, sowie das Tauschen zweier benachbarter Zeichen. Die kleinste benötigte Anzahl an Operationen entspricht dann der Distanz. Werden nur Ersetzungsoperationen verwendet, so ist die Damerau-Levenshtein-Distanz gleich der Hamming-Distanz. Aber im Gegensatz zur Hamming-Distanz ermöglicht die Definition als Minimierungsproblem die Verwendung von Zeichenketten unterschiedlicher Länge. Zusätzlich kann das Löschen und Einfügen von Zeichen die Distanz stark senken. Abb. 1 zeigt dieses Problem.

A	B	A	A	B	A	B	B	A	B	A	A	B	A	-
A	B	A	A	B	A	B	B	A	B	A	A	B	A	A

Abb. 1: Zwei Zeichenketten mit Damerau-Levenshtein-Änderung (rot) und Hamming-Übereinstimmung (grün)

Die Damerau-Levenshtein Distanz in Abb. 1 ist eins, da lediglich das hervorgehobene „A“ hinzugefügt bzw. gelöscht werden muss. Die Hamming-Distanz ist hingegen sieben. Dabei wurde eine angepasste Version der Hamming-Distanz verwendet, die „Null“-Zeichen

am Ende hinzufügt. Dadurch haben beide Zeichenketten die gleiche Länge. Insbesondere bei der Verwendung von Virtuellen Stochastischen Sensoren und der Berechnung von Ereignissequenzen treten solche Unterschiede sehr häufig auf.

Ein Nachteil der Damerau-Levenshtein-Distanz ist, dass die Laufzeit in $O(n^2)$ liegt. Zudem sind beide Distanz-Maße nicht direkt vergleichbar, da es auf den Anwendungsfall oder die Fragestellung ankommt.[14, 15]

2 Algorithmus

Die Proxel-Methode exploriert den Zustandsraum. Der dabei erzeugte Proxel-Baum entspricht einer Markov-Kette. Auf diesen Gedanken baut die hier vorgestellte Methode auf. Anstatt einer Markov-Kette wird, der mit der Proxel-Methode explorierter Zustandsraum, in ein EHMM umgewandelt. Auf diesem EHMM werden dann die Decoding-Algorithmen angewendet. Das EHMM kann gespeichert werden, da das EHMM systemspezifisch und unabhängig vom Trace ist. In diesem Kapitel stellen wir die Anpassung der Proxel-Methode vor. Abbildung 2 zeigt den Ablauf des Algorithmus zur Erzeugung der EHMM.

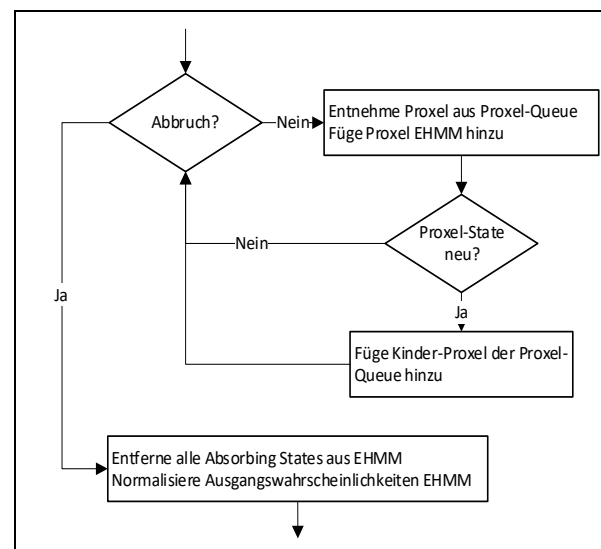


Abb. 2: Programmablaufplan zur Erzeugung der EHMM

2.1 Abbruchbedingungen

Da der Zustandsraum durchaus unendlich sein kann, ist eine Abbruchbedingung notwendig. In diesem Paper

wurde als Abbruchbedingung die Anzahl der Zustände im EHMM gewählt, da diese maßgeblich Einfluss auf Rechenzeit und Genauigkeit der Ergebnisse hat. Andere mögliche Abbruchbedingungen sind Zeitschritt des jüngsten oder ältesten Proxels, sowie die Wahrscheinlichkeit des wahrscheinlichsten Proxels in der Proxel-Queue oder die Gesamtwahrscheinlichkeit, die in der Proxel-Queue noch verbleibt. Es wird abgebrochen, wenn ein bestimmter Grenzwert unterschritten wird.

2.2 Proxel-Queue

Im ursprünglichen Proxel-Algorithmus wurde jeder Zeitschritt vollständig abgeschlossen, bevor der nächste exploriert wurde. Dies ist bei der vorgestellten Methode nicht ratsam. Für dieses Paper wurden die Proxel in einer Priority-Queue mit deren Wahrscheinlichkeiten als Sortierparameter gespeichert. Das wahrscheinlichste Proxel wird zuerst entnommen. Dadurch werden auch die wahrscheinlichsten Zustände zuerst exploriert und sind somit im EHMM enthalten.

2.3 EHMM

Das EHMM enthält alle Zustände und Transitionen. Beim Hinzufügen des Proxels zum EHMM enthält das Proxel alle notwendigen Informationen. Dazu gehört die Transition, die den Zustandswechsel verursacht hat, sowie deren Wahrscheinlichkeit. Informationen über die Ausgabesymbole sind transitionsspezifisch und modellabhängig aber konstant. Auch der Elternzustand ist im Proxel gespeichert, sodass der Zustand oder die Transition dem EHMM hinzugefügt werden können.

Ein Problem sind Senken. Das sind Zustände, die keine Ausgangstransitionen besitzen. Dies trifft auf die letzten hinzugefügten Zustände zu, da deren Kinderzustände noch nicht exploriert wurden. Bei der Berechnung „absorbieren“ diese die Wahrscheinlichkeit. Da diese in der Realität keine Senken sind, müssen sie aus dem EHMM entfernt werden. Dadurch können neue Senken entstehen, die ebenfalls entfernt werden müssen.

Desweiteren müssen auch alle Ausgangswahrscheinlichkeiten normiert werden, damit keine Wahrscheinlichkeit verloren geht. Der Grund für die fehlende Ausgangswahrscheinlichkeiten ist derselbe, wie bei den Senken. Einige Kinderproxel

wurden nicht exploriert, weshalb der Link zu den Zuständen fehlt.

3 Experimente

Für die Experimente wurde ein einfaches, nicht markov'sches Modell durchgerechnet. Sowohl die Proxel-Methode als auch die erzeugten EHMMs können das Evaluations- und das Dekodierungsproblem lösen. In diesem Paper betrachten wir nur das Dekodierungsproblem.

Für das Modell wurden 1000 Pfade mit dazugehörigem Trace generiert. Ziel ist es den echten Pfad aus dem Trace zu dekodieren.

Als Distanzmaß wurde zum Vergleich die Damerau-Levenshtein-Distanz verwendet, sowie die Hamming-Distanz. Da sich aufgrund der Art der Pfade die Hamming-Distanz und die Damerau-Levenshtein Distanz kaum unterscheiden, werden alle Diagramme in diesem Paper nur die Damerau-Levenshtein Distanz enthalten. Die Hamming Distanz ist um rund 10% größer als die Damerau-Levenshtein Distanz.

Als Vergleichsparameter zwischen der hier vorgestellten Methode und der ursprünglichen Proxel-Methode werden relative Distanz zwischen echtem und rekonstruiertem Pfad sowie die Rechenzeit verwendet. Für den Posterior bzw. Posterior-Viterbi-Algorithmus wurde nur die relative Distanz verwendet.

Für die Proxel-Methode wurden die Anzahl der Proxel pro Zeitschritt limitiert, für die EHMMs wurden die Anzahl der Zustände limitiert. Dabei wurden die Modelle mit 10k Zuständen und 100k Zuständen durchgerechnet. Durch das Entfernen aller Senken resultieren diese in einem EHMM mit 7k beziehungsweise 84k Zuständen. Da die EHMMs nur einmal erzeugt werden müssen und dann für jeden Trace wiederverwendet werden können, wurde die Erzeugung nicht in die Rechenzeit mit einbezogen. Diese enthält nur die reine Zeit, die zum Dekodieren benötigt wurde.

3.1 Model

Das System stellt den Bedienprozess einer Autovermietung dar. Die Autovermietung hat im Grundsystem eine Tür. Alle Kunden betreten und Verlassen die Autovermietung durch diese Tür. Passiert

ein Kunde die Tür, so wird dies über einen Sensor registriert. Es gibt Premium-Kunden, die mit Priorität behandelt werden, und eine längere Bedienzeit haben. Es gibt einen Angestellten. Ziel des VSS ist es, die Länge der Warteschlangen anhand des Türprotokolls zu bestimmen.

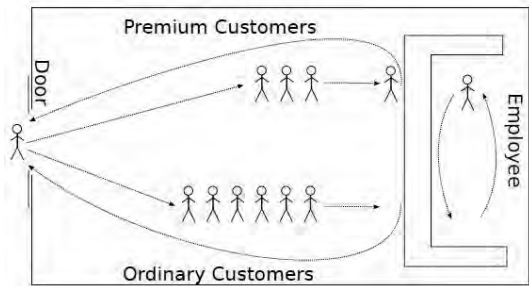


Abb. 3: Autovermietung mit zwei getrennten Warteschlangen für VIP und normale Kunden und einem Angestellten

Die Zwischenankunftszeit der normalen Kunden ist Normal-Verteilt: $N_{ord}(22,5, 5)$. Die Zwischenankunftszeit der VIP-Kunden ist ebenfalls Normal-Verteilt: $N_{vip}(45,5)$. Die Bearbeitungszeit pro Kunde ist Weibull-Verteilt: $W_{ord}(12,2)$ und $W_{vip}(16,2)$

Die Bearbeitungszeit pro Kunde (11.8 Zeiteinheiten) ist im Schnitt geringer als deren Zwischenankunftszeiten (15 Zeiteinheiten). Dies ist wichtig, da sich sonst die Kunden aufstauen würden. Ein Zustand enthält die Anzahl der Personen in den jeweiligen Warteschlangen sowie den Zustand des Angestellten. Zur Diskretisierung wurde ein Zeitschritt von 5 Zeiteinheiten gewählt.

3.2 Rechenzeit

Um die Rechenzeit zu vergleichen wurde die Zeit, die die jeweiligen Algorithmen benötigen, gemessen. Hier zeigt sich, dass die Proxel-Methode schneller als die Hybride-Proxel-Methode ist. Dies liegt daran, dass die Modelle mit deutlich weniger Zuständen/Proxel berechnet werden können, ohne Einbuße der Genauigkeit. Bei gleicher Anzahl an Proxel/Zuständen ist die Hybride-Proxel-Methode schneller. Die Proxel-Methode mit 1k-Zuständen ist doppelt so schnell, wie das EHMM. Das EHMM mit 84k Zuständen ist 34x langsamer als das EHMM mit 7k Zuständen, zur besseren Übersichtlichkeit ist diese daher nicht in Diagramm 1 enthalten. Die Werte in Diagramm 1 wurde normiert zum EHMM mit 7k Zuständen, um das

Verhältnis der Methoden besser darzustellen. Beide Methoden haben linearen Rechenaufwand in Abhängigkeit zu Pfadlänge.

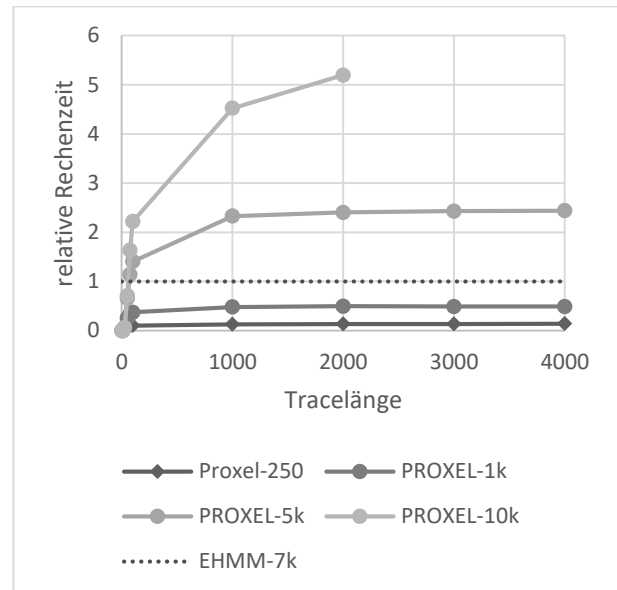


Diagramm 1: relative Rechenzeit normiert zu EHMM-7k

3.3 Genauigkeit

Die Hybride-Proxel-Methode zeigt ähnliche Ergebnisse, wie die Proxel-Methode. Die Ergebnisse für die EHMM mit 7k Zuständen sind geringfügig schlechter als die Proxel-Methode.

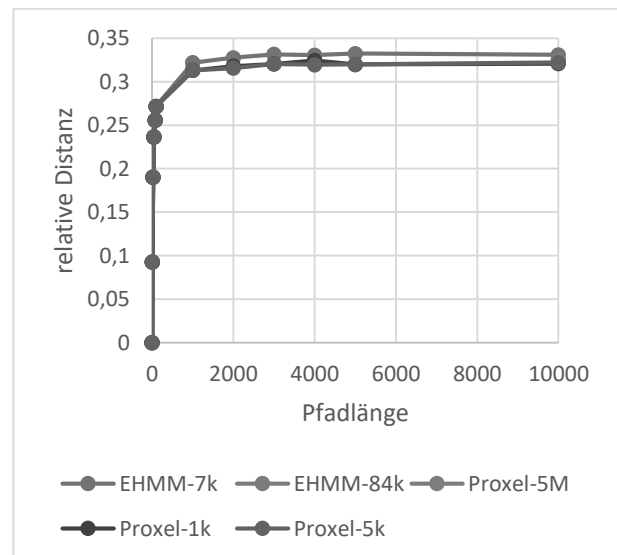


Diagramm 2: Durchschnittliche relative Distanz zum richtigen Pfad für die Proxel-Methode und die Hybride-Proxel-Methode (EHMM) mit verschiedener Begrenzung des

Zustandsraums.

Die Proxel-Methode liefert für dieses Modell sogar noch bis 250 Proxel pro Zeitschritt bessere Ergebnisse als die EHMM mit 7k Zuständen, wenn der Viterbi Algorithmus verwendet wird. Die Ergebnisse zwischen 250 Proxeln und 5M Proxeln unterscheiden sich dabei nicht. Hier ist die Proxel-Methode eindeutig überlegen.

Verwendet man allerdings den Posterior- oder den Posterior-Viterbi Algorithmus, so erzielt das EHMM bessere Ergebnisse. Diagramm 3 zeigt die durchschnittliche relative Distanz zu den jeweiligen echten Pfaden. Die Hybride-Proxel-Methode mit 7k hat die höchste Distanz mit dem Viterbi-Algorithmus. Die Hybride-Proxel-Methode mit 84k Zuständen und dem Viterbi-Algorithmus und die Proxel-Methode liefern etwas schlechtere Ergebnisse. Die geringste Distanz liefert der Posterior/Posterior-Viterbi-Algorithmus. Die Ergebnisse zwischen 7k und 84k und zwischen Posterior und Posterior-Viterbi unterscheiden sich dabei nicht.

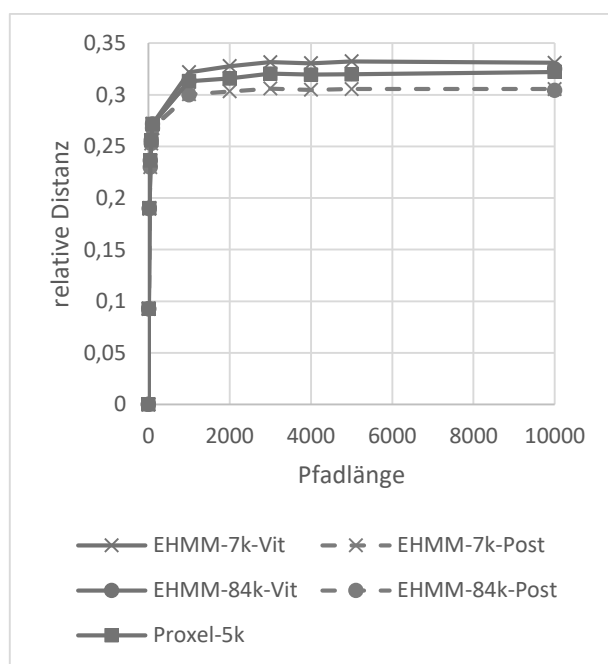


Diagramm 3: Durchschnittliche relative Distanz zum richtigen Pfad für die Proxel-Methode und die Hybride-Proxel-Methode (EHMM)

4 Result Discussion

Das Konvertieren der Proxel-Methode in ein EHMM ist nicht nur möglich, sondern bietet auch einige Vorteile. Im Gegensatz zur Proxel-Methode können nicht nur der

Forward-Algorithmus, sondern auch der Backward-Algorithmus ausgeführt werden. Dadurch kann nicht nur der Viterbi-Algorithmus zum Dekodieren verwendet werden, sondern auch Algorithmen wie Posterior oder Posterior-Viterbi-Algorithmus.

Unsere vorgestellte Methode zeigt nur kleine Einbuße bezüglich des Erkennungsgrades. Betrachtet man den Posterior bzw. Posterior-Viterbi-Algorithmus, so erzielt das EHMM sogar bessere Ergebnisse. Ob allerdings der Posterior oder Posterior-Viterbi-Algorithmus angewendet werden kann, ist Problem abhängig. Insbesondere der Posterior-Algorithmus muss nicht zwangsläufig einen echten Pfad ausgeben: Für zwei aufeinanderfolgende Zustände in der Ergebnissequenz muss es keinen Übergang von ersteren in den zweiten geben.

Das Berechnen der Forward und Backward-Wahrscheinlichkeiten hat allerdings noch andere Vorteile. So kann dadurch die Frage beantwortet werden, wie Wahrscheinlich ein Zustand s zum Zeitpunkt t ist, unter der Bedingung das ein Trace T beobachtet wurde. Dabei kann die Trace-Länge auch größer als t sein. Die Proxel-Methode kann diese Frage nicht beantworten.

Im Bezug auf Rechenleistung hingegen kann das EHMM nicht mit der Proxel-Methode mithalten. Selbst der Viterbi-Algorithmus benötigt bis zu 8x mehr Rechenzeit bei vergleichbaren Ergebnissen. Der Posterior-Algorithmus benötigt entsprechend 16x mehr Rechenzeit. Dies liegt daran, dass die Proxel-Methode nur valide Zustände exploriert. Das verwendete Modell erzeugt in jedem Zeitschritt eine Ausgabe, die τ -Werte hängen daher direkt mit den beobachteten Ausgaben zusammen. So ist der kleinste τ -Wert z.B. die Anzahl Zeitschritte seit der letzten Ausgabe. In den EHMMs sind alle Zustände unabhängig von den Beobachtungen gespeichert. Zudem wurde die Zeit zum Erzeugen der EHMMs nicht mit eingerechnet. Muss nur ein einzelner kurzer Pfad dekodiert werden, so ist der Overhead für die Konvertierung zu groß.

Zudem kann bei besonders langem Pfaden der Speicherverbrauch auch ein limitierender Faktor sein. Der Speicherverbrauch des Forward- und Backward-Algorithmus liegt in $O(n * |T|)$, wobei n die Anzahl der Zustände des EHMMs ist und $|T|$ die Tracelänge. Bei sehr langen Traces ist daher der Speicherverbrauch der

limitierende Faktor, da Viterbi und Posterior jeweils auf den Forward/Backward-Algorithmus aufbauen. Diese können, wie der Name suggeriert, nur in eine Richtung durchlaufen werden. Ein Zurückführen zu früheren Werten ist nicht möglich. Für das Backtracking werden alle Werte in der umgedrehten Reihenfolge benötigt. Daher müssen alle Werte für die Berechnung gespeichert werden. Um den Speicherverbrauch zu reduzieren haben wir auch eine angepasste Variante der Algorithmen entwickelt. Anstelle alle Berechnungen zu speichern, werden nur die Ergebnisse jedes $\sqrt{|T|}$ -tem Schritt gespeichert. Die Werte können dann vom letzten Zwischenergebnis aus berechnet werden. Der Speicherverbrauch reduziert sich so auf $O(n * \sqrt{|T|})$. Für den Viterbi-Algorithmus bedeutet dies ungefähr eine Verdopplung der Rechenzeit, da alle Berechnungen zweimal ausgeführt werden müssen. Der Rechenaufwand des Posterior-Algorithmus steigt um 50%, da nur entweder der Forward- oder der Backward-Algorithmus angepasst werden muss.

Die verwendeten Distanz-Maße Damerau-Levenshtein, sowie die Hamming-Distanz liefern ähnliche Ergebnisse. Dies ist der Art des Modells und der Pfade geschuldet. Da alle Ereignisse beobachtbar sind, sind die Pfadlängen und die Veränderung in den einzelnen Zuständen zeitlich genau bestimmbar. Dadurch verhält sich die Damerau-Levenshtein ähnlich zur Hamming-Distanz. Das Einfügen eines Zustandes und Löschen eines anderen Zustandes ist hier nicht hilfreich, um einen Pfad in einen anderen umzuwandeln. Auch das Vertauschen von Zuständen verringert die Distanz nicht, da nicht die Ereignisse, sondern die internen Zustände im Pfad enthalten sind. Werden zwei Events in falscher Reihenfolge erkannt, so ist der Anfangs und Endzustand dennoch derselbe.

Einschränkend ist allerdings zu sagen, dass nur ein Modell zum Vergleich verwendet wurde. Die Einsetzbarkeit und der Nutzen der Proxel-Methode, als auch der Konvertierung in ein EHMM hängen stark vom Modell und Anwendungsfall ab. Das verwendete Modell ist stationär, was dem EHMM entgegen kommt. Das Modell fällt immer wieder in dieselben Zustände zurück. Wäre dies nicht der Fall, dann würden viele der Zustände des EHMMs bei längeren Traces nur einmal benötigt werden. Das EHMM ist dann deutlich größer

als notwendig, was die Rechenzeit erhöhen würde. Zudem erzeugt jedes Ereignis eine Ausgabe. Bei nicht beobachtbaren Ereignissen, kann es sein, dass auch die Proxel-Methode deutlich mehr Zustände pro Zeitschritt benötigen würde.

Bezüglich der Geschwindigkeit ist anzumerken, dass die Experimente auf i7-6700HQ ausgeführt wurden. Der Prozessor ist schon mehrere Jahre alt. Auf neueren Prozessoren können die Ergebnisse anders ausfallen. Auch die Möglichkeit der Parallelisierung wurde nicht untersucht. Für die Proxel-Methode ist ein parallelisieren innerhalb eines Zeitschrittes nur schwer möglich. Bei den EHMMs ist dies allerdings sehr einfach, da es innerhalb eines Zeitschrittes pro Zustand nur Concurrent-Reads und keine Concurrent-Writes auf derselben Speicheradresse gibt.

Die sauberere Neuimplementation der EHMMs ist zudem 4x schneller als die in diesem Paper verwendete Version, da ein unnötiger Hash-Table-Lookup entfernt wurde. Damit können die EHMMs auch im Bereich Geschwindigkeit mit der Proxel-Methode mithalten. Auch ein Viterbi auf Basis eines Dijkstra kann zu zusätzlichen deutlichen Geschwindigkeitssteigerungen führen. Erste Tests legen nahe, dass auch die Verwendung von Sparse-Vectors einen erheblichen Geschwindigkeitsvorteil bringt. Ähnlich wie bei der Proxel-Methode werden dann auch nur die Zustände berechnet, deren Wahrscheinlichkeit nicht 0 ist. Wir haben uns aber gegen die neue Version für dieses Paper entschieden, um sicher zu gehen, dass wir nicht eine einseitige Optimierung für den neuen Algorithmus vornehmen. Auch bei der Proxel-Methode kann noch viel Optimierungspotenzial vorhanden sein. Neben der Optimierung der Datenstrukturen, scheint vor allem Step-Size-Control hier eine vielversprechende Idee zu sein.

