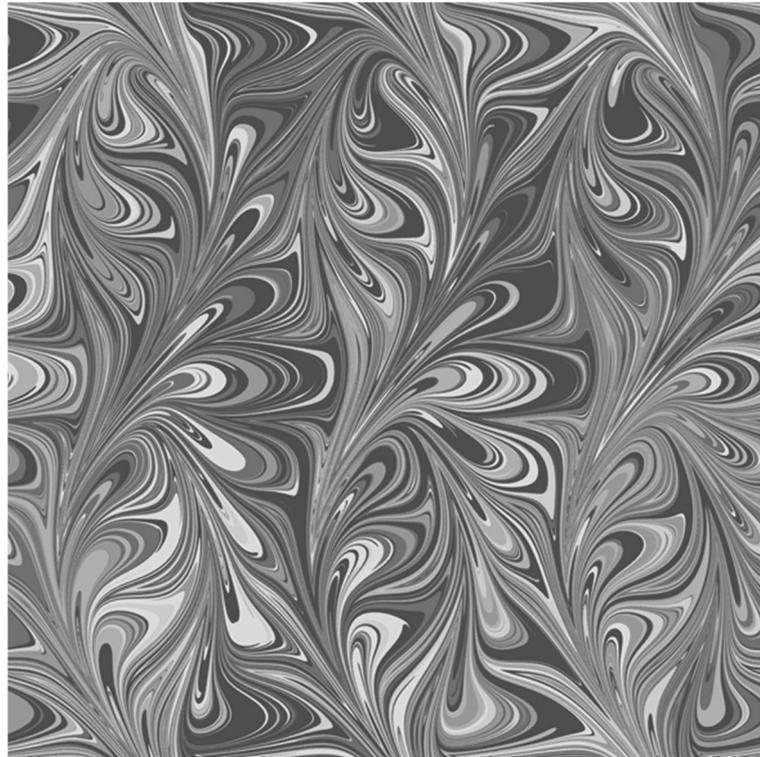


# ASIM WORKSHOP 2023

## STS / GMMS / EDU

### Proceedings Langbeiträge

REPORT 21



## ASIM Fachgruppenworkshop 2023

STS Simulation Technischer Systeme  
GMMS Grundlagen und Methoden in der Simulation  
EDU Simulation und Edukation



OTTO VON GUERICKE  
UNIVERSITÄT  
MAGDEBURG

INF

6. - 7. März 2023

Herausgegeben von Claudia Krull; Walter Commerell,  
Umut Durak, Andreas Körner, Thorsten Pawletta

ISBN ebook 978-3-903347-61-8  
ARGESIM Report 21  
[www.argesim.org](http://www.argesim.org)

DOI 10.11128/arep.21  
ASIM Mitteilung 185  
[www.asim-gi.org](http://www.asim-gi.org)



ASIM



ASIM



ASIM



# ASIM Books – ASIM Book Series – ASIM Buchreihen

Monographs / Proceedings

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

Book Series Fortschrittsberichte Simulation – Advances in Simulation

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

**ASIM WORKSHOP 2023**

**STS/GMMS/ EDU**

**Proceedings Langbeiträge**

**ASIM Fachgruppenworkshop 2023**

**STS Simulation Technischer Systeme**

**GMMS Grundlagen und Methoden**

**in Modellbildung und Simulation**

**EDU Simulation und Edukation**

**6.3. - 7.3. 2023, Universität Magdeburg**

Herausgegeben von

Claudia Krull; Walter Commerell, Umut Durak,

Andreas Körner, Thorsten Pawletta

ISBN ebook 978-3-903347-61-8

ASIM Mitteilung 185

ARGESIM Verlag, Wien, 2023

DOI 10.11128/arep.21

ARGESIM Report 21

[www.argesim.org](http://www.argesim.org)

## Bibliographic Data:

Publisher: ARGESIM Publisher, Vienna

Title: Proceedings Langbeiträge ASIM Workshop 2023 STS/GMMS/EDU

Subtitle: ASIM Fachgruppenworkshop 2023 - GMMS Grundlagen und Methoden in Modellbildung und Simulation, STS Simulation Technischer Systeme, EDU Simulation und Edukation - 6.3.-7.3. 2023, Universität Magdeburg

Author(s):

Editor(s): Claudia Krull; Walter Commerell, Umut Durak, Andreas Körner, Thorsten Pawletta

Series: ARGESIM Reports

Series Editors: Felix Breitenecker, Thorsten Pawletta, ASIM

Volume: ARGESIM Report no. 21

ISBN ebook: 978-3-903347-61-8 ARGESIM Verlag

DOI: 10.11128/arep.21

Publication Date: March 15, 2023

Number of Pages: 158 + iv pages

Cover: Digital Marbling Graphics by Graham Horton ©

Copyright © 2023 ASIM & 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 EUROSIM (in consideration of copyright regulations for related conference publications) and monographs on system simulation (mainly PhD theses).

## About ARGESIM

ARGESIM is a non-profit society generally aiming for dissemination of information on system simulation from research via development to applications of system simulation. ARGESIM is closely co-operating with EUROSIM, the Federation of European Simulation Societies, and with ASIM, the German Simulation Society. ARGESIM is an 'outsourced' activity from the Mathematical Modelling and Simulation Group of TU Wien, there is also close co-operation with TU Wien (organisationally and personally).

*ARGESIM Publisher* organizes publishing activities, with ISBN roots 978-3-901608-xx-y and 978-3-903347-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](http://www.sne-journal.org)
- Organisation and Publication of the ARGESIM Benchmarks for Modelling Approaches and Simulation Implementations → [www.argesim.org/benchmarks/](http://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/](http://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](http://www.mathmod.at)
- Administration and support of ASIM (German Simulation Society → [www.asim-gi.org](http://www.asim-gi.org)) and of EUROSIM (Federation of European Simulation Societies → [www.eurosim.info](http://www.eurosim.info))

ARGESIM – Arbeitsgemeinschaft Simulation News – Working Committee Simulation News – SNE Publication

Mommsengasse 19/8, 1040 Vienna, Austria; Tel +43-1-58801-10111, -10115; Fax +43-1-58801-910111

Email: [office@argesim.org](mailto:office@argesim.org), [office@sne-journal.org](mailto:office@sne-journal.org); WWW: [www.argesim.org](http://www.argesim.org), [www.sne-journal.org](http://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

Simulationstechnologie ist in vielen Bereichen und insbesondere im Entwicklungsprozess technischer Systeme fest verankert. Der jährliche Workshop der **ASIM/GI**-Fachgruppen **STS** (Simulation Technischer Systeme), **GMMS** (Grundlagen und Methoden in Modellbildung und Simulation) und **EDU** (Simulation und Edukation) bietet eine breite Plattform, um sowohl den aktuellen technologischen Stand als auch zukünftige Chancen in Theorie, Praxis und Ausbildung zu beleuchten.

Der **ASIM Workshop STS/GMMS/EDU** fand 2023 wieder als Präsenzveranstaltung statt, an der Fakultät für Informatik der Otto-von-Guericke-Universität Magdeburg, und bot eine offene Atmosphäre für lebendige Diskussionen und den Informations- und Erfahrungsaustausch zwischen Fachleuten aus Hochschulen, Forschungseinrichtungen und der Industrie.

Dank vieler engagierter Autorinnen und Autoren wurden hochqualitative Beiträge mit aktuellen Forschungsergebnissen eingereicht. Neben den traditionellen Inhalten des Workshops waren wieder aktuelle Themen wie „Maschinelles Lernen“ vertreten, aber auch aktuelle Anwendungen wie autonomes Fahren oder Umgang mit dem Klimawandel. Zwei interessante Keynote Vorträge bildeten den Rahmen des Programms. Sie beleuchteten die Themen „Saisonaler Energiespeicher auf Basis von Kalk“ und „Digital Marbling“, und führten zu angeregten Diskussionen im Nachgang.

Die zwei Tagungsbände enthalten die Beiträge des Workshops, die unter anderem folgende Themenschwerpunkte behandeln:

- Modellbasierte Mechatronikentwicklung
- Simulation in der Elektronikentwicklung
- Mathematische Verfahren in Modellbildung und Simulation
- Grundlagen und Methoden in Modellbildung und Simulation
- Simulation Cyber-Physischer Systeme
- Simulation Technischer Systeme
- Angewandte Simulationen
- Simulation und Digitalisierung in Aus- und Weiterbildung

Dieser Tagungsband, **Proceedings Langbeiträge ASIM Workshop 2023 STS/GMMS/EDU**, ISBN ebook 978-3-903347-61-8, DOI 10.11128/arep.21, ARGESIM Report 21, ASIM Mitteilung 185, publiziert die 17 Langbeiträge, die vom Programmkomitee begutachtet und angenommen wurden, mit individueller DOI.

Kurzbeiträge und Abstract-Beiträge sind zu finden in **Kurzbeiträge & Abstract-Beiträge ASIM Workshop 2023 STS/GMMS/EDU**, ISBN ebook 978-3-903347-62-5, DOI 10.11128/arep.22, ASIM Mitteilung 186. Für Form und Inhalt der in den Tagungsbänden enthaltenen Beiträge sind die Autoren selbst verantwortlich.

Entsprechend ASIMs Publikationsstrategie sind Tagungsband und Einzelbeiträge als Open Access in Basisversion auf [www.asim-gi.org](http://www.asim-gi.org) verfügbar. ASIM-Mitgliedern stehen die Vollversionen sowie die Sammlung der Vortragsfolien mit individuellem Login zur Verfügung (für Workshopteilnehmer auf der Tagungswebsite [www.asim-gi.org/magdeburg2023/programm](http://www.asim-gi.org/magdeburg2023/programm) bis Ende 2023 mit Gruppenlogin).

Für die Unterstützung und Mithilfe bei der Organisation des diesjährigen Workshops möchte ich mich sehr herzlich bei Frau Madeleine Breitzkreuz, Herrn Pascal Krenckel, Herrn Umut Durak und Herrn Thorsten Pawletta, sowie bei allen weiteren Beteiligten vor Ort am Institut für Simulation und Graphik bedanken. Außerdem gilt mein Dank den Reviewern für die tatkräftige Unterstützung bei der Auswahl und Verbesserung der Einreichungen.

Claudia Krull

Magdeburg, im März 2023



# Inhaltsverzeichnis – List of Content

## ASIM Workshop 2023 STS/GMMS/EDU – Langbeiträge

<b>EDUKATION und SIMULATION – EDUCATION and SIMULATION</b>	<b>1</b>
Implementing Thermodynamic Cyclic Processes Using the DLR Thermofluid Stream Library. <i>Peter Junglas</i> ; DOI: 10.11128/arep.21.a2102	3
Durchführung eines Simulationsprojektes im Simulationspraktikum. <i>Heiderose Stein, Tobias Uhlig, Oliver Rose</i> ; DOI: 10.11128/arep.21.a2106	11
Studieren mit digitalen Medien. <i>Nico Marten, Kathrin Thiele</i> ; DOI: 10.11128/arep.21.a2115	19
Industrial Methods and Digital Tools for Craft Professionals – A use case deploying Discrete Event Simulation and Virtual Reality. <i>Bastian Prell, Jörg Reiff-Stephan</i> ; DOI: 10.11128/arep.21.a2121	27
<b>SIMULATION TECHNISCHER SYSTEME – SIMULATION of TECHNICAL SYSTEMS</b>	<b>35</b>
Entwurf einer Modellbibliothek für die kostengünstige Funktionsentwicklung mechatronischer Systeme. <i>Sven Jacobitz, Jie Zhang, Xiaobo Liu-Henke</i> ; DOI: 10.11128/arep.21.a2119	37
Entwicklung der adaptiven lokalen Trajektorienplanung zur Realisierung des kollisionsfreien Fahrens. <i>Jie Zhang, Xiaobo Liu-Henke, Thomas Vietor</i> ; DOI: 10.11128/arep.21.a2107	45
Gekoppelte Simulation des Einspritz- und Verbrennungsvorgangs eines Industrie-Dieselmotors. <i>Daniel Jörss, Maximilian Ringel, Bert Buchholz, Christian Fink</i> ; DOI: 10.11128/arep.21.a2114	55
Automatisches Mapping des dynamischen Umfelds in cyper-physischen Systemen. <i>Taihao Li, Marian Göllner, Sven Jacobitz, Xiaobo Liu-Henke</i> ; DOI: 10.11128/arep.21.a2120	63
Konzept eines ganzheitlichen, hochflexibel konfigurierbaren HiL-Prüfsystems für den Test autonomer Fahrfunktionen. <i>Xiaobo Liu-Henke, Marian Göllner, Sven Jacobitz, Jie Zhang</i> ; DOI: 10.11128/arep.21.a2123	71
Towards a Scenario Toolkit for Autonomous Systems. <i>Ahmad Naja, Siddhartha Gupta, Umut Durak, Sven Hartmann</i> ; DOI: 10.11128/arep.21.a2117	77
Critical Scenario Techniques for Automated Vehicles: Literature Review. <i>Kaan Akgün, Umut Durak, Siddhartha Gupta</i> ; DOI: 10.11128/arep.21.a2124	87

Iterative Scenario-Based Testing in an Operational Design Domain for Artificial Intelligence Based Systems in Aviation. <i>Bojan Lukic, Jasper Sprockhoff, Alexander Ahlbrecht, Siddhartha Gupta, Umut Durak;</i> DOI: 10.11128/arep.21.a2108	95
Simulation-based Verification of Functions for Autonomous Drones. <i>Hamza Ghezali, Siddhartha Gupta, Umut Durak;</i> DOI: 10.11128/arep.21.a2118	103
<b>GRUNDLAGEN und METHODEN in MODELLBILDUNG und SIMULATION – BASIC PRINCIPLES and METHODS in MODELLING and SIMULATION</b>	<b>115</b>
Ein Gütemaß für Virtuelle Stochastische Sensoren - Beobachtbarkeit Typ-3. <i>Pascal Krenckel, Claudia Krull;</i> DOI: 10.11128/arep.21.a2116	117
Integrating Reinforcement Learning and Discrete Event Simulation Using the Concept of Experimental Frame: A Case Study With MATLAB/SimEvents. <i>Thorsten Pawletta, Jan Bartelt;</i> DOI: 10.11128/arep.21.a2122	125
Dynamic Time Warping und Synthesedaten zur Validierung von Seq2Seq für die Simulation. <i>Benjamin Wörrlein, Steffen Straßburger;</i> DOI: 10.11128/arep.21.a2113	133
Simulationsbasierte Untersuchung von Energieflexibilität bei der Optimierung in der Hinterlandentwässerung. <i>Marvin Hempel, Jens Heger;</i> DOI: 10.11128/arep.21.a2109	143
<b>HAUPTVORTRÄGE – INVITED LECTURES</b>	<b>151</b>
Saisonalenergiespeicher auf Basis von Kalk – Experimentelle Entwicklung und Systemsimulation (Abstract) <i>Marc Linder, Michael Böhm</i>	153
Digital Marbling: Simulation of a Traditional Bookbinder’s Craft (Abstract) <i>Graham Horton</i>	155
<b>Autorenindex – Author Index</b>	<b>157</b>

Langbeiträge (L) sind zu finden im

**ARGESIM Report 21 Proceedings Langbeiträge ASIM Workshop 2023 STS/GMMS/EDU**

ISBN ebook 978-3-903347-61-8, DOI 10.11128/arep.21, ARGESIM Report 21, ASIM Mitteilung 185

Kurzbeiträge (K) und Abstract-Beiträge (A) sind zu finden im

**ARGESIM Report 22 Kurzbeiträge & Abstract-Beiträge ASIM Workshop 2023 STS/GMMS/EDU**

ISBN ebook 978-3-903347-62-5, DOI 10.11128/arep.22, ARGESIM Report 22, ASIM Mitteilung 186

Abstracts der Hauptvorträge sind im **ARGESIM Report 21** und im **ARGESIM Report 22** zu finden

# Edukation und Simulation

## Education and Simulation

Implementing Thermodynamic Cyclic Processes Using the  
DLR Thermofluid Stream Library (L)  
*Peter Junglas*

Durchführung eines Simulationsprojektes im Simulationspraktikum (L)  
*Heiderose Stein, Tobias Uhlig, Oliver Rose*

Studieren mit digitalen Medien (L)  
*Nico Marten, Kathrin Thiele*

Industrial Methods and Digital Tools for Craft Professionals – A use case deploying  
Discrete Event Simulation and Virtual Reality (L)  
*Bastian Prell, Jörg Reiff-Stephan*

Digitalisierung in der Lehre durch Interaktive Notizbücher und automatisch  
bewertete Aufgaben (A)  
*Andreas Apostolatos, Sebastian Groß*

Langbeiträge (L) sind zu finden im

**ARGESIM Report 21 Proceedings Langbeiträge ASIM Workshop 2023 STS/GMMS/EDU**

ISBN ebook 978-3-903347-61-8, DOI 10.11128/arep.21, ARGESIM Report 21, ASIM Mitteilung 185

Kurzbeiträge (K) und Abstract-Beiträge (A) sind zu finden im

**ARGESIM Report 22 Kurzbeiträge & Abstract-Beiträge ASIM Workshop 2023 STS/GMMS/EDU**

ISBN ebook 978-3-903347-62-5, DOI 10.11128/arep.22, ARGESIM Report 22, ASIM Mitteilung 186

Abstracts der Hauptvorträge sind im **ARGESIM Report 21** und im **ARGESIM Report 22** zu finden



# Implementing Thermodynamic Cyclic Processes Using the DLR ThermoFluid Stream Library

Peter Junglas\*

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

\*[peter@peter-junglas.de](mailto:peter@peter-junglas.de)

**Abstract.** Simulation programs modeling cyclic processes can be used in thermodynamics lectures to promote understanding. Modelica-based simulation environments are a good starting point for the development of such programs, but the handling of the corresponding thermo-fluid standard library is very difficult for non-experts. The recently presented DLR ThermoFluidStream Library is a good alternative, that is easier to use. It provides most components that are needed in typical cases and includes full access to the Modelica media library.

It will be shown, how to use the ThermoFluidStream Library to create examples ranging from the simple Otto and Diesel cycles over the basic Joule-Brayton and Ericsson processes to the water/steam based Clausius-Rankine cycle. Though the construction of concrete processes with given thermodynamic state values and mass flow still requires some effort, one can apply a systematic approach to create working models for teaching purposes.

## 1 Introduction

Thermodynamics is a difficult subject for many engineering students, mainly because of its partly unintuitive nature using abstract notions like enthalpy and entropy. To promote understanding simulation programs can be used, which allow to “experiment” with state changes or complete cyclic processes, such as the collection of Java applets described in [1].

But the construction of such programs is a tedious and time-consuming task, especially if one wants to include examples that use more complex media than the simple ideal gas with constant specific heat capacity. Instead of writing such programs from scratch, one could use a simulation environment to describe the example models, and leave the actual computation to its solver. Modelica [2] with its physical modeling approach seems to be a good starting point, es-

pecially since a comprehensive free model library is available that describes the thermodynamic behaviour of many useful media [3]. Therefore it will be used in the following to build models of the standard processes that are examined in most introductory thermodynamics courses: the Otto and Diesel processes for closed systems and the Joule-Brayton and Ericsson processes for open systems [4, 5]. These models should run on any Modelica platform, especially on the freely available OpenModelica environment [6], and can be employed directly in a thermodynamics course.

The Modelica Standard Library (MSL) already contains an elaborate thermo-fluid library that provides basic components for one-dimensional thermo-fluid flow in pipes, vessels or machines [7]. But due to its very general approach it is much too complicated for the simple didactical applications addressed here. Additionally, corresponding models recurrently fail to run for reasons that are hard to find for non-specialists [8].

The recently presented DLR ThermoFluidStream Library [9] (“ThermoDLR”) seems to provide just the level of detail that is needed here: On the one hand it uses the full Modelica media library, on the other hand it offers components for vessels and machines that are much easier to handle than their MSL counterparts. And, most importantly, it uses a very clever, physically motivated scheme to achieve a high robustness [10] that should lead to models that generally run without delicate fine-tuning. This makes it a promising foundation for the construction of didactical examples.

A similar, but simpler and more limited approach has been presented in [11], which also describes a Modelica library for thermodynamical examples (“ThermoSimT”). Since its focus is on teaching Modelica, it does not use the complex standard Media library, but a greatly simplified version. Especially its steam/water model only shows basic modeling principles, but is of no practical use. Nevertheless it allows

to easily build models of all standard processes and will be used as a benchmark to assess the ease of use and versatility of the new ThermoDLR version.

In the following we will briefly describe the basic ideas of the ThermoSimT library, which provides models of cyclic processes for ideal gas with constant or temperature-dependent specific heat capacity and of the steam/water based Clausius-Rankine cycle. Then we will use the ThermoDLR library to implement similar models, utilizing the Media library to get valid practical results. Due to the didactical purpose and since we are only interested in equilibrium behaviour, the models have some unusual features: Heat transfer is done very fast, the characteristics of compressors or turbines don't really matter, and the values of the mass flow and several state variables are given in advance.

We will show, which problems appeared during the implementation, and present ways how to deal with them. This will help to produce similar models for own teaching purposes. As a starting point, all models described here can be downloaded freely from [12].

## 2 Cyclic Processes in ThermoSimT

Since [11] is a textbook on modeling and simulation, the main purpose of the ThermoSimT library is to teach the design and construction of a Modelica library. But thermo-fluid modeling is a very difficult task, therefore a lot of simplifying assumptions had to be made: The mass flow is constant and the flow has always the same direction, i. e. all connections are uniquely defined as input or output ports. The components have no states describing an internal change, but the thermodynamic variables just jump from the input to the output state. As a consequence, the described models are static, time changes can only be implemented by changing work or heat flows.

Since stream connectors [13] are much too advanced for an introductory textbook, the connector is based on the preliminary version of the thermo-fluid library described in [3]. ThermoSimT contains components for simple devices such as a cylinder, a heater, a pump and a turbine, together with source and sink components and a state measurement device that outputs all relevant thermodynamical variables. Pump and turbine are identically modeled as simple turbo machines based on an isentropic state change with a simple linear characteris-

tic

$$\dot{m} = K \omega.$$

The simple Media library covers the ideal gas with constant heat capacity ("simple air"), the NASA dry air model [14] and a simple model for steam and water, using ideal gas and ideal fluid equations together with a Clausius-Clapeyron based vapor pressure curve.

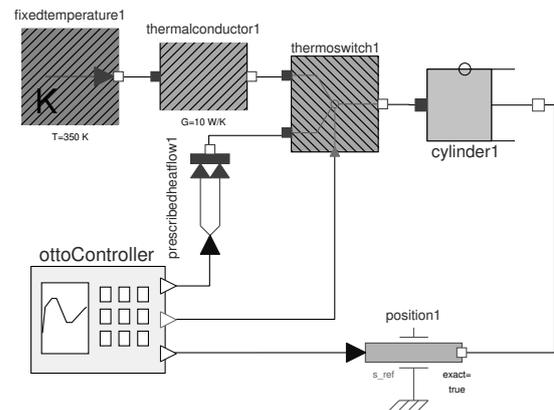


Figure 1: Otto cycle using ThermoSimT.

With these components a model of an Otto cycle can be built easily (cf. Figure 1). The thermodynamic computations are done in the cylinder component, while additional blocks provide a test stand defining the position of the piston and the amount of external heat. Models using simple air and dry air are provided, as well as a similar example for the Diesel process.

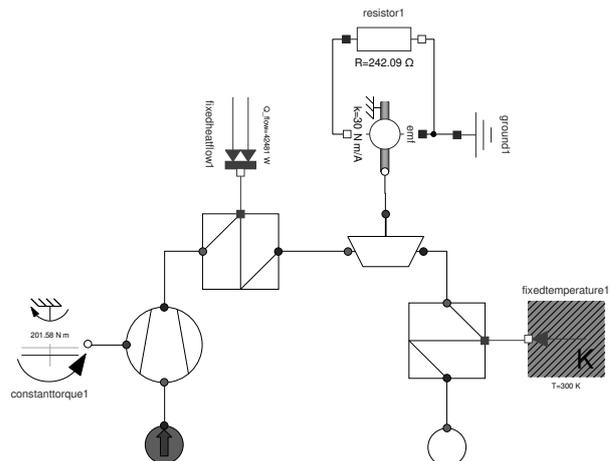


Figure 2: Joule-Brayton cycle using ThermoSimT.

Much more interesting from a modeling perspective is the model of the Joule-Brayton cycle (cf. Figure 2),

which describes the actual flow of the medium between components. The mechanical work for the compressor is provided by a constant torque block, while the external load at the turbine is modeled using a simple generator and a resistor. The ThermoSimT library does not work with a cyclic topology, but an additional cooler at least brings the state of the medium back to its initial state. Versions for simple air and dry air are provided, as well as an identically looking model using the simple steam/water medium, which actually makes it a Clausius-Rankine cycle.

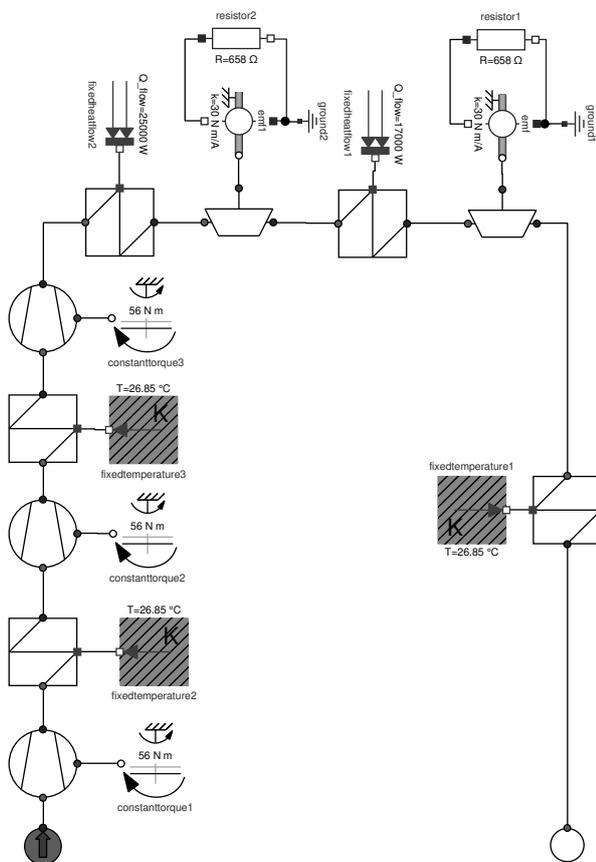


Figure 3: Ericsson cycle using ThermoSimT.

The final example is the Ericsson cycle, which contains several compressors and coolers as well as turbines and heaters, to approximate an isothermal behaviour in the turbo machines. It can be modelled easily with ThermoSimT (cf. Figure 3).

### 3 Otto and Diesel Processes

Since a cylinder model, which is at the heart of the Otto cycle model, is not included yet in the ThermoDLR library, one has to build it oneself. Fortunately, a complete volume model already exists, together with several variants. They all inherit from the parent class `PartialVolume`, which provides most of the variables and equations needed, together with optional `HeatPort`, `Inlet` and `Outlet`. To construct a `CylinderVolume`, one simply extends `PartialVolume`, adds a mechanical `Flange` and provides simple equations for the definition of the volume, the force and the work at the flange. To simplify the drawing of a T-s diagram, an explicit variable for the entropy is added.

Exchanging the cylinder model in the ThermoSimT Otto cycle is all that remains to do. The new model works immediately with `DryAir`, for `SimpleAir` one has to extend its range of validity by defining

```
SimpleAir(T_min=200, T_max=2000)
```

The Diesel controller component needs the internal pressure to create an isobaric process, which can be supplied by a sensor at the optional `Outlet` of the `CylinderVolume`.

### 4 Joule-Brayton Process with Ideal Gas

The construction of a model for the Joule-Brayton cycle seems to be almost trivial. One starts by replacing the thermodynamic components from the ThermoSimT example with their counterparts from ThermoDLR: `Source`, `Sink`, `Compressor`, `Turbine`, and `ConductionElement` as replacement of the `Heater`. All parameters have their default values except for the initial pressure and temperature at the `Source`. That the mass flow can not be defined anywhere, is due to the different concept of ThermoDLR:  $\dot{m}$  is a dynamic variable and is computed in the context of the whole model – here probably mainly depending on the compressor parameters. With the ideas from ThermoSimT in mind, this seems to be strange, and indeed: Starting the simulation results in the infamous error *“Failed to solve nonlinear system using Newton solver during initialization.”*

To find the reason of this problem, one examines a simple test model consisting of a `Source`, a

Compressor and a Sink. The Source defines starting pressure  $p_1 = 1$  bar and temperature  $T_1 = 300$  K, the Sink the final pressure  $p_2 = 6$  bar. Using the same torque as in the ThermoSimT model, the simulation doesn't run, which is not surprising, since the compressor from ThermoDLR uses a completely different characteristic curve and default operating point.

To find an adequate torque value, one has to analyze the equations used in the Compressor component. Using default parameters and neglecting regularisations for small variable values, they are:

$$\begin{aligned} \frac{p_2}{p_1} &= \frac{\omega^2}{\omega_{ref}^2} - \frac{\dot{m}^2}{\dot{m}_{ref}^2} + 1 \\ w_t &= \frac{\kappa}{\kappa-1} RT_1 \left( \left( \frac{p_2}{p_1} \right)^{\frac{\kappa-1}{\kappa}} - 1 \right) \\ &= \frac{\omega \tau}{\dot{m}} \end{aligned}$$

They define a quadratic pressure characteristic and compute the work using the explicit formula of an isentropic process for an ideal gas with constant heat capacity. Inserting the given state variables, the desired mass flow  $\dot{m} = 0.1$  kg/s and the default parameter values, one easily computes values for  $\omega$  and  $\tau$ .

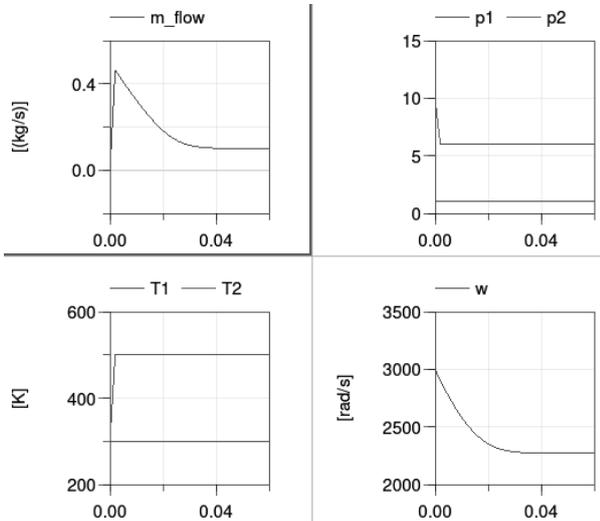


Figure 4: Results of the compressor test model.

Using this torque value for the constant torque and setting the initial value of the rotational velocity to  $\omega$  (at least approximately), the test model runs and has the correct results (cf. Figure 4). In accordance to the philosophy of the ThermoDLR library, the initial value

of  $\dot{m}$  is chosen to be 0, so that the equilibrium is reached by simulating the powering up of the system.

A similar computation can be done for the turbine. Unfortunately, the identical characteristic curve of compressor and turbine contains a very strict regularisation, whenever the pressure drops (as in the turbine). But of course it's simple to create a copy of the curve model and use the quadratic characteristic for the turbine as well. Trying to compute the torque for the given state variable values, one gets a negative value under a square root. This problem can easily be fixed by changing the operation point, setting  $\dot{m}_{ref} = 0.05$  kg/s. Finally one utilizes the linear characteristic of the simple generator to compute the resistance  $R$ , and gets the requested results.

For the heater, no new computations are necessary, since the needed heat is fixed by the thermodynamics. Combining the components (for a start without the final cooler) using the new parameter values, one gets a working Joule-Brayton process, which almost reproduces the required values. Only the pressure  $p_2$  after the compressor is slightly higher in the combined model, maybe due to the effect of several regularisations.

Instead of going through the complete equations, the fine-tuning can easily be done with a few manual parameter changes: First one lowers the torque at the compressor, until  $p_2$  reaches the desired value. Here, the MultiSensor components are very convenient, which display state variables directly in the graphical model. Now all pressure and temperature values are correct, only the mass flow is a bit too small. Better than fiddling with several parameters at once, one can use a simple scaling procedure: Increasing all works and heats by a constant factor  $q$  one can change only the mass flow, not the thermodynamical state. Setting  $q = \dot{m}_{desired} / \dot{m}_{actual}$ , increasing  $\dot{m}_{ref}$ ,  $\tau$  and  $\dot{Q}$  by a factor  $q$  and dividing  $R$  by  $q$ , one eventually reproduces all ThermoSimT values.

Finally one adds another ConductionElement with a large heat transfer coefficient to bring the temperature down to the start value. With the ThermoDLR library, one can now make it a real cycle by deleting the Sink and Source components and closing the circle with an intermediary Volume element. The initial state, which had been defined by the Source, is now given as initial value of the Volume.

## 5 Ericsson Process with Ideal Gas

To simplify the construction of an Ericsson model, a subsystem is built for a line of three compressors and intermediate coolers (cf. Figure 5), as well as a similar model of two turbines with intermediate heater. Both components contain sensors to output the complete work and heat supplied to the line. This will be useful to compute the overall efficiency in the complete model.

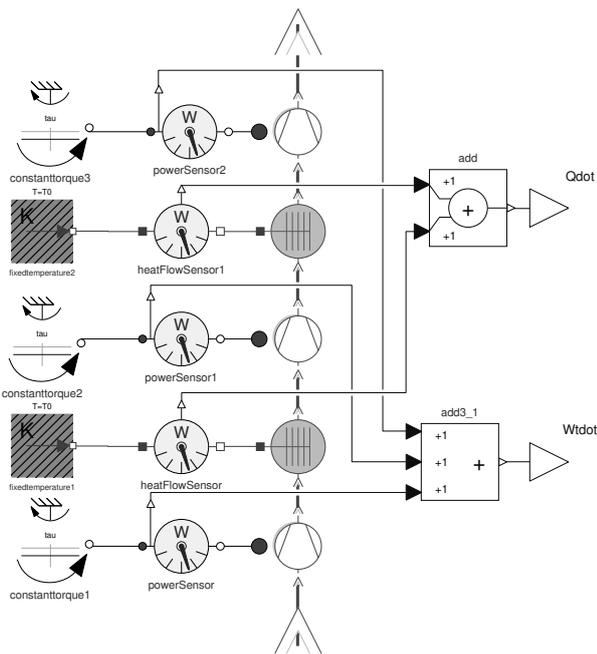


Figure 5: Line of compressors for the Ericsson cycle.

Providing correct parameter values, these compressor and turbine components show approximately isothermal behaviour. Using them in the Joule-Brayton model and guessing reasonable values for the work-related parameters  $\tau$  and  $R$ , the Ericsson model runs – at least, if one lowers the intermediate heating temperature a bit. As before, one can use fine-tuning to get the given pressure values and scaling to reach the correct mass flow. The resulting model is more stable, so that one can raise the intermediate heating temperature to the incoming value to better approximate an isothermal process.

Now we can make good use of the additional possibilities that are supplied by ThermoDLR, and include a heat exchanger that utilizes excess heat after the tur-

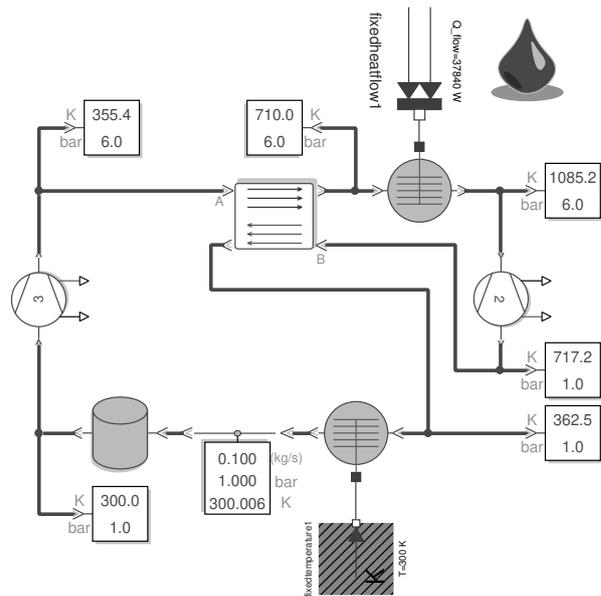


Figure 6: Ericsson cycle with heat exchanger using ThermoDLR.

bine for preheating the gas before entering the heater (cf. Figure 6). To reach the same maximal temperature as before, one starts with a low heat transfer coefficient  $k_{NTU}$ , then gradually lowers the supplied heat, while rising  $k_{NTU}$ . The final Ericsson model is much more realistic than the previous version based on ThermoSimT.

## 6 Joule-Brayton Process with Dry Air

Building a Joule-Brayton example based on dry air as a medium should be done easily by simply changing SimpleAir to DryAirNasa, but unfortunately, the new model doesn't run: To compute the temperature from the given enthalpy, the function  $h(T)$  has to be inverted – but the Dekker-Brent based solver is supplied with a start interval without a zero.

Looking closer at the DryAirNasa implementation, one finds another problem: The computation of the isentropic enthalpy is based on the approximation of constant heat capacity. This may be good enough for many practical applications, but in a teaching context one should present a correct solution. Since the utilised function `isentropicEnthalpy` is not declared as `replaceable`, one has to create a copy of the `DryAirNasa` class and its base class `SingleGasNasa` and supply a computation that

is based on the constant entropy. Additionally, the function `dp_tau_const_isentrop` that is used in ThermoDLR to define the compressor characteristics, is based on a constant heat capacity formula as well and has to be replaced by an exact version `dp_tau_const_isentrop_S`.

Using these more accurate methods, one is still faced with the original problem: The inversion gets a wrong start interval. This is due to the initial value  $\dot{m}(0) = 0$ : The direction of the flow is undefined initially and the algorithm tries values that are going into the wrong direction. Simply setting  $\dot{m}(0) = 0.1$  kg/s, which is given anyhow, succeeds and the model runs – but the values of pressure and mass flow are not the required ones. Obviously, the seemingly small difference between `SimpleAir` and `DryAir` leads to larger deviations than expected.

To find correct parameter values, one again starts with a simple test model for the compressor. Either by a direct computation – which can be easily done for dry air with a small Matlab script – or by a few trial and error steps, one finds values that lead to the required pressure and mass flow. Due to the internal inversions, the model is less stable than its `SimpleAir` counterpart: Even if one sets the correct torque value, one still has to supply an initial value for  $\omega$  that is large enough. For simplicity one can instead directly define  $\omega$  at the flange. The corresponding computation for the turbine again shows that  $\dot{m}_{ref}$  has to be lowered. After that, the computation leads to the correct value of the resistance  $R$ . Closing the cycle with a cooler and a volume element, one arrives at a Joule-Brayton model with dry air that – after a bit of the usual fine-tuning – reproduces the given state and mass flow values.

## 7 Clausius-Rankine Process with Standard Water

Basically, the Clausius-Rankine process is a Joule-Brayton process with water and steam as a medium: Though its technical realisation is much more complicated due to the phase transitions from water to steam and back, conceptually it consists of a pump, a heater, a turbine and a cooler. The media library includes `StandardWater`, a precise description of water and steam based on the IAPWS-IF97 formulation [15]. Yet, one cannot just use one of the Joule-Brayton models and change the medium, since the compressor and turbine components of ThermoDLR use explicit ideal gas

relations. For incompressible fluids the `Pump` component can be used, a suitable turbine component is not provided in the library. Also, completely different values of the state variables than before will be employed: The pressure ranges from 0.1 bar to 60 bar, the highest temperature should be 500 °C and the mass flow 10 kg/s.

To find working parameters, one can start with a simple test model for the pump. The ThermoDLR `Pump` component provides two different characteristics: a centrifugal pump and a simpler nominal pump, which is used in the following. Setting reasonable nominal values and basic parameters, one can once again use the model equations and the given state and mass flow to compute a correct value of the rotational speed  $\omega$ . Adding a standard heater (i. e. a `conductionElement`), one can reach the given temperature by first estimating, then fine-tuning the needed heat flow. The heating process includes the complete vaporisation of the water, succeeded by an overheating of the steam, but all this is automatically taken care of by the state equations used in IF97. Adding corresponding sensors, one can easily monitor the dryness fraction  $x$  everywhere in the cycle model.

A generic turbine model `TurbineG` that works for `StandardWater` (and any any other fluid medium) can be easily constructed: It inherits from the provided `PartialTurboComponent` and uses the explicit characteristic function `dp_tau_const_isentrop_S` that has already been defined for the `DryAir` example. Since the medium changes its phase from hot steam to wet steam inside the turbine, one can expect numerical difficulties, and in fact: Using default parameter values, the usual test model doesn't run, because the pressure reaches values below the triple point of water. To find working values, one initially starts with a simpler process, going from 60 bar to 30 bar instead of trying to reach 0.1 bar immediately. Changing  $\omega_{ref}$  und  $\dot{m}_{ref}$ , one soon gets a model that at least runs for a very short time. This makes it possible to study its behaviour and find the reason of the problems. Adapting  $\omega$ , the model finally runs to the end. Now one can use the usual fine-tuning to gradually lower the pressure to the requested value and to switch from the constant  $\omega$  input to the simple consumer model. The needed resistance value is completely unrealistic, but can finally be scaled to a meaningful value by changing the transformation coefficient of the simple generator model.

Now a first version of a Clausius-Rankine cycle can be built by combining the pump-heater and turbine models. The model runs only for a short time, before the final pressure again drops below the triple point. Enlarging the value of  $R$  by a factor of 100, the model runs to the end, though the state values are far from the requested ones. Once again one goes through the fine-tuning process, changing  $\omega$ ,  $\dot{Q}$  and  $R$ , to reach the requested state values. This is much more tedious than in the previous examples, because the very different properties of water and steam lead to a small range of working parameters. Additionally, their strong coupling sometimes results in a counterintuitive behaviour, as in the following scenario: One finds a too low temperature behind the heater and increases the heat flow, which leads to an increase of the mass flow and a falling temperature. Another problem is the instability of the standard DASSL solver that is used in the simulation program Dymola: Occasionally, the solver hangs for certain parameter values and has to be stopped. Changing to the ESDIRK23a solver drastically reduced the frequency of such events. With a certain amount of perseverance, results within 1 % of the requested values could be reached.

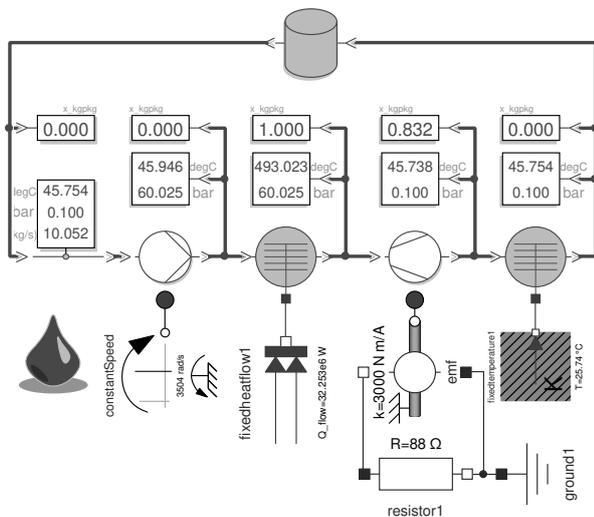


Figure 7: Clausius-Rankine cycle using ThermoDLR.

The next step is the addition of the cooler, which is supplied with a fixed temperature. Choosing the obvious value of  $45.80^{\circ}\text{C}$  – the condensation temperature at 0.1 bar –, one doesn't reach  $x = 0$ , therefore one has to use a smaller value and to increase the heat transfer coefficient  $U$  considerably. Finally adding the volume

element to close the circle, one again gets very different results and has to tune the cooler parameters once more, as well as the initial temperature of the volume. But in the end one reaches a complete Clausius-Rankine cycle with the requested state values (cf. Figure 7).

## 8 Conclusions

All examples of the simple ThermoSimT library could be implemented in ThermoDLR, as well as more complex processes using a heat exchanger or an accurate water model. ThermoDLR is designed to create models that run immediately, nevertheless it was nontrivial to build models that reproduce the given state values and mass flows. A systematic approach often worked that was based on the following steps:

- Start with simple models, where initial and final states are defined by `Source` and `Sink` components.
- Change the operation points of turbo machines to values near the given state values.
- Use (simplified) versions of the component equations to get good starting points for external parameters.
- Gradually fine-tune parameters to approximate the given thermodynamic states. If necessary, change the mass flow by a scaling procedure.
- Combine partial models and reiterate the fine-tuning process.
- Finally close the cycle with a `Volume` component and properly define its initial values.

Sometimes it was difficult to find proper parameters to make the model run at all or to change the state into the required direction – especially for a complex medium such as water and steam. In such cases, it proved useful to set the initial value of the mass flow directly to the required value or to define operation points very close to the target values. Generally, such a model runs at least for a very short time. This is helpful, because seeing the results of small parameter changes directly can provide clues on what to do to finally reach a running model. Other helpful measures were the introduction of control valves to decouple parts and define intermediate states, or even to change the solver. When the model finally works as required, it is generally more

stable than the previous versions and allows for easy parameter changes.

The design of the ThermoDLR library, and especially its ability to start with zero mass flow and model a power-up situation, makes the modeling of thermofluid systems much simpler for non-specialists than the complex Modelica Thermo-Fluid library. On the other hand, the consistent use of the mass flow as a state variable sometimes leads to unexpected behaviour, when parameters are changed or submodels combined. ThermoDLR provides useful basic components that are easy to understand down to the equation level and are easy to extend due to a simple, but convenient inheritance hierarchy. The supplied `MultiSensor` components are very helpful during the fine-tuning phase. Some important elements are missing yet, such as a generic turbine with a corresponding characteristic function or a cylinder volume, and had to be added here.

On the whole, ThermoDLR proved to be a very useful tool for the construction of cyclic process models that can be used for demonstration purposes in thermodynamics lectures. Of course, it requires some effort to familiarise oneself with the library, but going in simple steps, as has been shown before, the learning curve is not very steep and can be mastered by lecturers, who are not experts in thermo-fluid modeling.

## Acknowledgement

The author is grateful to Dirk Zimmer, Niels Weber and Michael Meißner for introducing him to the DLR ThermoFluid Stream Library and for providing helpful hints during the construction of the described models.

## References

- [1] Junglas P. Simulation Programs for Teaching Thermodynamics. *Global J of Engng Educ.* 2006; 10(2):175–180.
- [2] Modelica Association. *Modelica® – A Unified Object-Oriented Language for Systems Modeling, Language Specification Version 3.5.*  
URL <https://modelica.org/documents/MLS.pdf>
- [3] Casella F, Otter M, Proelss K, Richter C, Tummescheit H. The Modelica Fluid and Media library for modeling of incompressible and compressible thermo-fluid pipe networks. In: *Proc. 5th Int. Modelica Conference.* 2006; pp. 631–640.
- [4] Moran MJ, Shapiro HN, Boettner DD, Bailey MB. *Fundamentals of Engineering Thermodynamics.* Hoboken, NJ: John Wiley & Sons, 9th ed. 2020.
- [5] Cerbe G, Wilhelms G. *Technische Thermodynamik.* München: Carl Hanser, 19th ed. 2021.
- [6] Fritzson P, Pop A, et al. The OpenModelica Integrated Environment for Modeling, Simulation, and Model-Based Development. *Modeling, Identification and Control.* 2020;41(4):241–295.
- [7] Franke R, Casella F, Sielemann M, Proelss K, Otter M. Standardization of thermo-fluid modeling in Modelica.Fluid. In: *Proc. 7th Int. Modelica Conference.* 2009; pp. 122–131.
- [8] Drente P, Junglas P. Simulating a simple pneumatics network using the Modelica Fluid library. *SNE Simulation Notes Europe.* 2015;25(2):85–92.
- [9] Zimmer D, Weber N, Meißner M. The DLR ThermoFluidStream Library. In: *Proc. 14th Int. Modelica Conference.* Linköping, 2021; pp. 225–234.
- [10] Zimmer D. Robust object-oriented formulation of directed thermofluid stream networks. *Mathematical and Computer Modelling of Dynamical Systems.* 2020; 26(3):204–233.
- [11] Junglas P. *Praxis der Simulationstechnik.* Haan-Gruiten: Verlag Europa-Lehrmittel. 2014.
- [12] Junglas P. *Thermodynamic Cyclic Processes library in Modelica.*  
URL <http://www.peter-junglas.de/fh/simulation/thermocycle.html>
- [13] Franke R, Casella F, Otter M, Sielemann M, Elmqvist H, Mattson SE, Olsson H. Stream connectors – an extension of Modelica for device-oriented modeling of convective transport phenomena. In: *Proc. 7th Int. Modelica Conference.* 2009; pp. 108–121.
- [14] McBride BJ, Zehe MJ, Gordon S. NASA Glenn Coefficients for Calculating Thermodynamic Properties of Individual Species. *Nasa report tp-2002-211556,* NASA. 2002.
- [15] Wagner W, Cooper JR, et al. The IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam. *J Eng Gas Turbines and Power.* 2000;122(1):150–182.

# Durchführung eines Simulationsprojektes im Simulationspraktikum

Heiderose Stein<sup>1\*</sup>, Tobias Uhlig<sup>1</sup>, Oliver Rose<sup>1</sup>

<sup>1</sup>Institut für Technische Informatik, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Deutschland; \*[tobias.uhlig@unibw.de](mailto:tobias.uhlig@unibw.de)

**Abstract.** Im Zuge der Digitalisierung der Bundeswehr werden neue Systeme und Strukturen eingeführt. Simulationen sind ein Mittel, um diesen Herausforderungen zu begegnen. Die Kompetenz zur Modellbildung und Simulation wird in diesem Rahmen immer relevanter. Dabei kommt der Universität der Bundeswehr eine zentrale Rolle in der Ausbildung von Fachpersonal zu: Zur Vermittlung dieser Kompetenzen wird im Masterstudium Informatik und Ingenieurwesen, zusätzlich zur klassischen Wissensvermittlung mit Vorlesung und Übung, verpflichtend ein Simulationspraktikum durchgeführt. In diesem Rahmen bearbeiten die Studierenden ein Simulationsprojekt. Dieser Artikel beschreibt den strukturierten Aufbau und Inhalt des Praktikums. Er gibt einen Überblick über bestehende Simulationsprojekte und ihren Einsatz in der Lehre. Die Nutzung eines realitätsnahen, simulierten Projekts stellt dabei einen guten Kompromiss aus praktischer Wissensvermittlung und effizienter Durchführbarkeit dar. Das Simulationspraktikum hat in unserer Erfahrung den Lehrbetrieb bereichert und die Modellbildungs- und Simulationskompetenz der Studierenden gefördert.

## Einleitung

Mit der fortschreitenden Digitalisierung verändern sich bestehende Strukturen und Abläufe und existierendes Wissen muss erweitert werden. Diesen Bedarf sehen wir aktuell bei der Digitalisierungsoffensive der Bundeswehr. Traditionell hat die Bundeswehr auf Trainingssimulationen gesetzt, um ihr Personal zu schulen und bestimmte Fähigkeit zu vermitteln. Jetzt rückt auch die analytische Simulation in den Fokus, um Systeme zu verstehen und zu bewerten.

Simulationen machen abstraktes Wissen greifbar und experimentierbar - sie werden deshalb in verschiedenen Bereichen angewandt [1]. Ein erfolgreicher Einsatz von Simulationen hängt auch vom Vorwissen der Beteiligten ab. Deshalb liegt ein Schwerpunkt unserer Arbeit darauf, den Studierenden an der Universität der Bundeswehr die Kompetenzen der Modellbildung und analytischen Simulation zu vermitteln. Sie sollen ein Problem in Fragestellungen konkretisieren, und ein

vereinfachtes Modell ableiten können, mit dem übertragbare Antwortmöglichkeiten erarbeitet werden. Für die Studiengänge Informatik und Mathematical Engineering sind deshalb Modellierungs- und Simulationsgrundlagen Teil des Studiums.

In der Literatur werden vielfältige Erfahrungen, Beispiele und Vorschläge für die Vermittlung von Modellbildung und Simulation dargestellt (z.B. [2]). Eine besondere Rolle nehmen dabei Simulationspraktika ein, in denen das erworbene Wissen beispielhaft an Simulationsprojekten angewendet, gefestigt und vertieft wird ([3], [4], [5]). Wir wollen dazu unseren Praktikumsaufbau und unsere Erfahrungen mit einem simulierten Projektbeispiel in der Lehre ergänzen und einen strukturierten Aufbau für ein solches Praktikum vorstellen. Zudem wollen wir einen ersten Überblick geben, ob das Praktikum die Wissensvermittlung zur Simulation leisten kann, sowie Optionen für die Zukunft aufzeigen.

Im Folgenden Abschnitt gehen wir zunächst auf die bestehende Literatur zum Einsatz von Simulationspraktika in der Lehre ein. Dabei liegt der Fokus auf der Vermittlung von Simulationskompetenzen. Dann stellen wir die entwickelten Strukturen und Inhalte des Praktikums vor (Abschnitt 2). Anschließend diskutieren wir, inwiefern das Praktikum die Wissensvermittlung unterstützt (Abschnitt 3), und schließen mit einem Ausblick auf die zukünftige Perspektiven.

## 1 Methodik für Simulationsprojekte

Trainingssimulationen nehmen eine besondere Rolle bei der Ausbildung von Problemlösungskompetenzen ein. Denn Simulationen unterstützen die Entwicklung komplexer Fähigkeiten, wie sie für die Modellbildung und Simulation benötigt werden: Simulationen werden als besonders effektiv für das Verbessern komplexer Fähigkeiten, wie der generellen Problemlösefähigkeit, ge-

sehen: das Abbilden des Systems in einem Modell, das Experimentieren am Modell und die direkte Rückmeldung unterstützen das Lernen (vgl. [4], [6]). Vlachopoulos et al. [3] beschreiben positive Effekte auf Wissensanwendung, Problemlösungskompetenz und soziale Fähigkeiten. Insbesondere für das Erlernen von Simulationskompetenz, und für Studierende mit unterschiedlichem Vorwissen, beschreiben Collins et al. [7] die Effektivität von Simulationen an Beispielfällen, da sie die Fälle greifbar machen und als Entscheidungshilfe für die Studierenden wertvoll werden. Theoretische Grundlagen für die Effektivität von Simulationen, sowie einen generellen Überblick über simulationsbasierte Ansätze zur Vertiefung von Lehrwissen geben z.B. [3], [4], [8].

Im Rahmen unserer Arbeit liegt der Fokus darauf, wie man diese Mechanismen nutzen und analytische Simulation mittels eines Praktikums vermitteln kann. Dabei zeigen sich Simulationspraktika generell als hilfreiches Instrument für Simulationsdozenten (vgl. [2], [9], [10]). De Mesquita et al. [11] fassen aus Expertenbefragungen zusammen, dass Simulationsprojekte von der großen Mehrheit als wichtiges Instrument in der Lehre geschätzt werden (siehe Abbildung 1). Als Einschränkungen werden der hohe administrative Aufwand (vgl. [3], [12], [13]) und das Definieren des für die Studierenden passenden Inhalts (vgl. [14], [15], [16]) angegeben.

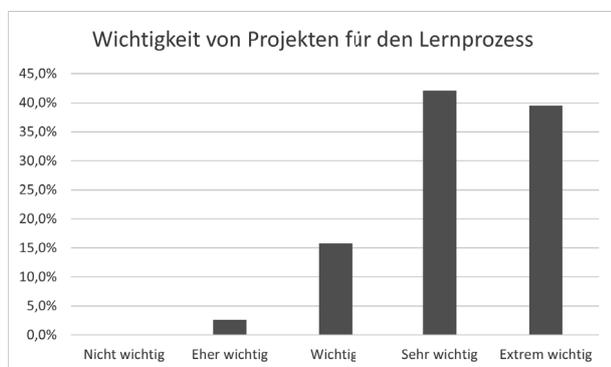


Abbildung 1: Bewertung von Simulationsprojekten nach [11]

Umfragen unter Studierenden bestätigen, dass Simulationen als besonders effektiv zum Entwickeln von Problemlösefähigkeit [5] und das Erlernen von (komplexen) Inhalten [17] eingeschätzt werden, und Simulationsprojekte generell positiv bewertet werden [18]. Spezifisch für die Vermittlung von Simulationskompetenzen fasst etwa Martin [13] zusammen, dass die Erfahrungen im Praktikum von den Studierenden positiv

gesehen werden – besonders dafür, Simulationen aus Kundendaten zu erstellen und Simulationsergebnisse zu vermitteln. Einschränkungen der positiven Bewertung ergeben sich z.B. wenn die Ziele unerreichbar scheinen [3] oder Studenten keine Verbindung zwischen Theorie und Praxis erkennen [19].

## 1.1 Aufbau und Bearbeitungsweise

Der Aufbau eines Simulationskurses ist generell unterteilt in eine theoretische Einführung, gefolgt oder unterstützt von Beispielen und einem abschließenden Simulationspraktikum. Eine Struktur für den Aufbau eines Simulationskurses beschreiben etwa [6] und [20]. Der Aufbau des Praktikums variiert dagegen nach Fallbeispiel (s.u.), Ziel und Voraussetzungen: Je nach Zielsetzung kann der Schwerpunkt der Bearbeitung z.B. auf der Datenauswertung (vgl. [21], [22]), der Modellierung (vgl. [23], [12]), der Implementierung oder Auswertung liegen. Je nach Voraussetzung sind z.B. zeitliche Begrenzungen gegeben (vgl. [2], [13]), unterschiedliches Vorwissen der Studierenden kann zusätzliche Lehrangebote nötig machen (z.B. [14], [15], [16]), oder durch die verwendeten Tools entstehen besondere Voraussetzungen. Bei den Tools ist grundsätzlich wichtig, dass sie den Studierenden bei geringem Einarbeitungsaufwand messbare Ergebnisse liefern, damit deren Nutzen gesehen wird. Bekannte Tools zu verwenden hilft, Einarbeitungsaufwände zu vermeiden ([20], [13]). Daher eignen sich besonders zeitnah eingeführte Tools [24] oder das Angebot von (freiwilligen) Einführungsworkshops [14]. Auch zur Bearbeitungsweise, ob allein oder im Team, gibt es unterschiedliche Einschätzungen (siehe [3], [11]): auf der einen Seite ist das Lernen von anderen Studenten hilfreich (vgl. z.B. [16], [25], [26]), auf der anderen soll jeder Student seine eigenen Fähigkeiten entwickeln.

## 1.2 Fallbeispiel

Die Auswahl des Fallbeispiels spielt eine wichtige Rolle für das Lernen, da die Studierenden darüber einen Bezug zum Thema herstellen oder über den Nutzen einen Zugriff auf die Methode herstellen (vgl. [7], [26]). Adelsberger et al. [6] empfehlen sogar, Studierenden mit unterschiedlichem Hintergrund thematisch unterschiedliche Fallbeispiele zur Verfügung zu stellen. Anhand der Erfahrungsberichte wird eine Vielzahl an verwendeten Fallbeispielen und Empfehlungen deutlich.

Diese fassen [11] in der folgenden Abbildung 2 zusammen.

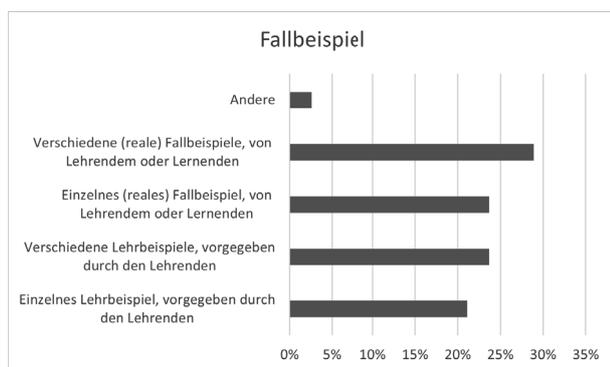


Abbildung 2: Empfehlungen zum Fallbeispiel nach [11]

Unterschiedliche Autoren beschreiben die ganze Spanne von Fallbeispielen, über von den Studenten selbst gestaltete Aufgaben bis hin zu praktischen Projekten mit realen Kunden: Kashefi et al. [24] stellen beispielsweise ein Praktikum mit zwei aufeinander folgenden Lehrbeispielen vor. Korecko et al. [15] beschreiben ein Simulationsprojekt, in dem die Studierenden ihr Thema selbst wählen können. Martin [13] präsentiert ein reales Fallbeispiel in Zusammenarbeit mit einer Firma. Pereira et al. [27] wählen ein Fallbeispiel, das mit einem realen Experiment gekoppelt ist, so dass die Simulationsergebnisse von den Studierenden direkt verglichen und bewertet werden können. Byrnett [28] beschreibt ein Fallbeispiel, bei dem der Lehrer die Rolle eines realitätsnahen Projektkunden einnimmt, es aber kein real gekoppeltes Experiment gibt.

Jedes Vorgehen hat seine Vor- und Nachteile, die die einzelnen Autoren im Detail beschreiben: bei Lehrbeispielen ist genau steuerbar, was die Studierenden lernen (sollen). Bei zwei unterschiedlichen Aufgaben können zusätzlich Erkenntnisse aus der ersten Aufgabe für die Lösung der zweiten verwendet werden; es kommt aber auch zu unerwünschten Übertragungslösungen (vgl. [24]). Bei einem selbst gewählten Thema ist die Motivation der Studierenden für das Thema eher höher, Aufwand und Nutzen sind aber für den Lehrenden schwer einzuschätzen. Bei einem realen Projekt ist der Nutzen direkt sichtbar, dafür ist das Praktikum an Projektrestriktionen und die Verfügbarkeit realer Daten gebunden. Bei einem vorgegebenen Beispiel mit realem Bezug sind Daten und der Umfang steuerbar, doch deren Bereitstellung bedeutet einen hohen Aufwand für den Lehrenden.

## 2 Entwickeltes Simulationspraktikum

Das beschriebene Simulationspraktikum ist Teil eines Kurses bestehend aus Vorlesung, Übung und Simulationspraktikum. In der Vorlesung werden vor allem die Grundlagen und das theoretische Grundwissen vermittelt. Die Übung findet parallel zur Vorlesung statt und vertieft die vorgestellten Konzepte; in der Übung werden dabei praktische Problemstellungen, die mit Simulationsmodellen bearbeitet werden, vorgestellt und teilweise im Selbststudium erarbeitet. Dabei kommen die Simulatoren AnyLogic, FlexSim und SIGMA zum Einsatz. Das Simulationspraktikum ist nach der theoretischen Einführung in das Thema und vor der abschließenden Fachprüfung angesiedelt, und geht über mehrere Wochen.

### 2.1 Ziele des Simulationspraktikums

Durch das Simulationspraktikum sollen die Studierenden ihr theoretisches Wissen anwenden, um selbstständig eine analytische Simulation eines Systems aus Kundendaten zu erstellen und damit die Kundenfragen zu beantworten. Das Simulationspraktikum hat damit die folgenden unmittelbaren Ziele:

- Den Vorlesungsstoff veranschaulichen und festigen.
- Verständnis für die Anwendbarkeit, Vorteile und Grenzen von Simulationsstudien vermitteln.
- Praktische Modellierungs- und Simulationserfahrung in einem Projekt geben.

Als Fernziel sollen die Studierenden auf eine Anwendung von Simulationen im Arbeitsfeld vorbereitet werden, um die Digitalisierung der Bundeswehr zu unterstützen.

Die Zielgruppe besteht aus Studierenden im Master, mit nach Fachrichtung – Informatik und Mathematical Engineering – unterschiedlichen Vorkenntnissen und Schwerpunkten. Das Praktikum durchlaufen in jedem Jahrgang etwa 20-30 Studierende. Projektmanagement und Teamarbeit sind Teil des Praktikums, daher wird das Projekt im Team bearbeitet. Die Studierenden bilden dazu selbstständig Gruppen von 3-5 Personen. Damit sollen zum einen die Studierenden auf spätere Projektarbeiten vorbereitet werden. Zum anderen erscheint der Austausch zwischen den Studierenden wichtig für die Wissensvermittlung (vgl. z.B. [16],

[26]), z.B. um den Code anderer Studierender lesen zu lernen ([25]).

## 2.2 Simulation eines Fallbeispiels

Als Fallbeispiel verwenden wir die Simulation eines realitätsnahen Beispiels; Fragestellungen für die Studierenden liegen neben einer Modellierung und Analyse des Ist-Zustands z.B. in einer Untersuchung möglicher Szenarien oder einer Optimierung. Durch die Simulation eines Fallbeispiels werden unter anderem die Vorteile einer Trainingssimulation auf das Praktikum übertragbar. So ist das Untersuchungsobjekt bekannt und steuerbar, und die Ergebnisse der Studierenden können am Simulationsmodell überprüft werden. Eingangs- und Ausgangsdaten sind gezielt erstellt und anpassbar, so dass geeigneten Datensets für die Modellierung erzeugt werden können (vgl. [21], [22]). Ein realitätsnahes Fallbeispiel kann die Vorteile eines realen Industrieprojekts annehmen, d.h. die Studierenden können direkt einen praktischen Nutzen erkennen (siehe [7], [13]). Dabei unterliegt es nicht den Anforderungen einer Zusammenarbeit mit Industriepartnern (vgl. [13]). Damit sind z.B. stetige Anpassung an die Bedürfnisse und Fortschritte der Studierenden möglich, indem einzelne Projektkomponenten erweitert oder verringert werden, oder die Studierenden eigene Fragestellungen in Ergänzung zur Aufgabenstellung einbringen. Dieses Steuern kann einen positiven Einfluss, sowohl auf die unmittelbare Motivation der Studierenden als auch auf die langfristigen Ergebnisse des Praktikums, entfalten (vgl. [29]).

Als Nachteil bringt die Simulation eines Fallbeispiels vor allem einen erhöhten initialen Aufwand mit sich: der Lehrende übernimmt die Erstellung eines realistischen Simulationsmodells sowie die Aufbereitung aus Kunden- und Mentorensicht – oder teilt diese Rollen noch auf. Durch existierende Fallbeispiele kann dieser Aufwand jedoch für folgende Praktika verringert werden (vgl. [13]). Im Verlauf des Praktikums sind so bislang Fallbeispiele aus verschiedenen Themenbereichen entstanden, die durch unterschiedliche Fragestellungen erweitert oder angepasst werden können. Ein Beispielprojekt im Praktikum ist die simulationsbasierte Analyse von Werksverkehrsstrategien. Der Ist-Zustand für den Verkehr ist, dass alle Servicetechniker mit eigenem Fahrzeug zu entsprechenden Serviceanfragen auf dem Werkgelände fahren. Die Studierenden sollen ein Vergleich dazu eine alternative Strategie, mit selbstfahrenden Fahrzeugen als Shuttleservice für die

Techniker, simulieren. Als historische Daten stehen dabei Serviceanfragen eines vergangenen Jahres zur Verfügung. Daraus ergeben sich für die Studierenden drei Hauptpunkte für das Praktikum: erstens die Analyse der historischen Daten und die Anpassung der entsprechenden Input-Modelle, zweitens die Modellierung und Simulation des Ist-Zustandes, inklusive der Verifikation und Validierung des Modells, und drittens die Erstellung des neuen Transportkonzeptmodells und dessen Vergleich mit dem aktuellen Modell.

## 2.3 Entwickelte Praktikumsstruktur

Die Struktur des entwickelten Praktikums orientiert sich an den einzelnen Schritten und Zielen der Bearbeitung eines Simulationsprojekts (vgl. [20]). Vorgehensmodelle beschreiben Leitlinien für das Erstellen einer Simulationsstudie (z.B. [30], [31]). Sie bilden den Modellbildungs- und Simulationsprozess komplett und in strukturiert nachvollziehbaren Schritten ab und können damit als Grundlage für die einzelnen Schritte der Projektbearbeitung dienen. Wir nutzen im Praktikum das Vorgehensmodell [32] siehe Abbildung 3 mit der Anpassung, dass Konzeptmodell und formales Modell zusammengefasst werden. Die Phasen der Projektbearbeitung und dazugehörigen Lernziele sind in Tabelle 1 dargelegt.

Die entwickelte Struktur sieht also wie folgt aus: In einem Eingangstreffen wird der Studentengruppe das Fallbeispiel vom Kunden vorgestellt, und der Tutor gibt den Studenten mithilfe genereller oder fallspezifischer Tutorials eine Einstiegshilfe in das verwendete Simulationstool. Die einzelnen Schritte und Aufgaben werden in regelmäßigen Treffen mit dem Dozenten diskutiert, und im Selbststudium bearbeitet. Im Praktikum wird den Studierenden dabei weitestgehend freigestellt, mit welchen Tools sie arbeiten, um den Einarbeitungsaufwand gering zu halten. In unserem Fall wird nur das Simulationstool vorgegeben, Tools zur Datenanalyse, Visualisierung und Modellierung dürfen frei gewählt werden. In den regelmäßigen Folgetreffen wird zudem der Projektfortschritt überprüft, dabei können Hilfestellungen, wie z.B. Verweise auf eigenständig zu erarbeitende Webinhalte (vgl. [4]) oder Anregungen zur Modellierung und Simulation gegeben werden. Zudem können Anpassungen vorgenommen werden, so dass die Ziele erreichbar und dennoch fordernd bleiben (vgl. [3]).

Tabelle 1: Einzelschritte und -lernziele für Modellierung, Datenanalyse und Simulation

Aufgabe	Primäres Lernziel	Vermittlungsmethode
Aufgabendefinition	Verstehen der Problemstellung: Selbstständiges formulieren der Ziele aus den Kundenfragen. Beschreiben der Aufgabe und identifizieren von Teilaufgabenstellungen	Unterstütztes Selbststudium: d.h. den Studierenden werden generelle Hilfestellungen und projektspezifische Informationen zur Verfügung gestellt, für Fragen sind wöchentliche Treffen vorgesehen
Systemanalyse und Modellformalisierung	Beschreiben der modellhaften Zusammenhänge und Größen (Konzeptionelles Modell) und Modellformalisierung mit UML. Beschreibung von Struktur, Zustand und Verhalten der Systemkomponenten.	Unterstütztes Selbststudium - ggf. ergänzt mit einem Beispielprojekt
Datenbeschaffung	Beschaffung relevanter und herausfiltern unnötiger Informationen	Selbständiges Einholen der relevanten Daten (Daten werden mit Simulationsmodell generiert).
Datenaufbereitung	Datenanalyse, um die Input-Modelle zu erstellen	Unterstütztes Selbststudium - ggf. ergänzt um weitere Daten und Besprechungsangebote
Implementierung	Erstellen des ausführbaren Simulationsmodells in einer vorgegebenen Simulationssoftware (AnyLogic)	Unterstütztes Selbststudium - Beispielimplementierungen und regelmäßige Besprechung der Implementierung mit dem Tutor
Experimente und Analyse	Experimente durchführen, auswerten, visualisieren und präsentieren	Unterstütztes Selbststudium und Gespräch mit Kunde/ Tutor, Präsentation im Kurs

### 3 Bewertung des Praktikums

Im Folgenden gehen wir auf die Bewertung des Praktikums ein, d.h. inwieweit das entwickelte Praktikum die Modellbildungs- und Simulationskompetenzen der Studierenden bislang unterstützt hat. Generell können qualitative und quantitative Bewertungen darüber vorgenommen werden, inwieweit das Praktikum die Simulationskompetenzen fördert (vgl. [4]). Qualitative Bewertungen umfassen Einschätzungen, z.B. Feedback und Selbstbewertung. Quantitative Bewertungen sind messbare Ergebnisse wie Testergebnisse und Expertenbewertungen. Hier werden das Feedback der Studierenden und Lehrenden, sowie der Einfluss der Praktikumseinführung auf die Prüfungsergebnisse untersucht.

#### 3.1 Feedback

Das Feedback der Studierenden zum Simulationspraktikum wird als Teil der Abschlusspräsentation erhoben, um einen Gesamteindruck zu erhalten (vgl. [3]). Das Studentenfeedback der letzten Jahre zeigt eine positive Bewertung des Praktikums – besonders die ersten Erfahrungen mit einem eigenen Simulationsprojekt und die Möglichkeit, eigene Ideen vorzustellen und mit der Simulation zu untersuchen. Zur Verbesserung werden

unterstützende Unterlagen, wie z.B. eine Dokumentationsvorlage, oder individuelle Wünsche nach bestimmten Themenschwerpunkten, geäußert.

Die Simulations- und auch Projektpraxis der Studierenden verbessert sich auch nach den Einschätzungen der Lehrenden. So wird im Verlauf des Projekts das Vorgehen der Studierenden auf die Schritte eines Simulationsprojekts angepasst. Dabei gehen die Studierenden dazu über, die Fragestellungen mit dem Simulationsmodell abzugleichen, und zu erfassen, welche Daten benötigt werden, welche Experimente durchzuführen sind, welche Annahmen dabei getroffen werden müssen und wodurch sich Grenzen ergeben. Bei der Projektarbeit verbessern sich insbesondere Planung und Teamarbeit über die Dauer des Projekts. Insgesamt decken sich damit Feedback von Studierenden und Lehrenden mit den in Abschnitt 1 geschilderten Erfahrungen.

Der Kontakt zu ehemalige Studierenden lässt vermuten, dass das Simulationspraktikum vor allem eine langfristige Wirkung entfaltet. Zum einen erinnern sich die Studierenden deutlich an ihr Projekt und die dabei gesammelten Erfahrungen. Zum anderen scheinen sie ein grundsätzliches Verständnis für die Möglichkeiten und den Einsatz von Simulationen mitzunehmen. Das zeigt sich z.B. wenn sich Studierende mit einem eigenen Thema für eine Studienarbeit oder mit einer Projekt-

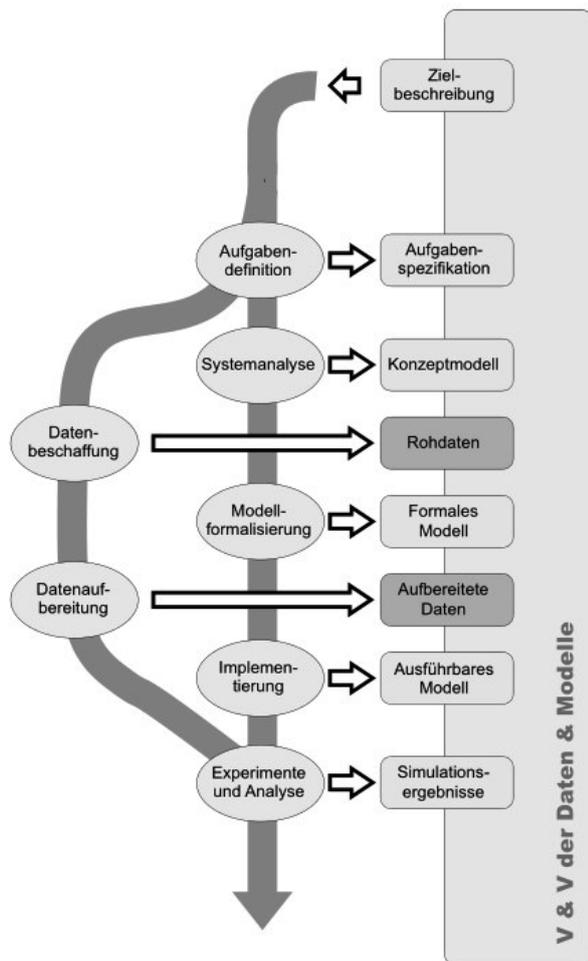


Abbildung 3: Vorgehensmodell

tanfrage aus dem Berufsleben melden. Dies deckt sich mit den Erfahrungen, dass die Kenntnis von Simulationen hilfreich für ihren späteren Nutzen in der Praxis ist ([2], [33]).

### 3.2 Prüfungsergebnisse

Die Ergebnisse der den Kurs abschließenden Prüfung zeigen, dass das Praktikum den Vorlesungsstoff veranschaulichen und festigen kann: die Notenverteilung hat sich zu besseren Noten verschoben, seit das Praktikum verpflichtender Bestandteil des Kurses ist (siehe Abbildung 4). Der Effekt ist dabei nicht so positiv wie das Feedback, was sich mit den Beobachtungen von [16] deckt. Das kann daran liegen, dass manche Studenten die Verbindung zwischen theoretischen Inhalten und Simulationsprojekt nicht herstellen (vgl. [19]),

oder dass zwischen der Vermittlung der theoretischen Inhalte und der Prüfung zu viel Zeit verstrichen ist (vgl. [24]). Zudem ist anzumerken, dass die Stichproben - aufgrund der geringen Anzahl an Studierenden je Jahrgang - klein sind. Der Trend entspricht allerdings dem grundsätzlich positiven Feedback.

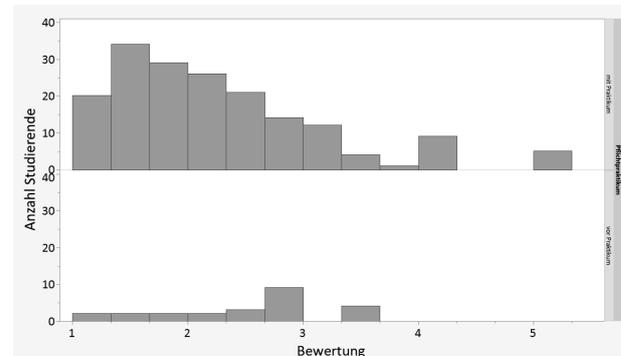


Abbildung 4: Notenverteilung vor (unten) und seit Einführung des Praktikums (oben)

## 4 Diskussion und Ausblick

Zusammenfassend ergibt sich, dass das entwickelte Praktikum ein wertvolles Lehrinstrument für die Ausbildung von Modellbildungs- und Simulationskompetenzen sein kann. Durch die Verwendung eines simulierten Fallbeispiels können praxisnahe Einblicke gewährt werden, ohne den Beschränkungen eines realen Projekts zu unterliegen. Dabei werden die grundlegenden Simulationsmechanismen zur Vermittlung genutzt: ein freies Experimentieren und Lernen am vereinfachten Modell. Durch die bereits erstellten Fallbeispiele kann der erhöhte Initialaufwand ausgeglichen werden. Die entwickelte Struktur erlaubt das gezielte Anpassen an die Lernenden sowie die verwendeten Tools, wie z.B. spezielle Simulatoren.