5 Conclusion and Future Work

Unsere Ergebnisse haben gezeigt, dass die Umwandlung in ein EHMM für das Dekodierungsproblem im Bereich Genauigkeit mit der Proxel-Methode mithalten kann. Im Bereich Rechenzeit erzielt unsere neue Methode schlechtere Ergebnisse. Allerdings zeigen bereits neue Untersuchungen, dass

dort noch deutlich Optimierungspotential besteht und die neue Methode durchaus mit der alten Methode auch im Bereich Rechenzeit mithalten kann.

Mit unserer vorgestellten Methode lassen sich allerdings nicht nur die Forward-Wahrscheinlichkeiten berechnen, sondern auch die Backward-Wahrscheinlichkeiten. Dadurch können deutlich mehr Anwendungsfelder abgedeckt werden. Zukünftige Untersuchungen müssen zeigen, ob dadurch sogar Smoothing und Training von Modellen möglich wird. Auch dies ist mit der Proxel-Methode bisher nicht möglich.

Allerdings wurde unsere Methode nur an einem Modell getestet. In wie weit die EHMMs für größere und nicht stationäre System anwendbar ist, muss noch untersucht werden. Ebenfalls ist es interessant wie die EHMMs mit nur teilweise-beobachtbaren Ereignissen performt. Während das Laufzeitverhalten der Proxel-Methode dadurch deutliche Verschlechterungen erfährt, wird das Laufzeitverhalten der EHMMs dadurch nicht beeinflusst. Aber ob dies auch für die Genauigkeit der Fall ist, muss noch untersucht werden.

Zudem wurde nur das Decoding untersucht. Ob die EHMMs auch das Evaluierungs-Problem genauso gut lösen können, wie die Proxel-Methode, müssen zukünftige Untersuchungen noch zeigen. Davon ist allerdings auszugehen, da der im Viterbi-Algorithmus verwendete Forward-Algorithmus sich nur wenig vom echten Forward-Algorithmus unterscheidet.

References

- [1] L. Fialho Müller, *Feasibility and Applicability of Virtual Stochastic Sensors for Human Activity Recognition in the Context of Ambient*, 2022.
- [2] T. Dittmar, C. Krull and G. Horton, *A new approach for touch gesture recognition: Conversive Hidden non-Markovian Models*, vol. 10, 2015, pp. 66-76.
- [3] T. Dittmar, C. Krull und G. Horton, *AN IMPROVED CONVERSIVE HIDDEN NON-MARKOVIAN MODEL-BASED TOUCH GESTURE RECOGNITION SYSTEM WITH AUTOMATIC MODEL CREATION*, Bergeggi, 2015.
- [4] T. Dittmar, C. Krull und G. Horton, *Evaluating a New Conversive Hidden non-Markovian Model Approach for Online Movement Trajectory Verification*, Porto, 2017.
- [5] C. Krull, *Virtual Stochastic Sensors: Formal Background and Example Applications*, Magdeburg: Shaker Verlag Düren, 2021.
- [6] C. Krull, R. Buchholz und G. Horton, „Virtual Stochastic Sensors: How to gain Insight into Partially Observable Discrete Stochastic Systems,“ in *s The 30th IASTED International Conference on Modelling, Identification and Control*, 2011.
- [7] G. Horton, „A NEW PARADIGM FOR THE NUMERICAL SIMULATION OF STOCHASTIC PETRI NETS WITH GENERAL FIRING TIMES,“ in *s European Simulation Symposium*, Dresden, 2002.
- [8] S. Lazarova-Molnar und G. Horton, *Proxel-Based Simulation of Stochastic Petri Nets*, SCS European Publishing House, 2004.
- [9] S. Lazarova-Molnar und G. Horton, *Proxel-Based Simulation of Stochastic Petri Nets Containing Immediate Transitions*.
- [10] F. Wickborn, C. Isensee, T. Simon, S. Lazarova-Molnar und G. Horton, *A New Approach for Computing Conditional Probabilities of General Stochastic Processes*, 2006.
- [11] D. R. Cox, „The analysis of non-Markovian stochastic processes by the inclusion of supplementary variables,“ *Mathematical Proceedings of the Cambridge Philosophical Society*, pp. 433-441, 1955.
- [12] C. Krull und G. Horton, *EXPANDED HIDDEN MARKOV MODELS: ALLOWING SYMBOL EMISSIONS AT STATE CHANGES*, Prague, 2007.
- [13] P. Fariselli und P. L. Martelli, *The posterior-Viterbi: a new decoding algorithm for hidden Markov models*, 2005.
- [14] V. I. Levenshtein, „Binary Codes Capable of Correcting Deletions, Insertions and Reversals,“ *Soviet Physics Doklady*, Bd. Vol. 10, p. 707, February 1966.
- [15] F. J. Damerau, „A technique for computer detection and correction of spelling errors,“ *Commun. ACM*, Bd. 7, pp. 171-176, March 1964.

git-Repository

Alle Quellcodes und Daten werden unter folgendem Link zur Verfügung gestellt: <https://github.com/Virtual-Stochastic-Sensors>

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

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

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

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

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

Introduction

Today's systems are often characterized by a high degree of variability. Variability modeling means to describe several system configurations. A system configuration represents one variant characterized by a sys-

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

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

The automated goal-driven selection of system variants is a prerequisite for automating simulation studies involving different system configurations. Accordingly, Schmidt [9] proposes an extended SES/MB-based software architecture to automate simulation studies in the MATLAB/Simulink environment. Based on [9], a software architecture is developed in [10] that supports the generation of executable simulation models for different target simulators. Among other things, this is based on the use of the *Functional Mock-up Interface* (FMI). In particular, the build method and the MB have been further developed so that, in addition to the SES, the

MB is also largely simulator-independent.

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

1 Some Basics of SES

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

In terms of variant and variability modeling, multi-aspect and specialization nodes represent variation points. In order to derive one specific system configuration by pruning an SES all variation points have to be resolved by evaluating the node attributes. However, not only attributes at nodes of variation points have to be

evaluated. Pruning can also involve adjustments to coupling relationships or attribute changes to entity nodes when resolving specializations. The result of each pruning operation is a *Pruned Entity Structure* (PES). A PES codes exactly one system configuration. There are several ways to implement a pruning method [4, 6]: (i) interactive pruning, (ii) automated pruning, (iii) enumerative pruning to derive all possible variants, (iv) selective pruning to derive one variant, etc. This paper focuses on the last one, the automated, goal-directed selection of exactly one system variant by pruning. For this purpose, we use the extension of the SES with information about pruning in the node attributes according to [8, 9, 12].

2 An Extended SES/MB-based Software Architecture

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

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

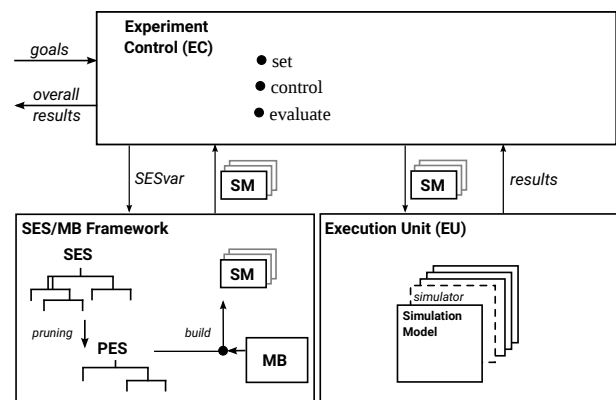


Figure 1: Extended SES/MB-based software architecture.

The EC is a higher-level control unit. It defines the experiment goals, steps, and settings. Experiment steps and settings can reactively depend on previous simulation results. The EC activates the other components and evaluates their operations. The EU is an interface to target simulators on which the generated SM is executed. It is a kind of wrapper and uses the *Application Programming Interface* (API) of a target simulator to execute the simulation and to collect results. The SES/MB framework provides an interface to communicate with the EC. The newly defined SESvars are used as input interface. As output, the framework returns the SM derived by pruning and generated by the build method to the EC.

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

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

3 Modeling and Pruning of SES with Multi-Aspect Nodes

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

Restructuring SES' to avoid hierarchies of multi-aspects and specializations as suggested in [5] is challenging to automatize. Additional attributes may be needed for the restructured SES describing the same set of system configurations and it is doubtful how values can be assigned to these attributes during pruning. In the next subsections it is demonstrated how hierarchies of multi-aspect nodes in combination with specialization nodes can be pruned automatically.

For illustration, we use a problem from the field of process industry, which is stepwise extended. Flexibilization in the process industry through modularization

leads to a large number of process alternatives and parameter variants in plant design and operation [15]. On the other hand, modularization increases the reusability of models or model components [16]. The variant management in modular plant design according to [17] could be efficiently handled with the SES/MB approach.

3.1 Single Multi-Aspect

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

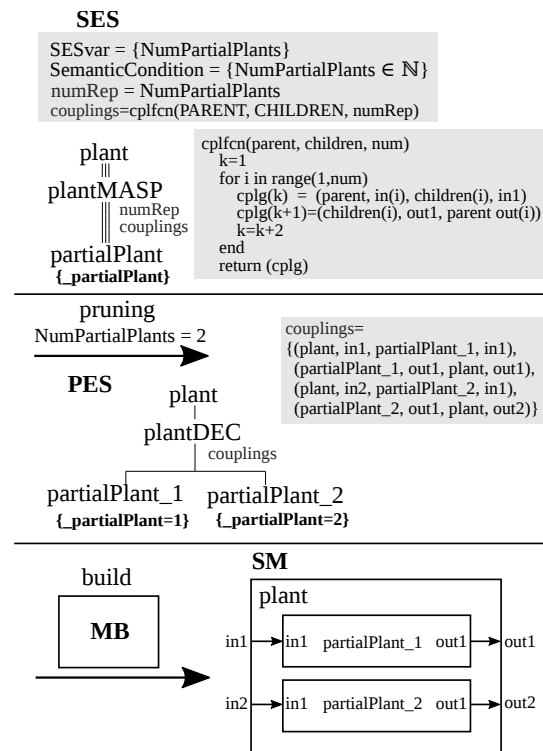


Figure 2: SES with a single multi-aspect, derivation of a possible PES, and the resulting model.

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

Above the SES tree the SESvar *NumPartialPlants* is defined as input interface. The following Semantic

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

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

Under the SES, Figure 2 shows the derivation of a possible PES by pruning. In the example, the value two was assigned to the SESvar *NumPartialPlants* before pruning. Accordingly, two replications of the entity node *partialPlant* are generated with the names *partialPlant_1* and *partialPlant_2*, and their attribute values are implicitly assigned to them. Also, the multi-aspect node is converted into an aspect node and the *couplings* attribute is computed using the SESfcn. Thus, the aspect node describes the parallel composition of the two partial plants.

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

3.2 Multi-Aspect with Succeeding Specialization

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

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

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

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

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

In the 1st pruning step the multi-aspect *plantMASP* is resolved and the two entities *partialPlant_1* and *partialPlant_2* are generated. The subsequent subtree starting with the specialization node is appended to each of the generated entities. The result of this step is called *intermediate PES*. In the 2nd step the variation point specified by the specialization *partialPlantSPEC* is resolved. Pruning a specialization results in the union of the parent node with one selected child. Children,

SES

```

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

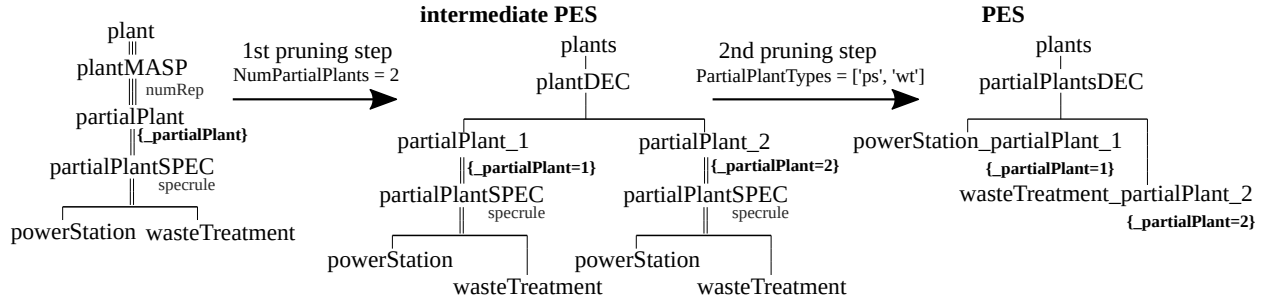


Figure 3: SES with a multi-aspect followed by a specialization and stepwise derivation of a possible PES.

which are not selected, are removed like the specialization node itself. The selection is controlled by the rules in the *specrule* attribute. In this example, the SESfcn *ppTypes* is called in the *specrule*. It receives as input arguments the value of the underscore attribute *_partialPlant* of the parent node and the vector defined in the SESvar *PartialPlantTypes*. Depending on the value in *_partialPlant* an element of the vector *PartialPlantTypes* is returned, which describes the category to be selected and thus the selection of a child node. The selected child node and the parent node are merged into one entity node, as indicated by the merged node name in the PES.

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

3.3 Several Multi-Aspects and Specializations in a Common Path

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

of possible system configurations exponentially.

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

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

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

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

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

SES

```

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

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

numRep1 = NumPartialPlants
specrule1 = {ppTypesFun(_partialPlant, PartialPlantTypes) == "ps"→ powerStation
    ppTypesFun(_partialPlant, PartialPlantTypes) == "cp"→ chemicalProduction
    ppTypesFun(_partialPlant, PartialPlantTypes) == "wt"→ wasteTreatment
}
numRep2 = cpNumFun(_partialPlant, PartialPlantTypes, NumChemicalSubProductions)
specrule2 = {cpTypesFun(_partialPlant, _chemicalSubProduction, PartialPlantTypes, ChemicalProductionTypes) == "ac"→ acid
    cpTypesFun(_partialPlant, _chemicalSubProduction, PartialPlantTypes, ChemicalProductionTypes) == "ba"→ base}
    
```

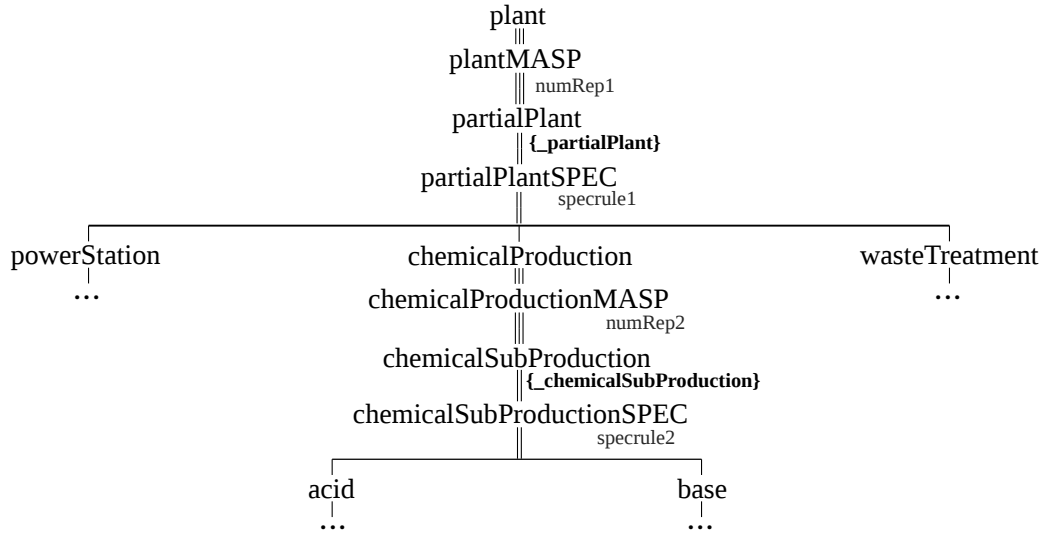


Figure 4: SES with two multi-aspects in one path and each followed by a specialization.

1st and 2nd pruning steps: The 1st pruning step is performed analogously to Section 3.1. According to the *numRep1* attribute at the *plantMASP* node, four entity nodes *partialPlant* are generated. Then, the 2nd pruning step is executed analogously to Section 3.2. By pruning the node *partialPlantSPEC* with the attribute *specrule1*, the four previously created entity nodes *partialPlant_1* ... *partialPlant_4* are specialized according to the value assignment of the SESvar *PartialPlantTypes* to: *powerStation_partialPlant_1*,

chemicalProduction_partialPlant_2, *chemicalProduction_partialPlant_3*, and *wasteTreatment_partialPlant_4*. The last operation of the 2nd pruning step is to attach the remaining subtree of the SES to the two entity nodes *chemicalProduction_partialPlant_2* and *chemicalProduction_partialPlant_3*. The result of these pruning steps is illustrated in Figure 5 as *intermediate PES 2*.

3rd pruning step: In the third pruning step the two nodes *chemicalProductionMASP* are resolved. For

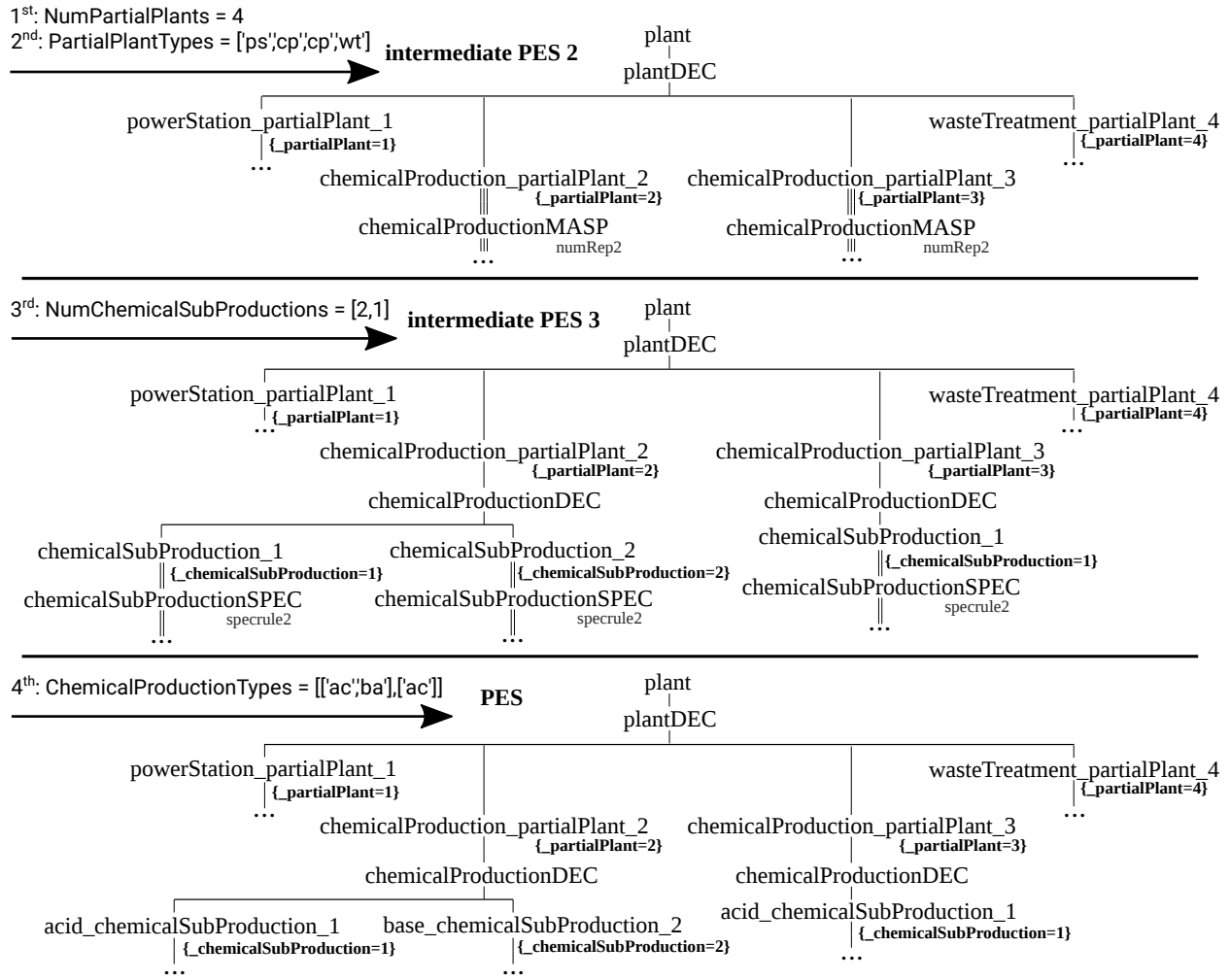


Figure 5: Step-by-step derivation of a system variant by pruning.

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

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

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

selected child node with the parent node is done analogously to Section 3.2. The final result of pruning is shown with the PES in Figure 5.

Short evaluation. The implicitly managed underscore attribute on entities generated during pruning in combination with SESfcns allows automated derivation of a system configuration (PES) specified with SES-var. However, the complexity of SESfcns increases significantly as the number of multi-aspects in a path increases. The axiom of uniformity specified for the SES states: nodes with the same name need to have the same variables and isomorphic subtrees. In the SES this axiom is fulfilled, but relaxed in the PES. In Figure 5 the nodes *chemicalProductionDEC* do not have isomorphic subtrees. In the context of simulation engineering this does not pose any problem. The extended pruning approach is implemented in the Python-based toolset [14].

4 Conclusion

The example has shown that the combination of multiple multi-aspects with subsequent specializations in an SES path supports a compact specification of a large number of system variants. The introduction of an implicitly managed attribute on entities generated by multi-aspects during pruning supports an automated derivation of a goal-directed system configuration (PES). Thus, an automatic model generation using the extended SES/MB architecture is supported.

References

- [1] Zeigler BP. *Multifaceted Modelling and Discrete Event Simulation*. USA: Academic Press Professional, Inc. 1984.
- [2] Rozenblit JW, Zeigler BP. Representing and constructing system specifications using the system entity structure concepts. In: *Proceedings of the 25th Winter Simulation Conference, Los Angeles, California, USA, December 12-15, 1993*, edited by Evans GW, Mollaghasemi M, Russell EC, Biles WE. ACM Press. 1993; pp. 604–611.
- [3] Zeigler BP, Kim TG, Praehofer H. *Theory of Modeling and Simulation*. USA: Academic Press, Inc., 2nd ed. 2000.
- [4] Rozenblit JW, Huang Y. Rule-Based Generation of Model Structures in Multifaceted Modeling and System Design. *INFORMS J Comput.* 1991;3(4):330–344.
- [5] Zeigler BP, Hammonds PE. *Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange*. Orlando, FL, USA: Academic Press, Inc. 2007.
- [6] Zeigler BP, Sarjoughian HS. *Guide to Modeling and Simulation of Systems of Systems*. Simulation Foundations, Methods and Applications. Springer. 2013.
- [7] Santucci JF, Capocchi L, Zeigler BP. System Entity Structure Extension to Integrate Abstraction Hierarchies and Time Granularity into DEVS Modeling and Simulation. *Simulation*. 2016;92(8):747–769.
- [8] Pawletta T, Schmidt A, Zeigler BP, Durak U. Extended Variability Modeling Using System Entity Structure Ontology Within MATLAB/Simulink. In: *Proceedings of the 49th Annual Simulation Symposium, ANSS '16*. San Diego, CA, USA: Society for Computer Simulation International. 2016; pp. 22:1–22:8.
- [9] Schmidt A. Variant Management in Modeling and Simulation Using the SES/MB Framework. Ph.D. thesis, Rostock University. 2019.
- [10] Folkerts H, Pawletta T, Deatcu C. Model Generation for Multiple Simulators Using SES/MB and FMI. *SNE Simulation Notes Europe*. 2019;31(1):25–32.
- [11] Folkerts H, Pawletta T, Deatcu C, Zeigler BP. Automated, Reactive Pruning of System Entity Structures for Simulation Engineering. In: *Proceedings of the 2020 Spring Simulation Conference, SpringSim '20*. San Diego, CA, USA: Society for Computer Simulation International. 2020; .
- [12] Deatcu C, Folkerts H, Pawletta T, Durak U. How to Define SES Trees for Variability Modeling. *SNE Simulation Notes Europe*. 2019;29(3):117–126.
- [13] RG CEA. MATLAB SES Tbx. https://github.com/cea-wismar/SES_Tbx_Matlab. 2022. Accessed Feb. 10, 2022.
- [14] RG CEA. Python-based SES/MB Architecture. https://github.com/cea-wismar/SESMB_Inf_Python. 2022. Accessed Feb. 10, 2022.
- [15] DECHEMA. Modulare Anlagen – Flexible chemische Produktion durch Modularisierung und Standardisierung - Status quo und zukünftige Trends. *Tech. rep.*, Germany. 2017.
- [16] von Wedel LA. An Environment for Heterogeneous Model Management in Chemical Process Engineering. Ph.D. thesis, RWTH Aachen. 2004.
- [17] Hady L, Wozny G. Multikriterielle Aspekte der Modularisierung bei der Planung verfahrenstechnischer Anlagen. *Chemie Ingenieur Technik*. 2012;84:597–614.

Analysis of Selected Literature on Co-simulation

Irene Hafner^{1*}, Štefan Emrich¹, Niki Popper^{1,2}

¹dwh GmbH, Neustiftgasse 57-59, 1070 Vienna, Austria; *irene.hafner@dwh.at

²TU Wien, Institute of Information Systems Engineering, Favoritenstraße 9-11, 1040 Vienna, Austria

Abstract. In this paper, a selection of literature on co-simulation has been classified along various aspects and analyzed to illuminate more and less popular topics. General divisions along the works' main topic, regarded model descriptions and considered coupling methods are presented, therein strong and loose coupling approaches, multirate schemes, iterative methods, or adaptive approaches and the progression of these shares over time, revealing tendencies towards more popular methods and areas that invite additional research.

Introduction

Co-simulation has become an important instrument in the area of modeling and simulation as the complexity of considered systems increases permanently. Depending on the disciplines they originate from as well as the level and depth of development, co-simulation methods can be classified by various different means, some of which have been presented in [1]. Consequently, literature on co-simulation can be associated with attributes of this structure and further aspects such as theoretical focus or the number of coupled subsystems.

The literature considered in this paper has been accumulated by strategic search in different catalogues and search machines (by the keywords “co-simulation”, “cooperative simulation”, “coupled simulation”, “hybrid simulation”, “multi-level simulation”, “hierarchical (co-)simulation”), contributions to attended conferences, recommendations by fellow researchers in the area of co-simulation and citations in papers found in the first iteration and again, in these, etc.

The selected work has been classified in two iterations. First, every publication has been carefully read, summed up and assigned certain properties in evidence, such as used model description, coupling algorithm, and application. These properties have then been complemented by methods for structuring developed in the classification found in [1]. In the second iteration, each work has been studied for these properties (or indisputable indications of them) and classified accordingly.

A complete list of these assignments is available in [2]. It is important to bear in mind that some properties could not be defined since the respective authors neither explicitly mention them nor could the property be found out from the paper's context.

1 Publications over the years

Selected by the method described above, a total of 139 publications remains to be considered in the statistical analysis. Regarding the publication dates of these, we observe that although multirate and cooperative simulation has been touched incidentally before the millennium (starting in 1960), research in this area has seen a significant upswing since. Table 1 lists the number of papers per five-year time frame, where publications before 2000 are grouped due to their small total number.

Table 1: Number of papers on co-simulation per time frame.

<2000	2000-2004	2005-2009	2010-2014	2015-2019
17	24	20	38	40

2 Main emphasis in the literature

Depending on the emphasis of the respective publication, each is assigned one topic out of “theory”, “application”, “survey”, “standard” or “framework”. This label describes whether the publication focuses *mainly* on theory, application, etc., not solely. Those where theory and application are quite balanced have been classified “both theory and application”. A majority (63%) of publications mainly covers theoretical aspects. 21 (15%) are applications of already known methods and for eight papers (6%), the theoretical and applied part are quite evenly matched. Eight publications are pure surveys (some limited to an application area of interest), two describe a standard (the HLA [3] and FMI [4]), and

thirteen present a framework. These shares, however, have changed throughout the years, as Figure 1 illustrates.

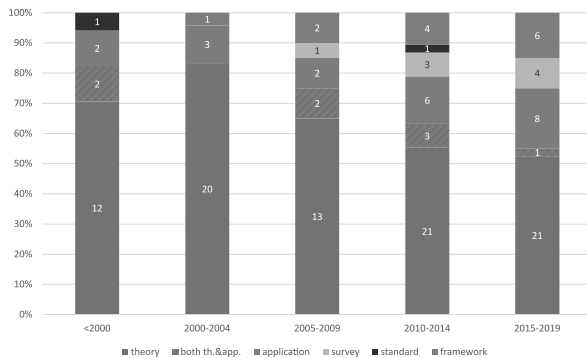


Figure 1: Variation of the main emphases' shares over time regarding five-year intervals. Numbers in the bars denote the quantity of publications per category in the respective time frame. Corresponding percentages can be read off the left-hand axis.

The outstandingly high share of theoretical papers in the years 2000-2004 amounts to more than 80%. The development of frameworks seems to have become more popular over the last two decades. Surveys start to occur in 2009 and have become more frequent ever since. This might simply be explained by the increasing amount of research in this area, thus raising the necessity of aggregating studies.

2.1 Theoretical subcharacterization of the literature

Although only 95 of the 139 publications exhibit a mainly theoretical or shared applied and theoretical focus, 122 (88%) consider at least one theoretical aspect. We further distinguish the following subcategories: error estimates, stability properties, coupling methods, performance, debugging, formalisms and classifications. Of all papers with theoretical aspect, a vast majority of 80% is investigating coupling methods, cf. Figure 2. Of these, over one third (36%) is dealing with the stability of these methods and 28% with error estimation (cf. Table 2). Note that these are not exclusive: again about one third thereof (12% of all that study coupling methods) considers both investigations on stability and error estimates for the regarded coupling methods. About one tenth is analyzing performance properties of these methods.

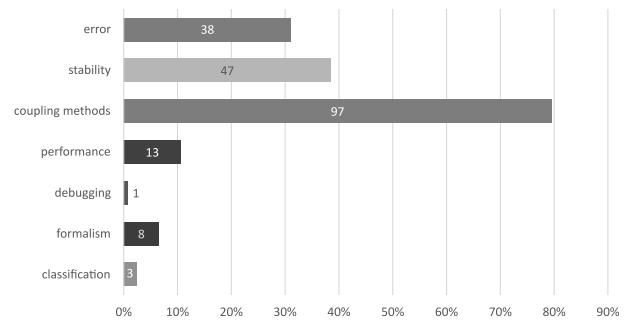


Figure 2: Share of publications considering specific theoretical subcategories. Absolute numbers are given in the bars, percentages are found on the horizontal axis.

Table 2: Number of publications on coupling methods combined with further theoretical categories (percentage of "coupling methods" in parentheses)

coupling methods and			
error	stability	performance	no further category
35 (36%)	27 (28%)	11 (11%)	40 (41%)

Taking a look at the overall parts of theoretical subcategories again in Figure 2, "stability" and "error" are the next most commonly investigated aspects with 39 and 31 percent, respectively. Within those, 19 publications consider both. 8 publications (11%) describe formalisms. These have mainly been published in the last five years. All three papers presenting classifications have also only come out in this time frame. These circumstances again relate to the increase of research and the variety of methods in co-simulation over the last two decades. In contrast, the share of publications including investigations on stability properties and error estimates, which had its peak in the time frame from 2005 to 2009 (at the expense of coupling methods), has dwindled in the last years.

3 Distinction by the state of development

Research on co-simulation methods originates from needs at different stages in the development of simulation models. On the one hand, facing a complex real system with partial systems differing to a great extent in their modeling requirements, these have to be approached with different techniques. The resulting separate simulations need to be coupled thereafter

to sufficiently represent the whole system (*integrate-and-collaborate*). On the other hand, complex systems within one physical domain may be described by one mathematical model that may consist of many equations which may be solved more efficiently by parallelization, possibly divided depending on stiffness. Again, the partial models will be simulated individually and coupled again in an overall simulation, but in this case, the need for separation arises from consideration of the system on the (mathematical) model level (*divide-and-conquer*). Whether a decomposition or collaborative coupling approach is used can be defined for 121 of the regarded publications. 60% of these are approaching their task with an integrate-and-collaborate strategy, 38% use divide-and-conquer methods and 2% utilize or compare both.

Over time, the share of works on integrate-and-collaborate approaches increases noticeably, from 25% to 80%. This consorts with the increase of surveys, standard and framework descriptions in the literature. These, as Figure 3 shows, exclusively represent an integrate-and-collaborate point of view.

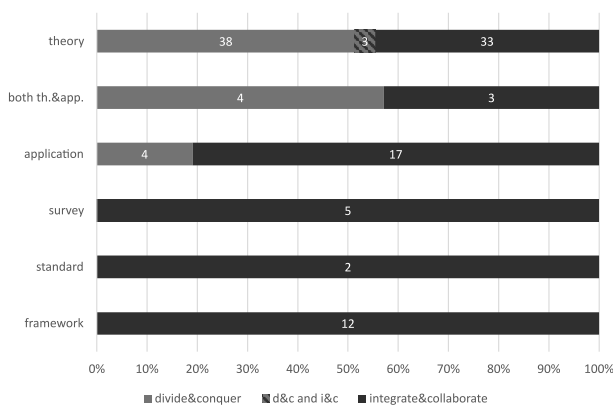


Figure 3: Decomposition of coupling point of view per main topic. The shares of divide-and-conquer and integrate-and-collaborative approaches are given separately for publications assigned to a specific main topic.

The illustration of this cross-connection further reveals that publications mainly presenting an application mostly address an integrate-and-collaborate problem, while for theoretical papers, shares of divide-and-conquer and integrate-and-collaborate approaches are quite balanced.

4 Distinction by field of application

The need for co-simulation arises in various fields of application which are closely related to the kind of coupling methods the systems are approached with and lead to varying developments. Note that many theory-based papers also apply their method to a benchmark example. 34 of the 139 considered papers do not include any application, the rest can be broken down as shown in Figure 4.

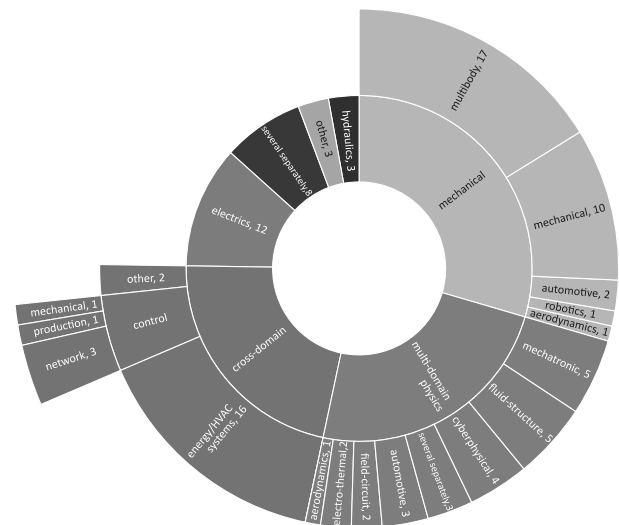


Figure 4: Breakdown of application areas found in the selected literature.

We can observe that the considered applications are mostly physical and almost all cover at least a physical aspect in one partial system. Applications in mechanics constitute the largest share with 30%. Within these, multibody simulations are most common, followed by general mechanical applications. 24% of all papers where an application is found combine two or more physical domains in various manifestations. While applications defined as “cross-domain” (22% of all 105) often also involve multiple physical domains, some non-physical model part is also included – e.g. control strategies or logistics. Summarized as “other” are specific isolated applications such as molecular dynamics or a HIL implementation. The lower-level category “several separately” describes publications where a concept is applied to several examples in different areas separately, not combining them. The sub-category of “multi-domain physics” that is also named “several

separately” likewise comprises several applications in one work but in this case, every separate example couples different domains.

In recent years, purely electrical applications have vanished while cross-domain applications have considerably increased.

5 Model descriptions in the considered literature

Different mathematical model descriptions also require individual solution algorithms and further coupling approaches. In 129 papers, one or more model descriptions out of *ordinary differential equations* (“ODE”), *differential algebraic equations* (“DAE”), *partial differential equations* (“PDE”), *agent based models* (“ABM”), *system dynamics* (“SD”), *synchronous data flow* (“SDF”), *finite element models* (“FEM”), *boundary element models* (“BEM”) and *discrete event systems* (“DE”) can be identified¹. ODEs and DAEs are by far the most frequently considered descriptions, as depicted in Figure 5.

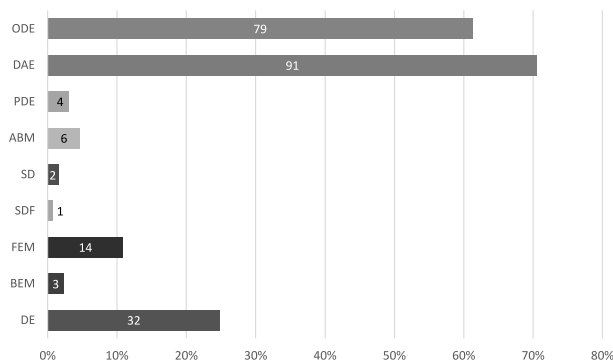


Figure 5: Amount of publications utilizing different kinds of model description.

Over half of the 129 publications (70) cover more than one model description. Thereof, 23 are exclusively dealing with ODEs and DAEs. While ODEs have always been considered in similar measure throughout the “history” told by this selection in research on co-simulation, DAEs have become a more popular topic of interest after the millennium.

33 papers explicitly cover hybrid co-simulation – in the sense of coupling continuous time models with DE systems. Research in this area has increased drastically

¹The two publications on standards have been excepted here since these are not restricted to one or few types of model descriptions.

over time: while none are found in the considered selection before the millennium, nine have been published from 2010-2014 and 21 after 2014. This correlates with the increased occurrence of DE systems.

6 Distinction of algorithms

One major point to be considered when categorizing co-simulation methods are the numerical approaches that come with the nature of the topic. In that respect, the focus can lie on the solution algorithms used by the participating subsystems or the coupling algorithm itself. Prior to the specific algorithms, however, different concepts to approach the intended cooperation can be discerned.

6.1 Distinction by coordination concept

Regarding the general concept of coupling coordination, we can distinguish whether or not an external orchestrator is used to organize the co-simulation. Co-simulations where communication is orchestrated *outside* one of the participating subsystems are classified as using an orchestrator. Without an external orchestrator, data exchange can be handled in one of the subsystems themselves, thus acting as the so-called *master* of one or more other subsystems (*minions*). Note that there are overlaps: If a master or orchestrator coordinates several subsystems while solving an equation, it may be considered as both. Whether or not an external orchestrator is used is defined for 103 papers. A majority (81%) of publications present approaches using an orchestrator. Only 18 publications (17%) do not use an orchestrator, 2 consider both variants. While this trend is also observable in most time frames, publications before the turn of the millennium stand out: within these, more than half do not use an external orchestrator.

6.2 Distinction by interfaces

Although slightly different interpretations for these terms are circulating the literature (cf.[1]), we adopt the term *distinction on “interface level”* from [5] for the division of coupling approaches into *strong* and *loose coupling* methods. *Strong coupling* allows different solvers but requires the same time steps in all subsystems, permanent exchange of coupling data and iteration in every time step while *weak* or *loose coupling* allows different, individual time steps in the partial systems. Whether loose or strong coupling methods are

applied can be determined for 115 of the 139 publications. An overwhelming majority of 87% focuses solely on loose coupling algorithms, nine papers (8%) consider only strong coupling methods and five percent cover both approaches. This general observation of a loose coupling dominance is also reflected in most time frames, although strong coupling slightly gains in popularity up to 2009, only to drop again afterwards.

6.3 Distinction by execution sequence

Regarding the sequence in which participating subsimulations are executed, we differ between *parallel* and *sequential* methods. Most commonly, parallel (also called *Jacobi* type) methods are applied. This does not necessarily require de facto computational parallelization on multiple cores, but means that every time data is exchanged, simulation time is the same in every subsystem, so no part can obtain future information on any other system. In sequential (*Gauß-Seidl* type) methods, values at a (in general, one) reference ahead can be used for more accurate approximations in “slower” subsystems. The sequence of execution is defined in 103 of the considered publications.