Für die Zukunft ergeben sich Möglichkeiten, die Bewertung des Praktikums und das Praktikum selbst anzupassen - gerade im Hinblick auf die Vorbereitung der Digitalisierung der Dienststellen der Bundeswehr. Für die Bewertung haben wir bislang vor allem das Feedback der Studierenden direkt nach dem Praktikum genutzt. Da inzwischen die ersten Studierenden ins Berufsleben eingetreten sind, werden auch diese Langzeiterfahrungen zugänglich. Hier kann das bislang erhobene Feedback systematisch ergänzt werden, etwa über Fragebögen. Dazu können die an der Dienststelle tatsäch-

lich benötigten Kompetenzen erfasst werden, um zu evaluieren, inwiefern sie durch das Praktikum vermittelt werden. Um die Entwicklung der Modellbildungs- und Simulationskompetenzen über einen längeren Verlauf sichtbar zu machen, wären z.B. auch systematische Feedback-Erhebungen während des Praktikums, im Anschluss an das Praktikum und mit Eintritt in die Dienststelle möglich. Das entwickelte Praktikum soll in Zukunft auch auf neue Bedarfe flexibel angepasst werden. Dazu sind die bestehende Struktur und mehrere Beispiele modular entwickelt worden. Diese können nun z.B. für weitere Personenkreise angepasst werden: Für Bachelorstudenten im Rahmen eines Praktikums, für Austauschstudenten im Rahmen einer Projektarbeit oder für einzelne Dienststellenangehörige in einer Kurzeinführung.

### Acknowledgement

Wir bedanken uns bei allen Mitwirkenden am Praktikum: das sind besonders unsere (ehemaligen) Kollegen Mihaela Hanea, Thomas Mayer, Falk Pappert, Alexander Roman, Tabea Werger, Shufang Xie und Zhugen Zhou, sowie unsere Studierenden.

Diese Forschungsarbeit wird durch dtec.bw –Zentrum für Digitalisierungs- und Technologieforschung der Bundeswehr gefördert. dtec.bw wird von der Europäischen Union –NextGenerationEU finanziert.

### Literatur

- [1] Kleint R, Mayer T, Uhlig T. Logistic Simulation to Support Military Rescue Chains. *IITSEC*. 2021;.
- [2] Smith JS, Alexopoulos C, Henderson SG, Schruben L. Teaching undergraduate simulation — 4 questions for 4 experienced instructors. In: *2017 Winter Simulation Conference (WSC)*. 2017; pp. 4264–4275. ISSN: 1558-4305.
- [3] Vlachopoulos, Makri A. The effect of games and simulations on higher education: a systematic literature review | *International Journal of Educational Technology in Higher Education* | Full Text. 2017;.
- [4] Chernikova, Heitzmann N, Stadler. Simulation-Based Learning in Higher Education: A Meta-Analysis. 2020.
- [5] Farashahi M, Mahdi Tajeddin. Effectiveness of teaching methods in business education: A comparison study on the learning outcomes of lectures, case studies and simulations. *The International Journal of Management Education*. 2018;16(1):131–142. MAG ID: 2792719890.
- [6] Adelsberger HH, Bick M, Pawlowski JM. Design principles for teaching simulation with explorative learning environments. 2000;pp. 1684–1691.
- [7] Collins AJ, Thaviphoke Y, Leathrum JF, Sturrock D. An Education of Simulation Discussion. In: *2019 Winter Simulation Conference (WSC)*. National Harbor, MD, USA: IEEE. 2019; pp. 3319–3331.
- [8] Krath J, Schürmann L, von Korflesch HFO. Revealing the theoretical basis of gamification: A systematic review and analysis of theory in research on gamification, serious games and game-based learning. *Computers in Human Behavior*. 2021;125:106963.
- [9] Freimer M, Schruben LW, Theresa M Roeder, Roeder TM, Standridge CR, Harmonosky CM, Ståhl I. You are going to teach simulation: now what? tips and strategies. *Proceedings of the 2004 Winter Simulation Conference, 2004*. 2004;2:2057–2065. MAG ID: 2107870405.
- [10] Standridge CR, Centeno MA, Johansson B, Björn Johansson, Ståhl I. Introducing simulation across the disciplines. 2005;pp. 2274–2279. MAG ID: 2156810694.
- [11] de Mesquita MA, da Silva BC, Tomotani JV. Simulation education: a survey of faculty and practitioners. 2019; pp. 3344–3355. MAG ID: 3007168921.
- [12] van der Zee D, Kotiadis K, Tako AA, Pidd M, Balci O, Tolk A, Elder M. Panel discussion: Education on conceptual modeling for simulation - challenging the art. In: *Proceedings of the 2010 Winter Simulation Conference*. Baltimore, MD, USA: IEEE. 2010; pp. 290–304.
- [13] Martin N. Bringing students to practice: performing a real-life simulation study in an introductory simulation course. In: *2018 Winter Simulation Conference (WSC)*. Gothenburg, Sweden: IEEE. 2018; pp. 4014–4025.
- [14] LaRue RJ, Monaco I, Latulippe DR. Enhancing Practical Learning in Undergraduate Chemical Engineering Courses via Integration of Commercial Process Modelling Software. 2021;pp. 123–131. MAG ID: 3127118273.
- [15] Korecko S. Project-Based Approach to Teaching Event-Driven Simulation. *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2019;pp. 1604–1608. MAG ID: 2956378762.
- [16] Komulainen TM. Integrating commercial process simulators into engineering courses. *IFAC Proceedings Volumes*. 2013;46(17):274–279. Number: 17.
- [17] Jennings D. Strategic Management: An Evaluation of the Use of Three Learning Methods. *Journal of*

- Management Development*. 2002;21(9). MAG ID: 1991557989.
- [18] Jimenez MN. Utilizing Interactive Simulation-Based Modules for Enhancing Learning Outcomes in Basic Computing Skills Course. *2013 Fourth International Conference on e-Learning "Best Practices in Management, Design and Development of e-Courses: Standards of Excellence and Creativity"*. 2013;pp. 67–73. MAG ID: 2009668966.
- [19] Wankat PC. Integrating the Use of Commercial Simulators into Lecture Courses. *Journal of Engineering Education*. 2002;91(1):19–23. MAG ID: 2022742644.
- [20] Garcia H, Centeno MA. S.U.C.C.E.S.S.F.U.L.: A framework for designing discrete event simulation courses. In: *Proceedings of the 2009 Winter Simulation Conference (WSC)*. Austin, TX, USA: IEEE. 2009; pp. 289–298.
- [21] Lynch CJ, Gore R, Collins AJ, Cotter TS, Grigoryan G, Leathrum JF. Increased Need for Data Analytics Education in Support of Verification and Validation. In: *2021 Winter Simulation Conference (WSC)*. Phoenix, AZ, USA: IEEE. 2021; pp. 1–12.
- [22] Leathrum JF, Collins AJ, Cotter TS, Lynch CJ, Gore R. Education in Analytics Needed for the Modeling & Simulation Process. In: *2020 Winter Simulation Conference (WSC)*. Orlando, FL, USA: IEEE. 2020; pp. 3236–3247.
- [23] Robinson S, Arbez G, Birta LG, Tolk A, Wagner G. Conceptual modeling: definition, purpose and benefits. 2015;pp. 2812–2826. MAG ID: 2291537941.
- [24] Kashefi A, Alwzinani F, Bell D. Perspectives on teaching modeling and simulation in a department of computer science. In: *2018 Winter Simulation Conference (WSC)*. Gothenburg, Sweden: IEEE. 2018; pp. 4058–4068.
- [25] Leathrum JF, Sokolowski JA, Shen Y, Audette M. Software Skills Required by M&S Graduates For Des Development. In: *2019 Winter Simulation Conference (WSC)*. National Harbor, MD, USA: IEEE. 2019; pp. 3285–3294.
- [26] Zha S, Estes MD, Michele D Estes, Xu L. A Meta-Analysis on the Effect of Duration, Task, and Training in Peer-Led Learning. 2019;12(1):5–28. MAG ID: 2949521130.
- [27] Pereira DC, Diego Crespo Pereira, del Rio Vilas D, David del Rio Vilas, Monteil NR, Nadia Rego Monteil, Prado RR, Rosa Rios Prado, Rodriguez AL. A project-based teaching experience for simulation and optimization education. *2011 7th International Conference on Next Generation Web Services Practices*. 2011;pp. 436–440. MAG ID: 2065533337.
- [28] Byrnett DL. Implementing student projects in a simulation course. *WSC '87*. 1987;pp. 77–81. MAG ID: 2004922659.
- [29] Yang YTC, Chang CH. Empowering students through digital game authorship: Enhancing concentration, critical thinking, and academic achievement. *Computers in Education*. 2013;68:334–344. MAG ID: 2014627836.
- [30] Furian N, Pardalos PM, O'Sullivan M, Walker CG, Vössner S, Neubacher D. A conceptual modeling framework for discrete event simulation using hierarchical control structures. *Simulation Modelling Practice and Theory*. 2015;56:82–96. MAG ID: 383151541.
- [31] Montevechi JAB, Pereira TF, da Silva CES, Miranda RdC, Scheidegger APG. Identification of the main methods used in simulation projects. In: *2015 Winter Simulation Conference (WSC)*. Huntington Beach, CA, USA: IEEE. 2015; pp. 3469–3480.
- [32] Rabe M, Spieckermann S, Wenzel S. *Verifikation und Validierung für die Simulation in Produktion und Logistik - Vorgehensmodelle und Techniken*. Springer. 2008.
- [33] Centeno M, Carrillo M. Challenges of introducing simulation as a decision making tool. In: *Proceeding of the 2001 Winter Simulation Conference (Cat. No.01CH37304)*. Arlington, VA, USA: IEEE. 2001; pp. 17–21.

# Studieren mit digitalen Medien

Nico Marten<sup>1\*</sup>, Kathrin Thiele<sup>1</sup>

<sup>1</sup>Institut für Konstruktion und angewandten Maschinenbau, Fakultät Maschinenbau, Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Straße 46/48, 38302 Wolfenbüttel; \**ni.marten@ostfalia.de*

**Abstract.** Inwiefern sind unsere Studierenden auf die Vielzahl von digitalen Tools, Informationsbeschaffungssystemen und Kommunikationsmöglichkeiten überhaupt vorbereitet? Alle Systeme und digitalen Zugänge werden ihnen zur Verfügung gestellt, aber reicht dies aus, um damit auch adäquat arbeiten zu können? Studierende der Fakultät Maschinenbau lernen (bis heute) das technische Zeichnen im ersten Semester mit Stift und Papier. Ihnen wird also das grundlegende Know-How schon früh gelehrt. Ist dieser Schritt auch in der digitalen Informationswelt nötig? Oder beherrschen die heutigen Studierenden (digital Natives) bereits alle ihnen zur Verfügung gestellten digitalen Tools und Informationen von Grund auf? Die hier ausgewertete Interviewstudie gibt einen Einblick in die Fähigkeiten (und Kompetenzen) die die Studierenden aktuell anwenden, um digital arbeiten zu können.

## Einleitung

Digitale Medien sind fester Bestandteil der Lehre an unseren Hochschulen. Insbesondere durch die Corona-Semester hat die Nutzung stark zugenommen. Einerseits werden gezielt digitale Medien zum Beispiel über Lernplattformen wie Moodle von den Lehrenden zur Verfügung gestellt. Andererseits arbeiten die Studierenden mit selbst recherchierten Unterlagen und digitalen Werkzeugen.

Digitale Medien 1.0	Digitale Medien 2.0
Internetseiten	Foto/Video Sharing
Internetbibliotheken	Wikis
Suchmaschinen	Instant Messaging
Plattformanwendungen	Social-Network

**Tabelle 1:** Digitale Medien 1.0 und 2.0 nach Grosseck [1]

Zu den digitalen Medien zählen neben den klassischen Dateien mit Informationen (wie z.B. Skripte)

auch interaktive Tools. Grosseck [1] unterteilt die digitalen Medien in zwei Bereiche, wie in Tabelle 1 zu sehen ist, wobei der Bereich digitale Medien 1.0 sich darauf beschränkt, Inhalte zur Verfügung zu stellen. Der Bereich 2.0 bezieht sich zusätzlich auf den Austausch und die digitale Kommunikation. In Lernplattformen wie Moodle lassen sich mehrere digitale Medien gleichzeitig abbilden. So sind Elemente der „Digitalen Medien 2.0“ wie beispielsweise Wikis oder Instant-Messaging eingefügt. Die Grenzen dieser Medien sind somit fließend.

In klassischen Grundlagenvorlesungen kommen eher die digitalen Medien 1.0 zur Anwendung. Dies erfordert an digitalen Kompetenzen die technische Fähigkeit mit den verwendeten Programmen umzugehen.

In der betrachteten Veranstaltung Mathematik 1 in der Fakultät Maschinenbau wird in Teilen das eigenständige Vorbereiten von Inhalten gefordert, welches nur teilweise in der Kontaktzeit besprochen wird. Es kann beobachtet werden, dass die Studierenden nicht nur mit den vorgegebenen Unterlagen arbeiten, sondern als zusätzliche Hilfe auch im Internet recherchieren.

Der Einsatz digitaler Medien setzt Kompetenzen voraus, die bei den Studierenden in unterschiedlicher Ausprägung vorhanden sind. Die technische Fähigkeit, mit den verwendeten Werkzeugen umzugehen, bringen die Studierenden heutzutage in der Regel mit. Bei weiteren digitalen Kompetenzen (bspw. das Recherchieren nach digitalen Inhalten) besteht Unterstützungsbedarf.

In den Bachelor- sowie Masterstudiengängen der Fakultät Maschinenbau an der Ostfalia Hochschule für angewandte Wissenschaften steht die Förderung von digitalen Kompetenzen zur Nutzung von digitalen Medien bisher wenig im Fokus.

Im Rahmen eines Projektes wird ein adäquates Förderkonzept erstellt. Dafür werden in diesem ersten Schritt die wesentlichen benötigten Kompetenzen herausgearbeitet. Zur Analyse der eingesetzten digitalen Kompetenzen im Studienalltag der Studierenden der Fakultät Maschinenbau wurden bisher keine Daten

erhoben. Dies ist speziell vor dem Hintergrund einer stark heterogenen Studierendenschaft mit unterschiedlichsten Vorausbildungen interessant. Ebenso sind die Auswirkungen der vergangenen Corona-Semester bzw. Online-Semester an der Fakultät Maschinenbau noch kaum erforscht. An der Fakultät fanden drei Semester (Sommersemester 2020 bis Sommersemester 2021) überwiegend online statt.

In Form von Studierendeninterviews ist untersucht worden, wie die Studierenden im Grundstudium mit den verschiedenen digitalen Medien umgehen. Der Fokus der Analyse liegt auf dem Nutzungsverhalten der Studierenden.

Aus dieser Analyse wird erarbeitet, welche digitalen Kompetenzen wesentlich sind. Dies ist die Basis, um später gezielt Unterstützungsangebote im Bereich der digitalen Medien und Kompetenzen zu entwickeln. Die Studie und ihre Ergebnisse werden in dieser Ausarbeitung vorgestellt.

Im ersten Teil wird der Aufbau und die Konzeption der Studie erläutert. Die Ergebnisse der Befragung werden dann anhand von sechs Kategorien untersucht und mit den digitalen Kompetenzen in Beziehung gesetzt.

Das Projekt wird vom Land Niedersachsen im Rahmen des Promotionsprogramms LernMINT (<https://lernmint.org>) gefördert.

## 1 Fragestellungen

Aus der Aufgabenstellung begründen sich verschiedene Fragestellungen, die in der Analyse näher untersucht werden sollen:

(F1): Welche digitalen Plattformen und Informationsbeschaffungsmöglichkeiten benutzen die Studierenden im Rahmen der Mathematik 1 Veranstaltung und im dazugehörigen Tutorium?

Diese Frage dient dazu, das Spektrum der digitalen Werkzeuge zu erfassen, welches von den Studierenden verwendet wird.

(F2): Wie nutzen die Studierenden digitale Medien im Kontext der Mathematik 1 Veranstaltung?

Diese Frage soll einen Überblick über die verschiedenen studentischen Lernprozesse und die Verwendung digitaler Medien geben.

(F3): Wie haben sich die Lernprozesse hinsichtlich der

Online-Kommunikation innerhalb und außerhalb der Vorlesung entwickelt?

Mit dieser Fragestellung werden die Auswirkungen der Corona-Semester auf die Lernsituationen untersucht. Dazu sollen Rückschlüsse aus den Erfahrungen der Studierenden mit der reinen Online-Lehre für die zukünftige Nutzung der digitalen Medien gezogen werden.

Als weiterführende Analyse sind die zentralen angewandten digitalen Kompetenzen unter der folgenden Fragestellung herauszuarbeiten:

(F4): Welche zentralen digitalen Kompetenzen müssen in Zukunft bei den Studierenden in den Lernsituationen der Mathematik 1 Veranstaltung gefördert werden?

Aus den Interviews werden die digitalen Kompetenzen herausgearbeitet, auf denen der Fokus im Projekt liegt. Für diese Kompetenzen werden später Konzepte zur Förderung erarbeitet.

## 2 Durchführung

Insgesamt wurden 11 Studierendeninterviews im Zeitraum von Oktober 2021 bis Mai 2022 durchgeführt. Im vorherigen Semester (März 2021 bis September 2021) fanden ausschließlich Online-Vorlesungen statt. Ab Oktober 2021 fanden die Vorlesungen teilweise online und teilweise in Präsenz statt.

	Semester	Studiengang	Vorausbildung
1	2.	Maschinenbau	Ausbildung
2	6.	Maschinenbau	Fachabitur
3	7.	Maschinenbau	Fachabitur
4	1.	Maschinenbau (dual)	Abitur
5	4.	Maschinenbau	Meisterausbildung
6	2.	Maschinenbau	Ausbildung
7	2.	Digital Engineering	Fachabitur
8	2.	Digital Engineering	Ausbildung
9	1.	Wirtschaftsingenieurwesen	Ausbildung
10	2.	Digital Engineering	Ausbildung / anderes Studium
11	7.	Maschinenbau	Abitur

**Tabelle 2:** Übersicht zu den Interviewteilnehmern

Alle Studierenden befinden sich bezüglich der bestandenen Prüfungen noch im Grundstudium (1. bis 4. Semester). Die Tabelle 2 zeigt die Übersicht der Interviewteilnehmer\*innen mit Semester und vorheriger Ausbildung. Die Auswahl der Studierenden spiegelt die Heterogenität der Studienanfänger\*innen an der Fakultät wider. Für das Interview wurde eine offene Gesprächsatmosphäre geschaffen, in der die Teilnehmenden ihre Ansichten und Erfahrungen ohne Bewertung der Aussagen äußern konnten. Es wird keine quantitative Analyse des Nutzungsverhaltens durchgeführt.

Grundlage für die Studie sind qualitative Interviews, bei denen die Studierenden nach Möglichkeit frei und offen ihre Erfahrungen teilen können. Dieser Aspekt ist vor allem in der Form des narrativen Interviews wiederzufinden. Ein Rahmen für das Interview wird durch die von Helfferich [2] entwickelte SPSS-Methode geschaffen. So entsteht der folgende Leitfaden mit fünf zentralen Aspekten:

1. Tagesablauf bezüglich Vor- und Nachbereitung einer Mathematik 1 Vorlesung
2. Digitale Recherche bei mathematischen Rechenproblemen (bspw. bei Übungsaufgaben) und bei der Recherche im Bezug auf Verständnis der theoretischen Inhalte
3. Kommunikation mit dem Kommilitonen im digitalen Raum
4. Änderungen in der Kommunikation und der Vorgehensweisen bei der Recherche bezogen auf die Corona-Semester
5. Vermittlung von Kenntnissen bezüglich der Anwendung von digitalen Tools oder Informationsbeschaffungsmöglichkeiten

Die nachfolgende Auswertung der Interviews beruht auf der strukturierenden Inhaltsanalyse mit deduktiver Kategoriebildung nach Mayring [3] (eine Übersicht darüber sowie das Vorgehen der inhaltlichen Kategoriebildung findet sich im Kapitel 5.5.4). Diese bietet die Chance theoretische Modelle direkt in die Kategoriebildung der Inhaltsanalyse mit einfließen zu lassen.

Als theoretische Basis für die Kategorien wird der „Digital Competence Framework for Citizens“ (auch DigComp 2.2 genannt) nach Vuorikari et al. [5] verwendet. Er besteht aus 5 Kompetenzfeldern mit jeweils mehreren Kompetenzen. Diese beziehen sich auf folgende Bereiche (Übersetzung von Autor\*in):

- Informations- und Datenkompetenzen
- Kommunikation und Zusammenarbeit
- Erstellung digitaler Inhalte
- Datensicherheit
- Problemlösen

Für die Analyse werden die theoretischen Grundlagen mit den Forschungsfragen verknüpft. Auf dieser Basis entsteht ein Categoriesystem zur deduktiven Auswertung der Interviews. Wie in Tabelle 3 zu sehen, ergeben sich sechs Analysekatoren.

Kategorien	
1	Rechercheprozesse
2	Kommunikationsplattformen
3	Kommunikationswege
4	Erfahrungen im Umgang mit digitalen Medien
5	Motivation
6	Bedürfnisse

**Tabelle 3:** deduktives Categoriesystem nach Mayring [3]

Die ersten vier Kategorien beziehen sich auf die digitalen Kompetenzen und den Umgang mit digitalen Medien. Die Motivation und die Bedürfnisse der Studierenden sind wichtig zu erfassen, damit ein Konzept zur Förderung entworfen werden kann.

So ist die Motivation nach Van Dijk [4] die Basis zur Aneignung von digitalen Technologien. Ein erfolgreiches Unterstützungsangebot muss sich an den Bedürfnissen der Studierenden orientieren.

## 3 Analyse

Grundlage für die Analyse sind die Aussagen und Begründungen der Studierenden. Die Aufteilung der Analyse orientiert sich an den sechs deduktiven gebildeten Kategorien (siehe Tabelle 3).

### 3.1 Rechercheprozesse

Der Rechercheprozess wird bezogen auf zwei Schwerpunkte ausgewertet. Einerseits werden die eingesetzten Tools und Recherchemöglichkeiten analysiert, andererseits auch auf die Vorgehensweisen eingegangen. Es werden die Erläuterungen der Studierenden für das

eigene Nutzungsverhalten angeführt.

Für die Recherche nach digitalen Inhalten werden unterschiedliche Tools und Plattformen verwendet. Speziell die Google-Suchfunktion ist eines der Hauptrecherchertools für die Inhalte der Mathematik 1 Veranstaltung. Bei detaillierterem Nachfragen, warum die Suchfunktion eines der zentralen Informationsquellen nach mathematischen Inhalten ist, wurden die bereits vorhandene praktische Erfahrung aber auch die schnellen (und nach den Studierenden) korrekten Ergebnisse der Google-Suchanfrage genannt. Zwar wird die Google-Suchanfrage häufig benutzt, jedoch haben sich nur wenige bisher mit den dahinterliegenden Algorithmen und den Ergebnissen der daraus resultierenden Suchanfragen auseinandergesetzt. Eine Bewertung der Suchergebnisse auf Grund der Herkunft der Inhalte oder Qualität erfolgt nur selten. Sobald die mathematischen Inhalte die gewünschte Frage beantworten, wird die Quelle akzeptiert und deren Ergebnisse verwendet.

Neben der Google-Suchanfrage benutzen Studierende auch das Videoportal Youtube als Informationsquelle. Hier wurde am häufigsten der Name „Daniel Jung“ mit seinen Mathematik-Videos erwähnt. Diese sind auf Grund ihrer einfachen Erklärungen und des praktischen Rechnens sehr beliebt.

Zusätzlich werden das digitale Skript und Bücher aus der Online-Bibliothek von Teilen der Studierenden als Grundlage für die Recherchen benutzt. Sie werden jedoch nicht priorisiert verwendet. Ein Argument hierfür sind die konkreteren und aufgabenspezifischeren Ergebnisse der Google-Suchmaschine.

Bei mathematischen Rechenproblemen, die sich aus den zu bearbeitenden Aufgaben ergeben, verwenden die Studierenden ähnliche Vorgehensweisen. Als eines der ersten Möglichkeiten zur Recherche wird die Google-Suchmaschine genannt. Dort wird nach ähnlichen Aufgabentypen wie in der Aufgabenstellung gesucht. Darüber hinaus werden bei konkreten Aufgaben Tools wie Photomath (<https://photomath.com/de>) und WolframAlpha (<https://www.wolframalpha.com>) benutzt. Photomath ermöglicht es den Studierenden Aufgaben direkt per Kamera auf dem Smartphone zu scannen und die entsprechende Lösung mit Lösungsweg zu betrachten. WolframAlpha ermöglicht es beispielsweise Funktionen mit ihren Eigenschaften darzustellen.

Die Vorgehensweisen der Studierenden sind bei allen zur Verfügung stehenden technischen Möglichkeiten sehr divers. So haben einige Studierende Bedenken bei der Nutzung von Tools wie Photomath. Sie begründen

dies in der Möglichkeit den Rechenweg direkt anzeigen zu lassen. Sie möchten diesen eher selbst herausfinden als ihn direkt vorgelegt zu bekommen. Eine negative Auswirkung auf das Verständnis wird befürchtet. Insgesamt ist die Recherche nach einem ähnlichen Aufgabentyp bei Problemen mit Übungsaufgaben sehr ausgeprägt. So erhoffen sich die Studierenden schnelle und verständliche Antwortmöglichkeiten.

### 3.2 Kommunikationsplattformen

Die angewandten Kommunikationsplattformen mit Anwendungsbeispielen sind in Tabelle 4 beschrieben. Bei den Videokonferenzsystemen benutzen die Studierende weniger das hochschuleigene System BigBlueButton (BBB). Die Benutzung von beispielsweise Discord wird damit begründet, dass diese Plattform bereits bekannter ist als BBB. Kurzfristige Kommunikationen zu Terminen der Studierenden oder auch zu einfacheren mathematischen Rechenaufgaben werden über die Instantmessengerdienste (z.B. Whatsapp) abgehandelt. Mailprogramme werden eher für die Kommunikation mit Mitarbeitenden oder Professor\*innen benutzt, eher selten für direkte Kommunikation zwischen Studierenden.

Kommunikationsplattform	Anwendungsbeispiele
<b>Videokonferenzsysteme (Discord, BigBlueButton oder Skype)</b>	Gemeinsames Arbeiten der Studierenden, Beantwortung von mathematischen Rechenproblemen und Fragestellungen bei theoretischen Inhalten
<b>Instantmessenger-Gruppen (Whatsapp oder Discord-Chatfunktion)</b>	Terminabsprachen, einfache Fragestellungen bei Problemen mit mathematischen Rechnungen
<b>Mailprogramme</b>	interne Kommunikation in der Hochschule (Konakt zu Mitarbeitern und Professor*innen)

**Tabelle 4:** digitale Kommunikationsplattformen und deren Anwendungsbeispiele der Studierenden

### 3.3 Kommunikationswege

In den Online-Semestern war die Kommunikation zwischen den Studierenden deutlich geringer als vorher. In den darauffolgenden Präsenzsemester ist die digitale Nutzung der Kommunikationswege stark lernotypabhängig. Manche Studierende arbeiten eher individuell an

den mathematischen Inhalten und Fragestellungen und bevorzugen teilweise die Distanz der Online-Vorlesung. Andere Studierende haben erst durch die Präsenz angefangen Lerngruppen zu bilden, welche nun teilweise in der Hochschule und teilweise im digitalen Raum stattfinden. Dies sind Anzeichen dafür, dass zwar digitale Kommunikation ein wichtiger Baustein bei der Zusammenarbeit der Studierenden ist, diese aber nicht ohne persönliche Treffen und Besprechungen funktioniert.

Die Fortführung der digitalen Form der Zusammenarbeit auch nach den Corona-Semestern beruht auf der Zeitersparnis bei den Studierenden. So müssen diese sich nicht erst einen Raum suchen, um zusammen an den Aufgaben arbeiten zu können. Sie treffen sich per Videokonferenzsystem (teilweise sehr regelmäßig, z.B. einmal die Woche) zur Bearbeitung der anstehenden Übungsaufgaben. Für die Kommunikation im digitalen Raum besitzen 9 von 11 befragten Studierenden ein Tablet mit Stift. Sie begründen den Kauf der digitalen Hardware auf der einen Seite mit den Corona-Semestern, auf der anderen Seite zählen sie die Vorteile der digitalen Mitschriften auf:

Die Unterlagen sind überall und jederzeit (im Tablet) verfügbar. Es existiert nur noch die Hardware, weitere Unterlagen oder Ordner müssen nicht mehr transportiert werden. Die Suche in den Dokumenten ist stark vereinfacht, ebenfalls kann über wenige Clicks direkt auf die Google-Suchfunktion umgeschaltet werden, sodass auch bei der Recherche Zeit eingespart wird. Weiter ist der Austausch mit den Kommilitonen durch die digitale Form vereinfacht. So lassen sich Fragestellungen schnell per Instantmessenger teilen oder Besprechungsinhalte per Videokonferenzsystem visualisieren.

### 3.4 Erfahrungen im Umgang mit den digitalen Medien

Die Erfahrungen im Umgang mit den digitalen Medien der Studierenden beruht zu großen Teilen auf praktisch erlernten Fähigkeiten. Nur selten haben sie mit deren Recherchevorgehen oder auch den entsprechenden Tools (bspw. Google-Suchmaschine oder daraus resultierende Ergebnisse) intensiv auseinandergesetzt. Dabei ist auch zu beachten, dass eine diverse Anzahl von Recherchemöglichkeiten in den Interviews genannt wurde. Diese sind auch abhängig vom individuellen Lernverhalten der Studierenden. Manche nehmen die Inhalte besser über Videos auf, andere wiederum bevorzugen die schriftliche Form in Büchern. Insgesamt suchen sie

sich zwar das für sich selbst optimale Medium aus, dennoch befassen sie sich wenig damit. Ebenfalls bietet die bisherigen Formen der Vorlesung (größtenteils in frontaler Vermittlung der Inhalte mit Bereitstellung von Skripten) nur wenig Ansätze für die Ausprägung der digitalen Recherchefähigkeiten der Studierenden, da die Inhalte und Literaturempfehlungen schon vom Dozenten validiert sind.

### 3.5 Motivation

Aus den Interviews lässt sich erkennen, in welcher Form die Studierenden motiviert arbeiten. Studierende berichten, dass insbesondere die Arbeit in Kleingruppen im digitalen Raum sehr effektiv ist.

Hingegen äußerten viele Studierende, dass das Format der Videoplattform das Konzept der Mathematik 1-Veranstaltung nicht unterstützt. Sie begründen dies damit, die Kommunikation zwischen der dozierenden Person und den Studierenden ein zentraler Baustein ist. Dieser ist im digitalen Raum (Videokonferenzsystem) stark reduziert.

Daneben erkennen viele Studierende, dass bei komplexeren Fragestellungen ihre Rechercheprozesse nicht zu ausreichenden Ergebnissen führen. An diesem Punkt kann ein Unterstützungsangebot anknüpfen.

### 3.6 Bedürfnisse

Ein wesentliches Bedürfnis der Studierenden bezüglich einer Recherche ist, dass diese schnell valide Ergebnisse liefert. Dies begründen sie mit wenig Zeit für die Mengen an gestellten Aufgaben.

Der zeitliche Aspekt der Studierenden muss bei der Entwicklung der Unterstützungsmaßnahmen berücksichtigt werden.

Daneben haben die Studierenden das Bedürfnis, ihre Rechercheergebnisse beurteilen zu können. Dies bereitet ihnen teilweise Schwierigkeiten.

## 4 Ergebnisse

Die digitalen Plattformen (F1) ergeben eine Vielzahl von angewendeten Tools für unterschiedliche Informationsbeschaffungs- und Kommunikationsprozesse. Besonders im Vergleich zu den herkömmlichen (nicht digitalen) Medien ist die Nutzung ausgeprägter. Die Recherche- sowie die Kommunikationsmedien wurden in Kapitel 3.1 und 3.2 erläutert.

Im Bezug auf die Nutzung der digitalen Medien (F2)

find eine Analyse hinsichtlich der Recherche sowie der Kommunikation statt (siehe Kapitel 3.1 und 3.3). Bei der Recherche wurde zwischen den Problemstellungen beim mathematischen Rechnen (z.B. bei Übungsaufgaben) und denen der theoretischen Inhalte (z.B. der An-eingung der Quotientenregel) unterschieden. Insgesamt lässt sich erkennen, dass verschiedene Studierende ähnliche Recherche- und Kommunikationsprozesse abhängig von ihrem Lernverhalten (individuell- oder gruppenorientiert) verwenden.

Bezogen auf die Erfahrungen und Auswirkungen aus den Online-Semestern (F3) vertrat ein großer Teil der Studierenden die Meinung, dass reine Online-Vorlesungen kontraproduktiv für deren Studium seien. Dies wird mit der geringeren Konzentrationsfähigkeit begründet. Viele Studierende berichten ebenfalls, dass das Lernen in Online-Arbeitsgruppen ihnen in den Online-Semestern geholfen hat. Diesen Aspekt gilt es bei der Planung von zukünftigen Unterstützungsangeboten zu beachten.

Um auch die letzte Fragestellung (F4) nach den digitalen Kompetenzen bewerten zu können, ist eine Analyse hinsichtlich des theoretischen Rahmens aus Kapitel 3 notwendig. Der beschriebene Kompetenzrahmen setzt sich aus fünf Kompetenzfeldern mit jeweils mehreren Kompetenzen zusammen. Im ersten Schritt werden die Kompetenzfelder eingeschränkt.

Kompetenzfeld	Kompetenz	
Informations- und Datenkompetenz	Browsing, Suche und Filterung	... von Daten, Informationen und digitalen Inhalten
	Auswerten	
	Verwalten	
Kommunikation und Zusammenarbeit	Interaktion	... mit Hilfe von digitaler Technologie
	Teilen	
	Zusammenarbeit	
Problemlösen	Lösung von technischen Problemen	
	Bedarfsermittlung von technischen Lösungen	

**Abbildung 1:** Übersicht über die angewandten Kompetenzen nach dem DigComp 2.2 [5]  
In der Übersicht in Abbildung 1 sind die ausgewähl-

ten Kompetenzfelder mit den dazugehörigen Kompetenzen zu sehen (von Autor\*in übersetzt).

Aus Kapitel 2.1 zeigt sich, dass die drei Kompetenzen des Bereichs der Information- und Datenkompetenzen die von den Studierenden beschriebenen Prozesse des Recherchierens von mathematischen Rechenproblemen und Inhalten widerspiegeln. Diese behandeln das Suchen, Auswerten und Verwalten von digitalen Inhalten.

In Abschnitt 3.2 und 3.3 wird die Zusammenarbeit mit Kommilitonen im digitalen Raum erläutert. Dafür werden die digitalen Kompetenzen der Kommunikation und Zusammenarbeit benötigt.

Für die Anwendung der beschriebenen Tools und Kommunikationsplattformen sind die Kompetenzen des digitalen Problemlösens unablässig. Sie geben an, inwiefern die Studierenden die Fähigkeiten beherrschen technische Problemstellungen (z.B. den Ausfall des Internets am Computer) eigenständig zu beheben. Dazu gehört die Fähigkeit, das richtige digitale Hilfsmittel in einer Situation auswählen zu können.

Die Aussagen der Studierenden beziehen sich hinsichtlich der digitalen Kompetenzen weniger auf die Erstellung und Entwicklung von digitalen Inhalten. Als Folgerung daraus wird das Feld der Erstellung digitaler Inhalte in diesem Zusammenhang nicht weiter betrachtet.

Der Bereich der Datensicherheit ist ein wichtiges Gebiet der digitalen Kompetenzen und ist hochschulweit von großer Bedeutung. Allerdings lässt die Vielzahl von zu betrachtenden Kompetenzen eine genauere Fokussierung dieses Feldes nicht zu.

Um im nächsten Schritt das Kompetenzlevel der einzelnen Studierenden einschätzen zu können, muss für jede Einstufung eine entsprechende Beschreibung mit Anforderungen vorhanden sein.

## 5 Fazit

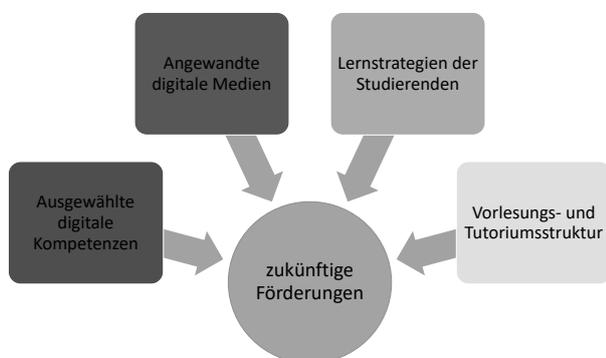
Insgesamt entsteht durch die durchgeführten Interviews eine objektivierte Beschreibung der Vorgehensweisen im Bezug auf Recherche und Kommunikation der Studierenden. Sie besitzen die technischen Fähigkeiten die vorhandenen Tools in ihren Lernalltag im Bezug auf die Mathematik 1 Veranstaltung zu integrieren. Bei den angewandten Rechercheprozessen reflektieren die Studierenden ihre Vorgehensweisen nur wenig. Diesbezüglich spielt auch die heterogene Vorausbildung in der

Studierendenschaft keine Rolle. Die Förderung der Reflexion dieser Prozesse ist ein zentraler Baustein für die zu entwickelnden Unterstützungsangebote für alle Studierenden.

Die Reflexion der Vorgehensweisen soll praxisnah an Anwendungsbeispielen erfolgen. Insgesamt soll so die Motivation und die Bedeutsamkeit des Themas für die Studierenden aufgezeigt werden. Nach Van Dijk [4] muss diese ausgeprägt sein, damit die Studierenden bereit sind ihre Kompetenzen zu stärken. Um dies zu erreichen ist es essenziell ihnen die Notwendigkeit zur digitalen Kompetenzoptimierung im Bereich der Mathematik 1 Veranstaltung, aber auch darüber hinaus, zu erläutern.

Um den Bedürfnissen der Studierenden entgegenzukommen ist der zeitliche Aufwand für die Unterstützungsangebote zu begrenzen. Dies soll über eine direkte Integration in die vorhandenen Abläufe der Mathematik 1 Veranstaltung geschehen. Beispielsweise könnte das Feedback von Rechercheprozessen direkt im Anschluss an vorhandene Übungsaufgaben mit höherem Informationsbeschaffungsbedarf erfolgen.

Für die zukünftigen Förderungen werden in der Abbildung 2 die Einflussfaktoren zusammengefasst.



**Abbildung 2:** Einflussfaktoren für die Erstellung der Förderung

Neben der hier beschriebenen Entwicklung des Unterstützungsangebots muss in Zukunft auch eine Anpassung der Räumlichkeiten und der Technik in der Hochschule erfolgen. Durch die Veränderungen des digitalen Lernverhaltens (bspw. Nutzung von Tablets zum Rechnen von Übungsaufgaben), entsteht ein höherer Bedarf nach digital ausgestatteten Lernplätzen mit Bildschirmen für Gruppenarbeit, Steckdosen und entsprechenden Anschlüssen für diverse Geräte. Dies muss bereitgestellt

werden, damit die Studierenden auch in Zukunft optimal mit den (digitalisierten) Materialien arbeiten können.

Im nächsten Schritt sollen die digitalen Kompetenzlevel bezogen auf den Kontext der Mathematik 1 Veranstaltung beschrieben werden. So ist es möglich individuelle Potenziale der Studierenden nach einer Einzelbewertung zu erkennen. Über eine Reflexion der Recherche- sowie der Kommunikationswege werden Unterstützungsmöglichkeiten in das akute Konzept mit integriert. Es wird so erhofft, dass die Studierenden die Reflexion der beschriebenen Prozesse direkt in den Studienalltag der Mathematik 1 Veranstaltung integrieren.

## Quellen

- [1] Grosseck, G. (2009). To use or not to use web 2.0 in higher education? *Procedia - Social and Behavioral Sciences*, 1, 478-482. <https://doi.org/10.1016/j.sbspro.2009.01.087>
- [2] Helfferich, C. (2011). *Die Qualität qualitativer Daten. Manual für die Durchführung qualitativer Interviews* (4. Auflage ed.). VS Verlag.
- [3] Mayring, P. (2015). *Qualitative Inhaltsanalyse. Grundlagen und Techniken* (12., überarbeitete Auflage ed.). Beltz. [https://doi.org/10.1007/978-3-658-21308-4\\_42](https://doi.org/10.1007/978-3-658-21308-4_42)
- [4] Van Dijk, J. (2013). Digitale Spaltung und digitale Kompetenzen. *Informationsgerechtigkeit: Theorie und Praxis der gesellschaftlichen Informationsversorgung*, 108-133.
- [5] Vuorikari, R., Kluzer, S., & Punie, Y. (2022). *DigComp 2.2, The Digital Competence framework for citizens : with new examples of knowledge, skills and attitudes*. Publications Office of the European Union. <https://doi.org/10.2760/115376>



# Industrial Methods and Digital Tools for Craft Professionals – A Use Case Deploying Discrete Event Simulation and Virtual Reality

Bastian Prell\*, Jörg Reiff-Stephan

Technical University of Applied Sciences Wildau, Hochschulring 1, 15745 Wildau, Germany;

\*[bastian.prell@th-wildau.de](mailto:bastian.prell@th-wildau.de)

**Abstract.** This paper presents an integrated approach to explore human factors in the craft sector. The embedded use case consists of craft workshops being 3D-modelled and simulated so that different scheduling rules can be deployed. The connected VR visualisation allows craft professionals to experience scheduling rules and their impact on different performance measurements in an environment intuitively resembling their actual workshops. The lessons learned from the simulation modelling are collected to be presented as a use case. A modular approach appears to be beneficial for the simulation modelling, offering flexibility and pragmatism but detail where needed. Apart from this anecdotal evidence, the study is accompanied by interviews. These are analysed based on abduction grounded theory which combines a strong focus on the actual corpus of qualitative data but considers theoretical foundations, when appropriate. This paper presents the study design as it has not yet been conducted completely but is ongoing.

## Introduction

Industry 4.0 is a well-known term for production scientists and practitioners. However, the typical craft workshop is usually only partially, if at all, automated. While the selective use of automation can be justified by smaller production quantities and lower economic efficiency, such argumentation cannot fully transfer to digitisation trends. Many Industry 4.0 solutions had a strong technical focus. In contrast to that, most recent approaches, with the label Industry 5.0, pay more attention to human factors in the production environment [1, 2]. This may offer connecting points to the value creation in the craft sector.

One of the tools used in this context is discrete event simulation (DES) to model and simulate material flows through the production facility. DES is linked to the term

digital factory. For example, it is used to virtually check production layouts for both new and replanned of systems or production scheduling. In contrast to that, the craft sector lags behind in applying digital tools. However, the requirements from both areas show certain similarities, as both value on-time delivery, throughput, capacity utilisation, lead-times etc.

## 1 Research Context

Compared to industrial value creation, increased labour productivity has been accomplished only to a lesser extent in the craft sector. Among other things, crafts are characterised by the fact that tools are operated manually. In contrast to that, industrial processes are mostly characterised by machine-guided tools. Ancillary processes, however, are not covered by this distinction but are similar to those in the industrial sector. Productivity describes the relationship between input factors and output over time, which is closely linked to the term technology from an economics point of view. New technologies are usually described by the term innovation. This section presents a brief introduction to the current state of the art for craft research, innovation diffusion, technology acceptance, simulation, and virtual reality (VR).

### 1.1 Craft Research

The craft sector is multifaceted and almost exclusively characterised by mostly small and some medium-sized enterprises [3]. Crafts account for 12.4 % of employment and 25 % of apprenticeships in Germany. Roughly 8 % of Germany's gross domestic product (GDP) is generated in the craft sector [4]. With the disproportionately large share of apprentices, it can be assumed on the one hand that there is sufficient digital literacy potential [5]; on the other hand, that the integration

of digital content into training is of great significance [6].

Compared to enterprises from the industrial sector, digital tools are used less intensively in the craft sector [7]. The difference and the need to catch up in terms of digitalisation, has already been the subject of various studies [8–11].

The craft sector has been affected disproportionately by the demographic change and will continue to do so: A large proportion of workers in this sector will retire in the next five to ten years. The general demographic of smaller cohorts following the so-called baby boomers won't be able to compensate for that. Furthermore, it can be observed that the choice of training by these overall fewer potential applicants, also in relative terms, was taken less and less in favour of craft professions [12]. At the same time, it is assumed that the demand for craft professionals will remain constant or even increase [13]. Thus, everything points to an increasing labour shortage in this sector. This calls for new technologies that could potentially increase labour productivity. Thus, those innovations need to emerge or be transferred and spread (diffuse) throughout the craft sector.

## 1.2 Innovation Diffusion

The fact that the craft sector does not use the potential of existing technologies to the same extent as the industrial sector raises the question of how innovations emerge in the craft sector. Innovation theory has produced various models to describe innovation processes. The traditional linear innovation model [14], which is based on a neo-classical understanding and thus often assumes an exogenous technology push, does not seem to be able to explain the different degrees of utilisation in industry and crafts: Technology and utilisation are open to both sectors. Considering individual perceptions and mutual influence between individuals and sectors seems to fit better to the focused craft sector. More current models from the group of systemic approaches, such as the *Technological Innovations Systems Approach*, for example, include those behavioural aspects [15].

Such behavioural aspects can arise, for example, from individual acceptance (or non-acceptance in case of rejection). Acceptance has been formalised through different models, whereas the *Technology Acceptance Model* (TAM) is the most commonly used. TAM has emerged less systematically than the aforementioned innovation

theory models, but from requirements management for specific solutions [16]. Originally designed to assess the acceptance of information and communication technology, especially computer programs, various extensions (TAM2 and TAM3) have been established for a wide range of applications and have proven to be robust [17]. The TAM assumes that acceptance is directly related to the actual use of a certain technology [18], as it is depicted in Figure 1. In the context of basic innovation theory terminology, rooted in Schumpeter's work, this relates to what makes up an innovation in the first place: Namely, an invention or idea that becomes established by wide application [19]. The TAM makes it possible to interview probands before and after exposing them to novel technologies and to determine influencing factors and weightings through factor analysis.

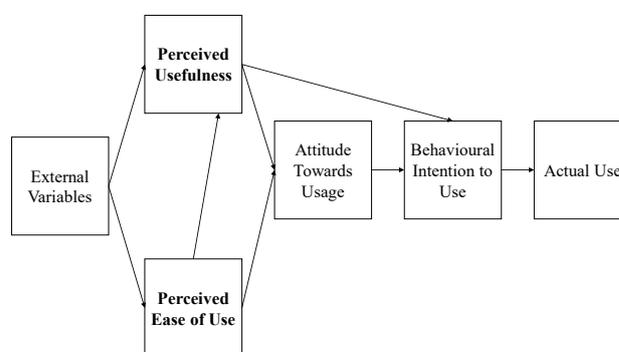


Figure 1. Modelled Technology Acceptance

## 1.3 Simulation in Production and Logistics

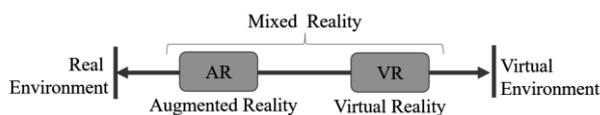
“A simulation is the imitation of the operation of a real-world process or system over time” which may be advised for a broad range of applications from operations research and systems analysis [20]. Simulation has been established for both planning new and replanning existing processes, systems, or factories in production and logistics. While a variety of different software suites has evolved to cover a wide area of application, the method of DES has become especially popular for many scopes of production and logistics. Bracht et al. classified visualisation as either to be dynamic or static to describe whether the visualisation is dynamically changing during the simulation run [21, 22]. Animation is a special type of visualisation, according to VDI 3633 Sheet 11 [23]. Nevertheless, animation is no synonym for simulation, which describes the method of computing a system's behaviour [22]. Wenzel points out that not only insights but also the communication of these insight ought to be achieved by simulation [24]. Common use cases are

found in the automotive sector, commissioning, chemical industry, or food industry. It is striking that non-industrial production has not been examined with simulations, even though layout planning or scheduling is relevant in this sector as well. Recent developments have led to the establishment of 3D simulation, which is easily understood also by non-experts.

#### 1.4 Virtual Reality

Digitisation and process optimisation projects are often accompanied by the expectation that the results will have a potentially positive impact on the profitability of value creation and are always associated with risks of failure due to project-inherent uncertainties [25].

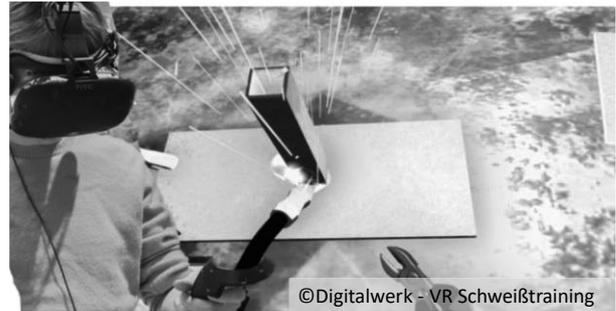
Hence, early transparency is of great importance, which can be supported by VR. Figure 2 depicts the so-called reality-virtuality continuum, according to [26]: The most left pole represents the real environment, with no virtuality. The augmentation of real objects by context-based information is called augmented reality. The reason for that is, that real objects make up the majority of elements in the user's field of vision resp. perception [27]. Meanwhile, VR is usually created by displays that exclude real-world objects. Instead, a virtually created world is displayed and mostly synchronised to head movements to realistically stimulate human senses [28]. VR is not enabling a completely virtual environment, as humans, generally speaking, have other senses that are still connected to reality and not yet stimulated virtually, e.g. sense of balance, smelling, etc. Together, AR and VR make up the term *mixed reality* [29].



**Figure 2.** Reality-Virtuality Continuum according to [26]

According to the classification of Reif either head-mounted displays or emulators support the visualisation. For the proposed use case, the used HTC Vive counts as a head-mounted display, which is powered by an external computer, running the simulation [30]. The provided hardware allows six degrees of freedom, for best immersion: Three for rotative movements, usually of the head like yawing, rolling, and pitching, and the three translatory motion forward, leftward, or rightward [31].

VR applications can be already found in practice: Education, production planning, or process optimisation as well as the visualisation of products for customers are common applications [32]. One study found training, planning, and communication to be the most relevant topics for VR in crafts [33]. Figure 3 shows a welding simulation, which is one of the examples used in the aforementioned study. Furthermore, visualisation is stated for planning, even though the authors don't state layout planning explicitly for production environments but kitchen.



**Figure 3.** Welding Training in VR

VR offers a rather intuitive orientation and the possibility of a controlled environment that can be of high value, e.g. in training scenarios that are not feasible otherwise, as they might be too expensive or too dangerous. Potential drawbacks of such a technology is the so-called motion sickness [34].

## 2 Research Design

The research design employs use case analysis as it is common in engineering science to anecdotally pass on experiences of designing and implementing technical systems. That way, the lessons learned can be deduced and documented. The novelty of this topic strongly points towards explorative research. That is why, the proposed design integrates grounded theory from the field of social science. It is used to create hypotheses about human behaviour and social life from field data, which consists mainly of interview transcriptions and observations [35]. More precisely, abduction grounded theory is used, as theoretical elements from literature, as presented in section 1, are considered as well. The use case presents a convenient way to get in touch with the research object (craft professionals) and exposing them to DES and scheduling, which are state-of-the-art digital industrial methods. This *technology shock* is then used to track reactions from individuals and their associated groups, respective the company. The remainder of this paper

discusses the identified use case and the planned data collection and analysis, while further aspects of the research design are displayed in section 4, once the use case has been described.

### 3 Use Case Description and Realisation

This section describes the activities planned to create a DES model and to display scheduling alternatives to craft professionals. Figure 4 represents the overall model of procedure. The main subtasks were adopted to a reduced extent from VDI 3633 [36]. The standard process (grey arrow) is deployed on an abstract level. This is to create modules of different layouts that may occur or make up actual workshops of craft companies. These modules were defined as flow shop or job shop layouts. Hence, modular simulation models for each of these two are implemented (blue arrow) using synthesised data, representing typical workshops of this sector. For the modelling of the actual workshop, the real-world data needs to replace the modules' dummy data, and parameters need to be set accordingly. Especially the 3D visualisation needs to reflect the specifics of the craft workshop. This can be done using 3D-design software such as Blender or CAD software like SolidWorks or Autodesk Inventor.

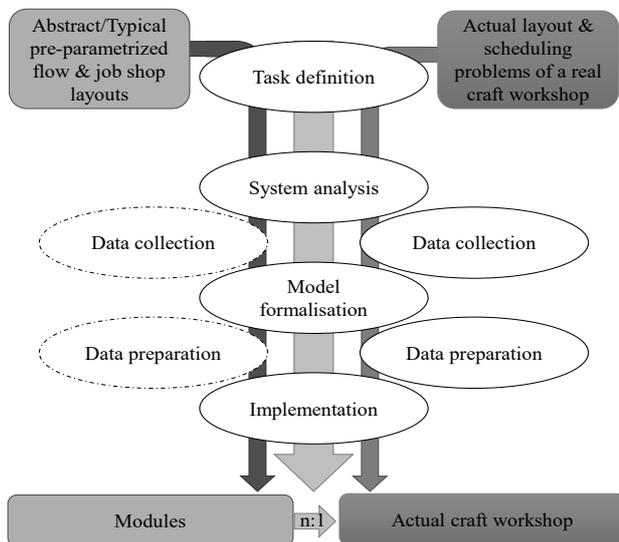


Figure 4. Model of Procedure for Simulation Creation

These 3D objects are then imported into the simulation software by Siemens Tecnomatix Plant Simulation. Only machines relevant for the schedule are designed in detail to ensure reasonable usage of resources. Other machinery

or the workshop surrounding can be integrated as panes displaying 2D photos of the respective objects. This allows a shortcut in modelling while ensuring intuitive resemblance. Finally, a functioning virtual 3D image of the workshop is created (orange arrow).

#### 3.1 Layout Patterns

This subsection presents the two main layout options that are thought to modular build up most craft workshops: Flow shop and job shop, which also determine the respective scheduling problems.

**Flow Shop Problem (FSP)** represents a scheduling problem where all jobs have the same processing sequence [37]. Usually, a flow shop is characterised by a flow-oriented layout, so that machines are arranged in series, as shown in Figure 5. This allows jobs to flow from an initial machine, through several intermediate machines to the final machine. This means that each operation after the first has exactly one direct predecessor, and each operation except the final machine has exactly one direct successor [38].

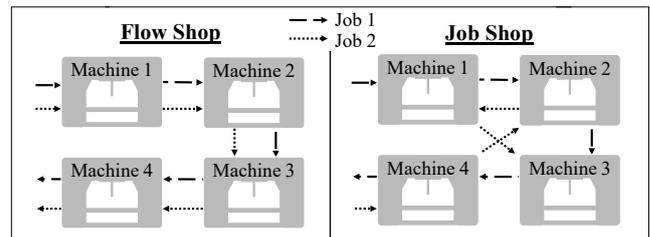


Figure 5. Flow Shop and Job Shop Layout Principles

Common objectives for FSPs are to find schedules that minimise longest completion time or makespan while maximising machine utilisation. This possibly leads to the highest throughput [39]. Flow shop production is usually associated with portfolios of only a few, highly standardised products, with predictable demand. Thus, the most common production strategy is *make-to-stock*. The number of constraints to consider, when scheduling, is significantly less than for a job shop layout. For instance, the setup time is rarely considered since setups do not occur often. A flow shop provides good visibility of occurring problems due to the linear layout.

**Job Shop Problem (JSP)** consists of jobs, running through a common production facility but having different machine orders [40]. Jilcha & Berhan [41] and Li et

al. [42] present JSP using the following definition: Job shop scheduling happens in a work location of a given set of general-purpose workstations where a variety of jobs, each with a specific set of operations, is processed in a given sequence. Jobs, therefore, have their individual routing and might use all machines of the respective layout or a subset of these (see Figure 5).

A JSP's objective is usually to minimise the makespan, subject to the constraints specified for each job. Overall characteristics of production in a job shop environment can usually be summarised as follows: Firstly, the variety of products is very high due to the possibility of customisation. Therefore, the production layout requires high flexibility which leads to low process standardisation. Additionally, demand is harder to predict than for the typical FSP-associated product portfolio, since it usually operates under a make-to-order strategy. Admittedly, not only customised or single pieces are produced in job shop configuration, but small batches as well. Finally, regarding the workers' skills, professionals working in job shop layouts need to master more tasks, hence they require more training in their respective qualifications.

Scheduling production in a job shop environment should consider the following aspects: The make-to-order strategy is linked to the target of on-time delivery. Overlooking the process is much more complex since every product has its specific routing.

### 3.2 Priority Rule-based Scheduling

Priority rules have been used for decades as a scheduling procedure for production. Its main objective is to define the sequence in which the pending jobs are processed, in a given period. It is known that priority rules will usually provide less optimal solutions in comparison to what may be given by other advanced algorithms, e.g., genetic algorithms, neural networks, simulated annealing, etc. However, it is commonly used because its implementation is easy and does not require expert knowledge nor skills [43]. As more complex approaches may be feasible for some craft companies, the authors assess rule-based scheduling to be more suitable for most companies in the craft sector. Hence, scheduling shall be limited in this paper's scope to priority rules. The Encyclopedia of Production and Manufacturing Management [44], as well as Koruca & Aydemir [43], present in their work the most commonly used rules for production scheduling. Ten of these rules are introduced in Table 1 and will be implemented in the simulation modules.

Figure 6 depicts the implications from these rules, using 5 jobs (J1-J5) with varying process times.

Rule	Description
<b>First come, First served (FCFS)</b>	Jobs will be scheduled and processed according to which was first to arrive either in the queue or to the machine.
<b>Last come, first served (LCFS)</b>	The last job to arrive is the one that is scheduled or processed next.
<b>Shortest processing time (SPT)</b>	When scheduling the job, the one with the shortest processing time among those in the queue is processed next. This reduces the work in process inventory, the total flow time, and the average job lateness.
<b>Longest processing time (LPT)</b>	This rule will schedule the job with the longest processing time among the jobs in the queue, next. While using this rule for scheduling the total completion time or makespan will be minimised.
<b>Earliest due date (EDD)</b>	When applying this rule, the job with the earliest due date will be processed next. The main aim of the rule is to reduce job lateness.
<b>Shortest remaining processing time (SRPT)</b>	The job that will have the priority of being processed is the one that has the shortest remaining processing time. The main aim of this rule is to minimise the total completion time or makespan and minimise the latest job delivery time.
<b>Longest remaining processing time (LRPT)</b>	The job that has the longest remaining processing time will be processed next by the machine. The aim of this rule is to maximise the capacity utilisation of machines/workstations in the work location.
<b>Slack time (ST)</b>	Is a variant of earliest due date rule, considering the remaining processing time (setups and lead time). The job that has the smallest amount of slack gets top priority. This rule integrates customer orientation and capacity utilisation.
<b>Next Queue (NQ)</b>	While scheduling, the NQ rule considers the queues at each of the succeeding machines/workstations where the jobs are heading. The priority goes to the job whose machine/workstation has the smallest queue.

**Table 1:** Priority-based scheduling rules

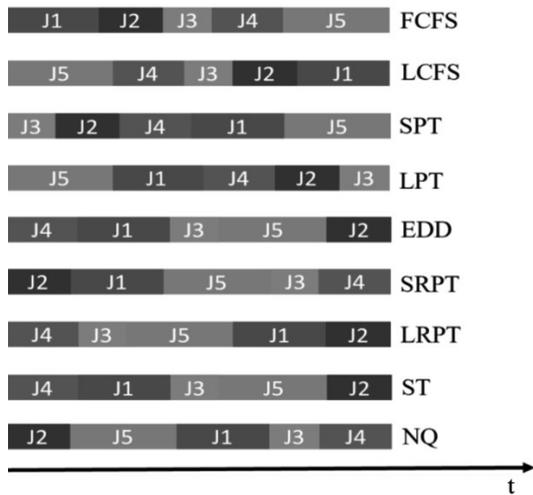


Figure 6. Priority-based scheduling rules, and their exemplary impact on the sequence of jobs

## 4 Further Course of the Study

According to the outlined research design, the use case of scheduling is embedded in the simulation and application of VR by craft professionals. The following subsections cover the methodological integration and the interviews.

### 4.1 Study and Use Case Integration

The use case analysis and the human factor analysis can be seen as two outcomes of this integrated work, which are different but not separate, as presented in Figure 7.

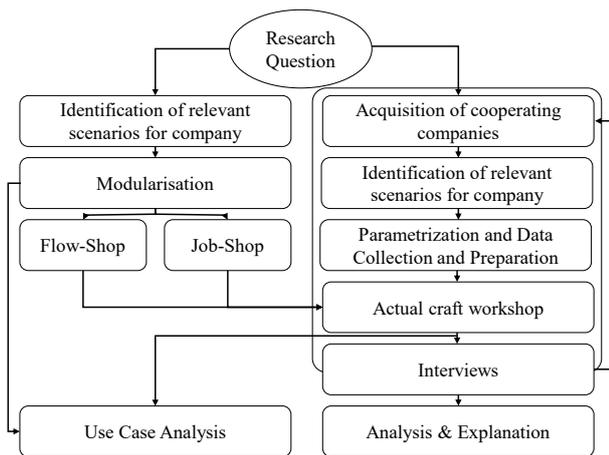


Figure 7. Model of Procedure for Conducting the Study

The work on the use case is currently ongoing as well as the acquisition of cooperating companies. The research questions have been defined as:

- How is the innovation process characterised in the crafts sector? What roles do various individual and social stakeholders play?
- What kind of innovation-type classifications are feasible for craft companies?
- Which special features can promote the prevalence of novel technologies and how are they related?

The presented research design offers a path to find underpinning explanatory hypotheses for those research questions. The study that encloses the actual applications is designed and the data collection and analysis are prepared accordingly to the following subsection. A minimal feasible model has been created and software evaluated that allows realistic but also pragmatic modelling of the actual machinery. The adaption to the first actual workshop is the next step to be conducted.

### 4.2 Interview Preparation

Conversational interviews are the preferred interview technique for qualitative research at the exploratory stage [45]. Semi-structured guidance can promote story-telling by the interviewees. The target is to capture accounts of experience and behaviour related to the research question respectively leading to hypotheses. Thus, the theoretical background of section 1 was considered for the design of the guiding questions [46]. Nonetheless, the interviews should only be carefully guided, avoiding any priming of interviewees when it comes to attitudes and sentiments.

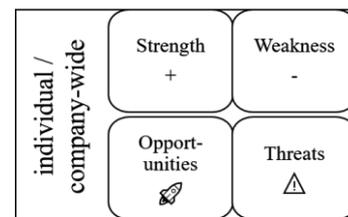


Figure 8. SWOT framework for Interview Guidance

Guidance can try to steer the conversation towards topics of internal or external communication, the used channels of communication and their recipients, usefulness, job relevance, ease of use, the perceived benefit and usage intention. For a similar approach, assessing a different use case the authors thought it useful to not reveal those subtopics directly nor the research question in order to achieve honest accounts. Therefore, the conversation was supported by a SWOT (Strength, Weakness, Opportunity and Weaknesses) template (Figure 8). This

template is shortly explained to retain the interviewees' attention for the rest of the interview. This was assessed to cover the actual research questions well. Nevertheless, the interviewees ought to be made aware of the abstract topic of technology transfer into the craft sector for reasons of integrity. Provided the interviewees give their permission, the interviews are to be directly voice recorded or otherwise transcribed with great detail. Furthermore, impressions from the interview setting should be noted. This so-called corpus of qualitative data is then analysed by interpretation, identification, and assignment of patterns and the deduction of explanatory hypotheses.

## 5 Conclusion and Outlook

This paper presents a mixed-methods approach to exploratively study the acceptance and diffusion of innovations within the craft sector while realising a DES use case. A DES model of an actual craft workshop is intended to employ different scheduling scenarios, and show the effect on different performance indicators that are relevant across the sectors of industry and crafts. Scheduling, in this case, represents an industrial method that is demonstrated using digital technology, which is novel to the craft sector. Both, method and technology are not used extensively in crafts and, therefore, offer innovation potential. How craft professionals assess this potential and respective usage within their field of work on an individual but also a company-wide level is explored through accompanying conversational interviews.

This paper presents the outline of the research design and the underpinned model of procedure for the actual realisation of the use case. The study is currently at the phase of the use-case realisation and a minimal viable DES has been implemented. The authors hope to employ the proposed design on different occasions, receiving thick feedback data that can be documented and analysed, to formulate detailed hypotheses for technology acceptance and innovation diffusion in the craft sector.

### Acknowledgements

This publication was made possible through the funding of the PhD program Innovation and Career Center - Integrated Engineering by the state of Brandenburg's Ministry of Research, Education and Culture MWFK (Germany).

## References

- [1] Günther, N., Reiff-Stephan, J.: Prescriptive Education im Zuge der Industrie 5.0. In: Jäkel, J. and Thiel, R. (eds.) AALE. pp. 315–320. VDE Verlag, Leipzig (2020). [https://doi.org/10.1007/978-3-642-51663-4\\_17](https://doi.org/10.1007/978-3-642-51663-4_17).
- [2] Prescher, T., Hellriegel, J., Schön, M., Baumann, A., Heil, M., Schulz, F.: Digitalisierung im Handwerk als Lernprozess fördern. CEUR Workshop Proc. 1669, 209–215 (2016).
- [3] statistisches Bundesamt: Produzierendes Gewerbe. (2019). <https://doi.org/2040720177004>.
- [4] Zentralverband des deutschen Handwerks: Kennzahlen des Handwerks, <https://www.zdh.de/daten-fakten/kennzahlen-des-handwerks/>, last accessed 2020/06/23.
- [5] Günther, N., Reiff-Stephan, J.: Future Skills für die Produktion von Morgen. In: AALE 2019. , (2019).
- [6] IG Metall: Das große Ding: Handwerk 4.0 Wie Betriebe und Beschäftigte erfolgreich den digitalen Wandel meistern Die Gewerkschaft für das Handwerk. (2015).
- [7] Peters, T.: Branchenanalyse SHK-Handwerk: Aktuelle Herausforderungen und Chancen. (2016).
- [8] Mittelstand 4.0-Kompetenzzentrum Hamburg: Digitalisierung im Hamburger Handwerk: Status Quo und Bedarfe 2021, <https://www.kompetenzzentrum-hamburg.digital/themen/handwerk/355-handwerk-status-quo-und-bedarfe-2021>, last accessed 2021/12/03.
- [9] Owen, A., Plöger, W., Hiltner, G., Reith, A.: Digitalisierungsbarometer für das Bau- und Ausbauhandwerk Kurzfassung Eine empirische 360° Analyse. (2020).
- [10] Thonipara, A., Höhle, D., Proeger, T., Bizer, K.: Digitalisierung im Handwerk - ein Forschungsüberblick. Göttinger Beiträge zur Handw. 36, (2020). <https://doi.org/10.3249/2364-3897-gbh-36>.
- [11] Veltkamp, N., Schulte, K.-S.: Digitalisierung des Handwerks. (2020).
- [12] Müller, K.: Dialog und Perspektive Handwerk 2025. I, (2016).
- [13] Creditreform Wirtschaftsforschung: Wirtschaftslage und Finanzierung im Mittelstand 2020/2021. 1–5 (2021).
- [14] Rogers, E.M.: Diffusion of innovations. Free Press, New York, NY [u.a.] (2003).
- [15] Greenacre, P., Gross, R., Speirs, J.: Innovation Theory: A review of the literature. ICEPT Work. Pap. 11, 1–46 (2012).
- [16] Davis, F.D.: User Acceptance of Information Systems: The Technology Acceptance Model (TAM), <http://www.jstor.org/stable/249008>, (1989).
- [17] Venkatesh, V., Davis, F.D.: Theoretical extension of the Technology Acceptance Model: Four longitudinal field studies. Manage. Sci. 46, 186–204 (2000). <https://doi.org/10.1287/mnsc.46.2.186.11926>.

- [18] Jockisch, M.: Das Technologieakzeptanzmodell, (2010). [https://doi.org/10.1007/978-3-8349-8484-5\\_11](https://doi.org/10.1007/978-3-8349-8484-5_11).
- [19] Hauschildt, J., Salomon, S.: Innovationsmanagement. Vahlen, München (2011).
- [20] Banks, J., Carson, J., Nelson, B., Nicol, D.: Discrete-event System Simulation. Prentice Hall (2010).
- [21] Bracht, U., Geckler, D., Wenzel, S.: Digitale Fabrik. Springer Heidelberg Dordrecht London New York (2011). <https://doi.org/10.1007/978-3-540-88973-1>.
- [22] Gutenschwager, K., Rabe, M., Spieckermann, S., Wenzel, S.: Simulation in Produktion und Logistik. Springer Berlin Heidelberg, Berlin, Heidelberg (2017). <https://doi.org/10.1007/978-3-662-55745-7>.
- [23] VDI Verein Deutscher Ingenieure: VDI Richtlinie 3633, Einführungsblatt (2014).
- [24] Wenzel, S.: Verbesserung der Informationsgestaltung in der Simulationstechnik unter Nutzung autonomer Visualisierungswerkzeuge. Verlag Praxiswissen, Dortmund (1998).
- [25] Knothe, T., Reiff-Stephan, J., Vladova, G., Ullrich, A.: Industrie 4.0 im Produktionsumfeld. In: Metamorphose zur intelligenten und vernetzten Fabrik. Springer Vieweg, Berlin (2017). <https://doi.org/10.1007/978-3-662-54317-7>.
- [26] Milgram, P., Takemura, H., Utsumi, A., Kishino, F.: Augmented reality: A class of displays on the reality-virtuality continuum. In: Telemanipulator and Telepresence Technologies. pp. 282–292 (1995). <https://doi.org/10.1117/12.197321>.
- [27] Azuma, R.T.: A Survey of Augmented Reality. Presence Teleoperators Virtual Environ. 6, 355–385 (1997). <https://doi.org/10.1162/pres.1997.6.4.355>.
- [28] Steuer, J.: Defining Virtual Reality: Dimensions Determining Telepresence. J. Commun. 42, 73–93 (1992). <https://doi.org/10.1111/j.1460-2466.1992.tb00812.x>.
- [29] Grothus, A., Thesing, T., Feldmann, C.: Digitale Geschäftsmodell-Innovation mit Augmented Reality und Virtual Reality. Springer Berlin Heidelberg, Berlin, Heidelberg (2021). <https://doi.org/10.1007/978-3-662-63746-3>.
- [30] Reif, R.: Development and Evaluation of an Augmented Reality Order Picking System, (2009).
- [31] Prithul, A., Adhanom, I.B., Folmer, E.: Embodied Third-Person Virtual Locomotion using a Single Depth Camera. In: Proceedings of Graphics Interface 2021. pp. 210–219. Canadian Information Processing Society (2021). <https://doi.org/10.20380/GI2021.24>.
- [32] KPMG: Neue Dimensionen der Realität. Eine Analyse der Potenziale von Virtual und Augmented Reality in Unternehmen. (2016).
- [33] HPI: Technologiemonitoring Steckbrief VR / AR. (2019).
- [34] Kim, E., Shin, G.: User discomfort while using a virtual reality headset as a personal viewing system for text-intensive office tasks. Ergonomics. 64, 891–899 (2021). <https://doi.org/10.1080/00140139.2020.1869320>.
- [35] Harmaz, K.: Constructing grounded theory: A practical guide through qualitative analysis. sage (2006).
- [36] VDI Verein Deutscher Ingenieure: VDI Richtlinie 3633, Blatt 1 (2014).
- [37] Rossit, D.A., Vásquez, Ó.C., Tohmé, F., Frutos, M., Safe, M.D.: The Dominance Flow Shop Scheduling Problem. Electron. Notes Discret. Math. 69, 21–28 (2018). <https://doi.org/10.1016/j.endm.2018.07.004>.
- [38] Baker, K., Trietsch, D.: Flow Shop Scheduling. In: Principles of Sequencing and Scheduling. pp. 283–317. Wiley (2018).
- [39] Tellache, N.E.H., Boudhar, M.: Flow shop scheduling problem with conflict graphs. Ann. Oper. Res. 261, 339–363 (2018). <https://doi.org/10.1007/s10479-017-2560-x>.
- [40] Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B.: Chapter 9 Sequencing and scheduling: Algorithms and complexity. Presented at the (1993). [https://doi.org/10.1016/S0927-0507\(05\)80189-6](https://doi.org/10.1016/S0927-0507(05)80189-6).
- [41] Kassu, J., Eshetie, B.: Job Shop Scheduling Problem for Machine Shop with Shifting Heuristic Bottleneck. Glob. J. Res. Eng. 158, (2015).
- [42] Li, Y., Goga, K., Tadei, R., Terzo, O.: Production Scheduling in Industry 4.0. Presented at the (2021). [https://doi.org/10.1007/978-3-030-50454-0\\_34](https://doi.org/10.1007/978-3-030-50454-0_34).
- [43] Koruca, H., Aydemir, E.: A Priority Rule Based Production Scheduling Module on Faborg-Sim Simulation Tool. Gazi Univ. J. Sci. 27, 1143–1155 (2014).
- [44] Swamidass, P.M. ed: Priority Scheduling Rules. In: Encyclopedia of Production and Manufacturing Management. pp. 527–528. Springer US, Boston, MA (2000). [https://doi.org/10.1007/1-4020-0612-8\\_708](https://doi.org/10.1007/1-4020-0612-8_708).
- [45] Dexter, L.A.: Elite and Specialized Interviewing. Northwestern University Press (1970).
- [46] Glaser, B.G., Strauss, A.L.: The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine Transaction (1967).

# Simulation Technischer Systeme

## Simulation of Technical Systems

Entwurf einer Modellbibliothek für die kostengünstige Funktionsentwicklung mechatronischer Systeme (L)

*Sven Jacobitz, Jie Zhang, Xiaobo Liu-Henke*

RobSimulation: Virtuelle Messzelle zur Planung, zum Testen und zur Optimierung von Prüfanlagen und –abläufen (A)

*F. Widdascheck, J. Riemenschneider, L. Schneebeli, D. Garth, C. Assmann, N. Lohse, G. Battenberg*

Entwicklung der adaptiven lokalen Trajektorienplanung zur Realisierung des kollisionsfreien Fahrens (L)

*Jie Zhang, Xiaobo Liu-Henke, Thomas Vietor*

Gekoppelte Simulation des Einspritz- und Verbrennungsvorgangs eines Industrie-Dieselmotors (L)

*Daniel Jörss, Maximilian Ringel, Bert Buchholz, Christian Fink*

Automatisches Mapping des dynamischen Umfelds in einem cyber-physischen Systemen (L)

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

Konzept eines ganzheitlichen, hochflexibel konfigurierbaren HiL-Prüfsystems für den Test autonomer Fahrfunktionen (L)

*Xiaobo Liu-Henke, Marian Göllner, Sven Jacobitz, Jie Zhang*

Towards a Scenario Toolkit for Autonomous Systems (L)

*Ahmad Naja, Siddhartha Gupta, Umut Durak, Sven Hartmann*

Critical Scenario Techniques for Automated Vehicles: Literature Review (L)

*Kaan Akgün, Umut Durak, Siddhartha Gupta*

Iterative Scenario-Based Testing in an Operational Design Domain for Artificial Intelligence Based Systems in Aviation (L)

*Bojan Lukic, Jasper Sprockhoff, Alexander Ahlbrecht, Siddhartha Gupta, Umut Durak*

Simulation-based Verification of Functions for Autonomous Drones (L)

*Hamza Ghezali, Siddhartha Gupta, Umut Durak*

Langbeiträge (L) sind zu finden im

**ARGESIM Report 21 Proceedings Langbeiträge ASIM Workshop 2023 STS/GMMS/EDU**

ISBN ebook 978-3-903347-61-8, DOI 10.11128/arep.21, ARGESIM Report 21, ASIM Mitteilung 185

Kurzbeiträge (K) und Abstract-Beiträge (A) sind zu finden im

**ARGESIM Report 22 Kurzbeiträge & Abstract-Beiträge ASIM Workshop 2023 STS/GMMS/EDU**

ISBN ebook 978-3-903347-62-5, DOI 10.11128/arep.22, ARGESIM Report 22, ASIM Mitteilung 186

Abstracts der Hauptvorträge sind im **ARGESIM Report 21** und im **ARGESIM Report 22** zu finden.



# Entwurf einer Modellbibliothek für die kostengünstige Funktionsentwicklung mechatronischer Systeme

Sven Jacobitz<sup>1\*</sup>, Jie Zhang<sup>1</sup>, Xiaobo Liu-Henke<sup>1</sup>

<sup>1</sup>Institut für Mechatronik, Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel; \*sve.jacobitz@ostfalia.de

**Kurzfassung.** Aufgrund der stetig steigenden Komplexität und des hohen Vernetzungsgrades mechatronischer Systeme ist der Einsatz einer strukturierten, systematischen Entwicklungsmethode wie dem Rapid Control Prototyping (RCP) notwendig. Essenziell hierfür ist eine durchgängige Unterstützung durch eine CAE-Plattform. Solche Plattformen sind jedoch sehr kostenintensiv, weshalb an der Ostfalia die kostengünstige Plattform LoRra entstanden ist. Zentrales Element der Plattform ist die Modellbibliothek, welche über den gesamten Entwicklungsprozess den Zugriff auf einen konsistenten, rückverfolgbaren Datenstand ermöglicht. Durch Versions- und Konfigurationsmanagement wird zudem die Wiederverwendbarkeit der Prozess-Artefakte gesteigert. Der folgende Beitrag stellt den Entwurf der LoRra-Modellbibliothek für die kostengünstige Funktionsentwicklung mechatronischer Systeme vor.

## Einleitung

Die Komplexität und Funktionsumfang mechatronischer Systeme nehmen immer weiter zu. Für kleine und mittelständische Unternehmen (KMU) stellt dieser Trend eine große Herausforderung dar. Um konkurrenzfähig zu bleiben, müssen sie immer mehr intelligente Hard- und Software in ihre Produkte integrieren. Neben der Anzahl an Funktionen eines Systems liegt dies auch am stetig steigenden Vernetzungsgrad der komplexen Softwarekomponenten, welche in starker Wechselwirkung miteinander stehen [1]. Zur Handhabung immer kürzerer Entwicklungszeiten bei höheren Qualitätsansprüchen ist eine strukturierte, systematische Entwicklung solcher Systeme unabdingbar [2]. Das durchgängig modellbasierte Rapid Control Prototyping (RCP) ist in diesem Zusammenhang eine häufig eingesetzte Methodik. Essenziell für RCP ist die durchgängige Unterstützung durch eine Computer Aided Engineering

(CAE) Werkzeugkette, um einen hohen Automatisierungsgrad zu erreichen. Etablierte CAE-Werkzeugketten sind sehr kostenintensiv, was insbesondere für KMU eine große Hemmschwelle zur Einführung des RCP-Prozesses darstellt [3]. Im Rahmen des durch die EU geförderten Forschungsprojektes *Low-Cost Rapid Control Prototyping-System mit Open-Source-Plattform für die Funktionsentwicklung von eingebetteten mechatronischen Systemen (LoCoRCP)* ist daher an der Ostfalia die kostengünstige Entwicklungsplattform LoRra entstanden [4].