Over half (55%) of these only investigate or apply parallel methods, almost a quarter exclusively focus on sequential ones and the rest discuss both approaches. The share of (purely) parallel methods gradually increases over time, peaking at 70% in the last five years. This might partly be explained by simultaneous progress in computational parallelization, which can be utilized to speed up coupled simulations, and partly by reasons of implementational simplicity.

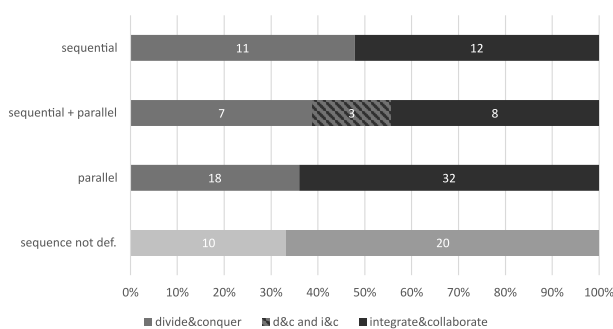


Figure 6: Nexus of sequence of execution and perspective of (de-)coupling.

Figure 6 illustrates the correlation between the sequence of subsystem execution and the perspective concerning the state of development in which the system

is partitioned and/or coupled. It is interesting to note that all three publications in which both an integrate-and-collaborate and a divide-and-conquer approach are considered also investigate both parallel and sequential methods. Publications on sequential algorithms show the highest share of those approaching their research question by a divide-and-conquer perspective. Neither property can be made out for six papers. Twelve define the sequence but not the state of development in which the (de-)coupling takes place. The thirty papers with unknown sequence but defined (de-)coupling perspective are included in the statistics, see the lighter colored bar in Figure 6.

6.4 Distinction by iterations

Depending on whether or not the coupling algorithm iterates over its macro steps, we differ between iterative (*implicit*) and non-iterative (*explicit*) approaches with or without a predictor-corrector step. The latter are also called *semi-implicit* as they allow step rejection and thus require rollback even if no iteration as such takes place. Whether the coupling algorithm uses iterations or not can be determined for 112 of the considered publications, whereat this rate has in general – except for the interval 2010-2014 – decreased over time.

In two thirds (74), only non-iterative master algorithms are used. This emphasizes the importance of the development of explicit methods, as – even though implicit approaches surpass explicit ones with respect to accuracy and stability – a majority adheres to explicit methods, be it due to performance reasons, software requirements, or simplicity. Eleven (15%) of the papers using non-iterative methods (amounting to 10% of all 112) apply a predictor-corrector method. A quarter (28) of the total are exclusively using iteration, of which one also applies a predictor-corrector method. Both iterative and non-iterative algorithms are considered in ten papers (8%), predictor-corrector steps in half of these. Figure 7 illustrates that while the research on iterative methods has increased in the years before 2010, it has diminished again in recent years. While this could be interpreted as preference of explicit methods in praxis (as 2000 to 2009 are also the years peaking in a focus on theory), this correlation is disproved by the cross-connection shown in Figure 8. One expectable observation to be made from there is that iterations are not defined in most surveys. Further, in most frameworks, non-iterative methods are applied or it is not defined whether or not iterations (can) take place. Taking a

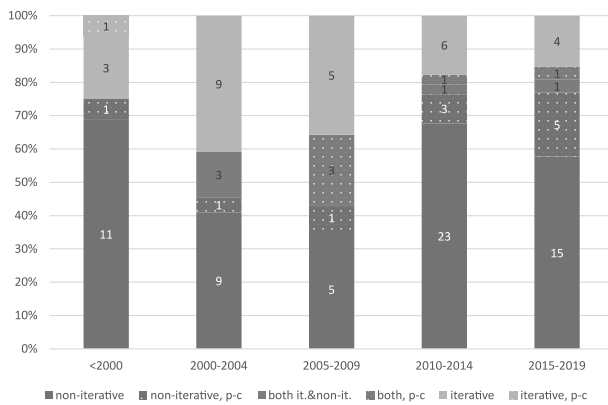


Figure 7: Variation of shares of iterative and non-iterative methods over the years.

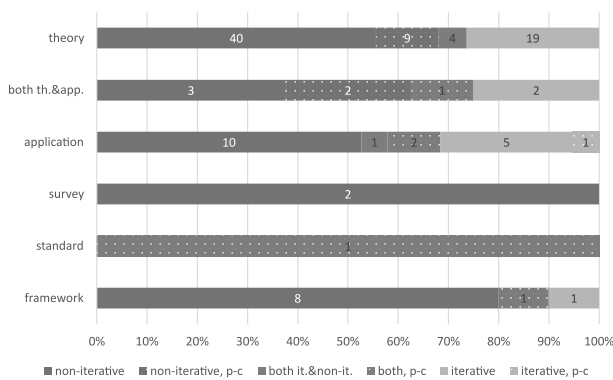


Figure 8: Connection between the main topic of publications and iterations in the master algorithm.

look at the nexus of the sequence of execution and the iterations of master algorithm steps (Figure 9), it is interesting to note that the category of publications exclusively using parallel methods also shows the highest share of non-iterative master algorithms. This implies that those using methods that are more prone to stability issues also do not aim to increase accuracy by iteration. *Both* iterative and non-iterative methods are only considered in publications which also use sequential algorithms (exclusively or in a comparison to parallel ones). There are 23 publications where neither sequence nor iterations are defined and four that define the sequence but do not reveal whether an iterative master algorithm is used.

6.5 Distinction by macro steps

Algorithms with a fixed macro step require all participating simulations to exchange data at previously

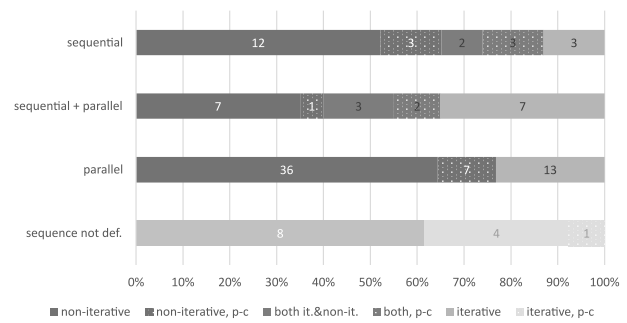


Figure 9: Nexus of sequence of execution and iterations of the master algorithm.

defined synchronization references in – generally – equidistant intervals. Other solutions adapt the macro step in the course of a simulation via step size control, either by step rejection and repetition with a smaller step size if certain tolerances are violated (see f.i. [6]) or by adaption before every macro step execution by predictive error estimation, f.i. via extrapolation or energy residuals [7, 8]. Some loose coupling methods do not necessarily require any synchronized time steps from the sub-simulations apart from the start time, see f.i. [9, 10]. Whether a fixed, adaptive, or individual step size without any simultaneous steps is used is defined for 100 publications. A clear majority of 72% is applying or developing a method with a fixed communication time step. Nevertheless, 35% are using some kind of macro step size control and four present methods allowing no common synchronization reference. There are overlaps, which are shown in detail in Figure 10. While from 2005 on, the amount of publications using fixed or adaptive macro steps does not vary substantially, an exceptional share of those in the category of fixed step sizes stands out in the time frame from 2000-2004. On the other hand, while in all other intervals, methods with fixed macro steps clearly dominate those with adaptive ones, shares are almost equal before 2000. All four works presenting master algorithms that do not require the subsystem solvers to have any step in common are found in the time frame from 2010-2014.

The connection between the kind of master step and utilization of iteration is shown in Figure 11. Not altogether surprisingly, the highest share of adaptive algorithms is found in publications covering non-iterative methods. Apart from the one standard theoretically allowing all three kinds of master steps and iterations (note that the distinction on iterations is not exclusive!), methods that do not require a common synchronization

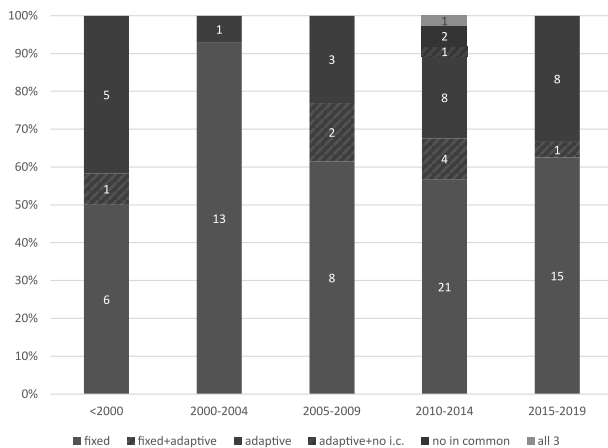


Figure 10: Change in percentages of fixed, adaptive or no common macro step usage in the selected literature over five-year time frames.

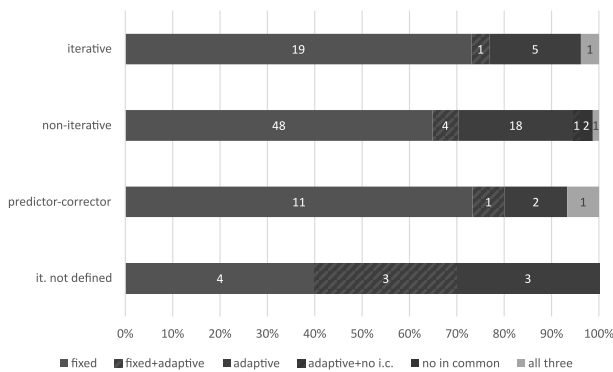


Figure 11: Connection between macro steps and iterations in the considered literature.

step are always non-iterative. In 17 publications, neither if iterations take place nor whether fixed, adaptive, or completely independent step sizes are used has been revealed. 22 did not specify the macro steps, but did clarify whether iterative methods have been used.

7 Distinction by the number of coupled subsystems

While partitions due to latency or activity are sometimes limited to dividing the overall system into two subsystems, further variations in time constants[11] or problems of the integrate-and-collaborate kind [12] often require the cooperation of several or even an arbitrary number of subsystems. The number of co-simulated systems has been defined in all but eight pub-

lications, all of which are surveys or formalism descriptions. More than half (56%) of these publications consider exactly two coupled subsystems. Of the remaining 58 papers, 22 are coupling more than two, yet an explicit, integer number of systems. 27% describe methods that can be applied to an arbitrary number of subsystems. Publications that present a theoretical approach for n coupled subsystems which have then been tested only on two have also been classified as allowing an arbitrary number of coupled systems.

Figure 12 shows how the different main orientations are partitioned by the number of considered subsystems.

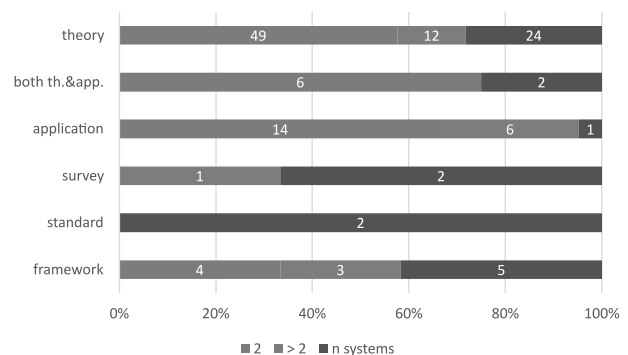


Figure 12: Papers' main emphases partitioned by different numbers of coupled subsystems.

It is not surprising that there are barely publications that focus on an application and still consider an arbitrary number of coupled subsystems. However, all mainly theoretical and applied work is composed predominantly of publications coupling only two systems. Surveys and standards almost always allow an arbitrary number.

The cross-connection of the number of coupled systems and their sequence of execution reveals that the considered literature barely covers investigations of sequential methods for an arbitrary number of systems. This may be explained by the impact of the order of execution in sequential approaches, which gains in possibilities and thus complexity with the number of systems.

7.1 Hierarchical approaches

Furthermore, we characterize whether a hierarchical approach is considered in the literature. The construction of a non-trivial hierarchy, which can increase numerical stability properties [13], is only possible for three or more systems, thus already restricting the selection. Only nine publications decidedly allow hierarchy, four

of which describe multirate methods where the integration algorithm itself is partitioned. This illuminates a general lack of existing studies and thus present potential in this area of research.

8 Conclusion

The analysis of the regarded literature has unveiled tendencies towards more popular methods. While most publications cover mainly theoretical aspects, applications are nevertheless manifold and range from mostly physical systems in one or many domains to cross-domain applications including complex controlled systems up to urban scale. Model descriptions are dominated by Differential (Algebraic) Equations but also cover AB or FE models and even DE systems. Hybrid systems, albeit sparsely represented, remain a challenge if approached via coupled simulations as well as they do in a mono-simulation. Non-iterative, parallel loose coupling methods are applied predominantly, even though iterative and sequential approaches entail higher accuracy and better stability properties. Similarly, fixed macro steps are more frequently used than adaptive algorithms, which, as well as the preference to avoid roll-back, may be explained by the implementational limitations of commonly known software tools that support co-simulation. The presented analysis has highlighted a lack of research of hierarchical co-simulation, which is addressed in [2].

References

- [1] Hafner I, Popper N. On the Terminology and Structuring of Co-simulation Methods. In: *Proceedings of the 8th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, EOOLT '17. Weßling, Germany: ACM. 2017; pp. 67–76. doi: 10.1145/3158191.3158203.
- [2] Hafner I. Cooperative and Multirate Simulation: Analysis, Classification and New Hierarchical Approaches. phd thesis, TU Wien, Vienna, Austria. 2021. doi: 10.34726/hss.2021.37460.
- [3] Dahmann JS, Fujimoto RM, Weatherly RM. The Department of Defense High Level Architecture. In: *Proceedings of the 29th conference on Winter simulation - WSC '97*. Atlanta, Georgia, United States: ACM Press. 1997; pp. 142–149.
- [4] Blockwitz T, Otter M, Akesson J, Arnold M, Clauss C, Elmqvist H, Friedrich M, Junghanns A, Mauss J, Neumerkel D, Olsson H, Viel A. Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. 2012; pp. 173–184. doi: 10.3384/ecp12076173.
- [5] Busch M. *Zur effizienten Kopplung von Simulationsprogrammen*. Kassel University Press. 2012.
- [6] Benedikt M, Stippel H, Watzenig D. An Adaptive Coupling Methodology for Fast Time-Domain Distributed Heterogeneous Co-Simulation. In: *SAE 2010 World Congress & Exhibition*. 2010; doi: 10.4271/2010-01-0649.
- [7] Zhang H, Liang S, Song S, Wang H. Truncation error calculation based on Richardson extrapolation for variable-step collaborative simulation. *Science China Information Sciences*. 2011;54(6):1238–1250. doi: 10.1007/s11432-011-4274-z.
- [8] Sadjina S, Kyllingstad LT, Skjong S, Pedersen E. Energy conservation and power bonds in co-simulations: non-iterative adaptive step size control and error estimation. *Engineering with Computers*. 2017;33(3):607–620.
- [9] González F, Naya MA, Luaces A, González M. On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics. *Multibody System Dynamics*. 2011;25(4):461–483. doi: 10.1007/s11044-010-9234-7.
- [10] Liang S, Zhang H, Wang H. Combinative Algorithms for the Multidisciplinary Collaborative Simulation of Complex Mechatronic Products Based on Major Step and Convergent Integration Step. *Chinese Journal of Mechanical Engineering*. 2011;24(03):355. doi: 10.3901/CJME.2011.03.355.
- [11] Striebel M. Hierarchical Mixed Multirating for Distributed Integration of DAE Network Equations in Chip Design. Ph.D. thesis, Bergische Universität Wuppertal. 2006.
- [12] Galtier V, Vialle S, Dad C, Tavella JP, Lam-Yee-Mui JP, Plessis G. FMI-based distributed multi-simulation with DACCOSIM. In: *Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, DEVS '15. San Diego, CA, USA: Society for Computer Simulation International. 2015; pp. 39–46.
- [13] Hafner I, Popper N. Investigation on Stability Properties of Hierarchical Co-Simulation. In: *Proceedings ASIM SST 2020*, 25. Symposium Simulationstechnik, vol. 59 of ARGESIM Report. Online-Tagung: ARGESIM Verlag. 2020; pp. 41–48. doi: 10.11128/arep.59.a59007.

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

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

Jörg Baumgartner²

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

²Fraunhofer Institut for Structural Durability and System Reliability LBF, Bartningstraße 47, 64289 Darmstadt, Germany

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

Introduction

Designer of castings components increasingly using simulation tools. There are typically three steps involved: (1) The design engineer creates a geometry based on the requirements and calculates the stresses resulting from the applied static operating loads and inertia forces by means of an FEM simulation using a structural analysis. In an optimization loop, component optimization can be performed in which the stresses are minimized under certain restrictions. (2) The casting process simulation is preformed, based on the 3D CAD model. There, the complete casting process from the pouring process to the solidification of the melt is analyzed in the form of a CFD simulation and the casting system (feeder, material allowance, etc.) is defined. (3) Based on the CAD data and, if necessary, the local material properties, the fatigue life is assessed and potential local weak spots in the component are identified. Until a final design is achieved, sub-steps have to be run through several times.

The use of simulation programs for the casting process, structural and fatigue analysis is currently still car-

ried out independently. An exchange of all relevant data takes not place when several calculation tools does are used, also due to the lack of software interfaces. A holistic view of the simulation data is missing.

Cast iron components are frequently exposed to high mechanical loads. To ensure safe operation of plant and machinery, the fatigue life of the component must be guaranteed under the assumed operating conditions over the planned service life. New or expanded areas of application and increased safety requirements increase the demands on detailed service life determination. From this point of view, the following question arises: What contribution can make the casting process simulation and structural analysis to the fatigue life calculation?

In addition to simulation-based fatigue analysis, the component's fatigue life can also be determined experimentally, although the corresponding tests are time-consuming and expensive. They also require extensive laboratory equipment. In addition, component tests of this magnitude are hardly feasible experimentally. Therefore, this route is only taken in a few cases, and experimental validation of simulation results is also too costly in most cases. In the present work, an attempt is made for the first time to take into account the experimental data determined at great expense in the simulation-based fatigue analysis. These data were determined on samples taken from large cast components.

1 Simulative Determination of Local Fatigue Limits

The following chapter presents two different approaches from a methodological point of view. The

first approach describes the calculation of fatigue limits based on a homogeneous material definition, i.e., the material definition is identical for the complete component. Locally varying material parameters are not considered. The second approach describes the integration of data from the casting process simulation into the fatigue analysis. Here, the definition of the local material properties is performed on a shared FE mesh. Both approaches have already been mentioned in [1, 2, 3]. In this paper, the data-technical realization is described in detail on the basis of an example.

1.1 Procedure with Global Material Data

FEMFAT is the world's leading solver for FE-based fatigue analysis and calculates structural durability of statically and dynamically loaded components on the basis of FE calculation results. FEMFAT as FE post-processor requires not only the data from the structural analysis (FE mesh and loads) but also the material data (strength values). Figure 1 shows the typical procedure for the calculation of local fatigue limits in FEMFAT. The example of a large casting of a wind turbine below explains the procedure in more detail. The *poll end* or *canister* (cast iron box on the end of the windshaft through which the sail stocks pass) of a wind turbine has to withstand high cyclic loads. It is therefore extremely important to ensure fatigue limits in areas of high stress.

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

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

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

A material class is selected from the material

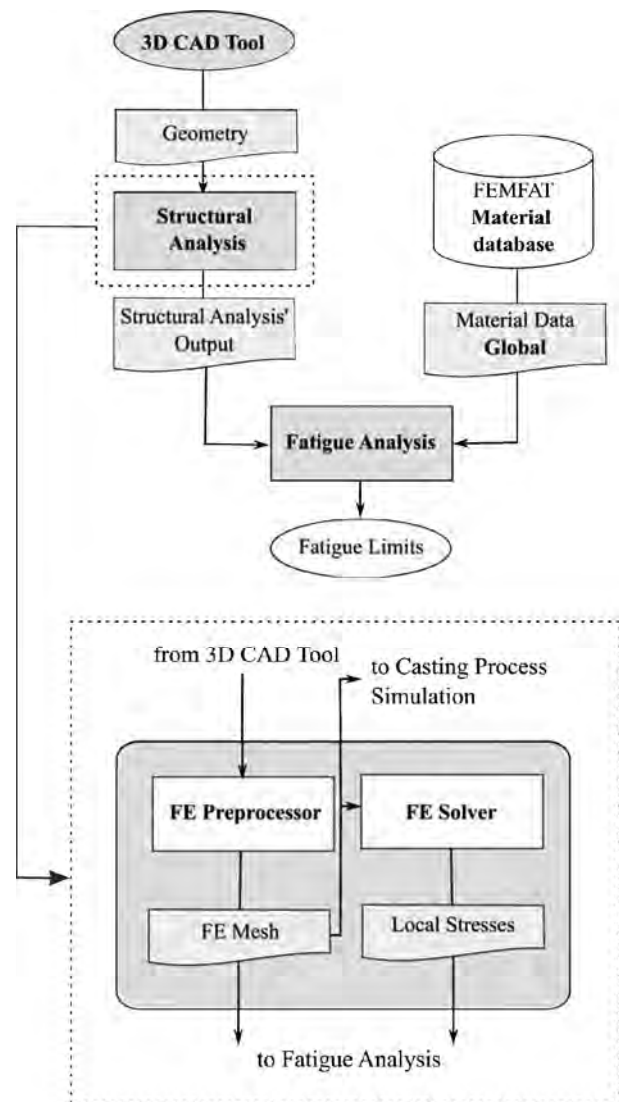


Figure 1: Calculation of fatigue limits with global material data

database. Based on preprogrammed ratios, the following missing material parameters are added to a predefined tensile strength for the calculation are automatically generated:

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

At the FE node, local fatigue limits are calculated from material parameters, which are additionally influenced by local component properties (e.g. notch effect) and loads. The basic procedure for calculating

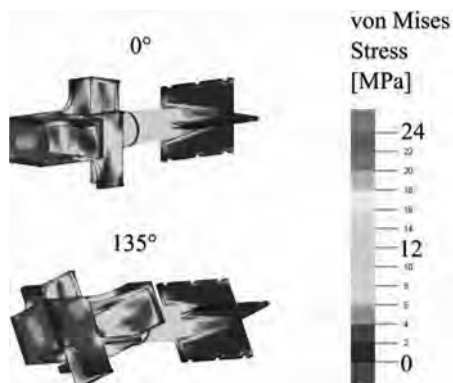


Figure 2: Simulated distribution of mechanical stresses at two different angular positions

fatigue limits is based on the influencing factors with which the fatigue strength is increased or decreased. The FKM (Forschungskuratorium Maschinenbau) standard describes these factors [4].

1.2 Integration of a Casting Process Simulation

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

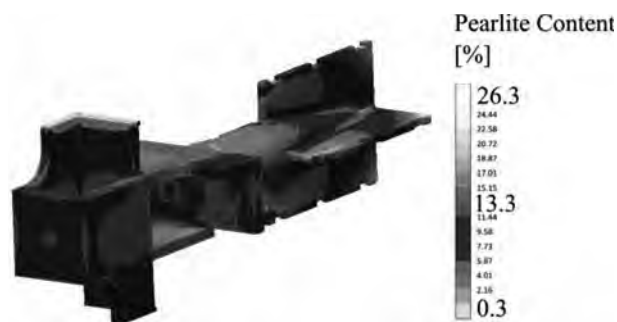


Figure 3: Simulated microstructure fraction in the component at room temperature

With regard to requirements on the accuracy of today's simulations in the casting industry, local material differences must be taken into account in the fatigue analysis. One way to deal with this is to use the local material data from the casting process simulation. From the

microstructure, the casting process simulation can determine local material values (tensile strength, Young's modulus, yield strength, elongation at break) in the casting in the next step (Figure 4).

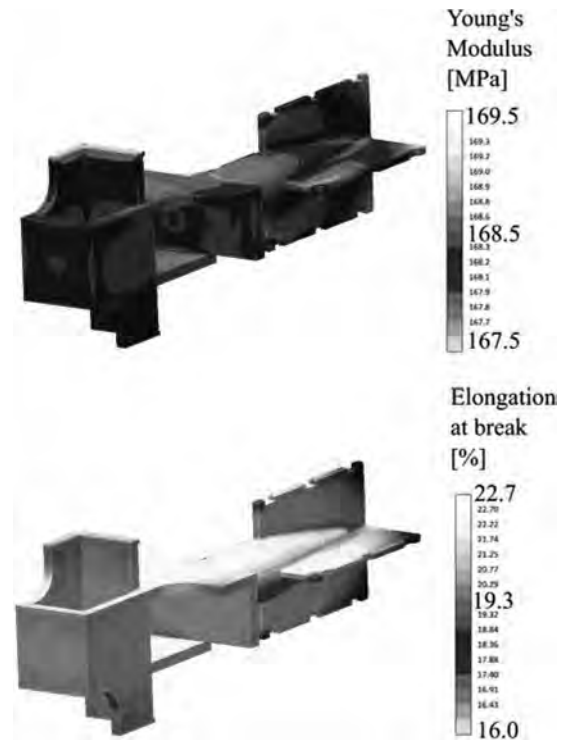


Figure 4: Simulated local material properties in the component at room temperature

Figure 5 shows the schematic description of the approach for calculating the local fatigue limits with integration of the local material data from the casting process simulation

For the calculation of the local material data of the poll end, the casting process simulation was carried out in the commercial software package MAGMAsoft 5.4.

The discretization for the numerical solving algorithm in the casting process simulation on the one hand and in the fatigue/structural analysis on the other hand is different: in MAGMAsoft the discretization is based on the finite volume method (FVM) and in the fatigue/structural analysis on the finite element method (FEM). There is a need for mapping the results from the casting process simulation to the FE mesh generated in the structural analysis. The mapping is realized by MAGMALink, a module of the casting process simulation. This makes the results of the casting process simulation usable for further processing in the fatigue

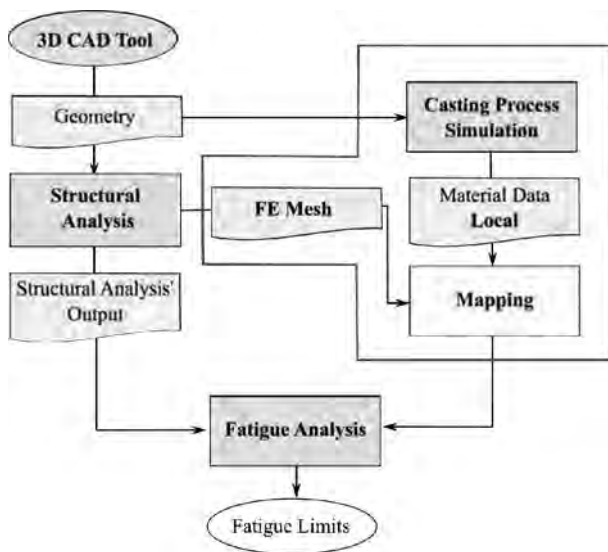


Figure 5: Calculation of fatigue limits with local material data from casting process simulation