Um die Wiederverwendbarkeit von Modellen und Funktionen zu gewährleisten, wird ein systematisches Datenmanagement mit den dazugehörigen Teilprozessen wie Versions- und Konfigurationsmanagement eingesetzt. Dies ist aufgrund der hohen Vielfalt, Flexibilität und Kurzlebigkeit unabdingbar [5]. Erste Vorschriften hierzu kamen Anfang der 1960er Jahre bei der NASA auf [6]. Nach Sax et. Al. [7] werden Inkonsistenzen während der Funktionsentwicklung zunehmend durch die hohe Variantenvielfalt hervorgerufen, was durch Einsatz eines entsprechenden Konfigurationsmanagements vermieden werden kann. Im Rahmen des RCP wird hierfür eine CAE-basierte Modellbibliothek, welche sämtliche relevanten Artefakte (Ergebnis eines Teilprozesses wie z. B. Modelle, Programmquelltext oder Dokumentation) verwaltet, benötigt [8].

In der klassischen Softwareentwicklung ist die Versions- und Konfigurationsverwaltung weit verbreitet. Eine Übersicht gibt [9]. Ein häufig verwendetes Open-Source-Werkzeug ist GIT [10]. Jedoch liegt der Schwerpunkt dieses Werkzeuges in der änderungsbasierten Verwaltung von Textdateien [11]. Eine Anwendung dieses änderungsbasierten Ansatzes ist jedoch auf im MBD gängige Datenformate nur wenig praktikabel [12], weshalb für die Nutzung in einer Modellbibliothek entsprechende Anpassungen notwendig sind.

Um Art und Umfang der notwendigen Anpassungen zu identifizieren wurden z. B. durch Niedzwiedz und Frei [13] systematische Untersuchungen durchgeführt. Hierbei wird ein Modell standardisiert aus Metadaten, Schnittstelleninformationen und Parametern aufgebaut. Basierend auf solch einen standardisierten Aufbau kann auch ein Versions- und Konfigurationsmanagement für komplexe, zusammengesetzte Modelle unter Einsatz einer System Entity Structure (SES) erfolgen [14].

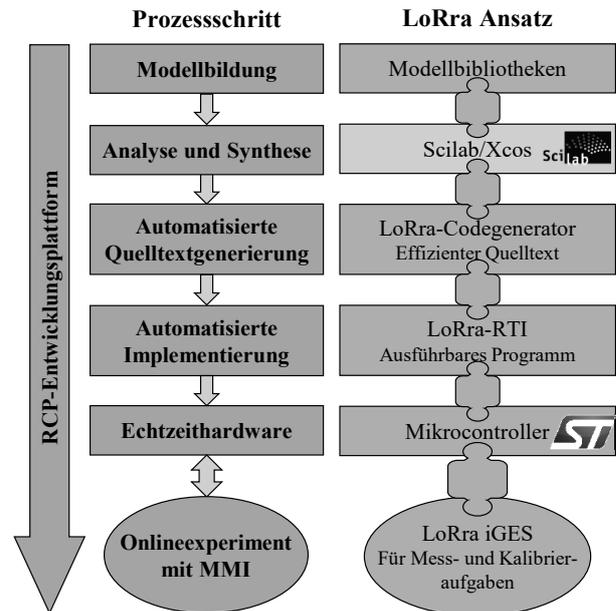
Im folgenden wird die CAE-basierte LoRra-Modellbibliothek konzipiert, entworfen und exemplarisch realisiert. Der weitere Beitrag ist wie folgt aufgebaut: Abschnitt 1 fasst die RCP-Entwicklungsmethodik zusammen und führt in die LoRra-Plattform ein. In Abschnitt 2 erfolgt anhand einer Analyse der Anforderung die Konzeption des Lösungsansatzes, welcher in Abschnitt 3 zu einem Entwurf konkretisiert wird. Abschließend erfolgt in Abschnitt 4 eine Beschreibung der Realisierung. Abschnitt 5 fasst die Ergebnisse zusammen und gibt einen Ausblick auf zukünftige Arbeiten.

## 1 Entwicklungsmethodik und -plattform

Aufgrund der hohen Systemkomplexität moderner vernetzter mechatronischer Systeme, wird der strukturierte, modellbasierte, verifikationsorientierte RCP-Prozess zur Entwicklung und Absicherung eingesetzt. Dieser besteht aus den Prozessschritten Modellbildung, Analyse / Synthese, automatisierte Generierung von Quelltext, automatisierte Implementierung auf einer Echtzeithardware sowie Onlineexperiment und wird durch Model-in-the-Loop- (MiL-), Software-in-the-Loop- (SiL-) sowie Hardware-in-the-Loop- (HiL-) Simulationen unterstützt [15].

Kennzeichnend für die vorgestellte Methodik sind die hohe Durchgängigkeit und der Automatisierungsgrad, von der Modellbildung, der modellbasierten Funktionsauslegung über die automatische Codegenerierung bis hin zur Echtzeitrealisierung (vgl. Abbildung 1 links). Sie wird durch eine durchgängige, voll automatisierte CAE-Plattform begleitet. Die modular aufgebaute, kostengünstige Entwicklungsplattform LoRra ist eine solche CAE-Plattform. Abbildung 1 illustriert den RCP-Entwicklungsprozess sowie die durchgängige Unterstützung mittels LoRra [4]. Von besonderer Relevanz ist hierbei eine zentrale Modellbibliothek, welche in sämtlichen Prozessschritten einen konsistenten, rückverfolgbaren Entwicklungsstand zur

verfügbar macht.



**Abbildung 1:** RCP-Entwicklungsprozess mit durchgängiger Unterstützung durch die LoRra-Plattform [4].

Für den Prozess der Modellbildung ist eine domänenübergreifende Modellbibliothek vorhanden. Mittels Versions- und Konfigurationsmanagement können Modellvarianten übersichtlich zusammengestellt und verwaltet werden. Das Open-Source-CAE-Werkzeug Scilab / Xcos wird zur Analyse und Synthese der Funktionen verwendet. Es bietet dabei einen ähnlichen Funktionsumfang wie das kommerziell häufig verwendete Matlab / Simulink. Das entstehende Funktionsmodell kann direkt in die Modellbibliothek integriert werden. Durch die offenen Schnittstellen der LoRra-API lassen sich zudem mit geringem Aufwand vorhandene Programme und Schnittstellentreiber einbinden. Mittels MiL-Simulationen können bereits in frühen Entwicklungsstadien Optimierungen und Tests der entwickelten Funktionen durchgeführt werden.

Durch den LoRra-Codegenerator wird mittels Modell-zu-Text-Transformation automatisiert effizienter, modularer C-Quelltext aus dem Funktionsmodell generiert. Durch offene funktionale Beschreibungen von Grundelementen des Modells, sogenannte Grundblöcke, ist der LoRra-Codegenerator flexibel erweiterbar. Der generierte Quelltext kann ohne manuelle Arbeiten, z.B. zur Optimierung und Test mittels SiL-Simulationen, wieder in das Xcos-Modell eingebunden werden.

Die Verbindung zu Modellen der Regelstrecke oder

weiteren Funktionen werden bei hinreichendem Funktionsstand durch Schnittstellenblöcke des LoRra Real-Time Interface (RTI) ersetzt. Hierdurch erfolgt ohne manuelle Programmierung die Ansteuerung der Echtzeithardware. In Kombination mit einer hardware-spezifischen RTI-Basissoftware, welche unter anderem ein Echtzeitbetriebssystem und standardisierte Schnittstellentreiber beinhaltet, wird somit eine automatisierte Implementierung auf der Echtzeithardware durch das RTI möglich. Als Echtzeithardware werden kostengünstige Mikrocontroller z.B. der Serie STM32H7 eingesetzt. Mittels HiL-Simulationen kann die entwickelte Funktion somit auch unter Echtzeitbedingungen optimiert und getestet werden. Als Mensch-Maschine-Interface (MMI) steht dabei die integrierte Graphikunterstützte Experimentiersoftware (iGES) zur Verfügung. Mit dieser lassen sich Onlineexperimente intuitiv steuern und überwachen sowie Messdaten aufzeichnen.

## 2 Konzeption der Modellbibliothek

### 2.1 Ansätze zur Entwicklung graphischer Bedienoberflächen

Für den strukturierten Softwareentwurf werden heutzutage standardisierte Architekturstile eingesetzt [16]. Diese dienen insbesondere zur Steigerung der Wiederverwendbarkeit, zur Strukturierung des Entwurfs sowie zur Schaffung eines einheitlichen Vokabulars. Über 25% der vorhandenen Stile dienen hierbei dem Entwurf einer Benutzeroberfläche [17]. Im Rahmen dieser Arbeit ist besonders das Modell / Präsentation / Steuerung (engl. Model / View / Controller, MVC) Prinzip relevant.

Der Architekturstil MVC in Anlehnung an [18] ist in Abbildung 2 illustriert. Hierbei werden die Visualisierung (Präsentation), die Steuerung und das Datenmodell getrennt voneinander mit definierten Schnittstellen realisiert. Die Steuerungskomponente reagiert auf Benutzereingaben in der graphischen Oberfläche und ändert bei Bedarf das Modell. Des Weiteren kann das Modell auch durch andere Softwarekomponenten geändert werden. Es benachrichtigt die Steuerung, über vorgenommene Änderungen, sodass diese die Präsentation aktualisiert. Aufgrund der geringen Komponentenverkopplung eignet sich dieses Prinzip besonders gut für MMIs, welche auf unterschiedlichen Zielplattformen eingesetzt werden [19]. So kann die Betriebssystem-

tem abhängige graphische Darstellung z. B. vollständig von der Steuerung und dem Modell entkoppelt werden.

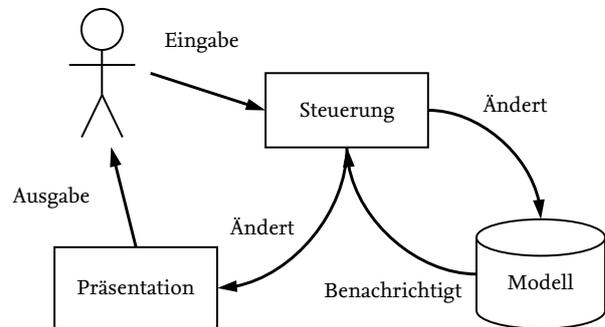


Abbildung 2: Architekturstil Modell / Präsentation / Steuerung.

### 2.2 Anforderungen an die Modellbibliothek

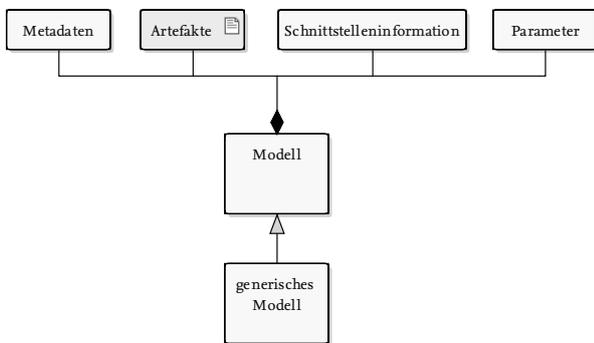
Wie aus Abschnitt 1 hervorgeht, ist die Modellbibliothek ein zentrales Werkzeug zum Datenmanagement in sämtlichen Prozessschritten. Um das Versions- und Konfigurationsmanagement in Gänze optimal zu unterstützen, werden übergeordnet folgende Anforderungen an die Modellbibliothek gestellt:

1. Versionierung sämtlicher enthaltener Daten sowie Unterstützung von Abläufen des Versionsmanagements (z. B. Prüfung, Freigabe).
2. Unterstützung der notwendigen Datenstrukturen zur Konfigurationsverwaltung in sämtlichen Prozessschritten sowie der Abläufe des Konfigurationsmanagements (z. B. Prüfung, Freigabe) zur Sicherstellung eines konsistenten Datenstands in sämtlichen Prozessschritten.
3. Hierarchische Strukturierung der Modelle in konfigurierbaren Kategorien und Hierarchieebenen.
4. Suchfunktion zum schnellen Finden spezifischer Modelle.
5. Unterstützung der Arbeit in verteilten Teams mit einer gemeinsamen Modellbasis.
6. Darstellung sämtlicher relevanter Modellinformationen in einer Übersicht.

## 2.3 Lösungsansatz

Um die Anforderungen an die Modellbibliothek umfassend zu erfüllen, wird zunächst der Aufbau eines Modells genauer betrachtet. Hierbei erfolgt eine Unterscheidung zwischen generischen und aggregierten Modellen.

Ein generisches Modell ist die kleinste in sich geschlossene Modelleinheit auf unterster Hierarchieebene. Sie untergliedert sich nicht in weitere hierarchisch geordnete Teilmodelle. Ein Beispiel für ein generisches Modell ist der elektrische Teil eines Gleichstrommotors, welcher sich durch Gl. (1) beschreiben lässt. Der Strukturierte Aufbau eines generischen Modells wird durch Abbildung 3 illustriert. Es besteht aus vier Komponenten:

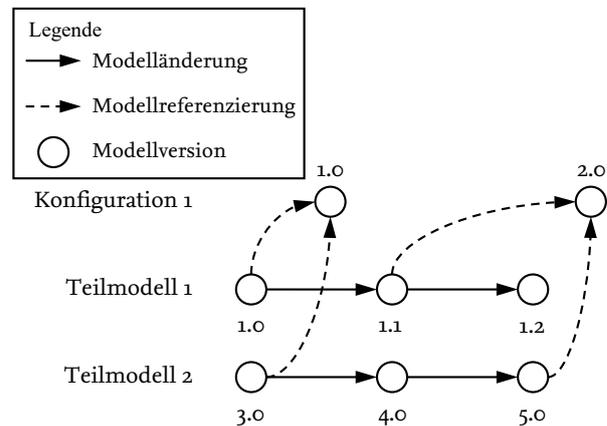


**Abbildung 3:** Strukturierter Aufbau eines generischen Modells.

- **Metadaten** beschreiben die übergeordneten Merkmale (z. B. Namen, Autor, allgemeine Beschreibung) des Modells.
- **Schnittstelleninformationen:** Datenstruktur, Einheiten und weitere relevante Informationen der Ein- und Ausgänge des Modells. Am Beispiel von Gl. (1) die Klemmspannung  $u$  in  $V$  als Eingang und der Motorstrom  $i$  in  $A$  als Ausgang.
- **Parameter:** Informationen und Werte zu den Parametern des Modells. Am Beispiel von Gl. (1) der Widerstand  $R$  in  $\Omega$  und die Induktivität  $L$  in  $H$ .
- **Artefakte** des Modells wie z. B. die (Xcos-)Modelldatei, der generierte C-Code oder die Modelldokumentation.

$$u = Ri + L \frac{di}{dt} - u_i \quad (1)$$

Ein aggregiertes Modell setzt sich aus weiteren Teilmodellen zusammen und bilden somit höhere Hierarchieebenen ab. Aggregierte Modelle werden als sogenannte Konfiguration in der Modellbibliothek abgebildet. Eine Konfiguration entsteht durch Integration definierter Versionsstände der Teilmodelle. Abbildung 4 illustriert das Prinzip.



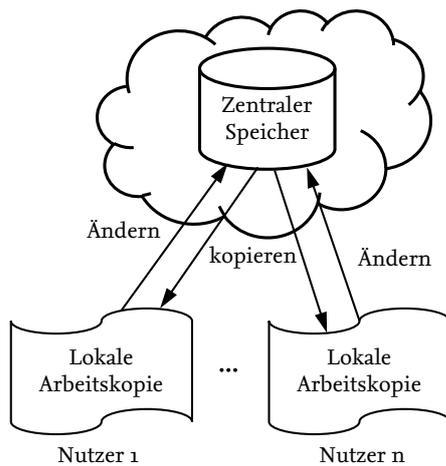
**Abbildung 4:** Prinzipielle Zusammensetzung einer Konfiguration.

Die Modelle sollen hierarchisch in einer Baumstruktur angeordnet werden. Diese Baumstruktur enthält Ordner (gruppierendes Hierarchieelement) und Modellelemente (sowohl generische als auch aggregierte Modelle). Sowohl für Gruppierungen als auch für einzelne Elemente können Benutzerrechte vergeben werden.

Um einen Modellzugriff für mehrere Benutzer zu ermöglichen, wird das Prinzip eines zentralen Speichers angewendet. Abbildung 5 verdeutlicht das Konzept. Über eine lokale Arbeitskopie erfolgt der Zugriff auf Modellartefakte. Änderungen werden an den zentralen Speicher, welcher als Datenbank fungiert, übermittelt und gespeichert. Nutzer können sich die geänderten Daten anschließend in ihre lokale Arbeitskopie kopieren.

## 3 Entwurf der Modellbibliothek

Das Konzept aus Abschnitt 2 wird im Folgenden konkretisiert und in einen Entwurf umgesetzt. Hierzu werden zunächst die Datenstrukturen und Schnittstellen sowie das Datenmanagement betrachtet und anschließend die graphische Oberfläche entworfen.



**Abbildung 5:** Konzept zur zentralen Speicherung von Modellen.

### 3.1 Datenstrukturen

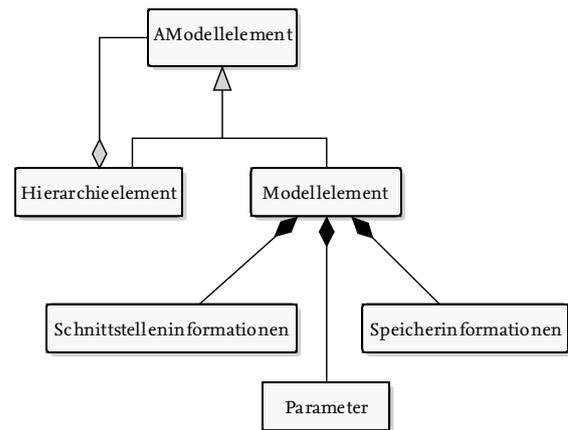
Für die Modellbibliothek werden verschiedene Datenstrukturen und Schnittstellen benötigt. Kernelement ist der hierarchische Modellbaum, welcher gleichzeitig als Datenbasis für die nach dem MVC-Prinzip aufgebaute graphische Oberfläche fungiert. Im folgenden wird exemplarisch die Datenstruktur und die Schnittstellen des Modellbaums entworfen.

Der Aufbau der Datenstrukturen erfolgt objektorientiert. Die abstrakte Klasse *AModellelement* repräsentiert die Grundstruktur für jedes Element des Baumes. Sie beinhaltet zentrale Daten wie Titel, Pfad im Baum oder Elternelement. Daraus abgeleitet werden die Klassen *Hierarchieelement* und *Modellelement*. *Hierarchieelement* beinhaltet eine Liste mit unterlagerten Elementen. *Modellelement* fasst neben weiteren Daten Schnittstelleninformationen, Parameter und Speicherinformationen des Modells zusammen. Abbildung 6 illustriert exemplarisch den Zusammenhang als UML-Klassenstruktur.

### 3.2 Datenmanagement

Das Datenmanagement der Modellbibliothek besteht im wesentlichen aus dem Versions- und Konfigurationsmanagement.

Um eine Versionierung und somit die konsistente Wiederverwendung in Form von Konfigurationen zu gewährleisten, muss die Modellbibliothek einen Versionsmanagementprozess unterstützen. Dies betrifft in erster Linie die Freigabe neuer Versionen. Wurden Än-



**Abbildung 6:** Klassenstruktur des hierarchischen Modellbaums.

derungen an einem Modell vorgenommen, darf die Nutzung in Form einer neuen Modellversion erst nach firmenspezifischen Freigabeprozessen erfolgen. Für die LoRra-Modellbibliothek bedeutet dies, dass von Benutzern vorgeschlagene Versionen erst für die Allgemeinheit freigegeben werden, nachdem die nach dem firmenspezifisch konfigurierten Prozess notwendigen Personengruppen dem zugestimmt haben.

Unterschiedliche Versionsstände eines Modells werden durch Versionsnummern der Form  $x.y$  gekennzeichnet. Hierbei wird  $x$  als Major-Version und  $y$  als Minor-Version bezeichnet. Wurden bei einer Änderung keine kompatibilitätsrelevanten Anpassungen am Modell vorgenommen (z. B. Fehlerbehebungen, Verhalten und Schnittstellen bleiben jedoch gleich), erfolgt lediglich die Inkrementierung der Minor-Version. Wurden Anpassungen vorgenommen, welche die Kompatibilität des Modells mit anderen Modellen beeinflussen (z. B. Änderungen an den Schnittstellen, Erweiterung der Funktionalität) wird die Major-Version inkrementiert und die Minor-Version zu 0 gesetzt. Die Zusammenstellung einer Konfiguration erfolgt durch Verknüpfung von Teilmodellen. Hierzu werden die entsprechenden Versionsnummern der Modelle referenziert. Abbildung 4 verdeutlicht das Prinzip.

### 3.3 Grafische Benutzeroberfläche

Die grafische Benutzeroberfläche (engl. Graphical User Interface, GUI) wird nach dem in Abschnitt 2.1 eingeführten MVC-Prinzip aufgebaut. Die Datenbasis hierfür (Modell) stellt der in Abschnitt 3.1 entworfene Modell-

baum dar. Hinzu kommen für verschiedene Aufgaben konzipierte Steuerungsklassen.

Abbildung 7 illustriert den Gesamtentwurf der graphischen Bedienoberfläche (Präsentation) der LoRra-Modellbibliothek. Das Hauptfenster ist in drei Bereiche unterteilt. Der Navigationsbereich (1) enthält die hierarchische Modellbaum der Bibliothek. Hierüber können Nutzende sowohl Aktionen für einzelne Modelle ausführen (z. B. öffnen oder bearbeiten) als auch einen ersten Überblick über den aktuellen Modellstatus erhalten. Zudem kann der Modellbaum durchsucht werden. Der Anzeigebereich (2) enthält verschiedene Ansichten zur Informationsdarstellung und -bearbeitung. Hier können beispielsweise die Metadaten und Modellartefakte angezeigt oder die Versionshistorie eingesehen werden. Zusätzlich ist im Werkzeugbereich (3) eine kontextabhängige Werkzeugleiste sowie die Menüstruktur zur Bedienung der Bibliothek angeordnet.

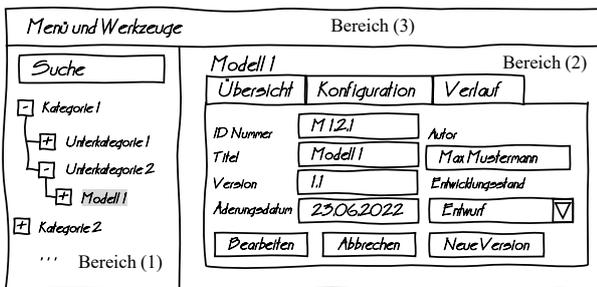


Abbildung 7: Entwurf der graphischen Oberfläche der Modellbibliothek.

## 4 Realisierung

Die Realisierung der Modellbibliothek erfolgt zunächst mit grundlegendem Funktionsumfang objektorientiert in Java als Eclipse Rich Client Platform (vgl. [20]). Das Eclipse-Framework bietet hierbei bereits viele zur Realisierung notwendige Mechanismen wie das *Standard Widget Toolkit* oder eine ereignisbasierte, Kopplungsarme Kommunikation zwischen verschiedenen graphischen Elementen. Zudem ist eine Vielzahl an Erweiterung mit offenen Schnittstellen nutzbar.

Die Versionsverwaltung wird gemeinsam mit der Anbindung an die zentrale Speicherinfrastruktur über das vorhandene Open-Source-Werkzeug GIT (vgl. [10]) realisiert. Hier sind bereits bewährte Mechanismen zur Versionierung vorhanden. Durch den Einsatz strukturierter, textbasierter, Modellbeschreibungen lassen sich

die einleitend erwähnten Einschränkungen umgehen. Zunächst erfolgt die Implementierung der Nutzerauthentifikation für den Atlassian Dienst Bitbucket. Eine spätere Erweiterung ist möglich.

Die strukturierte Modellbeschreibung erfolgt im JSON-Format (vgl. [21]). Listing 1 beinhaltet einen exemplarisch gespeicherten Modellbaum. Hierarchieelemente werden durch die Felder *title* (Anzeigetitel des Elements), *relPath* (relativer Dateipfad zum übergeordneten Hierarchieelement) und *children* beschrieben. Modellelemente enthalten die Felder *relPath*, *metaFileName* und *repoUrl* (URL zum Online-GIT-Repository). Sämtliche relevanten Metadaten werden in der unter *metaFileName* angegebenen Datei gespeichert.

Listing 1: Exemplarischer Modellbaum im JSON-Format.

```
{
  "title" : "root",
  "relPath" : "",
  "children" : [ {
    "title" : "Fahrzeugmodelle",
    "relPath" : "Fahrzeugmodelle/",
    "children" : [ ... ]
  }, {
    "title" : "Funktionsmodelle",
    "relPath" : "Funktionsmodelle/",
    "children" : [ {
      "title" : "VMS",
      "relPath" : "VMS/",
      "children" : [ ... ]
    }
  ],
  {
    "title" : "AMS",
    "relPath" : "AMS/",
    "children" : [ {
      "relPath" : "efm/",
      "metaFileName" : "efm.json",
      "repoUrl" : "https://tinyurl.com/repo_efm/"
    }, ... ]
  } ]
}, ... ]
}
```

Die Integration von Teilmodellen zu Konfigurationen erfolgt XML-basiert in Form einer SES. Hierdurch kann zunächst abstrakt die Struktur einer neuen Konfiguration erstellt werden. Durch Referenzierung der Teilmodelle unter Angabe von Version und Variante entsteht anschließend eine konkrete Konfiguration.

## 5 Zusammenfassung und Ausblick

Der vorliegende Beitrag stellt den Entwurf einer Modellbibliothek für die kostengünstige Funktionsentwicklung mechatronischer Systeme vor. Als Teil der auf Open-Source-Software basierenden durchgängigen RCP-Entwicklungsplattform LoRra bietet die Modellbibliothek in jedem Entwicklungsschritt eine konsistente und rückverfolgbare Datenbasis. Anhand der grundsätzlichen Anforderungen wurde ein Lösungsansatz für das Versions- und Konfigurationsmanagement hierarchischer Modelle sowie die zentrale Speicherung von Modellen erarbeitet. Anschließend wurden die Datenstruktur eines Modellbaums für hierarchische Modellkonfigurationen sowie die graphische Benutzeroberfläche entworfen. Abschließend erfolgte eine Zusammenfassung der Realisierung eines grundlegenden Funktionsumfangs.

Zukünftige Arbeiten befassen sich mit der weitergehenden Optimierung der Benutzerfreundlichkeit. Hierzu können weitere Funktionen zur Nutzerunterstützung (z.B. undo / redo) realisiert werden. Auch ist die Integration von graphischen Editoren, welche die Erstellung und Verwaltung von Konfigurationen vereinfachen, möglich. Für eine optimierte Übersicht in der Versionshistorie ist zudem eine Erweiterung des GIT-Werkzeugs *diff* vorgesehen, welche Änderungen in Xcos-Modellen visualisiert. Zuletzt ist eine Generalisierung der Nutzerauthentifikation möglich, sodass beliebige zentrale Speichersysteme genutzt werden können.

### Danksagung

Gefördert vom Niedersächsischen Ministerium für Wissenschaft und Kultur unter Fördernummer ZN3495 im Niedersächsischen Vorab der VolkswagenStiftung und betreut vom Zentrum für digitale Innovationen Niedersachsen (ZDIN).



## Literatur

- [1] Quantmeyer F, Liu-Henke X. Entwicklung eines hochflexiblen HiL-Systems zur Echtzeiterprobung des elektronischen Fahrzeugmanagements. In: *Effizienz, Präzision, Qualität*. Univ. 2013; pp. 1–10.
- [2] Liu-Henke X, Duym S. Modellgestützte Funktionsabsicherung des vernetzten mechatronischen Kraftfahrzeugs. In: *Mechatronik 2005*, VDI-Berichte. VDI-Verl. 2005; pp. 1073–1090.
- [3] Liu-Henke X, Feind R, Roch M, Quantmeyer F. Investigation of low-cost open-source platforms for developing of mechatronic functions with rapid control prototyping. In: *10th International Conference on Mechatronic Systems and Materials*. Opole, Polen. 2014; pp. 1–9.
- [4] Jacobitz S, Liu-Henke X. The Seamless Low-cost Development Platform LoRra for Model based Systems Engineering. In: *Proceedings of the 8th International Conference on Model-Driven Engineering and Software Development*. SCITEPRESS - Science and Technology Publications. 2020; pp. 57–64.
- [5] Broy M, Kuhmann M. *Projektorganisation und Management im Software Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg. 2013.
- [6] Versteegen G, Weischedel G. *Konfigurationsmanagement*. Berlin, Heidelberg: Springer Berlin Heidelberg. 2003.
- [7] Sax E, Reussner R, Guissouma H, Klare H. A Survey on the State and Future of Automotive Software Release and Configuration Management. 2017.
- [8] Zehetner J, Lu W, Watzenig D, Bernasch J. Co-Simulation und Modellbibliothek - Eckpfeiler für einen modernen Systementwicklungsprozess. In: *Berechnung, Simulation und Erprobung im Fahrzeugbau 2012*, VDI-Berichte. VDI-Verl. 2012; pp. 597–607.
- [9] Ratti N, Kaur P. Case Study: Version Control in Component-Based Systems. In: *Designing, Engineering, and Analyzing Reliable and Efficient Software*. IGI Global. 2013; pp. 283–297.
- [10] Preißel R, Stachmann B. *Git: Dezentrale Versionsverwaltung im Team : Grundlagen und Workflows*. Heidelberg: dpunkt.verlag, 5th ed. 2019.
- [11] Nugroho YS, Hata H, Matsumoto K. How different are different diff algorithms in Git? *Empirical Software Engineering*. 2020;25(1):790–823.
- [12] Schmitz D, Deng W, Rose T, Jarke M, Nonn H, Sanguanpiyapan K. Configuration Management for Realtime Simulation Software. In: *35th Euromicro*

*Conference on Software Engineering and Advanced Applications*. IEEE. 2009; pp. 229–236.

- [13] Niedzwiedz S, Frei S. A structured model library for the analysis of electric-vehicle drivetrains. In: *AmE 2012 - automotive meets electronics*, GMM-Fachbericht. VDE-Verl. 2012; pp. 21–26.
- [14] Durak U, Pawletta T, Oguztuzun H, Zeigler BP. System entity structure and model base framework in model based engineering of simulations for technical systems. In: *Proceedings of the Symposium on Model-driven Approaches for Simulation Engineering*. Society for Computer Simulation International. 2017; pp. 1–10.
- [15] 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. In: *Proceedings of the 16th International Conference on Software Technologies*. SCITEPRESS - Science and Technology Publications. 2021; pp. 215–223.
- [16] Starke G. *Effektive Softwarearchitekturen: Ein praktischer Leitfaden*. München: Hanser, 9th ed. 2020.
- [17] Henninger S, Corrêa V. Software pattern communities: current practices and challenges. In: *Proceedings of the 14th Conference on Pattern Languages of Programs - PLOP '07*. ACM Press. 2007; pp. 1–19.
- [18] Adams S. MetaMethods: The MVC paradigm. *HOOPLA!* 1988;1(4).
- [19] Liu Z, Li F, Liu H, Wu C, Zhang J. A Study of Cockpit HMI Simulation Design Based on the Concept of MVC Design Pattern. In: *Proceedings of the 2018 3rd International Conference on Modelling, Simulation and Applied Mathematics (MSAM 2018)*. Atlantis Press. 2018; pp. 82–84.
- [20] Steppan B. *Eclipse Rich Clients und Plug-ins*. München: Carl Hanser Verlag GmbH & Co. KG. 2015.
- [21] ISO/IEC 21778:2017: Information technology – The JSON data interchange syntax. *Standard*, International Organization for Standardization (ISO). 2017.

# Entwicklung der adaptiven lokalen Trajektorienplanung zur Realisierung des kollisionsfreien Fahrens in vernetzter Verkehrsumgebung

Jie Zhang<sup>1\*</sup>, Xiaobo Liu-Henke<sup>1</sup>, Thomas Vietor<sup>2</sup>

<sup>1</sup>Institut für Mechatronik Ostfalia Hochschule für angewandte Wissenschaften, Salzdhahmer Str. 46/48, 38302 Wolfenbüttel, Deutschland; \*[jie.zhang@ostfalia.de](mailto:jie.zhang@ostfalia.de)

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

**Abstract.** Der folgende Beitrag beschreibt die adaptive lokale Trajektorienplanung, welche als ein Teil der Trajektorienplanung des autonomen Fährbetriebs zur Kompensation der Unfähigkeit der globalen Trajektorienplanung bei Hindernisvermeidung in der dynamischen Verkehrsumgebung dient. Durch Integration der globalen und lokalen Trajektorienplanung wird eine optimale Trajektorie je nach Fahrstrategie (z.B. energieoptimal, zeitoptimal, distanzoptimal) ohne Konflikte mit anderen Verkehrsteilnehmern durch stetige Anpassung an die sich ändernde Verkehrsumgebung generiert. Anhand einer Pilotanwendung mit fahrerlosen Transportfahrzeugen (FTF) zum Materialtransport in der Intralogistik wird die entwickelte Funktion mithilfe eines virtuellen Prüfstandes abgesichert und aufgezeigt.

## Einleitung

Zentraler Bestandteil des autonomen Fährbetriebs ist neben dem Folgen einer ausgewählten Route die effiziente und konfliktfreie Trajektorienplanung, die sich mit dem Problem befasst, wie eine Reihe von Verhaltensweisen für die Fahrzeugführung in verschiedenen Szenarien schnell und präzise festgelegt wird und es so in die Lage versetzt, auf die sich dynamisch ändernde Verkehrsumgebung richtig zu reagieren [1]. So wird ein kontinuierlicher Fahrweg auf Basis der gewählten Route in Form einer Folge von Knotenpunkten zur Beschreibung der zeitlichen und räumlichen Bewegung eines Fahrzeugs inklusive Position, Fahrtrichtung und auch Translations- sowie Rotationsgeschwindigkeiten unter Berücksichtigung der Restriktionen des kinematischen sowie dynamischen

Verhaltens, ebenfalls unter Betrachtung der Zielkriterien (z.B. Energie, Sicherheit, Effizienz usw.), festgelegt, welche zur Optimierung der Fahrzeugführung beiträgt und als Führungsgröße einer Folgeregelung dient [2, 6]. Als Kopplungsglied der Zielführung mit der Fahrdynamikregelung steht die Leistung der Trajektorienplanung in direktem Zusammenhang mit der Fahrt von einer aktuellen Ist- zu einer vordefinierten Zielposition unter Beachtung der Geschmeidigkeit/Glätte des Fahrverhaltens.

Zur Erreichung des Ziels wird je nach der räumlichen und zeitlichen Dimension der Anwendungen und der Quelle der verfügbaren Verkehrsinformationen die Trajektorienplanung in zwei Teile, nämlich die globale und lokale Trajektorienplanung, unterteilt [3, 4]. Die globale Trajektorienplanung, auch statische Trajektorienplanung genannt, bezieht sich hauptsächlich auf die Bestimmung der Trajektorie zur Beschreibung der Positionen eines Fahrzeugs in Raum für die gesamte Fahrt von einer gegebenen Startposition bis zu einer Zielposition mit Hilfe der A-priori-Informationen, die durch eine digitale Karte bereitgestellt werden. Im Gegensatz dazu wird die lokale Trajektorieplanung, auch dynamische Trajektorienplanung genannt, als die Ermittlung des Übergangs der Fahrzeuge von einem Zustand in den nächsten in Echtzeit, basierend auf den über Mensch-Maschine-Interface (MMI) erfassten Komfortanforderungen und auf den Sensordaten oder verfügbaren Verkehrsinformationen durch V2X-Kommunikationstechnologie, der Fahrbahngrenzen, der Verkehrsregeln und der Hindernisse, insbesondere der dynamischen Hindernisse, welche in der globalen Trajektorienplanung mittels statischer Karte nicht vorhersehbar sind, definiert [3, 4, 5, 6].

Der vorliegende Beitrag fokussiert auf die Entwicklung der adaptiven lokalen Trajektorienplanung zur Vermeidung von Konflikten mit nicht vorhersehbaren Hindernissen (z.B. andere Fahrzeuge im selben Verkehrsszenario). Durch dessen Integration mit der globalen Trajektorienplanung erfolgt die konfliktfreie Fahrzeugführung in einer dynamischen Verkehrsumgebung. Durch eine Anwendung in einem Fahrerlosen Transportsystem (FTF) zum autonomen Materialientransport in der Intra-logistik wird die entwickelte lokale Trajektorienplanung in der virtuellen Simulationsumgebung abgesichert und demonstriert.

## 1 Stand des Wissens

In diesem Abschnitt wird zunächst der Ansatz Dynamic Window Approach (DWA) zur lokalen Trajektorienplanung und dann die Entwicklungsmethodik zur durchgängig modellbasierten und verifikationsorientierten Funktionsauslegung und -absicherung vorgestellt.

### 1.1 Dynamic Window Approach

Wie bereits erwähnt, ist es nicht sinnvoll während der globalen Trajektorienplanung die dynamischen und/oder plötzlich erscheinenden Hindernisse mitzubedenken. Eine Berücksichtigung erfordert eine große Rechenleistung und beeinträchtigt somit die Echtzeitfähigkeit. Zur Bewältigung dieser Herausforderung wird die lokale Trajektorienplanung auf Basis der globalen Trajektorie als Referenz entwickelt. Die Effizienz der lokalen Trajektorie hängt weitgehend von den verwendeten Ansätzen zur lokalen Trajektorienplanung ab [4, 6, 7].

DWA, ein weit verbreiteter und bewährter sowie effizienter Ansatz, wird in dieser Arbeit als Grundlage für die Entwicklung der adaptiven lokalen Trajektorienplanung verwendet. DWA wandelt die Fahrzeugkörperposition in Zeit und Raum in Bestimmung einer optimalen Kombination  $V(v, \omega)$  aus translatorischer und rotatorischen Geschwindigkeiten über einen bestimmten Zeitraum um, welcher als Prädiktionshorizont  $T$  bezeichnet wird [7]. Deshalb wird die Regelung der Fahrzeugposition über die Zeit als Regelung der Geschwindigkeit anhand des optimalen Geschwindigkeitsprofils  $V(v, \omega)$  in Echtzeit umgesetzt. Mithilfe einer speziell für die Aufgabenstellung entworfenen Bewertungsfunktion erfolgt eine verständliche und effektive Beurteilung, ob die betreffenden lokalen Trajektorien  $L_x$ , die anhand von Geschwin-

digkeitskombinationen innerhalb des Prädiktionshorizontes berechnet werden (vgl. Abb. 1), grundsätzlich fähig sind, das Fahrzeug sein vorbestimmtes Ziel konfliktfrei und effizient erreichen zu lassen. Anschließend wird die optimale Trajektorie durch die quantifizierte Lösungsgüte ermittelt. Gegenüber anderen Ansätzen liegen die Vorteile von DWA insbesondere darin, dass die ermittelte Geschwindigkeitskombination  $V$  als Sollwerte an die unterlagerte Fahrdynamikregelung zur Ansteuerung des Fahrzeugaufbaus direkt weiter übertragen werden kann [7, 8].

Zur Ermittlung der lokalen Trajektorien soll zuerst der Lösungsraum  $V_L$  etabliert werden, welcher mit den folgenden drei Randbedingungen abgegrenzt wird:

- Die maximale Geschwindigkeit  $V_m$  wird nicht nur durch die Fahrzeugkinematik beeinflusst, sondern auch gesetzlich durch die Obergrenze der zulässigen Höchstgeschwindigkeit begrenzt.
- Die Realisierbarkeit der Geschwindigkeit  $V_r$  muss unter Berücksichtigung der verfügbaren Leistung zum Antreiben und Bremsen innerhalb einer Abtastzeit  $\Delta T$  auf Basis der aktuellen Geschwindigkeit  $V_0(v_0, \omega_0)$  überprüft werden.
- Um die Fahrzeugsicherheit zu gewährleisten, muss das Fahrzeug beim Abbremsen mit maximaler Verzögerung  $(\dot{v}_{min}, \dot{\omega}_{min})$  auf null konfliktfrei mit Hindernissen bleiben. Daher wird die Sicherheitsgeschwindigkeit  $V_s$  durch Gl. 3 weiter eingegrenzt.

$$V_m = \{(v, \omega) | v \in [v_{min}, v_{max}], \omega \in [\omega_{min}, \omega_{max}]\} \quad (1)$$

$$V_r = \left\{ (v, \omega) \left| \begin{array}{l} v \in [v_0 + \dot{v}_{min} \cdot \Delta T, v_0 + \dot{v}_{max} \cdot \Delta T], \\ \omega \in [\omega_0 + \dot{\omega}_{min} \cdot \Delta T, \omega_0 + \dot{\omega}_{max} \cdot \Delta T] \end{array} \right. \right\} \quad (2)$$

$$V_s = \left\{ (v, \omega) \left| \begin{array}{l} v \leq \sqrt{2 \cdot dist(v_0, \omega_0) \cdot \dot{v}_{min}} \\ \omega \leq \sqrt{2 \cdot dist(v_0, \omega_0) \cdot \dot{\omega}_{min}} \end{array} \right. \right\} \quad (3)$$

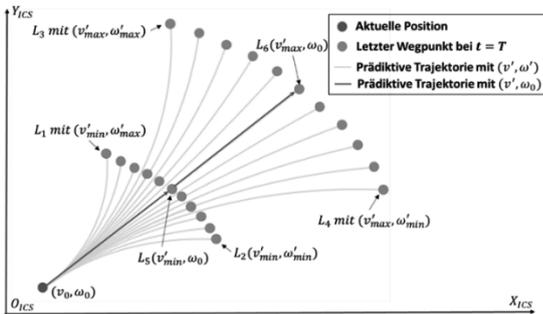
$$V_L = V_m \cap V_r \cap V_s \quad (4)$$

Mit der vordefinierten Geschwindigkeitsauflösung  $(\Delta v, \Delta \omega)$  wird der Lösungsraum  $V$  diskretisiert. Es ergeben sich insgesamt  $n_V$  Geschwindigkeitskombinationen bzw. Lösungen im Lösungsraum:

$$n_V = \frac{V(v, \omega)}{\Delta v} \cdot \frac{V(v, \omega)}{\Delta \omega} \quad (5)$$

Abbildung 1 stellt alle möglichen Trajektorien mit der Geschwindigkeit  $V(v', \omega')$  des Lösungsraums  $V_L$  dar. Der Bewegungsbereich des Fahrzeuges innerhalb eines Prädiktionshorizontes wird durch die Trajektorien  $L_{1,2,3,4}$  als Grenzen eingeschränkt. Es sei angenommen, dass die

Geschwindigkeit  $V$  innerhalb des Prädiktionshorizontes konstant bleibt.



**Abbildung 1:** Prädiktion der lokalen Trajektorien innerhalb eines Prädiktionshorizontes  $T$ .

Zur Bemessung der Lösungsgüte und zum Festlegen der optimalen Trajektorie wird eine anwendungsspezifische Bewertungsfunktion entworfen, in der verschiedene Zielkriterien durch entsprechenden Kenngrößen quantifiziert und je nach Anwendungsfall unterschiedlich gewichtet werden, um die inkonsistenten Prioritäten der Zielkriterien bei der Auswahl der Trajektorie widerzuspiegeln.

Bei der lokalen Trajektorienplanung werden normalerweise die Zielkriterien Sicherheit, Zeit und Distanz, betrachtet. Zur Erfassung dieser werden die entsprechenden Kenngrößen, bestehend aus Orientierung  $\theta$ , Fahrgeschwindigkeit  $\bar{v}$  und Abstand zu Hindernissen  $l_h$ , herangezogen.  $g_\theta$ ,  $g_{\bar{v}}$  und  $g_{l_h}$  sind die jeweiligen Gewichtungsfaktoren dieser Kenngrößen. Über die Auslegung der Gewichtungsfaktoren können verschiedenen Fahrstrategien (z.B. energieoptimal, zeitoptimal, distanzoptimal) verfolgt werden, wobei die ausgewählte Strategie mit der globalen Trajektorienplanung konsistent sein muss; so besteht die Möglichkeit zur Individualisierung des Fahrbetriebs. Dieser Prozess wird sukzessive fortgesetzt, bis das Fahrzeug die Zielposition erreicht.

$$G(L) = g_\theta \cdot \theta(L) + g_{\bar{v}} \cdot \bar{v}(L) + g_{l_h} \cdot l_h(L) \quad (6)$$

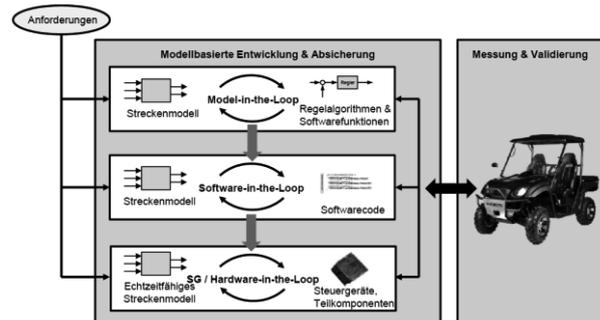
## 1.2 Modellbasierter Entwicklungs- und Absicherungsprozess

Die durchgängig modellbasierte und verifikationsorientierte Funktionsauslegung und -absicherung (vgl. Abb. 2) hat sich für die Entwicklung mechatronischer Systeme durch Optimierungen, Tests und Verifikationen in frühen Entwicklungsphasen als zeit- und kosteneffiziente Methode erwiesen, welche Model-in-the-Loop- (MiL-), Software-in-the-Loop- (SiL-) und Hardware-in-the-

Loop- (HiL-) Simulationen sowie die Echtzeitrealisierung durch Prototypen beinhaltet.

Ausgehend von Anforderungen und Lastenheft erfolgt die MiL-Simulation, in welcher Regelalgorithmen ohne Anspruch auf Echtzeitfähigkeit entwickelt und an einem System- bzw. Streckenmodell gegen ein komplexes Umgebungsmodell anhand von Szenarien erprobt werden. Aus den simulativ erprobten Regelalgorithmen wird im Rahmen der SiL-Simulation mittels automatischer Code-Generierung ein ausführbarer Target-Code erzeugt, welcher wiederum mit einem Systemmodell auf Funktion und Fehler getestet wird. Daraufhin erfolgt die HiL-Simulation, bei welcher ein um physische Teilkomponenten ergänztes, echtzeitfähiges Dynamikmodell zur Funktionsabsicherung und -optimierung verwendet wird.

Mithilfe dieser Methodik wird eine durchgängige Entwicklung basierend auf virtuellen Prototypen bis zum fertigen Produkt sichergestellt. Dieser Beitrag stellt aus der Methodik schwerpunktmäßig die Entwicklung und Verifikation der lokalen Trajektorienplanung mittels MiL-Simulation dar.



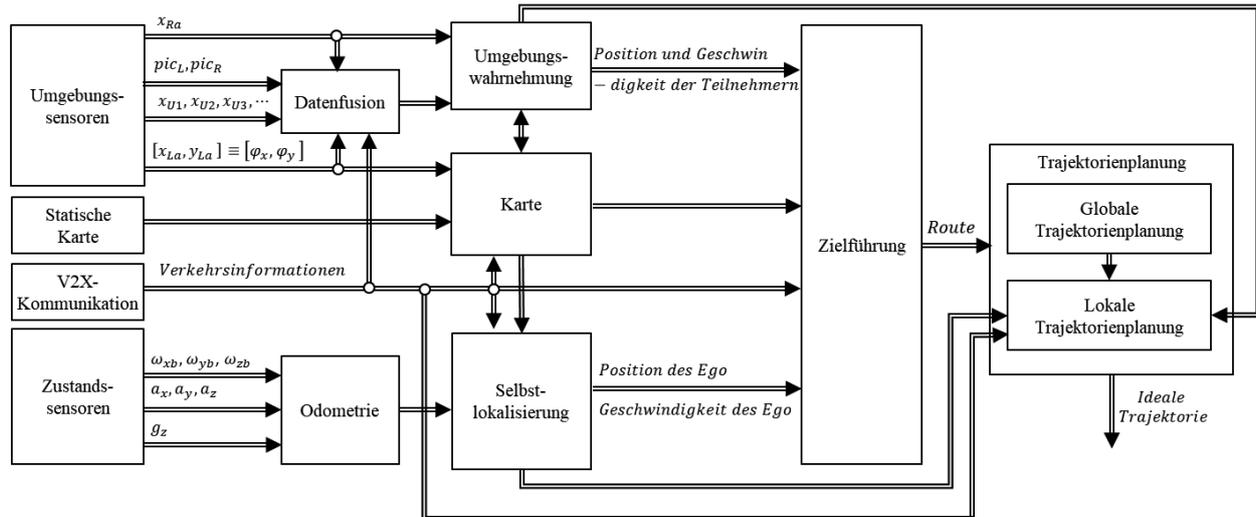
**Abbildung 2:** Durchgängig modellbasierte, verifikationsorientierte Entwicklungsmethodik nach [8].

## 2 Konzeption der adaptiven lokalen Trajektorienplanung

In diesem Abschnitt werden die Anforderungen an die lokale Trajektorienplanung erhoben und definiert. Auf Basis dessen werden die Funktionsstruktur und auch die informationstechnischen Schnittstellen zur Integration mit anderen Funktionsmodulen eines autonomen Fahrzeugs konzipiert.

### 2.1 Definition der Anforderungen

Anhand der mechatronischen Entwicklungsmethodik sind zum einen die Anforderungen zur Festlegung der Funktion und Struktur in Form von Lastenheften und



**Abbildung 3:** Funktionsstruktur eines autonomen Fahrzeugs zur Wahrnehmung, Planung und Entscheidungsfindung.

Spezifikationen erforderlich. Zum anderen dienen die Anforderungen als Vergleichsmaß bei der Funktionsabsicherung sowie Auswertung der Simulationsergebnisse [11, 12]. Ausgehend von der Analyse des Stands des Wissens und des angestrebten Anwendungsbereichs beim autonomen Fahren sowohl im offenen Straßenverkehr als auch in geschlossenen Produktionsumgebungen werden daher die folgenden wesentlichen Anforderungen an die adaptive lokale Trajektorienplanung definiert:

- Dynamische Anpassung an die sich ändernden Verkehrsumgebung
- Identifikation und Vermeidung potenzieller Konflikte
- Konsistente Fahrstrategie (z.B. energieoptimal, zeitoptimal, distanzoptimal) mit anderen Fahrfunktionen
- Einsatzbereitschaft für konfliktfreies autonomes Fahren in der Intralogistik und im Straßenverkehr
- Gewährleistung der Wiederverwendbarkeit, Parametrierbarkeit und Rekonfigurierbarkeit

## 2.2 Funktionsstruktur und Schnittstellendefinition

Anhand der Anforderungen werden die folgenden Teilfunktionen der lokalen Trajektorienplanung konzipiert. Deren Entwicklungsprozess wird im Abschnitt 3 weiter detailliert werden:

- **Identifizierung potenzieller Konflikte:** da sich die Fahrzeuge in einer sich ändernden Verkehrsumgebung befinden, sind vorübergehende Hindernisse, die nicht auf der Karte gespeichert wurden, wie z. B. plötzlich heruntergefallene Kisten oder sich plötzlich nähernde Verkehrsteilnehmer, nicht vorhersehbar. Daher muss

das Fahrzeug die Hindernisse mittels On-Board-Sensorsystemen oder V2X-Kommunikation erkennen und so die potenziellen Konflikte identifizieren.

- **Festlegung der lokalen Trajektorien zur Konfliktvermeidung:** Anhand der durch die Fahrstrategie ermittelten Zielkriterien wird eine anwendungsspezifische Bewertungsfunktion entworfen. Mithilfe der Bewertungsfunktion wird die am besten geeignete lokale Trajektorie aus dem Lösungsraum als spezifische Ausführungslösung für einen Abschnitt der globalen Trajektorie festgelegt.

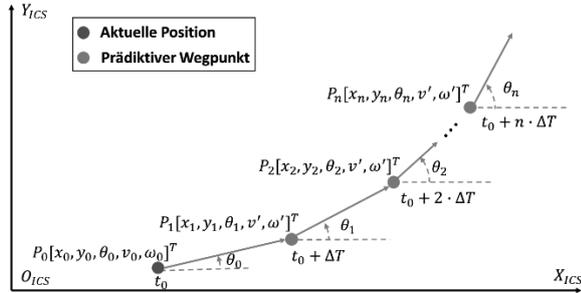
Zur Integration der lokalen Trajektorienplanung in die Funktionsstruktur eines autonomen Fahrzeugs werden die Schnittstellen zum Informationsaustausch, wie in Abbildung 3 dargestellt, anhand der aufgeführten Anforderungen definiert.

## 3 Entwicklung

Die im Abschnitt 2 konzipierten Funktionen sollen modellbasiert entwickelt werden. Deren Entwicklungsprozess wird im Folgenden detailliert beschrieben.

Bei der lokalen Trajektorienplanung werden auf Basis des aktuellen Fahrzeugzustands  $\underline{p}_0 [x_0, y_0, \theta_0, v_0, \omega_0]^T$  zum Zeitpunkt  $t = 0$  und dem Fahrzeugmodell alle möglichen Trajektorien  $\underline{L}$  mit einer zukünftigen Geschwindigkeit  $V(v', \omega')$  innerhalb eines Prädiktionshorizontes  $T$  innerhalb einer Abtastzeit  $\Delta T$  generiert. Hierbei wird die Trajektorie  $\underline{L}$  durch eine Reihe von Fahrzeugzuständen diskret dargestellt. Die folgende Abbildung 4 veranschaulicht den Vorgang.

$$\underline{\underline{L}}(t) = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \\ \theta_1 & \theta_2 & \dots & \theta_n \\ v' & v' & \dots & v' \\ \omega' & \omega' & \dots & \omega' \end{bmatrix} = \begin{bmatrix} \underline{P}_1 & \underline{P}_2 & \dots & \underline{P}_n \end{bmatrix} \quad (7)$$



**Abbildung 4:** Diskrete lokale Trajektorie mit Geschwindigkeit  $V(v', \omega')$  innerhalb eines Prädiktionshorizontes  $T$ .

Wie bereits erwähnt, wird hierbei zur Quantifizierung der Lösungsgüte der Trajektorien eine anwendungsspezifische Bewertungsfunktion (vgl. Gl. 6) verwendet. Obwohl durch Einstellung des Gewichtungsfaktors  $g_{l_h}$  des Zielkriteriums Sicherheit die meisten konflikthafte Trajektorien ausgefiltert werden können, ist es schwierig, durch die Bewertungsgleichung eine absolut konfliktfreie Trajektorie auszuwählen, da die Bewertungsgleichung darauf abzielt, einen Kompromiss der definierten Zielkriterien zu finden und somit eine systematische Optimierung zu erreichen. Zur Bewältigung dieser Schwachstelle werden die in der Bewertungsfunktion betrachteten Zielkriterien je nach Priorität in zwei Gruppen unterteilt. Zum einen gibt es harte Anforderungen (z.B. Sicherheit), welche nicht verhandelbar sind und auf jeden Fall erfüllt werden müssen. Die zweite Gruppe von Zielkriterien wird als weiche Anforderungen (z.B. Zeit, Distanz und Energie) bezeichnet, welche bei Ausführung der Fahrfunktionen zur Abwicklung der Transportaufträge im Interesse eines besseren Gesamtergebnisses hauptsächlich durch Abstimmung der Gewichtungsfaktoren kompromittiert werden. Sie werden nur dann thematisiert, wenn die harten Anforderungen bereits erfüllt sind. Zur klaren Unterscheidung der beiden Arten von Anforderungen wird die Bewertungsfunktion wie in Gl. 8 umformuliert, wobei  $S$  der vordefinierte Sicherheitsabstand ist, sodass konfliktbehaftete lokale Trajektorien direkt aus dem Lösungsraum herausgefiltert werden und damit das Zielkriterium Sicherheit als harte Anforderung immer erfüllt wird.

$$G(L) = g_{l_h} \cdot (g_\theta \cdot \theta(L) + g_{\bar{v}} \cdot \bar{v}(L)) \quad (8)$$

mit

$$g_{l_h} = \begin{cases} 0, & l_h(L) \leq S \\ 1, & l_h(L) > S \end{cases} \quad (9)$$

Da der Wertebereich der Kenngrößen der jeweiligen Zielkriterien unterschiedlich ist, müssen diese zum Vergleich miteinander mit einer geeigneten Skalierung normiert werden (beispielsweise die Normierung der Kenngröße Orientierung vgl. Gl. 10). Somit ergibt sich die normierte Bewertungsgleichung (vgl. Gl. 11):

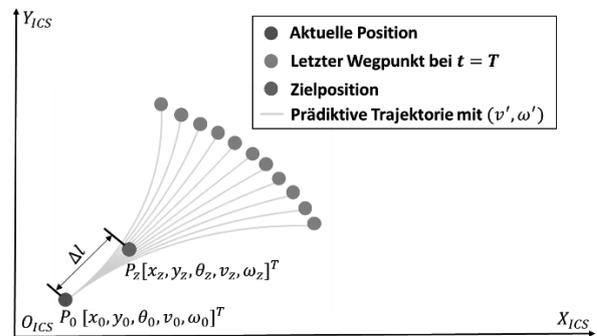
$$\theta_{nor}(L_i) = \frac{\theta_{max} - \theta(L_i)}{\theta_{max} - \theta_{min}} \quad (10)$$

$$\theta_{max} = \max [\theta(L_1), \theta(L_2), \dots, \theta(L_{n_{vw}})]$$

$$\theta_{min} = \min [\theta(L_1), \theta(L_2), \dots, \theta(L_{n_{vw}})]$$

$$G(L) = g_{l_h} \cdot (g_\theta \cdot \theta_{nor}(L) + g_{\bar{v}} \cdot \bar{v}_{nor}(L)) \quad (11)$$

Hierbei wird der Prädiktionshorizont  $T$  als eine konstante Zeiteinheit zur Beschreibung der zukünftigen Trajektorien verwendet, was dazu führt, dass das Fahrzeug mit der lokalen Trajektorie im letzten Trajektorienabschnitt an der Zielposition nicht genau anhalten kann und daher, wie in der Abbildung 5 veranschaulicht, eine unvermeidbare Abweichung  $\Delta l$  zwischen der Anfangsposition  $P_0$  des letzten Prädiktionshorizontes und der Zielposition  $P_z$  existiert.



**Abbildung 5:** Abweichung zwischen  $P_0$  und  $P_z$  mit einem konstanten Prädiktionshorizont  $T$  im letzten Trajektorienabschnitt.

Es sei angenommen, dass im letzten Schritt die Fahrzeugbewegung wie eine lineare Bewegung wegen des kleinen Zeitintervalls behandelt wird und die Abweichung  $\Delta l$  sich aus dem Abstand zwischen  $P_0$  und  $P_z$  ergibt (vgl. Gl. 12). Da die Fahrzeuggeschwindigkeit bis zum Erreichen der Zielposition auf null reduziert werden sollte, entspricht die höchste lineare Geschwindigkeit des Fahrzeugs im letzten Trajektorienabschnitt der linearen Geschwindigkeit am Anfang des Prädiktionshorizontes

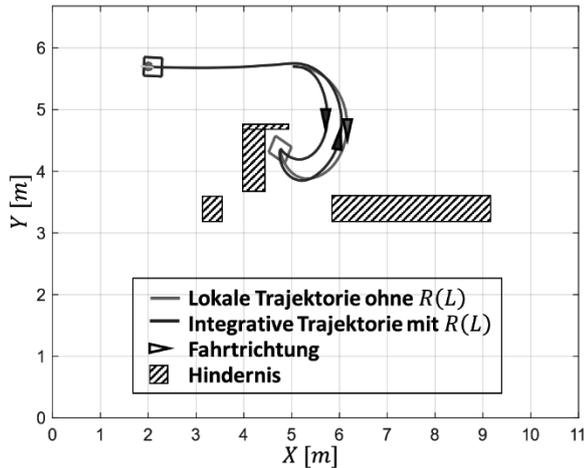
$v_0$ . Daher wird ein letztendlicher Prädiktionshorizont  $T'$  des letzten Trajektorienabschnittes ermittelt. Der Prädiktionshorizont  $T$  wird somit während der lokalen Trajektorienplanung abhängig von der Abweichung  $\Delta l$  umgeschaltet (Gl. 14).

$$\Delta l = v_0 \cdot T' = \sqrt{(x_0 - x_z)^2 - (y_0 - y_z)^2} \quad (12)$$

$$T' = \frac{\sqrt{(x_0 - x_z)^2 - (y_0 - y_z)^2}}{v'} \quad (13)$$

$$T = \begin{cases} T, & \Delta l > v_0 \cdot T \\ T', & \Delta l \leq v_0 \cdot T \end{cases} \quad (14)$$

In komplexen Szenarien mit zahlreichen Teilnehmern, insbesondere für Fahrzeuge in dicht besiedelten Wohngebieten oder freinavigierende FTF in einer kompakten Produktionsumgebung, neigen trotz lokaler Trajektorienplanung zur Konfliktvermeidung mit dynamischen Hindernissen oft dazu, in Sackgassen zu geraten. Obwohl es bei der Ermittlung der lokalen Trajektorie möglich ist, die Fahrtrichtung eines Fahrzeugs zu ändern, muss die Gewichtung der einzelnen Terme in der Bewertungsfunktion kompromittiert werden. In diesem Fall sollte die Orientierung  $\theta_{nor}$  relativ klein sein, was dem ursprünglichen Konzept widerspricht, dass zur Vermeidung von Umwegen die Orientierung  $\theta_{nor}$  so groß wie möglich sein sollte, um somit das Fahrzeug direkt in Richtung der Zielposition fahren zu lassen. Deswegen ist es schwierig für ein Fahrzeug sich selbständig durch die lokale Trajektorienplanung mit Gl. 11 aus einer Sackgasse zu befreien, sobald es in diese geraten ist (vgl. die rote Trajektorie in Abb. 6).



**Abbildung 6:** Schematische Darstellung eines Fahrzeugs in einer Sackgasse.

Zur Beseitigung des oben genannten Problems wird die Bewertungsfunktion um einen vierten Teil  $R$  zur

Adaption der Fahrtrichtung ausschließlich bei Sackgassen erweitert. Es ist auffällig, dass  $\theta_{nor}$  und  $R$  sich gegenseitig ausschließen sollten, was durch Konfiguration der entsprechenden Gewichtungsfaktoren bzw.  $g_\theta$  und  $g_\omega$  realisiert wird. Falls es keine gangbaren Trajektorien zur Auflösung der Konflikte gibt, d.h. die lineare Geschwindigkeit  $\bar{v}$  der potenziellen Trajektorien ein Lösungsraum gleich null ist, muss sich das Fahrzeug zur Änderung der Fahrtrichtung um die eigene Hochachse drehen und der Gewichtungsfaktor  $g_\omega$  einen geeigneten Wert  $C_1$  annehmen. Sobald eine praktikable Trajektorie zur Ausfahrt der Sackgasse vorliegt, soll im Gegensatz dazu der Gewichtungsfaktor  $g_\omega$  wieder zu null angenommen werden. Die lokale Trajektorienplanung mit und ohne Erweiterung von  $R$  wird in der Abbildung 6 gegenübergestellt und untenstehend veranschaulicht.

$$G(L) = g_{l_h} \cdot (g_\theta \cdot \theta_{nor}(L) + g_v \cdot \bar{v}_{nor}(L) + g_\omega \cdot R(L)) \quad (15)$$

mit

$$R(L) = |\omega'| \quad (16)$$

$$\begin{cases} g_\theta = 0, g_\omega = C_1, & \forall [\bar{v}(L_1), \bar{v}(L_2), \dots, \bar{v}(L_{n_{vw}})] = 0 \\ g_\theta = C_2, g_\omega = 0, & \exists [\bar{v}(L_1), \bar{v}(L_2), \dots, \bar{v}(L_{n_{vw}})] \neq 0 \end{cases} \quad (17)$$

Hierbei ist zu beachten, falls die lineare Geschwindigkeit einen zu großen Wert annimmt, biegt das Fahrzeug mit einem großen Kreis zum Ausweichen der Hindernisse ab (vgl. Abb. 7). Dieses Dilemma kann einerseits überwunden werden, durch Änderung des Gewichtungsfaktors  $g_v$  und andererseits indem das Vorzeichen (positive oder negative Richtung) des Gewichtungsfaktors in Abhängigkeit des Zustands des Fahrzeugs dynamisch angepasst wird. Unter Berücksichtigung des höheren Zeitaufwands und höheren benötigten Rechenleistung zur dynamischen Anpassung des Gewichtungsfaktors durch Online Training wird hier der zweite Ansatz zur dynamischen Anpassung des Vorzeichens der Gewichtungsfaktoren in die Bewertungsgleichung eingeführt, wodurch der übermäßige Einfluss eines bestimmten Faktors auf die Bewertungsfunktion unter speziellen Umständen unterdrückt und eine übermäßige Lenkhäufigkeit des Fahrzeugs reduziert wird.

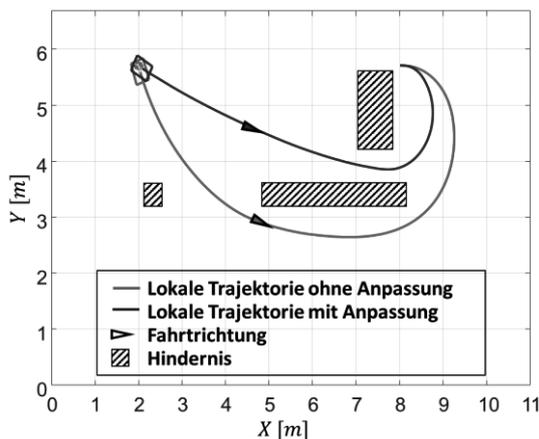
Die auslösende Bedingung zur Umschaltung des Vorzeichens von  $g_v$  wird durch die Randbedingungen (vgl. Gl. 20), die sich aus den Orientierungswinkeln  $\Delta\theta[L_i(P_{j+1}), L_i(P_j)]$  und der linearen Geschwindigkeit  $v'$  zusammensetzen, deklariert. Hierbei wird  $v_c$  als

Schwellenwert betrachtet und experimentell oder durch den Optimierungsprozess bestimmt. Sobald die lineare Geschwindigkeit  $v'$  der Trajektorie  $L_i$  den Schwellenwert  $v_c$  überschreitet und gleichzeitig die Trajektorie von der vordefinierten Zielposition weg orientiert ist, wird das Vorzeichen von dem zweiten Zielkriterium  $\bar{v}_{nor}(L_i)$  negativ, was zu einer Verringerung der Wahrscheinlichkeit diese Trajektorie  $L_i$  zu wählen führt. Abbildung 7 stellt den Einfluss des Vorzeichenwechsels des zweiten Zielkriteriums auf die lokale Trajektorienplanung dar. Die Trajektorien mit zu großer linearer Geschwindigkeit, die sich in der Richtung weg von der vordefinierten Zielposition orientieren, werden so herausgefiltert.

$$\begin{aligned} & \Delta\theta[L_i(P_{j+1}), L_i(P_j)] \\ &= \arctan\left(\frac{y_z - y_{P_{j+1}}}{x_z - x_{P_{j+1}}}\right) - \arctan\left(\frac{y_z - y_{P_j}}{x_z - x_{P_j}}\right) \end{aligned} \quad (18)$$

$$Q(L) = g_{L_h} \cdot [g_h \cdot \theta_{nor}(L) + g_v \cdot \bar{v}_{nor}(L) + g_\omega \cdot R(L)] \quad (19)$$

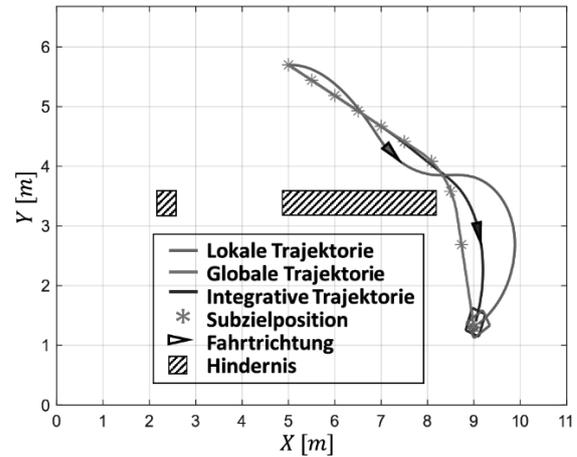
$$\begin{cases} g_v, & \neg(v' > v_c \cap \Delta\theta[L_i(P_{j+1}), L_i(P_j)] < 0) \\ -g_v, & v' > v_c \cap \Delta\theta[L_i(P_{j+1}), L_i(P_j)] < 0 \end{cases} \quad (20)$$



**Abbildung 7:** Schematische Darstellung des Einflusses vom Vorzeichenwechsel des zweiten Terms auf die lokale Trajektorienplanung.

Um sicherzustellen, dass die in der globalen Trajektorienplanung eingesetzte Fahrstrategie in der lokalen Trajektorienplanung konsistent beibehalten und nicht abgeschwächt wird, um den Rechenaufwand zu reduzieren, und um sicherzustellen, dass die bereits in der Karte gespeicherten stationären Hindernisse in der lokalen Trajektorienplanung nicht doppelt gerechnet werden, (d.h. sie wurden bereits in der globalen Trajektorienplanung berücksichtigt), sollen die globale und lokale Planung synergetisch miteinander integrieren werden, indem die diskreten Wegpunkte der globaler Trajektorie als

Subziele der lokalen Trajektorienplanung verwendet werden. Es sei dabei vorausgesetzt, dass die Länge der in der Abbildung 8 dargestellten integrativen Trajektorie viel kleiner ist als die der vollständigen lokalen Trajektorie. Somit ist auch die Transportzeit der integrativen Trajektorie vom Start- bis zur Zielposition geringer.



**Abbildung 8:** Gegenüberstellung der globalen, lokalen und integrativen Trajektorienplanung.

## 4 Funktionsabsicherung

### 4.1 Szenariobeschreibung

Mit Hilfe des virtuellen Prüfstandes werden die entwickelte Funktion und deren Synergien mit anderen Fahr-funktionen in einer realistischen Simulationsumgebung verifiziert und optimiert. Abbildung 9 stellt das Konzept des cyber-physisches Industrie-4.0-Labortestfeldes, welcher zur realitätsnahen Entwicklung und vor allem Erprobung der FTF, der IoT-basierten Kommunikation in Produktionsanlagen sowie der selbstoptimierten Intralogistik und Produktionsplanung konzipiert und entwickelt wird, im virtuellen Prüfstand dar, wobei eine Produktionsanlage mit drei Produktionsmaschinen, zwei Lagern, einem Warenausgang und vier autonomen FTF simuliert wird. Bei der Funktionsabsicherung werden 120 Transportaufträgen (TA) zufällig generiert und die FTF sollen selbst eine Trajektorie zwischen der Start- und Zielposition eines TA berechnen und konfliktfrei die Zielposition in einer dynamischen Verkehrsumgebung erreichen.

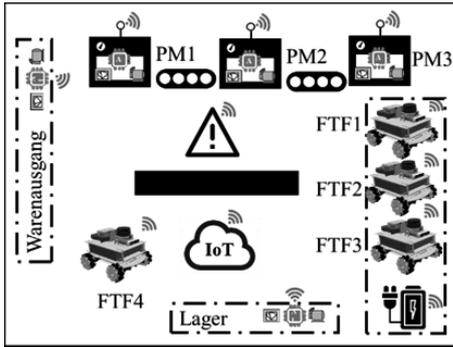


Abbildung 9: Cyber-physisches Industrie-4.0-Labortestfeld nach [13].

## 4.2 Simulationsergebnisse

Abbildung 10 stellt einen Ausschnitt aus dem Prozess der TA-Abwicklung mit vier FTF dar. Es ist ersichtlich, dass zur Vermeidung der Konflikte zwischen den einzelnen FTF die ausgeführten Trajektorien mithilfe der lokalen Trajektorienplanung während des Transports je nach Konfliktart in unterschiedlichem Maße von den entsprechenden globalen Trajektorien abweichen. Beispielsweise fahren FTF3 und FTF4 in entgegengesetzte Richtung und die globalen Trajektorien überschneiden sich teilweise. Um Konflikte zu vermeiden, kann FTF4 mit höherer TA-Priorität durch Abstimmung der FTF untereinander weiterhin der ursprünglich geplanten Trajektorie folgen, aber aufgrund einer drohenden Kollision mit FTF2 muss FTF4 von der globalen Trajektorie abweichen. Anhand dieser Simulationsergebnisse werden alle TA konfliktfrei ausgeführt.

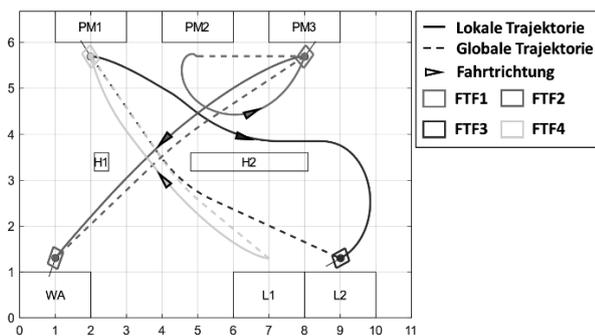


Abbildung 10: Ein Ausschnitt aus dem Prozess der TA-Ausführung durch FTF.

## 5 Zusammenfassung und Ausblick

Im vorliegenden Beitrag wird die adaptive lokale Trajek-

torienplanung mittels DWA zur Konfliktvermeidung insbesondere mit den dynamischen Akteuren konzipiert und entwickelt, sodass die Sicherheit der Fahrzeuge in der sich ändernden Verkehrsumgebung stetig gewährleistet wird. Durch Festlegung der Gewichtungsfaktoren ermöglicht diese darüber hinaus, konsistent bei der verwendeten Fahrstrategie und der Konfliktlösung zu bleiben. Mithilfe des Testszenarios wird die Funktionalität der adaptiven lokalen Trajektorienplanung in der Simulationsumgebung validiert und demonstriert.

Es sei bemerkt, dass Hauptprobleme bei der Entwicklung der lokalen Trajektorienplanung mittels DWA der Entwurf der anwendungsspezifischen Bewertungsfunktion mit tauglichen Zielkriterien und die Festlegung der Gewichtungsfaktoren, welche die Leistungsfähigkeit und Effizienz der lokalen Trajektorienplanung weitgehend bestimmen, sind. Im nächsten Schritt werden zur Verbesserung deren Leistungsfähigkeit die Gewichtungsfaktoren durch KI-Methoden systematisch optimiert werden.

## Danksagung

Gefördert vom Niedersächsischen Ministerium für Wissenschaft und Kultur unter Fördernummer ZN3495 im Niedersächsischen Vorab der VolkswagenStiftung und betreut vom Zentrum für digitale Innovationen Niedersachsen (ZDIN).



## References

- [1] Zhang J, Liu-Henke X. *Konfliktfreie, selbstoptimierte Trajektorienplanung für ein fahrerloses Transportfahrzeug zur Durchführung des autonomen Gütertransportes im Produktionsumfeld*. 25. Symposium Simulationstechnik - ASIM 2020, virtuelle Tagung, October 14 - 15 2020.
- [2] Göllner M, Jacobitz S, Li T, Liu-Henke X. *Modular Platform for Route Guidance in the Cyber-Physical Laboratory Test Field*. ASIM STS/GMMS & EDU Workshop 2022, July 25 - 26, Wien Österreich, 2022.
- [3] Li X, Hu X, Wang Z, Du Z. *Path Planning Based on Combination of Improved A-STAR Algorithm and DWA Algorithm*. 2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM), Manchester, United Kingdom, 2020, pp. 99-103, doi: 10.1109/AIAM50918.2020.00025.
- [4] Liang CL, Zhang XK, Han X. *Route planning and track keeping control for ships based on the leader-vertex ant*

- colony and nonlinear feedback algorithms*. Applied Ocean Research, 2020, Vol. 101, doi: 10.1016/j.apor.2020.102239.
- [5] Scherler S. *Zeit- und energieoptimierter Fahrbetrieb eines Elektrofahrzeugs mit Brennstoffzellen-Range-Extender im digital vernetzten Verkehr* [Dissertation]. Technische Universität Braunschweig, 2022.
- [6] Cai JC, Wan MF, Huang ZL, Liu Y. *An Improved DWA Path Planning Algorithm Integrating Global JPS Strategy*. 2022 2nd International Conference on Computer, Control and Robotics (ICCCR), Shanghai, China, 2022, pp. 20-26, doi: 10.1109/ICCCR54399.2022.9790216.
- [7] Xiong B, Jiang HK, Cui JJ, Lu K, Chen PY, Zhang M. *UAV Path Planning Based on Improved A\* and DWA Algorithms*. International Journal of Aerospace Engineering. 2021, Vol. 2021, doi: 10.1155/2021/4511252.
- [8] Fox D, Burgard W, Thrun S. *The Dynamic Window Approach to Collision Avoidance*. IEEE Robotics & Automation Magazine, 2002, 4(1):23-33.
- [9] 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 (ICSOFT), Online, July 6-8, 2021.
- [10] Zhang J, Liu-Henke X. *Virtueller Prüfstand zur simulationsgestützten Bewertung von cyber-physischen Industrie-4.0-Lösungen*. Virtueller ASIM STS/GMMS & EDU Workshop 2021, March 11 - 12, 2021.
- [11] 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.
- [12] 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.
- [13] Jacobitz S, Göllner M, Zhang J; Yarom OA, Liu-Henke X. *Seamless Validation of Cyber-Physical Systems Under Real-Time Conditions by Using a Cyber-Physical Laboratory Test Field*. IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE), Online, December 12-14, 2021.