analysis.

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

2 Consideration of Experimental Data

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

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

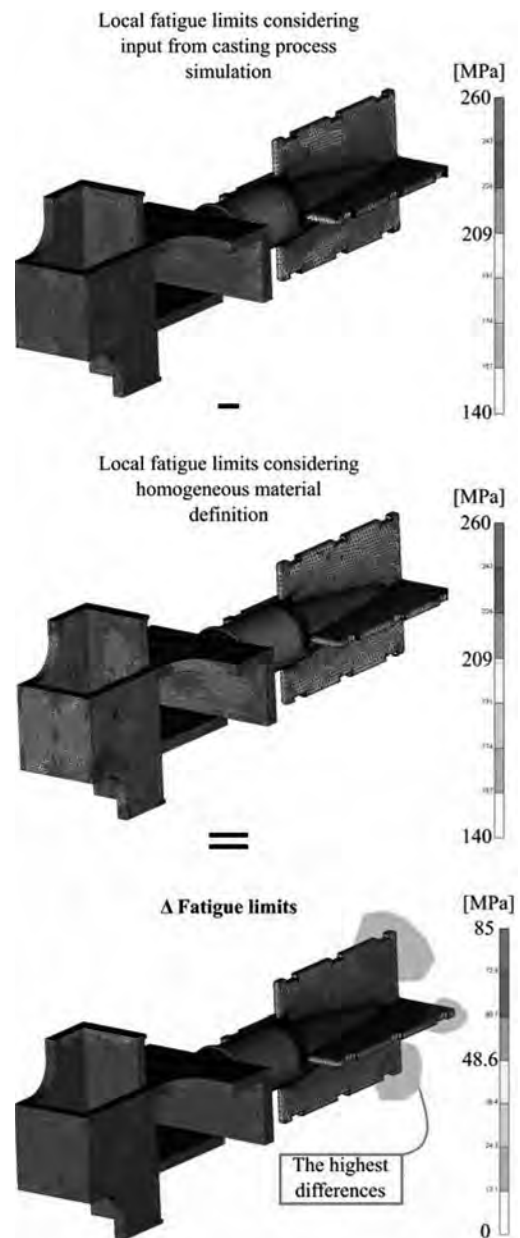


Figure 6: Fatigue limits with homogeneous material data and taking into account data from casting process simulation

the casting process simulation and is consequently not considered further.

From the casting process simulation, at this point not the local mechanical properties are required, as in the purely numerical approach, but the local microstructure. If experimental data on the microstructure and the associated fatigue limit are now available in a material

database, the microstructure resulting from the casting process simulation can be used to generate local fatigue limits (Figure 7).

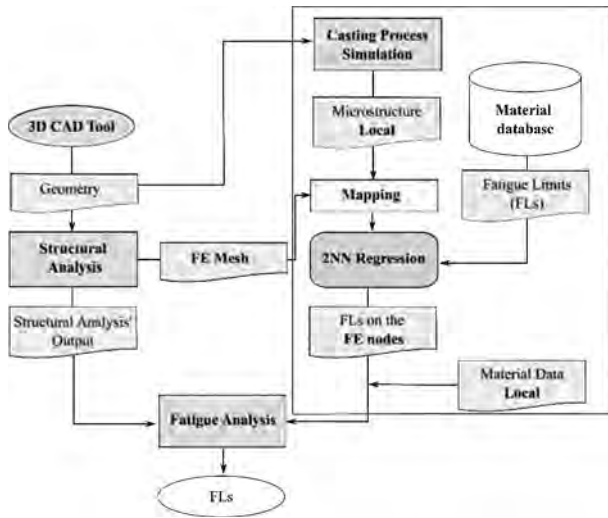


Figure 7: Calculation of local fatigue limits using experimental data

Since the experimental determination of microstructure/fatigue limits data involves a great deal of effort, only a few data sets are available. In order to determine the associated fatigue limits for all microstructure data from the casting process simulation, these must be determined approximately on the basis of the available experimental data.

Using the k-nearest neighbor algorithm (kNN) as a regression method [10], an individual fatigue limit can be generated for each FE node (Figure 8). Here, the value at a FE node is weighted by the distances d of the 2-nearest neighbors proportional to their distance. In the present work the regression method is implemented in form of the in-house developed MATLAB code. The fatigue limit $FL(x)$ for a node x from the experimentally determined fatigue limits of the two nearest neighbors x_1 and x_2 (both come nodewise from casting process simulation) is determined according to equation 1:

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

In the equation 1 the $d(x, x_1)$ and $d(x, x_2)$ represent the distances from the FE node to the next two neighbours (experimental data) in two-dimensional space that con-

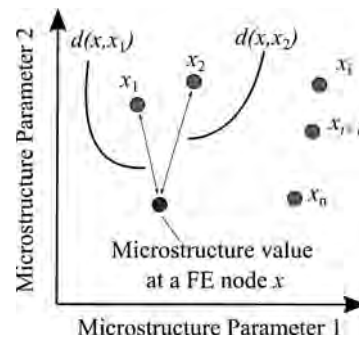


Figure 8: 2NN regression of fatigue limits from microstructure values

sists of two positive real numbers, the values for microstructure parameter 1 (x-axis) and microstructure parameter 2 (y-axis).

Finally, the obtained fatigue limits are modified via the FEMFAT internal algorithm using the stresses from the structural analysis. This allows a comparison of the fatigue limits from the three approaches described. Compared with the calculation using the local material properties from the casting simulation, the fatigue limits with the regression from the microstructure data increase again by between 10 % and 20 %. Figure 9 shows the areas with the highest differences between the approach based on experimental data and the approach based on the local material properties. The differences are located at the blade bearings and contact vials to the shaft.

Conclusion and Outlook

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

The purposeful coupling of not connected software tools in the foundry industry explains the complexity of the study for numerical determination of local fatigue limits. In order to combine the different software modules, a number of interfaces are required. Table 1 shows the formats of the respective interfaces used in the three

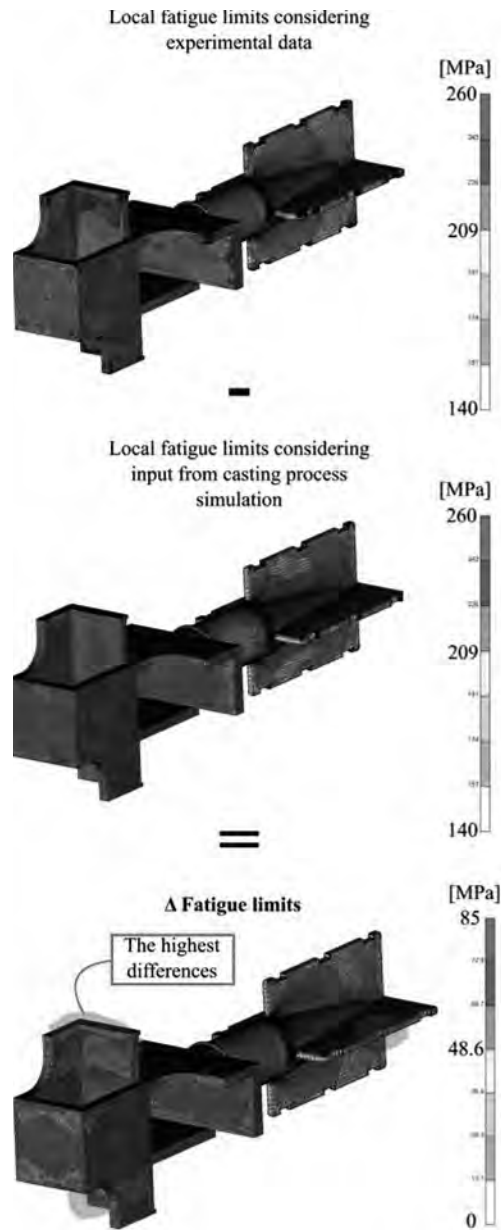


Figure 9: Fatigue limits considering data from casting process simulation and considering experimental data

approaches for the calculation of fatigue limits.

The integration of experimental data into the numerical process chain could be realized by a 2NN regression. Since the acquisition of experimental data on fatigue limits is very time-consuming, the algorithm is currently based on very little data. The incorporation of more experimental data will improve the prediction ac-

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

Table 1: Overview of the used file formats

curacy. These should also cover significantly more microstructure classes with the associated fatigue limits. An evaluation of the method presented here is only possible if the fatigue limit of a component is determined both experimentally and numerically.

Acknowledgement

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

References

- [1] Chen X, Engler-Pinto C, King M, Li M, Prabhu E, Su X. Aluminum Cylinder Head High Cycle Fatigue Durability Including the Effects of Manufacturing Processes. 2012; pp. 2012-01-0540. URL <https://www.sae.org/content/2012-01-0540/>
- [2] Gaier C, Eberhard D, Aichberger W, Dannbauer H. Betriebsfestigkeitsanalyse mit Berücksichtigung von

Ergebnissen aus der Prozesssimulation. Wiesbaden, Germany; .

Advanced Theory and Simulations. 2020;3(7):2000031.
URL <https://onlinelibrary.wiley.com/doi/10.1002/adts.202000031>

- [3] Puchner K, Aigner R, Rinnergschwentner L, Dannbauer H. Consideration of local material characteristics for the fatigue assessment of cast component. In: *Proceedings of the 1st Congress for intelligent Combining of Design, Casting, Computer Simulation, Checking and Cyclic Behaviour for efficient Cast Components*. Darmstadt; pp. 155–164.
- [4] Rennert R. *Rechnerischer Festigkeitsnachweis für Maschinenbauteile aus Stahl, Eisenguss- und Aluminiumwerkstoffen*. FKM-Richtlinie. Frankfurt am Main: VDMA-Verl, 6th ed. 2012.
- [5] Mercier JP, Zambelli G, Kurz W. Microstructures. In: *Introduction to Materials Science*, pp. 239–259. Elsevier. 2002;.
URL <https://linkinghub.elsevier.com/retrieve/pii/C20090291483>
- [6] Olofsson J. Local microstructure-based material performance and damage in design and finite element simulations of cast components. *Journal of Computational Design and Engineering*. 2018; 5(4):419–426.
URL <https://academic.oup.com/jcde/article/5/4/419/5728982>
- [7] Thomser C, Bodenburg M, Sturm JC. Optimized Durability Prediction of Cast Iron Based on Local Microstructure. *International Journal of Metalcasting*. 2017;11(2):207–215.
URL <http://link.springer.com/10.1007/s40962-016-0091-x>
- [8] Systemzuverlässigkeit FIFBU. Maßgeschneiderte Bauteileigenschaften durch Integration von Fertigungs- und Funktionssimulation : BMBF-Verbundprojekt MABIFF ; Abschlussbericht des Verbundprojektes. *Tech. rep.*, Technische Informationsbibliothek u. Universitätsbibliothek. 2011. Artwork Size: Online Ressource (103 S., 8,63 MB) Medium: application/pdf Version Number: 1.0.
URL <https://www.tib.eu/suchen/id/TIBKAT:68815087X/>
- [9] Hack M, Jung D, Egner-Walter A. Optimierter Lebensdauer-Berechnungsprozess von Eisengussbauteilen unter Berücksichtigung des lokalen Gefüges*. *Materials Testing*. 2012;54(7-8):497–502.
URL <https://www.degruyter.com/document/doi/10.3139/120.110356/html>
- [10] Zhang Z, Gu GX. Finite-Element-Based Deep-Learning Model for Deformation Behavior of Digital Materials.

Hochaufgelöste Energieprofile durch hybride Simulation

Benjamin Wörrlein, Steffen Straßburger

Fachgebiet Informationstechnik in Produktion und Logistik, Technische Universität Ilmenau, Max-Planck-Ring 12, 98693 Ilmenau, Deutschland; benjamin.woerrlein@tu-ilmenau.de; steffen.strassburger@tu-ilmenau.de;

Abstract. The price of a commodity, as electricity, is determined on a commodity market. A market is efficient when the supply and demand in the market are at an equilibrium. Efficient markets run on information. Information can cause a spontaneous and instantaneous change within the supply and demand in a market. The market communicates this new equilibrium through the change of the price of a commodity. In the electricity market the supplier and consumer communicate through electrical load profiles. A load profile signals when and how much energy should be consumed within a certain time frame without causing a change in the price of electricity. Creating such load profiles is commonly done by the supplier of energy by means of standard load profiles. Here we propose a data-driven simulation-based method that allows for the consumer to create its own specific load profile, which potentially will bring down the cost of energy consumed.

Einführung

Lastprofile beschreiben den Energiebedarf eines Systems über einen bestimmten Zeitraum. Energieerzeuger bedienen sich Lastprofilen, um einem Konsumenten mitzuteilen, wann wieviel Energie abgerufen werden kann. Weicht der tatsächliche Energiebedarf eines Konsumenten von dem im Lastprofil beschriebenen Energieangebot ab, muss der Konsument entsprechend der Differenz ein erhöhtes Netzentgelt entrichten [1]. Standardlastprofile werden von Energieerzeugern für Abnehmer mit einem Jahresverbrauch von kleiner 100 MWh erstellt [2]. Für Abnehmer mit einem höheren Jahresverbrauch kann im Rahmen der registrierenden Leistungsmessung ein spezifisches Lastprofil für eine definierte Zeitperiode (täglich, wöchentlich, monatlich, ...) erstellt werden [2, 3].

Dieses spezifische Lastprofil wird auf Basis der in der registrierenden Leistungsmessung erhobenen Daten erstellt. Das erstellte Lastprofil soll nun möglichst nah das

Abrufverhalten von Energie eines Konsumenten beschreiben. Anhand eines zwischen Erzeuger und Konsument vereinbarten Lastprofils wird nun ein Lieferabkommen geschlossen. Weicht der Konsument in seinem Abrufverhalten vom vereinbarten Lastprofil ab, erklärt sich der Erzeuger weiter bereit ihm Energie zu liefern, diese ist aber in Abhängigkeit der erhöhten Volatilität höher bepreist. Dieses Vorgehen hat mehrere Nachteile:

1. Der Erzeuger hat hier einen Informationsvorsprung vor dem Konsumenten. Durch das Vorhandensein mehrerer Abnehmer, fällt diesem in der Regel die Erstellung des Lastprofils zu. Der Erzeuger seinerseits hat nun das Interesse die Lastprofile der Konsumenten möglichst an dem für ihn optimalen Abrufverhalten und nicht dem der Konsumenten zu orientieren.
2. Spezifische Lastprofile werden für definierte Perioden erstellt. Das optimale Abrufverhalten des Verbrauchers kann nun aber eine so starke Volatilität aufweisen, dass dieses nicht in einem periodischen Lastprofil abgebildet werden kann. Hier kann für den Konsumenten ein Optimierungsproblem der Zielgrößen Energiebezugskosten und bspw. der Gesamtanlageneffektivität (*Overall Equipment Effectiveness – OEE*) entstehen.
3. Der Konsument hat nur begrenzte Möglichkeiten auf die vor Abruf vereinbarten Lastprofile Einfluss zu nehmen. Dies liegt vorrangig daran, dass er im Falle von erwarteter starker Volatilität diese nicht beschreiben kann. Volatilität geht mit einer verringerten Prognosegüte in Abhängigkeit zum betrachteten Zeitraum einher.

Wir schlagen hier eine datengetriebene Simulationsmethode vor, um den oben beschriebenen Nachteilen entgegenzuwirken. Basis dieser Methode ist eine Hybride Simulation (HS) nach [4], welche eine diskret-ereignisorientierte Simulation (DES) mit einem Modell des Maschinellen Lernens (ML) verbindet. Bei dem ML-Modell handelt es sich hierbei um ein sog. Sequence-to-Se-

quence-Modell (Seq2Seq) [5, 6]. Seq2Seq-Modelle lernen zwei Sequenzen aufeinander abzubilden und werden standardmäßig für Übersetzungen verwendet. Die HS soll für eine mechanische spanende Fertigung mit mehreren parallelen CNC-Maschinen angewendet werden. Auf jeder Fertigungslinie soll anhand eines Produktionsplans eine Folge von Fertigungsaufträgen (FA) abgearbeitet werden. Der Produktionsplan enthält lediglich den Startzeitpunkt eines FA und eine Beschreibung des FA, wie den NC-Code, den Fertigungsmodus (Schruppen oder Schlichten) etc. Dieser Ansatz wurde schon für einen einzelnen FA vorgeschlagen [7], validiert und dessen Einbindung in DES diskutiert [8]. Der hier vorgeschlagene Ansatz soll sich nun mit der Erstellung von Lastprofilen für eine gesamte Fertigung beschäftigen.

Ein Seq2Seq-Modell erlernt den Zusammenhang zwischen der Beschreibung eines FA und dessen Zeitreihe des Energiebedarfs. Belegt nun in der DES der FA eine Fertigungslinie wird das trainierte Seq2Seq-Modell geladen und mit den Beschreibungen des FA aktiviert. Das Seq2Seq-Modell gibt nun eine hochaufgelöste Zeitreihe des Energiebedarfs des FA in Abhängigkeit seiner Beschreibung zum Aktivierungspunkt innerhalb der Simulation wieder. Somit kann für jede Fertigungslinie ein eigenes spezifisches Lastprofil erzeugt werden. Die Summe aller Fertigungslinien ergibt das spezifische Lastprofil des betrachteten Produktionssystems.

Ziel dieser Veröffentlichung ist es die Ergebnisse der oben beschriebenen HS vorzustellen und auf etwaige Schwächen und Stärken der beschriebenen Methode einzugehen. Die Validierung des Ansatzes geschieht anhand realer Daten eines Fertigungssystems für eine Fertigungsschicht.

1 Datengetriebene HS

1.1 Hybride Modelle zur Erzeugung von Lastprofilen

In einer klassischen DES können FA anhand ihres jeweiligen Startzeitpunktes im Produktionsplan abgebildet werden. Die Länge eines FA wird hier zuerst in Abhängigkeit zu einer aus den Daten ermittelten Verteilungsfunktionen bestimmt. Der eigentliche Energiebedarf wird anschließend meist statusbasiert über einen mittleren Leistungsbedarf bestimmt. Dieses Vorgehen kann nachteilig sein.

Erstens ändert sich ein Produktionsplan ständig in

Abhängigkeit zu den Anforderungen an das Produktionssystem und kann über eine Fertigungsschicht hinaus kaum vorherbestimmt werden. Er eignet sich also kaum zur Erstellung eines Lastprofils über längere Perioden.

Zweitens bilden die Verteilungsfunktionen der Länge und der statusbasierte Abruf des Energiebedarfs der einzelnen FA lediglich aus den Daten ermittelte statistische Größen, wie Mittelwert und Standardabweichung, ab. Dies ist einerseits nachteilig, da sie von einer starken Vergleichbarkeit der FA ausgehen und andererseits nur einen gemittelten Energiebedarf wiedergeben. Etwaige Lastspitzen, wiederkehrende Muster oder andere Charakteristika des Energiebedarfs eines FA gehen somit verloren, und stehen bei der Erstellung eines hochaufgelösten Lastprofils nicht mehr zur Verfügung.

Als ursächlich hierfür kann die grundsätzliche Limitation ereignisdiskreter Simulationsansätze angesehen werden, dass Zustandsänderungen zwischen zwei Ereignissen nicht abbildbar sind. Somit wird der Energiebedarf der Einfachheit halber zwischen den Start- und Endereignissen eines Auftrages als konstant betrachtet.

Um diese Limitation zu überwinden, bietet sich der Ansatz der hybriden Simulation an [4], der es ermöglicht, den ereignisdiskreten Modellteil um ein weiteres Modellierungsparadigma zu ergänzen, welches eine höheraufgelöste Abbildung des Energiebedarfs ermöglicht.

Eine mögliche Variante hierzu kann die kombinierte Simulation sein, also die Kombination eines ereignisdiskreten und eines kontinuierlichen Modellteils [9]. Wenn es die Natur des technischen Systems zulässt, kann dessen Energiebedarf über Differentialgleichungen modelliert werden [10]. Dies ist jedoch bei vielen Produktionsressourcen, wie z.B. CNC-Maschinen, nicht gegeben.

In derartigen Fällen greift man typischerweise auf hochaufgelöste gemessene Zeitreihen des Energiebedarfs zurück. Diese können direkt in der hybriden Simulation verwendet werden, indem sie z.B. über Tabellenfunktionen in Kombination mit dem System-Dynamics-Ansatz in die Simulation „eingespielt“ werden [11]. Über diesen Mechanismus lassen sich bereits Lastprofile ganzer Fertigungslinien erstellen.

Wesentliche Limitation dieses Ansatzes ist jedoch, dass nur bereits bekannte und gemessene Zusammenhänge wiedergegeben werden können. Eine Prognose des Energiebedarfs unbekannter Aufträge ist nicht möglich. Weiterhin ist keine Abbildung von Ursache-Wirkungszusammenhängen zwischen technologischen Steuerparametern (z.B. halbe Vorschubgeschwindigkeit, langsa-

mere Aufwärmphase) und dem resultierenden Energiebedarf möglich.

Eine Abhilfe hierfür könnte der in diesem Beitrag vorgestellte Ansatz zur hybriden Simulation darstellen, bei dem der ereignisdiskrete Modellteil um ein spezielles Modell des maschinellen Lernens ergänzt wird, das eine hochaufgelöste Zeitreihe des Energiebedarfs eines Auftrags auf Basis von dessen NC-Code prädiziert.

1.2 Rekurrente Netze und Encoder-Decoder Architekturen

Künstliche neuronale Netze (KNN) werden zur Identifikation von Mustern in komplexen Datenstrukturen verwendet. Ändern sich Muster über die Zeit, wird diese zeitliche Abfolge von Mustern als Sequenz verstanden. Damit ein KNN zeitliche Muster verarbeiten kann, müssen rekurrente Verbindungen in der Netztopologie vorhanden sein, welche eine Rückkopplung abstrahierten Wissens zulassen [12, 13]. Solche rückgekoppelten bzw. rekurrenten neuronalen Netze (RNN) eignen sich besonders für Daten, welche in sequentieller Form vorliegen [14].

Handelt es sich bei den Daten eines KNN um Sequenzen, werden diese als *Sequence to Sequence* (Seq2Seq) Architekturen bezeichnet. Durch die Aufnahmeschicht eines KNN findet eine Codierung der Eingangssequenz statt. Wird die Eingangssequenz in eine spezifische neuronale Schicht codiert, so ist dies ein Encoder. Wird eine Zielsequenz aus einer neuronalen Schicht heraus generiert, so wird dieser Teil einer Netzwerktopologie als Decoder bezeichnet [14].

Ist es Aufgabe eines Seq2Seq-Modells, Sequenzen unterschiedlicher Länge auf einander abzubilden, werden solche Strukturen allgemein als rekurrente Encoder-Decoder-Netzwerke (RNN-ED) bezeichnet [14]. Sollen Sequenzen unterschiedlicher Länge und unterschiedlicher Attribute aufeinander abgebildet werden, so müssen diese um eine zusätzliche Beschreibungsart, einen *Kontext* erweitert werden.

Der Kontext C ist der finale Zustandsvektor der verdeckten Schicht des Encoders und soll diesen auf den Decoder abbilden [14]. Der Vektor selbst lässt sich anhand einer verdeckten Schicht beschreiben [14]. So wird bei der RNN-ED-Architektur der Kontext C als ein Resultat der finalen verdeckten Schicht des Encoders mit der Eingangssequenz X_T , beschrieben. Da der Encoder in der Trainingsphase seinen finalen Zustand weitergibt, muss die ganze Sequenz X_T durchlaufen worden sein.

Weiterführende Erläuterungen zum hier verwendeten Encoder-Decoder können [5, 6, 14] entnommen werden.

2 Konzept

Ziel des Simulationsmodells ist es, für einen Fertigungsverbund von Maschinen (siehe Abb. 1) ein Gesamtlastprofil zu erzeugen. Die abgebildeten Maschinen sollen als Eingangsparameter lediglich Beschreibungen der Fertigungsaufträge erhalten, welche vor einer etwaigen Schicht verfügbar wären. Diese sind hier Startzeitpunkte, Betriebsmodi und NC-Code der FA auf jeder Maschine.

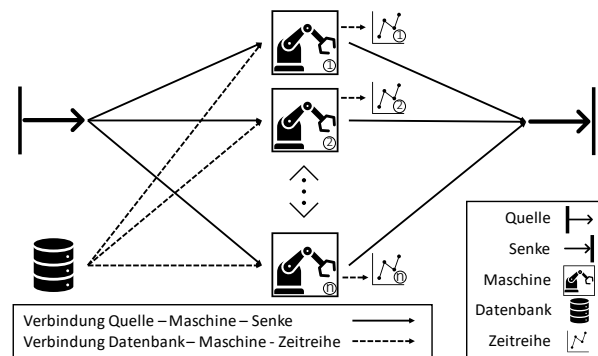


Abbildung 1: Modell des untersuchten Fertigungsverbunds

Erreicht die Simulationszeit die Startzeit des FA, wie in den Beschreibungen des FA hinterlegt, wird die Maschine belegt und arbeitet nun den FA ab. Hierbei wird die Länge des FA und die benötigte Energie vom trainierten Seq2Seq-Modell anhand der Beschreibung des spezifischen FA prognostiziert.

2.1 Generierung von Zeitreihen

Wie eingehend erwähnt wird eine Menge von Eingangssequenzen X und Zielsequenzen Y als Sequenzpaarungen $\{X_i, Y_i\}$ in einem Seq2Seq-Modell verwendet. Hierfür wird der Spanprozess eines FA auf einer Werkzeugmaschine (WZM) betrachtet.

Ein NC-Code beschreibt eine Abfolge notwendiger technologischer Schritte bis zur Beendigung eines FA und kann somit als eine konkrete Beschreibung einer dem Prozess zugrundeliegenden Zustandsfolge verstanden werden. Der NC-Code bestimmt also maßgeblich das Verhalten innerhalb des Spanraums einer WZM. Weiter gilt ein FA erst als abgeschlossen, wenn der NC-Code einmal komplett durchlaufen wurde.

Weiter wird die Beschreibung des FA um entspre-

chende Betriebsmodi, Schruppen und Schlichten, ergänzt. Der NC-Code, gemeinsam mit dem Betriebsmodus stellen hier die Eingangssequenz X_i eines RNN-ED dar.

Basis der Ausgangszeitreihen Y_i quasi-kontinuierlicher Ausgabewerte ist der Strombedarf [W] desselben FA bei Durchlauf des NC-Codes im jeweiligen Betriebsmodus. Der zeitliche Strombedarf gibt konkret Aufschluss darüber, wann mit wieviel benötigter Leistung gerechnet werden muss, sobald über die Einsteuerung eines FA entschieden werden muss.

In der Trainingsphase wird nun ein ungewichtetes KNN, bestehend aus einem RNN-ED, anhand der Eingangs- und Zielsequenzen $\{X_i, Y_i\}$ parametrieren (siehe Abb. 2).

Ist die Parametrierung des RNN-ED abgeschlossen, benötigt dieses lediglich die Eingangssequenz, um die Ausgangssequenz zu erzeugen. Aufgabe der Inferenzphase ist es nun, ein aussagekräftiges Strombedarfsprofil \hat{Y}_i auszugeben. Inferenzierung ist hier das logische Schließen aus Daten.

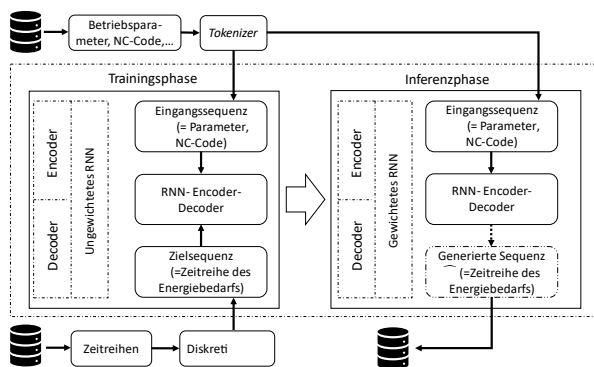


Abbildung 2: Bestandteile einer RNN-Encoder-Decoder Topologie für Sequenzen unterschiedlicher Länge und Symbolik während der Trainings- und Inferenzphase [8].

Die so postulierte Methode der Erzeugung quasi-kontinuierlicher Zeitreihen stellt, in Kombination mit den Modellierungsmöglichkeiten diskret-ereignisgesteuerter Simulationssysteme, eine Methode der hybriden Simulation dar.

2.2 Inferenz zur Laufzeit

Um die Generierung der Zielsequenz \hat{Y}_i innerhalb einer Simulation zu gewährleisten, muss die Inferenzmethode als Funktion hinterlegt sein (siehe Code 1).

```
def seq2seq(job):
```

```
    load seq2seq model
    seq2seq(job)
    return ts_job
```

Code 1: Pseudocode der Inferenzmethode

Hierfür muss lediglich das parametrisierte Modell *seq2seq model* geladen werden und diesem die Beschreibungen des spezifischen FA *job* als Parameter übergeben werden. Die Inferenzmethode kann nun innerhalb des Simulationsmodells einer WZM aufgerufen werden und gibt eine Zeitreihe *ts_job* zurück.

Das hier verwendete Modell der Fertigung (siehe Code 2) besteht aus einem Maschinenmodell auf welchem sich zwei Prozesse, Warten und Fertigen, abwechseln.

```
class machine(name):
```

```
    self.name=name
    self.parts_made=0
    self.jobs=[list of jobs and descriptions]
    self.ts_energy= empty series
```

```
def waiting(self):
```

```
    timeout_job=
    self.jobs[parts_made[start time]]
    yield timeout(timeout_job)
    yield machining
```

```
def machining(self):
```

```
    seq2seq(jobs[parts_made[description]])
    yield timeout(len(ts_job))
    self.parts_made +=1
    concat(ts_energy,ts_job)
    yield waiting
```

Code 2: Pseudocode des Simulationsmodells

Die Maschine wird hier in SimPy, einem objektorientiertem Simulationsparadigma (vgl. [15]), beschrieben. Das Objekt, die Maschine, wird durch mehrere Variablen beschrieben. Sie verfügt über einen Namen, einen Teilezähler, eine Liste von geplanten FA, sowie über ein, anfangs leeres, Datenobjekt, in welchem die Zeitreihen der einzelnen FA gespeichert werden. Weiter verfügt die Maschine über zwei Funktionen, Warten und Fertigen. Die Maschine wartet so lange bis die Simulationszeit der Startzeit eines FA entspricht. Darauf folgend wird Fertigungsprozess begonnen.

Der Fertigungsprozess (*machining*) wird anhand der Seq2Seq Funktion beschrieben. Die Funktion erhält die Beschreibung des aktuellen FA und gibt eine Zeitreihe dessen Energiebedarfs *ts_job* zurück. Die Länge des FA

ist gleich der Anzahl der einzelnen Elemente der Zeitreihe. Die Maschine wird nun für die Länge der Zeitreihe blockiert. Anschließend wird die Zeitreihe im Datenobjekt *ts_energy* gespeichert, die Zählvariable der abgeschlossenen FA um 1 erhöht und die Maschine beginnt wieder zu warten. Dies geschieht, bis alle FA abgearbeitet wurden.

3 Versuchsvorbereitung

3.1 Vorbereitung der Seq2Seq-Methode

Dem Seq2Seq-Modell werden in der Trainingsphase einerseits Eingangssequenzen, bestehend aus den jeweiligen NC-Codes und Betriebsmodi, andererseits Zielsequenzen, bestehend aus Zeitreihendaten eines FA gegenübergestellt. Es wurde eine Menge ($i = 51$) Sequenzpaarungen an einer WZM über mehrere Tage aufgenommen. Diese müssen entsprechend der in [16] vorgestellten empirischen Ergebnisse vorverarbeitet werden, damit das Modell einen aussagefähigen Zusammenhang zwischen den beiden Mengen erlernt.

Die Eingangssequenzen \mathbb{X} werden um verschiedene Betriebsmodi $\{x_1, x_2\}$, in denen die WZM betrieben werden kann, erweitert. Diese Betriebsmodi spiegeln eine gängige Arbeitsroutine bei der Bearbeitung eines FA wider. Der NC-Code läuft zum ersten Mal $\{\text{Schruppen} = x_1\}$, um eine größere Menge an überschüssigem Material abzutragen und dem Material seine Form zu geben. Danach wird der gleiche NC-Code noch mehrere Male ausgeführt $\{\text{Schlichten} = x_2\}$, um die Oberfläche des nun in Form gebrachten Materials zu glätten.

Die Grundlage der Werte für die Zielzeitreihe \mathbb{Y} bildet der reale Strombedarfs eines FA beim Durchlauf eines NC-Codes. Die Zeitreihendaten wurden unter Feldbedingungen aufgezeichnet und haben die gleiche Taktung $\Delta t = 500$ ms.

Der Energiebedarf der WZM und damit die Zeit, die für die Bearbeitung eines spezifischen Auftrags benötigt wird, wird zunächst bei jeder Bearbeitung eines Auftrags überwacht und als Zeitreihendatensatz gespeichert. Anschließend wird das Seq2Seq Modell mit den oben beschriebenen Trainingsdaten parametrisiert und für die Weiterverwendung in der Simulation gespeichert [8].

3.2 Vorbereitung der Referenzzeitreihen

Zur Validierung des durch HS erzeugten Lastprofils soll

parallel hierzu ein Lastprofil rein aus den Daten des zugrundeliegenden Systems erstellt werden (Siehe Abb. 3). Hierzu wird die Messzeitreihe von 6 Tagen tageweise auf 6 Maschinen umgelegt.

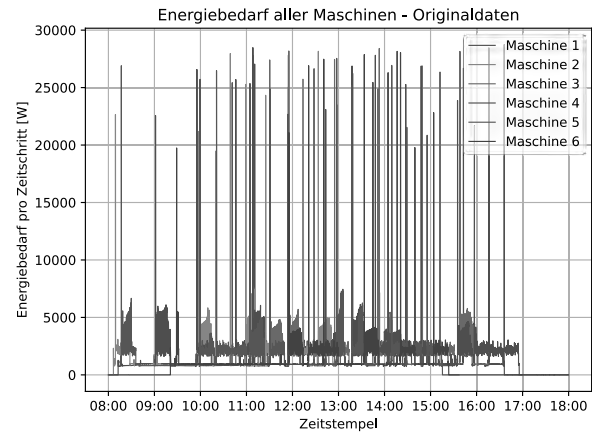


Abbildung 3: Originaldaten des Strombedarfs für alle Maschinen der Fertigung einer Schicht.

Die Daten müssen hierfür vorverarbeitet werden. Einerseits betrachtet die später verwendete Seq2Seq Methode nur den Energiebedarf bei Ausführung eines FA. Die Rohdaten müssen daher um die Zeiträume, in denen keine Fertigung stattfindet, bereinigt werden. Andererseits soll aus den Daten ein Gesamtlastprofil erstellt werden. Hierfür müssen die Zeitreihen anhand eines gemeinsamen Index aufsummiert werden. Die Rohdaten wurden mit der gleichen Messstrecke von 500 ms aufgenommen. Die einzelnen Zeitpunkte unterliegen jedoch einer Messtoleranz, welche zu einem leicht unterschiedlichen Zeitstempel führt.

Weiter werden Lastprofile zur Kommunikation des zeitlichen Energiebedarfs zwischen Verbraucher und Versorger verwendet. Da jeder Versorger über eine unterschiedliche Energieinfrastruktur verfügt (Speicher, Kraftwerke, Netze, etc.) muss das erzeugte Lastprofil an den Bedürfnissen des Versorgers ausgerichtet sein. Diese Bedürfnisse entsprechen der Möglichkeit des jeweiligen Versorgers effizienter Energie in seinem Verteilnetzesystem zur Verfügung zu stellen. Die Lastprofile müssen daher unterschiedliche Fragen nach bspw. mittlerem, minimalem oder maximalem Energiebedarf für unterschiedliche Zeiträume beantworten. Ein Mangel an öffentlicher Information über die versorgerseitige Erstellung von Lastprofilen trägt hierzu weiter bei [17].

Diese Probleme werden dadurch gelöst, dass die Zeitreihen einer *Resampling* Methode unterzogen werden.

Resampling ist der Prozess der Aggregation eines Intervalls an Datenpunkten zu einem neuen Datenpunkt, samt neuem Zeitstempel. Die *Resampling* Methode benötigt hierfür 2 Parameter, die Schrittlänge des Intervalls und die Aggregationsmethode, welche beide anhand des Anwendungsfall bestimmt werden können.

Die Schrittlänge sollte sich einerseits am Betrachtungshorizont des Lastprofils orientieren. Diese wird bedingt durch die registrierende Leistungsmessung auf 15 Minuten Intervalle gelegt. Für die Aggregationsmethode wird eine Mittelwertberechnung, als robustes Vergleichsmaß für den mittleren Energiebedarf, gewählt.

Weiter soll der Minimalbedarf während eines Intervalls bestimmt werden, um eine genauere Aussage über das stromverbrauchende System zu treffen. Hierfür wird wieder ein Intervall von 15 Minuten gewählt, aber dieses Mal mit einer Minimalwertberechnung.

Die Energiebezugskosten sind nicht allein von der Menge des benötigten Stroms, sondern auch vom Zeitpunkt des Bedarfs, in Form vom Versorger zur Verfügung gestellter Kapazität, abhängig. Daher wird die Betrachtung der Zeitreihen noch um ein weiteres *Resampling* erweitert. Hier wird eine Maximalwertmethode angewandt, um den maximalen Bedarf an Energie in einem Intervall darzustellen. Weiter wird das Intervall auf 1 Minute gesetzt. Dies liegt hier darin begründet, dass der maximale Energiebedarf nicht nur über einen längeren Zeitraum dargestellt werden soll, sondern auch in Form von Lastspitzen. Wird ein FA um einen Zeitraum verschoben, verschieben sich auch die Lastspitzen des FA um eben diesen. Dieser Zeitraum ist wiederum abhängig von der Regelstrecke des Bezugssystems. Diese wurde vor Datenerhebung für den Werker der Maschine auf 1 Minute gesetzt.

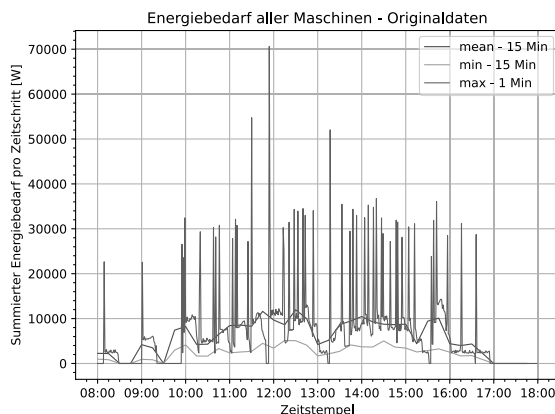


Abbildung 4: Ergebnisse der bereinigten Daten, nach Resampling mit einer Mittel-, Minimal- und Maximalwertmethode.

Im Anschluss an das *Resampling* werden alle 3 so erzeugten Zeitreihen für alle Maschinen aufsummiert (siehe Abb. 4). Die Methode der Synthese eines Lastprofils aus Einzellastprofilen ist nicht neu und wurde schon bei [2, 18] diskutiert.

Die gewählten *Resampling* Parameter werden für die mit der HS erzeugten Zeitreihen wiederverwendet, um eine Vergleichbarkeit der beiden Datensätze zu erreichen.

4 Versuchsdurchführung

Die HS wird in Python mit SimPy [15] modelliert und durchgeführt. Als Framework für die verwendete Seq2Seq-Methode wird Tensorflow [19] verwendet. Für die GPU Berechnung des Tensorflow Modells stand eine RTX 2080 Ti mit 4352 CUDA Kernen zur Verfügung. Die Parametrierung des Modells dauert auf dieser ca. 6 Tage. Die Inferenzierung eines FA dauert hingegen nur wenige Sekunden.

4.1 Ergebnisse der HS

Die HS wird nun gestartet und gibt Zeitreihen nach der in Code 2 beschriebenen Methode aus. Diese werden noch zur Laufzeit für jede Maschine in einem eigenen Datensatz gespeichert. Anschließend werden die inferenzierten Zeitreihen nach, der in Kapitel 3.2 beschriebenen Methode, aggregiert. Die nun erzeugten Lastprofile (vgl. Abb. 5) stellen den Strombedarf eines Fertigungssystems dar, welche nur durch ein Seq2Seq-Modell, Beschreibungen und Startzeitpunkte der FA erzeugt wurden.

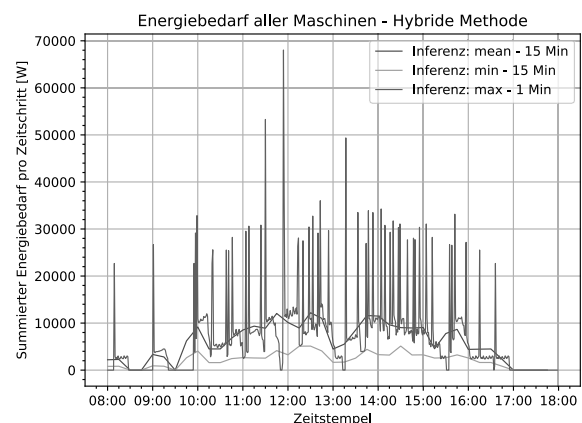


Abbildung 5: Ergebnisse der durch Seq2Seq erzeugten Daten, nach Resampling mit einer Mittel-, Minimal- und Maximalwertmethode.

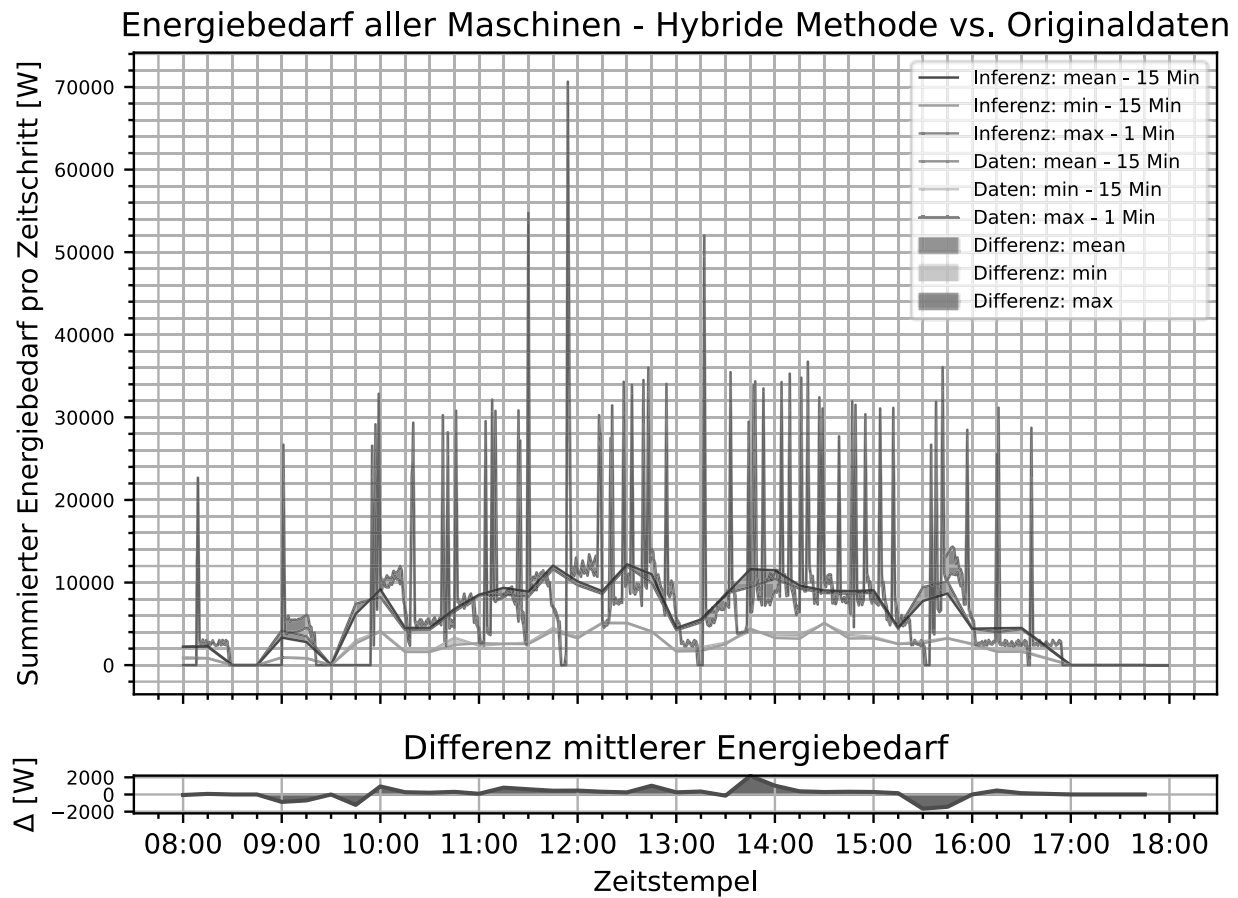


Abbildung 4: Oben: Visueller Abgleich der Ergebnisse der hybriden Methode mit denen der abgespielten Daten. Auffällig ist die starke Übereinstimmung der Trainings- und Inferenzdaten, die auch aus der unten dargestellten Differenz des mittleren Energiebedarfes über die Zeit abzulesen ist.

4.2 Validierung der HS am realen Datensatz

Vergleicht man die verschiedenartig erzeugten Lastprofile von Abb. 4 und Abb. 5 so ist eine deutliche Ähnlichkeit auffällig. Diese Ähnlichkeit wird weiter unterstrichen durch einen visuellen und analytischen Abgleich der beiden Lastprofilmenngen.

Im direkten Vergleich (vgl. Abb. 6- oben) sind die Unterschiede, in Form von farbig schattierten Flächen, kaum wahrzunehmen. Dies betrifft alle gegenübergestellten Lastprofile, sowie die Lage der konkreten Lastspitzen. Besonders deutlich wird dies in der über die Zeit abgetragenen Differenz des mittleren Energiebedarfes (vgl. Abb. 6- unten).

Vergleicht man die Lastprofile analytisch anhand ihres konkreten Energiebedarfes über die Gesamtzeit in Kilowattstunden [kWh], so wird dies noch weiter untermauert (vgl. Tab. 1).

Hierzu wird die *Riemann'sche* Summe aller einzelnen Lastprofile gebildet und miteinander verglichen. Die Methode der *Riemann'schen* Summen [20] wird dann angewandt, wenn das Integral über einer Funktion über diskrete Datenpunkte approximiert werden soll.

[kWh]	Seq2Seq	Original- daten	<i>Seq2Seq</i> <i>Daten</i>
Mean	59.47	58.07	1.02
Min	22.18	23.19	0.96
Max	69.02	69.67	0.99

Tabelle 1: Vergleich der Energiemengen der unterschiedlichen Methoden und Lastprofile.

5 Kritische Betrachtung

Die grundlegende Funktionalität des beschriebenen hybriden Simulationsmodells wurde im Testszenario bestätigt.

Für eine abschließende Bewertung der verwendeten Methoden ist es ratsam, die qualitative und quantitative Datenbasis des Seq2Seq-Modells zu erweitern. Der hier verwendete Datensatz ist von geringer Größe und bezieht sich nur auf einen Fertigungsauftrag in unterschiedlichen Bearbeitungsmodi. Eine methodische Erweiterung der Seq2Seq-Methode ist daher notwendig, um auch unterschiedliche Fertigungsaufträge mit unterschiedlichen NC-Codes für das Training verwenden zu können. Ein Ansatzpunkt hierzu könnte die Verwendung synthetischer Trainingsdaten sein, welche kostengünstig und transparent zu generieren wären.

Im Erfolgsfall könnte dann eine Lösung entwickelt werden, die auf Basis nicht in den Trainingsdaten vorhandener NC-Codes plausible Stromverbrauchsprognosen für neue FAs generiert. Dies hätte ein hohes praktisches Potenzial und wäre auch aus wissenschaftlicher Sicht ein bedeutender Erfolg.

Darüber hinaus fehlt dem vorgeschlagenen Seq2Seq-Modell eine geeignete Bewertungsmethode für einzelne Zeitreihen eines FA [8]. Es gilt zu diskutieren, ob dies auch der Fall für die hier vorgestellten Gesamtzeitreihen ist. Der Abgleich in Abbildung 6 hat gezeigt, dass ein Gesamtlastprofil, welche aus einzelnen inferenzierten Lastprofilen erstellt wurde, mit Lastprofilen, welche rein aus dessen Trainingsdaten erstellt wurde, verglichen werden kann. Da Trainingsdaten zwingend für Parametrieren des ML-Modells vorhanden sein müssen, könnten diese in ihrer Gesamtheit, wie hier vorgestellt, zur Validierung des Lernverhaltens verwendet werden. Weiter weisen die generierten Gesamtlastprofile die gleiche Struktur im Hinblick auf ihren Gesamtzeitraum und Zeitstempel auf und können somit durch Methoden der Zeitreihenanalyse wie dem *Mean Absolute Error* oder der *Riemann'schen* Summe verglichen werden.

Die Weiterentwicklung der oben beschriebenen Methode des maschinellen Lernens und ihre Anwendung für hybride Simulationsmodelle ist derzeit Gegenstand laufender Forschung.

Die Übertragung der Grundidee auf andere Formen von Eingangssequenzen und Zeitreihen anderer Messwerte ist ebenfalls denkbar und ein möglicher Gegenstand weiterer Untersuchungen.

Literaturverzeichnis

- [1] Bundesnetzagentur. 2005. Verordnung über die Entgelte für den Zugang zu Elektrizitätsversorgungsnetzen (Stromnetzentgeltverordnung -StromNEV). StromNEV.
- [2] Benjamin Jacobsen und Maximilian Stange. 2020. Vorgehensmodell zur Simulation von gebündeltem Energiebedarf. In *Proceedings ASIM SST 2020*. ARGESIM Publisher Vienna, 295–301. DOI: <https://doi.org/10.11128/arep.59.a59042>.
- [3] Bundesnetzagentur. 2005. *Verordnung über den Zugang zu Elektrizitätsversorgungsnetzen (Stromnetzzugangsverordnung - StromNZV)*. StromNZV.
- [4] Navonil Mustafee, Sally Brailsford, Anatoli Djanatliev, Tillal Eldabi, Martin Kunc, und Andreas Tolk. 2017. Purpose and benefits of hybrid simulation: Contributing to the convergence of its definition. In *Proceedings of the 2017 Winter Simulation Conference (WSC)*. IEEE Press, Piscataway, NJ, 1631–1645. DOI: <https://doi.org/10.1109/WSC.2017.8247903>.
- [5] Ilya Sutskever, Oriol Vinyals, und Quoc Le V. 2014. Sequence to Sequence Learning with Neural Networks. In *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems*. MIT Press, Cambridge, MA, USA.
- [6] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, und Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar.
- [7] Benjamin Wörrlein, Sören Bergmann, Niclas Feldkamp, und Steffen Straßburger. 2019. Deep-Learning-basierte Prognose von Stromverbrauch für die hybride Simulation. In *Simulation in Produktion und Logistik 2019*. Verlag Wissenschaftliche Scripten, Auerbach, 121–131.
- [8] Benjamin Wörrlein und Steffen Straßburger. 2020. On the Usage of Deep Learning for Modelling Energy Consumption in Simulation Models. *SNE* 30, 4, 165–174. DOI: <https://doi.org/10.11128/sne.30.tn.10536>.
- [9] Anna C. Römer, Martina Rückbrod, und Steffen Straßburger. 2018. Eignung kombinierter Simulation zur Darstellung energetischer Aspekte in der Produktionssimulation. In *ASIM 2018 : 24. Symposium Simulationstechnik*. ARGESIM/ASIM, Wien, 73–80.
- [10] Thorsten Pawletta, Artur Schmidt, und Peter Junglas. 2017. A Multimodeling Approach for the Simulation of Energy Consumption in Manufacturing. *SNE* 27, 2, 115–124. DOI: <https://doi.org/10.11128/sne.27.tn.10377>.
- [11] Anna C. Roemer und Steffen Strassburger. 2019 - 2019. Hybrid System Modeling Approach for the Depiction of the Energy Consumption in Production Simulations. In *2019 Winter Simulation Conference (WSC)*. IEEE, 1366–1377. DOI: <https://doi.org/10.1109/WSC40007.2019.9004772>.
- [12] Rüdiger W. Brause. 1995. *Neuronale Netze. Eine Einführung in die Neuroinformatik* (2., überarbeitete und

- erweiterte Auflage). Leitfäden der Informatik. Vieweg+Teubner Verlag, Wiesbaden.
- [13] Andreas Zell. 2003. *Simulation neuronaler Netze* (4., unveränd. Nachdr). Oldenbourg, München.
 - [14] Ian Goodfellow, Yoshua Bengio, und Aaron Courville. 2016. *Deep Learning*. MIT Press.
 - [15] Klaus Müller und Tony Vignaux. 2002. *Simpy. a process-based discrete-event simulation framework* (2002). URL: <http://simpy.readthedocs.org>.
 - [16] Benjamin Wörrlein und Steffen Straßburger. 2020. Sequence to Sequence Modelle zur hochaufgelösten Prädiktion von Stromverbrauch. In *Proceedings ASIM SST 2020*. ARGESIM Publisher Vienna, 149–157. DOI: <https://doi.org/10.11128/arep.59.a59021>.
 - [17] C. M. Colson und M. H. Nehrir. 2009. An alternative method to load modeling for obtaining end-use load profiles. In *41st North American Power Symposium*. IEEE, 1–5. DOI: <https://doi.org/10.1109/NAPS.2009.5484036>.
 - [18] Noah D. Pflugradt. 2016. *Modellierung von Wasser und Energieverbräuchen in Haushalten*. Dissertation. Universitätsbibliothek Chemnitz, Chemnitz.
 - [19] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems* (2015). URL: <https://www.tensorflow.org>.
 - [20] Shigeru Takahashi. 1955. Notes on the Riemann-sum. *Proc. Japan Acad. Ser. A Math. Sci.* 31, 1. DOI: <https://doi.org/10.3792/pja/1195525845>.

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

Marian Göllner^{1*}, Sven Jacobitz¹, Taihao Li¹, Xiaobo Liu-Henke¹

¹Institut für Mechatronik, Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel; *mar.goellner@ostfalia.de

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

Introduction

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

An intelligent vehicle must be capable of performing the four basic tasks of measuring, recognition, planning and execution in order to achieve its intrinsically set or extrinsically motivated goal. First of all, it must detect (measure) its environment with sensors

and process (recognize) the sensor information for environment perception. Based on that, the autonomous system makes decisions about its behavior and its interaction with the environment (planning) and finally realizes these decisions with the help of its actuators to execute the basic movement.

Motivation

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

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

Methodology

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

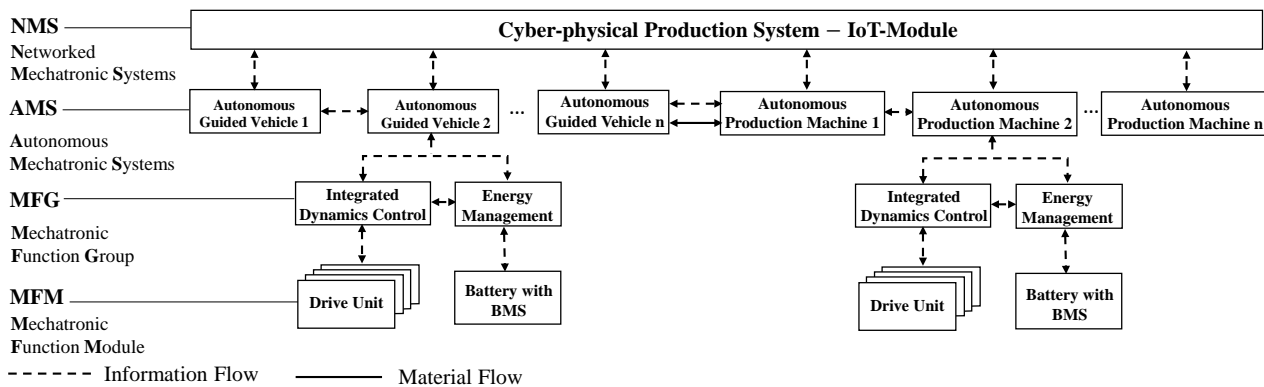


Abbildung 1: Mechatronic structuring and hierarchization of a cyber-physical industry 4.0 production plant

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

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

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

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

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

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

1 State of the art

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

1.1 Mapping

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

1.2 Route Planning

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

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

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

2 Concept of doplar

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

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

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

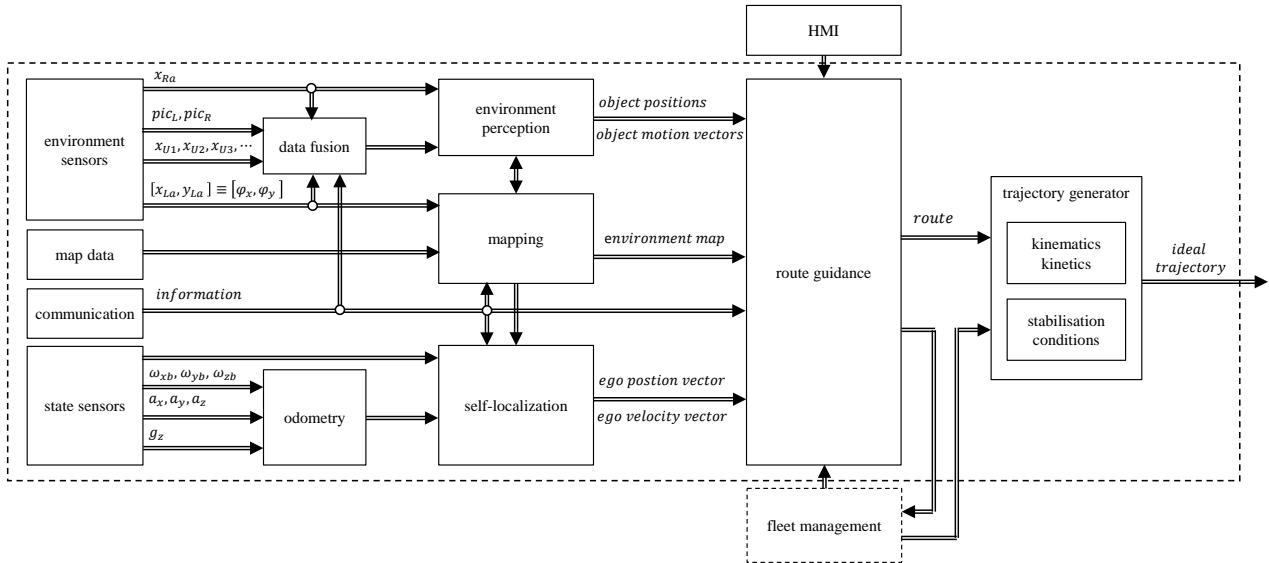


Abbildung 2: Concept of the domain-specific configurable modular platform for route guidance and trajectory planning of intelligent vehicles (doplar)

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

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

3 Design of doplar's route guidance

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

3.1 Mapping

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

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

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

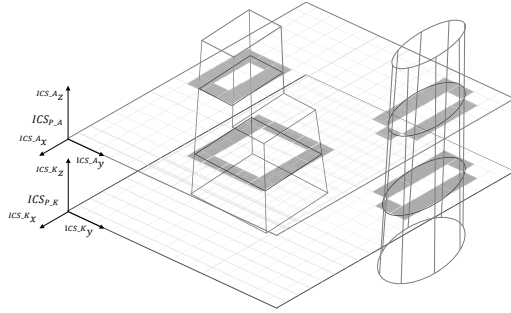


Abbildung 3: Creating a 2.5-dimensional, geometric map from data of the environment perception

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

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

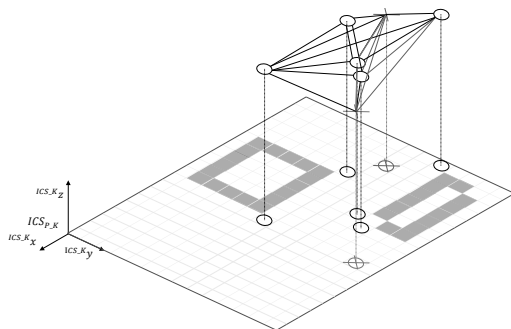


Abbildung 4: Creating a topological graph map based on a 2.5-dimensional geometric map

3.2 Route guidance

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