# Gekoppelte Simulation des Einspritz- und Verbrennungsvorgangs eines Industrie-Dieselmotors

Daniel Jörss<sup>1</sup>, Maximilian Ringel<sup>1</sup>, Bert Buchholz<sup>2</sup>, Christian Fink<sup>1\*</sup>

<sup>1</sup>Department of Mechanical / Process and Environmental Engineering, Wismar University of Applied Sciences, Philipp-Müller-Str. 14, 23966 Wismar, Germany

\**Christian.Fink@hs-wismar.de*

<sup>2</sup>Chair of Piston Machines and Internal Combustion Engines, Rostock University, Albert-Einstein-Straße 2, 18059 Rostock, Germany

**Abstract.** Um die zukünftigen Abgasemissionsgrenzen einhalten zu können, kommt einer sehr präzisen Regelung von Einspritzmenge und Einspritzbeginn eine entscheidende Bedeutung zu. Das Einspritzsystem, welches den Forderungen bestmöglich gerecht wird, ist das Common-Rail System. Das hier präsentierte Modell beschreibt ein 1D-Hydraulik Injektormodell in Simscape™, welches an ein nulldimensionales Verbrennungsmotormodell ankoppelt. Zur Validierung des Injektormodells wurden Messungen an einem Einspritzverlaufsindikator durchgeführt. Dabei konnten gute Ergebnisse zwischen Simulation und Experiment erzielt werden. Im zweiten Schritt wurde ein in Matlab® Simulink® entwickeltes Modell [1] mit einem phänomenologischen Verbrennungsmodell nach Barba [2] erweitert. Als Eingangparameter wird der Einspritzverlauf in das Verbrennungsmodell übergeben. Unter Berücksichtigung weiterer Teilmodelle wird daraus der Brennverlauf für das Verbrennungsmotormodell berechnet. Zwischen dem aus der Druckverlaufsanalyse (DVA) berechneten Brennverläufen und der Simulation konnten gute Ergebnisse erzielt werden. Des Weiteren zeigen die experimentellen und simulierten Zylinderdruckverläufe hohe Übereinstimmungen.

## Einleitung

Mit der Einführung der Common-Rail Einspritztechnik ist es möglich, die Druckerzeugung und die Einspritzung zeitlich sowie örtlich zu entkoppeln. Der notwendige Einspritzdruck in dem System wird durch eine separate Hochdruckpumpe erzeugt. Der unter Druck stehende Kraftstoff wird anschließend in

einem Hochdruckrail gespeichert und ist somit das Bindeglied zwischen Pumpe und Injektor. Anstelle von druckgesteuerten Einspritzdüsen werden in Common-Rail-Systemen elektrisch betätigte Injektoren verwendet. Durch die elektrische Ansteuerung des Injektors können der Zeitpunkt und die Dauer der Einspritzung bestimmt werden. Auf diese Weise ergeben sich große Freiheiten hinsichtlich der Gestaltung von Mehrfacheinspritzungen sowie der Applikation unterschiedlicher Kraftstoffdrücke je nach Lastpunkt. Dadurch ist es möglich, Einspritzstrategien so zu entwickeln, dass alleine mit innermotorischen Maßnahmen erhebliche Emissionsreduktionen erreicht werden können. Auf diese Weise kann ggf. auf eine anschließende Abgasnachbehandlung verzichtet oder die notwendige Nachbehandlung deutlich reduziert werden. Dabei stellen sich der zeitliche Verlauf der Einspritzrate sowie die Zerstäubungsqualität der in den Brennraum eingebrachten Kraftstoffmenge als wichtige Parameter heraus. Für eine bestmögliche Anpassung jedes Betriebspunktes im Motorenkennfeld sind oft umfangreiche Prüfstandsversuche notwendig. Mit heutigen Simulationsverfahren gelingt es, komplexe Vorgänge im Verbrennungsmotor detailliert abzubilden, um ein besseres Verständnis des Betriebs- und Emissionsverhaltens zu bekommen sowie die Auswirkungen unterschiedlicher Applikationen zu analysieren.

In diesem Bericht soll das Modell eines Common-Rail Injektors mit Anbindung an ein nulldimensionales Verbrennungsmotormodell [1] beschrieben, sowie erlangte Simulationsergebnisse mit Messergebnissen des Prüfstandes verglichen werden.

# 1 Modellbeschreibung

Zur Simulation des Common-Rail Injektors wurde in Simulink® die physikalische objektorientierte Toolbox Simscape™ gewählt. Konventionelle Common-Rail Injektoren bestehen aus drei wesentlichen Baugruppen, das Steuerventil, die Einspritzdüse sowie ein mechanisch-hydraulisches Kopplungselement. Bezogen auf Abbildung 1 entscheidet der Druck im Ventilsteuerraum (2) darüber, ob die Einspritzdüse öffnet oder schließt bzw. geöffnet oder geschlossen bleibt. Wie hoch der Druck im Ventilsteuerraum ist, hängt zum einen vom Raildruck und zum anderen vom Zustand des Steuerventils ab. Ist das Steuerventil geöffnet, strömt Kraftstoff über eine Abströmdrossel aus dem Ventilsteuerraum ab und der Druck fällt. Schließt das Steuerventil die Abströmdrossel, steigt der Druck im Ventilsteuerraum an und erhöht über den Ventilsteuerkolben die auf die Düsenadel wirkende Kraft bis diese schließt und den Einspritzvorgang beendet. Ein fallender Druck reduziert die auf den Steuerkolben wirkende Kraft auf die Düsenadel, welche einem Kräftegleichgewicht unterliegt. Ab Unterschreitung eines bestimmten Druckes öffnet die Düsenadel und der Einspritzvorgang beginnt.

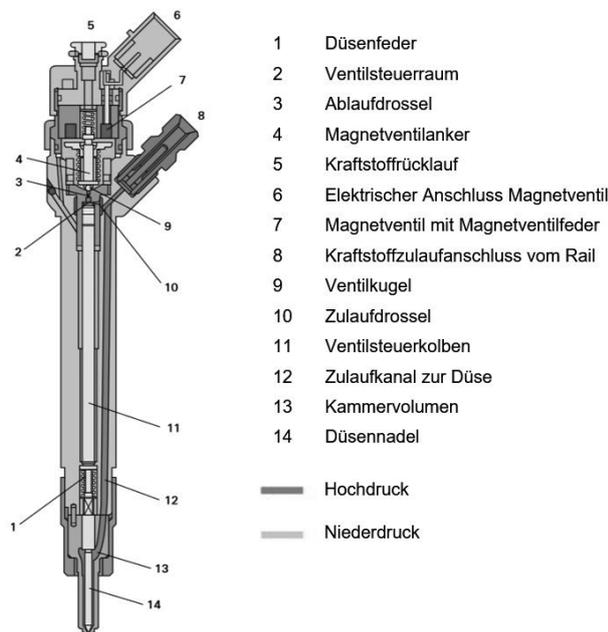


Abbildung 1: Aufbau Magnetventil-Injektor [3]

In der Simulationsumgebung wird der Injektor in Teilsysteme unterteilt, bei denen verschiedene Domä-

nen von Simscape™ verwendet werden. Die geometrischen Abmaße der einzelnen Bauteile sowie die Volumina des zu simulierenden Injektors wurden im Vorfeld messtechnisch ermittelt. Parameter für das Magnetventil konnten aus [4],[5] entnommen werden.

## 1.1 Magnetventilmodell

Das Magnetventilmodell umfasst insgesamt vier Domänen aus der Simscape™-Bibliothek. In Abbildung 2 ist die Verknüpfung der einzelnen Blöcke dargestellt. Der

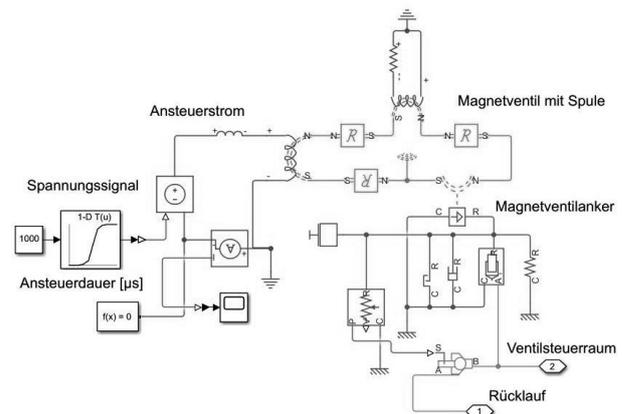


Abbildung 2: Aufbau Magnetventil in Simscape™

Aufbau dieses Teilmodells wurde in Anlehnung an [4] weiterentwickelt. Wie auch in [4] wird hier als Eingangsgröße ein pulswidenmoduliertes Spannungssignal vorgegeben. Unter Berücksichtigung des ohmschen Widerstandes sowie der Induktivität ergibt sich hieraus der Ansteuerstrom. Aus dem zeitlichen Verlauf dieses Signals ergibt sich die Ansteuerdauer für das Magnetventil. Dieser Zusammenhang ist exemplarisch, bei einer Ansteuerdauer von 1000 µs, in Abbildung 3 dargestellt. Über den Strom entsteht in der Spule des Magnetventils ein Magnetfeld, das den Magnetventilanker anzieht. Überwiegt die Magnetkraft die der entgegen gerichteten Federkraft sowie die der Druckkraft, die auf der Ventilkugel wirkt, beginnt sich die Ventilkugel aus dem Sitz zu heben. In Abbildung 4 ist der zeitliche Verlauf zwischen Ansteuerstrom und der Hubbewegung des Magnetventilankers dargestellt. Der negative Anteil des Hubverlaufs lässt sich auf eine Stauchung und Dehnung der Bauteile zurückführen, die durch den hohen Systemdruck hervorgerufen wird. Der hohe Strom in der Anzugsphase bewirkt ein schnelles Öffnen, womit der Durchfluss durch die Abflussdrossel freigegeben wird. Nach kurzer Zeit beginnt die Hal-

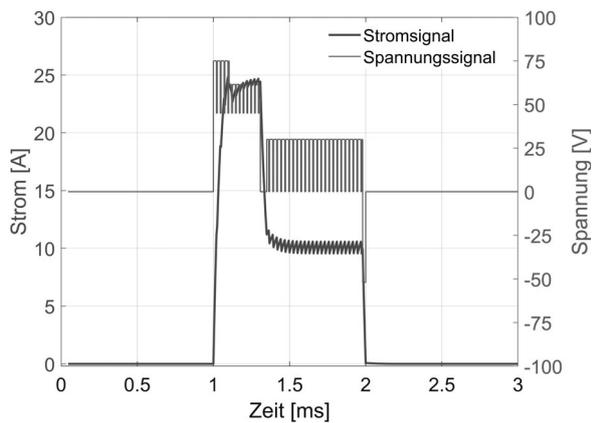


Abbildung 3: simulierter Strom- und Spannungsverlauf

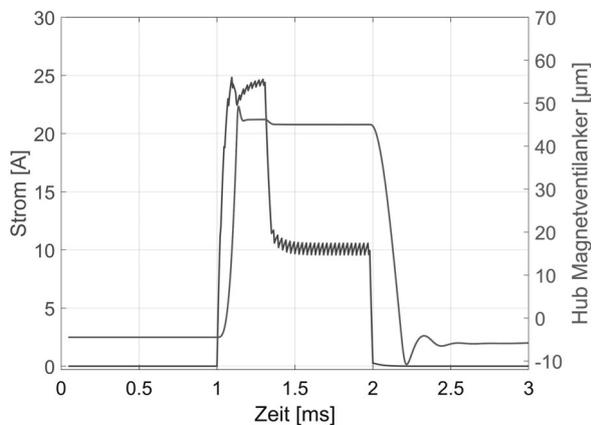


Abbildung 4: Zusammenhang Steuerstrom und Magnetventilhubverlauf

tephase, in der die Ventilkugel weiterhin geöffnet bleibt. Mit dem Ende der Ansteuerung drückt die Magnetventilfeder den Magnetanker nach unten, schließt den Ventilsitz und beendet somit den Durchfluss durch die Abflussdrossel.

## 1.2 Mechanisch-hydraulisches Modell

Dieses Modell beschreibt die Kopplung zwischen den hydraulischen und den mechanischen Komponenten im Injektor. Als Eingangsgrößen dienen die Hubbewegung des Magnetventilankers aus dem Magnetventilmodell und ein vorgegebener Kraftstoffdruck. In Abbildung 5 ist dieser Ansatz in Simscape dargestellt und ist dem Aufbau des vermessenden Injektors nachempfunden. Die beiden Hydraulikzylinderblöcke repräsentieren darin die Düsennadel und den Ventilsteuerkol-

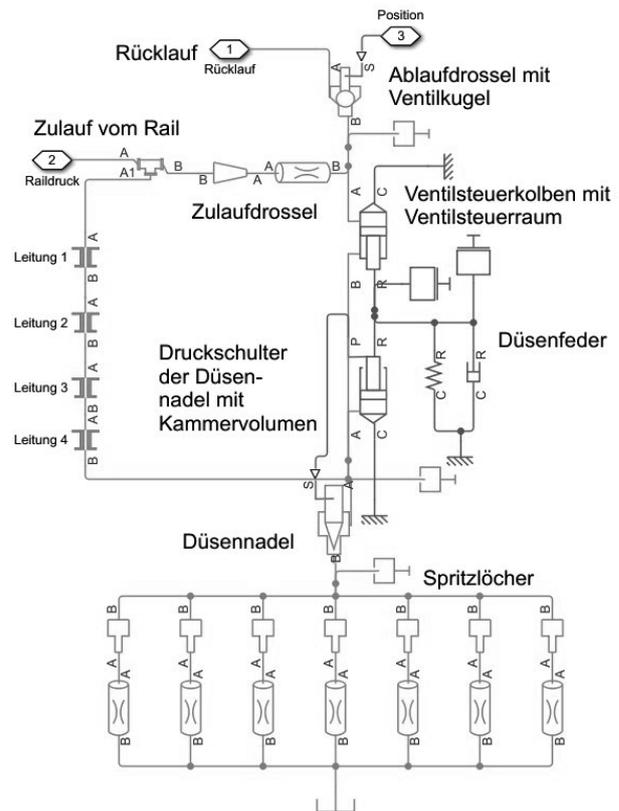
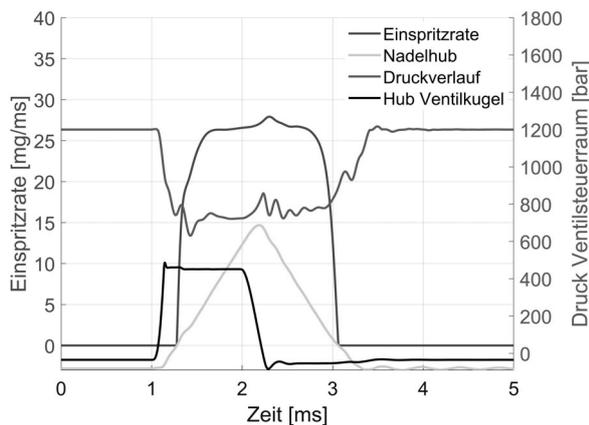


Abbildung 5: Aufbau des Injektormodells in Simscape™

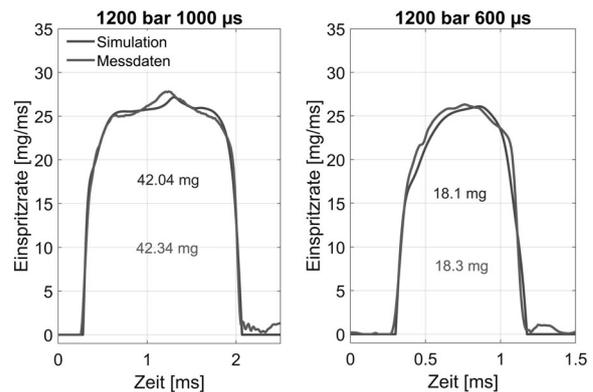
ben, mit ihren dazugehörigen Massen. Wie in [4] werden auch hier die Spritzlöcher aus einer Kombination aus Querschnittsänderung und Drosselstelle einzeln dargestellt. Im Ruhezustand liegen im Ventilsteuererraum sowie im Kammervolumen derselbe Druck an. Auf der Fläche des Ventilsteuerkolbens und der Druckschulter der Düsennadel wirkt jeweils eine Kraft. Wegen der Flächenverhältnisse wird die Düsennadel durch die größere Kraft des Ventilsteuerkolbens in ihren Sitz gedrückt. Durch die Ansteuerung des Magnetventils hebt sich die Ventilkugel aus dem Sitz und öffnet daraufhin die Ablaufdrossel. In der Folge strömt der Kraftstoff aus dem Ventilsteuererraum, wodurch der Druck sinkt. Daraus folgt, dass der Druck im Ventilsteuererraum kleiner ist als der Druck im Kammervolumen. Die Druckdifferenz bewirkt, dass die Kraft an der Düsennadel größer ist als die im Ventilsteuererraum und führt zum Öffnen der Düsennadel. Vom Querschnittsverhältnis zwischen Zu- und Ablaufdrossel des Ventilsteuererraums hängt die Öffnungsgeschwindigkeit, mit der sich die Düsennadel bewegt, ab. Der Zusammenhang zwischen Einspritzrate, Nadelhubverlauf, Druckverlauf


**Abbildung 6:** Simulationsergebnisse Injektormodell

im Steerraum und Hubbewegung der Ventilkugel sind in Abbildung 6 dargestellt. Mit dem Ende der Ansteuerung des Magnetventils verschließt die Ventilkugel die Ablaufdrossel und stoppt den Durchfluss. Dadurch baut sich der Druck im Ventilsteerraum wieder auf, womit wieder eine erhöhte Kraft auf den Steuerkolben entsteht. Die Schließgeschwindigkeit wird maßgeblich durch die Zulaufdrossel bestimmt. Überschreiten die Kraft aus dem Ventilsteerraum und die Kraft der Düsenfeder nun die entgegengesetzte Kraft auf der Düsennadel, so beginnt sich die Düsennadel zu schließen. Die Einspritzung endet, wenn die Düsennadel den Düsenkörpersitz erreicht und die Spritzlöcher verschließt.

## 2 Modellvalidierung

Zur Validierung des Simulationmodells werden die Berechnungsergebnisse mit Messdaten des Einspritzverlaufsindicators verglichen. In Abbildung 7 ist ein Vergleich der Einspritzverläufe, am Beispiel von zwei verschiedenen Betriebspunkten, dargestellt. Der Messdatenverlauf ist in beiden Fällen der mittlere Verlauf aus 150 aufeinanderfolgenden Einspritzvorgängen, die am Einspritzverlaufsindikator gemessen wurden. Ein Vergleich der Einspritzverläufe zeigt, dass Simulation und Messdaten qualitativ gut übereinstimmen. Die Abweichung der eingespritzten Mengen zwischen Simulation und Experiment beträgt rund 1%. Ähnlich gute Ergebnisse wurden auch an anderen Betriebspunkten festgestellt, was die Anwendbarkeit dieses Injektormodells bestätigt.


**Abbildung 7:** Vergleich der Einspritzverläufe zwischen Messung und Simulation

## 3 Verbrennungsmodell

Der hier vorgestellte Ansatz nach Barba [2] beschreibt ein globales phänomenologisches Verbrennungsmodell zur Berechnung des Brennverlaufes. Dieses Modell verwendet als Eingangsgröße den Einspritzverlauf, wodurch sich gegenüber empirischen Modellen mehr Anwendungsmöglichkeiten von Parameterstudien ergeben.

Wie in [6] ist auch hier der Grundgedanke, dass die Kraftstoffumsetzung durch den Zündverzug und die Kraftstoffverdampfung gesteuert wird. Als erster Modellbaustein, für die Verdampfung des eingebrachten Kraftstoffes, folgt die Berechnung des Primärtropfendurchmessers  $d_{Tr0}$  [7] nach Gleichung 1.

$$d_{Tr0} = c \cdot d_{D,eff} \cdot (Re \cdot We)^{-0.28} \quad (1)$$

Die Reynolds- und die Weber-Zahl werden nach Gleichung 2 und 3 berechnet.

$$Re = \frac{u_{Tr0} \cdot d_{D,eff}}{v_{B,fl}} \quad (2)$$

$$We = \frac{u_{Tr0}^2 \cdot d_{D,eff} \cdot \rho_Z}{\sigma_{B,fl}} \quad (3)$$

Die Ausgangstropfengeschwindigkeit  $u_{Tr0}$  wird direkt aus dem Einspritzverlauf nach Gleichung 4 berechnet.

$$u_{Tr0} = \frac{dm_E}{dt} \cdot \frac{1}{\rho_{B,fl} \cdot A_{D,eff}} \quad (4)$$

Mit dem  $d^2$ -Gesetz in Gleichung 5 lässt sich ausgehend vom Ausgangstropfendurchmesser und einem empirischen Faktor  $\beta$  die Verdampfungszeit für jeden Rechenschritt berechnen.

$$d_{Tr}^2 = d_{Tr0}^2 - \beta \cdot t \quad (5)$$

In der Literatur werden für den Koeffizient  $c$  Werte von 8.7 [7], 4.0 [2] und 25.0 [8] angegeben. Für den Verdampfungskoeffizient  $\beta$  wird der Wert mit  $7 \cdot 10^{-6}$  angegeben [2][10]. Neben der Verdampfungszeit muss noch zusätzlich der Zündverzug mitberücksichtigt werden. Der Zündverzug ist typischerweise die Zeit zwischen Einspritzbeginn und der ersten signifikanten Wärmefreisetzung. Nach Gleichung 6 unterteilt Barba [2] diesen in einen physikalischen und einen chemischen Anteil.

$$\tau_{ZV} = \tau_{ZV,phy} + \tau_{ZV,chem} \quad (6)$$

Die Anteile lassen sich nach den Gleichungen 7 und 8 berechnen.

$$\tau_{ZV,phy} = c_0 \cdot u_{Tr0}^{-1.68} \cdot d_{D,eff}^{0.88} \quad (7)$$

$$\tau_{ZV,chem} = c_1 \cdot \left(\frac{p_z}{p_0}\right)^{c_2} \cdot \lambda_{Zn}^{c_3} \cdot e^{\frac{T_A}{T_Z}} \quad (8)$$

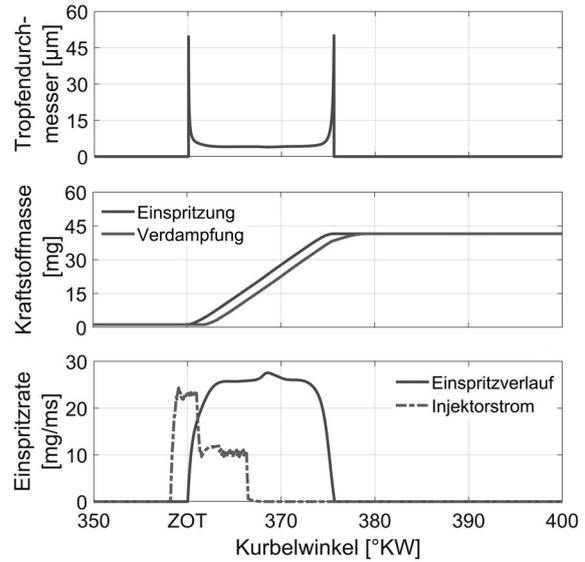
Da sich die Temperaturen und Drücke während der Zündverzugsberechnung ändern, wird deshalb das Zündintegral nach Gleichung 9 verwendet.

$$1 = \int_{t_{EB}}^{t_{VB}} \frac{1}{\tau_{ZV}} dt \quad (9)$$

Das Zündereignis erfolgt, wenn das Integral den Wert 1 erreicht hat. In Abbildung 8 sind die Simulationsergebnisse des Verdampfungsmodells dargestellt. Aus dem Einspritzverlauf wird für jede Rechenschrittweite ein Tropfendurchmesser sowie die dazugehörige Verdampfungszeit berechnet.

In Simulink<sup>®</sup> werden die Verdampfungszeit und der Zündverzug als Zeitverzögerung an ein "Transport Delay" übergeben. Damit lässt sich die kumulierte eingespritzte Kraftstoffmasse als Eingangsgröße, um diese Zeitspanne verschieben.

Den wichtigsten Parameter für das Verbrennungsmodell stellt die zur Verfügung stehende unverbrannte dampfförmige Kraftstoffmasse dar. Die Beschreibung der Diffusionsverbrennung nach Gleichung 10 basiert auf einen Frequenzansatz, der neben



**Abbildung 8:** Simulationsergebnisse Verdampfungsmodell

einer charakteristischen Mischungslänge auch eine Mischungsgeschwindigkeit, die für die Turbulenz im Brennraum steht, verwendet [2].

$$\frac{dm_{B,V}}{dt} = \frac{u'}{l_{Diff}} \cdot m_{B,UV} \quad (10)$$

Nach Gleichung 11 setzt sich die Mischungslänge  $l_{Diff}$  aus der Brennraumgeometrie, dem momentanen Verbrennungsluftverhältnis und aus der Anzahl der Düsenstritzlöcher zusammen.

$$l_{Diff} = \sqrt[3]{\frac{V_Z}{\lambda \cdot Anz_D}} \quad (11)$$

Auf Grund der auftretenden Turbulenz im Brennraum wird die Mischungsgeschwindigkeit  $u'$  in eine Grund- und eine Einspritz-turbulenz unterteilt. Dabei werden auftretende Effekte, die durch Einlassströmung oder Drall entstehen, in der Grundturbulenz erfasst und über die mittlere Kolbengeschwindigkeit abgebildet, was eine grobe Vereinfachung darstellt. Durch vektorielle Addition der Grundturbulenz und der Einspritz-turbulenz lässt sich Gleichung 12 wie folgt schreiben [2].

$$u' = \sqrt{c_G \cdot c_m^2 + c_{Kin} \cdot k} \quad (12)$$

Die beiden Parameter  $c_G$  und  $c_{Kin}$  erlauben eine Wichtung der beiden Anteile sowie einen Angleich des Diffusionsmodells an Messdaten [2]. Somit lässt sich

Gleichung 10 folgendermaßen formulieren:

$$\frac{dm_{B,V}}{dt} = \frac{\sqrt{c_G \cdot c_m^2 + c_{Kin} \cdot k}}{\sqrt[3]{\frac{V_Z}{\lambda \cdot Anz_D}}} \cdot m_{B,UV} \quad (13)$$

Die spezifische kinetische Turbulenzenergie  $k$  fasst die Diffusion des Kraftstoffes und die dem System zugeführte kinetische Energie, infolge der Einspritzung, zusammen. Dieses stark vereinfachte  $k - l_I$ -Modell wird entsprechend Gleichung 14 beschrieben.

$$\frac{dk}{dt} = -c_{Diss} \cdot \frac{1}{l_I} \cdot k^{\frac{3}{2}} + c_E \cdot \frac{dk_E}{dt} \quad (14)$$

Die zugeführte kinetische Energie ergibt sich aus dem zeitlichen Einspritzverlauf sowie der jeweils dazugehörigen Ausgangstropfengeschwindigkeit. Zusätzlich geht Barba von einer globalen Betrachtung des Zylinders aus, indem die gesamte Zylinderfüllung als Skalierung für die gesuchte kinetische Energiedichte herangezogen wird [2]. Dies lässt sich nach Gleichung 15 wie folgt beschreiben:

$$\frac{dk_E}{dt} = \frac{1}{2} \cdot \frac{dm_E}{dt} \cdot u_{T,r0}^2 \cdot \frac{1}{m_Z} \quad (15)$$

Mit der verbrannten Kraftstoffmasse aus Gleichung 13 und dem unteren Heizwert für Dieseldieselkraftstoff wird der zugeführte Wärmestrom nach Gleichung 16 berechnet.

$$\frac{dQ_B}{dt} = H_U \cdot \frac{dm_{B,V}}{dt} \quad (16)$$

Für die Simulationsrechnung ersetzt dieses Modell den zuvor verwendeten Ersatzbrennverlauf nach Vibe [9] in der Energiebilanz. In Abbildung 9 ist die Gegenüberstellung des Brennverlaufs aus der Druckverlaufsanalyse experimentell ermittelter Messdaten (DVA) und des phänomenologischen Modells dargestellt. Besonders im zweiten Teil der Hauptverbrennung zeigt sich, dass das phänomenologische Modell eine gute Übereinstimmung liefert. Durch den Bezug auf den Einspritzverlauf lässt sich das phänomenologische Modell vielseitig anwenden, besonders für die Verwendung von weiteren Teileinspritzungen sowie bei Variationen hydraulischer Parameter des Einspritzsystems.

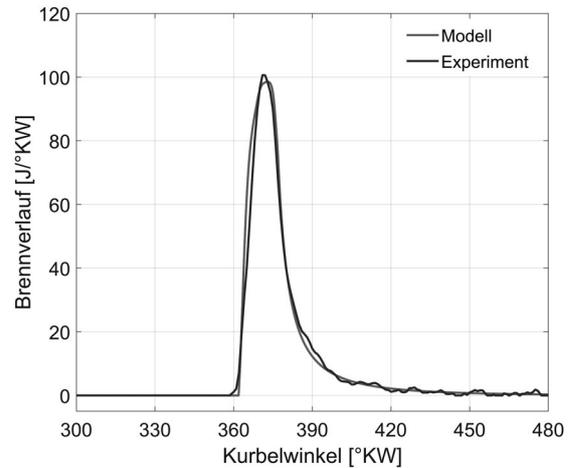


Abbildung 9: Vergleich Brennverlauf aus Druckverlaufsanalyse (DVA) mit Simulationsmodell

## 4 Validierung des Gesamtmodells

Zur Validierung des Gesamtmodells werden die Berechnungsergebnisse mit Messdaten vom Motorenprüfstand verglichen. Die Eingangsparameter wie Injektorbestromungsdauer, Raildruck und Ladedruck sind denen vom Prüfstand nachempfunden. In Ab-

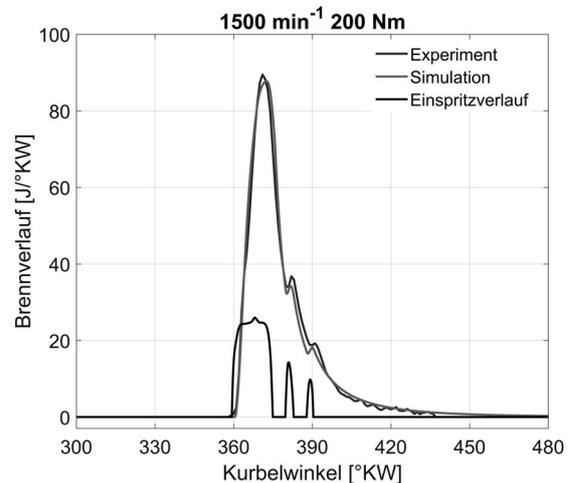
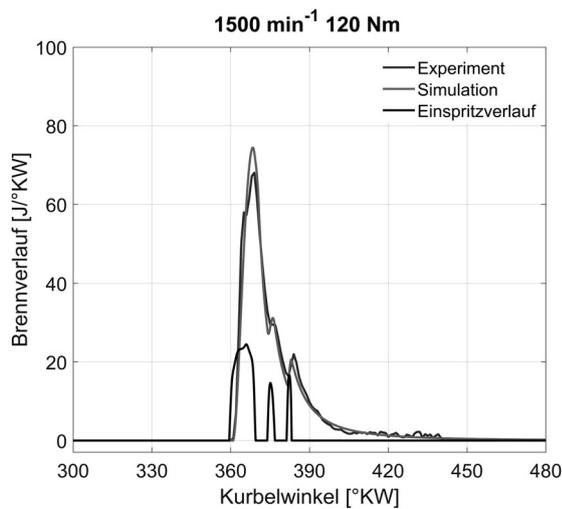
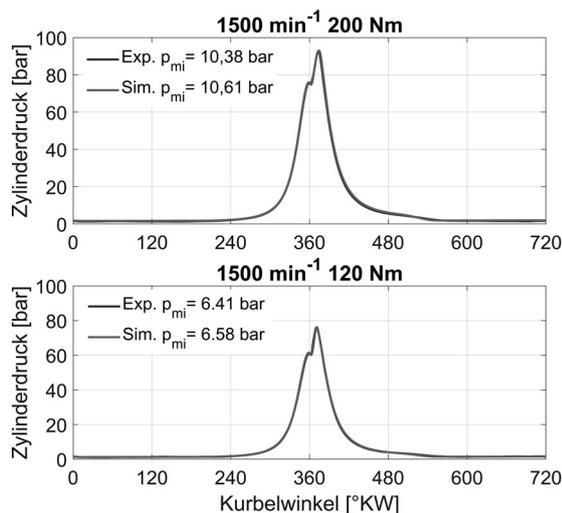


Abbildung 10: Vergleich des experimentell (DVA) und simulativ ermittelten Brennverlaufs bei Anwendung einer Mehrfacheinspritzung hohe Last



**Abbildung 11:** Vergleich des experimentell (DVA) und simulativ ermittelten Brennverlaufs bei Anwendung einer Mehrfacheinspritzung mittlere Last

Abbildung 10 und 11 ist ein Vergleich der Brennverläufe bei zwei verschiedenen Lastpunkten dargestellt. Es ist zu erkennen, dass das gekoppelte Simulationsmodell auch bei angewandeter Mehrfacheinspritzung eine hohe Übereinstimmung mit experimentell ermittelten Daten aufweist. Die nach der Haupteinspritzung einsetzende Wärmefreisetzung, die durch die angelegten Nacheinspritzungen entsteht, sind dabei deut-



**Abbildung 12:** Zylinderdruckverlauf über ein Arbeitsspiel im Vergleich von Experiment und Simulation

lich zu erkennen. Eine weitere Validierung erfolgt mit Hilfe der entsprechenden Zylinderverläufe in diesen Betriebspunkten, die in Abbildung 12 mit der Simulation gegenüber gestellt sind. Dabei zeigt sich, dass das Verbrennungsverhalten sehr gut wiedergegeben wird. Ein Vergleich des Mitteldruckes, als Größe zur Charakterisierung der Motorlast, zeigt eine Abweichung von 2 % zwischen Messung und Simulation. Ähnlich gute Ergebnisse wurden auch in anderen Betriebspunkten mit Ein- und Mehrfacheinspritzung festgestellt, was die Anwendbarkeit des Simulationsansatzes und der gewählten Modelle bestätigt.

## 5 Zusammenfassung und Ausblick

Der in diesem Beitrag dargestellte Modellansatz beschreibt ein 1D-Hydraulik Injektormodell in Simscape™, welches mit einem in Simulink® umgesetzten phänomenologischen Verbrennungsmodell gekoppelt wurde. Mit dem Injektormodell steht ein leistungsfähiges Werkzeug zur Verfügung, mit dem es möglich ist, das Injektorverhalten detailliert zu beschreiben. Dazu bietet das Verbrennungsmodell eine sehr gute Grundlage zur Berechnung der Wärmefreisetzung im Brennraum. Ein Vergleich der Simulationsergebnisse zeigte sehr gute Übereinstimmungen mit den Messergebnissen von den Prüfständen, was eine Anwendbarkeit dieser Modellansätze unterstützt. Ein wesentliches Ziel für zukünftige Arbeiten ist die Weiterentwicklung des Modells hinsichtlich der Betrachtung dynamischer Betriebszenarien. Des Weiteren steht die Implementierung eines NO-Emissionsmodells im Fokus, um das Emissionsverhalten in Parameterstudien zu analysieren und Optimierungsmaßnahmen abzuleiten.

### Danksagung

Die hier vorgestellten Ergebnisse wurden im Rahmen des öffentlichen geförderten Projektes SIDYN (FKZ:13FH043PX8) erarbeitet. Die Autoren danken dem BMBF für die Unterstützung des Vorhabens.

### Nomenklatur

- $\beta$  - Verdampfungskoeffizient [ $\text{m}^2/\text{s}$ ]
- $\lambda$  - Verbrennungsluftverhältnis [-]
- $\lambda_{zn}$  - Lokal vorherrschendes Luftverhältnis [-]

$v_{B.fl}$  - Kinematische Viskosität flüssig [ $m^2/s$ ]  
 $\rho_{B.fl}$  - Dichte Kraftstoff flüssig [ $kg/m^3$ ]  
 $\rho_Z$  - Dichte [ $kg/m^3$ ]  
 $\sigma_{B.fl}$  - Oberflächenspannung flüssig [ $N/m$ ]  
 $\tau$  - Zeit [ $s$ ]  
 $\tau_{ZV}$  - Zündverzögerung [ $s$ ]  
 $A_{D,eff}$  - Effektive Düsendurchschnitt [ $mm^2$ ]  
 $c$  - Allgemeine Konstante [-]  
 $c_m$  - Mittlere Kolbengeschwindigkeit [ $m/s$ ]  
 $d_{D,eff}$  - Effektive Düsendurchmesser [ $mm$ ]  
 $H_U$  - Unterer Heizwert [ $J/kg$ ]  
 $k$  - Kinetische Energiedichte [ $m^2/s^2$ ]  
 $l_{Diff}$  - Mischungslänge [ $m$ ]  
 $l_l$  - Längenmass [ $m$ ]  
 $m_B$  - Kraftstoffmasse [ $kg$ ]  
 $m_E$  - Eingespritzte Kraftstoffmasse [ $kg$ ]  
 $p_0$  - Umgebungsdruck [ $bar$ ]  
 $p_{mi}$  - Indizierter Mitteldruck [ $bar$ ]  
 $p_z$  - Zylinderdruck [ $bar$ ]  
 $Q_B$  - freigesetzte Energie (bei der Verbrennung) [ $J$ ]  
 $Re$  - Reynoldszahl [-]  
 $T_A$  - Aktivierungstemperatur [ $K$ ]  
 $t_{EB}$  - Zeit Einspritzbeginn [ $s$ ]  
 $t_{VB}$  - Zeit Verbrennungsbeginn [ $s$ ]  
 $T_Z$  - Mittlere Gastemperatur im Zylinder [ $K$ ]  
 $u'$  - Turbulenzintensität [ $m/s$ ]  
 $u_{T,r0}$  - Ausgangstropfengeschwindigkeit [ $m/s$ ]  
 $V_Z$  - Zylindervolumen [ $m^3$ ]  
 $We$  - Weberszahl [-]

- [5] Payri, R., Salvador, F.J., Carreres, M., De la Morena, J. *Fuel temperature influence on the performance of a last generation common-rail diesel ballistic injector. part ii: Id model development, validation and analysis* Energy Conversion and Management, 114:376-391, 2016.
- [6] Constien, M., R., Woschni G. *Vorausberechnung des Brennverlaufes aus dem Einspritzverlauf für einen direkteinspritzenden Dieselmotor*, MTZ 53, Nr.7/8, 1992.
- [7] Varde, K.S., Popa, D.M., Varde L.K. *Spray Angle and Atomization in Diesel Sprays*, SAE 841055, 1984.
- [8] Kožuch, P. *Ein phänomenologisches Modell zur kombinierten Stickoxid- und Rußberechnung bei direkteinspritzenden Dieselmotoren*, Stuttgart, Universität Stuttgart, Diss., 2004.
- [9] Vibe, I. *Brennverlauf und Kreisprozess von Verbrennungsmotoren*, VEB Verlag Technik, Berlin, 1970.
- [10] Grill, M. *Objektorientierte Prozessrechnung von Verbrennungsmotoren*, Stuttgart, Universität Stuttgart, Diss., 2006.

## References

- [1] Jörss, D., Herrmann, B., Fink, C. *Modellierung des Betriebsverhaltens eines Diesel-Industriemotors als Stromerzeuger*, Virtueller ASIM Workshop der Fachgruppe STS/GMMS & EDU, 2021, DOI: 10.11128/arep.45
- [2] Barba, C. *Erarbeitung von Verbrennungskennwerten aus Indizierdaten zur verbesserten Prognose und rechnerischen Simulation des Verbrennungsablaufes bei Pkw-DE-Dieselmotoren mit Common-Rail-Einspritzung*, Zürich, ETH Zürich, Diss., 2001.
- [3] Audi AG, [http://www.volkspage.net/technik/ssp/ssp/SSP\\_227.PDF](http://www.volkspage.net/technik/ssp/ssp/SSP_227.PDF), letzter Zugriff: 08.12.2022
- [4] Vermeiden, J.C. *Fuel injection equipment characterisation for short and multiple injections experimental setup design and model-based sensitivity analysis*, Eindhoven University of Technology, 2020.

# Automatisches Mapping des dynamischen Umfelds in cyber-physischen Systemen

Taihao Li<sup>1\*</sup>, Marian Göllner<sup>1</sup>, Sven Jacobitz<sup>1</sup>, Xiaobo Liu-Henke<sup>1</sup>

<sup>1</sup>Institut für Mechatronik, Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel; \*[ta.li@ostfalia.de](mailto:ta.li@ostfalia.de)

**Kurzfassung.** Im Rahmen des Niedersächsischen Zukunftslabors für Mobilität hat die Fachgruppe um Prof. Liu-Henke ein Cyber-Physisches-Labortestfeld zur Echtzeit-Erprobung und -Optimierung intelligenter, vernetzter, autonomer Fahrfunktionen entwickelt. Eine wesentliche Voraussetzung für die autonome Fahrt in cyber-physischen Systemen ist das automatische Mapping. Dieses muss die Umgebung hinreichend genau abbilden und dazu dynamisch an geänderte Umfeldbedingungen angepasst werden. In diesem Beitrag wird unter Nutzung des Robot Operating Systems (ROS) eine Funktion zur kontinuierlichen, automatischen Kartierung des Umfelds im CPS-Testfeld ausgelegt. Durch Auswertung von Zustandssensoren zur Umfelderkennung und anschließender Sensordatenfusion wird die Grundlage zur Erstellung einer dynamischen Umfeldkarte gelegt.

## Einleitung

Die intelligente Mobilität wird in der Hightech-Strategie des Bundesministeriums für Bildung und Forschung [1] als eine der wichtigsten Zukunftstechnologien bewertet. Das automatisierte und vernetzte Fahren spielt dabei eine wesentliche Rolle um die Anforderungen der Mobilität der Zukunft zu erfüllen und einen sauberen, leisen, zuverlässigen und verbrauchsgünstigen Personen- und Güterverkehr zu ermöglichen. Die Fachgruppe für Regelungstechnik und Fahrzeugmechanik unter der Leitung von Frau Prof. Liu-Henke erforscht als Mitglied des niedersächsischen Zukunftslabors [2] Lösungen durch Nutzung digitaler Technologien und baut zu diesem Zweck ein cyber-physisches Labortestfeld für die Entwicklung intelligenter und vernetzter Mobilitätsfunktionen auf [3]. Ziel dieses CPS-Labortestfelds ist es, ein zu testendes autonomes System wiederholt in Echtzeit unter Laborbedingungen zu verifizieren. Derzeit unterstützt das Testfeld die Entwicklung und Erprobung innovativer Funktionen in zwei Anwendungsfällen: dem autonomen Fahren im intelligenten Straßenverkehr und der fahrerlosen Intralogistik in einer Industrie-4.0 Umgebung. Dazu steht eine autonome Mobilplattform (AGV) zur Verfügung, die

sich durch modulare und rekonfigurierbare Bauweise auf den jeweiligen Anwendungsfall anpassen lässt.

Damit die autonome Mobilplattform (AGV) innerhalb der dynamischen Umgebung des Labortestfeldes navigieren kann, ist es nötig eine lokale dynamische Karte mithilfe fahrzeugeigener Sensorik aufzubauen, diese auf ein globales Koordinatensystem zu referenzieren und sich selbst in dieser Karte zu lokalisieren. Die Positionsbestimmung ist nach Cox [4] dabei eines der grundlegendsten Probleme, welches hier detailliert beschrieben werden soll.

Dazu ist dieser Artikel wie folgt gegliedert: In Abschnitt 1 wird zunächst die Methodik vorgestellt, welche bei der Entwicklung der Kartenerstellungsfunktion zur Anwendung kam. Im Kapitel 2 wird ein Überblick über den aktuellen Stand der Technik auf diesem Gebiet gegeben. Abschnitt 3 erläutert die Konzeption der Funktionen. Entwurf und Ausarbeitung dieser Funktion erfolgt in Abschnitt 4 basierend auf der vorherigen Konzeptionen. Abschnitt 5 beschreibt die Bewertung der Kartenerstellung. Zum Schluss wird die Arbeit zusammengefasst und ein Ausblick gegeben.

## 1 Methodik

Die Entwicklung des Kartenerstellungsalgorithmus ist ein iterativer Prozess, welcher auf der Methodik der mechatronischen Komposition beruht. Die Entwicklung solcher Funktionen erfolgt auf der Grundlage von Anforderungen und Modellen. Der erste Schritt besteht daher darin, die Anforderungen zu definieren. Danach folgt die Abbildung der realen Funktion mit Hilfe mathematischer Gleichungen, auch Modellbildung genannt. Anschließend wird die Funktion konzeptioniert und realisiert. Abschließend wird die entwickelte Funktion validiert werden, um festzustellen, ob sie die Anforderungen erfüllt.

Um den iterativen Prozess darzustellen, wird in die-

ser Arbeit die Funktionsentwicklung nach dem mechatronischen Kreislauf [5] durchgeführt. Abb. 1 stellt den mechatronischen Entwicklungskreislauf der Entwicklung der Kartenerstellungsfunktion dar.

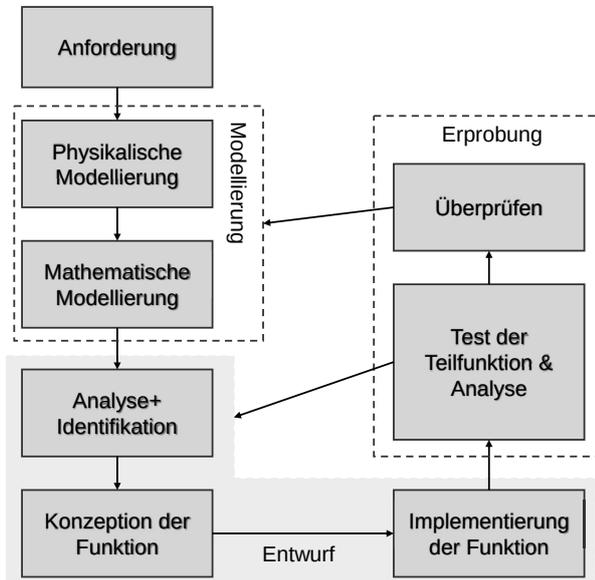


Abbildung 1: Mechatronischer Entwicklungskreislauf

Der Entwurfsprozess beginnt mit der Modellbildung basierend auf dem realen System, welches gemäß der Anforderungen reduziert bzw. vereinfacht wird, sodass sich zunächst ein physikalisches Modell ergibt. Dieses wird mithilfe physikalischer Gesetzmäßigkeiten in ein mathematisches Modell überführt, welches wiederum bspw. in Form von Signalflussplänen im Rechner abgebildet und mithilfe von CAE-Werkzeugen und entsprechender Numerik simuliert werden kann.

Der Modellbildungsprozess umfasst zudem Messungen am realen System, um zum einen die Parameter des mathematischen Modells zu identifizieren und zum anderen die Simulation zu validieren. Eine anschließende Analyse der Simulationsergebnisse lässt Schlussfolgerungen über die grundlegenden statischen und dynamischen Eigenschaften des realen Systems zu, auf dessen Grundlage die Konzeption der Funktionen erfolgt. Es werden sowohl die Funktionsarchitektur als auch Ansätze zu dessen Auslegung und Optimierung festgelegt. Komplexe Funktionen werden zudem nach dem verallgemeinerten Kaskadenprinzip in hierarchische Teilfunktionen zerlegt, um die Funktionskomplexität zu reduzieren und so den Auslegeprozess handhabbar zu machen. Dazu werden bereits in einem frü-

hen Entwicklungsstadium Hard- und Softwareanforderungen berücksichtigt und Schnittstellen für die funktionsübergreifende Kommunikation definiert.

Der Erprobungsprozess wird parallel zur Entwicklung durchgeführt. Wenn eine Teilfunktion entwickelt worden ist, wird diese getestet und analysiert.

## 2 Stand des Wissens

### 2.1 Robot Operating System (ROS)

Um die Funktionen der Mobilplattform mit verschiedener Hardware schnell zu entwickeln, verwendet diese Arbeit das Robot Operating System (ROS) als Entwicklungsplattform. Abb.2 zeigt ein Beispiel für die Datenübertragung zwischen verschiedenen Geräten in einer ROS-Umgebung. Die Kommunikation erfolgt über das Bereitstellen (publish) von Nachrichten durch Knoten (Nodes) welche wiederum von anderen Knoten abonniert (subscribe) werden können. Aus Abb.2 ist ebenfalls ersichtlich, dass in einem Gerät mehr als ein Knoten gesetzt werden kann. weiterhin ist ein ROS-Master für die Kommunikation erforderlich, welcher die Verwaltung und Registrierung von Knoten und deren bereitgestellte Nachrichten verantwortet. Die Nachrichten werden zudem in drei Kategorien unterschieden, Thema-, Aktion- und Service-Nachrichten, um deren Priorisierung und Zyklidizität zu deklarieren.

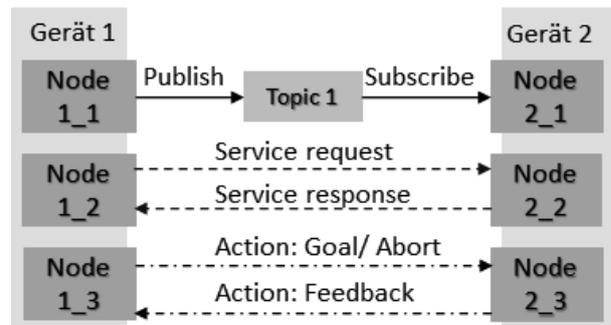


Abbildung 2: Architektur der Kommunikation im ROS

### 2.2 Darstellung der Karte

Die Erstellung von Umgebungsmodellen ist für die Entwicklung autonomer / hochautomatisierter mobiler Robotersysteme unerlässlich. Das Erstellen des Umgebungsmodells erfolgt mittels Algorithmen zur simultanen Lokalisierung und Kartierung (eng. Simultaneous

Localization and Mapping (SLAM)) welche im nächsten Abschnitt näher vorgestellt werden. Zur Darstellung und Speicherung dieses Umgebungsmodell eignen sich Topologische und geometrische Karten.

**Topologische Karte:** Bei der topologischen Karte wird die Umgebung in Form eines topologischen Graphen dargestellt. Die Topologie vermeidet die direkte Messung von Umgebungsdaten, erfordert keine weiteren Kartendetails und konzentriert sich mehr auf die Beziehung zwischen Kartenelementen. erstmalige verwendet für die freie Navigation findet sich im Jahr 1988 bei Kuiper und Byun [6].

**Geometrische Karte:** Eine geometrische Karte ist ein Kartenmodell, das die Umgebung genau darstellen kann. In [7] wurde z.B. eine kontinuierliche Karte, welche Merkmale in der Umgebung durch Punkte und Linien darstellt, zur freien Navigation verwendet. Eine weitere Art der geometrischen Karte ist die Gitterkarte. In dieser wird die kontinuierliche Umgebung in mehrere diskreten Gitter zerlegt wie dies z.B. in [8] gezeigt wurde.

### 2.3 Etablierte Simultaneous Localization and Mapping (SLAM) Verfahren

Die derzeit am häufigsten genutzten SLAM-Verfahren wurden in verschiedenen Studien [9, 10, 11, 12] vorgestellt und bewertet. Dabei wurden im wesentlichen die vier SLAM-Algorithmen GMapping, Cartographer, Karto-SLAM und HectorSLAM miteinander verglichen. Aus den Bewertungen ist zu schlussfolgern, dass Cartographer in kleinen und großen Szenarien und in jedem Bewegungszustand des Roboters die genaueste und realitätstreuere Karte erstellt und zudem auch sehr robust gegenüber Störungen ist. Allerdings haben die verwendeten Sensoren und die Größe der Umgebung Auswirkungen auf die Kartierungsergebnisse dieses Algorithmus. Deswegen kann in dieser Arbeit der Cartographer SLAM-Algorithmus nur durch ein-satzspezifische Optimierung der Parameter während der Verwendung angepasst werden, um die Umgebung genau kartieren zu können.

## 3 Konzept der Funktion

Das Ziel dieser Arbeit ist es, eine autonome mobile Plattform in die Lage zu versetzen, eine zuverlässige und genaue Karte in einer unbekannt, unstrukturierter Umgebung zu erstellen.

Ein Problem des derzeitigen Ansatzes besteht darin, dass beim händischen Mapping nur feste und konstante Objekte in der Umgebung sowie die Länge und Breite des Raums abgebildet werden können. Eine genauere Beschreibung des Raumes ist mit der händischen Kartierung nicht möglich. Das Problem führt daher zu der grundlegenden Anforderung, mobile Plattformen in die Lage zu versetzen, eine detaillierte und genaue Karte in einer unstrukturierten und flexiblen Umgebung zu erstellen. Um dieses Ziel zu erreichen, wird eine funktionale Architektur, wie in der Abbildung 3 dargestellt, vorgeschlagen.

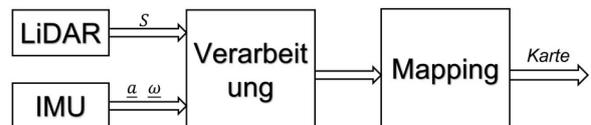


Abbildung 3: Konzeption der Funktion

Für die Kartenerstellung ist es notwendig, entsprechende Sensoren zu nutzen, die die Umgebung zuverlässig abtasten und Rückschlüsse auf Relativbewegungen des Fahrzeugs zulassen. Hier werden also nicht nur Umgebungsdaten, sondern auch die Bewegungsdaten des eigenen Fahrzeugs benötigt. Die Umgebungsdaten werden dabei von LiDAR-Sensoren der Plattform in Form eines Scans erfasst. Ein Scan  $S$  ist eine Sammlung, die aus einer endlichen Anzahl  $K$  von Laserpunkten  $s_i$  besteht und durch die Formel 1 ausgedrückt wird.

$$S = \{s_i \mid i \leq K, i \in \mathbb{N}\} \quad (1)$$

Bewegungsdaten werden von einer IMU des Fahrzeuges gesammelt. Sie misst die Beschleunigung  $\underline{a}$  in den Achsen  $x, y, z$  und die Winkelgeschwindigkeit  $\underline{\omega}$  um die Achsen  $x, y$  und  $z$ . Die gesammelten Messdaten müssen zu einem Zeitpunkt einmal transformiert werden, um sie in einem einheitlichen Koordinatensystem miteinander fusionieren zu können. Nur so ist es möglich nicht nur eine Karte aufzubauen, sondern gleichzeitig auch die Position der Plattform innerhalb dieser zu ermitteln (SLAM).

## 4 Entwurf der Kartenerstellung

Die Kartenerstellung basiert wie bereits erwähnt auf dem Cartographer Algorithmus [13]. Abb. 4 zeigt den Ablauf der Kartenerstellung. Diese Funktion enthält die folgenden Schritte: Filterung, Transformation,

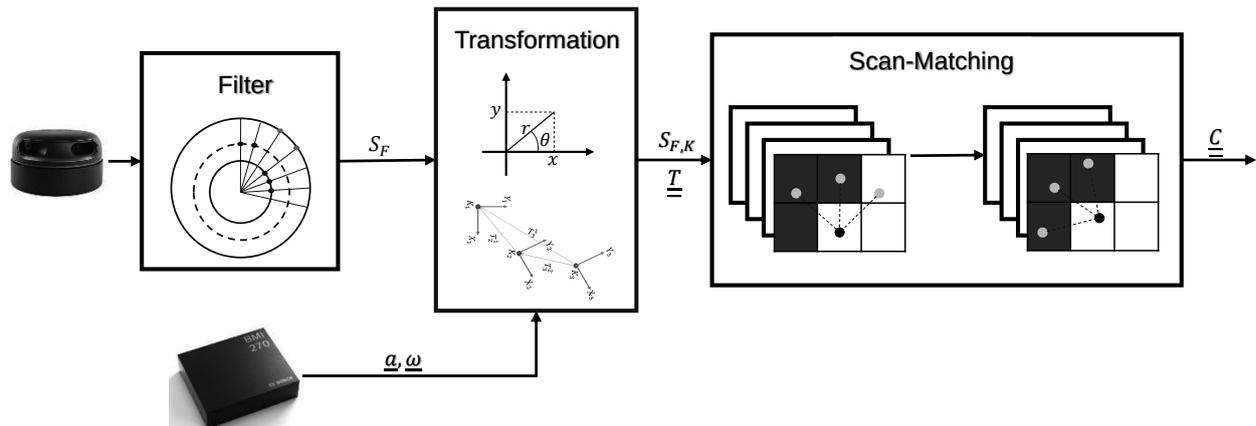


Abbildung 4: Entwurf der Kartierung

Scan-Matching und Kartenaktualisierung. In den folgenden Abschnitten werden diese einzelnen Bestandteile detailliert erläutert. Weiterhin werden die verwendeten Sensoren und die Datenstruktur der Karte vorgestellt.

#### 4.1 Verwendete Sensorik

In diesem Abschnitt werden zunächst die verwendeten Sensoren der Plattform vorgestellt. Der in dieser Arbeit verwendete LiDAR-Sensor ist ein kostengünstiger RPLiDAR A2, der mittels rotierendem Einzellaserkopf die Scandaten  $S$  als 2-dimensionale Koordinatenpunktemenge liefert. Rohdaten werden in Polarkoordinaten dargestellt. Jeder Laserpunkt  $\underline{s}_i$  ist definiert über einen Entfernungswert  $d_i$  und einen Winkel  $\theta_i$ , durch die Formel  $\underline{s}_i = (d_i, \theta_i)$ .

Das in dieser Arbeit verwendete IMU ist ein BMI055 Chip innerhalb einer Intel Realsense D435i, ein 6-Achsen IMU, welches einen Beschleunigungsmesser und ein Gyroskop mit jeweils drei zueinander orthogonalen Achsen umfasst.

#### 4.2 Filterung der Sensordaten

Die Qualität der Kartenerstellung ist abhängig von der Qualität der Sensordaten. Deswegen ist es notwendig, dass die Sensordaten vor der Kartenerstellung gefiltert werden. Beispielsweise führt die Montage LiDAR-Scanners auf dem Roboterchassis dazu, dass dieser den Roboterkörper selbst mitscannet. Ebenfalls möglich ist, dass ein großer Messwert stark rauscht, was ebenfalls eine Filterung notwendig macht.

Zur Filterung der LiDAR-Daten wird eine Pass-Through-Filterung durchgeführt, um weiter entfernte oder nahe Messwert zu filtern. In dieser Filterung werden zwei Schwellwerte eingestellt, nämlich „ $d_{min}$ “ und „ $d_{max}$ “. Wenn der gemessene Wert Formel 2 erfüllt, kann dieser Laserpunkt zur Kartenerstellung verwendet werden. Der gefilterte Laserpunkt wird als  $\underline{s}_{F,K_k}$  gezeichnet. Ebenso enthält jeder gefilterte Laserpunkt  $\underline{s}_{F,K_k}$  auch entsprechende Entfernungsinformationen  $d_k$  und Winkelinformationen  $\theta_k$ . Mithilfe dieser Filterung können alle Laserpunkte, die sich negativ auf der Kartenerstellung auswirken würden, zuvor gelöscht werden.

$$\{d_i \in \mathbb{R} \mid d_{min} < d_i < d_{max}\} \quad (2)$$

#### 4.3 Definition des Koordinatensystem

Die Definition des Koordinatensystem ist der Schlüssel zu einer genauen Beschreibung der Position des Fahrzeugs. Abb.5 zeigt zunächst alle benötigten Koordinatensysteme. In dieser Arbeit werden rechtshändige Koordinatensysteme genutzt; wenn die jeweilige X-Achse nach vorne zeigt, zeigt also die jeweilige Y-Achse nach rechts.

- AGV-Koordinatensystem  $BCS_{AGV}$ : Dieses Koordinatensystem ist ein Körperkoordinatensystem. Es befindet sich in der geometrischen Mitte des AGVs. Die positive Richtung der X-Achse des AGV-Koordinatensystems wird entlang der Vorwärtsrichtung des Fahrzeugs definiert.
- LiDAR-Koordinatensystem  $BCS_{LiDAR}$ : Die Position des Koordinatensystem entspricht der realen

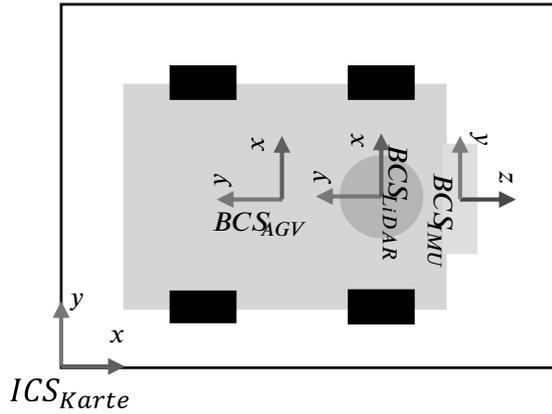


Abbildung 5: Definition der Koordinatensysteme

Position des Sensors im AGV. Hier zeigt die positive X-Achse zur rechten Seite des AGVs.

- IMU-Koordinatensystem  $BCS_{IMU}$ : Das Körperkoordinatensystem des IMU - hier zeigt die Z-Achse Richtung Front des Fahrzeugs.
- Kartenkoordinatensystem  $ICS_{Karte}$ : Dieses Koordinatensystem wird als globales Koordinatensystem definiert, welches während der Kartenerstellung fest ist. Dieses Koordinatensystem wird im ersten Rechenschritt an der Position des AGV-Koordinatensystems initialisiert.

Die Koordinatensysteme der beiden Sensoren  $BCS_{LiDAR}$  und  $BCS_{IMU}$  werden entsprechend der realen Position im Fahrzeug definiert (siehe auch Abb.5). Zur Erstellung der Karte müssen zunächst alle erfassten Sensordaten in kartesischen Koordinaten vorliegen. Für die LiDAR-Scandaten ist dazu die in Formel 3 beschriebene Umwandlung der Polarkoordinaten in das kartesische Koordinatensystem nötig.

$$\begin{aligned} x_k &= d_k \cos \theta_k \\ y_k &= d_k \sin \theta_k \end{aligned} \quad (3)$$

Nach der Umwandlung werden die Scandaten wie in Formel 4 dargestellt, wobei  $\underline{s}_{F,K_k}$  ein umgewandelter Laserpunkt ist.

$$S_{F,K} = \{ \underline{s}_{F,K_k} \mid j \leq M, i \in \mathbb{N} \} \quad (4)$$

Anschließend werden die umgerechnete Daten aus den jeweiligen lokalen Koordinatensystemen  $BCS_{Sensor} = \{BCS_{LiDAR}, BCS_{IMU}\}$  in das globale Koordinatensystem  $ICS_{Karte}$  transformiert. Im weiteren Verlauf wird jeder Laserpunkt  $\underline{s}_{F,K_k}$  für die Kartierung meh-

rere Koordinatentransformationen durchlaufen, deren Ablauf durch die Formel 5 und Formel 6 definiert ist.

$$\underline{s}_{F,K_k}^{BCS_{AGV}} = \underline{T}_{BCS_{LiDAR}}^{BCS_{AGV}} \cdot \underline{s}_{F,K_k} \quad (5)$$

$$\underline{s}_{F,K_k}^{ICS_{Karte}} = \underline{T}_{BCS_{AGV}}^{ICS_{Karte}} \cdot \underline{s}_{F,K_k}^{BCS_{AGV}} \quad (6)$$

Wobei  $\underline{T}_{BCS_{Fahrzeug}}^{ICS_{Karte}}$  und  $\underline{T}_{BCS_{LiDAR}}^{BCS_{Fahrzeug}}$  zwei Transformationsmatrizen sind. Eine allgemeine Form der Transformationsmatrix besteht aus zwei Teilen, der Drehmatrix  $\underline{R}$  und der Translationsmatrix  $\underline{t}$ , wie durch die Formel 7 dargestellt.

$$\underline{T} = \begin{pmatrix} \underline{R} & \underline{t} \\ 0 & 1 \end{pmatrix} \quad (7)$$

Die Transformationsmatrix  $\underline{T}_{BCS_{Fahrzeug}}^{ICS_{Karte}}$  hängt von der Pose des Fahrzeug  $\underline{\zeta}$  ab. Die Pose enthält die Position des Fahrzeugs  $\underline{t}_f$  und die Ausrichtung  $\theta_f$ , die durch IMU (Formel 8) geschätzt wird. Die Pose  $\theta_f$  wird durch die Formel 9 in eine Transformationsmatrix  $\underline{T}_{BCS_{Fahrzeug}}^{ICS_{Karte}}$  umgewandelt.

$$\underline{\zeta} = f(\underline{a}, \underline{\omega}, \underline{T}_{BCS_{IMU}}^{BCS_{AGV}}) \quad (8)$$

$$\underline{\zeta} = \begin{bmatrix} \underline{R}(\theta_f) & \underline{t}_f \\ 0 & 1 \end{bmatrix} \quad (9)$$

#### 4.4 Datenstruktur des verwendeten Kartentypus

In dieser Arbeit wird die Gitterkarte als Untertypus der geometrischen Karten zur Darstellung und Speicherung des Umgebungsmodells verwendet. Im Gitterkartentypus wird wiederum die Belegungsgitterkarte ausgewählt. Wie oben erwähnt, wird in dieser Karte der Zustand jedes Gitters durch Wahrscheinlichkeitswert bestimmt. Die Gitterkarte und die Elemente der Gitterkarte werden als eine Matrix  $\underline{C}$  mit den Matrizenelementen  $c_{ij}$  definiert (Formel 10):

$$\underline{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix} \quad (10)$$

Jedes Gitter hat drei Zustände, nämlich frei, belegt und unbekannt. Diese drei Zustände können jeweils

durch die Nummern 0, 1 und -1 dargestellt werden. Wegen des Rauschens der Messung kann ein Laserpunkt nicht immer mit einem bestimmten Gitterpunkt zusammenfallen. Daher wird der Zustand jedes Gitters durch eine Wahrscheinlichkeit bestimmt. Wenn der Laserpunkt einmal in einem Gitter liegt, wird der Wahrscheinlichkeitswert, dass das Gitter als belegt eingestuft werden kann, erhöht. Der Aktualisierungsprozess wird in dem nächsten Abschnitt beschrieben.

#### 4.5 Scan-Matching

Das Ziel des Scan-Matching ist eine optimale Plattform-Pose  $\underline{\zeta}_f$  zu finden, und so den Positionierfehler zwischen einem Scan-Laserpunkt und dem entsprechenden Kartegitter zu minimieren. In diesem Fall wird dieser Prozess als ein nicht-lineares kleinste-Quadrate Problem (Formel 11) bezeichnet.

$$\min e = \sum_{k=1}^M \left( \underline{C}_k - \underline{T}_f \cdot s_k \right)^2 \quad (11)$$

Wobei  $s_k$  ein Laserpunkt des Scans im AGV-Koordinatensystem ( $BCS_{AGV}$ ), d.h.  $s_{F,K_k}^{BCS_{AGV}}$  ist. Der Index des Laserpunktes in einem Scan ist  $k$ . Die Anzahl der Laserpunkte in einem Scan wird hier mit  $M$  bezeichnet. Jeder Laserpunkt wird vom AGV-Koordinatensystem in das Karten-Koordinatensystem ( $ICS_{Karte}$ ) transformiert.  $\underline{C}_k$  ist die Koordinate der korrespondierenden Karte.  $\underline{T}_f$  ist eine vereinfachte Notation für  $\underline{T}_{BCS_{Fahrzeug}^{ICS_{Karte}}}$ . Der Zweck der Gleichung ist es, eine optimale Transformationsmatrix  $\underline{T}_f$  zu finden, welche die Abweichung zwischen Laserpunkt und dem korrespondierenden Gitter minimiert. Der Anfangswert der Matrix  $\underline{T}_f$  wird durch die IMU entsprechend der Formeln 8 und 9 geschätzt.

#### 4.6 Kartenaktualisierung

Jedes Gitter der Belegungskarte hat einen initialen Wahrscheinlichkeitswert für den Belegungszustand  $P(c_{ij} = 1)$ . Der Zustand des gesamten Gitters kann mit der Formel 12 beschrieben werden. Wenn eine neue Beobachtung eingeht, kann die Aktualisierung der Karte gemäß der Bayes'schen Formel erfolgen, wobei  $z$  die Beobachtung selbst ist. Um die Berechnung zu vereinfachen, kann die Formel 13 Logarithmiert werden.

$$\text{odds}(c_{ij}) = \frac{P(c_{ij} = 1)}{P(c_{ij} = 0)} \quad (12)$$

$$\text{odds}(c_{ij} | z) = \frac{P(z | c_{ij} = 1)}{P(z | c_{ij} = 0)} \text{odds}(c_{ij}) \quad (13)$$

Abb. 6 zeigt des Aktualisierungsprozess der Karte. Zum Zeitpunkt  $t = 0$  wird jedes Gitter der Karte mit null Log-odds-Werten initialisiert. Diese Initialisierung ist identisch mit der Wahrscheinlichkeit, dass die Zellen frei sind. Zum Zeitpunkt  $t = 1$  werden neue Messungen von dem LiDAR-Sensor empfangen, der in diesem Beispiel viele einzelne Strahlen aussendet. Auf dieser Abbildung ist deutlich zu sehen, dass die gelben Zellen als besetzt und die hellgrünen Zellen als freier, leerer Raum gemessen werden. Der Wert von  $\text{odds}_{belegt}$  wird als 0,8 und der Wert von  $\text{odds}_{frei}$  als  $-0,5$  angenommen.

Anschließend werden die Werte der beobachteten Gitter aktualisiert. Die Werte der unbeobachteten Gitter bleiben unverändert. Wenn der Log-odds-Wert einen Schwellenwert überschreitet, wird das Gitter als Belegt gekennzeichnet. Wenn der Log-odds-Wert unter einem Schwellenwert liegt, wird das Gitter als frei betrachtet.

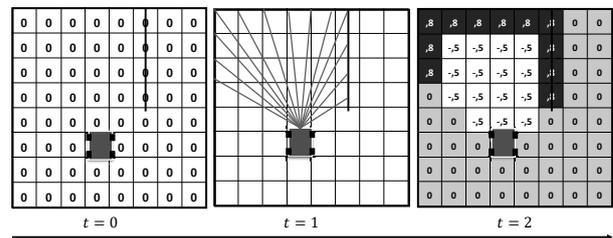


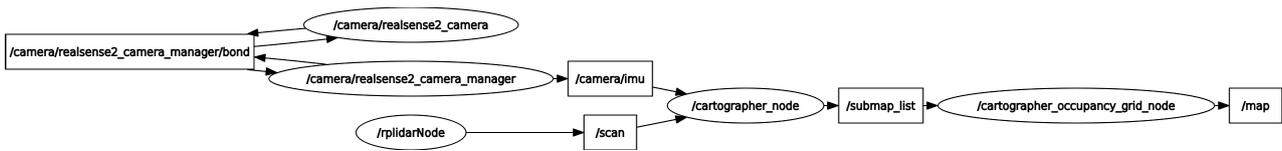
Abbildung 6: Aktualisierung der Karte

## 5 Realisierung der Kartenerstellung

Dieses Kapitel beschreibt die Realisierung der Kartenerstellung auf der Mobilplattform innerhalb der ROS-Umgebung.

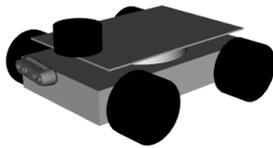
### 5.1 Einrichtung und Konfiguration in ROS

Vor der Kartierung muss zunächst der räumliche Zustand des Fahrzeugs in ROS beschrieben werden, wobei ROS eine eigene Koordinatensystemdefinition



**Abbildung 7:** Knotendiagramm

nutzt. Daher müssen die in diesem Artikel in Entwurf definierten Koordinatensysteme gemäß den ROS-Koordinatensystemregeln REP105 und REP103 angepasst werden. Danach können die Beziehungen zwischen den verschiedenen Sensorkoordinatensystemen auf der Plattform mithilfe derer CAD-Modelle bestimmt und in einer URDF-Datei, einem Werkzeug zur Beschreibung des Objektmodells, eingetragen werden. Nach der Beschreibung der Beziehungen jedes Koordinatensystem kann das Modell der Plattform im ROS wie in Abbildung 8 dargestellt werden.

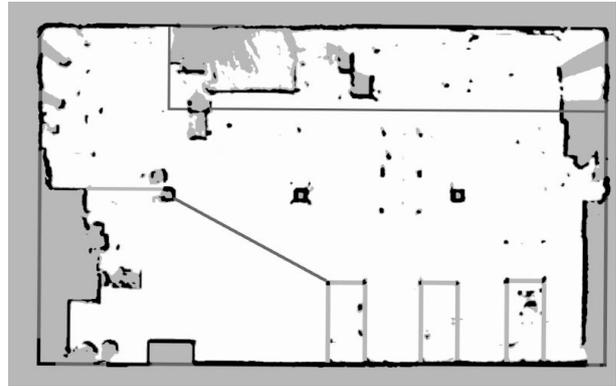


**Abbildung 8:** Fahrzeug-Modell im ROS

Um die Kartenerstellung in ROS umzusetzen, werden zunächst alle benötigten Knoten konfiguriert. Abb. 7 zeigt das entsprechende Knotendiagramm. Dadurch können die erforderlichen Informationen für die Kartenerstellung korrekt zwischen den Knoten übertragen werden.

## 5.2 Ergebnis

Abb. 9 zeigt die erstellte Karte. Die grauen Linien im Diagramm sind die Hilfslinien, die zum Aufteilen der Bereiche verwendet werden, um eine Fabrik zu simulieren. Um die Genauigkeit der erstellten Karten zu überprüfen, muss eine Karte durch externe Messungen zur Verifikation erstellen und mit der durch den Algorithmus erstellten Karte verglichen werden. Die geometrische Daten des Raums werden durch einen Laserentfernungsmesser gemessen. Dann wird die Verifikationskarte mit der erstellte Karte ausgerichtet und aufeinander gelegt, wie die Abb. 10 gezeigt. In X-Richtung ist der Messfehler 3 cm und in Y-Richtung 2 cm groß. Die Fehler in beide Richtungen sind kleiner als die Auflö-



**Abbildung 9:** Die erstellte Karte



**Abbildung 10:** Vergleich der Karten

sung der Karte. Deswegen kann diese erstellte Karte für weitere Arbeiten als Basis verwendet werden.

## 6 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde eine Funktion für eine autonome Mobilplattform zur Kartenerstellung realisiert. Der große Vorteil der Funktion ist, dass die Mobilplattform in einer unstrukturierten Umgebung automatisch und schnell eine zuverlässige Karte erstellen kann. Gleichzeitig kann die Plattform auch selbst die eigene

Position bestimmen. In dieser Arbeit werden die benötigten Koordinatensysteme und der Informationsfluss definiert, der für die Erstellung der Karte erforderlich ist. Deshalb ist dieser auch übergreifend für verschiedene Mobilplattformen und verschiedene Sensoren zu verwenden. Zum Schluss wird die erstellte Karte verifiziert und gezeigt, dass die durch den Algorithmus erstellte Karte als eine Referenzkarte verwendet werden kann.

Allerdings enthielt diese Arbeit auch Teile, die verbessert werden können. Der Zustand des Fahrzeugs könnte durch mehr Sensoren besser geschätzt werden, umso die Eigenlokalisierung zu verbessern. Außerdem kann die Auflösung der Karte aufgrund der Leistungsfähigkeit des Computers nicht erhöht werden, was jedoch in Zukunft optimiert werden kann. Sobald die Karte erstellt ist, wird sie als statische Referenzkarte für nachfolgenden Arbeiten wie die Lokalisierung und Navigation verwendet.

## Danksagung

Gefördert vom Niedersächsischen Ministerium für Wissenschaft und Kultur unter Fördernummer ZN3495 im Niedersächsischen Vorab der VolkswagenStiftung und betreut vom Zentrum für digitale Innovationen (ZDIN).



## Literatur

- [1] Bundesministerium für Bildung und Forschung (BMBF). *Die neue Hightech-Strategie - Innovationen für Deutschland*. Berlin. 2014.
- [2] Zentrum für digitale Innovationen Niedersachsen (ZDIN). *Digital klar voraus. Branchenübergreifende Digitalisierung und Weiterentwicklung am ZDIN*. Oldenburg: OFFIS e. V. 2022.
- [3] Liu-Henke X, Jacobitz S, Göllner M, Zhang J, Scherler S, Yarom OA. Cyber-physical Industry 4.0 laboratory test field to simulate self-optimizing intralogistics. In: *2020 19th International Conference on Mechatronics-Mechatronika (ME)*. IEEE. 2020; pp. 1–6.
- [4] Cox JJ. Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on robotics and automation*. 1991; 7(2):193–204.
- [5] Liu-Henke X. *Mechatronische entwicklung der aktiven feder-/neigetechnik für das schienen-fahrzeug RailCab*. VDI Verlag. 2005. 149 p.
- [6] Kuipers B, Byun YT. A Robust, Qualitative Method for Robot Spatial Learning. In: *AAAI*, vol. 88. 1988; pp. 774–779.
- [7] Tomatis N, Nourbakhsh I, Siegwart R. Simultaneous localization and map building: A global topological model with local metric maps. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1. IEEE. 2001; pp. 421–426.
- [8] Li Y, Ruichek Y. Occupancy grid mapping in urban environments from a moving on-board stereo-vision system. *Sensors*. 2014;14(6):10454–10478.
- [9] Fan X, Wang Y, Zhang Z. An evaluation of Lidar-based 2D SLAM techniques with an exploration mode. In: *Journal of Physics: Conference Series*, vol. 1905. IOP Publishing. 2021; .
- [10] Filipenko M, Afanasyev I. Comparison of various slam systems for mobile robot in an indoor environment. In: *2018 International Conference on Intelligent Systems (IS)*. IEEE. 2018; pp. 400–407.
- [11] Santos JM, Portugal D, Rocha RP. An evaluation of 2D SLAM techniques available in robot operating system. In: *2013 IEEE international symposium on safety, security, and rescue robotics (SSRR)*. IEEE. 2013; pp. 1–6.
- [12] Yagfarov R, Ivanou M, Afanasyev I. Map comparison of lidar-based 2d slam algorithms using precise ground truth. In: *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE. 2018; pp. 1979–1983.
- [13] Hess W, Kohler D, Rapp H, Andor D. Real-time loop closure in 2D LIDAR SLAM. In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2016; pp. 1271–1278.