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

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

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

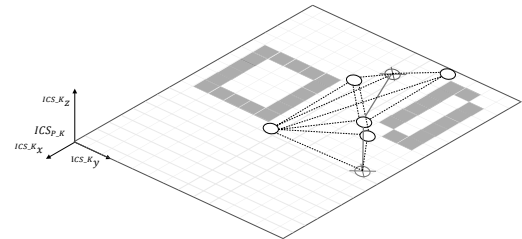


Abbildung 5: Result of Dijkstra's algorithm for finding the optimal route in a topological graph map

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

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

The integration of dynamic information from wireless communication in the cyber-physical traffic sys-

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

3.3 Trajectory planning

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

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

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

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

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

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

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

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

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

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

4 Demonstration of the platform doplar

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

4.1 Pilot application in the domain of Industry 4.0

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

The realization of the AGV is shown in Figure 6. Mecanum wheels are used to transmit the drive torque

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

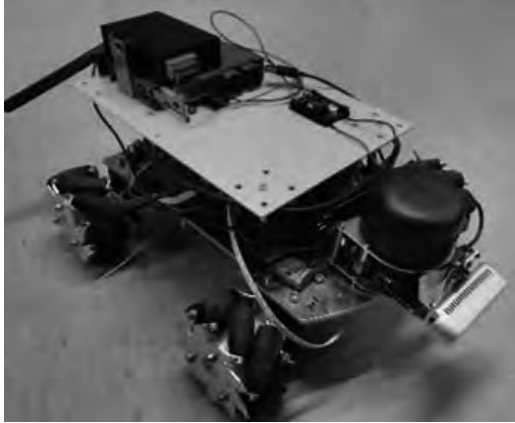


Abbildung 6: Realization of the AGV

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

4.2 exemplary results

First, the doplar was used to generate a graph from the floor plan, which enables route guidance. The fleet management initially initiates the route guidance for the three AGVs. On the basis of the routes generated, the time to complete the order is determined so that the AGV with the shortest transport time can be selected for the order. A result of the route guidance system is shown in Figure 8. The three determined routes, which also seem optically plausible, are recognizable. AGV 3 has the shortest route to the order location and therefore also requires the shortest transport time so that this AGV should be selected for the transport order.

This pilot application shows exemplary the functionality of the platform doplar from generating a graph

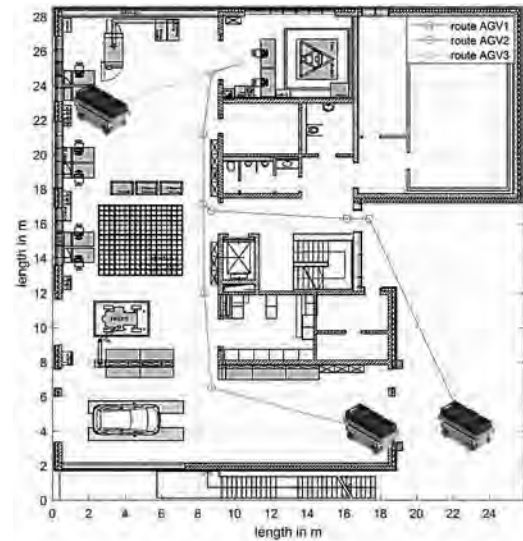


Abbildung 7: Result of the route guidance for the three AGVs as basis for the fleet management for the distribution of the transport order

map from a floor plan to route guidance. further examples are concluded in [2].

5 Conclusion and Outlook

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

For the AGV that has received the order, doplar generates an optimal trajectory for its completion. In the following work steps, the developed platform doplar is to be integrated into other domains and real systems, examined under real-time conditions and further optimized.

Acknowledgement

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

References

- [1] Liu-Henke X. *Mechatronische Entwicklung der aktiven Feder-/Neigetechnik für das Schienenfahrzeug RailCab: Zugl.: Paderborn, Univ., Diss., 2005*, vol. 589 of *Fortschritt-Berichte VDI Reihe 12, Verkehrstechnik/Fahrzeugtechnik*. Düsseldorf: VDI-Verl., als ms. gedr ed. 2005.
- [2] Jacobitz S, Göllner M. Seamless validation of cyber-physical systems under real-time conditions by using a cyber-physical laboratory test field. In: *IEEE International Conference on Recent Advances in Systems Science and Engineering*. 2021; .
- [3] Kortenkamp D, Weymouth T. Topological Mapping for Mobile Robots Using a Combination of Sonar and Vision Sensing. In: *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 2)*, AAAI'94. USA: American Association for Artificial Intelligence. 1994; pp. 979–984.
- [4] Thrun S, Burgard W, Fox D. A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots. *Machine Learning*. 1998;31(1/3):29–53.
- [5] Kweon IS, Kanade T. High-resolution terrain map from multiple sensor data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1992; 14(2):278–292.
- [6] Neira J, Tardos JD. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*. 2001; 17(6):890–897.
- [7] Leonard JJ, Durrant-Whyte HF. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*. 1991; 7(3):376–382.
- [8] Benkert M, Widmann F, Wolff A. The Minimum Manhattan Network Problem: A Fast Factor-3 Approximation.
- [9] Walter R. Metrische Räume. In: *Einführung in die Analysis*, edited by Walter R, pp. 270–327. Berlin: de Gruyter. 2007;.
- [10] Dijkstra EW. A note on two problems in connexion with graphs. *Numerische Mathematik*. 1959;1(1):269–271.
- [11] Hart PE, Nilsson NJ, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*. 1968;4(2):100–107.
- [12] Bellman R. On a routing problem. *Quarterly of Applied Mathematics*. 1958;16(1):87–90.
URL
<http://www.jstor.org/stable/43634538>
- [13] Floyd RW. Algorithm 97: Shortest path. *Communications of the ACM*. 1962;5(6):345.
- [14] Warshall S. A Theorem on Boolean Matrices. *Journal of the ACM*. 1962;9(1):11–12.
- [15] Kleene SC. Representation of Events in Nerve Nets and Finite Automata. In: *Automata Studies. (AM-34)*, edited by Shannon CE, McCarthy J, Annals of Mathematics Studies, pp. 3–42. Princeton, NJ: Princeton University Press. 1956;.
- [16] Golfarelli M, Maio D, Rizzi S. Elastic correction of dead-reckoning errors in map building. In: *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, vol. 2. 1998; pp. 905–911 vol.2.
- [17] Scherler S, Liu-Henke X, Henke M. Predictive Energy Management for an Electric Vehicle with Fuel Cell Range Extender in Connected Traffic Systems. In: *2020 19th International Conference on Mechatronics - Mechatronika (ME)*. IEEE. 02.12.2020 - 04.12.2020; pp. 1–8.
- [18] Göllner M, Zhang J, Liu-Henke X. Model predictive trajectory control for automated driving of a spherical electrical drive. In: *2018 IEEE International Systems Engineering Symposium (ISSE)*. IEEE. 10/1/2018 - 10/3/2018; pp. 1–6.
- [19] Göllner M. Fusion structure for environment perception and model-predictive navigation of autonomous vehicles. 25.06.2019.

Service-oriented Architecture for Automatic Evaluation of Urban Logistical Concepts

Richard Pump^{1*}, Lasse Bienzeisler², Torben Lelke², Volker Ahlers¹, Arne Koschel¹

¹University of Applied Sciences and Arts Hannover, 30459 Hannover, Germany; *richard.pump@hs-hannover.de

²Technische Universität Braunschweig, Institute of Transportation and Urban Engineering, 38108 Braunschweig, Germany

Abstract. With the rapidly increasing availability of logistics concepts new tools are required to provide accurate estimations of concept effectiveness to urban planners. In this paper we describe a service oriented architecture providing automated concept evaluation through agent-based traffic simulation, combining general traffic models with specialised logistical models into singular simulations. Key focus of the paper is the integration architecture utilized to combine the different tools from a previously developed manual workflow. The architecture is designed to incorporate multiple different tools into a combined workflow, providing non-domain users with a highly specialised tool.

Introduction

Urban Logistics is currently changing at a rapid pace. An ever increasing consciousness for environmental factors, new regulatory measures, new services and a pandemic are moving populations from traditional solutions towards utilization of novel logistics concepts [1, 2]. This development confronts urban planners with new challenges as the number of design options for a city logistics system continues to increase, challenging the selection of the most suitable measures to improve the quality of urban mobility and thus environmental, economic and social effects. New tools are needed to support decision making in municipal planning agencies to increase utilization and effectiveness of logistics solutions [3]. A team consisting of traffic engineers, business IT specialists, computer scientists and municipal government employees was tasked in the research project *USEFUL* to create a simulation-based decision support tool for urban logistics [4, 5].

An agent-based simulation approach using the multi-agent transport simulation (MATSim) [6] was chosen to evaluate the impacts of logistics concepts

on urban traffic patterns within the city of Hannover. For the evaluation, different key performance indicators (KPIs) were defined. Based on these simulation results, the observed impacts on the KPIs were evaluated in order to derive recommendations for urban planners. Consequently a hand-crafted traffic model [7], adapted to the resulting needs and behavioral patterns stemming from logistics concepts was developed, which in turn fed a semi-automated evaluation tool chain. Due to the comparatively long run times of the agent-based simulations and the high computational effort, all scenarios were simulated manually and only the corresponding impacts were stored in the tool. These results are on display on the project's web page¹, where users can gather information about the concepts, the evaluation areas and predicted effects on different KPIs.

However, the workflow with the semi-automatic traffic model generation and the evaluation framework is currently only applicable to the city of Hannover. In particular, the setup of the traffic model is limited by available data sources and therefore not directly transferable to other cities and municipalities. To provide similar assistance to other communities, a completely automated architecture needs to be designed and implemented. The resulting tool has to utilize public and private data sources to create an accurate baseline model of the area of interest, before standardized changes to the model implement novel logistics concepts.

To tackle these issues – and increase the usefulness of the web-based tool – the follow-up project *USEFUL-XT* aims to increase the transferability of the *USEFUL* workflow. In this paper we present the architecture to combine previously manual workflows using a service-oriented approach. A high-level view of the different services is given, as the implementation of the services is still in progress.

The result is the concept of an integrated, service-

¹<https://useful.uni-hannover.de>

oriented and cloud-based framework capable of automatically collecting and processing the required data, creating a simulation model and performing the resulting simulation runs. As the specified simulation scenario only need to be simulated once, the computation needs are also dynamic and efficient resource usage demands a dynamically scalable solution with adequate storage space to store results.

Section 1 will give an overview of related work. In the following Section 2 the current workflow of the project is presented, which is then expanded to an automatic workflow in Section 3. Remaining challenges are discussed in Section 4 and a conclusion is given in Section 5.

1 Related Work

A similar tool to the *USEFUL* web application was provided by [8]. The Urban Transport Roadmap provides statistical evaluations for many regulatory measures and estimates the impacts of technological development on urban traffic. However, the tool focuses on very rough estimates for many different cities and allows in-depth analysis of combinations of measures.

Another theoretical framework for automated evaluation of urban traffic was proposed by [9]. In their proposed system, a user can utilize a partially automated co-creation process to construct simulation models of a city. After model creation, partial changes to the model are proposed to the user, which, when chosen, are in turn simulated within multiple simulation tools. At the same time a rule-based system watches the simulation for inconsistencies and provides input to the user to correct these. The publication focuses on the process of creating the tool, rather than the tool itself. It is unclear if the tool can provide automated model creation for multiple areas and logistics concepts. Furthermore, the proposed system of [9] does not include an expert system to fast-track concept evaluation. The proposed system remains theoretical.

In another work, [10] present a generalized pipeline for the generation of agent-based transport models in France. This pipeline presents a standardized approach for processing data sources to synthesize an synthetic population and assign mobility behaviour to the agents. The paper also includes a detailed validation of the method. However, the described method is focused entirely on the generation of a model for private traffic omitting commercial traffic and is adapted to the specif-

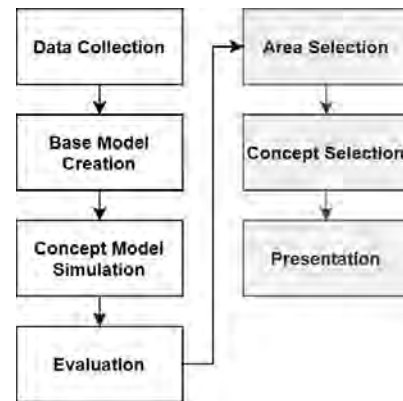


Figure 1: Current manual workflow of the project *USEFUL*.
User interactive parts marked in grey.

ically in France available data.

An example for a microscopic modeling approach for commercial traffic is presented in [11]. Therein, the authors then employ land use and structural data to generate businesses and the respective employees and vehicle fleets. By applying the German commercial travel survey, each vehicle is equipped with a daily schedule.

Meanwhile, many other simulation models and studies have been created to identify issues and solutions for urban logistics [12]. However, model creation was not completely automated and often only small areas of urban logistics were modelled. Our proposed system approaches urban logistics from a more abstract point of view, allowing a rough estimation of impacts for many concepts and areas, through automated model creation, simulation and evaluation.

2 Current Solution

The current workflow combines manual data collection, preparation, and model calibration as well as simulation and evaluation. A lightweight, web-based decision support tool was created, not slowed down by long simulation run times, as users want to use the tool at a quick glance. The goal of the process was the easy-to-understand representation of logistic concept impacts on specific KPIs, which are presented to users in a web-based decision support tool. Traffic and logistic simulations are performed using MATSim to benchmark the impact of new logistics concepts. The process can be split into seven parts, as shown in Figure 1.

The first step of the workflow is data collection, as many different sources of information have to be uti-

lized to manually collect the necessary traffic supply and demand information. Several open data sources like OpenStreetMap (OSM) and public transport providers are considered to create a road network with an integrated public transport system for the traffic supply needed in the MATSim model.

To generate a realistic traffic demand, multiple studies and non-public data from companies and municipalities are consolidated into a synthetic population. Furthermore, different input surveys are used to derive the travel or logistic demand from the synthetic population. From the collected data a *basecase* model is created, representing the status quo of traffic within the research area. A manual calibration of the model using traffic counts, as well as a validation against different mobility parameters is performed to achieve a high model accuracy. Building upon the previously created *basecase*, demand changes due to the implementation of novel logistic concepts are replicated in the model and need to be simulated subsequently. Each concept is parameterized in different dimensions (e.g. usage of electric vehicles, total acceptance percentage, etc.), studying concepts more in-depth. A large amount of simulation runs were needed for every area of interest and each parameter combination, since multiple logistics concepts were compared to provide a useful decision support.

Simulation results are in turn evaluated with regards to eight KPIs: acceptance, emissions, area usage, costs, implementation potential, traffic impacts, economic feasibility, ecological break-even-point. For the evaluation, the *basecase* is considered as the starting point and trends on the KPIs through utilization of logistic concepts are derived.

After all possible scenarios are simulated, evaluation data is combined in a database queryable through a web-application. Users can now select a research area, a concept (possibly modify it through parameters) and are presented with the evaluated simulation results.

While this workflow produces good results for pre-defined areas, adaption to new areas is labor intensive and slow, diminishing the usefulness for other parties. Furthermore, since all possible combinations of areas, concepts and parameters need to be simulated a priori, timely simulation requires a large amount of computational resources. A optimized process would allow the users to select the area of interest, a specific concept and then receive the simulation results on-demand (either through a new simulation study or previously performed simulations). Consequently, we designed this

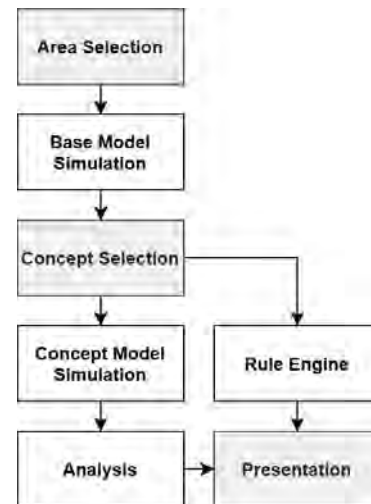


Figure 2: Planned workflow for the *USEFUL*-Application. User interactive parts marked in grey.

concept of an automated simulation framework. This goal-process is shown in Figure 2.

The improved workflow also utilizes a rule-based expert-system to give impact estimations almost instantly, as opposed to the simulation, where turnaround times of multiple days, if not weeks are to be expected. To achieve this workflow, major challenges must be solved, most importantly the automatic creation of models based on automatically collected data. While the previous process already uses automated tools at each step in the workflow, data transfer between steps was handled manually, increasing the total time needed. Implementing the new workflow requires a clear definition of software interfaces, processes and deployment into an easily scalable environment.

3 Service-oriented cloud-based workflow

Service orientation and cloud usage are computing patterns that address multiple issues also encountered in the simulation workflow of *USEFUL*. Service orientation allows the easy integration of quickly evolving tools into combined workflows while providing abstract interfaces, decoupling different steps of software processes [13]. This in turn allows the easy combination of many different software tools into a combined application. Cloud-based computing provides quickly scalable infrastructure by combining different applications

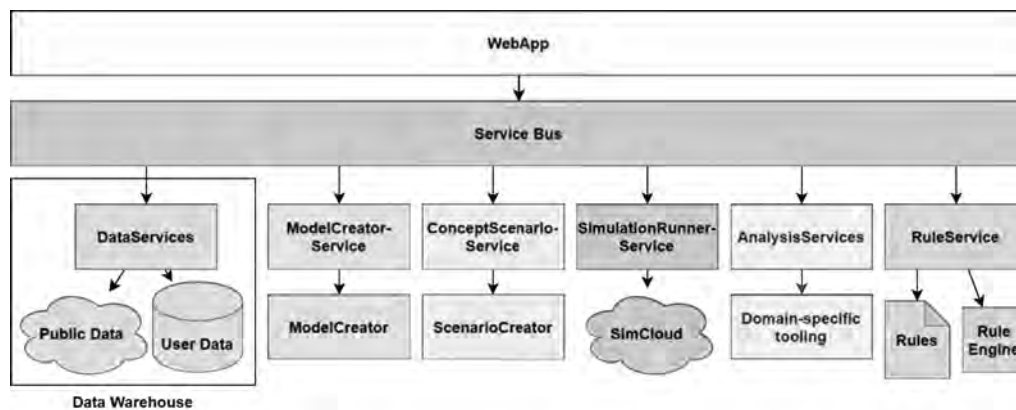


Figure 3: Proposed Architecture for the system. Each step of the previous workflow of the project is encapsulated into a service, which can be invoked in order from the user-accessible web-application

on the same hardware, therefore reducing the cost for hosting of applications.

With a cloud-based simulation service, users can request simulations ad-hoc, paying only for the computational power that is actually needed to run the simulation, instead of investing in expensive simulation hardware. Since primary user groups are located in communal administration and other non-software firms, this approach increases the attractiveness of the solution to the end user. While the architecture is aimed at a cloud deployment, it is not necessary to rely upon an external cloud provider. Through the usage of a private cloud, the entire software can run on-premise, if e.g. data security or privacy requirements dictate it.

In general service-orientation is implemented by encapsulating specific tasks of the workflow into independent services that can be recombined for other workflows. Figure 3 shows the proposed service-oriented architecture for the automated solution.

The workflow of the proposed system is controlled by the already existing WEBAPP, providing user interaction and service orchestration. Since only a single workflow – the evaluation of logistic scenarios – is to be expected, a flexible workflow-engine is not necessarily needed and implementation complexity can be reduced. In case the workflow changes for the different simulations, an increased workload to implement the changes is to be expected, but does not outweigh the reduction of implementation complexity. As the WEBAPP has already been developed in *USEFUL*, it can easily be adapted to the necessary changes.

The SERVICE BUS provides stable service endpoints for usage through the WEBAPP (or other appli-

cations) and connects them to the service implementations. This effectively decouples declaration and definition of different tasks, allowing easy redevelopment of different parts of the overall workflow without leading to widespread changes. Many different existing solutions can be adapted for this particular purpose. Communication between the SERVICE BUS and other components is technology-dependent and can be implemented in multiple different ways. As the chosen simulation tool MATSim utilizes xml-files for configuration and modelling, data transfers will also utilize the xml-standard to achieve design homogeneity.

The SERVICES implement the steps of the workflow and utilize technologies independent of the rest of the application. This allows technology selection according to the domain-specific needs, as well as flexibility when services are updated to new methods. A description of the services is given in the following sections.

Some cross-cutting concerns are life cycle management and security, as these are needed at all levels of the application. While life-cycle management needs to be decentralized due to the technical nature of the domain, security can be handled at the SERVICE BUS and deployment environment. E.g. access to services from external sources can be blocked by a firewall and internal compartmentalization is achieved by extending the existing role-based-access-control to service bus level.

3.1 DataServices

The data services, are responsible of adapting different sources of data into the system. To create simulation models for urban logistics, different information

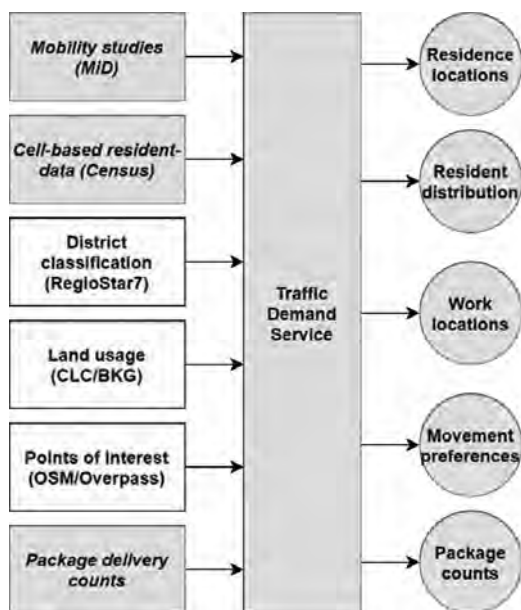


Figure 4: High-level view of the Demand data service. White background marks machine readable sources, slanted writing marks manually interpreted Data.

sources are summarized to create the traffic supply as well as the traffic demand.

As these data sets are built from many different sources, the services are responsible for aggregating this data and providing the necessary information in a standardized format. The data services consolidate the processed data in a data warehouse and allocate it to the associated services for further processing. While some data might be pulled from standardized web services, other need to be manually interpreted, prepared and provided in a suitable format. Consequently most of the technical and logical complexity of handling different data is localized in these services.

Figure 4 shows a high-level view of one of the main data services, the TRAFFIC DEMAND SERVICE. This Service is responsible for providing information about traffic demand and population behavior, which in turn is used to generate traffic demand within the MODELCREATORSERVICE. Information for this service is gathered from public APIs like OSM/Overpass, public statistical data, manually extracted from documents, and non-public studies. Data pulled from public APIs can easily be converted into the necessary XML structure, utilized by consuming services. Other Data, like the aforementioned public statistical data, needs manual conversion and own storage infrastructure, as doc-

ument structures vary vastly between different sources or iterations of the same studies. For example, the study *Mobility in Germany 2017*² added a new excel-based tool for data exploration, which was not available for the studies predecessor, created in 2008. While automatic extraction of the study is possible, the created tool would only work for this iteration of the study. The resulting effort/effect-balance is not favourable for automation. To make manually extracted data available for the automatic workflow, it is entered into a database, which provides a machine readable data format. Since the TRAFFIC DEMAND SERVICE collects data from different sources and provides a unified interface for consumers, it can be viewed as a data warehouse.

As part of the modelling process, the traffic demand service also provides data for the automated model calibration service. Thus, mode share information of the simulation area as well as other mobility parameters must be specified and provided by the user. An essential element for the quality of the model is the availability of real traffic data from the simulation area particularly consisting of vehicle count data of different sections of the network. While this detector data can be accessed online for specific network connections, it is often not directly available. Therefore, this missing component must be provided by the user.

3.2 ModelCreatorService

After all necessary input data is collected and processed into the data warehouse, the MODELCREATORSERVICE is executed. This service contains three consecutive modeling steps that gradually generate an agent-based transport model for individual traffic along with a service-based model for commercial traffic (Figure 5).

In the first step, a synthetic population consisting of households with corresponding agents is generated based on the grid cells that are deposited in the data warehouse. Grid cells are selected according to the area of interest included in the execution request. Each of the agents in the population represents one person living in the specific cell and is assigned with certain socio-demographic attributes, such as gender or age, as well as attributes that impact the mobility, such as the possession of a driving licence or public transport ticket. In addition to these agents, companies are synthesized as location for work activities and as instances that generate commercial or logistic traffic. The companies are

²<http://www.mobilitaet-in-deutschland.de/publikationen2017.html> - last visited: 25.04.2022

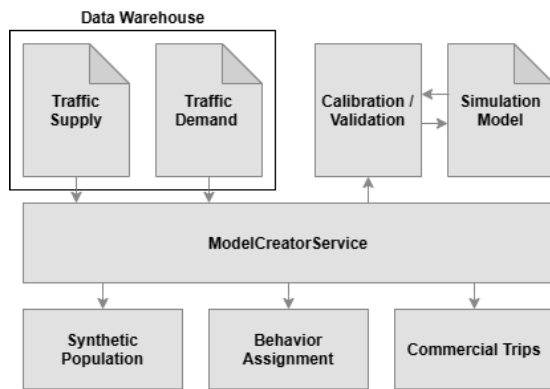


Figure 5: Architecture of the ModelCreatorService

differentiated by industry sectors and also obtain attributes such as the number of vehicles and the number of agents working at that location.

In the second step of the model creation, each agent is assigned a mobility behavior corresponding to the predefined attributes. The behavior is defined by a sequence of activities the agent performs during a day and the necessary trips that the agent must perform to reach those activities. These patterns are provided by the data services and also contain additional information such as the transportation modes used and the travel distance for the trips. The location choice for the activities is then carried out on the basis of these information along with land-use data.

The last element of the MODELCREATORSERVICE is the generation of commercial trips. In this process, the defined vehicles for each company are assigned with a sequence of services they have to operate on the simulated day. The services act in a similar way to the activities of the agents and are linked to the population and other companies. For example, if a craftsman performs a service for an agent, the agent has to be at home.

A special focus is laid on logistical aspects of the model. Specifically, the parcel demand is used to generate service sequences for parcel delivery vehicles of multiple delivery services. The resulting vehicle routing problem is solved using a ruin-and-recreate-algorithm and resulting trips are assigned to the vehicles [14].

The output of the MODELCREATORSERVICE consists of several independent XML-files, representing the model in its entirety. This model then has to be calibrated and validated using additional data sources from the warehouse that were not used for the model creation.

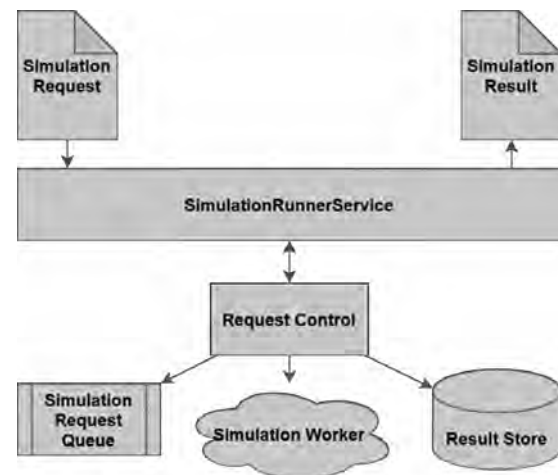


Figure 6: Simulation Runner Service responsible for asynchronous execution of simulation requests.

3.3 ConceptScenarioService

After the *basecase* has been created, further changes to the model are necessary to inspect different logistics concepts. In the CONCEPTSCENARIOSERVICE sub-components make changes to the model according to the concepts. For example, some of the traffic resulting from grocery runs is replaced by a centralized grocery distribution system. To add a new concept, new components can easily be added to the service, without changing other parts of the system. Since the logistical concepts are not in service at the research area, validation of the changed model proves difficult.

3.4 SimulationRunnerService

The simulation service is responsible for scheduling and executing the simulations. A detailed analysis through simulations can be provided by utilizing a worker cloud where simulation tasks can be asynchronously executed.

As shown in Figure 6, SIMULATION REQUESTS are accepted by the SIMULATIONRUNNERSERVICE, where they can be stored in a queue until resources are available for execution. The SIMULATION REQUEST contains XML-Files which were generated in the MODELCREATORSERVICE along with configuration parameters for the simulation, which have to be provided by the user. Data is then fed into the SIMULATION WORKERS, which execute MATSim simulations on dynamically scaling infrastructure e.g. through the use of

Docker containers. After the simulation runs are finished, data is pushed into a RESULT STORE for subsequent requests to prevent repeated simulation of the same model/parameter-combination.

3.5 AnalysisServices

The final step before presenting concept impacts to the user is the analysis of simulation results. The analysis is carried out by applying models for different KPIs to parts of the simulation output (e.g. additional traffic caused by the given concept). As models are created by domain experts, an integration for widely used tools like Microsoft Excel is required. The different models compare information from the simulation of the modified models to the *basecase* and derive trends for KPIs. This allows the easy-to-understand representation in the final visualization for the user.

One example of ANALYSIS SERVICES is the AREA MODEL, which gives an estimation on the expected change of area needed for the implementation of a given logistical concepts. The AREA MODEL uses information like the amount and type of delivery vehicle used to implement a given concepts and then calculates additional area needs for e.g. construction of micro hubs. This is in turn compared to the reduction of e.g. parking spaces to calculate the net difference. Since the effects of concepts are also interconnected, the AREA MODEL feeds its result directly into the COST MODEL, which predicts cost changes.

3.6 RuleService

As an alternative to the time-consuming simulation path (shown in Figure 2), a rule-based expert system can be utilized to estimate the impacts KPIs. The RULESERVICE uses the JSONRULEENGINE and multiple sets of Rules to estimate Results, as each concept reacts differently to structural and behavioural peculiarities of different concepts. While not providing the accuracy achieved by the simulations, the RULESERVICE allows the preliminary study of concepts effects on non-simulated areas to identify promising combinations for further study.

The rules are built upon previous simulation results for each logistics concept and model the connection between concept effects and structural attributes of areas. As no high accuracy is expected until a large amount of simulations have been carried out, the first iteration of the rules engine only predicts three categories of

change, instead of a numerical scale. As more and more simulations are carried out, new results are incorporated into the rules and model prediction can be increased by including the new results into the rule creation process. However, the rule engine cannot predict changes in radically new areas or for unknown logistical concepts.

3.7 WebApp

After the impacts of the concept have been evaluated, a web-based decision support tool presents the results to the user. This WEBAPP is built upon Laravel³, using a modern design to attract users. Functionalities for selecting areas and concepts as well as dynamical presentation of results are already implemented, therefore only additions to already existing components are necessary. Furthermore, the targeted workflow is almost identical to the previous workflow from a user perspective, so no major design changes result. More information on the WEBAPP and the rule-based expert system can be found in [15].

4 Remaining challenges

While the technical integration of all existing tools can easily be achieved through the use of available software solutions, some procedural issues remain. First and foremost the validation of the simulation models remains problematic, as multiple data sources for similar information are required. Data has proven to be relatively scarce, as privacy of residents or workers needs to be protected and some data are simply not collected. Furthermore, validation is currently a time-consuming, manual process, that has not been fully automated yet.

Another challenge is the calibration of the simulation models, as this is again a manual process, requiring a human-in-the-loop approach to properly interpret information and adapt parameters accordingly. An automated process could involve a meta-optimization approach such as evolutionary algorithms, but would in turn increase resource demand and runtime.

Since open data projects are utilized, data quality is inconsistent, as data tends to be biased. [16] state one of the main reasons for this is the inhomogeneous contribution patterns of its contributors. This issue is not fixable through the proposed system, as automatic data quality improvement is out of scope for its requirements.

³<https://laravel.com>

5 Conclusion

In this paper an architecture for the automation of simulation and decision support has been presented. Built upon a previously developed workflow, the system aims to allow users to evaluate novel logistics concepts in specific research areas. The system shall provide rough estimates on effects on key performance indicators of traffic like emissions and population behaviour. This allows users to evaluate logistics concepts in theory and aids in preliminary selection, reducing the need for cost-intensive prototype-projects in the real world.

While the technical aspects of the proposed system do not pose showstopping challenges, domain specific workflows, like simulation validation need to be developed further to implement the system. In future work a prototypical implementation as well as the definition and evaluation of automated processes for simulation validation and calibration are planned.

Acknowledgement

This work was supported by the Federal Ministry of Education and Research of Germany (project USEFUL, grant no. 03SF0547, and USEFUL-XT, grant. no 03SF0609). We would like to thank our colleagues from the other institutions and the City of Hannover.

References

- [1] Kunze O. Replicators, Ground Drones and Crowd Logistics A Vision of Urban Logistics in the Year 2030. *Transportation Research Procedia*. 2016;19:286–299. Transforming Urban Mobility. mobil.TUM 2016. International Scientific Conference on Mobility and Transport. Conference Proceedings.
- [2] Dablanc L. E-commerce trends and implications for urban logistics. *Urban logistics Management, policy and innovation in a rapidly changing environment*. 2019;pp. 167–195.
- [3] Lagorio A, Pinto R, Golini R. Research in urban logistics: a systematic literature review. *International Journal of Physical Distribution & Logistics Management*. 2016;46:908–931.
- [4] Pump R, Gohe S, Trott M, auf der Landwehr M, Koschel A, Ahlers V, Gusig L, von Viebahn C. Towards Extending USEFUL-ness for Urban Logistics with Service-orientation. In: *SERVICE COMPUTATION 2021*, SERVICE COMPUTATION, International Conferences on Advanced Service Computing. Porto, Portugal: IARIA. 2021; pp. 13–19.
- [5] Bienzeisler L, Lelke T, Wage O, Huck LM, Friedrich B. Uncertainty and Variability Analysis of Agent-Based Transport Models. *Transportation Research Procedia*. 2022;62:719–726.
- [6] Horni A, Nagel K, Axhausen K, eds. *Multi-Agent Transport Simulation MATSim*. London: Ubiquity Press. 2016.
- [7] Bienzeisler L, Lelke T, Wage O, Thiel F, Friedrich B. Development of an Agent-Based Transport Model for the City of Hanover Using Empirical Mobility Data and Data Fusion. *Transportation Research Procedia*. 2020; 47:99–106.
- [8] de Stasio C, Fiorello D, Fermi F, Martino A, Hitchcock G, Kollamthodi S. On-line Tool for the Assessment of Sustainable Urban Transport Policies. *Transportation Research Procedia*. 2016;14:3189–3198. Transport Research Arena TRA2016.
- [9] Dossou PE, Vermersch A. Development of a decision support tool for sustainable urban logistics optimization. *Procedia Computer Science*. 2021;184:476–483. The 12th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 4th International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.
- [10] Hörl S, Balac M. Synthetic population and travel demand for Paris and Île-de-France based on open and publicly available data. *Transportation Research Part C: Emerging Technologies*. 2021;130:103291.
- [11] Reiffer A, Heilig M, Kagerbauer M, Vortisch P. Microscopic Demand Modeling of Urban and Regional Commercial Transport. *Procedia Computer Science*. 2018;130:667–674.
- [12] Jlassi S, Tamayo S, Gaudron A. Simulation Applied to Urban Logistics: A State of the Art. In: *City Logistics 3*. John Wiley & Sons, Ltd. 2018; pp. 65–87.
- [13] Rosen M, Lublinsky B, Smith KT, Balcer MJ. *Applied SOA: service-oriented architecture and design strategies*. John Wiley & Sons. 2012.
- [14] Zilske M, Joubert JW. Freight Traffic. In: *The Multi-Agent Transport Simulation MATSim*, edited by Horni A, pp. 155–156. Ubiquity Press. 2016;.
- [15] Heumann M, Pump R, Breitner MH, Koschel A, Ahlers V. Towards Sustainable Transport: A Strategic Decision Support System for Urban Logistics Operations. *Innovation Through Information Systems*. 2021;pp. 367–381.
- [16] Basiri A, Haklay M, Foody G, Mooney P. Crowdsourced geospatial data quality: challenges and future directions. *International Journal of Geographical Information Science*. 2019;33(8):1588–1593.

SNE SIMULATION NOTES EUROPE

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

SNE seeks to serve scientists, researchers, developers and users of the simulation process across a variety of theoretical and applied fields in pursuit of novel ideas in simulation and to enable the exchange of experience and knowledge through descriptions of specific applications. **SNE** puts special emphasis on the overall view in simulation, and on comparative investigations, as benchmarks and comparisons in methodology and application. Additionally, **SNE** welcomes also contributions in education in / for / with simulation.

SNE is the official membership journal of EUROSIM, the federation of European simulation societies and simulation groups, so **SNE** is open for postconference publication of contributions to conferences of the EUROSIM societies, and for special issues organized by EUROSIM societies.

SNE, primarily an electronic journal, follows an open access strategy, with free download in basic layout. Members of EUROSIM societies are entitled to download SNE in an elaborate and extended layout. Print **SNE** is available for specific groups of EUROSIM societies, and as print-on-demand from TU Verlag, TU Wien (www.tuverlag.at).



www.sne-journal.org

ASIM SST 2022

TU Wien, 25.-27.7.2022

---- *eine Tagung wie es früher einmal war* ----



ALGORITHMIC ART BY VLATKO CERIC

VCERIC.NET

ISBN print
ISBN 978-3-903311-19-0
TU Verlag, Wien, 2022

ISBN ebook
ISBN 978-3-901608-97-1
ARGESIM Verlag, Wien, 2022

DOI ID
10.11128/arep.20