# Konzept eines ganzheitlichen, hochflexibel konfigurierbaren HiL-Prüfsystems für den Test autonomer Fahrfunktionen

Xiaobo Liu-Henke<sup>1\*</sup>, Marian Göllner<sup>1</sup>, Sven Jacobitz<sup>1</sup>, Jie Zhang<sup>1</sup>

<sup>1</sup>Institut für Mechatronik Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel, Deutschland; \*x.liu-henke@ostfalia.de

**Kurzfassung.** Autonomes Fahren und vernetzte cyber-physische Verkehrssysteme stellen immer größer werdende Herausforderungen an die Entwicklung und Absicherung erweiterter Fahrerassistenzsysteme und autonomer Fahrfunktionen. Insbesondere Echtzeitoptimierung und Test sind hierbei mit enormem Aufwand und Risiko verbunden. Ein ganzheitlicher, flexibel konfigurierbarer, echtzeitfähiger Prüfstand für das gesamte Fahrzeug würde hier Abhilfe schaffen. Der folgende Beitrag beschreibt das Konzept des ganzheitlichen, hochflexibel konfigurierbaren Echtzeit-Prüfsystems für intelligente Fahrzeuge in kooperierenden cyber-physischen Verkehrssystemen (ERAGON), welches gerade an der Ostfalia Hochschule für angewandte Wissenschaften aufgebaut wird.

## Einleitung

Die Mobilität befindet sich durch die zunehmende Digitalisierung und Vernetzung von Fahrzeugen in einem disruptiven Wandel. Der autonome Fahrbetrieb von Elektro-Hybridfahrzeugen in hochvernetzten cyber-physischen Systemen (CPS) ist eine der Kerntechnologien im digitalen Transformationsprozess der Mobilität. Die Nutzungsvielfalt autonomer Fahrzeuge erfordert dabei immer vielfältigere Sensorik sowie immer komplexere und intelligentere Algorithmen aus den Bereichen der modernen Regelungstechnik und der Künstliche Intelligenz (KI). Daraus resultieren Systeme, die noch umfangreicher und komplexer sind als die bereits vorhandenen hochgradig vernetzten elektronischen Fahrzeugfunktionen [1].

Die Entwicklung solcher Systeme ist hoch komplex und erfordert fachübergreifende, interdisziplinäre Entwurfsprozesse wie die bewährte Rapid Control Prototyping (RCP-)Methodik aus der Mechatronik-Forschung. In einem Top-Down-Verfahren wird hierbei zunächst die Systemkomplexität durch Strukturierung unter Anwendung von Modularisierung und Hierarchisierung reduziert. Auf den oberen Hierarchieebenen des vernetzten mechatronischen Systems (VMS) sowie des autonomen mechatronischen Systems (AMS) sind hierbei zunehmend intelligente autonome Fahrfunktionen und Fahrerassistenzsysteme (engl. ADAS) wie das elektronische

Fahrzeugmanagement und eine intelligente, kooperative Zielführung verortet. Anschließend erfolgt in einem Bottom-Up-Prozess der modellbasierte Entwurf jedes einzelnen Teilsystems mit den Absicherungsprozessen Model-in-the-Loop (MiL), Software-in-the-Loop (SiL) und Hardware-in-the-Loop (HiL). Solch ein strukturiertes Vorgehen ist unabdingbar zur Auslegung und Absicherung vernetzter mechatronischer Systeme [2]. Um die integrierte Gesamtfunktionalität autonomer, vernetzter Fahrzeuge zu untersuchen ist daher ein Gesamtfahrzeugprüfstand, welcher das Gesamtsystem aus Fahrbahn, vernetzten Fahrzeugen und vernetzter Fahrumgebung genauestens abbildet und gleichzeitig die Sensorik des Fahrzeugs anregt unabdingbar.

Eine Auswertung des nachfolgenden Standes des Wissens und Forschung ergibt, dass zwar sehr viele Teillösungen zur zuverlässigen Absicherung für ADAS, hochautomatisierte und autonome Fahrfunktionen auf AMS- und VMS-Ebene existieren und diese auch kontinuierlich weiterentwickelt werden. Was allerdings fehlt, ist ein ganzheitliches Prüfsystem, welches eine flexible Konfiguration sowohl des Prüflings als auch der Prüfumgebung unter realistischen und reproduzierbaren Bedingungen auf allen Ebenen der mechatronischen Strukturierung ermöglicht.

Im vorliegenden Beitrag wird daher das Konzept des ganzheitlichen, hochflexibel konfigurierbaren Echtzeit-Prüfsystems für intelligente Fahrzeuge in kooperierenden cyber-physischen Verkehrssystemen (ERAGON) vorgestellt. Dieses befindet sich aktuell an der Ostfalia Hochschule im Aufbau.

## 1 Stand des Wissens

Nach [3] ist die valide Funktionsabsicherung eine große Herausforderung auf dem Weg zum autonomen Fahren. Es muss gewährleistet werden, dass die ausgelegten Funktionen im Gesamtsystem nachweislich in Hinblick

auf Ausgangsgüte und Missinterpretationswahrscheinlichkeit sicher sind [4]. Es ist notwendig möglichst viele Situationen, mit denen das Fahrzeug konfrontiert werden könnte, reproduzierbar zu testen. Möchte man diese Aufgabe unter Realbedingungen im Straßenverkehr vollziehen, so werden hunderte Millionen Testkilometer notwendig [5]. Es ist also sinnvoll, den Test unter Realbedingungen mit simulationsgestützten Verfahren zu ergänzen, sodass bestimmte, seltene Situationen trotzdem gefahrlos und reproduzierbar getestet werden können. Dazu ist ein komplexes echtzeitfähiges Prüfsystem notwendig, welches durch eine Mischform aus realen und simulativen System und Umgebungskomponenten möglichst realistische Prüfbedingungen erzeugt.

Zu diesem Zweck setzen Chen et. al. [6] eine integrierte Simulations- und Testplattform für selbstfahrende Fahrzeuge ein. Deren Plattform bietet die Möglichkeit ein reales Fahrzeug auf einem abgesperrten Testgelände zu testen. Die Besonderheit an deren Ansatz ist, dass die Sensorsignale (GPS, IMU, Lidar und Kamera) aus einem hochgenauen virtuellen Simulationsszenario stammen und von einem realen Steuergerät im Fahrzeug zu realen Fahrbefehlen verarbeitet werden. Im Testverlauf wird die Sensorik also vollständig virtuell abgebildet, wodurch ungewollte Aggregationseffekte auftreten können, obwohl ein reales Fahrzeug vorhanden ist [7]. Die entwickelten Fahrfunktionen lassen sich vorab auch ohne die realen Komponenten in der Simulationsumgebung in Offline- und Online-Simulationen testen. Die in Chen et. al. vorgestellte Plattform ist hardwareseitig nicht flexibel gestaltet. Sie ist an ein spezifisches Forschungsfahrzeug mit einem prototypischen Steuergerät mit festen Schnittstellen gebunden. Somit lassen sich mit diesem Aufbau weder unterlagerte noch auf V2X-Kommunikation basierende Funktionen absichern. Aufgrund der Hardware-Architektur ist die Plattform ebenso wenig dazu geeignet KI-Algorithmen zu berücksichtigen.

Die Vehicle-in-the-Loop Methodik zur Bewertung automatisierter Fahrfunktionen im virtuellen Verkehr aus Solmaz [8] beruht auf einem ganz ähnlichen Aufbau wie Chen. Auch hier wird ein reales Fahrzeug auf einem abgesperrten Testgelände in einer simulierten Umgebung eingesetzt. Der wesentliche Unterschied besteht jedoch in der Ausführung des Steuergeräts für die Fahrfunktionen. Im Gegensatz zu dem Steuergerät aus Chen, lässt die MicroAutoBox II aus Solmaz eine flexible Modifikation des Systems unter Test durch RCP zu. Dies ist jedoch auf die AMS-Ebene beschränkt. So sind verschiedene Al-

gorithmen, auch aus dem Bereich der KI, implementierbar sowie theoretisch auch zusätzliche Signale z. B. aus der V2X-Kommunikation integrierbar. Das Prüfsystem selbst ist jedoch weder flexibel konfigurierbar noch frei von Aggregationseffekten. Die Reproduzierbarkeit der Testszenarien ist bei Prüfsystemen auf realen Teststrecken stets fragwürdig, da sich trotz simulierter Sensorsignale beispielsweise das Reifenverhalten witterungs- und temperaturabhängig ändert.

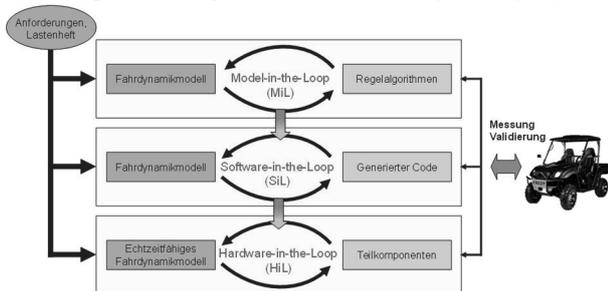
In Kanchwala [9] wird ein Echtzeit HiL-Fahrzeugsimulator vorgestellt, der ein reales Fahrzeug in einen stationären Versuchsaufbau einspannt. Hier werden Fahrwiderstände über vier Elektromotoren simuliert, die direkt mit den Rädern gekoppelt werden. Die zugehörigen Soll-Drehmomente werden von einem Simulationsrechner bestimmt, auf dem eine Simulation des Testszenarios erfolgt. Der Fokus dieses Systems liegt jedoch auf einer Untersuchung der längsdynamischen Fahreigenschaften, z.B. in virtuellen Offroad-Umgebungen. Untersuchungen des Lenkverhaltens oder Berücksichtigung von Sensor und Kommunikationssystemen sind nicht möglich. Die Flexibilität des Fahrzeugsimulators beschränkt sich auf den Test verschiedener Fahrzeuge. Eine flexible Konfiguration des Prüfaufbaus ist nicht vorgesehen.

Ying et al. [10] setzen zur Funktionsabsicherung autonomer Fahrzeuge eine Vehicle-in-the-Loop Simulations- und Testplattform ein. Der dazugehörige Gesamtfahrzeugprüfstand besitzt vier individuelle Anregungseinheiten, welche die Abbildung von drei Freiheitsgraden auf ein reales Fahrzeug ermöglicht. Die Sensoren des Fahrzeugs (Kamera, Lidar und Radar) werden mit Signalen stimuliert, die auf Basis virtueller Verkehrsszenarien generiert werden. Dadurch ermöglicht die Prüfumgebung reproduzierbare und vollständig kontrollierbare Testszenarien. Die Simulationsumgebung erlaubt die Einbindung von virtueller V2X-Kommunikation. Diese ist jedoch nicht in der Realität abgebildet. Somit ist keine vollständige Absicherung der Fahrfunktionen auf der VMS-Ebene möglich. Das zu prüfende Fahrzeug ist zudem nicht modifizierbar. Es ermöglicht zwar eine Absicherung verschiedener Fahrfunktionen in unterschiedlichen Prüfstands-Konfigurationen, jedoch kein Prototyping frei definierbarer Funktionen.

## 2 Methodik

Für die Entwicklung komplexer mechatronischer Systeme wird die durchgängig modellbasierte, verifikations-

orientierte RCP-Methodik eingesetzt. Mittels der mechatronischen Strukturierung wird das vernetzte CPS durch Modularisierung und Hierarchisierung in hierarchisch angeordnete Teilsysteme mit vier Hierarchieebenen geteilt: mechatronische Funktionsmodule (MFM), mechatronische Funktionsgruppen (MFG), autonome mechatronische Systeme (AMS) und vernetzte mechatronische Systeme (VMS) [11]. Ergebnis der Strukturierung ist eine funktionale Zergliederung des Gesamtsystems in gekapselte Module. Die hierarchisch angeordneten Module mit ihren Teilfunktionen besitzen eindeutig definierte physikalische und informationstechnische Schnittstellen in horizontaler und vertikaler Richtung und bilden die Basis für die spätere Integration zum Gesamtsystem. [12].



**Abbildung 1:** Durchgängig modellgestützter Funktionsentwicklungs- und Absicherungsprozess.

Nach der hierarchischen Strukturierung und Definition aller Schnittstellen erfolgt in einem Bottom-Up-Prozess die modellbasierte und funktionsorientierte Auslegung jedes einzelnen Moduls. Beginnend mit der untersten und zugleich vitalsten Ebene MFM erfolgt auf allen Hierarchieebenen anschließend die Integration zu höherwertigen Funktionen. Der Entwurf sowie deren Integration zum Gesamtsystem (mechatronische Komposition) erfolgt unter Anwendung des modellbasierten mechatronischen Entwicklungskreislaufs. Es folgen Model-in-the-Loop-Simulationen (MiL), in welchen Regelalgorithmen und Künstliche Intelligenz auf Basis eines physikalischen bzw. mathematischen Ersatzmodells entwickelt und an einem Fahrzeugmodell erprobt werden. Aus den simulativ erprobten Algorithmen wird im Rahmen der Software-in-the-Loop-Simulation (SiL) mittels automatischer Codegenerierung ein ausführbarer Programmcode erzeugt und in einem Virtuellen Prüfstand offline getestet. Anschließend erfolgt die Hardware-in-the-Loop-Simulation (HiL), bei welcher ein um physische Teilkomponenten ergänztes echtzeitfähiges Fahrzeugmodell zur Funktionsabsicherung und -optimierung der Algorithmen und intelligenten

Funktionen unter realistischen Echtzeitbedingungen online verwendet wird [13]. Abbildung 1 illustriert den Prozess.

### 3 Konzeption des Prüfstands

Das Echtzeit-Prüfsystem für intelligente Fahrzeuge in kooperierenden cyber-physischen Verkehrssystemen „ERAGON“ besteht hauptsächlich aus vier Modulen: Prüfstandsmodul, Echtzeitinformationsverarbeitung, Software- und MMI-Kommunikationsmodul. Hierdurch ist das System flexibel konfigurierbar und für verschiedene Prüflinge einsetzbar. Abbildung 2 verdeutlicht das Zusammenspiel.

Alle unterlagerten Prüfsysteme sind über definierte Schnittstellen hierarchisch angeordnet. Somit ist das „ERAGON“ in der Lage für unterschiedliche Untersuchungszwecke in verschiedenen Varianten flexibel verwendet werden zu können.

#### 3.1 Prüfstandsmodul

Das Prüfstandsmodul besteht im Wesentlichen aus einer komplexen Anregungseinheit zur Simulation und Stimulation eines Prüflings durch das simulierte Umfeld. Eine umfangreiche Umgebungssimulation dient zur Erzeugung einer virtuellen 3D-Welt und zur Repräsentation der realen Objekte in der Umgebung des Fahrzeugs. Die Sensoren können auf unterschiedliche Weise mit den Daten aus der Simulation angeregt werden. Eine Möglichkeit ist Over-The-Air mittels physikalischen Anregungsgrößen die errechneten Hindernisse einzuprägen, z.B. bei Ultraschallsystemen mittels Ultraschallwellen über eine Schwingmembran. Ein weiterer Kanal zur Einspeisung von Umfeldsignalen ist eine zusätzliche V2X-Entwicklungsplattform für einfachen Zugang zur V2X-Kommunikation, sodass keine Implementierung der spezifischen Kommunikationsprotokolle und Softwareschichten erforderlich sind.

Mit o. g. Systemen wird eine umfangreiche Grundlage der Implementierung und Ankopplung einer umfassenden Verkehrssimulation gewährleistet. Das Kernmodul kann damit ein vollständiges Verkehrsszenario reproduzierbar und realitätsgetreu simulieren. Diese hohe Augmentation von Realität und Simulation ist für das Testen von höherwertigen autonomen und vernetzten Funktionen auf der Strukturebene AMS und VMS unablässig und macht einen umfassenden Test der Fahrfunktion in einer CPS-Verkehrsumgebung in den Strukturebenen AMS / VMS erst möglich.

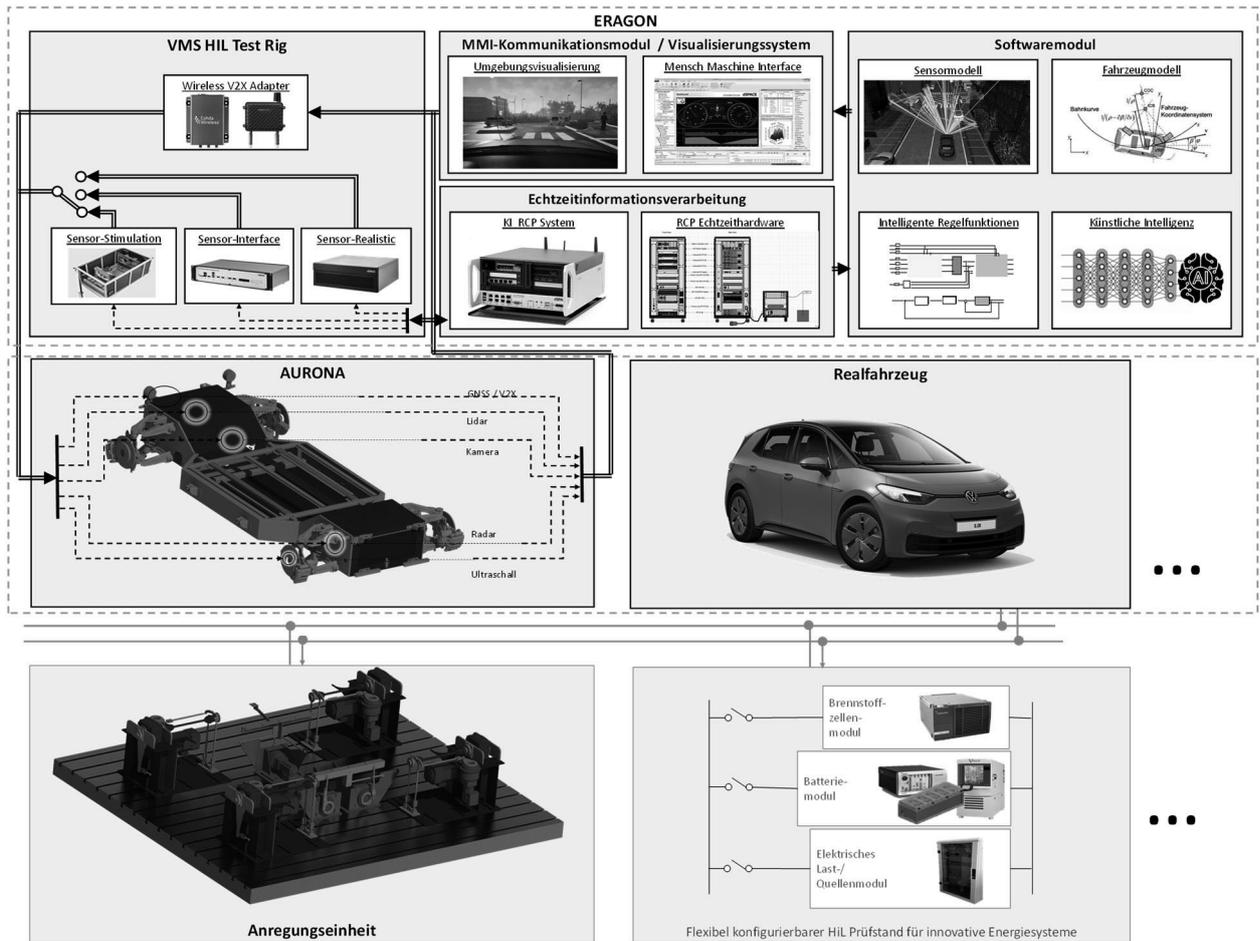


Abbildung 2: Konzept des ganzheitlichen Prüfsystems ERAGON.

### 3.2 Softwaremodul und MMI

Das Softwaremodul beinhaltet sämtliche zu testende Reglerfunktionen und virtuellen Systeme wie die Sensor- und Fahrzeugmodelle. Zusätzlich erfolgt hier die Kalibrierung, Skalierung und Signalkonditionierung. Das Softwaremodul wird als verteilte Berechnung auf mehreren Prozessoren des Echtzeitinformationsverarbeitungsmoduls unter Echtzeitbedingungen digital berechnet und verarbeitet. Zur Applikation und Kommunikation dient das Kommunikationsmodul mit Mensch-Maschine-Interface, welches auch zur Visualisierung dient. Hierfür sind entsprechende Benutzeroberflächen (GUIs) vorgesehen. Über das RTI (Real-Time Interface) der Target-Hardware erfolgt die Ansteuerung bzw. Automatisierung des Prüfstands.

### 3.3 Echtzeitinformationsverarbeitung

Das Modul der Echtzeitinformationsverarbeitung mit seinen RCP-Systemen stellt die Informationsverarbeitung sicher. Das Modul ist mit RCP-Systemen einschließlich

KI-RCP-System zum schnellen Prototyping von intelligenten Regelfunktionen, zum maschinellen Lernen von Künstlicher Intelligenz mittels leistungsfähiger Prozessortechnologie für anspruchsvolle Echtzeit-anforderungen ausgerüstet und verfügt über ein schnelles Datenverarbeitungs- und Speichersystem zum Speichern und Verwalten großer Datenmengen.

### 3.4 Prüflinge und Versuchsfahrzeug

Der Prüfstand ist für einen flexiblen Einsatz konzipiert. Hierfür sind definierte Schnittstellen zu verschiedenen Prüflingen vorgesehen. Hier können beispielsweise reale Fahrzeuge, Funktionsträger oder spezielle Versuchsaufbauten wie ein Einspurmodell eingesetzt werden. Initial wird ein Autonomer rekonfigurierbarer Funktionsträger für nachhaltige Mobilität (AURONA) als Prüfling vorgesehen. Als ein RCP-Funktionsträger ist AURONA mit allen aktuellen für die hochautomatisierten bzw. autonomen Fahrfunktionen notwendigen Umfeldsensoren (Kamera, Radar, Ultraschall, Lidar, etc.) ausgestattet, deren Schnittstellen an verschiedenen Stellen geöffnet werden.

So können Hardwarebestandteile des sensorischen Systems über Bypassing umgangen werden. Die Schnittstellen sind so konfiguriert, dass eine flexibel konfigurierbare Anregung der Sensoren über den „ERAGON“ gewählt werden kann. Es ist sowohl möglich über die Sensorstimulation den physikalischen Sensoreingang anzuregen als auch direkt die Elektronik desselben Sensors mit Rohdaten zu bespielen.

Weiterhin ist AURONA mit einer 5G-basierten V2X Kommunikationseinheit ausgestattet über die es mit der V2X-Entwicklungsplattform des „ERAGON“ angeregt, aber auch mit anderen V2X-Einheiten bidirektional kommunizieren kann. Somit können auch kooperative Fahr-funktionen untersucht werden. Das „AURONA“ verfügt zudem über ein Antriebs- und Lenksystem, welches ein Testen des Einflusses der aktorischen Elemente eines autonomen Fahrzeugs ermöglicht. Es ist somit in der Lage Stellbefehle vollumfänglich realistisch umzusetzen. Diese werden über die Anregungseinheit des „ERAGON“ gegenaktuiert, wodurch sich ein realistisches und geschlossenes Fahrverhalten des „AURONA“ oder auch eines anderen Prüflings ergibt.

## 4 Zusammenfassung

Im vorliegenden Beitrag wurde das Konzept eines ganzheitlichen, hochflexibel konfigurierbaren HiL-Prüfsystems „ERAGON“ für den Test autonomer Fahrfunktionen vorgestellt. Dieses ist in einer Closed-Loop zusammen mit dem Funktionsträger „AURONA“ dazu in der Lage die gesamte Wirkkette eines autonomen Fahrzeugs ausgehend von der Infusion von Sensorrohdaten über das Entwickeln und Testen von KI-Funktionen bis zur Anregung von realistischen Fahrsituationen zu simulieren und zu stimulieren.

### Danksagung

Gefördert als Teil der Reaktion der EU auf die Covid-19-Pandemie. Die Finanzierung erfolgte aus REACT-EU Mitteln des EFRE unter dem Förderkennzeichen ZW 7-85195402. Die Verantwortung für den Inhalt liegt bei den Autoren.



EUROPÄISCHE UNION  
Europäischer Fonds für  
regionale Entwicklung

Als Teil der Reaktion  
der EU auf die  
Covid-19-Pandemie  
finanziert



## Literatur

- [1] Liu-Henke, X.; Yarom, O.; Scherler, S.: *Virtual Development and Validation of a Function for an Automated Lateral Control using Artificial Neural Networks and Genetic Algorithms*. 91st IEEE Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, May 25 - 28, 2020
- [2] 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 (ICSOFT), Online, July 6-8, 2021
- [3] Aeberhard, M.; et al., *Experience, Results and Lessons Learned from Automated Driving on Germany's Highways*, in IEEE Intelligent Transportation Systems Magazine, vol. 7, no. 1, pp. 42-57, 2015.
- [4] Mitsch, S.; Ghorbal, K.; Platzer, A.: *On Provably Safe Obstacle Avoidance for Autonomous Robotic Ground Vehicles*. Robotics Science and Systems (RSS), 2013.
- [5] Wachenfeld, W.; Winner H.: *Die Freigabe des autonomen Fahrens*. Technische Universität Darmstadt. in M. Maurer et al. (Hrsg.), *Autonomes Fahren*, 2015, DOI 10.1007/978-3-662-45854-9\_28
- [6] Chen, S.; Chen, Y.; Zhang S.; Zheng, N.: *A Novel Integrated Simulation and Testing Platform for Self-Driving Cars with Hardware in the Loop*. in IEEE Transactions on Intelligent Vehicles, vol. 4, no. 3, pp. 425-436, Sept. 2019, doi: 10.1109/TIV.2019.2919470
- [7] Jacobitz, S.; Göllner, M; Zhang, J; Yarom, O. A.; Liu-Henke, X.: *Seamless Validation of Cyber-Physical Systems Under Real-Time Conditions by Using a Cyber-Physical Laboratory Test Field*. IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE), Online, December 12-14, 2021.
- [8] Solmaz, S.; Rudigier, M.; Mischinger, M.: *A Vehicle-in-the-Loop Methodology for Evaluating Automated Driving Functions in Virtual Traffic*. 2020 IEEE Intelligent Vehicles Symposium (IV), 2020, pp. 1465-1471, doi: 10.1109/IV47402.2020.9304811.
- [9] Kanchwala, H.; Dhillon, J. S.: *A real-time hardware-in-the-loop vehicle simulator*. 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), 2020, pp. 182-187, DOI: 10.1109/INDIN45582.2020.9442138
- [10] Ying, G.; Zhao, X.; Xu, Z.; Cheng J.; Wang, W.: *An Indoor Vehicle-in-the-Loop Simulation Platform Testing Method for Autonomous Emergency Braking*. April 2021, Journal of Advanced Transportation 2021(3):1-12, DOI: 10.1155/2021/8872889
- [11] Liu-Henke, S.; Scherler, S.; Göllner, M.: *System Architecture of a full active Autonomous Electric Vehicle*. IEEE Transportation Electrification Conference and Expo (ITEC), Chicago, IL, USA, June 22 - 24, 2017
- [12] Liu-Henke, X.: *Mechatronische Entwicklung der aktiven*

*Feder-/Neigetechnik für das Schienenfahrzeug RailCab.*  
VDI-Fortschritt-Berichte, Reihe 12, Nr. 589, VDI-Ver-  
lage. Düsseldorf, 2004

- [13] Jacobitz, S.; Liu-Henke, X.: *A Real-Time Interface for Xcos – Demonstrated on a Battery-management System.* 2nd Scilab Conference, Berlin, Germany, October 14, 2019.

# Towards a Scenario Toolkit for Autonomous Systems

Ahmad Naja<sup>1\*</sup>, Siddhartha Gupta<sup>1</sup>, Umut Durak<sup>2</sup>, Sven Hartmann<sup>1</sup>

<sup>1</sup>Institute for Computer Science, TU Clausthal, Clausthal-Zellerfeld, Germany; \* [ahmad.naja@tu-clausthal.de](mailto:ahmad.naja@tu-clausthal.de)

<sup>2</sup>German Aerospace Center (DLR), Braunschweig, Germany; [umut.durak@dlr.de](mailto:umut.durak@dlr.de)

**Abstract.** Scenario-based approaches have recently been widely adopted in the automotive and aviation industries. They aim to define and manage the test cases in a better way, thereby significantly reducing the risks and defining the safety argument for the system. To improve and complement the safety conditions, the Operational Design Domain (ODD) defines the operational boundaries of a driving automation system to specify the scope of the safety case, represented by human and machine-readable language. Many approaches and standards that involve essential modelling and safety concepts have been introduced to represent scenario-based simulation. In addition, some parts of these approaches were implemented as graphical tools, such as System Entity Structure (SES) and Pruned Entity Structure (PES) tools, which are based on an ontology and its derived structure. However, an extensive implementation covering scenario modelling and management, ODD, and assessment is still missing. This paper proposes an adapted scenario-based approach based on the related research and implements it using a robust GUI tool called Operational Domain Modeling Environment (ODME). ODME is progressing to fill the gap by covering the modelling functions and safety approaches in one comprehensive environment with a wide range of capabilities and features.

## Introduction

The subject of modeling aims to represent a certain aspect of reality for a particular purpose. Systems, processes, and phenomena can all be represented by models. Afterward, the system behaviors can be generated through the simulation process using the model [1]. Creating and developing models is considered as an initial step to simulate and test these systems based on different use cases, called scenarios.

A scenario describes a system's behavior based on its operating conditions and situations, changing of its

parameters through time, and the mutual interaction of its components with each other [2]. Scenario modelling helps researchers better plan and lower risks by exploring a variety of prospective outcomes, comparing them using specific standard metrics, and testing decisions<sup>i</sup>. Developing a scenario goes through significant steps, starting by defining the scenario by the stakeholders and finishing by generating the necessary executable specifications. Scenarios as executable specifications are input to the simulation environment [3].

One of the biggest problems facing the car industry is ensuring the safety of autonomous vehicles. Scenario-based development and test methods are potential methods for testing and validating autonomous driving capabilities [4]. The scenario-generating process of the scenario-based approach carries forward a safety argument and is crucial for the system's release. So, the scenarios must be created and documented methodically. In addition, they have to be traceable throughout the development process [4]. Technology firms and automakers use the Operational Design Domain (ODD) concept to specify the safe operating conditions for their Automated Driving Systems (ADS). An ODD establishes where the ADS are intended to function correctly by definition [5]. Therefore, scenarios and ODD complement each other to provide a safety argument for an autonomous vehicle.

Organizations and researchers develop many standards to represent the scenario computationally, such as OpenSCENARIO from the Association for Standardization of Automation and Measuring Systems (ASAM). ASAM played a lead role in setting standards that cover the whole simulation pipeline. Figure 1 shows a simulation process workflow based on a family of ASAM standards, which illustrates the different steps a scenario goes through, beginning with a concept description and ending with testing results.

<sup>i</sup><https://www.synario.com/scenario-modeling-what-you-need-to-know/>

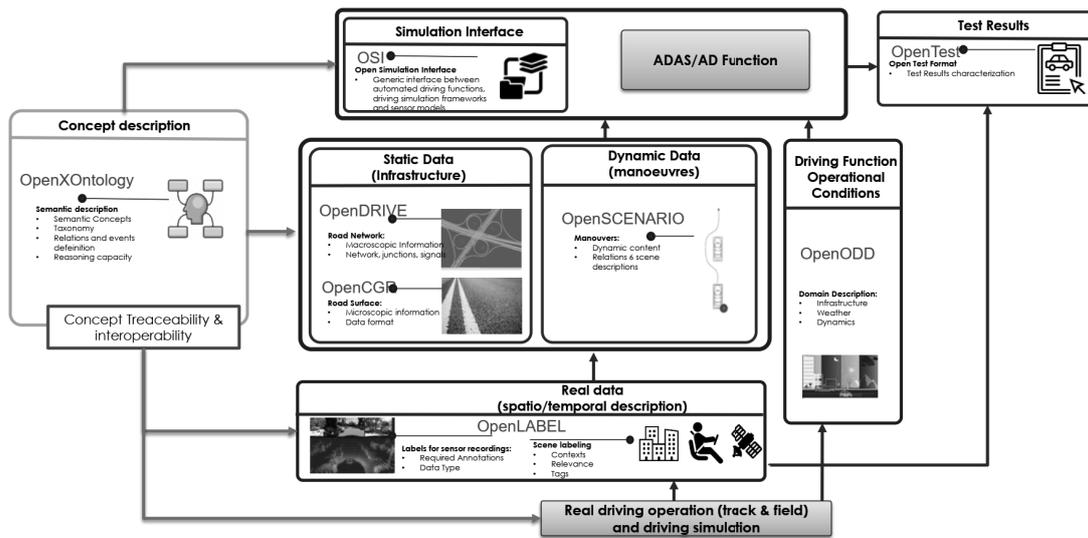


Figure 1: Simulation Process Workflow Based on a Family of ASAM Standards [6]

OpenSCENARIO is the point of interest used for driving simulation and virtual development, testing and validation of driving assistance functions, automated and autonomous driving <sup>ii</sup>.

There are some concerns with the approach illustrated in Figure 1. Many standards are still under development and have not been shown to be compatible with each other. Many third-party tools have been shown to use one of these standards, mostly focusing on OpenSCENARIO, but none can handle the complete workflow. A more integrated and compact workflow is needed along with its associated tool-set.

This paper provides new insights to define an adapted approach based on the ASAM simulation workflow in Figure 1 and other research work. This adapted approach will compact the simulation process based on the scenario modelling, scenario generation, management and ODD definition. Furthermore, a modelling environment that implements the new approach will be introduced, called the Operational Domain Modeling Environment (ODME). The presented model-based scenario development employs System Entity Structures (SES) based metamodeling that offers the development of models and the generation of executable software entities with successive model transformations through technical spaces [7]. The SES and its related project, Pruned Entity Structures (PES) tools presented in [8], was a starting point for achieving the suggested goals.

After refactoring, it will be ready to expand and implement the scenario-based approach.

The paper begins by giving a background in section I about SES, scenario representation, Operational Design Domain, ASAM Standards and SES and PES Tools. In section II, the newly suggested approach will be introduced and illustrated. Section III will implement the theoretical approach in a practical modelling environment. Finally, section IV gives a short conclusion and shows some recommended views for future work.

## 1 Background

### 1.1 System Entity Structure (SES)

SES is described as a framework for knowledge representation of system coupling, decomposition, and taxonomy [9] and is considered an enhancement in the discipline of system theory-based approaches to modelling and simulation [10]. Utilizing interactions between decomposition, coupling, and taxonomies allows for the succinct specification of a family of models [11]. SES is further described as a formal ontology framework specifying a system's components and hierarchical relationships [12]. Figure 2 shows SES nodes and relationships and a simple SES tree example.

An SES is represented as a labelled tree. The entity, Aspect, Specialization, and Multi Aspect are the four different types of nodes [14].

<sup>ii</sup> [www.asam.net/standards/detail/openscenario/](http://www.asam.net/standards/detail/openscenario/)

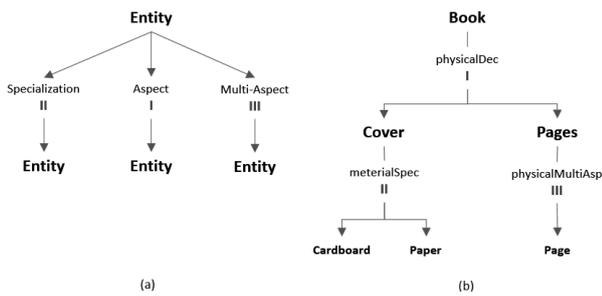


Figure 2: (a) SES Nodes and Relationships, [12] and (b) an Example.[13]

- **Entity:** It depicts system elements that are artificial or real. It is an object of interest that may have variables linked to it. Other node types are utilized to describe parent and child entities.
- **Aspect:** It indicates the decomposition relationship of an Entity node. It represents processes for breaking down larger objects into more fine-grained ones.
- **Specialization:** It represents an Entity's taxonomy. Specialization denotes groups or families of distinct forms that an object may take.
- **Multi Aspect:** This particular type of aspect describes a multiplicity connection and shows that the parent entity is composed of many entities of the same type.

Several axioms define the SES as well, which are: uniformity, strict hierarchy, alternating mode, valid brothers, attached variables, and inheritance [15].

Given its foundations in the theory of modelling and simulation and its expressive strength and clarity with only a few axioms, SES is suggested as the foundation of the proposed intermediate metamodel in the Simulation Model package. Thus, SES is appropriate as a straightforward intermediate metamodel that establishes a formal framework that is clear and understandable [16].

## 1.2 Scenario Computational Representation

The machine-readable format for SES and Pruned Entity Structure (PES) is called computational representation. There are two significant operations in the conceptual space, as shown in Figure 3.

An SES Ontology—a specific SES—is constructed utilizing the constructs, structure, and axioms of SES.

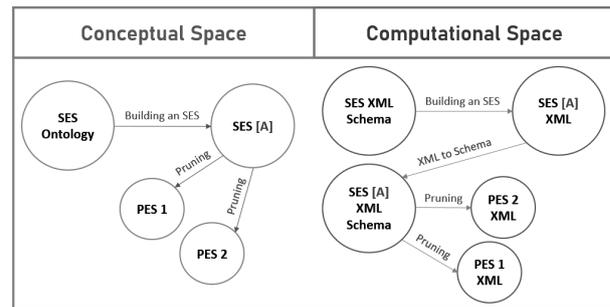


Figure 3: Computational Representation [17]

The pruning process generates the Pruned Entity Structures (PESs) from this specific SES. According to [18], we may express the SES ontology with an XML Schema and the specific SES as an XML file in the computational space. Then, they proposed creating an XML that specifies a particular SES to an XML Schema. The construction and validation of PESs during pruning finally employ this schema. PESs ultimately become XML instances. The schema for the SES ontology could be defined using the XML Schema Definition Language (XSD). An XML document's restrictions and structure may be described using XSD [19].

## 1.3 Operational Design Domain (ODD)

SAE J3016 defines the Operational Design Domain (ODD) for a driving automation system as "Operating conditions under which a given driving automation system, or feature thereof, is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and the requisite presence or absence of certain traffic or roadway characteristics." In short, the ODD establishes the limits that the driving automation system is intended to work within and, as a result, will only function when these criteria are met, as shown in Figure 4.

The ODD restricts where the automated driving system (ADS) is valid and thus confines the scope of the safety case and the verification. Use cases are required to give a strategy for a set of operating conditions (OCs) and verify that it remains within the range of the ODD

ODD can control the coverage of scenarios and provide a list of operation conditions, which can be used later as an effective input for the system assessment process. Some requirements are important when ODD is

<sup>iii</sup> [www.claytex.com/tech-blog/](http://www.claytex.com/tech-blog/)

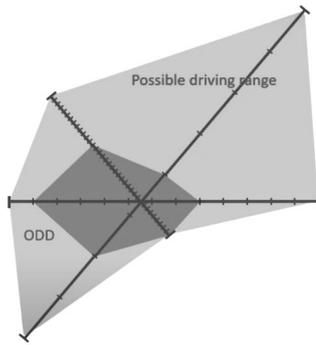


Figure 4: Operational Design Domain (ODD) [20]

defined. For instance, ODD should be generated in a format understandable for humans, such as tables. On the other hand, it has to be machine-readable, such as an XML file.

#### 1.4 ASAM Standards

In the automobile sector, development and test tool-chains can be typically built to ASAM standards, which are public specifications. The use of ASAM standards is optional. To determine the Standard-compliance of products, ASAM advises and promotes best practices. Throughout vehicle development, ASAM standards specify interfaces, protocols, file formats, and data models. ASAM-based tools and products provide seamless data exchange and simple integration into already-existing value chains<sup>iv</sup>.

ASAM provides a family of standards for the simulation domain, which have repeatedly proven themselves in various development processes. These standards must interact to generate a global view of the simulation process. In the following, some related ASAM standards are mentioned [21]:

- **OpenXOntology:** An ontology-based framework for notions like roads, lanes, and traffic participants is provided by OpenXOntology. The ASAM OpenXOntology comprises several interconnected components, including core, domain, and application ontologies.
- **OpenODD:** For connected autonomous cars (CAVs), OpenODD seeks to create a format that

may describe a specified Operational Design Domain. The ODD defines the functional requirements for connected autonomous vehicles and outlines the environmental characteristics that CAVs must be able to control.

- **OpenSCENARIO:** The dynamic content of the world is defined by OpenSCENARIO, for instance, the anticipated behaviour of traffic participants and how they should interact with one another and their environment.
- **OpenDRIVE:** The primary goal of OpenDRIVE is to offer a description of the road network that can be used as input into simulations to create and verify advanced driver assistance systems (ADAS) and Autonomous driving features.

#### 1.5 SES and PES Tools

Initial scenario workflow using SES[22] was implemented using SES tools. PES Tools provided the pruning capabilities for the tool. It added to and even finished work outlining a formal strategy for creating a scenario specification language [23]. Figure 5 shows the workflow of SES-PES projects.

In SES Tools, The user may access a wide variety of widgets using the graphical user interface that helps with modelling. It uses a collection of elements and axioms to describe knowledge about system connection, taxonomy, and decomposition [24]. While the SES model developed is pruned using PES Tools, several scenarios are produced. Pruning is a technique that creates a distinct system structure from a domain model; the outcome is known as a Pruned Entity Structure(PES).

An SES Model represents a family of models for a certain application domain. All of a system's potential configurations are taken into account when using SES modelling. An SES model tree has to be trimmed to get a specific configuration. To achieve this configuration, a domain model tree that is a selection-free tree, pruning removes extraneous structure based on the definition of a realistic frame. A domain model is often reduced by pruning by eliminating options for an entity with numerous attributes and specializations made up of several entities. A domain model tree may be pruned by giving the values of the variables, choosing one entity from the available specialization node possibilities, and indicating the cardinality in a Multi Aspect node [8].

<sup>iv</sup> <https://www.asam.net/standards/standard-compliance/>

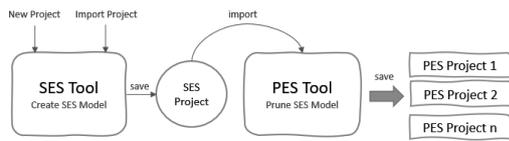


Figure 5: SES and PES Projects Structure

Although the tool-set provided the modelling and simulation community with many advantages, it is currently not scalable and requires several enhancements. Using two tools and managing the pruned models was a critical problem that made the project complicated and hard to use.

## 2 General Scenario-based Approach for Autonomous Systems

The proposed scenario-based approach uses the fundamental elements of many works involving scenarios such as the ASAM workflow [6] and other research works like [25], [26], integrating many features of scenario simulation principles.

The simulation model process in the proposed approach goes through a sequence of fundamental steps, from knowledge acquisition to finding the best scenario to be executed. The domain model should be created depending on some knowledge generated by experts or raw data. A domain model defines an abstract representation of all the elements of the simulated system, so it needs to be pruned to produce a particular use case or scenario. Scenario Manager aims to organize the created scenarios and their attributes and help to arrange scenarios based on different metrics. Afterward, the selected scenario will be executed physically or virtually to be tested and assessed. Finally, the metrics will be evaluated to send feedback to the domain model to be improved. Figure 6 illustrates the suggested scenario-based approach.

**Knowledge Source.** A common way to generate relevant scenarios is to use the knowledge of domain experts and formulate the scenarios manually. Data from real driving situations could be another input for scenario modelling (not the focus of our current approach).

**Metamodel (SES Ontology.)** An ontology offers a vocabulary for a specific domain by computerizing the

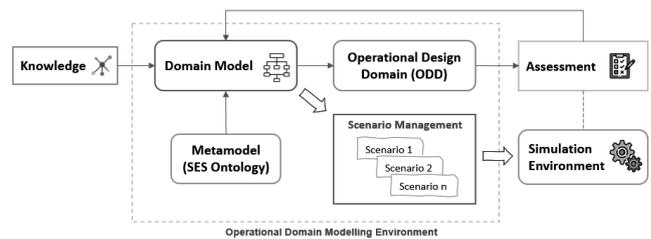


Figure 6: Proposed Scenario-based Approach Workflow

specification of the meaning of definitions and describing the concepts and relationships that are significant in a given domain. As ontologies explain domain relationships and entities in a simple and machine-interpretable way, they serve as a bridge between humans and computer systems. Model transformations are offered as an automated way to produce an executable scenario definition, whereas metamodeling is proposed as a technique to develop a graphical modelling language. To include all the components of a scenario that may be simulated in autonomous cars, SES is used for metamodeling [23]. Upon this metamodel, a scenario modeling approach is developed.

**Domain Modelling.** In this step, the knowledge will be used to create a model based on SES ontologies that describe the system in human-readable representation. This model will be considered an abstraction of all possible scenarios, so it needs to be pruned later to generate derived models or individual scenarios. Moreover, the variables and constraints used in the model will be fundamental to defining ODD.

**Scenario Management.** The created domain model will be derived in different scenarios by the pruning process. However, the generated scenarios need a mechanism to be organized. The scenario manager aims to achieve this goal by developing a method to facilitate storing and managing the scenarios. Scenarios also need to have certain metrics associated with them.

**Define Operational Design Domain (ODD).** In this block, the operating conditions that form the system's boundary will be defined, as shown in Table 1. These operational conditions will be used as input for the assessment process.

Value Range	Aspect
Pedestrian	sporadically
Road Types	highway
Time of Day	any
Speed Range	$\leq 130$ kph
Visibility	$\geq 40$ m

Table 1: ODD Table Example

**Scenario Execution.** Once the scenario test cases have been identified, they must be executed on real physical systems or simulation tools. During the execution, the results should be recorded and stored to be analyzed and evaluated. This step is essential to specify the safety properties and is required to develop executable specifications.

**Assessment Process.** Evaluating the quality and the performance characteristics, such as safety and efficiency, is crucial in developing autonomous systems. In this block, the results and remarks of execution will be processed and analyzed to evaluate the scenario, hazard analysis and risk assessment. After that, The scenario's metrics will be updated and sent as feedback to the domain modelling block.

### 3 Operational Domain Modeling Environment (ODME)

ODME is a robust environment that contains all functionalities the user needs to create and prune models and manage pruned scenarios via the scenario manager. It is a successor to the SES tools implementing the functionalities contained in the dotted line in the proposed scenario-based approach in Figure 6. Figure 7 shows the general view of the ODME main window in domain modelling mode.

Developing the ODME project went through four important levels:

1. **Planning:** Before starting with implementation, many related projects and research were reviewed to collect ideas and create a special innovative approach. The SES and PES Tools project was also chosen as a base and starting point for ODME.
2. **Reconstruction:** two important steps were implemented in this level:

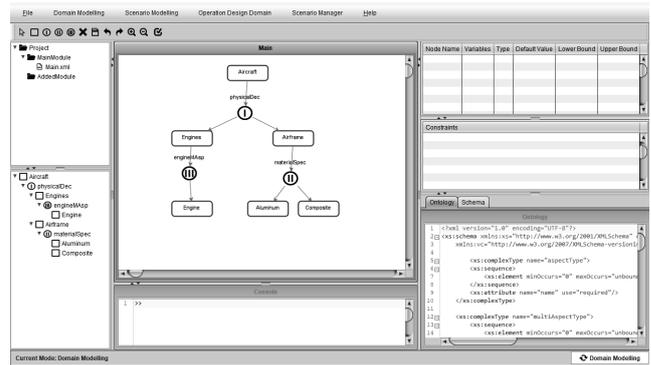


Figure 7: ODME Main Window in Domain Modelling Mode

- **Refactoring Process:** cleaning the code, fixing bugs, and improving the project's structure.
- **Build Mode Switch mechanism:** create a smooth and simple solution to change between Domain Modelling and Scenario Modeling modes after merging SES and PES tools.

3. **Upgrade:** We added many new features to ODME, such as importing/exporting the model as a template and saving the model as a PNG file, in addition to some improvements in GUI design. The essential new feature in ODME is the scenario manager, which allows the user to create multi-scenario models for one domain model and delete and change the metrics of scenarios. The scenario files will be managed automatically and stored in separate folders that can be moved to other devices, making the created projects portable.
4. **Test and Documentation:** by using the new environment to create real use case scenarios, documenting, and suggesting more advanced features for future work.

As mentioned before, ODME consists of two modelling modes, domain and scenario modelling, as well as the scenario manager.

#### 3.1 Domain Modelling Mode

ODME facilitates the user creating a domain model as a visual representation of real situation objects by a wide set of widgets and features. The components in the domain of the problem, and the connections between them, are represented by the Domain Model. All system entity structure components introduced in section

[II] are supported. ODME added many new features which help to improve usability and make the environment reliable and user-friendly.

While ODME starts, the main window will appear in Domain Modelling mode, and a new project called Main will be generated automatically with a root node. In the Drawing Panel, the user can create a domain model using the different types of nodes in the Tool-Bar. While a node is connected to the graph model, the Synchronized Tree Window will be updated to show it. When the model is saved, the ontology and schema will be generated. Figure 8 shows a simple domain model created by ODME.

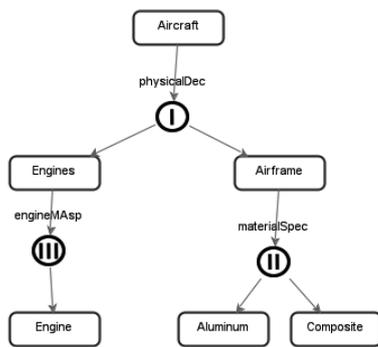


Figure 8: Domain Model Example

### 3.2 Scenario Modelling Mode

After creating a domain model, ODME provides a simple mode switch mechanism, which saves the domain model and forwards it to scenario modelling mode. In this mode, the model can only be pruned to create scenarios. The nodes which are eligible for pruning will be highlighted.

ODME support three different types of pruning [10]:

- **Multi Aspect Node Pruning** Before pruning, a Multi Aspect node must define its cardinality or a total number of aspects. Cardinality is assessed, and a certain number of aspects of the same kind are formed when a Multi Aspect node is pruned. Figure 9 shows the result after pruning the "engine-MAsp" entity in Figure 8. Based on the cardinality number, which will be defined by the user (Three, for example), three engines will be generated after pruning.
- **Specialization Node Pruning** One child must be chosen to create a valid variation per the special-

ization requirement. Figure 8 demonstrates that the entity Airframe has two options: composite and aluminum. The entity that remains after pruning can, therefore, either be of type Aluminum or type composite. In Figure 9, the completed pruned entity Composite Airframe will be generated.

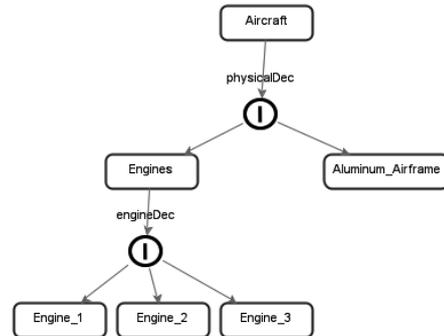


Figure 9: Scenario Model Example

- **Entity variable Pruning** Entities which have variables can only be pruned. So, they will be highlighted with green in scenario modelling mode. This pruning can be achieved by updating the value of an entity's variables. After pruning the entity variable, the variable table will also be updated with the new value. Figure 10 shows the variable table of the "Node" entity after pruning of "Var\_2".

Node Name	Variables	Type	Default Val...	Lower Bou...	Upper Bou...
Node	Var_2	int	5	0	10
Node	Var_3	string	none		
Node	Var_1	boolean	none		

Figure 10: Entity Variables Table after Pruning

### 3.3 Scenario Management

The domain model can be pruned to have many scenarios. The number of scenarios generated can extend to a large number, as the variability factors are many. It requires a mechanism to organize scenarios by labelling them, and prioritizing them before exporting and executing them. Scenario manager is one of the new features in ODME used to manage the scenarios created by the tool. The following points can summarize the main goals of a scenario manager:

- Capturing all the scenarios and creating a scenarios list to organize them, in addition, to helping to access any scenario.

- Assigning criteria as classification metrics to the scenarios manually.
- Providing run-time scripts to execute the scenario directly and show the result by linking the scenarios to the simulation environment, such as Matlab and Gazebo.
- Having the possibility of a mechanism to accept feedback from the assessment and adjust the scenarios accordingly.

The last two points are still under development. Figure 11 illustrates the main functionalities of the scenario manager.

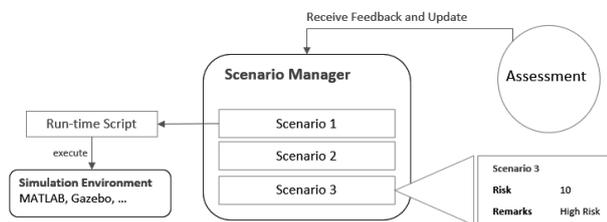


Figure 11: Scenario Manager

Scenarios List is a simple functionality in the tool to switch between scenarios, delete scenarios and change their metrics. By double-clicking on one of the scenarios, the metrics update window will be opened, where the user can update the risk value of the scenario in addition to writing some remarks. Figure 12 shows the scenarios list.

Name	Risk	Remarks
InitScenario		
Scenario2	10	High Risk
Scenario3	2	Low Risk

Figure 12: Scenario Manager List

## 4 Conclusion and Future Work

In this paper, we proposed a scenario-based approach using SES ontology, which integrates many fundamental simulation concepts, like domain modelling, scenario management, and operational design domain. Furthermore, we implemented this approach in a robust tool called ODME. Nevertheless, ODME is still an active project open to many ideas and improvements. There are several directions for future work, aiming to

implement ODD in our tool so that it could be used for the assessment process. In addition, the scenario manager needs to be optimized to compare different scenarios. The future work also includes developing a test data generator using machine learning algorithms. This data can help test a broader range of scenarios and define the operational conditions more precisely.

## References

- [1] P. K. Davis and R. H. Anderson. Improving the composability of dod models and simulations. pages vol. 1, no. 1, pp. 517. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, 2004.
- [2] D. A. Kononov, V.V. Kulba, S.S. Kovalevsky, and S.A. Kosjachenko. Development of scenario spaces and the analysis of dynamics of behaviour of social and economic system. Preprint, 1999.
- [3] Umut Durak, Okan Topçu, Robert Siegfried, and Halit Oğuztüün. Scenario development: A model-driven engineering perspective. 2014.
- [4] Till Menzel, Gerrit Bagschik, Leon Isensee, Andre Schomburg, and Markus Maurer. From functional to logical scenarios: Detailing a keyword-based scenario description for execution in a simulation environment. 2019.
- [5] S. O.-R. A. D. Committee et al. Sae j3016. taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. tech. rep., 2016.
- [6] Oihana Otaegui. Standardisation on interfaces and formats for ccam validation. connected and automated driving virtual conference, April 2021.
- [7] D. Gasevic, D. Djuric, and Devedic V. Model driven engineering and ontology development. Springer, 2009.
- [8] Bikash Chandra Karmokar. Application agnostic ses modeling environment and interactive pruning tool. TU Clausthal.
- [9] T. Kim, C. Lee, E. Christensen, and et al. System entity structuring and model base management. pages 20: 1013–1024. IEEE Trans Syst Man Cybern, 1990.

- [10] T. Ören and B. Zeigler. System theoretic foundations of modeling and simulation: a historic perspective and the legacy of a wayne wymore. pages 88: 1033–1046, 2012.
- [11] J. Rozenblit and B. Zeigler. Representing and construction of system specifications using the system entity structure concepts. In: Proceedings of the 1993 winter simulation conference, Los Angeles, CA, 12–15 Dec 1993.
- [12] B. Zeigler and P. Hammonds. Modeling and simulation-based data engineering: introducing pragmatics into ontologies for net-centric information exchange. London: Academic Press, 2007.
- [13] H. Lee and B. Zeigler. System entity structure ontological data fusion process integrated with c2 systems. pages 4: 206–225. J Defense Model Simul Appl Methodol Technol, 2010.
- [14] B. Zeigler. Object-oriented simulation with hierarchical, modular models: intelligent agents and endomorphic systems. San Diego, CA: Academic Press Professional, 1990.
- [15] B. Zeigler. Multifaceted modeling and discrete event simulation. Orlando, FL: Academic Press, 1984.
- [16] Umut Durak. Extending the knowledge discovery metamodel for architecture-driven simulation, modernization. page Vol. 91(12) 1052–1067. Simulation: Transactions of the Society for Modeling and Simulation International, 2015.
- [17] Bikash Chandra Karmokar, Umut Durak, Sven Hartmann, and Bernard P. Ziegler. Towards a standard computational representation for system entity structures.
- [18] B. P. Zeigler and P. E. Hammonds. Modeling and simulation-based data engineering: introducing pragmatics into ontologies for net-centric information exchange. Elsevier, 2007.
- [19] H. S. Thompson, N. Mendelsohn, D. Beech, and M. Maloney. W3c xml schema definition language (xsd) 1.1 part 1: Structures. page W3C Working Draft Dec vol.3. The World Wide Web Consortium (W3C), 2009.
- [20] Bernhard Kaiser. Application story of odd as part of safety assurance: The significance of a well-structured odd specification for the ad safety and sotif process. ANSYS, June 2021.
- [21] Asam sim:guide, standardization for highly automated driving. ASAM e.V.
- [22] U. Durak, S. Jafer, R. Wittman, S. Mittal, S. Hartmann, and B. P. Zeigler. Computational representation for a simulation scenario definition language. page page 1398. In 2018 AIAA Modeling and Simulation Technologies Conference.
- [23] B. Chandra Karmokar, U. Durak, S. Jafer, B. N. Chhaya, and S. Hartmann. Tools for scenario development using system entity structures. page page 1712. In AIAA Scitech 2019 Forum.
- [24] B. P. Zeigler and P. E. Hammonds. Modeling and simulation-based data engineering: Introducing pragmatics into ontologies for net-centric information exchange. Elsevier, 2007.
- [25] Stefan Riedmaier, Thomas Ponn, Dieter Ludwig, Bernhard Schick, and Frank Diermeyer. Survey on scenario-based safety assessment of automated vehicles.
- [26] Demin Nalic, Tomislav Mihalj, Maximilian Baeumler, and Matthias Lehmann. Scenario based testing of automated driving systems: A literature survey.



# Critical Scenario Techniques for Automated Vehicles: Literature Review

Umut Durak<sup>1</sup>, Siddhartha Gupta<sup>2</sup>, Kaan Akgün<sup>3</sup>

<sup>1</sup>DLR Institute of Flight Systems, Lilienthalplatz 7, 38108 Braunschweig, Germany; umut.durak@dlr.de

<sup>2</sup>Institut für Informatik, Technische Universität Clausthal, 38678, Clausthal-Zellerfeld, Germany; siddhartha.gupta@tu-clausthal.de

<sup>3</sup>Institut für Informatik, Technische Universität Clausthal, 38678, Clausthal-Zellerfeld, Germany; kaan.akguen@tu-clausthal.de

**Abstract.** As Automated Driving Systems (ADS) become more widely used, there is rising worry regarding their safety and security. Traditional verification techniques, such as evaluating the vehicle's performance over a certain number of miles or kilometres, are insufficient to assess ADS risk. Instead, alternate approaches, such as scenario-based testing, which involves assessing the vehicle's performance in simulated situations that match real-world settings, are required to determine the system's safety and performance. This article presents a literature review on the importance of critical scenario analysis in ensuring the safety and reliability of ADS. Critical scenario analysis is a complete approach for identifying, quantifying, prioritising, selecting, and validating the most critical situations for ADS development and testing. ADS can reduce the frequency of accidents and fatalities caused by human error. Through this literature review, readers will understand the importance and benefits of critical scenario analysis methods and approaches, allowing them to systematically evaluate risks and opportunities and better comprehend the potential outcomes of future scenarios.

## Introduction

The increasing number of vehicle accidents and fatalities is a global concern, with the World Health Organization reporting that 1.3 million deaths occur yearly [1]. The introduction of Automated Driving Systems (ADS) has the potential to drastically reduce this number, as it aims to remove human error – the leading cause of such accidents. This is because ADSs are meant to remove human mistakes, which are the primary cause of automobile accidents. The Society of Automobile Engineers (SAE) has classified different levels of automation for ADSs, ranging from no automation (level 0) to full automation (level 5) [2]. ADSs with an SAE level of 3 or above have a higher degree of automation and can accomplish many driving activities without human involvement [2]. As the level of automation

increases, so does the level of safety concern, including the potential for system malfunction or failure [3]. To ensure the reliability and security of these systems, ADS must undergo comprehensive security evaluations before being deployed to the public[3],[4].

The most common procedures for validating ADS involve evaluating the vehicle's performance over a certain number of miles or kilometres and assessing the system's safety and dependability [6]. This is done by measuring the miles or kilometres a vehicle can travel without incident. The assumption is that the further a vehicle can go without incident, the safer it is. However, these techniques are not considered sufficient for evaluating the risk of ADS as they do not consider the various operational scenarios that a vehicle may encounter in the real world [5]. As an alternative, scenario-based testing may be used to more accurately assess the system's safety and performance [6]. Scenario-based testing involves evaluating the vehicle's performance in simulated scenarios replicating real-world conditions, such as variable weather, traffic patterns, and road conditions. The process involves deliberately modifying the conditions and parameters of the simulated scenario to test the vehicle's ability to operate safely and effectively. This method aims to evaluate the safety and reliability of the ADS by exposing it to a wide range of potential conditions and situations it may encounter in real-world operations. Virtual testing is scenario-based testing that uses computer simulations to replicate these scenarios instead of physical testing. This approach allows for a more efficient and cost-effective ADS testing method [29].

Critical scenarios refer to specific driving situations with a high probability of causing safety issues or problems in the ADS [10]. These scenarios may present a significant risk to the safety of the autonomous vehicle, its passengers, and other road users. They may pose a challenge to the proper functioning of the ADS. These scenarios can include but are not limited to, situations

such as heavy traffic, poor weather conditions, complex road layouts, and unexpected obstacles. Identifying and testing for these critical scenarios is an essential step in ensuring the safety and reliability of ADS before they are released to the public [10].

The literature review focuses on specific methods proposed in previous research for identifying critical scenarios likely to present significant risks or challenges to the safety or operation of ADS. These methods can include techniques for identifying and quantifying the potential impact of different scenarios, prioritising them based on their likelihood or potential impact, selecting the most critical ones, and evaluating them can increase performance to focus on during the development of the ADS. These methods aim to help ensure that the ADS is thoroughly tested and validated against a wide range of scenarios likely to be encountered in real-world operations before deployment to improve its safety and reliability. First, section 1 introduces fundamental concepts related to scenario-based safety validation for ADS. Then, Section 2 describes the literature review, which is used to identify relevant information on critical scenario analysis. This process involves searching through academic journals, conference proceedings, and other relevant sources to identify articles, papers, and studies that address critical scenario analysis in the context of ADS.

This paper presents a comprehensive overview of various techniques in a cohesive manner, allowing the reader to gain a holistic understanding of the subject matter. Bringing together a diverse range of concepts and approaches facilitates a deeper understanding of the interconnectedness of these different aspects and how they can be applied in practice.

## 1. KEY CONCEPTS

ADS are vehicles designed to operate and navigate without human intervention. These systems use a combination of sensors, cameras, and other technologies to sense their surroundings and decide how to drive and operate the vehicle. SAE has defined many levels of autonomy for ADS, ranging from Level 0, which requires human intervention for all driving activities, to Level 5, which can do all driving functions without human intervention. Many levels of autonomy for ADS, ranging from Level 0, which requires human intervention for all driving activities, to Level 5, which can do all driving functions without human intervention. The criteria for each level are clearly defined, including the tasks the system can perform and the situations in which it can perform them without human intervention. ADS aims to increase safety by using advanced technologies to

perceive and analyse the driving environment, make decisions, and control the vehicle. By relying on these technologies rather than human drivers, ADS can potentially reduce the frequency of accidents and fatalities caused by human error, such as distracted or impaired driving [11],[12].

ISO/PAS 21448:2019, as cited by Geyer et al. and Ulbrich et al., defines a scenario as a sequence of events occurring in a particular order and in different locations or settings (referred to as "scenes"). Scenarios help describe the temporal series of images portrayed by scenes and actions, and occurrences that can enhance them[13]-[15].

Thorn et al. employed a method of testing called scenario-based testing to evaluate the performance of ADS. This method involves creating simulated scenarios replicating real-world conditions and situations that the ADS may encounter during operation. For example, scenario-based testing can include testing the vehicle's performance in simulated scenarios replicating real-world conditions such as variable weather, traffic patterns, or road conditions. These scenarios evaluate the ADS' ability to safely and effectively navigate different driving conditions [16].

Menzel et al. introduced the concept of using different levels of abstraction to represent and analyse potential scenarios that an autonomous vehicle may encounter. Functional scenarios are described using semantic or linguistic notations and are defined at a high level of abstraction. Logical scenarios are represented using state-space level with parameter ranges and are defined at a medium level of abstraction. Finally, concrete scenarios are described using concrete parameter values and are defined at a low level of abstraction. Using these different levels of abstraction allows researchers to better understand a vehicle's capabilities and limitations and develop strategies for improving safety and reliability [17].

Although not a topic for discussion in this paper, it should be noted that Operational Design Domain (ODD) is a concept mainly used in ADS to define the operating conditions and limitations of the vehicle to ensure safe operation. ODD specifies the range of environmental conditions within which the ADS is designed to operate. In reality, ADS is the combination of hardware and software that collectively performs the entire dynamic driving task (DDT) on a sustained basis, regardless of whether it is limited to a specific ODD. Specifying the ODD helps to set the boundaries of safe operation for the vehicle. At the same time, the ADS is responsible for navigating the car and reacting to the environment and traffic based on the ODD-defined parameters. Also

included in ADS is the concept of ontology, which is a formal representation of knowledge that describes concepts and relationships within a specific domain [22]. In the context of ADS, an ontology can be used to represent knowledge about the driving scene, including information about the road layout, traffic signs, and the behaviour of other vehicles. The ontology can also be used to represent the relationships between concepts, such as how traffic signs constrain the behavior of vehicles. Using an ontology to represent this knowledge allows for automation in creating traffic scenarios for testing and simulation, which is a more efficient process than manually creating these scenarios by experts [9]. This can be useful to identify critical scenarios in the design of ADS and to evaluate the performance of the ADS in a wide range of conditions.

Additionally, ontology metamodels can provide a standardised, structured way of representing and organising knowledge about the environment and the vehicle's capabilities [21]. Therefore, ODD and ontology metamodels may be considered vital components in ensuring the safety and performance of ADS. However, noting that ODD is not always required for scenario generation [7],[8],[9],[21],[22].

Critical scenarios generally refer to specific driving situations that present a high risk or danger to the safety and operation of ADS. These scenarios may include potential collisions, traffic violations, or other dangerous situations. Analysing critical scenarios is essential for developing strategies to enhance the safety and performance of ADS [18].

According to Zhang et al., critical scenarios are defined as difficult circumstances under which an autonomous vehicle can operate safely. The authors specify the specific parameters, such as weather and road conditions, and external circumstances, such as traffic density and presence of pedestrians, necessary for the vehicle to run safely. A comprehensive set of essential scenarios should be defined to ensure that the vehicle operates effectively in various locations and scenarios. Examining critical scenarios aids in understanding and developing strategies for improving the vehicle's performance and safety limits [19].

## 2. CRITICAL SCENARIO ANALYSIS

Critical scenario analysis is a comprehensive process used to develop and test ADS. This process aims to ensure the safety and reliability of ADS by thoroughly testing their capabilities under a range of scenarios based on real-world data and expert opinions. The following sections will present a literature review and explain the

critical scenario analysis process. The process starts with critical scenario identification and quantification, followed by critical scenario prioritisation, validation, selection, and evaluation.

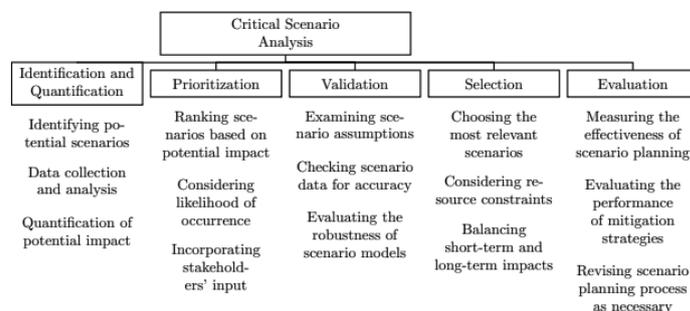


Figure 1: Overview of Critical Scenario Analysis[97]

### 2.1 Critical Scenario Identification & Quantification

Critical scenario identification and quantification is identifying possible scenario types, grouping them into categories, and giving numerical numbers to each scenario to compare and rank them. The purpose of scenario identification is to develop a well-defined set of scenarios that may be utilised to assess a specific problem or circumstance, identify potential risks and opportunities, and develop strategies for managing them. Scenario quantification is an essential step in the scenario management process. It helps to identify the most critical scenarios and concentrate on the issues most likely to have a significant impact.

Articles by Kramer et al. and Papa and Ferreira present solutions for meeting critical scenario identification and quantification requirements. Kramer et al. propose combining existing methodologies for hazard analysis and risk assessment (HARA) with further advancements and recommend using an integrated HARA strategy early in the development phase of ADS so that the results may be included during the design and testing phases. They also propose a method for identifying and specifying hazards related to ADS. The technique consists of modelling the system design architecture and intended system design, identifying hazards, performing a causal chain analysis, deriving the triggering conditions for hazards, and quantifying and assessing the associated risks. Papa and Ferreira use a mix of exploratory and back-casting approaches to scenario building, allowing flexibility in the process and customisation for the specific situation under investigation. They define scenarios as hypothetical futures highlighting the policy quandaries and behavioural conflicts that may arise as ADSs transition from speculation to reality. They provide two alternatives, one optimistic and the other pessimistic, and identify critical decisions related to the deployment

and governance of ADS. Additionally, Song et al. propose a new research strategy for selecting critical ADS test scenarios, which includes studying the system requirements and implementation of the autonomous driving functionalities, defining the system's essential parameters, describing the parameter's ratio and dispersion, and establishing feasible object operations and severity levels for the ADS, and using the modeFrontier tool to develop an optimisation technique for identifying critical test scenarios for ADS. These are the main approaches for critical scenario identification and quantification being utilised today [22],[23],[24].

## 2.2 Critical Scenario Prioritisation

Critical scenario prioritisation is used in ADS to prioritise tests for ADS based on diversity and cost. Critical scenario prioritisation requirements include minimising the data needed for autonomous vehicle testing while maintaining accuracy and detecting almost all inserted defects.

Several solutions have been proposed to meet the requirements of critical scenario prioritisation. Deng et al. offer STRaP, a method that enables the alignment of messages from different channels in the recording and the conversion of each frame of the aligned recording into a vector based on a driving situation data structure. Additionally, the STRaP method allows for slicing the aligned and vectorised recording into components based on the correlation of successive vectors and the prioritisation of the classes based on their coverage of driving scene aspects and rarity. Birchler et al. present two evolutionary techniques for selecting ADS experiments in virtual settings based on diversity measures derived from static road characteristics. They demonstrate that their strategy enhances the detection of security issues and that multi-objective meta-heuristics better single-objective procedures for promoting SDC tests. The solutions provided by this article, such as the Singular-Objective Genetic Algorithm (SO-SDC-Prioritizer), Multi-Objective Genetic Algorithm (MO-SDC-Prioritizer), and the Black-box Greedy Algorithm, all aim to minimise the amount of data needed for autonomous vehicle testing while maintaining accuracy and detecting almost all inserted defects [26],[27].

## 2.3 Critical Scenario Selection

Critical scenario selection is choosing the most appropriate scenario for a given situation using the results of scenario evaluation, quantification, and prioritisation. The goal of scenario selection is to select the scenario that offers the best balance of benefits, risks, and uncertainties based on the specific goals and objectives of the situation.

Several strategies have been presented to address the essential scenario selection requirements. Wang et al. recommended choosing the best scenario for a given condition. Riedmaier et al. distinguish between techniques based on testing and falsification in their literature review. Testing-based approaches to safety assessment involve evaluating the safety of a system by testing it under a range of scenarios selected to assess the system's safety under various conditions. Examples include N-wise sampling, Interactive Design of Experiments (DoE), automated method for regression testing, Satisfiability Modulo Theories (SMT), Signal Temporal Logic (STL), Randomization procedures, and Developing logical scenarios. These methods all ensure that the selected scenarios represent real-world conditions and can identify potential issues in the ADS. Falsification is a method of testing that seeks out instances that break safety criteria for ADSs. Examples include using real-world accident data, increasing the criticality of specific scenarios, identifying critical scenarios within established parameter ranges, using simulation-based falsification, using reinforcement-based adaptive stress testing learning, using optimisation methods such as Particle Swarm Optimization, Differential Evolution Genetic Optimization, Simulated Annealing and using machine learning models such as random forest for scenario selection [28 -85]

## 2.4 Critical Scenario Validation

Critical scenario validation is the process of checking scenarios to ensure that they are accurate and realistic. This can involve comparing the scenarios with real-world data and expert opinions and adjusting them as needed. The goal of scenario validation is to ensure that the scenarios being considered are based on the most precise and up-to-date data and accurately reflect the real-world situations that ADS may encounter.

Various solutions have been proposed to meet the requirements of critical scenario validation. Wang et al. propose VAAFO, an ADS validation approach that combines the benefits of test drives with virtual assessment. This approach includes a route analysis unit, a reality design adjustment unit, an evaluation unit, and a scenario record unit. Second, Elrofai et al. propose a method for efficiently and systematically evaluating the safety of advanced ADS by integrating deterministic algorithms with data science to uncover events concealed in massive volumes of the dataset. Finally, Weng et al. propose a total and effective probabilistic solution to the security assessment challenge. This approach is based on set invariance, divides the given data into separate groups, and validates each group independently. These solutions aim to improve the accuracy and reliability of critical scenario validation by providing methods for

comparing scenarios to real-world data, adjusting them as needed, and evaluating the safety and security of ADS [86],[87],[88].

## 2.5 Critical Scenario Evaluation

Critical scenario evaluation assesses the likelihood and potential impact of different scenarios to identify the most critical ones and focus on the issues most likely to have a significant effect. Requirements for critical scenario evaluation include the ability to analyse decision-making in uncertain settings, identify the best possible actions to take in each situation, and evaluate the performance of the autonomous system in critical scenarios.

The articles provided describe various solutions for meeting these requirements. Huang et al., for example, present scenario analysis as a method for identifying critical scenarios. Van Der Pol suggests using Markov Decision Process (MDP) to analyse decision-making in uncertain settings. Sutton et al. present reinforcement learning (RL) as a machine learning approach that uses MDPs to identify the best possible actions to take in each situation. Levine et al. present REINFORCE, a specific type of RL algorithm that uses a policy gradient method called Monte Carlo sampling to find the optimal policy. Ren and colleagues present Neural architecture search (NAS) as a parameter optimisation strategy for constructing a learning algorithm for a given task. Shalev-Shwartz et al. present the Responsible and Safe Social Scenario (RSS) method, which guarantees that an operator cannot be the cause of an incident. Dosovitskiy et al. present Carla, an open-source simulator for ADS that aims to provide realistic settings and a ROS bridge to test and evaluate the system's performance in critical scenarios. These are the most commonly used approaches for critical scenario evaluation [89 - 96].

## 3. CONCLUSION AND FUTURE WORK

The scenario-based methodology presented in this work helps discover, measure, and prioritise relevant situations of critical scenarios to assist ADS development. As a result, researchers may better understand the capabilities and limits of ADS and devise ways to improve their safety and performance by methodically creating and analysing demanding scenarios. Future studies on this issue might include developing and testing the scenario-based approach to critical scenario analysis. This may also require conducting more complete assessments of the technique using more extensive and more diverse case datasets, as well as investigating the impact of different aspects, such as the ADSs level and the

complexity of the environment, on the approach's performance. Furthermore, it may be worthwhile investigating possible applications of the technique outside of the creation of ADS, such as designing other complex systems or decision-making in dynamic situations. Overall, it is imperative to study more in this field to understand better crucial circumstances and their importance in developing sophisticated technology.

## References

- [1] World Health Organization. Number of Road Traffic Deaths. [Online] . Available: <https://www.who.int/data/gho/data/themes/topics/topic-details/GHO/road-traffic-mortality>
- [2] SAE International, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," SAE Tech. Paper J3016, 2018.
- [3] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," SAE Int. J. Transp. Saf., vol. 4, no. 1, pp. 1524, 2016.
- [4] P. Koopman and M. Wagner, "Toward a framework for highly automated vehicle safety validation," in Proc. WCX World Congr. Exper., 2018, pp. 113, DOI: 10.4271/2018-01-1071.
- [5] A. Shetty, M. Yu, A. Kurzhanskiy, O. Grembek, H. Tavaafoghi, and P. Varaiya, "Safety challenges for autonomous vehicles in the absence of connectivity," Transportation Research Part C: Emerging Technologies, vol. 128, p. 103133, 2021.
- [6] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" Transportation Research Part A: Policy and Practice, vol. 94, pp. 182–193, 2016.
- [7] IEEE Transactions on Intelligent Transportation Systems (2017). A survey of autonomous driving system design and development.
- [8] Czarnecki, K. (2018). Operational Design Domain for Automated Driving Systems. Taxonomy of Basic Terms ", Waterloo Intelligent Systems Engineering (WISE) Lab, University of Waterloo, Canada.
- [9] Klück, F., Li, Y., Nica, M., Tao, J., & Wotawa, F. (2018, October). Using ontologies for test suite generation for automated and autonomous driving functions. In 2018 IEEE International symposium on software reliability engineering workshops (ISSREW) (pp. 118-123). IEEE.
- [10] Khastgir, S., Brewerton, S., Thomas, J., & Jennings, P. (2021). Systems approach to creating test scenarios for automated driving systems—reliability engineering & system safety, 215, 107610.
- [11] Society of Automotive Engineers International. (2019). Autonomous driving. Retrieved from [http://www.sae.org/standards/content/j3016\\_201901/](http://www.sae.org/standards/content/j3016_201901/)
- [12] IEEE Transactions on Intelligent Transportation Systems. (2019). Autonomous driving: Challenges and opportunities. 20(4), 1718-1731.
- [13] ISO/PAS 21448:2019(E) (2019). "Road vehicles - Safety of the intended functionality ."ISO
- [14] S. Geyer et al. (2014). "Concept and development of a unified ontology for generating test and use-case catalogs for

assisted and automated vehicle guidance. "In: IET Intelligent Transportation Systems, Vol. 8, Iss. 3, pp.193-189.

[15] S. Ulbrich et al. (2015). "Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving." IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, pp. 982-988.

[16] Thorn, E., Kimmel, S. C., Chaka, M., & Hamilton, B. A. (2018). A framework for automated driving system testable cases and scenarios (No. DOT HS 812 623). United States. Department of Transportation. National Highway Traffic Safety Administration.

[17] T. Menzel, G. Bagschik, and M. Maurer. Scenarios for Development, Test, and Validation of Automated Vehicles. arXiv preprint:1801.08598, pages 1–7, 2018.

[18] Khastgir, S., Brewerton, S., Thomas, J., & Jennings, P. (2021). Systems approach to creating test scenarios for automated driving systems. Reliability engineering & system safety, 215, 107610.

[19] Zhang, X., Tao, J., Tan, K., Törngren, M., Sánchez, J. M. G., Ramli, M. R., ... & Felbinger, H. (2021). Finding critical scenarios for automated driving systems: A systematic literature review. arXiv preprint arXiv:2110.08664.

[20] Colwell, I., Phan, B., Saleem, S., Salay, R., & Czarnecki, K. (2018, June). An automated vehicle safety concept based on runtime restriction of the operational design domain. In 2018 IEEE Intelligent Vehicles Symposium (IV) (pp. 1910-1917). IEEE.

[21] Abmann, U., Zschaler, S., & Wagner, G. (2006). Ontologies, metamodels, and the model-driven paradigm. In Ontologies for software engineering and technology (pp. 249-273). Springer, Berlin, Heidelberg.

[22] Kramer, B., Neurohr, C., Büker, M., Böde, E., Fränzle, M., & Damm, W. (2020, September). Identification and quantification of hazardous scenarios for automated driving. In International Symposium on Model-Based Safety and Assessment (pp. 163-178). Springer, Cham.

[23] Papa, E., & Ferreira, A. (2018). Sustainable accessibility and the implementation of automated vehicles: Identifying critical decisions. Urban Science, 2(1), 5.

[24] Song, Q., Tan, K., Runeson, P., & Persson, S. (2022). Critical Scenario Identification for Realistic Testing of Autonomous Driving Systems.

[25] Deng, Y., Zheng, X., Zhang, M., Lou, G., & Zhang, T. (2022, November). Scenario-based test reduction and prioritisation for multi-module autonomous driving systems. In Proceedings of the 30<sup>th</sup> ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 82-93).

[26] Birchler, C., Khatiri, S., Derakhshanfar, P., Panichella, S., & Panichella, A. (2022). Single and Multi-objective Test Cases Prioritisation for Self-driving Cars in Virtual Environments. Proceedings of the ACM on Measurement and Analysis of Computing Systems.

[27] Wang, J., Pun, A., Tu, J., Manivasagam, S., Sadat, A., Casas, S., ... & Urtasun, R. (2021). Advise: Generating safety-critical scenarios for self-driving vehicles. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 9909-9918).

[28] Riedmaier, S., Ponn, T., Ludwig, D., Schick, B., & Diermeyer, F. (2020). Survey on the scenario-based safety assessment of automated vehicles. IEEE Access, 8, 87456-87477.

Just 18 deleted

[29] T. Ponn, D. Fratzke, C. Grandt, and M. Lienkamp, "Towards certification of autonomous driving: Systematic test case generation for a comprehensive but economically-feasible assessment of lane keeping assist algorithms," in Proc. 5th Int. Conf. Vehicle Technol. Intell. Transp. Syst., 2019, pp. 333342.

[30] H. Beglerovic, A. Ravi, N. Wikström, H.-M. Koegeler, A. Leitner, and J. Holzinger, "Model-based safety validation of the automated driving function highway pilot," in Proc. 8th Int. Munich Chassis Symp. (Proceedings), P. P. E. Pfeffer, Ed. Wiesbaden, Germany: Springer, 2017, pp. 309329.

[31] E. Rocklage, H. Kraft, A. Karatas, and J. Seewig, "Automated scenario generation for regression testing of autonomous vehicles," in Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC), Oct. 2017, pp. 476483.

[32] B. Kim, A. Jarandikar, J. Shum, S. Shiraishi, and M. Yamaura, "The SMT-based automatic road network generation in vehicle simulation environment," in Proc. 13th Int. Conf. Embedded Software. (EMSOFT), Oct. 2016, pp. 110.

[33] B. Kim, T. Masuda, and S. Shiraishi, "Test specification and generation for connected and autonomous vehicle in virtual environments," ACM Trans. Cyber-Phys. Syst., vol. 4, no. 1, pp. 126, Jan. 2020.

[34] I. Majzik, O. Semerath, C. Hajdu, K. Marussy, Z. Szatmari, Z. Micskei, A. Voros, A. A. Babikian, and D. Varro, "Towards system-level testing with coverage guarantees for autonomous vehicles," in Proc. ACM/IEEE 22nd Int. Conf. Model Driven Eng. Lang. Syst. (MODELS), Sep. 2019, pp. 8994.

[35] S. Khastgir, G. Dhadyalla, S. Birrell, S. Redmond, R. Addinall, and P. Jennings, "Test scenario generation for driving simulators using constrained randomisation technique," in Proc. SAE World Congr. Exper. Warrendale, PA, USA, 2017.

[36] L. Huang, Q. Xia, F. Xie, H.-L. Xiu, and H. Shu, "Study on the test scenarios of level 2 automated vehicles," in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2018, pp. 4954.

[37] M. Althoff and J. M. Dolan, "Reachability computation of low-order models for the safety verification of high-order road vehicle models," in Proc. Amer. Control Conf. (ACC), Jun. 2012, pp. 35593566.

[38] M. O'Kelly, A. Sinha, H. Namkoong, R. Tedrake, and J. C. Duchi, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 98279838.

[39] D. Åsljung, J. Nilsson, and J. Fredriksson, "Comparing collision threat measures for verification of autonomous vehicles using extreme value theory," IFAC-PapersOnLine, vol. 49, no. 15, pp. 5762, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2405896316309855>

[40] D. Åsljung, J. Nilsson, and J. Fredriksson, "Using extreme value theory for vehicle level safety validation and implications for autonomous vehicles," IEEE Trans. Intell. Vehicles, vol. 2, no. 4, pp. 288297, Dec. 2017.

[41] D. Zhao, "Accelerated evaluation of automated vehicles," Ph.D. dissertation, Univ. Michigan, Ann Arbor, MI, USA, 2016.

[42] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated evaluation of

automated vehicles safety in lane change scenarios based on importance sampling techniques," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 595607, Mar. 2017.

- [43] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, "Accelerated evaluation of automated vehicles in car-following maneuvers," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 733744, Mar. 2018.
- [44] Z. Huang, D. Zhao, H. Lam, D. J. LeBlanc, and H. Peng, "Evaluation of automated vehicles in the frontal cut-in scenario An enhanced approach using piecewise mixture models," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/June. 2017, pp. 197202.
- [45] Z. Huang, H. Lam, and D. Zhao, "Sequential experimentation to efficiently test automated vehicles," in *Proc. Winter Simulation Conf. (WSC)*, Dec. 2017, pp. 30783089.
- [46] Z. Huang, H. Lam, and D. Zhao, "Towards affordable on-track testing for autonomous vehicle Kriging-based statistical approach," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 16.
- [47] Z. Huang, Y. Guo, M. Arief, H. Lam, and D. Zhao, "A versatile approach to evaluating and testing automated vehicles based on kernel methods," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 47964802.
- [48] Z. Huang, H. Lam, D. J. LeBlanc, and D. Zhao, "Accelerated evaluation of automated vehicles using piecewise mixture models," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 28452855, Sep. 2018.
- [49] Z. Huang, H. Lam, and D. Zhao, "An accelerated testing approach for automated vehicles with background traffic described by joint distributions," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 933938.
- [50] Z. Huang, M. Arief, H. Lam, and D. Zhao, "Synthesis of different autonomous vehicles test approaches," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 20002005.
- [51] Z. Huang, M. Arief, H. Lam, and D. Zhao, "Evaluation uncertainty in data-driven self-driving testing," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 19021907.
- [52] S. Zhang, H. Peng, D. Zhao, and H. E. Tseng, "Accelerated evaluation of autonomous vehicles in the lane change scenario based on subset simulation technique," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 39353940.
- [53] M. Arief, P. Glynn, and D. Zhao, "An accelerated approach to safely and efficiently test pre-production autonomous vehicles on public streets," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 20062011.
- [54] M. O'Kelly, A. Sinha, H. Namkoong, R. Tedrake, and J. C. Duchi, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 98279838.
- [55] X. Wang, H. Peng, and D. Zhao, "Combining reachability analysis and importance sampling for accelerated evaluation of highly automated vehicles at pedestrian crossing," in *Proc. Dyn. Syst. Control Conf.*, 2019.
- [56] S. P. Olivares, N. Rebernik, A. Eichberger, and E. Stadlober, "Virtual stochastic testing of advanced driver assistance systems," in *Advanced Microsystems for Automotive Applications*. Cham, Switzerland: Springer, 2016, pp. 2535.
- [57] Y. Akagi, R. Kato, S. Kitajima, J. Antona-Makoshi, and N. Uchida, "A risk-index base sampling method to generate scenarios for the evaluation of automated driving vehicle safety\*," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 667672.
- [58] D. Asljang, M. Westlund, and J. Fredriksson, "A probabilistic framework for collision probability estimation and an analysis of the discretisation precision," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 5257.
- [59] L. Stark, M. Düring, S. Schoenawa, J. E. Maschke, and C. M. Do, "Quantifying vision zero: Crash avoidance in rural and motorway accident scenarios by combination of ACC, AEB, and LKS projected to German accident occurrence," *Traffic Injury Prevention*, vol. 20, no. 1, pp. S126S132, Jun. 2019.
- [60] G. E. Mullins, "Adaptive sampling methods for testing autonomous systems," Ph.D. dissertation, Univ. Maryland, College Park, MD, USA, 2018.
- [61] Stark, L., Kühn, A., & Kühn, A. (2017). Deriving test cases from accident data for the safety assessment of automated vehicles. In *Transportation Research Part C: Emerging Technologies* (Vol. 80, pp. 134-144).
- [62] Stark, L., Kühn, A., & Kühn, A. (2018). Test case selection from an accident database for the safety assessment of automated vehicles. In *Transportation Research Part C: Emerging Technologies* (Vol. 94, pp. 1-11).
- [63] Fahrenkrog, F., Kluge, B., & Kühn, A. (2018). Deriving safety test cases from real-world accident data for the assessment of automated vehicles. In *Transportation Research Part C: Emerging Technologies* (Vol. 93, pp. 314-326).
- [64] Geiger, M., Kühn, A., & Kühn, A. (2019). Deriving representative test cases for the safety assessment of automated vehicles from an extensive accident database. In *Transportation Research Part C: Emerging Technologies* (Vol. 101, pp. 199-213).
- [65] A. Pierson, W. Schwarting, S. Karaman, and D. Rus, "Learning risk level set parameters from data sets for safer driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 273280.
- [66] M. Althoff and S. Lutz, "Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 13261333.
- [67] M. Klischat and M. Althoff, "Generating critical test scenarios for automated vehicles with evolutionary algorithms," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 23522358.
- [68] F. Gao, J. Duan, Y. He, and Z. Wang, "A test scenario automatic generation strategy for intelligent driving systems," *Math. Problems Eng.*, vol. 2019, pp. 110, Jan. 2019.
- [69] Q. Xia, J. Duan, F. Gao, T. Chen, and C. Yang, "Automatic generation method of test scenario for ADAS based on complexity," *SAE Tech. Paper 2017-01-1992*, 2017, DOI: 10.4271/2017-01-1992.
- [70] Q. Xia, J. Duan, F. Gao, Q. Hu, and Y. He, "Test scenario design for intelligent driving system ensuring coverage and effectiveness," *Int. J. Automot. Technol.*, vol. 19, no. 4, pp. 751758, Aug. 2018.
- [71] J. Wang, C. Zhang, Y. Liu, and Q. Zhang, "Traffic sensory data classification by quantifying scenario complexity," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 15431548.

- [72] C. Zhang, Y. Liu, Q. Zhang, and L. Wang, "A graded offline evaluation framework for intelligent vehicle's cognitive ability," in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2018, pp. 320325.
- [73] Y. Qi, Y. Luo, K. Li, W. Kong, and Y. Wang, "A trajectory-based method for scenario analysis and test effort reduction for highly automated vehicle," in Proc. SAE Tech. Paper Series, Warrendale, PA, USA, 2019.
- [74] T. Ponn, C. Gnandt, and F. Diermeyer, "An optimisation-based method to identify relevant scenarios for type approval of automated vehicles," in Proc. 26th Int. Tech. Conf. Enhanced Saf. Vehicles (ESV). Washington, DC, USA: National Highway Traffic Safety Administration, 2019.
- [75] T. Ponn, A. Schwab, F. Diermeyer, C. Gnandt, and J. Záhorský, "A method for the selection of challenging driving scenarios for automated vehicles based on an objective characterisation of the driving behavior," in 9. Tagung Automatisiertes Fahren. Munich, Germany: Technical Univ. Of Munich, 2019.
- [76] P. Koopman and F. Fratrick, "How many operational design domains, objects, and events?" in Proc. SafeAI, 2019, pp. 14.
- [77] A. Corso, P. Du, K. Driggs-Campbell, and M. J. Kochenderfer, "Adaptivestress testing with reward augmentation for autonomous vehicle validation," in Proc. IEEE Intell. Transp. Syst. Conf. (ITSC), Oct. 2019, pp. 163168.
- [78] H. Beglerovic, M. Stolz, and M. Horn, "Testing of autonomous vehicles using surrogate models and stochastic optimisation," in Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC), Oct. 2017, pp. 16.
- [79] G. E. Mullins, "Adaptive sampling methods for testing autonomous systems," Ph.D. dissertation, Univ. Maryland, College Park, MD, USA, 2018.
- [80] C. E. Tuncali, T. P. Pavlic, and G. Fainekos, "Utilising S-TaLiRo as an automatic test generation framework for autonomous vehicles," in Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC), Nov. 2016, pp. 14701475.
- [81] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski, "Simulation-based adversarial test generation for autonomous vehicles with machine learning components," in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2018.
- [82] C. E. Tuncali, G. Fainekos, D. Prokhorov, H. Ito, and J. Kapinski, "Requirements-driven test generation for autonomous vehicles with machine learning components," IEEE Trans. Intell. Vehicles, early access, Nov. 25, 2019, DOI: 10.1109/TIV.2019.2955903.
- [83] B. Gangopadhyay, S. Khastgir, S. Dey, P. Dasgupta, G. Montana, and P. Jennings, "Identification of test cases for automated driving systems using Bayesian optimisation," in Proc. IEEE Intell. Transp. Syst. Conf. (ITSC), Oct. 2019, pp. 19611967.
- [84] M. Nabhan, M. Schoenauer, Y. Tourbier, and H. Hage, "Optimising coverage of simulated driving scenarios for the autonomous vehicle," in Proc. IEEE Int. Conf. Connected Vehicles Expo (ICCVE), Nov. 2019.
- [85] Hungar, H. (2020, October). A concept of scenario space exploration with criticality coverage guarantees. In International Symposium on Leveraging Applications of Formal Methods (pp. 293-306). Springer, Cham.
- [86] Wang, C., & Winner, H. (2019, October). Overcoming challenges of validation automated driving and identification of critical scenarios. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC) (pp. 2639-2644). IEEE.
- [87] Elrofai, H., Worm, D., & Op den Camp, O. (2016). Scenario identification for validation of automated driving functions. In Advanced Microsystems for Automotive Applications 2016 (pp. 153-163). Springer, Cham.
- [88] Weng, B., Capito, L., Ozguner, U., & Redmill, K. (2021). Towards guaranteed safety assurance of automated driving systems with scenario sampling: An invariant set perspective. IEEE Transactions on Intelligent Vehicles, 7(3), 638-651.
- [89] Huang, W., Wang, K., Lv, Y., & Zhu, F. (2016, November). Autonomous vehicles testing methods review. In 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) (pp. 163-168). IEEE.
- [90] Van Der Pol, E. (2016). Deep reinforcement learning for coordination in traffic light control. Master's thesis, University of Amsterdam.
- [91] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT Press, 2018.
- [92] S. Levine, "Cs 294: Deep reinforcement learning," 2017.
- [93] Ren, P., Xiao, Y., Chang, X., Huang, P. Y., Li, Z., Chen, X., & Wang, X. (2021). A comprehensive survey of neural architecture search: Challenges and solutions. ACM Computing Surveys (CSUR), 54(4), 1-34.
- [94] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," CoRR, vol. abs/1708.06374, 2017. [Online]. Available: <http://arxiv.org/abs/1708.06374>
- [95] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in Proceedings of the 1st Annual Conference on Robot Learning, 2017, pp. 1–16.
- [96] Karunakaran, D., Worrall, S., & Nebot, E. (2020). Efficient falsification approach for autonomous vehicle validation using a parameter optimisation technique based on reinforcement learning. arXiv preprint arXiv:2011.07699.
- [97] Figure 1

# Iterative Scenario-Based Testing in an Operational Design Domain for Artificial Intelligence Based Systems in Aviation

Bojan Lukic<sup>1\*</sup>, Jasper Sprockhoff<sup>1</sup>, Alexander Ahlbrecht<sup>1</sup>, Siddhartha Gupta<sup>1</sup>,  
Umut Durak<sup>1</sup>

<sup>1</sup>German Aerospace Center (DLR), Institute of Flight Systems, Lilienthalplatz 7, 38108 Braunschweig, Germany

\*[bojan.lukic@dlr.de](mailto:bojan.lukic@dlr.de)

**Abstract.** The use and development of Artificial Intelligence (AI) based systems is becoming increasingly prominent in different industries. The aviation industry is also gradually adopting AI-based systems, for instance, with Machine Learning algorithms for flight assistance. There are several reasons why adopting these technologies poses additional obstacles in aviation compared to other industries. One reason are the strong safety requirements which lead to obligatory and thorough assurance activities such as testing to obtain certification. Therefore, a systematic approach is needed for developing, deploying, and assessing test cases for AI-based systems in aviation. This paper proposes a method for iterative scenario-based testing for AI-based systems. The method contains three major parts: First, a high-level description of test scenarios; second, the generation and execution of these scenarios; and last, monitoring of parameters during scenario execution. Parameters are refined, and the steps are repeated iteratively. The method forms a basis for developing iterative scenario-based testing solutions. As a domain-specific example, a practical implementation of this method is illustrated. For an object detection application used on an airplane, flight scenarios, including multiple airplanes are generated from a descriptive scenario model and executed in a simulation environment. The parameters are monitored using a custom Operational Design Domain monitoring tool and refined in the process of iterative scenario generation and execution. The proposed iterative scenario-based testing method helps in generating precise test cases for AI-based systems while having a high potential for automation.

## Introduction

The practical use of Machine Learning (ML) applications for Artificial Intelligence (AI) based systems in aviation is still in an early stage. One reason is the premature nature of guidelines illustrating the proper implementation of those applications. Specifically, the additional and strict requirements and constraints for

introducing new systems in the aviation industry pose an obstacle. This makes the implementation and certification of ML algorithms for autonomy challenging. Recently, the European Union Aviation Safety Agency (EASA) [1] and Society of Automotive Engineers (SAE) [2] each published early versions of fundamental guidelines, discussing the implementation of ML applications in aeronautical systems. These guidelines provide guidance for implementing level 1 ML applications, which can assist humans. Due to the premature nature of these guidelines, the certifiability of ML applications in aviation, especially for fully AI-based systems, is not yet given. Yet, similar to traditional software, it is certain that specific verification artifacts need to be provided to increase trust. Typical artifacts include the results of conducted tests. As defined in the EASA guidance, implementing AI-based systems requires the exact definition of their Operational Design Domain (ODD).

The ODD defines the conditions under which a system operates correctly. In the domain of AI-based systems, the ODD defines the execution boundaries under which the AI-based system is designed and defines the parameters which need to be satisfied for the system to properly operate [3]. The definition of parameter boundaries for the correct behavior of an AI-based system becomes especially important when working in safety-critical domains such as aviation. For instance, the ODD of an aviation system can help with the definition of design assurance levels [4]. In this context, the definition of the system's ODD guarantees the generation of precise test cases for high test coverage. One systematic approach for developing test cases for AI-based systems in their operational domain is model-

based testing using the Model-Based Systems Engineering (MBSE) methodology. Due to its highly descriptive nature and model-centric approach [5], MBSE is an appropriate methodology to model systems on all levels of abstraction, making it useful in the development process of test cases for ML applications [6]. The iterative scenario testing concept presented in this work is amongst others exemplified with methods from MBSE.

This paper discusses the generation of test scenarios for an AI-based system. The use case is about using a computer vision algorithm to perform object detection and determine the distance to other aircraft to predict dangerous situations. The scenarios represent different situations with foreign aircraft used for testing. The detailed use case is explained in [7]. A method for iterative scenario-based testing of AI-based systems is presented in the scope of this work. Three essential parts of the method are defined: A high-level description of the scenarios to be executed, the testing environment in which test scenarios are executed, and a monitoring tool for narrowing down the parameter boundaries for the ODD of the respective system. A domain-specific implementation of this methodology is also presented. For modeling the systems involved and developing test cases, the MBSE tool Cameo<sup>1</sup> is used. The simulation is executed in FlightGear, a highly customizable open-source software for flight simulation<sup>2</sup>. The scenarios are generated in a model-based approach in Cameo and then executed in a FlightGear instance. Parameters are monitored using a custom Python library. The findings show that the presented iterative scenario-based testing method facilitates the definition and refinement of test scenarios for AI-based applications.

The remaining paper is structured as follows: In Section 1, related work and the status quo of scenario-based testing with a model-based approach are discussed. Section 2 presents the development of a domain-independent method for iterative scenario-based testing. The implementation of this methodology is presented in Section 3 with tools used for defining scenarios, executing them, and monitoring them.

<sup>1</sup>Dassault Systemes, 2022, Cameo Systems Modeler, available at <https://www.3ds.com/products-services/catia/products/no-magic/cameo-systems-modeler/>.

<sup>2</sup>FlightGear developers & contributors, 2021. FlightGear, Available at <https://www.flightgear.org/>.

## 1 Related Work

In [8], Jafer and Durak discuss the complexity of simulation scenario development in aviation. They propose ontology-based approaches to develop an aviation scenario definition language (ASDL). According to the authors, ontologies provide invaluable possibilities to tackle the complexity of simulation scenario development. Durak presents a model-driven engineering perspective for scenario development in [9]. The use of metamodels for generating executable scenarios is demonstrated with a sample implementation. Durak's work is closely related to the research presented in the work at hand, specifically the development of conceptual metamodels for generating executable scenarios.

Simulation-based data and scenario generation for AI-based airborne systems is discussed by Gupta in [10]. In the work, the authors aim to answer the question of what needs to be simulated for synthetic data and scenario generation in the simulation engineering process of an AI-based system. The used methods are a simulation-based data generation process adapted from EASA's first usable guidance for Level 1 machine learning applications and the scenario-based approach using SES, which is explained more thoroughly in the publications of Durak [11], [12] as well as Karmokar [13]. The work in [10] is succeeded with [14], which discusses behavioral modeling for scenario-based testing in aviation and introduces an enhanced approach for scenario-based testing called Operational Domain Driven Testing.

Closely related, [15] demonstrates the testing of black box systems, such as AI-based applications for autonomous road vehicles, in their ODD. The framework introduced by the authors is used to learn monitors in a feature space and prevent the system from using critical components when exiting its ODD. Scenario-based testing of autonomous road vehicles is discussed in [16] and [17]. The authors present an automated scenario-based testing methodology for vehicles using advanced AI-based applications. The work shows that the presented formal simulation approach effectively finds relevant tests for track testing with a real autonomous vehicle.

In [18], Hungar presents scenario-based testing for automated road vehicles. The outcome of the work is the PEGASUS method, which is used to assess highly automated driving functions. According to the author, the most important steps for scenario-based testing involve capturing all evolutions, i.e. variants, of func-

tional scenarios, formalization of them, systematic testing, the analysis of critical regions, and finally, the development of a risk chart.

Closely related to [9], the work presented in this paper discusses model-driven scenario development. In addition to the methodologies discussed in the related work, an iterative scenario parameter adjustment and generation process is introduced, forming the iterative scenario-based testing method. The method is illustrated with an exemplary generation of test scenarios for an AI-based demonstrator. In the next section, the methodology for this domain- and tool-independent iterative scenario-based testing method is presented.

## 2 Iterative Scenario-Based Testing Concept

The related work shows that there are many ways to realize scenario-based testing for AI-based systems. Especially when talking about domain-specific tools, a variety of testing strategies are possible. A generalization of these testing strategies can help with defining universal testing methods. To achieve that, a fundamental, tool-independent method is needed to describe the basic methodology for iterative scenario-based testing on a high level of abstraction. This method can then be used to build some domain-specific testing tools. For such iterative scenario-based testing, three fundamental components have been identified:

### Scenario Model

First, a high-level description of the testing scenarios needs to be defined. This high-level model can be achieved by describing the scenarios' fundamental components. Modeling tools or formalized methods can for instance be used to formulate the scenarios and derive all required scenario variations from the high-level model. The method shall be capable of generating an arbitrary number of scenarios with high parameter variation from the high-level description to achieve satisfactory test coverage for the application to be verified.

### Environment for Scenario Execution

Second, an environment for executing the derived scenarios should be selected. The environment can be of different types, such as simulated, real system, or a mix of both, e.g. real systems extended with elements from augmented reality. These environments have different advantages and disadvantages. A simulated system can

be deployed quickly, offers consistent conditions, and is cost-effective. The biggest drawback of simulated environments is their sim-to-real gap. The gap refers to the applicability of simulations to real-life environments, as many simulated environments cannot fully offer all relevant conditions as a real system. The biggest advantage of a real system is its closeness to the real-life environment in which the tested application is designed to operate in. Real systems are hard to deploy and costly. Especially when talking about automated and accelerated testing, real systems can pose a financial and temporal bottleneck in the testing process.

### ODD Monitoring

Last, a monitoring tool is required for verification and for tracking all parameters that are necessary for and can have some variance on scenarios. By tracking these parameters and verifying the application to be tested, a precise ODD can be defined for the system. With feedback from the monitoring tool, parameters can be adjusted, or new parameters can be chosen for a new iteration of scenario generation. The tools for monitoring in the chain of scenario-based testing can be arbitrarily chosen as long as they are capable of monitoring parameters in real-time for synchronization purposes.

The described method is of an iterative nature. Each component feeds the next with some information. This loop is depicted in Figure 1.

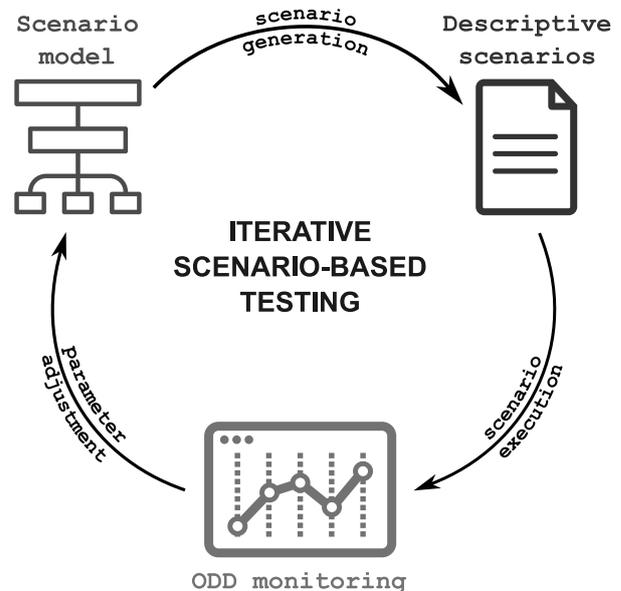


Figure 1: Iterative scenario-based testing

The execution of test scenarios can be accomplished in a simulated environment as well as a real system. Although both approaches are important to consider, the method depicted in Figure 1 is tailored towards testing in simulated environments. For generating application-readable scenario descriptions with the scenario modeling tool, some application, e.g. script, is needed. Similarly, after scenario execution and monitoring, some application is needed which feeds the result logs to the scenario modeling tool, decides on parameter adjustment, and triggers new scenario generation. The use of such intermediate applications and scripts enables high automation and optimization of the method. In ideal circumstances, the iterative scenario-based testing method forms a closed loop with automated test scenario generation, execution, and real-time monitoring of parameters.

### 3 Exemplary Implementation

This section explains an exemplary implementation to demonstrate the derived method. For the implementation, domain-specific tools were selected that can be exchanged depending on the use case. The exemplary implementation of discussed method can be divided into three components: First, the MBSE-based scenario description and generation using Cameo; second, the execution of scenarios defined in generated XML files with the flight simulator FlightGear; and last, the monitoring of parameters during scenario execution with a custom ODD monitoring tool. The basic flow of information and steps are illustrated in Figure 2.

The high-level model of the scenarios is described with a profile diagram in Cameo. Profile diagrams are defined in the System Modeling Language. Additionally, extensions are used to increase the modeling capabilities with profile diagrams. One configuration of a specific scenario is generated with a block definition diagram, which can be transformed and exported into the desired XML scenario files with the help of scripts. XML files are generated for the use case on hand, since FlightGear uses a XML format for the scenario execution. However, other domain-specific formats can be used as well. The scenarios are executed within an instance of FlightGear and the parameters are monitored with a custom ODD monitoring tool. A more detailed description of the implementation is shown in the following subsections.

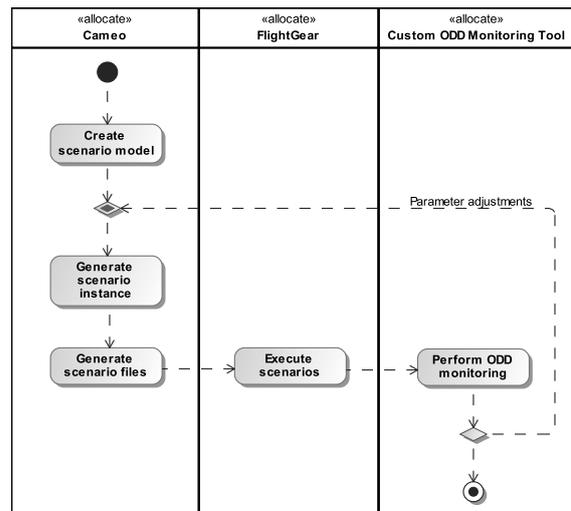


Figure 2: Flow of information and steps for iterative scenario-based testing used in this work

#### 3.1 Scenario Format and MBSE-Based Scenario Generation

A high-level description of the necessary files for scenario execution can be observed in Figure 3. Along with scenario files, flight plan files are needed for scenario execution, as will be explained shortly.

The scenario files include various tags which define the inputs, objects, and attributes when executing them in FlightGear. An important tag is the `<entry>` tag which defines objects used in a scenario and can include the following additional tags: `<callsign>` – the identification of the aircraft, `<type>` and `<model>` – the type and model of the aircraft, `<flightplan>` – the flight plan which the scenario refers to, and `<repeat>` – a Boolean flag that indicates whether the scenario shall be repeated once or infinitely often.

The flight plans, which the scenario files refer to, are also in XML format. The most important tag in the flight plan is the `<wpt>` tag, which can include the following additional tags: `<name>` – the name of the waypoint, `<lat>` – the latitude of the entry that refers to the flight plan, `<lon>` – the longitude, `<alt>` – the altitude, `<ktas>` – the *knots true airspeed*, `<on-ground>` – if the specified object starts from the ground or not, `<gear-down>` – if the landing gear is retracted or extended, and `<flaps-down>` – if the flaps are retracted or extended. FlightGear offers many more configuration files which can be adjusted to change environmental parameters as

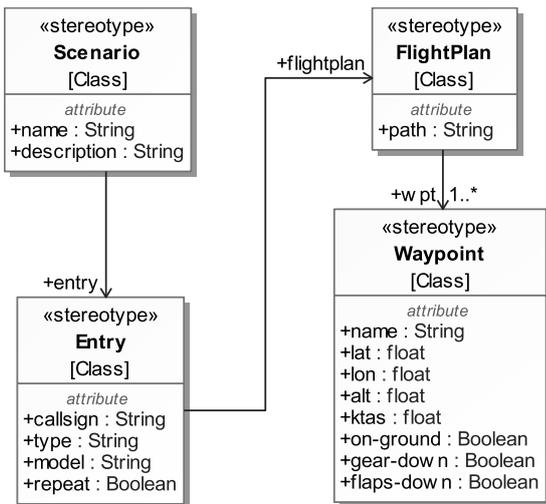


Figure 3: High-level description of the configuration files for FlightGear

well as parameters of entities and other components of interest for scenario-based testing. For simplicity, only the scenario and flight plan files along with their parameters are discussed here. Some high-level description, i.e. metamodel, of the scenario and flight plan files is needed to generate arbitrary test scenarios.

Figure 4 depicts one instance of the high-level description of the scenario and flight plan files.

### 3.2 Scenario Execution

The scenarios are executed within FlightGear. The respective XML files can be executed manually in a FlightGear instance, or referred to as parameters for automatic execution with startup of FlightGear. For automation purposes, we chose the latter. As explained in the previous subsection, one or more entries, e.g. planes, can be defined in a scenario file, with each flying according to a predefined route. In this instance, one passenger airplane is defined, which narrowly passes the user's plane. Figure 5 shows a screenshot of the scenario during execution in FlightGear.

### 3.3 ODD Definition and Monitoring

The ODD defines the conditions under which a system operates properly. Several parameters can have a variance on the scenarios executed in FlightGear, some of which were defined above. Additional parameters, such

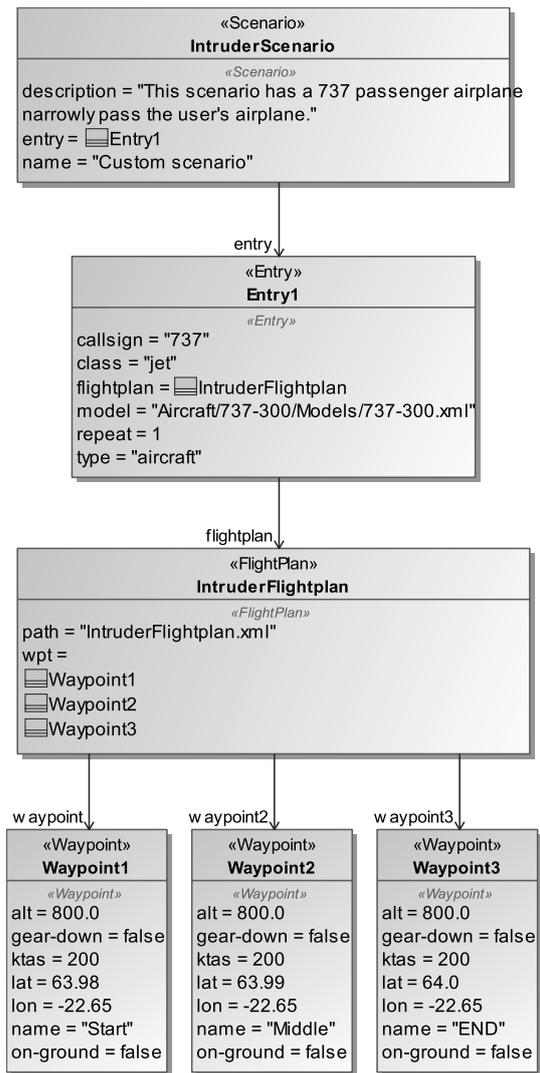


Figure 4: Block definition diagram of one scenario and flight plan configuration



Figure 5: Passenger airplane narrowly passing user's Cessna

as weather conditions, need to be considered. A high-level description of the domain model for the ODD of the AI-based system used on an airplane is depicted in Figure 6.

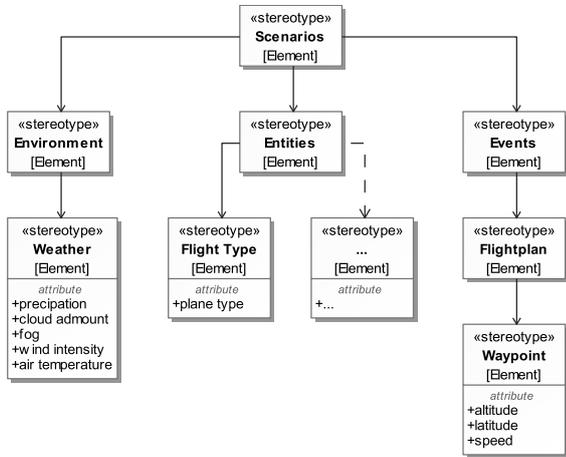


Figure 6: Domain model for the ODD of the scenario-based testing method

The parameter boundaries for the use case of object detection during scenario execution can be determined in an iterative process. Due to the high number of parameters to be considered, a manual exhaustive search for parameter boundaries is highly time-consuming. Therefore, some tool is needed which can track the necessary parameters during scenario execution and give feedback on the result of the tested application.

For monitoring these parameters in FlightGear, a public Python library<sup>3</sup> for fetching parameters from FlightGear’s property tree is used. The AI-based system tested in this example is an object detection application. The result of the domain model’s object detection and desired parameters can be logged using the monitoring tool. The feedback generated from the tool can then be used to adjust the values of selected parameters in Cameo, generate new scenarios, and narrow down the parameters to fit the ODD of the application. This iterative process can be used to narrow down the ODD boundaries of each parameter with every iteration.

A simplified ODD for the system with the two parameters altitude and speed can be defined as follows: “The application performs correct object detection of

<sup>3</sup>Munyakabera Jean Claude, 2022. flightgear\_interface, Available at: [https://github.com/ironmann250/flightgear\\_interface](https://github.com/ironmann250/flightgear_interface)

intruding airplanes of type Boeing 737 within following parameter boundaries:

- Altitude of intruding airplane relative to own airplane in feet,  $\Delta$  alt: -100 to 100.
- Cumulative speed of intruding airplane as well as own airplane in knots true airspeed,  $\Sigma$  ktas: 0 to 500.”

Table 1 shows an exemplary log recorded during execution of a scenario in FlightGear. For completeness and to reflect other relevant parameters currently covered in the scenario model, the latitude and longitude of the intruding airplane are logged along with the aforementioned altitude and speed.

Log #	lat	lon	$\Delta$ alt	$\Sigma$ ktas	detect
1	63.970	-22.65	100	400	no
2	63.974	-22.65	100	399	no
3	63.978	-22.65	100	400	no
4	63.982	-22.65	100	399	yes
5	63.986	-22.65	100	400	yes
6	63.990	-22.65	100	403	yes
7	63.994	-22.65	100	399	no
8	63.998	-22.65	100	400	no

Table 1: Exemplary log of parameters monitored during scenario execution in FlightGear.

The table shows that a successful object detection is on hand for logs four to six. Therefore, the predefined ODD holds for the combination of parameters on hand. Now, single parameters can be adjusted for a potential reevaluation of the predefined ODD. In this case, the altitude of the intruding airplane relative to the own airplane is increased by 100 feet. First, a scenario with a new configuration of attributes needs to be generated, similar to the one depicted in Figure 4. In this case, the altitude is adjusted to reflect the definition for the new test case. Lastly, the necessary XML files are generated from the configuration model. Now, scenario execution in FlightGear and parameter monitoring can be performed. Table 2 shows the log for the second iteration of parameter monitoring.

As shown in the second table, the object detection is successful for logs three to five. The predefined ODD still holds for the combination of parameters on hand. However, the ODD can now be adjusted and phrased

Log #	lat	lon	$\Delta$ alt	$\Sigma$ ktas	detect
1	63.970	-22.65	200	400	no
2	63.974	-22.65	200	399	no
3	63.978	-22.65	200	401	yes
4	63.982	-22.65	200	400	yes
5	63.986	-22.65	200	400	yes
6	63.990	-22.65	200	400	no
7	63.994	-22.65	200	400	no
8	63.998	-22.65	200	400	no

Table 2: Exemplary log of parameters monitored during scenario execution in FlightGear.

more precisely in line with the altered parameter. The ODD for the application can therefore be rephrased as follows:

“[...] within following parameter boundaries:

- Altitude of intruding airplane relative to own airplane in feet,  $\Delta$  alt: -100 to 200.
- [...]”

The loop of parameter adjustment, scenario generation, execution, and monitoring can be repeated until the changes in detection results fall below some predefined value and an ODD with some desired precision has been determined. The example for ODD monitoring and parameter adjustment presented above is simplistic and, for instance, does not consider constraints. Many more parameters can be and need to be considered when defining a precise ODD for the underlying application. Also, the granularity for testing parameter boundaries of the ODD needs to be determined accurately. For instance, a higher logging frequency of parameters can be chosen, which makes the tests more precise but also increases the testing effort. Also, instead of a Boolean for the result of the object detection, the more granular confidence of the object detector from the machine learning application can be used as a metric. The framework in itself requires fine-tuning and more testing to provide the right conditions for successful iterative scenario-based testing of various systems.

The exemplary implementation of model-based scenario generation and ODD monitoring in this section follows the method presented in Figure 1. Domain-specific tools such as Cameo, XML files, and a Python application were used to build a framework for iterative scenario-based testing. The implementation can be seen

as a minimal working example, demonstrating the iterative scenario-based testing method explained in Section 2. The implementation can be developed further to allow for closed-loop scenario-based testing with automated scenario generation, execution, and monitoring.

## 4 Conclusion and Discussion

The use of ML applications in AI-based systems such as airplanes is steadily increasing. The thorough testing of these systems is a fundamental part of their development process. Certain industries, such as aviation, impose strict requirements and constraints for the use of AI-based applications, increasing the testing efforts required to certify and use these applications. Additionally, ML applications are often considered a black box. Therefore, black box testing methods need to be put in place that are as rigorous as current testing methods for common software systems.

This work depicts a method for iteratively testing an AI-based system which performs object detection in an airplane. For this purpose, a scenario-based testing loop was developed, including the three steps of generating application-readable scenario descriptions from models, execution of these scenarios, and parameter monitoring with model parameter adjustments. In addition to generating arbitrary test cases, the presented method illustrates the approximation of boundaries for the ODD of the ML application with iterative parameter adjustments.

This method can be further optimized by connecting its components, i.e. the high-level scenario description, scenario execution, and ODD monitoring, and creating a closed loop with automated scenario generation, execution, and parameter adjustment. Additionally, test oracles that determine the success or failure of individual tests should be investigated. The granularity of test cases, i.e. only success and failure evaluation or more finely grained evaluations, is important. These findings will be investigated in future research.

### Acknowledgement

The presented research is financed by and part of the project *Model-Based Systems Engineering for Artificial Intelligence* (MBSE4AI). We thank everyone who provided support, insight, and expertise that greatly assisted the research.

## References

- [1] EASA. *EASA Concept Paper: First usable guidance for Level 1 machine learning applications*. Tech. rep. Apr. 2021. URL: <https://www.easa.europa.eu/en/downloads/126648/en>.
- [2] EUROCAE WG-114/SAE and G-34 Artificial Intelligence Working Group. *Artificial Intelligence in Aeronautical Systems: Statement of Concerns*. SAE International. Apr. 2021. DOI: <https://doi.org/10.4271/AIR6988>.
- [3] On-Road Automated Driving (ORAD) Committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Standard. SAE International, Apr. 2021. DOI: [https://doi.org/10.4271/J3016\\_202104](https://doi.org/10.4271/J3016_202104).
- [4] *DO-178C - Software Considerations in Airborne Systems and Equipment Certification*. Standard. RTCA, Dec. 2011. URL: <https://www.d0178.org/>.
- [5] A. Wayne Wymore. *Model-Based Systems Engineering*. 1993. ISBN: 9780203746936. DOI: 10.1201/9780203746936.
- [6] Azad M. Madni. “MBSE Testbed for Rapid, Cost-Effective Prototyping and Evaluation of System Modeling Approaches”. In: *Applied Sciences* 11.5 (2021). ISSN: 2076-3417. DOI: 10.3390/app11052321.
- [7] Jasper Sprockhoff et al. “Model-Based Systems Engineering for AI-Based Systems”. In: *AIAA SCITECH 2023 Forum*. Jan. 2023. DOI: 10.2514/6.2023-2587.
- [8] Shafagh Jafer and Umut Durak. “Tackling the Complexity of Simulation Scenario Development in Aviation”. In: *Proceedings of the Symposium on Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems*. Society for Computer Simulation International, 2017. ISBN: 9781510840300.
- [9] Umut Durak et al. “Scenario Development: A Model-Driven Engineering Perspective”. In: *SIMULTECH 2014 - 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*. SCITEPRESS – Science and Technology Publications, 2014, pp. 117–124. URL: <https://elib.dlr.de/94626/>.
- [10] Siddhartha Gupta et al. “From Operational Scenarios to Synthetic Data: Simulation-Based Data Generation for AI-Based Airborne Systems”. In: *AIAA SCITECH 2022 Forum*. Jan. 2022. DOI: 10.2514/6.2022-2103.
- [11] Umut Durak et al. “Using System Entity Structures to Model the Elements of a Scenario in a Research Flight Simulator”. In: *AIAA Modeling and Simulation Technologies Conference*. 2017. URL: <https://elib.dlr.de/112664/>.
- [12] Umut Durak et al. “Computational Representation for a Simulation Scenario Definition Language”. In: *2018 AIAA Modeling and Simulation Technologies Conference*. DOI: 10.2514/6.2018-1398.
- [13] Bikash Chandra Karmokar et al. “Tools for Scenario Development Using System Entity Structures”. In: Jan. 2019. DOI: 10.2514/6.2019-1712.
- [14] Siddhartha Gupta and Umut Durak. “Behavioural Modeling for Scenario-based Testing in Aviation”. In: *AIAA SCITECH 2023 Forum (not yet published)*. Jan. 2023.
- [15] Hazem Torfah et al. “Learning Monitorable Operational Design Domains for Assured Autonomy”. In: *Proceedings of the International Symposium on Automated Technology for Verification and Analysis (ATVA)*. Oct. 2022.
- [16] Daniel Fremont et al. “Formal Scenario-Based Testing of Autonomous Vehicles: From Simulation to the Real World”. In: Sept. 2020, pp. 1–8. DOI: 10.1109/ITSC45102.2020.9294368.
- [17] Francis Indaheng et al. “A Scenario-Based Platform for Testing Autonomous Vehicle Behavior Prediction Models in Simulation”. In: *ArXiv abs/2110.14870* (2021).
- [18] Hardi Hungar. “Scenario space exploration for establishing the safety of automated vehicles”. In: *3rd China Autonomous Driving Testing Technology Innovation Conference, 2020*. Dec. 2020. URL: <https://elib.dlr.de/139626/>.

# Simulation-based Verification of Functions for Autonomous Drones

Hamza Ghezali<sup>2</sup>, Siddhartha Gupta<sup>2</sup>, Umut Durak<sup>1</sup>

<sup>1</sup>Institute of Flight Systems, German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Braunschweig, Germany

<sup>2</sup>Institute of Informatics, Clausthal University of Technology, Julius-Albert-Str. 4, 38678 Clausthal-Zellerfeld, Germany

**Abstract.** The world is witnessing considerable interest and a rapid shift towards the autonomous aerial domain. Aerial robotics brings a lot of solutions to some existing problems and has benefits in many fields, such as transportation of people and goods, navigation, military defence, games, agriculture and so on. Small aerial robots, also known as drones, have critical applications due to their flexibility and suitable small size. Developing these small electronic devices is an arduous task involving careful planning and testing. In addition, Their development and delivery to end users should not only be finished in a shorter amount of time and also with high liability. One of the main concepts to address this issue is using a new testing technique called Behavior Driven Development (BDD). This work uses BDD in component-based software engineering for aerial robotics as an agile technique. This testing approach offers an exciting method to write different possible scenarios in simple, easy-to-read, and understandable feature files. An example of applying this method for drone application development with Robot Operating System and Gazebo is shown in this paper.

## Introduction

We live in a time where "flying robots" are no longer only in science fiction movies. In the beginning, aerial robots were mainly considered for military surveillance, but currently, they have a vital role in targeting and special missions. Over time their involvement in the civilian domain has drastically increased. Some examples are aerial photography, remote sensing, agricultural and wildlife surveys, disaster response, and delivering products. Aerial robots are complex Cyber-Physical Systems (CPS) with various kinds and many sensors, actuators, controllers, and communication systems. Robot Operating Systems (ROS) is a framework for large-scale robotic applications which offers a peer-to-peer communication infrastructure. It uses a central broker

for service registration and discovery [1]. It is becoming a famous architecture for developing software components and aerial robotics. Some of the recent studies on the utilisation of ROS for aerial robotics include ([2]; [3]; [4]; [5]). There are testing strategies developed in the industry for ROS-based robotics software already ([6]; [7]; [8]). Most of them follow the Test Driven Development (TDD) approach, which has its roots in agile methods [9]. Modern software engineering practices tend to remove disconnects among its activities by employing continuous practices to achieve agile processes. After Test-Driven Development (TDD) bridged the gap between implementation and testing, Behavior Driven Development (BDD) aims at the gap between the end-user and the implementation. It establishes a practice based on behavior specifications from the end-user's perspective. It builds upon TDD and promotes a ubiquitous semi-formal language for the specification of behavior that is accessible to all the stakeholders of the system. BDD aims to come up with an executable and a human-readable specification of the system, which can be used for automated acceptance testing. There is a growing interest in the aerial robotics domain towards agile software engineering methods. However, being complex Cyber-Physical Systems, system-level testing of aerial robots requires closed-loop simulation-based approaches [10]. Gazebo is an open-source robot simulator for developing algorithms, designing robots, performing tests, or training AI systems using realistic scenarios ([11]; [12]). Gazebo-ROS integration has also been developed in the last decade [13]. It has then been used also for simulations of aerial robotics ([14]; [15]; [16]; [2]). This paper will present an approach that integrates BDD with Gazebo to enable agile simulation-based verification of ROS-based applications for aerial robotics.

# 1 Background

## 1.1 Behavior Driven Development

BDD is defined as "implementing an application by describing its behavior from the perspective of its stakeholders" [17]. It promotes a ubiquitous semi-formal language for the specification of behaviors that is accessible to all the stakeholders of the system. It is structured around features, which can be defined as the system's capabilities that benefit its users. A feature is usually described in BDD by a title, a brief narrative, and a number of scenarios that serve as acceptance criteria. Scenarios are concrete examples to illustrate the desired behaviors of the system. When the concrete examples are executable, they turn the criteria into an acceptance test. BDD calls this automated acceptance testing. The ubiquitous language idea is based on Evans [18], who stresses that the linguistic divide or the language fracture between the domain expert and the technical team leads to only vaguely described and vaguely understood requirements. Gherkin is the common language for writing features [19]. While it is not a Turing Complete language, it has a grammar enforced by a parser. Each Gherkin feature may contain an arbitrary number of scenarios, which serve as acceptance criteria for the feature. A scenario typically consists of multiple steps describing the actions needed to stimulate the scenario and check the outcome. The steps of a scenario are distinguished into three different classes.

- Given prefaces, a step describes some initial state of the application or the world surrounding the application.
- When introducing a mutation of state on the application or the world. Often this can also be described as the occurrence of an event.
- Then is used to describe an expected result after the arrival of an event in a prior When step.

## 1.2 Behave Framework

Behave [20] is a framework written in python to perform BDD and allows the development of test cases using the Gherkin language. Behave is opiated on source file organisation where a test case is placed inside the directory feature. Multiple features can be put into different files having the extension *.feature*. A feature may have multiple steps, and those step implementations reside in a nested directory called steps. A background consists of a series of steps like scenarios. It allows

you to add some context to the scenarios of a feature. A background is executed before each scenario of this feature but after any before hooks. The background description is for the benefit of humans reading the feature. Again, the background name should be a descriptive yet brief title, illustrating the background operation being performed or the requirement being met. A background section may exist only once within a feature file. In addition, a background must be defined before any scenario or scenario outline. Scenarios describe the discrete behaviours being tested. They are given a title, like backgrounds and the scenario description, for the benefit of humans reading the feature text. Scenario outlines may be used when you have expected conditions and outcomes to go along with your scenario steps. An outline utilises placeholders in the step definition, replaced by values from a table. You may have several example tables in each scenario outline. Steps take a line each and begin with a keyword - one of "Given", "When", "Then", "And", or "But". In a formal sense, the keywords are all Title Case, though some languages allow all-lowercase keywords where that makes sense.

## 1.3 Robot Operating System (ROS)

ROS is an open-source meta-operating system to ease the process of building a robot. ROS shares similarities with traditional operating systems by having process management and scheduling features. "It provides a structured communications layer above the host operating systems of a heterogeneous compute cluster" [21]. ROS is currently one of the most popular platforms for the collaboration of robotics solutions. The key concepts in ROS are nodes, ROS master, topics, messages and services.

**Nodes:** They are the smallest functional unit inside ROS architecture. They execute a particular functionality and help to achieve modularity. They can be thought of as processes. Just as a normal operating system manages processes and their communication, ROS manages nodes and communication between them.

**Messages:** Nodes communicate through messages. ROS has over 200 predefined messages that may contain simple values and more complex data like text, images, etc. Users also can create their messages.

**ROS master:** ROS supports a peer-to-peer network of ROS nodes which communicate through messages. There is an individual node called ROS master, which maintains the registry of all the active nodes in the net-

work. It also contains a parameter server which any node can use to store and access the global state. It acts as a lookup service for information on the nodes and the communication between them. A visual representation of a ROS parameter server can be seen below. The diagram shows that multiple nodes like camera and GNSS are running together in the case of a standard robotic system. The ROS master is aware of all the nodes and their communication. The communication between nodes takes place in two ways:

**Client Based:** This is similar to a client-server architecture where a node will send a request to another node for particular information. The server node will send the required information back to the client. This client-server system can be realised in two ways – services and actions –. The use of services is generally more common compared to actions. The major difference between the two is that services are synchronous, and the client node will wait for the server to send back the information before it continues its process. In contrast, in actions, the client node will not wait for the server response and continue processing.

**Pub-Sub:** Another method for the nodes to communicate with each other is through topics. A topic is a channel which acts as a pipe where a publisher can send information that a subscriber can access. There is no request, but new information is shared until the subscriber node has subscribed to a particular topic. This method of communication is the most common in ROS communication.

## 1.4 Gazebo Simulator

Gazebo is a simulator for robot research that is closely tied to ROS for simulating robots' behaviour ([11];[12]). Its main features include:

- A real-time physics engine which makes use of various technologies like ODE, Bullet, simbody and DART
- High-quality graphics making use of the OGRE engine.
- Rich set of sensors and plugins with noise generation and direct access to the Gazebo API.

The two essential executables of Gazebo are – gzserver and gzclient. The gzserver is responsible for the physics, the engine and sensor generation. The gzclient is responsible for the graphical user interface, the command line interface and the custom applications. The key elements required for a gazebo simulation are worlds, models, and plugins. The world consists of a scene, physics, models, plugins and light in a .world

file using the Simulation Description Format (SDF). It consists of the environment in which Gazebo executes a particular simulation. The gzserver loads it. Models are also described in an SDF, but it includes the description of a single model like a drone. Another vital element is plugins. They define custom behavior - especially at the startup of other elements of the Gazebo simulation - and allow communication interface with external programs like ROS nodes. There are six types of plugins: world, model, sensor, visual, GUI and system, each corresponding to a particular kind of simulation element and its role. Currently, Gazebo supports plugins to be written in C++.

## 2 Simulation Based Verification using BDD and Gazebo

Aerial robots, like other CPS, are composed of networks of computers, sensors, and actuators. The software is usually used coupled with other software, networked sensors, and actuators, so it cannot be tested in isolation. Therefore, the testing process must enclose interaction with other components and the physical environment. Simulation-based testing utilises component and environment virtualisation [22], which enables convenient means of signal manipulation and behavior monitoring. The X-in-the Loop (XiL) testing describes different configurations for simulation-based verification. XiL environments provide interfaces to the system under test. The idea is to embed the system under test in a synthetic environment. Input vectors simulate data signals to an algorithm or a system, and the processed output values are monitored. A unique feature of XiL testing is the possibility to use the same stimulus for different configurations, and the behavior is directly comparable. Two XiL examples would be Software-in-the-Loop (SiL) and Hardware-in-the-Loop (HiL) testing. SiL testing executes the code within a simulated feedback loop, essentially on the development environment. It is non-real-time and targets functional verification. HiL is the testing of the executable on the target platform in a real-time setting. Beyond functional requirements, it also enables the validation of non-functional requirements. A common practice in aerial robotics is to have a dedicated flight control system and a companion computer for onboard mission management. There is a strong trend in the aerial robotics community around open-source flight control systems, particularly PX4 [23]. Its system architecture

allows a native integration to ROS using Micro Air Vehicle Link (MAVLink) [24]. PX4 further provides capabilities to make both SIL and HIL testing with Gazebo using PX4 SITL [25]. Using this infrastructure, executing the features defined in Gherkin using Behave to test ROS nodes with the flight control system and the physics-based simulation in Gazebo is possible.

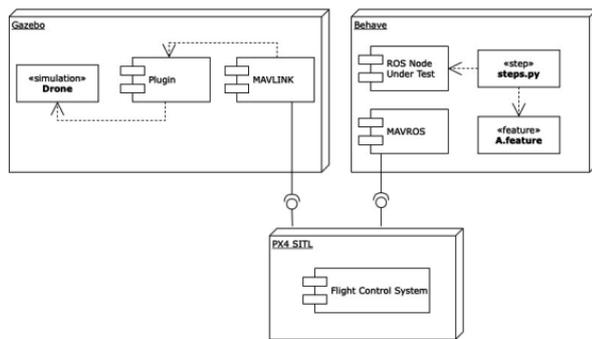


Figure 1: SITL test architecture

Figure 1 presents a simplified representation of the architecture for SIL testing. Typically, all the elements of the SIL test architecture are on the development computer and connected via UDP. PX4 SITL uses Gazebo with a plugin which provides a MAVLink connection. The specific configuration of the PX4 SITL then enables the flight control system to run in a simulation mode using firmware generated explicitly for this environment. Behave runs the step scripts executing the Gherkin feature files on the test harness side and exercises the ROS Node Under Test functions. PX4 supports the integration to ROS via MAVROS [26]. MAVROS is a ROS package that enables the connection of ROS deployments to MAVLink-enabled flight control systems, such as PX4 [27]. The test harness then utilises MAVROS to connect to the flight control system. Based on the test requirements, other ROS nodes or interactions may exist between the test harness and Gazebo. A representative case to demonstrate the utilisation of the proposed approach would be testing a ROS node whose goal is to publish data from a distance sensor. The Hokuyo sensor is used for ground proximity tracking to prevent controlled flight into the terrain. One of the requirements of the Hokuyo ROS node could be written in natural language as follows: *Hokuyo node shall provide the drone altitude above ground level when the drone is flying higher than 0.2 meters and lower than 14 meters above the ground level.* It is then

possible to specify this requirement with three scenarios—examples of these scenarios as Gherkin features are in Figure 9.

## 2.1 Data Flow between ROS nodes

The diagram in figure 2 demonstrates the different ROS nodes, the data flow between them, and how these nodes interact to obtain the desired data. The diagram consists of four nodes:

- 1- ROS Master node:** It represents the node responsible for registering and controlling all the other nodes to facilitate communication between them.
- 2- Laser scan node:** This node publishes the drone altitude data received from Laser
- 3- GPS node:** This node publishes the drone altitude data received from GPS
- 4- Processing node:** This node subscribes to both the drone altitude data received from Laser and GPS and uses this information to control the drone accordingly.

## 3 Implementation steps

### 3.1 First step: Adding Hokuyo LIDAR sensor to the drone

To add the Hokuyo sensor to the drone <sup>1</sup>, the following snippets have been added to the SDF file of the drone. The first is used to link it to the body of the drone, as shown in figure 3; the second shows its topic name, `"/spur/laser/scan"`, as shown in figure 4, and the third contains its characteristics shown in figure 5.

The added sensor was rotated 90 degrees using pose coordinates so that the Hokuyo beams could be directed vertically to detect the ground, as shown in figure 6. The Hokuyo LIDAR publishes its data or distance to obstacles via a rostopic named `"/spur/laser/scan"` as seen in figure 4 in the `topicName` tag under the `plugin` tag.

### 3.2 Second step: Control the drone (Ground detection, change mode, arm, takeoff and land)

#### 3.2.1 Ground detection

There are two ways, in our case, to detect the ground: The Hokuyo sensor sends a range of 1024 beams (i.e. of type array) in different directions, shown in figure 6

<sup>1</sup>Gazebo Add a Sensor to a Robot: <https://classic.gazebosim.org/tutorials>

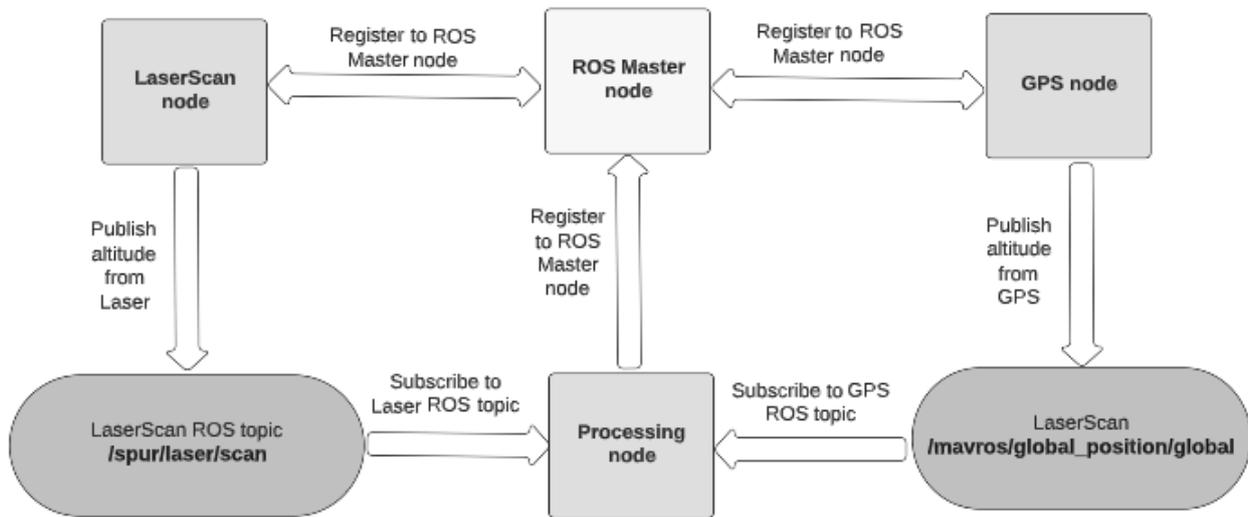


Figure 2: data flow between ROS nodes via ROS topics and services.

```

<joint name="hokuyo_joint" type="fixed">
  <pose>0 0 0.03 0 0 0</pose>
  <parent>base_link</parent>
  <child>hokuyo_link</child>
</joint>

```

Figure 3: The SDF snippet used to attach the Hokuyo LIDAR to the drone's body using the joint tag.

```

<plugin name="hokuyo_node" filename="libgazebo_ros_laser.so">
  <robotNamespace></robotNamespace>
  <topicName>/spur/laser/scan</topicName>
  <frameName>/hokuyo_sensor_link</frameName>
</plugin>

```

Figure 4: The Rostopic name through which Hokuyo publishes its data

as thin blue lines. This work aimed to detect the ground after being hit with some beams sent by the Hokuyo LIDAR.

There are two ways to perform this detection task:

- First method (using the minimum of the 1024 beams): This solution can only work if the drone is flying in a clean environment without obstacles, such as trees, buildings, or other obstacles. Suppose there are any obstacles inside the maximum range of the LIDAR; then, the minimum distance represents the distance to the closest obstacle relative to the drone. For this reason, the second method solves this problem.

- Second method (using the index of the beam directed vertically to the ground):

After a short time, when the drone takes off, all the distances corresponding to the 1024 indexes of the array were printed to figure out which one represents the minimum, knowing there were no obstacles around to avoid mistakes in detecting other objects than the ground. After multiple tests, it was found that the index for hitting the ground vertically is index number 256.

Since the environment is clean and no obstacles exist nearby, we used the `min()` function for ground detection.

### 3.2.2 Change mode, arm, takeoff and land

The first step was to get the published distance LIDAR data sent from the sensor by subscribing to its rostopic using `rospy.Subscriber('/spur/laser/scan', LaserScan, scan_callback, queue_size=10)`. A callback function named "scan\_callback" was defined to implement and write the code based on three conditions (figure 7):

- Distance sensor altitude of the drone less than 0,2m: Three functions corresponding to three commands are called to change the mode to offboard or guided mode, arm the drone, and then takeoff.
- Distance sensor altitude of the drone between 0,2m and 20m: In this case, the drone is considered to be

```

<!--add Lidar-->
<link name="hokuyo link">
  <pose>0 0 0 3.14159 1.52549 3.14159</pose>
  <collision name="collision">
    <pose>0 0 0.3 0 0 0</pose>
    <geometry>
      <box>
        <size>0.1 0.1 0.1</size>
      </box>
    </geometry>
  </collision>
  <visual name="visual">
    <pose>0 0 0.27 0 0 0</pose>
    <geometry>
      <mesh>
        <uri>model://hokuyo/meshes/hokuyo.dae</uri>
      </mesh>
    </geometry>
  </visual>
  <inertial>
    <mass>0.016</mass>
    <inertia>
      <ixx>0.0001</ixx>
      <ixy>0</ixy>
      <ixz>0</ixz>
      <iyy>0.0001</iyy>
      <iyz>0</iyz>
      <izz>0.0001</izz>
    <!-- low inertia necessary to avoid not disturb the drone -->
  </inertia>
</inertial>
  <sensor type="ray" name="laser">
    <pose>0 0 0.3 0 0 1.57</pose>
    <visualize>true</visualize>
    <update_rate>10</update_rate>
    <ray>
      <scan>
        <horizontal>
          <samples>1024</samples>
          <resolution>1</resolution>
          <min_angle>-3.141593</min_angle>
          <max_angle>3.141593</max_angle>
        </horizontal>
      </scan>
      <range>
        <min>0.1</min>
        <max>30</max>
        <resolution>0.1</resolution>
      </range>
    </ray>
    <plugin name="hokuyo_node" filename="libgazebo_ros_laser.so">
      <robotNamespace></robotNamespace>
      <topicName>/spur/laser/scan</topicName>
      <frameName>/hokuyo_sensor_link</frameName>
    </plugin>
  </sensor>
</link>

```

Figure 5: The SDF snippet used to add the Hokuyo LIDAR to the drone using the link tag

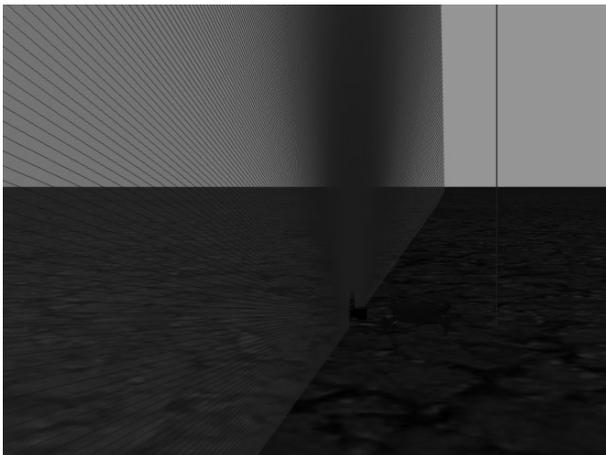


Figure 6: Hokuyo beams directed vertically to detect the ground

flying in the desired range. therefore, no action needs to be taken.

- Distance sensor altitude of the drone higher than 20m: A function corresponding to the land command is called if the received altitude is higher than the maximum altitude of 20m.

### 3.3 Qgroundcontrol as visualisation tool

Qgroundcontrol <sup>2</sup> was used to visualise the status of the drone (figure 8). It is helpful software that can be used to configure PX4 parameters and mission planning. It shows valuable information about the drone, such as altitude, location, connection status, and errors as notifications (written as well as audio notifications), which helps the user or the developer to detect and see errors and problems hidden in the code and correct them very quickly.

### 3.4 Third step: Gherkin scenarios in a feature file and their implementation as steps in python

As represented in figure 9, the feature file contains different scenarios; each scenario consists of three main keywords: Given, When and Then. The meaning of these are briefly explained below <sup>3</sup>:

- Given: It is represented by the keyword Given in the .feature file. It means under a certain situation or state
- When: It is represented by the keyword When in the .feature file. It means when a certain condition or event is met
- Then: It is represented by the keyword Then in the .feature file. It means a specific outcome must occur as a result.

#### 3.4.1 Implementation of "Given" Hokuyo LIDAR distance sensor node running

//rework In this work, the initial situation is to check whether the Hokuyo LIDAR node is running i.e. it is publishing the LIDAR distance data. A bash script checks whether or not the LIDAR topic is publishing data (figure 11). In other words, it appears in the output of the "rostopic list" command. Searching the output of the latter command and grepping the Hokuyo LIDAR

<sup>2</sup>QGroundControl User Guide: <https://docs.qgroundcontrol.com/master/en/>

<sup>3</sup>Gherkin Reference: <https://cucumber.io/docs/gherkin/reference/>

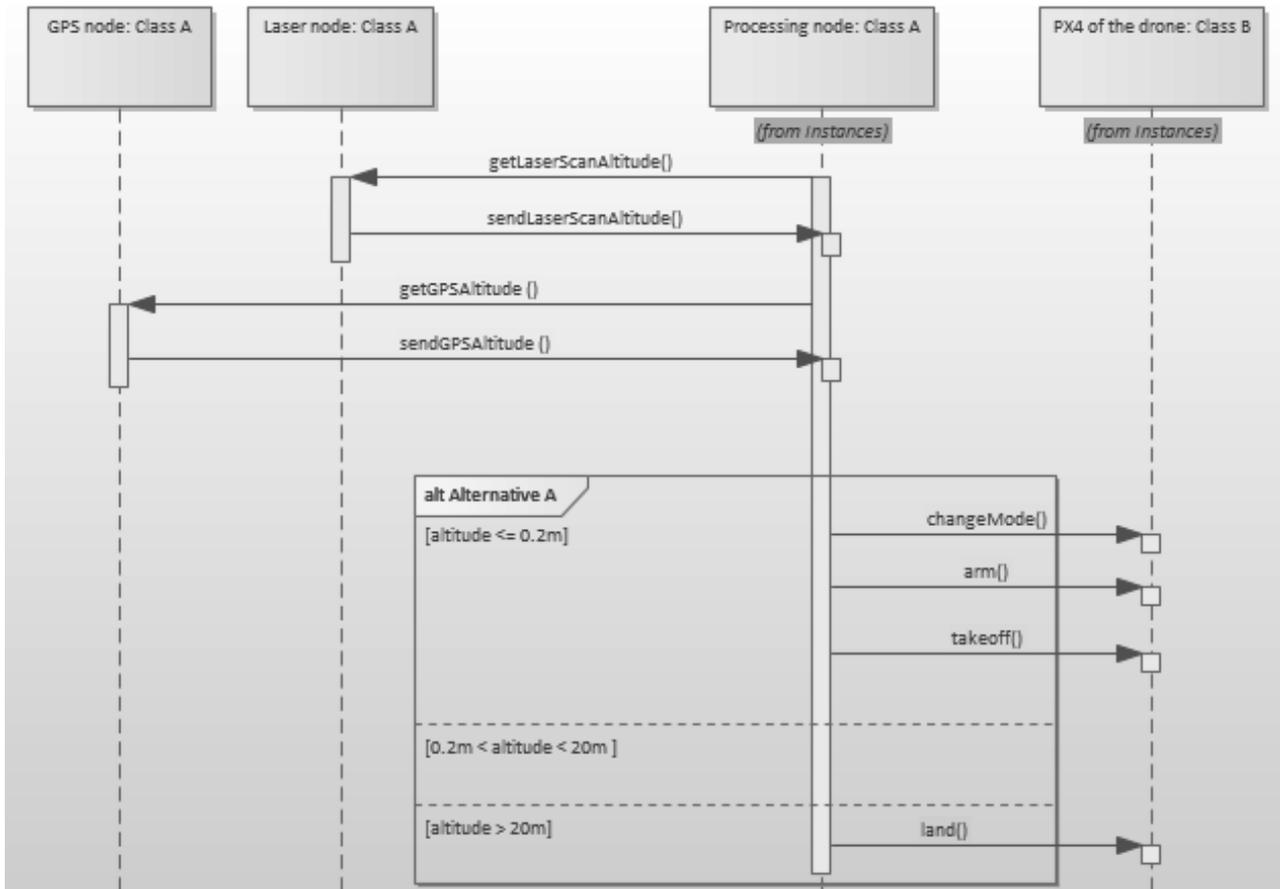


Figure 7: Sequence diagram showing the flow of commands to control the drone.



Figure 8: Picture of the QGroundControl GUI

topic name `"/spur/laser/scan"`. If the script finds it exists successfully with 0, otherwise, it exits with another value and the checking for the node running test fails.

### 3.4.2 implementation of "When" events (three event possibilities) and "Then" their expected corresponding outcomes

The task was to check and test three possibilities of the drone altitude:

- First event when the drone is under 0.2m: In this case, the altitude received from the Hokuyo sensor must be minus infinity
- Second event when the drone is flying between 0.2m and 14m: In this case, the difference between the altitude received from the Hokuyo sensor and the one obtained from GPS must be less than 0.30m
- Third event when the drone is flying above 14m: In this case, the altitude received from the Hokuyo sensor must be plus infinity

```

Feature: Precision altimeter with Hokuyo sensor

Scenario: Lower than 0.2 meters AGL
Given Hokuyo sensor ROS node is running
When The drone is at an altitude less than 0.2 meters above ground level
Then Range Message of Hokuyo sensor ROS node is minus infinity.

Scenario: Lower than 14 meters and higher than 0.2 meters AGL
Given Hokuyo sensor ROS node is running
When The drone is flying lower than 14 meters and higher than 0.2 meters above ground level
Then The difference between the drone's altitude above ground level and Range Message of Hokuyo
sensor ROS node is less then 0.30 meters.

Scenario: Higher than 14 meters AGL
Given Hokuyo sensor ROS node is running
When The drone is flying higher than 14 meters above ground level
Then Range Message of Hokuyo sensor ROS node is plus infinity.

```

Figure 9: Gherkin scenarios.

## 4 Results and Discussion

As shown in the output of the behave tests for the 3 cases mentioned before (lower than 0,2m, between 0,2m and 14m, and higher than 14m), only one event test case should run successfully at a time. The other two test cases must fail because the drone can only be in one of the 3 cases mentioned above (figure 10).

// Rewrite For the Test whether the Hokuyo node is publishing its topic, it can be seen in the output that the test runs successfully for ALL 3 cases (first line of each event test is green) regardless in which event the drone is, which is an expected result since the publication of distance topic is independent from the altitude of the drone.

For the sake of clarification, the output of the tests can be divided into three main sections:

- Given node running and publishing altitude and the drone altitude lower than 0,2m: As shown on the top of the behave test output, the corresponding event and outcome tests are in green (figure 12) and whereas the context and outcome are in red for the other 2 test cases
- Given node running and publishing altitude and the altitude of the drone between 0,2m and 14m: For the middle case of the behave test output, the corresponding event and outcome tests are in green (figure 13) and whereas the event and outcome are in red for the other 2 test cases

- Given node running and publishing altitude and the drone altitude higher than 14m: For the bottom case of the behave test output, the corresponding event and outcome tests are in green (figure 14) and whereas the event and outcome are in red for the other 2 test cases

## 5 Conclusion and Future Work

In conclusion, it can be said that integration of ROS, Gazebo, PX4 Autopilot, and QGroundcontrol provides an efficient way to simulate cyber-physical systems, such as the drone example we used in this project, and allows for efficient and smooth communication between these different components. Concerning the testing part, the BDD framework was a handy tool that facilitated our tests with SITL, including ROS, Gazebo and the drone PX4 autopilot firmware. It is a user-friendly framework that provides an easy-to-understand high-description language for all involved stakeholders, even those without a technical background. In this work, a Hokuyo LIDAR sensor was used, which is limited in its performance because it sends beams in all directions around it. It is suitable for object detection in general, but for detecting a specific object in a particular desired direction, like in our situation, using a directed Laser would be more efficient, which could improve the precision of the received data. Moreover, the goal of this project was mainly for learning purposes using simula-

```

OK laser scan is in rostopic list
Feature: Precision altimeter with TerraRanger One # ../drone.feature:3

Scenario: Lower than 0.2 meters AGL # ../drone.feature:5
  Given TerraRanger One ROS node is running # drone_steps.py:11 0.653s
  When The drone is at an altitude less than 0.2 meters above ground level # drone_steps.py:18 0.287s
  Then Range Message of TerraRanger One ROS node is minus infinity. # drone_steps.py:25 0.698s

Scenario: Lower than 14 meters and higher than 0.2 meters AGL
:10
  Given TerraRanger One ROS node is running
1 0.683s
  When The drone is flying lower than 14 meters and higher than 0.2 meters above ground level
2 0.100s
  Traceback (most recent call last):
    File "/home/lowe/.local/lib/python3.6/site-packages/behave/model.py", line 1329, in run
      match.run(runner.context)
    File "/home/lowe/.local/lib/python3.6/site-packages/behave/matchers.py", line 98, in run
      self.func(context, *args, **kwargs)
    File "drone_steps.py", line 37, in drone_flying_lower_thanb_14m_and_highen_than_0dot2m
      assert low_altitude_threshold < drone_terranger_altitude < high_altitude_threshold
  AssertionError

  Then The difference between the drone's altitude above ground level and Range Message of TerraRanger One ROS node is less then 0.30 meters.

Scenario: Higher than 14 meters AGL # ../drone.feature:15
  Given TerraRanger One ROS node is running # drone_steps.py:11 0.668s
  When The drone is flying higher than 14 meters above ground level # drone_steps.py:48 0.096s
  Traceback (most recent call last):
    File "/home/lowe/.local/lib/python3.6/site-packages/behave/model.py", line 1329, in run
      match.run(runner.context)
    File "/home/lowe/.local/lib/python3.6/site-packages/behave/matchers.py", line 98, in run
      self.func(context, *args, **kwargs)
    File "drone_steps.py", line 52, in drone_flying_higher_than_14m
      assert drone_terranger_altitude > high_altitude_threshold
  AssertionError

  Then Range Message of TerraRanger One ROS node is plus infinity. # None

Failing scenarios:
../drone.feature:10 Lower than 14 meters and higher than 0.2 meters AGL
../drone.feature:15 Higher than 14 meters AGL

0 features passed, 1 failed, 0 skipped
1 scenario passed, 2 failed, 0 skipped
5 steps passed, 2 failed, 2 skipped, 0 undefined
Took 0m2.586s
    
```

Figure 10: Test output when drone altitude less than 0,2m

```

# HOKUYO LIDAR SENSOR ROSTOPIC PUBLISHING
@given('TerraRanger One ROS node is running')
def terranger_ros_node_running(nodeCheckScriptExitValue):
    nodeCheckScriptExitValue= os.system('sh ./node_running_script.sh')
    assert nodeCheckScriptExitValue==0
    
```

Figure 11: Python snippet to check that Hokuyo sensor is publishing its data successfully

```

Scenario: Lower than 0.2 meters AGL
  Given TerraRanger One ROS node is running
  When The drone is at an altitude less than 0.2 meters above ground level
  Then Range Message of TerraRanger One ROS node is minus infinity.
    
```

Figure 12: Zoomed in output, drone altitude < 0,2m

```

Scenario: Lower than 14 meters and higher than 0.2 meters AGL
  Given TerraRanger One ROS node is running
  When The drone is flying lower than 14 meters and higher than 0.2 meters above ground level
  Then The difference between the drone's altitude above ground level and Range Message of TerraR
  s less then 0.30 meters. # drone_steps.py:48 0.210s
    
```

Figure 13: Zoomed in output, 0,2m < drone altitude < 14m

```

Scenario: Higher than 14 meters AGL
  Given TerraRanger One ROS node is running
  When The drone is flying higher than 14 meters above ground level
  Then Range Message of TerraRanger One ROS node is plus infinity.
    
```

Figure 14: Zoomed in output, drone altitude > 14m

tion in the loop (SITL). For more practical projects with the hardware (HIL), choosing a high-performance dis-

tance sensor like Teranger one <sup>4</sup> would give better and more precise results.

<sup>4</sup>TeraRanger One - the lightweight and low-cost ToF distance measurement sensor, 14m, 8grams: <https://www.terabee.com/sensors-modules/lidar-tof-range-finders/>

## References

- [1] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, Andrew Ng. "ROS: an open-source Robot Operating System". In ICRA workshop on open source software 2009 May 12 (Vol. 3, No. 3.2, p. 5).
- [2] Gupta, S., and U. Durak. 2020. "RESTful Software Architecture for ROS-based Onboard Mission System for Drones". In AIAA Scitech 2020 Forum, 0239.
- [3] Sankhe, C., B. Ahuja, A. Coutinho, C. Bhangale, and N. Giri. 2020 "Implementation of ROS in Drones for Animate and Inanimate Object Detection". In Advanced Computing Technologies and Applications, 579–589. Springer.
- [4] Honig, W., and N. Ayanian. 2017. "Flying multiple UAVs using ROS". . In Robot Operating System (ROS), 83–118. Springer.
- [5] Lamping, A. P., J. N. Ouwerkerk, and K. Cohen. 2018. "Multi-UAV Control and Supervision with ROS". . In 2018 Aviation Technology, Integration, and Operations Conference, 4245.
- [6] Fabresse, L., J. Laval, and N. Bouraqadi. 2013. "Towards test-driven development for mobile robots". . In Proceedings of the ICRA 2013 Workshop on Software Development and Integration in Robotics (SDIR VIII).
- [7] Dieber, B., R. White, S. Taurer, B. Breiling, G. Caiazza, H. Christensen, and A. Cortesi. 2020. "Penetration testing ROS". . In Robot Operating System (ROS), 183–225. Springer.
- [8] Paikan, A., S. Traversaro, F. Nori, and L. Natale. 2015. "A generic testing framework for test driven development of robotic systems". . In International Workshop on Modelling and Simulation for Autonomous Systems, 216–225. Springer.
- [9] Beck, K. 2003. Test-driven development: by example. Addison-Wesley Professional. Bernardeschi, C., A. Fagiolini, M. Palmieri, G. Scrima, and F. Sofia. 2018. "Ros/gazebo-based simulation of co-operative uavs". . In International Conference on Modelling and Simulation for Autonomous Systems, 321–334. Springer.
- [10] Lee, E. A. 2010. "CPS foundations". In Design Automation Conference, 737–742. IEEE. Meier, L., D. Honegger, and M. Pollefeys. 2015. "PX4: A node-based multithreaded open-source robotics framework for deeply embedded platforms". . In 2015 IEEE international conference on robotics and automation (ICRA), 6235–6240. IEEE.
- [11] Gazebo Robotics Foundation 2014. "Gazebo, robot simulation made easy". . <http://gazebo.org/>. Accessed: 2021-03-01.
- [12] Koenig, N., and A. Howard. 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), Volume 3, 2149–2154. IEEE.
- [13] Takaya, K., T. Asai, V. Kroumov, and F. Smarandache. 2016. "Simulation environment for mobile robots testing using ROS and Gazebo". . In 2016 20th International Conference on System Theory, Control and Computing (ICSTCC), 96–101. IEEE.
- [14] Meyer, J., A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. Von Stryk. 2012. "Comprehensive simulation of quadrotor uavs using ros and gazebo". . In International conference on simulation, modeling, and programming for autonomous robots, 400–411. Springer.
- [15] Bernardeschi, C., A. Fagiolini, M. Palmieri, G. Scrima, and F. Sofia. 2018. "Ros/gazebo based simulation of co-operative uavs". . In International Conference on Modelling and Simulation for Autonomous Systems, 321–334. Springer.
- [16] Sciortino, C., and A. Fagiolini. 2018. "ROS/Gazebo-Based Simulation of Quadcopter Aircrafts". . In 2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI), 1–6. IEEE. Takaya, K., T. Asai, V. Kroumov, and F. Smarandache. 2016. "Simulation environment for mobile robots testing using ROS and Gazebo". In 2016 20th International Conference on System Theory, Control and Computing (ICSTCC), 96–101. IEEE.
- [17] Chelimsky, D., D. Astels, B. Helmkamp, D. North, Z. Dennis, and A. Hellesoy. 2010. "The RSpec Book: Behaviour Driven Development with Rspec". . Cucumber, and Friends, Pragmatic Bookshelf 3:25. Dieber, B., R. White, S. Taurer, B. Breiling, G. Caiazza, H. Christensen, and A. Cortesi. 2020. "Penetration testing ROS". In Robot Operating System (ROS), 183–225. Springer.
- [18] Evans, E. 2004. *Domain-driven design: tackling complexity in the heart of software*. . Addison-Wesley Professional.
- [19] Wynne, M., A. Hellesoy, and S. Tooke. 2017. *The cucumber book: behaviour-driven development for testers and developers*. Pragmatic Bookshelf. .
- [20] Rice, Benno and Jones, Richard and Engel, Jens 2017. "Welcome to behave!". . <https://behave.readthedocs.io/en/stable/>. Accessed: 2021-03-01

- [21] Bpin, K. 2018. *Robot Operating System Cookbook*. Packt Publishing. .
- [22] Koch, C. B., U. Durak, and D. Muller. 2018. “Simulation-based verification for parallelization of model-based “ applications”. . In Proceedings of the 50th Computer Simulation Conference, 1–10.
- [23] Meier, L., D. Honegger, and M. Pollefeys. 2015. “PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms”. . In 2015 IEEE international conference on robotics and automation (ICRA), 6235–6240. IEEE.
- [24] Koubaa, A., A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui. 2019. “Micro air vehicle link (mavlink) in a nutshell: A survey”. . IEEE Access 7:87658–87680.
- [25] PX4 2021a. “Gazebo Simulation”. . <https://docs.px4.io/master/en/simulation/gazebo.html>. Accessed: 2021- 03-15.
- [26] PX4 2021b. “ROS (Robot Operating System)”. . <https://docs.px4.io/master/en/ros/>. Accessed: 2021-03-15.
- [27] PX4 2021c. “ROS with MAVROS Installation Guide”. . [https://docs.px4.io/master/en/ros/mavros installation.html](https://docs.px4.io/master/en/ros/mavros/installation.html). Accessed: 2021-03-15.



# Grundlagen und Methoden in Modellbildung und Simulation

## Basic Principles and Methods in Modelling and Simulation

Virtuelle Stochastische Sensoren – Verhaltensrekonstruktion mit verallgemeinerten Modellen (A)

*Claudia Krull*

Ein Gütemaß für Virtuelle Stochastische Sensoren - Beobachtbarkeit Typ-3 (L)

*Pascal Krenckel, Claudia Krull*

Konzept zur simulationsbasierten dynamischen Layoutplanung für eine Matrixproduktion (K)

*Tolga Turmaz, Nils Kalbe, Felix Borgmann*

Integrating Reinforcement Learning and Discrete Event Simulation Using the Concept of Experimental Frame: A Case Study With MATLAB/SimEvents (L)

*Thorsten Pawletta, Jan Bartelt*

Dynamic Time Warping und Synthesedaten zur Validierung von Seq2Seq für die Simulation (L)

*Benjamin Wörrlein, Steffen Straßburger*

From Classic Simulation Circle to Data-Integrated Integrated Simulation (A)

*N. Popper, F. Breitenecker, M. Bicher, B. Glock, I. Hafner, M. Mujica Mota, G. Mušić, T. Pawletta, C. Rippinger, M. Rössler, G. Schneckenreither, C. Urach, M. Wastian, G. Zauner, M. Zechmeister*

Monte Carlo Simulation zur Untersuchung von Gruppenentscheidungen und deren Konsensbildung (A)

*Jana Görs, Graham Horton*

The Principle of Solving a Mixed Boundary Value Problem with the help of Hierarchical Matrices (A)

*Christina Schwarz*

Simulation und Maschinelles Lernen in Supply Chains (K)

*Marc Münnich, Florian Zumpe, Pierre Grzona*

Simulationsbasierte Untersuchung von Energieflexibilität bei der Optimierung in der Hinterlandentwässerung (L)

*Marvin Hempel, Jens Heger*

Langbeiträge (L) sind zu finden im

**ARGESIM Report 21 Proceedings Langbeiträge ASIM Workshop 2023 STS/GMMS/EDU**

ISBN ebook 978-3-903347-61-8, DOI 10.11128/arep.21, ARGESIM Report 21, ASIM Mitteilung 185

Kurzbeiträge (K) und Abstract-Beiträge (A) sind zu finden im

**ARGESIM Report 22 Kurzbeiträge & Abstract-Beiträge ASIM Workshop 2023 STS/GMMS/EDU**

ISBN ebook 978-3-903347-62-5, DOI 10.11128/arep.22, ARGESIM Report 22, ASIM Mitteilung 186

Abstracts der Hauptvorträge sind im **ARGESIM Report 21** und im **ARGESIM Report 22** zu finden



# Ein Gütemaß für Virtuelle Stochastische Sensoren - Rekonstruktionsgüte Typ-3

Pascal Krenckel\*, Claudia Krull

Institut für Simulation und Graphik, Otto-von-Guericke-Universität Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Deutschland; \*krenckel@ovgu.de

**Abstract.** Virtuelle Stochastische Sensoren wurden an der Otto-von-Guericke-Universität Magdeburg entwickelt, um partiell-beobachtbare diskrete stochastische Modelle zu analysieren und das Systemverhalten zu rekonstruieren. Dabei existiert noch keine Beschreibung der Güte eines Virtuellen Stochastischen Sensors. In diesem Paper stellen wir ein Verfahren zur Bestimmung der Rekonstruktionsgüte vor. Dabei wird für zwei Modelle die Güte bestimmt. Diese Modelle wurden zusätzlich verändert, und die Auswirkung auf die Güte mit der erwarteten Güte verglichen. Unsere Ergebnisse zeigen, dass schon nach wenigen Zeitschritten ein Unterschied zwischen der Güte der verschiedenen Versionen der Modelle zu erkennen ist. Mit einem Gütemaß ist es möglich, verschiedene Sensorsetups zu vergleichen und diese zu optimieren.

## Einführung

Virtuelle Stochastische Sensoren (VSS) können benutzt werden, um das Verhalten von partiell-beobachtbaren diskreten stochastischen Systemen zu rekonstruieren. Ähnlich zu den Virtuellen Sensoren [1, 2] messen VSS Größen, die nicht direkt messbar sind. Zum Beispiel kann ein Beschleunigungssensor benutzt werden, um die Geschwindigkeit abzuleiten. Auch ein VSS leitet das Systemverhalten aus einer gegebenen Sequenz von Beobachtungen ab – eine Zeitreihe physischer Sensordaten. Im Gegensatz zu Virtuellen Sensoren sind die Beziehungen zwischen Systemverhalten und Beobachtungen bei VSS nicht analytischer, sondern stochastischer Natur. Ein großes Problem ist dabei die Rekonstruktionsqualität des VSS. In diesem Paper stellen wir eine neue Definition der Rekonstruktionsgüte Typ-3 vor - ein Maß der Rekonstruktionsgüte eines VSS für die Pfadrekonstruktion. Durch die Rekonstruktionsgüte kann nicht nur die Anwendbarkeit eines VSS bestimmt werden, sondern auch verschiedene Sensorsetups verglichen werden. Dies ermöglicht es, das Sensorsetup zu optimieren.

## 1 Hintergrund

### 1.1 Virtuelle Stochastische Sensoren

VSS wurden 2011 eingeführt. 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“. Die häufig verwendeten Hidden Markov Modelle (HMM) sind auch VSS [3, 4]. Es können aber auch Modelltypen mit gedächtnisbehaftetem Verhalten wie Augmented Stochastic Petri Nets (ASPN) verwendet werden. Ein VSS besteht immer aus einem Modell und einem Lösungsverfahren. Dabei gibt es drei Zielstellungen: Evaluierung, Decoding und Training. [3, 5]

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 (Spur) die Sequenz der internen Zustände (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. [4]

Beim Training wird versucht, das Parameterset eines Modells zu finden, das die höchste Wahrscheinlichkeit hat eine gegebene Menge von Beobachtungen zu erzeugen. Für gedächtnisbehaftete Modelle mit diskreten Zustandsräumen existieren bereits Lösungsverfahren die das Evaluierungsproblem und das Decodingproblem lösen können. Die Proxelmethode 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. [6, 7, 8, 9]

## 1.2 Beobachtbarkeit

Der Begriff Beobachtbarkeit wurde von Kálmán im Kontext von linear dynamischen Systemen eingeführt. Beobachtbarkeit beschreibt dort, ob sich der Systemzustand rekonstruieren lässt. Ist der Systemzustand bekannt, können für jeden nachfolgenden Zeitpunkt die Zustandsvariablen bestimmt werden. Hat der Ausgabevektor weniger Dimensionen als der Eingabevektor, so wird mehr als ein Messwert benötigt, um den Systemzustand zu rekonstruieren. [10, 11, 12]

Für VSS ist diese Definition nicht geeignet. Der Begriff der Beobachtbarkeit soll im Kontext der VSS ein Maß für die Rekonstruktionsgüte des VSS sein und nicht nur ein binäres Maß. In [13] wurde erstmals die Beobachtbarkeit und damit ein Maß für die Güte eines VSS definiert. In früheren Publikationen haben wir den Begriff Beobachtbarkeit verwendet. Um Verwechslungen mit dem binären Maß aus der Regelungstechnik zu vermeiden, haben wir uns dazu entschieden, den Begriff in Rekonstruktionsgüte umzubenennen.

Im normalen Anwendungsfall ist die Ground-Truth, die echte Sequenz der internen Zustände, nicht bekannt. Die Aufgabe des VSS ist es, genau diese Sequenz zu berechnen. Im praktischen Einsatz, wäre es hilfreich, wenn insbesondere vor der Installation des Sensors bestimmt werden kann, welche Qualität der Sensor haben wird. Genau zur Lösung dieses Problems wurde die Rekonstruktionsgüte im Kontext der Virtuellen Stochastischen Sensoren eingeführt. Die Rekonstruktionsgüte wurde dabei in drei Typen unterteilt, die in den folgenden Abschnitten kurz erläutert werden. Sowohl in den vorangegangenen Arbeiten als auch für diese Arbeit wird angenommen, dass sowohl die Zeit als auch der Zustandsraum diskret sind. Ist dies nicht der Fall, so muss das Modell vorher diskretisiert werden. [13, 14]

**Detektierbarkeit.** Die Rekonstruktionsgüte Typ-1 beschreibt mit welcher Sicherheit ein auftretendes Ereignis beobachtet werden kann. Art und Eindeutigkeit werden dabei nicht betrachtet. Dadurch ist die Rekonstruktionsgüte Typ-1 leicht zu berechnen. Allerdings schränkt dies die Aussagekraft auch stark ein. So ist es in seltenen Fällen sogar möglich, dass zusätzliche Ausgaben die Rekonstruktionsgüte erhöhen, aber die Güte des Virtuellen Stochastischen Sensors senken.

Eine Typ-1 Rekonstruktionsgüte von 100% hat dennoch eine besondere Bedeutung. In diesem Fall ist jeder Zustandswechsel detektierbar. Dadurch ist ein Event-basierter Ansatz zum Lösen möglich. Die von Robert Buchholz entwickelten CHMMs sind genau so definiert, und ermöglichen die Verhaltensrekonstruktion partiell beobachtbarer diskreter stochastischer Systeme mit kontinuierlichem Zeitverlauf. Diese Methode ist um ein vielfaches schneller und genauer als das allgemeine VSS Lösungsverfahren. [13, 14, 15]

**Zustandsrekonstruktion.** Die Rekonstruktionsgüte Typ-2 löst das Problem der schlechten Aussagekraft der Rekonstruktionsgüte Typ-1. Die Rekonstruktionsgüte Typ-2 ist ein direktes Maß für die Güte der Rekonstruktion des internen Zustandes und ist definiert über die Summe aller Quadrate der Zustandswahrscheinlichkeiten. Die aktuelle Definition nutzt keine Ähnlichkeitsmaße, Zustände sind entweder identisch oder nicht. Eine exakte Berechnung der Rekonstruktionsgüte Typ-2 ist allerdings in der Praxis nicht möglich. Sowohl der Zustandsraum als auch die Anzahl der Spuren steigen exponentiell mit der Anzahl der Zeitschritte an. [13]

Die Rekonstruktionsgüte Typ-2 ist ein gutes Maß für die Güte der Zustandsrekonstruktion. Ist allerdings eine Sequenz von Zuständen oder Ereignissen von Interesse, dann ist die Rekonstruktionsgüte Typ-2 nicht anwendbar.

**Pfadrekonstruktion.** Da die Rekonstruktionsgüte Typ-2 für Aussagen über die Rekonstruktionsgüte der Zustands- oder Ereignissequenzen nicht anwendbar ist, wurde die Rekonstruktionsgüte Typ-3 eingeführt. Die in [13] vorgestellte Definition beschreibt die Rekonstruktionsgüte Typ-3 als erwartete Distanz zwischen allen Pfaden. Dafür werden entsprechende Distanzmaße benötigt. Welche dies sind, ist stark problemabhängig. Sowohl in [13] als auch in dieser Arbeit beschränken wir uns auf die Hamming-Distanz und die Damerau-Levenshtein-Distanz. Vorteil der Verwendung von Distanzmaßen ist, dass dadurch Ähnlichkeiten abgebildet werden können. Kleine Unterschiede in den Pfaden können so auch kleine Unterschiede in der Distanz zur Folge haben.

Allerdings gilt auch hier, dass in der Praxis eine exakte Berechnung der Rekonstruktionsgüte Typ-3 nicht möglich ist. Sowohl die Anzahl der Pfade, als auch die Anzahl der Spuren wächst exponentiell mit der Anzahl der Zeitschritte an. Schon nach wenigen Zeitschritten ist die Berechenbarkeitsgrenze erreicht. Eine Abschätzung mit der Monte-Carlo-Methode ist nicht möglich, da es nicht trivial ist, zwei Pfade mit gleicher Spur zu sampeln.

Zudem ist die Rekonstruktionsgüte Typ-3 über die Differenz aller Pfade zueinander definiert. In der Realität ist allerdings nur die Distanz zwischen dem Ground-Truth Pfad und dem berechneten Pfad von Interesse. Die Distanz zwischen zwei unwahrscheinlichen Pfaden ist nicht von Bedeutung. [6, 13, 14] Aus diesem Grund stellen wir hier eine neue Definition der Rekonstruktionsgüte Typ-3 vor, die nicht nur relevanter für den letztendlichen Anwendungsfall ist, sondern auch durch die Monte-Carlo-Methode abgeschätzt werden kann.

### 1.3 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 und 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 sinnvoll verglichen werden können. Vorteil der Hamming-Distanz ist, dass sie sehr einfach und effizient zu berechnen ist.

Die Damerau-Levenshtein-Distanz ist die Anzahl der Operationen, die es benötigt, um die Zeichenketten ineinander umzuformen. Die Operationen sind: Einfügen, Ersetzen, Löschen eines Zeichens, sowie das Tauschen zweier benachbarter Zeichen. Die kleinste benötigte Anzahl an Operationen entspricht der Distanz. [16, 17] 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. Abbildung 1 zeigt diesen Fall.

A	B	A	A	B	A	B	A	B	B	B	A	B	A	A	A	B	A	-
A	B	A	A	B	A	B	A	B	A	B	A	B	A	A	A	B	A	A

**Abb. 1:** Zwei Zeichenketten mit Damerau-Levenshtein-Änderung (rot) und Hamming-Abweichung (gelb)

Die Damerau-Levenshtein Distanz in Abbildung 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

VSS und der Berechnung von Ereignissequenzen treten solche Unterschiede sehr häufig auf.

Ein Nachteil der Damerau-Levenshtein-Distanz ist, dass die Laufzeit der Berechnung in  $O(n^2)$  liegt. Zudem sind beide Distanz-Maße nicht direkt vergleichbar, da es auf den Anwendungsfall oder die Fragestellung ankommt.

## 2 Neu Definition der Rekonstruktionsgüte Typ-3

Die Rekonstruktionsgüte Typ-3 versucht die Qualität des rekonstruierten Pfades eines VSS zu quantifizieren, vor dem tatsächlichen Einsatz. In [13] wurde die Rekonstruktionsgüte Typ-3 als erwartete Distanz zwischen allen möglichen Pfaden definiert. Die neu definierte Rekonstruktionsgüte Typ-3 ist die erwartete Distanz zwischen allen Pfaden und dem rekonstruierten Pfad.

Damit die Rekonstruktionsgüte Typ-3 in dem Bereich von  $[0;1]$  liegt wird die berechnete Distanz zweier Pfade zusätzlich durch einen Normierungsfaktor geteilt und die Differenz zu 1 gebildet. Für die Hamming und Damerau-Levenshtein-Distanz entspricht der Normierungsfaktor der Pfadlänge.

$$O_3(t) = 1 - \frac{E(\text{Distanz}_{\text{Decoding}})}{N(t)}$$

$E(\text{Distanz}_{\text{Decoding}})$  ist die erwartete Distanz zwischen Ground-Truth und dem decodierten Pfad, also der erwarteten Distanz zwischen der Realität und dem VSS Output. Daher ist die neue Definition semantisch passender als die bisherige.

Der Erwartungswert  $E(\text{Distanz}_{\text{Decoding}})$  hängt von zwei Zufallsprozessen ab, dem äußeren (den Systemausgaben) und dem inneren (der Systementwicklung). Dabei ist der decodierte Pfad von der Spur abhängig.

$$\begin{aligned} E(\text{Distanz}_{\text{Decoding}}) &= E\left(E(\text{Distanz}_{\text{Decoding}} | \text{Spur})\right) \\ &= E\left(\frac{P(\text{Distanz}_{\text{Decoding}}, \text{Spur})}{P(\text{Spur})} \times \text{Distanz}_{\text{Decoding}}\right) \\ &= E\left(P(\text{Distanz}_{\text{Decoding}} | \text{Spur}) \times \text{Distanz}_{\text{Decoding}}\right) \\ &= \sum_{s \in \text{Spur}} \sum_{p \in \text{Pfad}} P(s) \times P(p|s) \times \text{Distanz}(p, \text{dec}(s)) \end{aligned}$$

$\text{Distanz}(\dots)$  berechnet die Distanz zwischen dem Pfad  $p$  und dem decodierten Pfad  $\text{dec}(s)$  aus Spur  $s$ .  $P(s)$

ist die Wahrscheinlichkeit die Spur  $s$  zu beobachten und  $P(p|s)$  ist die Wahrscheinlichkeit des Pfades  $p$  unter der Bedingung, dass  $s$  beobachtet wurde. Die Pfadlänge  $|p|$  ist dabei konstant - die Anzahl der Zeitschritte  $t + 1$ . Damit ergibt sich für die Rekonstruktionsgüte:

$$O_3(t) = 1 - \frac{\sum_{s \in \text{Spur}} \sum_{p \in \text{Pfad}} P(s) \times P(p|s) \times \text{Distanz}(p, \text{dec}(s))}{N(t)}$$

Mit einem sinnvollen Normierungsfaktor ergeben sich folgende Grenzwerte:

- $O_3(t) = 1$   
Die Distanz zwischen den Pfaden ist 0. Der Pfad ist bezogen auf das Distanzmaß vollständig rekonstruierbar.
- $O_3(t) = 0$   
Die Distanz zwischen den Pfaden ist maximal. Der Pfad ist nicht rekonstruierbar. Dieser Wert ist jedoch nicht zu erreichen, da die Distanz des Pfades zu sich selber Null ist.

Die Anzahl der Pfade wächst exponentiell. Im Gegensatz zur vorherigen Definition wird aber jeder Pfad nur mit dem decodierten Pfad verglichen. Dies reduziert den Rechenaufwand deutlich. Dennoch ist auch dies in der Praxis nicht berechenbar.

## 2.1 Abschätzung

Die Distanz kann allerdings durch einen Maximum-Likelihood-Schätzer sehr gut abgeschätzt werden. Mithilfe einer Monte-Carlo Simulation werden verschiedene Pfade mit entsprechender Spur erzeugt. Aus der Spur wird mithilfe des Decodings ein entsprechender Pfad rekonstruiert und die Distanz zum richtigen Pfad bestimmt. Durch Anwenden des Satz von Bayes erhalten wir eine Formel, die durch die Monte-Carlo Simulation abgeschätzt werden kann:

$$\begin{aligned} & \sum_{s \in \text{Spur}} \sum_{p \in \text{Pfad}} P(s) \times P(p|s) \times \text{Distanz}(p, \text{dec}(s)) \\ &= \sum_{s \in \text{Spur}} \sum_{p \in \text{Pfad}} P(s) \times \frac{P(p \wedge s)}{P(s)} \times \text{Distanz}(p, \text{dec}(s)) \\ &= \sum_{s \in \text{Spur}} \sum_{p \in \text{Pfad}} P(p \wedge s) \times \text{Distanz}(p, \text{dec}(s)) \\ &= \sum_{s \in \text{Spur}} \sum_{p \in \text{Pfad}} P(p) \times P(s|p) \times \text{Distanz}(p, \text{dec}(s)) \end{aligned}$$

Das Sampeln des Pfades  $p$  mit der entsprechenden Wahrscheinlichkeit  $P(p)$  ist per Konstruktion möglich. Das Sampeln einer Spur  $s$  für einen Pfad ist ebenfalls tri-

vial möglich. Diese Abschätzung ist zum einen erwartungstreu, zum anderen kann durch die Streuung des Datensatzes ein Konfidenzintervall berechnet werden.

Im folgenden Experimente Kapitel zeigen wir, dass die Abschätzung berechenbar und die Ungenauigkeit sehr klein ist.

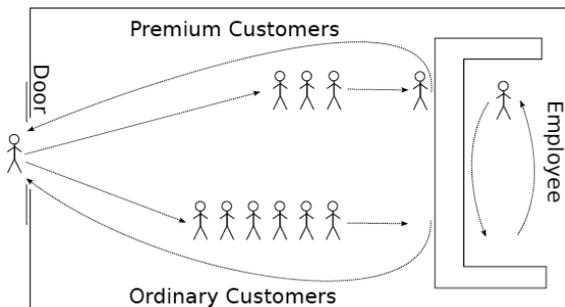
## 3 Experimente

Für die Experimente wurden zwei einfache Modelle mit unterschiedlicher Zielsetzung verwendet. Diese Modelle werden in verschiedenen Varianten verwendet, deren Systemzustand je nach Setup, unterschiedlich gut rekonstruierbar sein sollten.

Es wurden drei verschiedene Decoding-Verfahren, sowie die Hamming-Distanz und Damerau-Levenshtein-Distanz als Distanzmaß verwendet. Dabei wurde ein Pfad einmal als Sequenz von internen Zuständen und einmal als Sequenz von Ereignissen betrachtet. Zur Abschätzung der Rekonstruktionsgüte wurden alle Experimente mit wachsender Pfadlänge von bis zu 100.000 simuliert. Für Pfade mit einer Länge von bis zu 1.000 wurden die System je Pfadlänge 1.000-mal simuliert. Pfade mit einer Länge von über 1.000 wurden hingegen 100-mal simuliert. Jede Simulation entspricht einem zufälligen Ground-Truth Pfad mit entsprechender Beobachtungssequenz, aus der ein Pfad rekonstruiert werden kann. Um die Aussagekraft der Rekonstruktionsgüte Typ-3 zu testen, haben wir jedes Modell manipuliert und so die Rekonstruktionsgüte verändert.

### 3.1 Car-Rental-Service

Das erste 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.



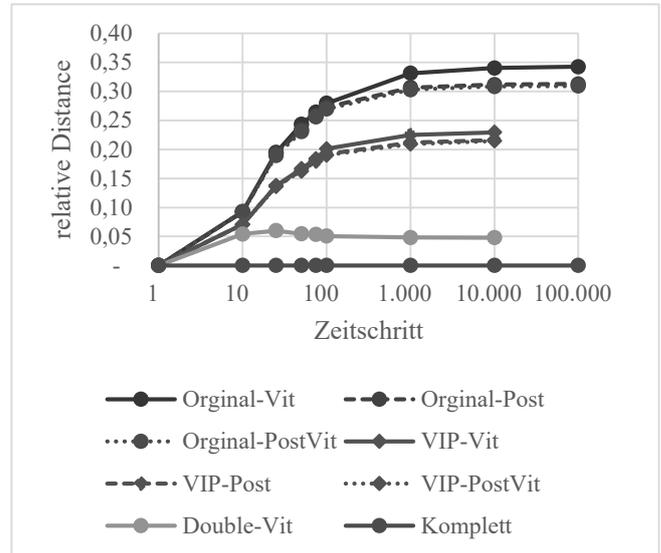
**Abb. 2:** Autovermietung mit zwei getrennten Warteschlangen für Premium und normale Kunden und einem Angestellten

Die mittlere Bearbeitungszeit pro Kunde (11.8 Zeiteinheiten) ist geringer als die mittlere Zwischenankunftszeit (15 Zeiteinheiten). Dies ist wichtig, da sich sonst, die Kunden aufstauen würden. Da das Ziel die Bestimmung der Warteschlangenlänge ist, wird hier der Pfad als Sequenz von internen Zuständen betrachtet. Der Diskretisierungszeitschritt im Lösungsverfahren beträgt 5 Zeiteinheiten. Im Schnitt tritt damit in 2/3 aller Zeitschritte ein Ereignis ein. Um die berechnete Rekonstruktionsgüte mit der „gefühlten“ Rekonstruktionsgüte zu vergleichen, wurden vier Varianten dieses Modells erzeugt.

1. Im Grundmodell existiert eine Tür mit einer Lichtschranke. Jedes Passieren der Tür erzeugt dasselbe Ausgabesymbol. (*Original*)
2. In der zweiten Version kann durch eine doppelte Lichtschranke die Richtung der Personen bestimmt werden. (*Double*)
3. In der dritten Version gibt es einen extra Eingang nur für Premiumkunden. 30% der Premiumkunden benutzen diesen. Ein Premiumkunde muss nicht durch dieselbe Tür ein und austreten. (*VIP*)
4. In der vierten Version wird zwischen Eingang und Ausgang und Normal- und Premiumkunde unterschieden. Jedes Ereignis ist exakt beobachtbar. (*Komplett*)

Die vier verschiedenen Modelle wurden verwendet, um die berechnete Rekonstruktionsgüte mit der gefühlten Rekonstruktionsgüte zu vergleichen. Für die gefühlte Rekonstruktionsgüte ergibt sich folgendes Ranking: Variante 1 hat die geringste Rekonstruktionsgüte, da zwar alle Ereignisse detektierbar sind, aber weder zwischen Kunden, noch zwischen Betreten und Verlassen unterschieden werden kann. Bei Version zwei (Doppelte Lichtschranke) ist das Betreten und Verlassen eindeutig zu unterscheiden. Premiumkunden betreten die Autovermietung deutlich seltener (1:2). Die doppelte Lichtschranke (Version 2) sollte also eine deutliche Verbesserung der

Rekonstruktionsgüte zur Folge haben. Version 3 hingegen sollte die Rekonstruktionsgüte nur leicht verbessern, da nur wenig Events besser zu beobachten sind. Variante vier ist zu 100% beobachtbar. Daher ist folgende Rangfolge zu erwarten: 4, 2, 3, 1.



**Diagramm 1:** Relative Hamming Distanz für Viterbi, Posterior und Posterior-Viterbi für alle 4 Varianten

Diagramm 1 zeigt den Verlauf der relativen Distanz über die Pfadlänge für die Hamming-Distanz des Posterior-Viterbi-Algorithmus. Da die Werte im Bereich von bis zu 0,3 liegen, ist hier, zur besseren Übersichtlichkeit, die relative Distanz und nicht deren Differenz zu 1 abgebildet. Daher ist ein größerer Wert in den Diagrammen schlechter. Außerdem ist die x-Achse in logarithmischer Skala eingeteilt. Gut zu erkennen ist, dass die Rekonstruktionsgüte konvergiert. Von 1.000 zu 10.000 Zeitschritten verändert sich der Wert der Rekonstruktionsgüte nur noch um 0.01. Von 10.000 zu 100.000 ist kein signifikanter Unterschied mehr zu sehen. Das 99%-Konfidenzintervall hat eine Breite von maximal  $\pm 0.01$ .

Posterior und Posterior-Viterbi Algorithmus liefern annähernd dieselben Ergebnisse. Der Viterbi-Algorithmus ist geringfügig schlechter. Deutlich ist zu erkennen, dass das Originalmodell die schlechtesten Werte liefert. Danach folgt das VIP-Modell (3) das leicht bessere Werte liefert. Das Modell mit der doppelten Lichtschranke (2) und das komplett-beobachtbare System (4) liefern deutlich bessere Ergebnisse. Dies entspricht der erwarteten Reihenfolge. Auch deutlich zu erkennen ist, dass bereits

nach wenigen Zeitschritten die Reihenfolge der verschiedenen Versionen bezüglich der Rekonstruktionsgüte sichtbar ist. Die Ergebnisse für die Damerau-Levenshtein-Distanz sind hier nicht abgebildet, da sie für dieses Model dieselben Werte liefert, wie die Hamming-Distanz. Dies liegt an der Art, wie die Pfade betrachtet werden. Ein Pfad entspricht dem internen Zustand des Systems. Werden zwei Events in der falschen Reihenfolge erkannt, so unterscheiden sich die Zustände nur nach dem ersten Event. Nach dem zweiten Event hätten beide Pfade denselben internen Zustand. Außerdem sind alle Ereignisse beobachtbar. Die zusätzlichen Operationen (Tauschen, Hinzufügen, Löschen) machen daher kaum einen Unterschied.

### 3.2 Quality-Tester

Das zweite System beschreibt einen Quality-Tester. Das System besteht aus zwei Quellprozessen (Source 0 und Source 1), die mit unterschiedlicher Häufigkeit Items produzieren. Die Items werden dann einem Qualitätstest unterzogen. Der Tester gibt dann für jedes Item den Zeitstempel und den Zustand des Items aus (OK oder NOK). Dabei ist nicht bekannt von welcher Quelle das Item stammt. Ziel ist es, zuzuordnen, welche Quelle welches Item produziert hat. In unserem Fall ist die Art des Ausgabesymbols nicht wichtig, da beide Quellen mit gleicher Wahrscheinlichkeit defekte Items produzieren. Abb. 3 zeigt den Quality-Tester.

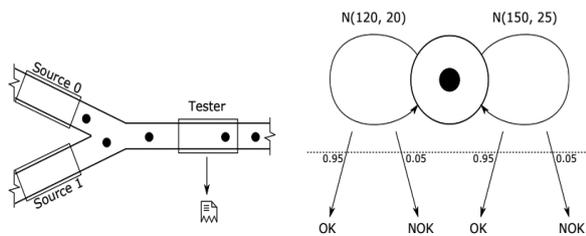


Abb. 3: Systemabbildung und Zustandsraum Model des Quality-Testers

Interessant ist dieses Modell, da es mit nur zwei Transitionen auskommt. Alle internen Zustände sind dieselben. Daher muss der Pfad hier als Sequenz von Ereignissen betrachtet werden. Aus diesem Grund werden hier auch nur die Daten auf Basis des Viterbi-Decodings vorgestellt. Posterior und Posterior-Viterbi sind nicht geeignet, um Event-Sequenzen zuverlässig zu bestimmen. Dies liegt daran, dass sie nur die Wahrscheinlichkeiten der einzelnen Zustände betrachten. Die Zustandsübergänge, die die Events darstellen, spielen bei Posterior keine

Rolle. Um die Rekonstruktionsgüte zu reduzieren, wurde das Model angepasst. Jedes Item wird nur mit einer 50% Wahrscheinlichkeit getestet (Half).

**Komplette Ereignissequenz.** Da im Grundmodell alle Ereignisse beobachtbar sind, werden alle Zeitschritte ohne Event mit einem Null-Event aufgefüllt. Ein Null-Event bedeutet, dass kein Ereignis eingetreten ist. Da Null-Events den Großteil des Pfades ausmachen, ist die relative Distanz auch entsprechend klein. Zudem zeigt sich auch in diesem Experiment, dass Hamming-Distanz und Damerau-Levenshtein-Distanz ähnliche Ergebnisse liefern.

Anders verhält es sich aber mit der Manipulierten Variante. Da nur 50% der Ereignisse beobachtbar sind, muss hier auch der Zeitschritt, in dem das Ereignis auftrat, korrekt rekonstruiert werden. Dadurch steigt die relative Distanz stark an. Hier zeigt sich auch ein Unterschied zwischen Hamming- und Damerau-Levenshtein-Distanz. Diagramm 2 zeigt die relativen Distanzen.

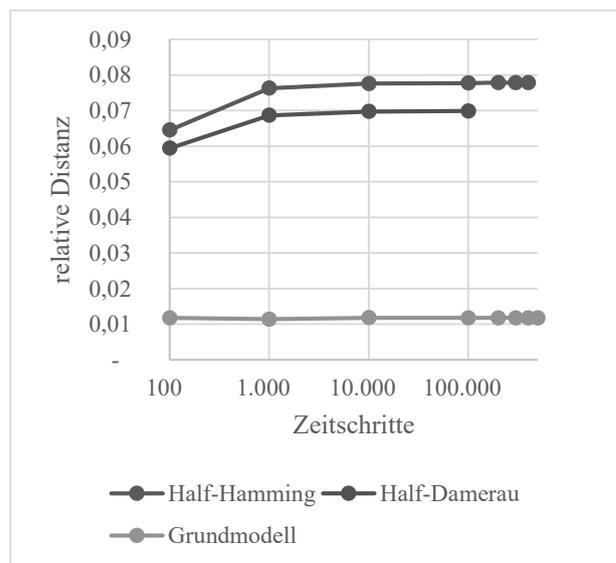
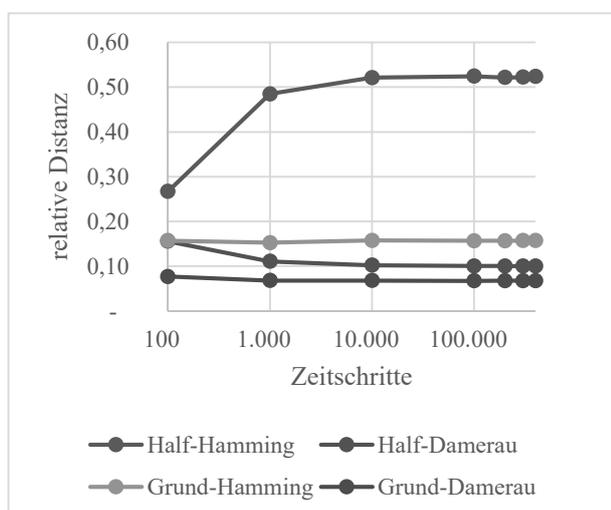


Diagramm 2: relative Distanz für das Grundsystem und dem System mit reduzierten Ausgaben mit dem Viterbi-Algorithmus.

Für das Grundmodell erzielt die Damerau-Levenshtein-Distanz nur im Falle, dass beide Quellen in aufeinanderfolgenden Zeitschritten ein Item produzieren und diese falsch zugeordnet wurden, bessere Ergebnisse. Dann ist die Hamming-Distanz für diesen Sequenzabschnitt zwei, während die Damerau-Levenshtein-Distanz eins ergibt. Diese Fälle sind im Grundmodell sehr selten. Im Modell mit reduzierter Rekonstruktionsgüte sieht das allerdings anders aus. Wird ein Item das keine Ausgabe produziert

einem benachbarten Zeitschritt zugeordnet, so ist auch hier die Damerau-Levenshtein Distanz kleiner. Die Ergebnisse zeigen, dass die Rekonstruktionsgüte schnell konvergiert und dass auch rechnerisch das Modell mit reduzierter Rekonstruktionsgüte schlechtere Ergebnisse liefert. Allerdings zeigt sich hier, dass die Werte selber von geringer Aussagekraft sind, da hauptsächlich Null-Events auf Null-Events zugeordnet wurden, was bei den verwendeten Modellen keine große Schwierigkeit darstellt, da fast alle Events Null-Events sind.

**Eventreihenfolge.** Um das Problem der Null-Events zu lösen, wurde im zweiten Experiment mit den beiden Modellen, der Pfad nur auf die Events ungleich dem Null-Event eingekürzt. Dadurch werden die Pfade deutlich kürzer.



**Diagramm 3:** relative Distanz des Quality-Testers der verschiedenen Systeme mit Damerau- und Hamming-Distanz des Experiments „Eventreihenfolge“

Auch hier zeigt sich, dass die Rekonstruktionsgüte schnell konvergiert. Zusätzlich zeigt sich hier, dass die Damerau-Levenshtein-Distanz deutlich besser performt als die Hamming-Distanz. Insbesondere ist die relative Distanz des Modells mit reduzierter Rekonstruktionsgüte  $>50\%$ . Dies liegt daran, dass die Hamming-Distanz eigentlich für dieses Problem nicht verwendet werden kann. Da nicht jedes Ereignis beobachtet wird, ist auch die Anzahl der produzierten Items nicht bekannt. Es entstehen Pfade mit unterschiedlicher Länge. Damit die Hamming-Distanz damit umgehen kann, wurde der längere Pfad abgeschnitten und die abgeschnittene Länge der Distanz zugerechnet.

Bei dem Grundmodell kann ein derartiges Problem

nicht auftreten, da jedes Ereignis beobachtet wird. Wird allerdings ein Ereignis falsch erkannt, so bedeutet dies in fast allen Fällen, dass das darauffolgende Ereignis mit hoher Wahrscheinlichkeit auch falsch erkannt wurde.

## 4 Ergebnisdiskussion

Für die verwendeten Modelle war es möglich, die Rekonstruktionsgüte selbst für viele Zeitschritte zu berechnen. Es zeigt sich, dass insbesondere Posterior und Posterior-Viterbi sehr gut bei der Rekonstruktion der internen Zustände funktionieren. Allerdings sind diese Algorithmen bei der Rekonstruktion des Event-Pfads nicht verwendbar. Dies ist zu erwarten, da insbesondere Posterior keine Pfade, sondern nur Zustandssequenzen zurückgibt.

Auch zeigt sich, dass die Hamming-Distanz nicht sinnvoll angewendet werden kann, wenn die Ereignisse nicht zu 100% zu beobachten sind.

Zu beobachten ist auch, dass bei allen Experimenten die Rekonstruktionsgüte Typ-3 konvergiert. Mit größerem zeitlichem Abstand wird der Einfluss verschiedener Systemzustände immer geringer und nähert sich 0 an.

Der Unterschied in der Rekonstruktionsgüte Typ-3 war bereits nach wenigen Zeitschritten zu erkennen. Dies ist eine wichtige Entdeckung. Für die hier verwendeten Modelle ist ein Berechnen der Rekonstruktionsgüte Typ-3 für lange Sequenzen nicht notwendig, wenn nur die Reihenfolge der Güte der VSS interessant ist. So lässt sich das Sensorsetup optimieren und das beste Sensorsetup bestimmen.

Einschränkend ist allerdings zu sagen, dass bei der Rekonstruktion des Pfades für die sehr langen Spuren einzelne Pfade nicht rekonstruiert werden konnten. Dies liegt an der verwendeten Methode zur Rekonstruktion. Auch wenn diese Fälle sehr selten auftraten, so sind dies grade die unwahrscheinlichen Pfade, bei denen eine hohe Distanz zum dekodierten Pfad zu erwarten ist. Dadurch wird die Distanz geringfügig unterschätzt.

Ein hier nicht untersuchter Einflussfaktor ist die Länge eines Zeitschrittes. Je kürzer die Zeitschritte, desto genauer, aber auch rechenaufwändiger sind die Ergebnisse. Der in diesem Paper verwendete Algorithmus setzt voraus, dass nur ein Ereignis im selben Zeitschritt auftritt. Insbesondere bei der Autovermietung treten alle Ereignisse mit einer Dichte von  $1/2$  bis zu  $2/3$  auf. Dies betrifft allerdings auch die letztendliche Verwendung des VSS. Auch dort besteht dasselbe Problem, sodass die Aussagen zur Rekonstruktionsgüte dennoch korrekt sind.

Wird ein anderer Zeitschritt zur Lösung eines VSS verwendet, dann muss dies allerdings auch bei der Berechnung der Rekonstruktionsgüte des VSS berücksichtigt werden.

## 5 Zusammenfassung und Ausblick

Die Rekonstruktionsgüte ist ein wichtiges Werkzeug, um den potentiellen Nutzen eines Virtuellen Stochastischen Sensors zu bestimmen. Wir haben in diesem Paper eine Definition vorgestellt, die Bezug zur Verwendung hat. Unsere Experimente haben gezeigt, dass die Rekonstruktionsgüte Typ-3 nicht nur berechenbar ist, sondern bereits wenige Zeitschritte ausreichen, um die Modelle zu vergleichen. Dies ist notwendig, um das Sensorsetup zu optimieren, oder zu bestimmen ob ein Virtueller Stochastischer Sensor für einen bestimmten Anwendungsfall einsetzbar ist.

Aus dieser Arbeit ergeben sich allerdings auch einige weitere Fragen. So ist bisher noch nicht geklärt was eine Rekonstruktionsgüte mit einem spezifischen Wert genau für ein Problem bedeutet. Dies ist problemabhängig. Insbesondere die Distanzfunktion hat starken Einfluss auf die Rekonstruktionsgüte. Wie sich die Rekonstruktionsgüte mit anderen Distanzmaßen und größeren Modellen verhält, muss in zukünftigen Arbeiten noch untersucht werden. So könnte z.B. für die Autovermietung die mittlere Quadratische Abweichung von Interesse sein. Ob sich für ein derartiges Distanzmaß ein sinnvoller Normierungsfaktor findet, oder ob das Distanzmaß selber aussagekräftiger ist, ist auch eine Frage die noch geklärt werden muss.

## 6 Verweise

- [1] S. Kabadayi, A. Pridgen und C. Julien, „Virtual sensors: abstracting data from physical sensors,“ *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM'06)*, pp. 586-592, 2006.
- [2] L. Fortuna, S. Graziani, A. Rizzo und M. Xibilia, *Soft Sensors for Monitoring and Control of Industrial Processes*, 2007.
- [3] C. Krull, *Virtual Stochastic Sensors: Formal Background and Example Applications*, Magdeburg: Shaker Verlag Düren, 2021.
- [4] L. Rabiner und B. Juang, „An introduction to hidden Markov models,“ *IEEE ASSP Magazine*, Bd. Volume 3, Nr. 1, pp. 4-16, Jan 1986.
- [5] C. Krull, R. Buchholz und G. Horton, „Virtual Stochastic Sensors: How to gain Insight into Partially Observable Discrete Stochastic Systems,“ in *The 30th IASTED International Conference on Modelling, Identification and Control*, 2011.
- [6] G. Horton, „A NEW PARADIGM FOR THE NUMERICAL SIMULATION OF STOCHASTIC PETRI NETS WITH GENERAL FIRING TIMES,“ in *European Simulation Symposium*, Dresden, 2002.
- [7] S. Lazarova-Molnar und G. Horton, *Proxel-Based Simulation of Stochastic Petri Nets*, SCS European Publishing House, 2004.
- [8] S. Lazarova-Molnar und G. Horton, *Proxel-Based Simulation of Stochastic Petri Nets Containing Immediate Transitions*.
- [9] F. Wickborn, C. Isensee, T. Simon, S. Lazarova-Molnar und G. Horton, *A New Approach for Computing Conditional Probabilities of General Stochastic Processes*, 2006.
- [10] Kálmán, *LECTURES ON CONTROLLABILITY AND OBSERVABILITY*, Stanford University, 1970.
- [11] J. Lunze, *Regelungstechnik 2: Mehrgrößensysteme, Digitale Regelung*, Springer-Verlag Berlin Heidelberg GmbH, 2002.
- [12] F. M. Ham und R. G. Brown, „Observability, Eigenvalues, and Kalman Filtering,“ in *IEEE Transactions on Aerospace and Electronic Systems* 2, 1983.
- [13] P. Krenckel, *Beobachtbarkeit von Virtuellen Stochastischen Sensoren*, Magdeburg: Otto-von-Guericke-Universität Magdeburg, 2019.
- [14] P. Krenckel, C. Krull und G. Horton, *Observability of Virtual Stochastic Sensors: Observability Type-1*, Palma de Mallorca, 2019, pp. 8-12.
- [15] R. Buchholz, *Conversive Hidden non-Markovian models*, Magdeburg, 2012.
- [16] F. J. Damerau, „A technique for computer detection and correction of spelling errors,“ in *Communications of the ACM*, Bd. 7, New York, NY, USA, Association for Computing Machinery, 1964, pp. 171-176.
- [17] V. I. Levenshtein, „Binary Codes Capable of Correcting Deletions, Insertions and Reversals,“ *Soviet Physics Doklady*, Bd. 10, p. 707, February 1966.

# Integrating Reinforcement Learning and Discrete Event Simulation Using the Concept of Experimental Frame: A Case Study With MATLAB/SimEvents

Thorsten Pawletta\*, Jan Bartelt

Research Group Computational Engineering and Automation (CEA), Wismar University of Applied Sciences, Philipp-Müller-Straße 14, D-23966 Wismar, Germany; [\\*thorsten.pawletta@hs-wismar.de](mailto:*thorsten.pawletta@hs-wismar.de), [jan.bartelt@hs-wismar.de](mailto:jan.bartelt@hs-wismar.de)

**Abstract.** Reinforcement Learning (RL) is an optimization method characterized by two interacting entities, the agent and the environment. The environment is a Markov Decision Process (MDP). The goal of RL is to learn how an agent should act to achieve a maximum cumulative reward in the long-term. In discrete-event simulation, the dynamic behavior of a system is represented in a model (DESM) that is executed via a simulator. The concept of Experimental Frame (EF) provides a structural approach to separating the DESM into the Model Under Study (MUS) and its experimental context. Here, we explore the integration of a discrete event MUS as an environment for RL using the concept of EF. After discussing the methodological framework, a case study using MATLAB/Simulink and the SimEvents blockset is considered. The case study starts with an introduction of the discrete-event MUS for which a control strategy shall be developed. The MUS is reused in three experiments using specific EFs. First, an EF for the design of a heuristic control strategy with ordinary simulation runs is presented. Then, based on the methodological approach, specifics of the EF are considered when using a self-implemented Q-agent and the RL toolbox of MATLAB/Simulink.

## Introduction

In modeling and simulation (M&S), a model describes the dynamic behaviour of a real or virtual system. The execution of the model is performed using a simulator. In the versatile use of a model, it should be developed independently from the context of use. The reference to a concrete experiment can be mapped by an Experimental Frame (EF). An EF specifies the conditions under which a system is observed or a model experimented with (Zeigler [12], Zeigler et al. [14], Traore and Muzy [11]). The

model used is called the Model under Study (MUS). Depending on the EF, the same MUS can be used in different experimental contexts, such as a parameter study, sensitivity analysis, optimization, etc. The EF and MUS form the simulation model (SM). Discrete event simulation models (DESM) are characterized by a finite number of states over a continuous time base.

The EF implements the interface for a Simulation-Based Experiment (SBE). Inspired by Breitenecker's [1] approach to structuring SBEs, Pawletta et al. [5] and Schmidt [7] introduced the concept of Simulation Method (SimMeth) and Experiment Method (ExpMeth). The SimMeth controls the execution of the simulation runs via a simulator and ExpMeths are arbitrary numerical methods. ExpMeths are used for pre- and post-processing or to control the SimMeth, such as in simulation-based optimization experiments (Carson and Maria [2]; Schmidt [7]).

Reinforcement Learning (RL) (Sutton and Barto [8]) in combination with a dynamic system simulation can be considered as a SBE. However, RL is an optimization method for Markov Decision Processes (MDPs). The MDP is modeled as an environment and an agent acts as a controller. The goal is to learn how the agent should act to achieve a maximum cumulative reward in the long-term. In contrast to a DESM, an MDP is a discrete time process and the time base is only used for the sequential ordering of states. Not all states of the MUS are usually of interest to the RL. Accordingly, the states of the MUS must be converted into MDP-compliant states. Due to the methodological differences, the combination of the two methods, RL and discrete event simulation, often lead in practice to implementations that are difficult to maintain and MUS that are not generally usable.

Here, the practical integration of both methods using the concept of EF is explored by means of a case study and using MATLAB/Simulink as well as the SimEvents blockset (MathWorks [9]). We start with some basics to SBEs, EF, RL and the usage of RL in a SBE. Then, the MUS is introduced for which a control strategy is developed. To present the reusability of the MUS in the context of different experiments using specific EFs, we start with the design of a heuristic controller. This is followed by two experiments on RL-based controller design.

This work is based on Pawletta and Bartelt [6]. More details on the theoretical background and related work is provided there.

## 1 Basics

Based on Pawletta and Bartelt [6], we briefly review the basics of structuring SBEs, the RL method, and the use of RL as a method of an SBE.

### 1.1 Structuring Simulation-Based Experiments

Schmidt [7] divides SBEs into three classes. We consider only the first two classes. The execution of one or more simulation runs by a SimMeth constitutes a simple SBE, if the SimMeth is invoked directly by the user or a supervisory Experiment Control (EC). An EC defines the goals and steps of an experiment and automates the experiment execution.

In a complex SBE, the SimMeth is controlled by an ExpMeth, for example, by a numerical optimization method. Figure 1 shows the basic structure of a complex SBE. Both the SimMeth and ExpMeth define process parameters ( $P_{\text{ExM}}$ ,  $P_{\text{SimM}}$ ).

The EF separates the MUS from a specific context of use to improve the reusability of the MUS. Formally, Zeigler [13] defines the function of an EF with the tuple.

$$EF = \langle T, I, C, O, \Omega_I, \Omega_C, SU \rangle \quad (1)$$

T represents the time base, I and O the set of input and output variables of the MUS (equivalent to  $I_{\text{MUS}}$  and  $O_{\text{MUS}}$  in Fig. 1), C the set of run control variables,  $\Omega_I$  the set of admissible input segments,  $\Omega_C$  the set of admissible control segments, and SU the set of summary mappings. Set  $\Omega_I$  refers to the input variables of the MUS and to the input/output relationships in the EF. Set  $\Omega_C$  defines the experimental constraints. The experiment objectives are mapped to *interest variables*. Set SU defines the determination of the interest variables based on the MUS outputs.

The interest variables are typical output variables of the EF. The implementation of an EF is done using three types of components, as illustrated in Figure 2 (Zeigler [13]; Zeigler et al. [14]). The *generator* (Gen) initializes the configurable parameters of the MUS and calculates the input segments for the MUS which can also be inputs of the *Acceptor* (Acc) or *Transducer* (Trans). The Acc defines the admissible control segments and monitors their compliance. The output of the Acc is run control information. The Trans calculates the SU.

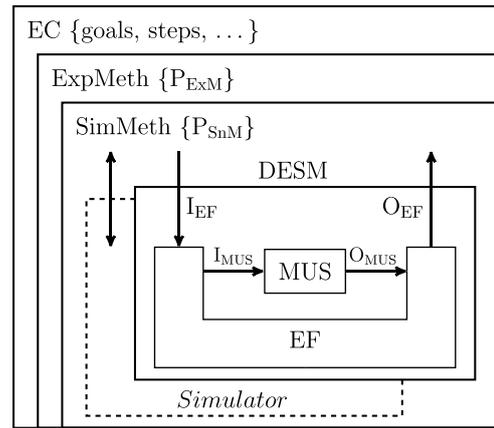


Figure 1. Basic structure of a complex SBE.

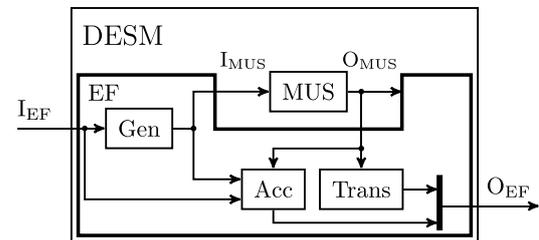
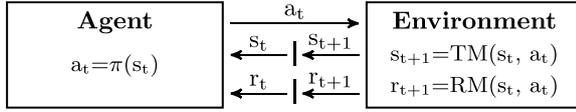


Figure 2. Basic structure of a DESM with MUS and EF. An EF does not necessarily have to contain all three components and the coupling relationships are not fixed.

### 1.2 The Method of Reinforcement Learning

According to Sutton and Barto [8], RL focuses on the sequential decision-making by an agent that interacts with a real or virtual environment. The agent is trained by its interactions with the environment. The goal of RL is to learn a behavioral strategy  $\pi: S \rightarrow A$  for the agent that assigns an action  $a \in A$  to each state  $s \in S$  of the environment. Thus, the agent can act as a controller for the environment. Using RL, a distinction is made between the training and deployment of an agent, although the agent can continue learning during deployment. The basic RL

framework is shown in Figure 3.



**Figure 3.** Basic RL framework.

In model-free RL, the agent only knows the allowed action set  $A$  at the start of training. The states  $s \in S$  of the environment are unknown to the agent. When an action  $a_t \in A$  takes effect, the environment determines its next state  $s_{t+1}$  as well as a reward value  $r_{t+1}$  using a state transition model  $TM: S \times A \rightarrow S$  and reward model  $RM: S \times A \rightarrow R$ . The next state and the reward value are sent back to the agent. The index  $t$  marks a sequence of states in the sense of a MDP. Through iterative interactions with the environment, the agent obtains information about possible states of the environment and the benefits of actions, gradually improving its behavioral strategy  $\pi$ . A variety of different learning strategies have been developed for RL agents such as Q-learning, Deep Q Networks etc.

We briefly consider Q-learning that uses formula (2) to learn a strategy  $\pi$  using a table function called the Q-matrix. A matrix element  $Q(s, a)$  represents the estimated benefit of an action  $a_t$  when it is performed in the state  $s_t$  of the environment. The updated  $Q(s_t, a_t)$  value of the current state/action tuple  $(s_t, a_t)$  is calculated from the previous  $Q(s_t, a_t)$  value, the currently received reward  $r_{t+1}$ , and the maximum Q-value ( $\max_a Q(s_{t+1}, a)$ ) of all possible actions in the currently received next state  $s_{t+1}$ . The variables  $\alpha$  and  $\gamma$  are hyperparameters, i.e. they must be defined before training, but can still be adjusted during training.

$$Q(s_t a_t) \leftarrow Q(s_t a_t) + \alpha [r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (2)$$

The training takes place in independent episodes. Each episode starts in an initial state  $s_0$  of the environment and ends when a target state  $s_{target}$  or abort state  $s_{abort}$  is reached. At the beginning of the training, the agent selects an action  $a \in A$  randomly. This is called *exploration*. As the learning process progresses, the agent increasingly uses the knowledge it has acquired to select an action which is called *exploitation*. The ratio  $\epsilon$  of exploration to exploitation is adjusted over the course of the training. After the completion of a defined number of episodes, the behavioral strategy  $a = \pi(s)$  is derived from

the training data.

### 1.3 Integrating Reinforcement Learning into a Simulation-Based Experiment

When integrating RL and dynamic system simulation, the MUS forms the environment for the RL agent. The goal of such an SBE is

- to learn the best possible behavioral strategy of the agent,
- to extract this strategy from the training data, and
- to use it as a controller for the MUS.

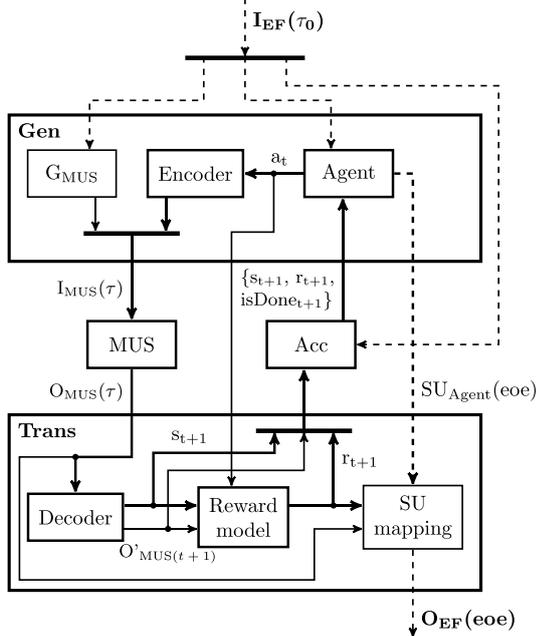
The first two items are defined with an ExpMeth *training* that controls a SimMeth to execute simulation runs. The ExpMeth training contains the following basic steps:

- Set the RL-specific experiment parameters PExM such as the learning rate, exploration rate, maximum number of episodes, Q-matrix etc.
- Set the simulation execution parameters PSnM for the SimMeth, such as the simulator to be used, the simulation time interval etc.
- Set the DESM parameters for the EF and the MUS and prepare the DESM for executing using a SimMeth.
- Initialize statistical variables, such as those used to record the total reward per episode etc,
- Compute the defined number of episodes, i.e. call the SimMeth into a loop to execute the DESM, update the statistical variables after each episode, and check whether to abort the training or continue with another episode.
- Determine and save the best policy  $\pi$ , and plot essential learning results.

Figure 4 shows the basic structure of a DESM with an EF for RL in the training phase. The variables  $\tau$  and  $t$  represent the different time bases.  $\tau$  is the continuous time of the MUS and  $t$  the discrete time for ordering the sequential states of the RL. The input variables  $I_{EF}$  are initialized at the simulation start time  $\tau_0$ , at the beginning of an episode. Results are get back via  $O_{EF}$  at the *end of an episode* (*eof*).

The *Gen* is subdivided into three subcomponents. *Gen.GMUS* initializes the parameters of the MUS at  $\tau_0$  and calculates input segments  $\Omega_I$  for the MUS inputs  $I_{MUS}(\tau)$  over the course of an episode. *Gen.Agent* maps the RL agent. It is part of the *Gen* because its output, an action  $a_t$ , produces inputs of the MUS. The agent's parameters

are initialized via the input interface  $I_{EF}(\tau_0)$ . In addition to the ordinary agent inputs' new state  $s_{t+1}$  and the reward value  $r_{t+1}$ , the third input  $isDone_{t+1}$  is a Boolean value that signals the end or cancellation of an episode. At the *eof*, the agent creates a summary mapping  $SU_{Agent}$  that contains RL-specific values such as the *number of steps*, the *total reward*, or the strategy learned so far (e.g. the *Q-matrix*). The  $SU_{Agent}$  is passed to the *Trans. Gen.Encoder* defines a mapping  $i(\tau)=h(a_i)$  to transform a single action value  $a_i$  into MUS compatible input values  $i(\tau) \in I_{MUS}(\tau)$  as introduced by Choo et al. [3].



**Figure 4.** Basic structure of a DESM with a MUS and EF for RL in the training phase.

The *Trans* is also subdivided into three subcomponents. *Trans.Decoder* defines (i) the calculation of the *interest values*  $O'_{MUS(t+1)}$  from the current outputs  $O_{MUS}(\tau)$  of the MUS related to the time base of the RL (i.e.  $O'_{MUS(t+1)}=f(O_{MUS}(\tau))$ ), and transformation of the interest values  $O'_{MUS(t+1)}$  to a state  $s_{t+1}$  in the RL space (i.e.  $s_{t+1}=g(O'_{MUS(t+1)})$ ). Thus, all interest values of the MUS are mapped into one state for the RL and for each particular interest value there is only one corresponding state in the RL space (Choo et al. [3]). The *Trans.Reward-model* maps the reward calculation. The reward value characterizes a state transition  $s_t \rightarrow s_{t+1}$  in the RL space. Defining the reward calculation is sometimes a difficult problem. Our own experiments showed that the reward value can sometimes be computed very efficiently based on the  $O'_{MUS(t+1)}$  values. The component *Trans.SU-mapping* implements the overall SU of an episode and passes

it at the *eof* to the output  $O_{EF}$ .

The *Acc* checks compliance with the constraints for the episode using defined run control information. Run control variables can be initialized via  $I_{EF}(\tau_0)$ . Typical run conditions to be monitored include the simulation interval  $[\tau_0, \tau_{final}]$  of the MUS and thus the maximum duration of an episode, the detection of illegal states or the reaching of a target state. The *Acc* checks all the relevant quantities and sets the *isDone* value, before sending the tuple  $(s_{t+1}, r_{t+1}, isDone_{t+1})$  to the *Gen.Agent*.

When deploying a learned strategy, we have to distinguish whether it is used with or without further learning of the agent. For an experiment *deployment without training* the EF simplifies as shown in Pawletta and Bartelt [6]. No explicit ExpMeth is required. The SimMeth is called directly in the EC according to the number of simulation runs to be executed.

## 2 Case Study

The basic implementation of the approach to integrate RL and discrete-event simulation introduced in Section 1.3 will be demonstrated by a case study using MATLAB/Simulink and the SimEvents blockset. The objective is to develop a control strategy for a MUS with discrete-event dynamics. First, the most general possible modeling of the MUS, i.e. without concrete references to an experiment, is discussed. Then, the same MUS is used in three experiments using different EFs: (i) to design a heuristic strategy, (ii) to learn a strategy with a self-implemented Q-agent, and (iii) to learn a strategy using MathWorks' dedicated RL toolbox (MathWorks [10]).

### 2.1 Model Under Study and General Objectives of the Control Design

The MUS is a simple server line consisting of an entity generator, a convertible operating unit, and two downstream servers connected in parallel with separate input queues as shown in Figure 5. The operating unit can process two types of entity ( $jobType=1 \mid 2$ ). A separate processing time can be defined for each entity type ( $procT1, procT2$ ). A retooling time ( $retoolingT$ ) is necessary when the entity type is changed in the operation unit. The calculation of the type of entity and retooling time dependent processing time is done during simulation runtime using two Simulink functions (not shown in Fig. 5). After processing, branching into one of the two FiFo queues of the downstream servers takes place depending on the en-

tity type. The downstream servers have different processing times ( $saleT1$ ,  $saleT2$ ). The definition of the different time values is determined by a value vector  $param=[procT1, procT2, retoolingT, saleT1, saleT2]$  at input port3 at the simulation start time  $\tau_0$ . Entities are generated via input events ( $msgGenJob$ ) at input port1. The entity type ( $jobType$ ) to be generated follows on from the value at input port2. After an entity has been processed in the operating unit, the MUS generates an output event ( $y\_msgFinish$ ) at output port1. Furthermore, the current tool setting ( $sSetting$ ) of the operating unit, the current queue lengths ( $y\_#jobsQ1$ ,  $y\_#jobsQ2$ ), and the number of completed entities on the downstream servers ( $y\_#jobs1sold$ ,  $y\_#jobs2sold$ ) are output as data from port2 to port6. Hence, input set  $I_{MUS}$  and output set  $O_{MUS}$  are defined by:

- $I_{MUS} = \{msgGenJob(\tau), type(\tau), param(\tau)\}$
- $O_{MUS} = \{y\_msgFinish(\tau), y\_sSetting(\tau), y\_#jobsQ1(\tau), y\_#jobsQ2(\tau), y\_#jobs1sold(\tau), y\_#jobs2sold(\tau)\}$

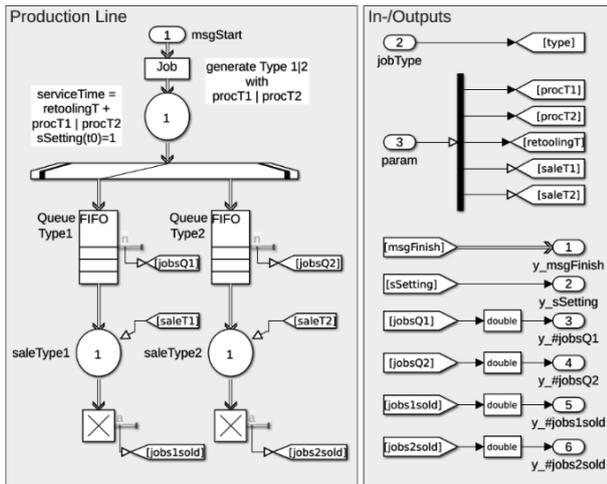


Figure 5. Structure of the MUS in SimEvents

The MUS represents the dynamic system behavior independent of a concrete experiment. The goal of the following experiments is to design a controller with the best possible injection strategy of the two entity types into the MUS so then the queues have the most balanced stock of both types available for the downstream servers.

## 2.2 Designing a Heuristic Strategy

The top-level structure of the DESM for designing a heuristic control strategy is shown in Figure 6. The MUS

named *Propline* implements the input and output interface described in Section 2.1 with  $I_{MUS}(\tau)$  and  $O_{MUS}(\tau)$ . The  $I_{EF}/O_{EF}$  of the EF are not visible on the top-level structure of the DESM. This interface is realized via workspace variables. The EF consists of five components, of which *Parameters*, *Controller* and *Encoder* form the generator *Gen* according to Figure 2. With the exception of the *Encoder*, the components of the EF operate purely signal-oriented.

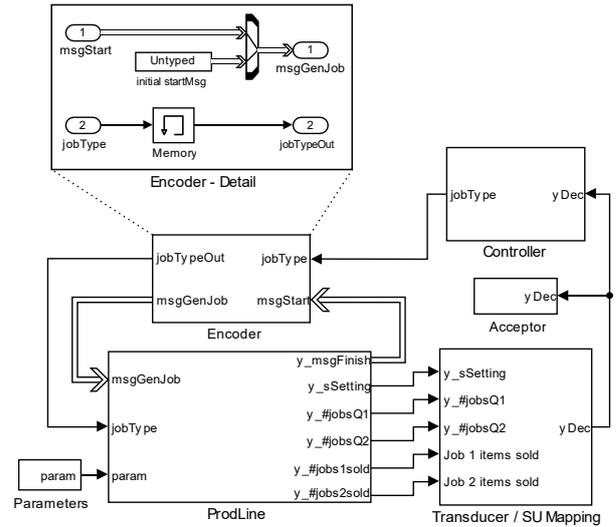


Figure 6. Top-level structure of the DESM with MUS and EF for designing a heuristic control strategy, and substructure of the Encoder block

The *Transducer* monitors the signal-oriented outputs of the MUS, maps the variables of interest  $O'_{MUS}$  for this experiment in a vector  $yDec=[sSetting, \#jobsQ1, \#jobsQ2]$ , and provides the vector as output variable. Moreover, the *Transducer* generates the SU mapping, by providing the time trajectories of the  $O'_{MUS}$  quantities as EF outputs  $O_{EF}$  via the data workspace.

The *Acceptor* controls the termination of the simulation after a specified time interval  $[\tau_0, \tau_{final}]$  has elapsed. It evaluates the interest variables  $\#jobsQ1$  and  $\#jobsQ2$  and terminates the simulation run abnormally if the difference between the two quantities exceeds a maximum value.

The *Controller* implements the heuristic strategy. It determines the next entity type to be generated based on the current values of the interest variables received by the *Transducer*. The goal is to minimize the difference between the two queue contents ( $\#jobsQ1$  and  $\#jobsQ2$ ) while respecting the current tool setting ( $sSetting$ ). The

result is passed on as a signal (*jobType*) to the *Encoder*.

The *Encoder* works event-driven (see Fig. 6). If the operation server of the MUS is free, it sends an event *y\_msgFinish* to the *Encoder*. Thereupon the *Encoder* sends an event *msgGenJob* to the MUS and forwards the signal *jobType* that codes the entity type to be generated. At the start of a simulation run, the *Encoder* generates the initial input event *msgGenJob* and sets the entity type (*jobType=1*) to be generated for the MUS. *Parameters* generates the constant input segments for the MUS vector *param* for initializing the MUS parameters.

This is a simple SBE. The EC defines the parameters  $P_{Sim}$  to be varied and directly calls the SimMeth to execute simulation runs.

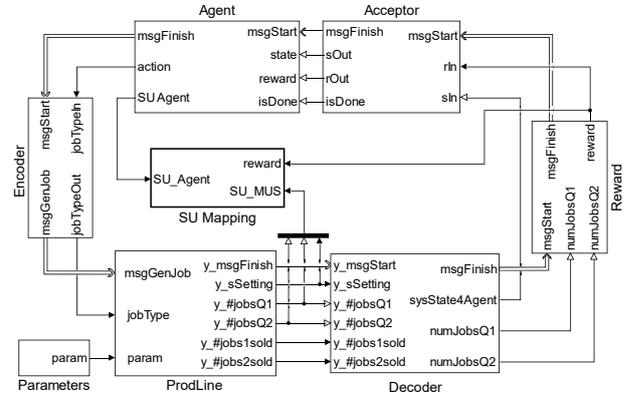
### 2.3 Learning a Strategy Using a Self-Implemented Q-Agent

The top-level structure of the DESM for this SBE is shown in Figure 7. The identical MUS named *Prodline* is integrated into an RL-specific EF according to Figure 4. As in the previous experiment, the EF interface ( $I_{EF}/O_{EF}$ ) is implemented via Workspace variables. *Parameters*, *Agent* and *Encoder* form the generator *Gen*, and *Decoder*, *Reward* as well as *SU Mapping* form the transducer *Trans* (cf. Fig. 2). To learn a strategy, the agent requires unique state-action tuples ( $s_t, a_t$ ) as well as associated next state  $s_{t+1}$  and reward values  $r_{t+1}$ . Hence, the two time bases  $t$  and  $\tau$  were introduced in Subsection 1.3 for the EF and the MUS. Accordingly, the components of the EF are implemented event-oriented, with the exception of *Parameters* and *SU Mapping*. The component *Parameters* is identical to the previous experiment.

At simulation start time  $\tau_0$ , an episode is started by the *Agent* sending an event *msgGenJob* and setting an action value  $a_t = \{1|2\}$  at the output port *action*. In this case, the outputs of the *Agent* are compatible with the inputs of the MUS in value and timestamp with respect to the global simulation clock. Hence, the *Agent*'s outputs are only forwarded by the *Encoder* to the MUS *ProdLine* that generates a new entity with *jobType=action value*.

When an entity has completed on the operation unit, an output event  $y\_msgFinish(\tau)$  is sent from the MUS to activate the *Decoder* and study-specific output data( $\tau$ ) is passed signal-oriented to the *SU Mapping* for trajectory recording. The *Decoder* selects the information relevant to the RL from the *MUS outputs*( $\tau$ ) and calculates the new state  $s_{t+1}$  of the RL space. To limit the RL space, the *Decoder* truncates the two queue contents ( $\#jobsQ1$  and  $\#jobsQ2$ ) to a maximum length. The new state  $s_{t+1}$  is thus

calculated from the two limited queue contents and the current tool setting (*sSetting*) of the operating unit, and output at the port *sysState4Agent*.



**Figure 7.** Top-level structure of the DESM with MUS and EF for learning a strategy using a self-implemented agent component.

After decoding, the reward calculation is activated by an event *msgFinish*. Contrary to the general approach, the reward is not calculated using the RL-related state  $s \in S$  but on the basis of MUS-related interest variables  $O'_{MUS(t+1)}$ , in this case  $\#jobsQ1$  and  $\#jobsQ2$ . In terms of content, both approaches are identical but the second one resulted in a much better structured reward computation.

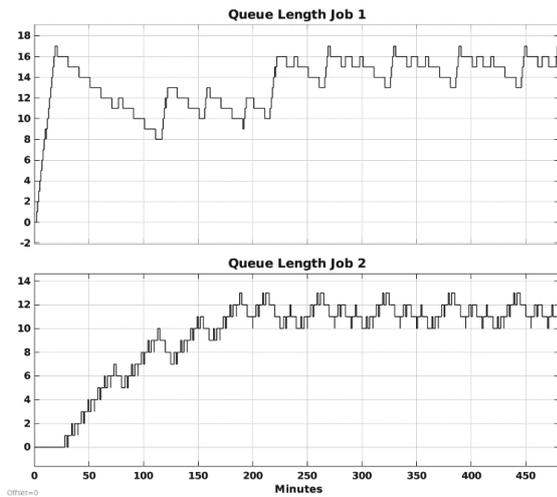
After the reward calculation, the *Acceptor* is activated by an event *msgFinish*. In this experiment, no constraints are defined for  $s_{t+1}$  and  $r_{t+1}$ , so they are only passed to the *Agent*. Only a control segment is defined for the simulation time interval  $[\tau_0, \tau_{final}]$ , which specifies the length of an episode. At the termination of an episode, the *Acceptor* schedules an internal event with an infinitesimal time advance. The time delay is necessary for data updates in the *Agent* and *SU Mapping* at the end of an episode. The *Acceptor* activates the *Agent* via an event *msgFinish* and signals using the boolean variable *isDone* whether the end of an episode (*eoe*) has been reached or not.

The *Agent* evaluates the boolean *isDone* value. If *isDone* is false, it executes its learning rules, calculates a new *action* value, and generates an output message *msgFinish* to activate the *Encoder*. In case of *isDone* is true it performs a data update *SU\_Agent(eoe)* and the episode is terminated by the *Acceptor*.

This is a *complex SBE*. The EC defines the parameters  $P_{ExM}$  and  $P_{Sim}$  (cf. Fig. 1), and calls an ExpMeth *training* as described in Subsection 1.3, which calls the SimMeth in a loop to execute the DESM for one episode at a time.

Figure 8 shows the simulation results of the MUS using a learned control strategy after 5000 episodes.

The event-oriented implementation of the EF components was done using SimEvents' *Discrete Event Charts*, which call MATLAB functions. This makes the algorithms of the components, such as the learning approach of the agent, easily interchangeable. In Pawletta and Bartelt [6], the algorithms of this experiment are presented in more detail and the full implementation is available on Github (FG CEA [4]).

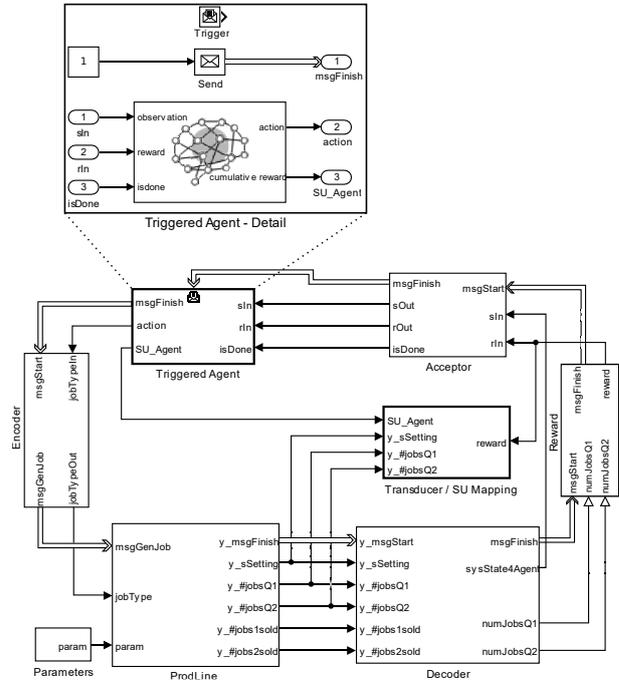


**Figure 8.** Time trajectories of the queue lengths computed using the learned strategy of a self-implemented Q-agent after 5000 episodes.

## 2.4 Learning a Strategy Using a Dedicated RL Toolbox

The MathWorks offers a dedicated RL toolbox for MATLAB/Simulink (MathWorks [10]). This provides an agent block for Simulink, which is configured from MATLAB. Different learning approaches can be configured in the form of agent types as well as hyperparameter settings. Furthermore, the toolbox provides different methods, such as a training method called *train*. In this experiment we use the Q-learning agent and the *train* method of the RL toolbox. It must be noted that the documentation of the toolbox does not contain any hints or examples for the use with event-oriented MUS implemented with the SimEvents blockset. According to the documentation, the agent block of the RL toolbox works signal-oriented. A signal is in Simulink a time-varying quantity that has values at all points in time. Accordingly, the agent block is designed for continuous or discrete-time models with equidistant sampling.

The top-level structure of the DESM for this SBE is shown in Figure 9. With the exception of the *Triggered Agent* block, the DESM corresponds completely to the DESM in Figure 7, i.e., all other components of the EF as well as the MUS were adopted unchanged. Hence, only the *Triggered Agent* is discussed below.



**Figure 9.** Top-level structure of the DESM with MUS and EF for learning a strategy using the RL Toolbox of MATLAB/Simulink, and substructure of the *Triggered Agent*.

The *Triggered Agent*, implements a so-called triggered subsystem and encapsulates the RL agent block of the toolbox. The inner structure of the *Triggered Agent* and the input/output interface of the encapsulated RL agent are also shown in Figure 9. Analogous to the *Agent* in the previous model (cf. Fig. 7), the *Triggered Agent* is activated by the *Acceptor* per event (*msgStart*) when a new state  $s_{t+1}$  of the RL space (input port *sIn*) and a new reward value (input port *rIn*) as well as the boolean *isDone* value have been calculated. If *isDone* is false, the encapsulated RL agent calculates the next *action value*  $a_t$  as well as updates the *cumulative reward value*, and then the *Triggered Agent* activates the *Encoder* by event (*msgFinish*). In the other case, if *isDone* is true, the encapsulated RL agent computes the complete data update of the episode  $SU\_Agent(eoe)$  and, in contrast to the previous model, immediately terminates the episode. I.e., the termination of the episode by the acceptor according to the

previous example is skipped.

The EC defines the parameters  $P_{ExpM}$  and  $P_{SimM}$ , and calls the RL toolbox specific ExpMeth *train*, which uses an RL toolbox specific SimMeth.

### 3 Conclusions

The integration of discrete-event simulation and RL methods has a high application potential for both M&S and AI applications. On the basis of the concept of EF and the general structure of complex SBE, it has been shown how a clear methodological separation can be made so that the MUS, EF, SimMeth, simulator and AI methods – as ExpMeth – can be developed independently and reused in different contexts. The methodological considerations have been practically underpinned by a case study implemented with MATLAB/Simulink and the SimEvents blockset.

In particular, the three experiments of the case study demonstrate that MUS can be developed independently of their experimental context. As shown, this is also true when integrating with the RL method. The adaptation to a concrete experiment can be done by a specific EF, higher-level ExpMeths and a supervisory EC. The basic structure of an EF and the communication relationships in SBE using the RL method were presented.

SBEs in combination with the RL method are characterized by a large number of methodological parameters and variants of agents. Accordingly, the specification of such experiment variants and their automated execution based on the System Entity Structure and Model Base (SES/MB) approach will be investigated in a next step.

### References

- [1] Breiteneker, F. (1992) Models, methods and experiments - a new structure for simulation systems. *Mathematics and Computers in Simulation*, 34(3):231–260.
- [2] Carson, Y., and Maria A. (1997) Simulation optimization: methods and applications. In *Proceedings of the 1997 Winter Simulation Conference*, 118-126.
- [3] Choo, B., Graham, C., Stephen, A., Dadgostari, F., and Beling, P.A. (2020). Reinforcement learning from simulated environments: an encoder decoder framework. In *Proceedings of the SCS SpringSim '20 (Virtual) Conference*, 12 pages.
- [4] FG CEA (2022). Integration of RL and Discrete Event Simulation: A Case study using MATLAB/ Simulink/ SimEvents, Wismar Univ. of Applied Sciences Wismar, <https://github.com/cea-wismar>.
- [5] Pawletta, T. (2019) Specification and execution of simulation models and experiments. *Discussion talk at MS Workshop 'One simulation model is not enough'*, Univ. of Rostock, Dep. of Computer Science, [https://www.cea-wismar.de/pawel/Forschung/Poster\\_Slides/2019-04-23-Presi\\_FG-CEA\\_UnivRo-WS\\_reducedSize.pdf](https://www.cea-wismar.de/pawel/Forschung/Poster_Slides/2019-04-23-Presi_FG-CEA_UnivRo-WS_reducedSize.pdf).
- [6] Pawletta T., Bartelt J. Integration of Reinforcement Learning and Discrete Event Simulation Using the Concept of Experimental Frame. In Mota M.M., editor. *Eurosim Congress 2023*; 2023 Jul; Amsterdam, Netherlands. 8 pages (submitted 2022-Dec-30).
- [7] Schmidt, A. (2019). Variant management in modeling and simulation using the SES/MB framework. In *Advances in Simulation*, Bd. 30, TU Publisher Vienna (in German).
- [8] Sutton, R.S., and Barto, A.G. (2018). *Reinforcement learning: an introduction – 2<sup>nd</sup> edition*. MIT Press.
- [9] The MathWorks (2022-1). SimEvents. <https://mathworks.com/simevents/reinforcement-learning.html>, ©1994-2022 The MathWorks, Inc.
- [10] The MathWorks (2022-2). Reinforcement Learning Toolbox. <https://mathworks.com/products/reinforcement-learning.html>, ©1994-2022 The MathWorks, Inc.
- [11] Traore, M.K., and Muzy, A. (2006). Capturing the dual relationship between simulation models and their context. *Simulation Modeling Practice and Theory*, Elsevier, 14(2006), 126-142.
- [12] Zeigler (1976). *Theory of modeling and simulation – 1<sup>st</sup> edition*. John Wiley & Sons.
- [13] Zeigler (1984). *Multifaceted Modelling and Discrete Event Simulation*. Academic Press.
- [14] Zeigler, B.P., Muzy, A., and Kofman, E. (2018). *Theory of modeling and simulation – 3<sup>rd</sup> edition*. Elsevier, Academic Press.

# Dynamic Time Warping und Synthesedaten zur Validierung von Seq2Seq für die 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](mailto:benjamin.woerrlein@tu-ilmenau.de); [steffen.strassburger@tu-ilmenau.de](mailto:steffen.strassburger@tu-ilmenau.de)

**Abstract.** Seq2Seq is a machine learning method that allows to translate sequences into other sequences. This method has been tried in hybrid simulation of machine tools. The method has been used to generate time series of energy consumption of jobs from the corresponding numerical control code that runs on a machine tool. Seq2Seq suffers from various problems. Firstly, the creation of training data is costly. Secondly, standard Seq2Seq metrics only allow for the evaluation of a prediction of one timestamp at a time, not an entire time series. Thirdly, training metrics are failing when vanilla data is used, as two identical numerical control codes can result in deviating time series. This causes confusion for the model in the training loop, as it is not clear which time series should be considered correct. Here we propose a holistic framework to all three problems, that contains synthetic data, additional metrics for time series and dynamic time warping.

## Einführung

Sequence to Sequence (Seq2Seq) [3, 16] ist eine Klasse von Methoden des maschinellen Lernens (ML), welche es ermöglicht zwei Sequenzen unterschiedlicher Länge und unterschiedlicher Beschreibungen aufeinander abzubilden. Seq2Seq bedient sich hierbei künstlicher, neuronaler Netz und ist dem Deep Learning zugeordnet [4].

Im Kontext der Simulation wurde Seq2Seq schon erfolgreich verwendet um *numerical control codes* (NC-Codes) einer Werkzeugmaschine (WZM) in eine Zeitreihe des Energiebedarfs derselben Maschine zu übersetzen. Hierfür wird zuerst ein Trainingsdatensatz auf Basis einzelner Fertigungsaufträge (FA) erstellt. Dieser enthält pro FA einmal den NC-Code und weiter die gemessene Zeitreihe. Mit ebendiesem Datensatz wird das Seq2Seq-Modell trainiert. Das trainierte Modell kann nun mit einem NC-Code aktiviert werden und gibt anschließend eine Zeitreihe anhand des NC-Codes aus [25]. Weiter kann das Modell innerhalb einer hybriden Simulation

verwendet werden, um beispielweise die Länge eines Fertigungsauftrages oder den Energiebedarf von Maschinen zu prognostizieren [22, 26].

Die Seq2Seq-Methode leidet an mehreren Problemen:

### 1. Trainingsdaten:

Die Beschaffung von Trainingsdaten ist kosten- und zeitintensiv [13, 21]. Gerade bei ML-Methoden ist dieser Effekt besonders stark spürbar, da die Menge an Trainingsdaten mit einem verbesserten Lernverhalten gleichgesetzt wird [5]. Als Lösungsansatz schlagen wir hier die Verwendung von synthetischen Daten vor [13, 21].

### 2. Fehlende Metriken für mehrdeutige Datensätze:

Trainingsdaten für maschinelles Lernen müssen im Regelfall eindeutig sein. Eindeutig heißt hier, dass für jede Stichprobe des Trainingsdatensatzes genau eine Lösung vorliegt. Diese Bedingung wird bei der Erstellung von Trainingsdaten, die NC-Codes und Zeitreihen erhalten, verletzt, wenn bei gleichem NC-Code eine (leicht) unterschiedliche Zeitreihe gemessen wird. Wir schlagen hier vor eine neue Lernmetrik einzuführen, welche die generierten Zeitreihen am Ende einer Trainingsperiode mit einer Vergleichszeitreihe vergleicht.

### 3. Keine Vergleichszeitreihen:

Generierte Zeitreihen mit Trainingszeitreihen zu vergleichen, scheint auf den ersten Blick trivial. Das Problem liegt in zwei Punkten begründet. Erstens vergleicht das Seq2Seq-Modell während des Trainings nicht komplette Zeitreihen, sondern jeden Datenpunkt in den Zeitreihen einzeln [3, 16]. Ein Vergleich von Gesamtzeitreihen findet nicht statt. Zweitens gibt es bei mehrdeutigen Datensätzen keine eindeutige Vergleichszeitreihe, welche verwendet werden könnte, um ganze Zeitreihen miteinander zu vergleichen. Wir schlagen vor, dieses Problem durch *Dynamic Time Warping* (DTW) [1, 15] zu lösen.

In den folgenden Kapiteln werden die Grundlagen der vorgeschlagenen Methode und ein darauf aufbauendes Konzept zur Lösung aller drei Probleme vorgeschlagen. Kern des Konzeptes ist es, nach jedem Trainingslauf eine

Anzahl von Zeitreihen anhand aller möglichen NC-Codes zu generieren. Anschließend werden die generierten Zeitreihen mit durch DTW erzeugten Referenzzeitreihen verglichen. Dies geschieht iterativ am Ende jedes Trainingslaufes. Der Ansatz wird daher hier als *Iterating over Metrics* (IOM) bezeichnet.

Im Anschluss werden die zur Implementierung notwendigen Bestandteile des Konzepts näher erläutert und die Ergebnisse der daraus abgeleiteten Methode vorgestellt.

## 1 Grundlagen des IOM-Ansatzes

Zum besseren Verständnis wird kurz auf die verwendeten Methoden des vorgestellten IOM-Ansatzes eingegangen.

### 1.1 Sequence-to-Sequence

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. Um zeitliche Muster verarbeiten zu können, müssen rekurrente Verbindungen im KNN vorhanden sein, welche eine Rückkopplung abstrahierten Wissens zulassen [27]. Solche rückgekoppelten bzw. rekurrenten neuronalen Netze (RNN) eignen sich besonders für Daten, welche in sequentieller Form vorliegen [4].

Handelt es sich bei den Daten 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 neuronale Schicht codiert, so ist dies ein Encoder. Wird eine Zielsequenz aus einer neuronalen Schicht heraus generiert, so wird dieser Teil als Decoder bezeichnet [4].

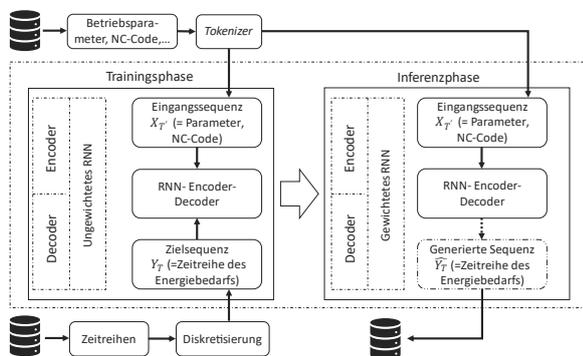


Abbildung 1: Bestandteile einer RNN-Encoder-Decoder Topologie [24].

Eine solche Topologie wird auch allgemein als rekurrente Encoder-Decoder-Netzwerke (RNN-ED) bezeichnet [4]. Abb. 1 beschreibt ebenso ein KNN für den Anwendungsfall von NC-Codes und Zeitreihen.

Beim Training werden sequenziell alle Daten eines Sequenzpaares  $\{X_i, Y_i\}$  durchlaufen und aufeinander abgebildet. Zuerst wird die Eingangssequenz  $X_i$  in einen sogenannten Kontextvektor  $C$  encodiert. Dieser wird anschließend auf die Ausgangssequenz  $Y_i$  decodiert.

Das ML-Model lernt hierbei den Verlauf von  $Y_i$  in dem es den Zustand des Kontextvektors für jeden Eintrag  $y_{i,t}$  aktualisiert. Die Änderung des Zustands des KNN wird hierbei durch eine Fehlerfunktion bestimmt. Das Modell versucht anschließend den Fehler zu minimieren. Der Fehlerwert beschreibt den Unterschied zwischen dem zu lernenden  $y_i$  und dem vorhergesagten Ergebnis  $\hat{y}_i$ . Die Vorhersage  $\hat{y}_i$  geschieht in der letzten Schicht des KNN. In dieser wird eine sog. *Softmax*-Aktivierungsfunktion verwendet. Diese hat zum Nachteil, dass sie je nur einen Eintrag von  $y_i$  mit der generierten Zeitreihe vergleichen kann [4].

Weiterführende Erläuterungen zum hier verwendeten Encoder-Decoder können [3, 4, 16] entnommen werden.

### 1.2 Synthesedaten durch Simulation

Ein grundlegendes Problem bei der Anwendung von Deep Learning ist die Verfügbarkeit von Trainingsdaten. Diese müssen gesammelt, auf Unstimmigkeiten hin untersucht, mit analytischen Methoden vorverarbeitet usw. werden. Dies geht teils mit hohen Kosten und Zeitaufwand einher. Im hier zugrundeliegenden Beispiel des Datensatzes von NC-Codes und Zeitreihen des Energiebedarfs von Fertigungsaufträgen wird dies deutlich.

Verfügt die betrachtete Werkzeugmaschine über keine passende Schnittstelle, muss der NC-Code und der Zeitraum, in dem dieser läuft, händisch aufgenommen und in ein Datenformat übertragen werden.

Noch aufwendiger verhält sich die Erstellung des Datensatzes der Energiezeitreihen. Hierfür muss eine geeignete Messtechnik und Software identifiziert, implementiert und schlussendlich angewandt werden. Dies stellt hohe Anforderung an verfügbares Personal, sowie die Verfügbarkeit von WZM und Messtechnik.

Sind die Voraussetzungen für eine Aufnahme des NC-Codes und der Zeitreihen erfüllt, stellt sich anschließend die Frage, ob die WZM im betrachteten Zeitraum eine Auslastung hat, die so hoch ist, dass damit eine

große Anzahl von Trainingsdaten erzeugt werden kann. Gerade im Hinblick auf die Größe der im Deep Learning verwendeten Datensätze (>10k) scheint dies unwahrscheinlich.

Abhilfe kann hier die Verwendung von Synthesedaten schaffen. Synthesedaten sind künstlich erzeugte Daten, welche in Struktur (Länge, Merkmalen, Merkmalshäufigkeit, etc.) den Originaldaten ähneln. Der Vorteil von Synthesedaten ist, dass diese kostengünstig, transparent und reproduzierbar erstellt werden können. Synthesedaten können durch ein Simulationsmodell generiert werden und anschließend einem ML-Model als Trainingsgrundlage dienen.

Die Verwendung von Synthesedaten im Maschinellen Lernen ist nicht neu und stellt ein eigenes Forschungsfeld dar [20]. So haben *Melo et. al.* [13] gezeigt, dass Synthesedaten zum Training von Bilderkennungsmodellen verwendet werden können. Weitere Anwendungsfälle für die Anwendung von Synthesedaten im Deep Learning sind bspw. die zerstörungsfreie Prüfung von Stahl [2], Objekterkennung [23], zur Erstellung von Fahrzeugbegrenzungsrahmen im autonomen Fahren [20] oder Fußgängererkennung in Bilddaten [7].

Auch die Erstellung von Synthesedaten durch Simulationstechniken ist kein Novum. So wurde Data Farming verwendet, um eine Datenbasis für Deep Learning in der Simulation von Produktionssystemen [9] oder der Objekterkennung für Roboter [19], zu schaffen. Weitere Anwendungsfälle waren bspw. die Erstellung von Mobilitätsdaten [8, 12], Bilddaten zur Herzgewebebestimmung [10] oder Daten der Fertigungsplanung und -steuerung [6].

### 1.3 Dynamic Time Warping

Zeitreihen beschreiben den Verlauf eines Merkmals über die Zeit, meist für eine feste Messstrecke, d.h. einem gleichen Intervall zwischen den einzelnen Messpunkten. Im Resultat ist eine Zeitreihe lediglich eine Sammlung von zeitlich sortierten Datenpunkten.

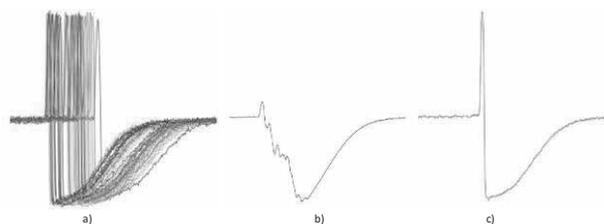
Der Vergleich von Zeitreihen miteinander erscheint zuerst wie ein triviales Problem. Vergleicht man die Zeitreihen einer festen Periode miteinander, beispielsweise der Temperaturverlauf über einen Tag, so könnte man die Zeitreihen für zwei Tage einfach übereinanderlegen, den Mittelwert dieser bilden, um das Temperaturmittel zu bestimmen; oder sie voneinander abziehen, um die Temperaturdifferenz zwischen zwei Tagen zu bestimmen.

Bei Zeitreihen, die keiner festen Periode folgen gestaltet sich der Vergleich von Zeitreihen schwieriger. Beispielsweise im hier genannten Anwendungsfall von Zeitreihen des Energiebedarfs eines FA. Werkzeugmaschinen sind dynamische Systeme, d.h. dass diese ihren Zustand bei jedem Ereignis ändern, was zur Folge hat, dass die Zeitreihen des gleichen NC-Codes und der gleichen Maschine unterschiedlich aussehen werden. Der Unterschied ist hier in beiden Dimensionen der Zeitreihe, der Anzahl von Datenpunkten und der Merkmalsausprägung der Zeitreihe festzustellen.

Möchte man nun solche Zeitreihen dynamischer Systeme miteinander vergleichen, stellen sich folgende Probleme.

Erstens haben die Zeitreihen eine unterschiedliche Anzahl an Datenpunkten. Die Länge einer Zeitreihe wird durch ihre Anzahl an Datenpunkten bestimmt. Besteht hier eine Abweichung können die zusätzlichen Datenpunkte einer Zeitreihe nicht mit den Datenpunkten der anderen Zeitreihe verglichen werden, da diese nicht vorhanden sind. Streicht man nun die zusätzlichen Datenpunkte, so könnte man die Zeitreihen wieder miteinander vergleichen, aber es würde potenziell wichtige Datenpunkte dabei verloren gehen. Alternativ könnte man die kürzere Zeitreihen mit Werten befüllen, bis sie die Länge der längeren Zeitreihe erreicht hat. Das Problem hierbei ist es, einen Wert zu finden mit dem man die kürzere Zeitreihe auffüllt, ohne dabei die Zeitreihe nachteilig zu ändern.

Zweitens können sich charakteristische Verläufe auf den Zeitreihen durch die fehlenden Datenpunkte in ihrer Lage verschoben haben. Solche Verläufe wären bspw. der Temperaturverlauf über den Tag mit einem Hoch zur Mittagszeit oder wie im hier genannten Anwendungsfall wiederkehrende Muster im Spannverlauf einer WZM.

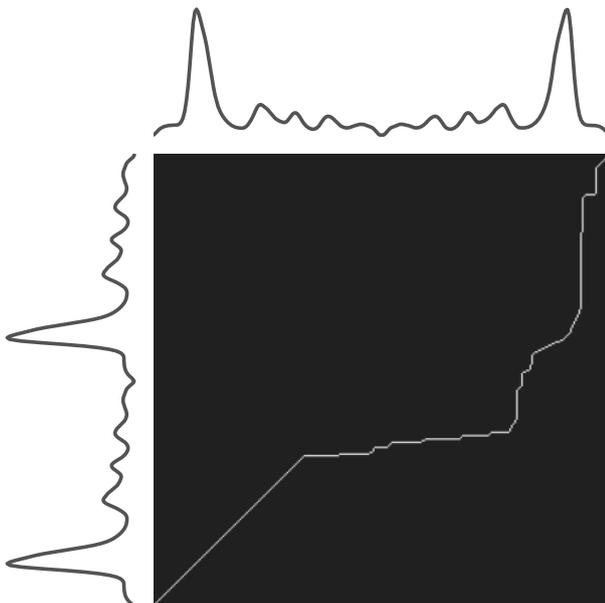


**Abbildung 2:** Vergleich von a) einer Menge an Zeitreihen, b) deren gemittelte Zeitreihe und c) eine durch DTW gemittelte Zeitreihe (Abbildungen entnommen aus [14]).

Betrachtet man Abb. 2 werden die oben genannten Probleme noch einmal verdeutlicht. Abb. 2 a) stellt hier

eine Menge an Zeitreihen dar, welche alle leicht unterschiedlich lang sind und deren Charakteristika (Steigung, Gefälle, Anstieg auf Ausgangswert) sich in Lage und Ausprägung unterscheiden. Würde man die Zeitreihen nur mit Werten auffüllen (hier dem Mittelwert der Zeitreihen) könnte man den Mittelwert aller Zeitreihen bilden. Abb. 2 b) stellt das Ergebnis dieser gemittelten Zeitreihe dar. Es ist erkennbar, dass diese gemittelte Zeitreihe, außer in der Länge, nicht mehr mit den Ausgangszeitreihen vergleichbar ist.

Eine Methode die Zeitreihen unterschiedlicher Länge mit unterschiedlicher Lage ihrer Charakteristika miteinander vergleichen kann heißt *Dynamic Time Warping* (DTW) [1, 15]. DTW ist ein Algorithmus, welcher eine Menge an Zeitreihen miteinander über eine sog. Distanzmatrix vergleicht. Die Distanzmatrix vergleicht die Datenpunkte der beiden Zeitreihen durch ein Distanzmaß, wie euklidische Distanz, Manhattan Distanz usw. Beim Vergleich von zwei Datenpunkten wird deren Unterschied in der Distanzmatrix notiert. Anschließend wird ein Pfad in der Distanzmatrix aufgespannt, der sog. *Warping Path* (siehe Linie in Abb. 3). Der optimale *Warping Path* ist der Pfad, der in Summe die geringste Distanz durch die Distanzmatrix aufweist.



**Abbildung 3:** Distanzmatrix von zwei durch Dynamic Time Warping verglichenen Zeitreihen (links und oben zu sehen) [17].

Liegen die Datenpunkte nah beieinander, steigt der *Warping Path* in jeder Dimension gleich stark an (vgl. Abb. 3 *Warping Path* – links unten).

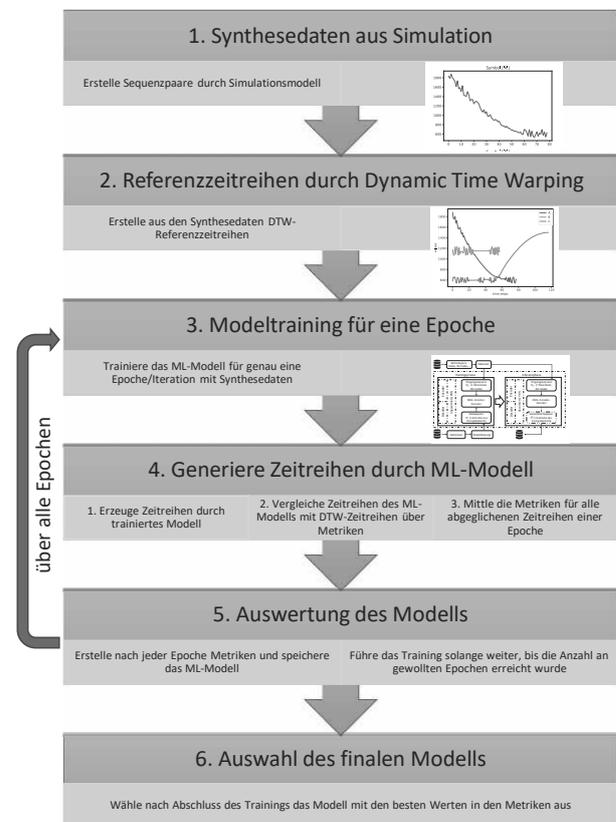
Weichen die Datenpunkte der beiden Zeitreihen voneinander ab, so flacht sich der *Warping Path* ab oder steigt stark an.

Wurde ein *Warping Path* gefunden, kann dieser verwendet werden, um eine durch DTW gemittelte Zeitreihe zu erstellen. Als Beispiel sei hier Abb. 2 c) angeführt. Die hier gezeigte Zeitreihe ist in ihrer Länge und Form klar mit den einzelnen Zeitreihen aus Abb. 2 a) vergleichbar.

Weiterführende Literatur zu Dynamic Time Warping kann [1, 14, 15, 17] entnommen werden.

## 2 Konzept

Der IOM-Ansatz besteht aus mehreren Schritten (vgl. Abb. 4).



**Abbildung 4:** Konzept des IOM-Ansatzes

Zuerst, um dem Kosten- und Verfügbarkeitsproblem von Trainingsdaten zu entgehen, werden diese über ein Simulationsmodell erstellt.

Im zweiten Schritt werden Referenzzeitreihen erstellt. Diese dienen dazu, durch das Modell erzeugte Zeitreihen vergleichbar zu machen. Dies geschieht anhand von *Dynamic Time Warping*, wobei die Datengrundlage die zuvor erzeugten Synthesedaten darstellen.

Drittens wird nun das Training des ML-Modells gestartet. Das Modell trainiert genau für eine Iteration (Anm.: Zur Verringerung der Berechnungszeiten wurden 10 Epochen in einer Iteration zusammengefasst).

Im vierten Schritt wird das trainierte Modell verwendet, um Zeitreihen zu generieren. Der Vergleich von kompletten Zeitreihen ist nötig um mit dem in Kapitel 1.1 beschriebenen Problem umzugehen, dass die im Modell verwendete *Softmax*-Funktion nur schrittweise einzelne Datenpunkte vergleicht, nicht aber eine gesamte Zeitreihe.

Die Güte der generierten Zeitreihen wird anschließend über einen Vergleich mit den durch DTW erzeugten Referenzzeitreihen bestimmt. Wurden über Metriken Vergleiche für alle betrachteten Zeitreihen erstellt, werden diese im Anschluss zusammengefasst. Hierbei wird der Mittelwert jeder einzelnen Metrik für alle Zeitreihen gebildet.

Im anschließenden fünften Schritt geht das Modell wieder ins Training über und wiederholt Schritt 3 und 4 so lange bis eine vorher festgesetzte Anzahl an Epochen erreicht wurde. Auch hier wird wieder bei jedem Durchlauf eine Zusammenfassung der Metriken erstellt und das trainierte Modell gespeichert.

Im sechsten und letzten Schritt wird das Modell ausgewählt, welches die besten Resultate in den betrachteten Metriken hatte.

### 3 Versuchsvorbereitung

Zur Umsetzung des IOM-Konzeptes ist es nötig verschiedene Vorbereitungsschritte durchzuführen. Diese sind einerseits die Erstellung von Synthesezeitreihen. Weiter müssen Vergleichszeitreihen über *Dynamic Time Warping* erstellt werden. Anschließend müssen die im vorhergehenden Kapitel genannten Metriken erstellt und in das Modeltraining eingebunden werden.

#### 3.1 Erstellung Synthesezeitreihen

Zur Erstellung von Synthesezeitreihen wird das Simulationstool *Anylogic* (vgl. Abb. 5) verwendet. In *Anylogic* wird ein diskret ereignisorientiertes Simulationsmodell angelegt. In der Quelle des Modells werden zufällig Fertigungsaufträge (A, B oder C) erzeugt. Diese gelangen über eine Warteschlange in einen Warteraum. Es kann sich immer nur ein FA im Warteraum befinden. Jedem FA wird eine eigene Wartezeit zugeordnet. Die Wartezeit im Warteraum folgt einer stetigen Gleichverteilung von  $[0;95;1]$ . Sobald der Warteraum mit einem FA belegt ist,

wird eine dem FA zugehörige Funktion ausgeführt. Diese Funktionen unterscheiden sich für alle drei Typen von FA (vgl. Abb.6). Es handelt sich hier um mathematische Funktionen, welche als Eingangsparameter die verbleibende Wartezeit des FA erhalten. Die Ausgaben der Funktionen ändern sich daher über die Zeit. Der berechnete Wert wird anschließend durch eine weitere Gleichverteilungsfunktion geführt und ausgegeben.

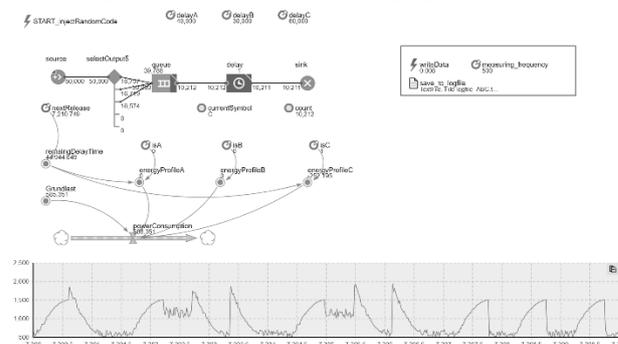


Abbildung 5: *Anylogic* Simulationsmodell.

Durch die Gleichverteilung im Warteraum und in den Ausgabefunktionen wird sichergestellt, dass die Zeitreihen gleicher FA sich jedes Mal leicht unterscheiden. Dies trägt dem eingangs beschriebenen Problem Rechnung, dass sich zwei gemessene Zeitreihen des gleichen FA nie exakt in Länge und Werteverlauf gleichen. Abb. 6 stellt Stichproben der erzeugten Zeitreihen dar. Die Gleichverteilung in den Ausgabewerten ist hier klar zu erkennen.

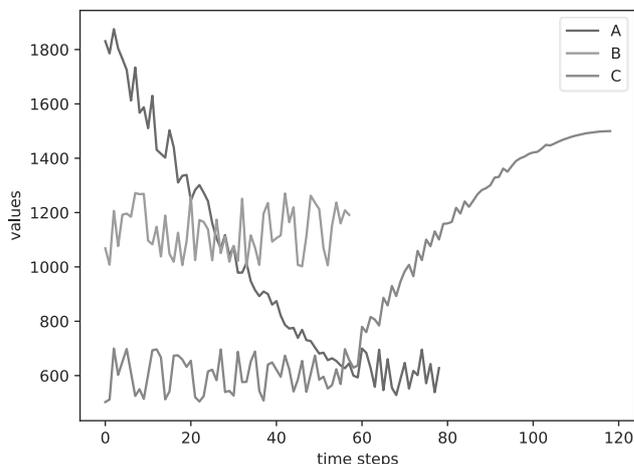
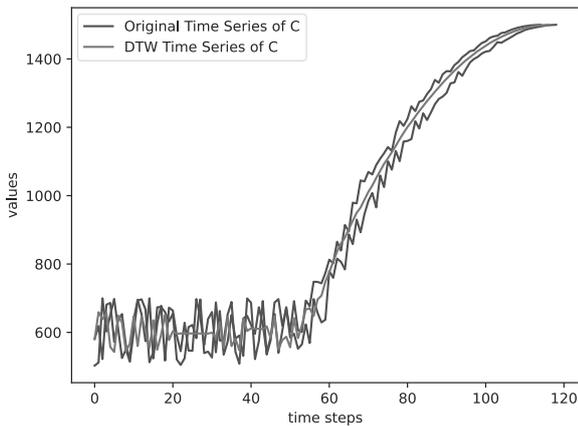


Abbildung 6: Stichprobe der durch *Anylogic* erzeugten Zeitreihen.

#### 3.2 Generierung der Referenzzeitreihen durch Dynamic Time Warping

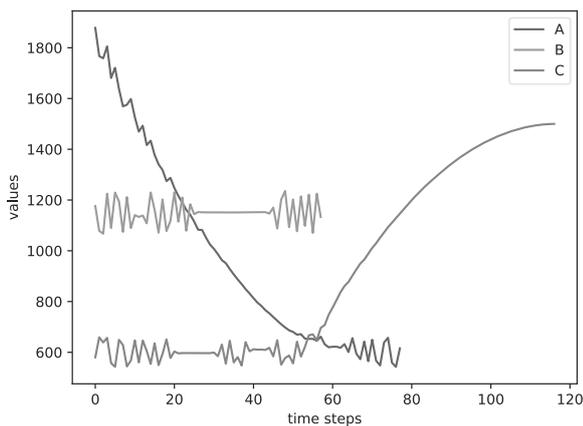
Zur Erstellung von Referenzzeitreihen wird *Dynamic Time Warping* verwendet. DTW erzeugt aus einer Menge

von Zeitreihen eine Referenzzeitreihe, welche in Länge und Struktur einen minimalen Gesamtfehler zur Ausgangsmenge aufweist. Die Berechnung erfolgt anhand der Python Bibliothek *tslearn* [17]. Der spezifisch hier verwendete DTW Algorithmus ist *softDTW* [11]. *softDTW* weist eine höhere Fehlertoleranz in der Erstellung des *Warping Path* auf, welche sich positiv auf die Erstellung von gemittelten Zeitreihen auswirkt [11].



**Abbildung 7:** Vergleich von 2 Zeitreihen des Fertigungsauftrags C mit der für C durch *softDTW* erzeugten Zeitreihe.

Abb. 7 zeigt eine durch *softDTW* erzeugte Zeitreihe im Vergleich zu zwei Stichproben des Fertigungsauftrags C. Es ist deutlich zu erkennen, dass die *softDTW* Zeitreihe die Stichproben gut zusammenfasst, diese aber nicht einfach kopiert, sondern in Form und Länge nachahmt.



**Abbildung 8:** Mit *softDTW* erzeugte Zeitreihen der drei Fertigungsaufträge.

Abb. 8 zeigt die *softDTW* Zeitreihen für alle 3 FA. Im Vergleich mit Abb. 6 wird der Unterschied zwischen den Ausgangs- und DTW-Zeitreihen deutlich. Die Zeit-

reihen weisen alle eine ähnliche Länge und Form auf, unterscheiden sich aber leicht, da sie aus der gesamten Menge an Zeitreihen berechnet wurden.

### 3.3 Einbinden der Metriken in das Seq2Seq-Modell

Während des Modelltrainings sollen die generierten Zeitreihen mit den DTW-Zeitreihen als Referenz verglichen werden. Im Anschluss daran werden die Werte der Metriken der einzelnen Zeitreihen für eine gesamte Trainingsepoche gemittelt.

Hierfür wurden zwei Metriken verwendet. Einmal wird der *mean squared error* (MSE) angewandt. Dieser vergleicht die Zeitreihen, indem er die Differenz zwischen jedem einzelnen Datenpunkt berechnet und diese anschließend quadriert. Im Anschluss werden alle Differenzen miteinander addiert, um eine Aussage über die Güte der Gesamtzeitreihe zu erhalten. Die Quadratur der Differenz hat zwei Vorteile. Erstens werden alle Differenzen positiv, was in der anschließenden Addition dazu führt, dass sich ein negativer und positiver Fehlerwert nicht gegenseitig auslöschen. Zweitens werden durch die Quadratur große Fehler stärker gewichtet als einzelne.

Die zweite Metrik *sigma length* betrachtet lediglich eine Dimension der Zeitreihe, nämlich ihre Länge. Hier wird die Anzahl der Datenpunkte in beiden Zeitreihen ermittelt und anschließend durcheinander geteilt. Desto näher das Ergebnis an 1 liegt, desto ähnlicher sind sich generierte und Referenzzeitreihe in ihrer Länge.

```
def results(ml_ts, dtw_ts, iteration):
```

```
    mean_squared_error_accuracy =
    mean_squared_error(ml_ts, dtw_ts)

    sigma_length_accuracy =
    len(ml_ts) / len(dtw_ts)

    results =
    {"Iteration": iteration,
     "MSE": mean_squared_error_accuracy,
     "Sigma_Length": sigma_length_accuracy}
```

```
    return results
```

**Code 1:** Pseudocode der Metriken *MES* und *sigma length* für eine Zeitreihe (*ml\_ts* steht für die durch das ML-Modell erzeugten Zeitreihen; *dtw\_ts* für die durch DTW erzeugten).

Wird ein Trainingslauf beendet, werden mit dem trainierten Modell Zeitreihen erzeugt. Diese werden anschließend mit denen in Code 1 beschriebenen Metriken

verglichen und die Resultate der Metriken für jede Zeitreihe gespeichert. Anschließend wird nun eine Zusammenfassung für die gesamte Iteration berechnet. Dies soll eine ganzheitliche Aussage über das Modell zu einem bestimmten Trainingsstand ermöglichen. Hierbei werden alle MSE und *sigma length* Resultate gemittelt. Der Mittelwert erlaubt eine robustere Aussage über die Fähigkeit des Modells Zeitreihen zu erzeugen als die Betrachtung von Einzelzeitreihen.

## 4 Versuchsdurchführung

Die Erstellung der Synthesezeitreihen wurde in *Anylogic* durchgeführt. Als Programmiersprache wurde *Python* verwendet. Die Berechnung der DTW-Zeitreihen geschah in *tslearn* [17], die Erstellung des ML-Modells in *Tensorflow* [18]. Als Hardware standen zur Verfügung ein *Ryzen 7 2700 X* Prozessor, eine *RTX 2080Ti* Grafikkarte mit 4352 Cuda Kernen, sowie eine SSD mit 480 GB Speicher.

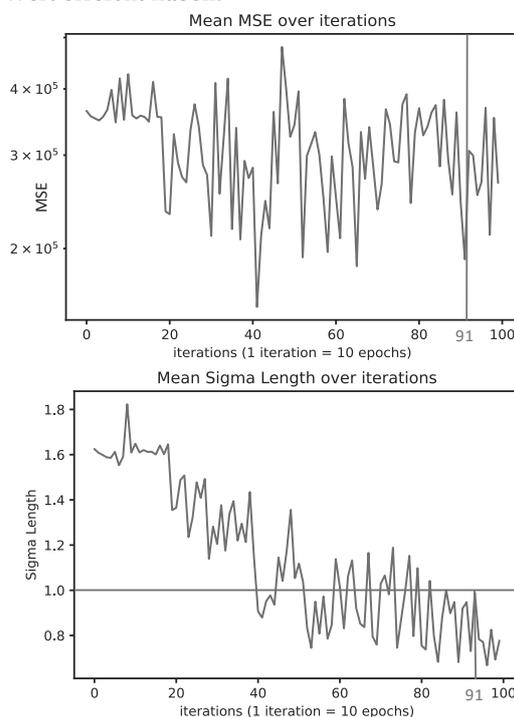
Der IOM-Ansatz sieht vor das ML-Modell nach jeder Epoche zu speichern. Dies konnte aber mit der vorhandenen Speicherkapazität nicht umgesetzt werden. Daher wurde entschieden das Modell je 10 Epochen (=1 Iteration) trainieren zu lassen und nur für die letzte Epoche einer Iteration das Modell zu speichern und Metriken zu erheben. Das Modell trainierte für 1000 Epochen.

Bei den Trainingsdaten handelt es sich um den in Kapitel 3 beschriebenen, wobei die Sequenzpaare aus jeweils 2 FA gebildet wurden, um die Varianten im Datensatz zu erhöhen. Der Datensatz hatte eine Größe von 10000 Sequenzpaaren.

### 4.1 Abgleich der Metriken

Wie im IOM-Konzept beschrieben, sollen am Ende jeder Iteration gemittelte Metriken gebildet werden. Der Verlauf dieser Metriken wurde in Abb. 9 aufgetragen. Hier werden einmal Ergebnisse des MSE und des *sigma length* dargestellt. Um nun das optimale Modell auszuwählen, wird das Optimum der beiden Metriken bestimmt. Das Optimum einer Metrik ist der Wert, in dem die Metrik das beste Resultat erzielt. Bei MSE wäre dies 0, da MSE die Differenz zwischen allen Datenpunkten betrachtet. Bei *sigma length* hingegen wäre es 1, da hier ein Verhältnis der Länge einer Zeitreihe zur anderen untersucht wird. Zur Veranschaulichung wurde das Optimum bei *sigma length* grafisch eingetragen (blaue Linie). Alle Ite-

rationen, die nun auf der blauen Linie liegen, stellen Modelle dar, die in der *sigma length* Metrik den optimalen Wert erreicht haben.



**Abbildung 9:** Metriken des IOM-Ansatzes. Oben: gemittelter MSE. Unten: gemittelter *sigma length*. Die blaue Linie markiert das Optimum von *sigma length* (=1). Die grüne Linie ist die ausgewählte Iteration.

In einem zweiten Schritt werden nun Iterationen, in denen *sigma length* = 1 gilt auf ihren Wert in der MSE-Metrik hin untersucht. Die grüne Linie in Abb. 9 bei Iteration 91 (=Epoche 910) entspricht der Iteration, bei der die Werte für beide Metriken am niedrigsten sind. Es wird daher das Modell der Epoche 910 ausgewählt.

### 4.2 Visualisierung der Inferenz- und Dynamic Time Warping Zeitreihen

Das Modell kann nun verwendet werden, um Zeitreihen zu generieren. Abb. 10 zeigt alle 9 möglichen Varianten auf die das ML-Modell trainiert wurde. Die blaue Zeitreihe wurde vom ML-Modell generiert. Die orange Zeitreihe ist die durch DTW erzeugte Referenzzeitreihe. Der visuelle Abgleich bestätigt das Funktionieren des IOM-Ansatzes.

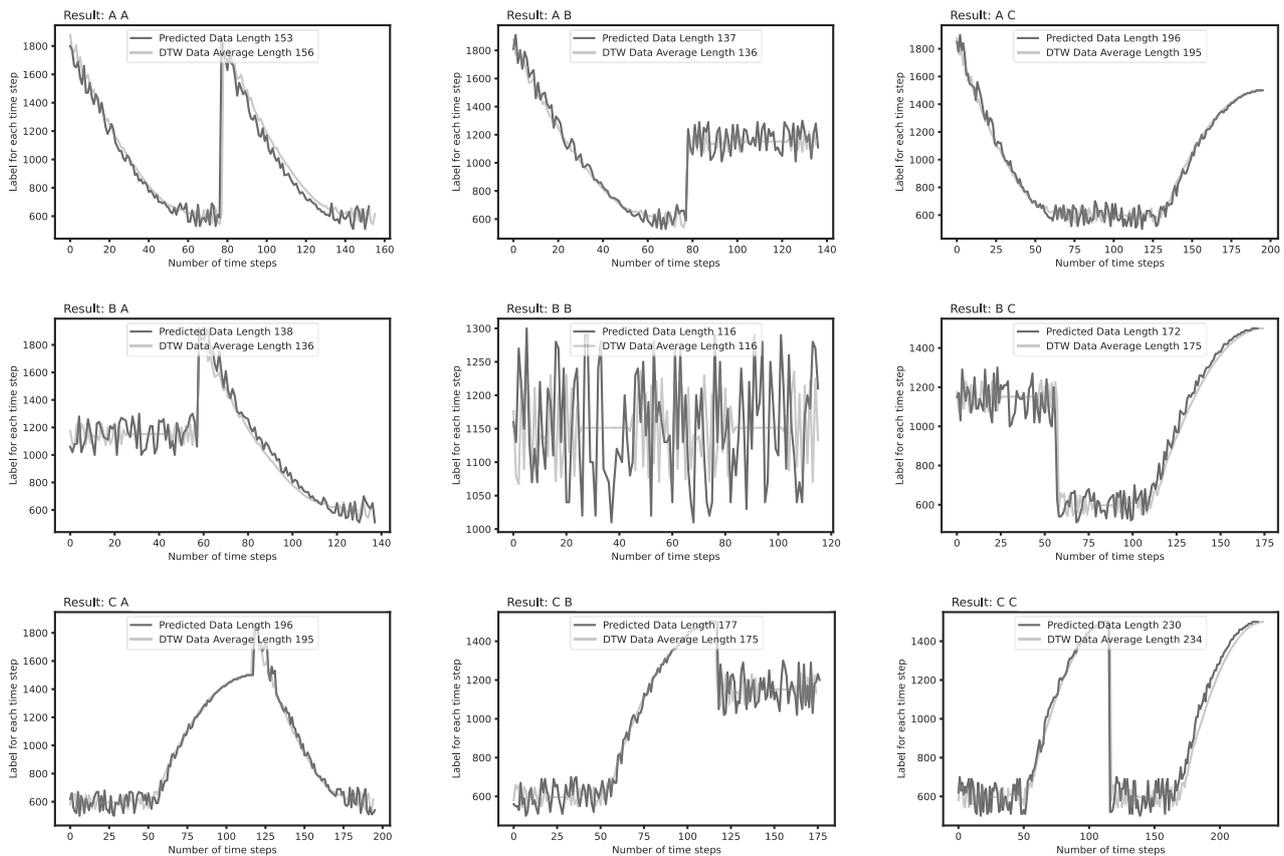


Abbildung 10: Vergleich aller bei Epoche 910 erstellten Zeitreihen mit den DTW-Referenzzeitreihen. Die Legende in den Varianten gibt die Längen der beiden Zeitreihen wieder.

## 5 Kritische Betrachtung

Der vorgestellte IOM-Ansatz konnte die an ihn gestellten Anforderungen erfüllen.

Einerseits wurde gezeigt, dass durch Simulation erzeugte Synthesedaten für das Training von ML-Modellen verwendet werden können. Diese konnten kostengünstig, transparent und auf den Anwendungsfall passend erzeugt werden.

Weiter wurde durch die Implementierung von speziellen Metriken erreicht, dass nun Gesamtzeitreihen miteinander verglichen werden konnten und nicht nur wie im Seq2Seq Basis Model einzelne Datenpunkte über die *Softmax*-Funktion.

Daraufhin wurde gezeigt, dass DTW-Zeitreihen dazu verwendet werden können, Referenzzeitreihen für den Abgleich in den Metriken zu bilden. Die Verwendung von DTW war nötig, um das Problem der Nichteindeutigkeit in einer Menge von Zeitreihen, mit gleichen Beschreibungen (bspw. gleicher NC-Code) zu lösen.

Abschließend konnten so Zeitreihen generiert werden, welche sich zu den analytisch erzeugten Dynamic Time Warping Zeitreihen kaum unterscheiden. Die Zeitreihen sind in Form und Länge fast identisch.

Die Methode selbst bietet Potential für weitere Forschungsfragen. So könnte sie weiter validiert werden, indem der Datensatz um ein Vielfaches vergrößert oder mehrere verschiedene FA in der Simulation erzeugt werden. Weiter hat das ML-Model hier nur Zeitreihen von NC-Codes erzeugt, welche schon in den Trainingsdaten vorhanden waren. Es gilt zu untersuchen inwieweit das Model Zeitreihen generieren kann, im Falle eines unbekanntenen NC-Codes.

Nachteil der vorgestellten Methode ist die Anforderung an Rechen- und Speicherleistung. So muss für jede Iteration ein Modell gespeichert werden, obwohl nach Bestimmung des optimalen Modells alle anderen gelöscht werden können. Auch die Berechnung der Metriken übersteigt bei epochenweiser Erstellung die hier für das Training des Modells aufgewandte Zeit.

## 6 Literaturverzeichnis

- [1] Donald J. Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *KDD workshop*, 359–370.
- [2] Aleksei Boikov, Vladimir Payor, Roman Savelev, and Alexandr Kolesnikov. 2021. Synthetic Data Generation for Steel Defect Detection and Classification Using Deep Learning. *Symmetry* 13, 7, 1176. DOI: <https://doi.org/10.3390/sym13071176>.
- [3] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and 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.
- [4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.
- [5] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md M. A. Patwary, Yang Yang, and Yanqi Zhou. 2017. *Deep Learning Scaling is Predictable, Empirically*.
- [6] Sanjay Jain, Anantha Narayanan, and Yung-Tsun T. Lee. 2019. Infrastructure for Model Based Analytics for Manufacturing. In *2019 Winter Simulation Conference (WSC)*. IEEE. DOI: <https://doi.org/10.1109/wsc40007.2019.9004893>.
- [7] Hadi Keivan Ekbatani, Oriol Pujol, and Santi Seguí. 2017. Synthetic Data Generation for Deep Learning in Counting Pedestrians. In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods*. SCITEPRESS - Science and Technology Publications. DOI: <https://doi.org/10.5220/0006119203180323>.
- [8] Amir T. Kutjev and Yana A. Bekeneva. 2021. Simulation Software for Generating Data in Monitoring Systems. In *Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElCon-Rus)*. January 26-28, 2021 : St. Petersburg and Moscow, Russia, 2021. IEEE, Piscataway, NJ, 486–489. DOI: <https://doi.org/10.1109/ElCon-Rus51938.2021.9396098>.
- [9] Tobias Lechler, Martin Sjarov, and Jörg Franke. 2021. Data Farming in Production Systems - A Review on Potentials, Challenges and Exemplary Applications. *Procedia CIRP* 96, 230–235. DOI: <https://doi.org/10.1016/j.procir.2021.01.156>.
- [10] Michael Loecher, Luigi E. Perotti, and Daniel B. Ennis. 2021. Using synthetic data generation to train a cardiac motion tag tracking neural network. *Medical image analysis* 74, 102223. DOI: <https://doi.org/10.1016/j.media.2021.102223>.
- [11] Marco Cuturi and Mathieu Blondel. 2017. Soft-DTW: a Differentiable Loss Function for Time-Series. In *Proceedings of the 34th International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR, 894–903.
- [12] Pasqual Martí, Jaume Jordán, Javier Palanca, and Vicente Julian. 2022. Charging stations and mobility data generators for agent-based simulations. *Neurocomputing* 484, 196–210. DOI: <https://doi.org/10.1016/j.neucom.2021.06.098>.
- [13] Celso M. de Melo, Antonio Torralba, Leonidas Guibas, James DiCarlo, Rama Chellappa, and Jessica Hodgins. 2022. Next-generation deep learning based on simulators and synthetic data. *Trends in cognitive sciences* 26, 2, 174–187. DOI: <https://doi.org/10.1016/j.tics.2021.11.008>.
- [14] François Petitjean, Germain Forestier, Geoffrey I. Webb, Ann E. Nicholson, Yanping Chen, and Eamonn Keogh. 2016. Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. *Knowl Inf Syst* 47, 1, 1–26. DOI: <https://doi.org/10.1007/s10115-015-0878-8>.
- [15] Petitjean François, Forestier Germain, Webb Geoffrey I., Nicholson Ann E., Chen Yanping, and Keogh Eamonn. 2014. Dynamic Time Warping Averaging of Time Series Allows Faster and More Accurate Classification. In *2014 IEEE International Conference on Data Mining*, 470–479. DOI: <https://doi.org/10.1109/ICDM.2014.27>.
- [16] Ilya Sutskever, Oriol Vinyals, and 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.
- [17] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. 2020. Tslern, A Machine Learning Toolkit for Time Series Data. *Journal of Machine Learning Research* 21, 118, 1–6.
- [18] TensorFlow Developers. 2022. *TensorFlow*. Zenodo.
- [19] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. *Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World*.
- [20] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil,

- Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. 2018. *Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization*.
- [21] Aleksei Triastcyn and Boi Faltings. 2018. *Generating Artificial Data for Private Deep Learning*.
- [22] Benjamin Woerrlein and Steffen Strassburger. 2020. A Method for Predicting High-Resolution Time Series Using Sequence-to-Sequence Models. In *2020 Winter Simulation Conference (WSC)*. IEEE, 1075–1086. DOI: <https://doi.org/10.1109/WSC48552.2020.9383969>.
- [23] Matthew Z. Wong, Kiyohito Kunii, Max Baylis, Wai H. Ong, Pavel Kroupa, and Swen Koller. 2019. Synthetic dataset generation for object-to-model deep learning in industrial applications. *PeerJ. Computer science* 5, e222. DOI: <https://doi.org/10.7717/peerj-cs.222>.
- [24] Benjamin Wörrlein and 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>.
- [25] Benjamin Wörrlein and 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>.
- [26] Benjamin Wörrlein and Steffen Straßburger. 2022. Hochaufgelöste Energieprofile durch hybride Simulation. In *ASIM SST 2022 Proceedings Langbeiträge*. 26. ASIM Symposium Simulationstechnik, 25.07.-27.07.2022, TU Wien. ASIM Mitteilung, 180. ARGESIM Verlag, Wien, 243–251. DOI: <https://doi.org/10.11128/arep.20.a2004>.
- [27] Andreas Zell. 2003. *Simulation neuronaler Netze* (4., unveränd. Nachdr). Oldenbourg, München.

# Simulationsbasierte Untersuchung von Energieflexibilität bei der Optimierung in der Hinterlandentwässerung

Marvin Hempel<sup>1\*</sup>, Jens Heger<sup>1</sup>

<sup>1</sup>Institut für Produktionstechnik und -systeme, Leuphana Universität Lüneburg, Universitätsallee 1, 21335 Lüneburg, Germany; \*marvin.hempel@leuphana.de

**Abstract.** In this paper, the issue of adjusting pump activities to fluctuating energy availability in the context of hinterland drainage systems is addressed. Through mathematical modeling and optimization techniques, the potential for flexibility within regular pumping operations is evaluated. In addition to the intuitive approach which considers only the tides of the adjacent water body for the purpose of enhancing energy efficiency, electricity price patterns are also taken into consideration to promote grid-friendly behavior and reduce CO<sub>2</sub> emissions.

## Einleitung

Die fortschreitende Energiewende verändert die Art und Weise, wie ein zeitgemäßer Umgang mit energieintensiven Anlagen umzusetzen ist. Durch den steigenden Anteil an erneuerbaren Energiequellen im Stromnetz erhöht sich naturgemäß die Fluktuation der Energieverfügbarkeit [1]. Diese Schwankungen haben einen direkten Einfluss auf die Energiepreise am Strommarkt. Tendenziell sind die Strompreise bei einem hohen Energieaufkommen aus erneuerbaren Quellen geringer und umgekehrt. Daraus ergibt sich neben dem Gedanken der Energieeffizienz ein zusätzliches Potenzial der Flexibilisierung, durch dessen Nutzung betreiberseitig Energiekosten gesenkt, gleichzeitig indirekte CO<sub>2</sub>-Emissionen reduziert und das Stromnetz entlastet werden können.

Infolgedessen sollten die kosten- und energieintensive Pumpvorgänge bei der Hinterlandentwässerung durch intelligente Steuerungssysteme an die heute deutlich komplexeren Bedingungen am Strommarkt angepasst werden, um ganzheitlich optimierte Abschöpfstrategien zu gewährleisten. Die klimawandelbedingte Zunahme der Intensität von Regenereignissen und die steigenden Strompreise erschweren es den Schöpfwerksbetreibern ohnehin geeignete Entwässerungsmaßnahmen auf der Grundlage veralteter Konzepte anzuwenden. Es

reicht teilweise nicht aus manuelle, binäre Entscheidungen zu treffen und bei hohen Innen- bzw. niedrigen Außenpegelständen die Pumpen unter Volllast zu betreiben.

Durch Drehzahlregelung der Pumpen besteht z.B. die Möglichkeit unterschiedliche Wirkungsgrade zu erreichen, wobei eine geringe Drehzahl einen deutlich reduzierten Energieverbrauch zur Folge hat. Ebenfalls ist aber auch ein längerer Zeitraum notwendig, um dasselbe Volumen an Wasser abzuleiten und die Förderhöhe, die überwunden werden kann, reduziert sich. Die Wahl der optimalen Drehzahl hängt zu jeder Zeit unmittelbar von den aktuellen Pegelständen und dessen Änderungsrate als auch vom aktuellen Strompreis ab.

Um der globalen Zunahme an CO<sub>2</sub>-Emissionen und den steigenden Energiekosten entgegenzuwirken, müssen also umfangreiche Kenntnisse generiert werden, die die vielschichtigen Zusammenhänge zwischen den kontrollierbaren (z.B. Pumpdauer, Drehzahl der Pumpe) und unkontrollierbaren (z.B. Gezeiten, Pegelstände, Witterungsbedingungen) Einflussfaktoren eines Entwässerungssystems abbilden. In Zukunft kann, mithilfe von Prognosedaten für Niederschlag, Gezeiten und Stromverfügbarkeit, eine angebotsorientierte Flexibilität des Energieverbrauchs erreicht werden.

## 1 Stand der Technik

Das Gebiet der Energieflexibilität ist aufgrund der vorherrschenden Energiewende ein sehr aktuelles Thema, das für viele Industriezweige beforscht wird [2]. In der Arbeit von Reinhart et al. wird Energieflexibilität als die Möglichkeit von Produktionssystemen zur schnellen und kosteneffizienten Anpassung an Änderungen am Energiemarkt beschrieben [3].

In der Literatur werden durch die Inklusion von Energieverbrauchsdaten in die klassischen Simulationsmo-

delle für Produktionsprozessplanung energetische Optimierungen durchgeführt. Mithilfe der zeitlichen Flexibilität einzelner Produktionsschritte kann eine optimierte Anpassung an die Schwankungen des Energieangebots bzw. der Strompreise umgesetzt werden [4, 5].

Auch für Anwendungen aus dem Bereich der Wasserwirtschaft werden aufgrund der Kosten- und Energieintensität Möglichkeiten untersucht, um die Energieeffizienz der Prozessschritte zu steigern und Kosten zu senken. Zur Modellierung solcher Pumpsysteme werden in der Regel mathematische Modelle gebildet, die den mathematischen Zusammenhang zwischen Zu- und Abfluss von Wasser in einem Tank oder mehreren Tankkaskaden beschreiben. In der Literatur werden weitestgehend die Wasserversorgung oder die städtische Entwässerung betrachtet [6, 7]. Die Betrachtung von Entwässerungssystemen in küstennahen Regionen spielt kaum eine Rolle. Aber insbesondere für die Hinterlandentwässerung ergeben sich durch die Schwankungen am Strommarkt neuartige Methoden zur Entwässerungsoptimierung. Der klassischen Methode des Pumpens bei niedrigen Tideständen stehen nun Möglichkeiten für komplexere Pumpplanungsprozesse gegenüber, die die schwankenden Strompreisstrukturen berücksichtigen [8, 9].

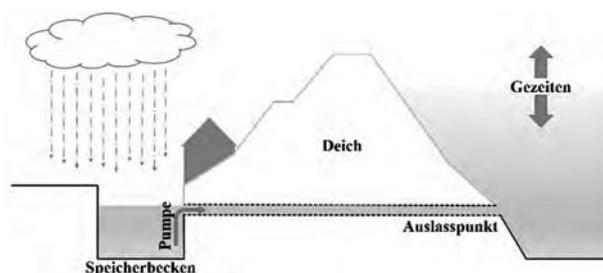
## 2 Szenariobeschreibung

Der Unterhaltungsverband Kehdingen ist einer von 114 Verbänden in Niedersachsen und betreibt auf einer Fläche von rund 27.000 ha entlang der Elbe und im direkten Umland rund 160 Schöpfwerke mit insgesamt 400 Pumpen. Die Hauptaufgabe des Unterhaltungsverbandes besteht darin, das aufkommende Wasser aus dem natürlichen Niederschlagsgebiet abzuleiten. Aufgrund der niedrigen Höhenlagen (unter Normalhöhennull (NHN)) des Gebiets ist eine natürliche Vorflut nur selten möglich und das gesammelte Wasser wird bei Bedarf unter hohem Energieeinsatz mithilfe von Pumpen aus dem Unterhaltungsgebiet in die umliegenden Tidengewässer gehoben. In der Vergangenheit ist dabei für den Unterhaltungsverband und dessen Mitglieder ein jährlicher Energieverbrauch von circa 2,5 – 3 Mio. kWh entstanden. In 2021 bedeutete dies einen Ausstoß von durchschnittlich ca. 1300 t CO<sub>2</sub>-Äquivalenten.

In einem klassischen Entwässerungssystem wird zwischen verschiedenen Arten von Schöpfwerken unterschieden, die je nach Ebene im System unterschiedliche Aufgaben zu bewältigen haben. Die sogenannten Polderschöpfwerke sammeln das abzuführende Wasser direkt

im Binnenland und leiten es über Pumpen und Rohrsysteme in Speicherbecken auf höhere Höhenniveaus ab. Auf diesen höheren Ebenen können entweder Stufenschöpfwerke das Wasser erneut auf ein höheres Niveau heben oder es wird unmittelbar von Mündungsschöpfwerken in angrenzende Tidengewässer abgeleitet. Die Anzahl der spezifischen Schöpfwerke variiert und ist den jeweiligen Bedingungen angepasst. Dabei spielen Faktoren wie die Gebietsgröße und die Gesamtförderhöhe eine Rolle. Generell kann davon ausgegangen werden, dass verhältnismäßig viele Polderschöpfwerke im Binnenland verteilt sind, einige Stufenschöpfwerke bei gesondertem Bedarf eingesetzt werden und eine relativ geringe Anzahl an Mündungsschöpfwerken an den Deichkanten zur Verfügung stehen. Die Leistung der Mündungsschöpfwerke ist dementsprechend um ein Vielfaches höher, als die der Schöpfwerke auf unterer Ebene, um das gesamte abzuführende Niederschlagswasser aus dem Gebiet verarbeiten zu können.

Einige Schöpfwerke besitzen feste Überlaufkanten. Dies bedeutet, dass das abzuleitende Wasser zu jedem Zeitpunkt über ein Rohrsystem auf ein Niveau gehoben wird, das über dem Wasserspiegel des nächsten Speicherbeckens gelegen ist. Von dort aus kann es in das nachgelagerte Sammelbecken fließen. Die momentane Förderhöhe ist daher ausschließlich vom Pegelstand des vorgelegerten Sammelbeckens abhängig.



**Abbildung 1:** Schematische Darstellung eines Schöpfwerks zum Abpumpen von Niederschlagswasser unter dem Deich in Tidengewässer.

Insbesondere bei Mündungsschöpfwerken findet ein anderes Konzept Anwendung, da hier das Wasser unter dem Deich in ein Tidengewässer gepumpt wird. Dessen Pegelstand ist in der Regel höher als der Auslasspunkt des Pumpendurchlaufs (siehe Abbildung 1). Daher ergibt sich für dieses Konzept die momentane Förderhöhe durch die Differenz des Außen- und Innenpegelstandes

der vor- und nachgelagerten Gewässer. Da die Pegelstände von Tidengewässern aufgrund der Gezeiten zyklisch um mehrere Meter variieren können, ergeben sich Zeitfenster mit relativ hohen bzw. geringen Förderhöhen.

Um spezifische Merkmale des Systems bei schwankenden Pegelständen zu analysieren, werden Messdaten bei unterschiedlichen, systemrelevanten Parameterzuständen erhoben. Dafür wird die Motordrehzahl der Pumpe durch eine zeitdiskrete sinusförmige Anpassung zwischen 65 % und 100 % der händlerspezifischen Nenndrehzahl im Arbeitsbereich unter Realbedingungen variiert. Mithilfe von implementierten Messsystemen (Durchfluss-, Leistungs- und Pegelstandmessungen) können systembezogene Datenpunkte aufgenommen werden.

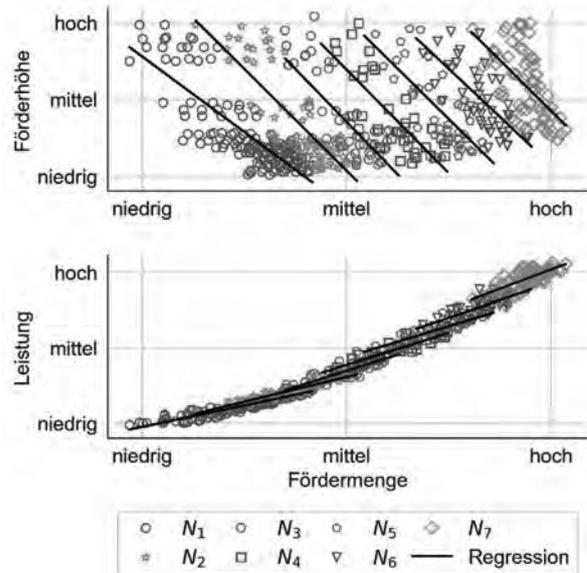
### 3 Mathematisches Modell

In diesem Abschnitt wird die mathematische Modellierung eines einzelnen Mündungsschöpfwerks mit einer drehzahlregelbaren Pumpe dargestellt und basierend auf diesem Modell werden unterschiedliche Szenarien hinsichtlich einer mathematisch optimierten Abschöpfstrategie miteinander verglichen.

#### 3.1 Beschreibung des Modells

Die Grundlage für die nachfolgende mathematische Modellierung ist die gemischt-ganzzahlige nichtlineare Optimierung (MINLP) aus der Arbeit von Fecarotta et al. [6]. Aufgrund der Gegebenheit, dass ein Entwässerungssystem in der Regel aus einem Verbund mehrstufiger Pumpen und Speicherbecken besteht, die eine systematische Interdependenz aufweisen, wird das Problem durch Linearisierungs- und Diskretisierungstechniken in ein gemischt-ganzzahliges lineares Optimierungsproblem (MILP) überführt. Die Reduktion der Komplexität soll unter Realbedingungen eine Berechnung von mehrstufigen Pumpplänen in vertretbarer Zeit gewährleisten.

Zu Beginn ergeben sich durch die Klassierung der gemessenen, systemspezifischen Datenpunkte der Pumpe  $N = 7$  äquidistante Drehzahlbereiche ( $N_1$ : 65 %– 70 %,  $N_2$ : 70 %– 75 %, ...). Durch die Anwendung von linearen Regressionsverfahren lassen sich im weiteren Verlauf diskretisierte, lineare Systemkennlinien ableiten. Diese Kennlinien definieren den Zusammenhang zwischen Förderhöhe bzw. Leistung und Durchfluss (Fördermenge) der Pumpe und geben einen Einblick in die generelle Funktionsweise einer entsprechenden Schöpfwerkspumpe im Arbeitsbereich (siehe Abbildung 2).



**Abbildung 2:** Gemessene Datenpunkte mit resultierenden Systemkennlinien einer installierten Schöpfwerkspumpe im Drehzahlbereich ( $N_1 - N_7$ ) von 65 % - 100 % der vorgegebenen Nenndrehzahl.

Es lassen sich vier verschiedene Vektoren zur Abbildung der Kennlinien erzeugen. Dabei bilden die Vektoren  $M_H$  bzw.  $M_P$  die Steigung der Regressionsgraden für die Förderhöhen- bzw. Leistungskennlinie ab.  $B_H$  bzw.  $B_P$  bilden den Ordinatenabschnitt der Kennlinien ab.

Die Grundlage der mathematischen Optimierung bildet in dieser Arbeit eine multikriterielle Minimierung der Summe aus den akkumulierten Energieverbräuchen  $P_t$  und den Energiekosten  $K_t$  mit einem Gewichtungsfaktor  $w$ :

$$\min ZF = \sum_{t \in T} (w \cdot P_t + (1 - w) \cdot s \cdot K_t) \cdot dt \quad (1)$$

Hierbei gibt  $T$  die Anzahl und  $dt$  die Dauer der Planungsperioden an. Die Energiekosten setzen sich aus den Energieverbräuchen und den Energiepreisen  $p_t$  der einzelnen Perioden zusammen ( $K_t = P_t \cdot p_t$ ). Durch einen Skalierungsfaktor  $s$  werden die Kosten an den Wertebereich der Energieverbräuche angepasst. Dabei nimmt  $s$  den Durchschnittswert von  $p_t$  an. Mithilfe des Gewichtungsfaktors  $w$  kann die Präferenz für die unterschiedlichen Zielkriterien beliebig variiert werden.

Die Entscheidungsvariablen des Optimierungsmodells werden durch die binären Variablen  $x_{t,n} \in \{0,1\}$  mit  $n \in \{0,1, \dots, N\}$  repräsentiert, wobei  $x_{t,0} = 1$  die Deaktivierung der Pumpe in Periode  $t$  darstellt. Für alle  $n \in$

$\{1, \dots, N\}$  repräsentiert  $x_{t,n}$  die Wahl einer der  $N$  spezifischen Drehzahlbereiche.

Das System unterliegt des Weiteren verschiedenen Restriktionen, die den Rahmen der Optimierungsmöglichkeiten durch eine Beschränkung des Lösungsraums vorgeben. Diese Restriktionen werden im folgenden Verlauf formal dargestellt.

Durch eine diskretisierte Kontinuitätsgleichung:

$$H_{in_t} = H_{in_{t-1}} + \frac{dQ_{in_t} - dQ_t}{S} dt \quad (2)$$

wird beispielsweise sichergestellt, dass die Änderung des Innenpegelstandes  $H_{in_t}$  im Speicher in konsekutiven Zeitfenstern konsistent zum durchschnittlichen Ab- ( $dQ_t = \frac{Q_t + Q_{t-1}}{2}$ ) und Zufluss ( $dQ_{in_t} = \frac{Q_{in_t} + Q_{in_{t-1}}}{2}$ ) des Tanks ist. Dieser setzt sich aus den Ab-  $Q$  bzw. Zuflüssen  $Q_{in}$  zweier konsekutiver Zeitfenstern zusammen.

Da der Pegel des Tanks sowohl einen Minimal- ( $c_{lb}$ ) als auch einen Maximalpegelstand ( $c_{ub}$ ) nicht unter- bzw. überschreiten darf, werden Kapazitätsrestriktionen definiert, die einen Leerlauf- bzw. Überlaufschutz darstellen:

$$c_{lb} \leq H_{in_t} \leq c_{ub} \quad (3)$$

Mithilfe folgender Gleichung wird die Förderhöhe  $H_{f_t}$  in jedem Zeitfenster in Abhängigkeit vom Innen- und Außenpegelstand  $H_{out_t}$  definiert:

$$H_{f_t} = H_{out_t} - H_{in_t} \quad (4)$$

Die nachfolgenden vier Ungleichungen ermöglichen eine diskrete Auswahl aus den  $N$  zur Verfügung stehenden Regressionsgraden der unterschiedlichen Drehzahlbereiche. Dabei wird durch die Wahl ausreichend hoher Werte für  $H_{f,max}$  und  $P_{max}$  sichergestellt, dass die Restriktionen für nicht-gewählte Drehzahlbereiche  $x_{t,n} \forall n \in N \setminus \{0\}$  aufgeweicht werden, um den verfügbaren Lösungsraum nicht zu beschränken.

$$M_{H_n} \cdot Q_t + B_{H_n} - H_{f,max} \cdot (1 - x_{t,n}) \leq H_{f_t} \quad (5)$$

$$M_{H_n} \cdot Q_t + B_{H_n} + H_{f,max} \cdot (1 - x_{t,n}) \geq H_{f_t} \quad (6)$$

$$M_{P_n} \cdot Q_t + B_{P_n} - P_{max} \cdot (1 - x_{t,n}) \leq P_t \quad (7)$$

$$M_{P_n} \cdot Q_t + B_{P_n} + P_{max} \cdot (1 - x_{t,n}) \geq P_t \quad (8)$$

Die beiden nachstehenden formalen Zusammenhänge gewährleisten, dass der generierbare Durchfluss bzw. die benötigte Leistung gleich Null gesetzt werden, sofern die Pumpe deaktiviert ist. Durch die Wahl ausreichend hoher Werte für  $Q_{max}$  und  $P_{max}$  werden die Restriktionen für

eine aktive Pumpe soweit aufgeweicht, dass diese im Arbeitsbereich nicht beschränkt wird.

$$(1 - x_{t,0}) \cdot Q_{max} \geq Q_t \quad (9)$$

$$(1 - x_{t,0}) \cdot P_{max} \geq P_t \quad (10)$$

Es wird durch folgende Gleichung außerdem definiert, dass die Pumpe innerhalb eines Zeitfensters entweder deaktiviert ist ( $n = 0$ ) oder exakt ein einziger Drehzahlbereich der Pumpe aktiviert ist ( $n \in N \setminus \{0\}$ ).

$$\sum_{n=0}^N x_{t,n} = 1 \quad (11)$$

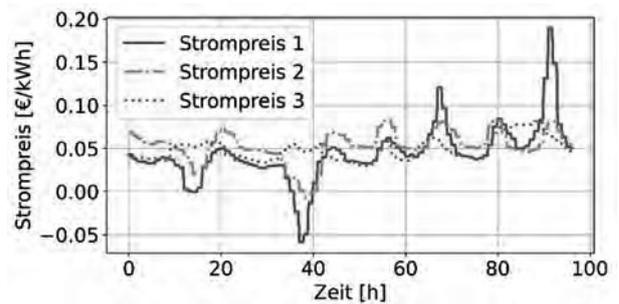
Aus Gründen der Plausibilität gelten sowohl für den Durchfluss als auch für die Leistung die Nichtnegativitätsbedingungen:

$$Q_t \geq 0 \quad (12)$$

$$P_t \geq 0 \quad (13)$$

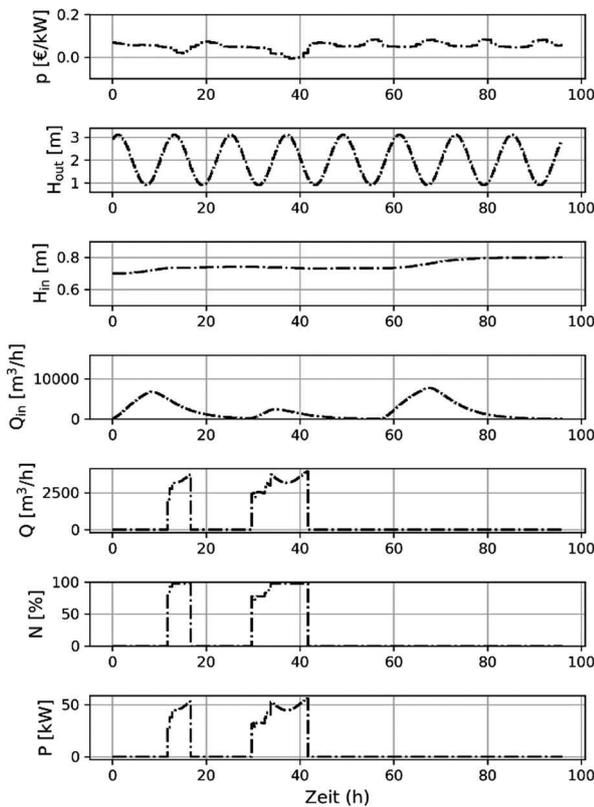
### 3.2 Durchführung der Experimente

Bei der Durchführung der nachfolgenden Experimente werden unterschiedliche Datenwerte in das vorgestellte MILP eingepflegt, um mithilfe des Gurobi Solvers einen optimalen Pumpplan für vier aufeinanderfolgende Tage (bzw. 384 Zeitschritte mit einem Zeitfenster von 15 Minuten) zu berechnen. Neben der Implementierung von drei unterschiedlichen Strompreisverläufen (Strompreis 1, 2 und 3) mit unterschiedlich starken Strompreisschwankungen (siehe Abbildung 3), wird durch eine Sinusfunktion mit 12-stündigem Zyklus das Gezeitenverhalten eines Tidengewässers simuliert.



**Abbildung 3:** Viertägige Verläufe mit viertelstündiger Auflösung unterschiedlicher, vergangener Großhandelspreise für Strom in Deutschland/Luxemburg auf Basis der Daten der Bundesnetzagentur [11].

Der Anfangsfüllstand des Sammelbeckens wird auf einen festen Wert bei halber Höhe des abpumpbaren Pegelstandes gesetzt. Des Weiteren wird mithilfe einer Linienspeicherkaskade zur Umwandlung von Niederschlags- in Abflussdaten der Zufluss zum Sammelbecken der Pumpe modelliert [10]. Hierbei wird ein Zuflussmuster aus drei zufällig aufeinanderfolgenden, unterschiedlich intensiven Zuflüssen gewählt. Außerdem wird durch die Skalierung des Zuflussmusters mit dem Faktor 1,5 ein zweiter, stärkerer Zufluss modelliert, um unterschiedliche Auswirkungen zu untersuchen.



**Abbildung 4:** Exemplarisches, optimiertes Pumpszenario für vier Tage bei schwankendem Strompreis (Strompreis 2) und schwachem Zuflussmuster für  $w = 0$ .

In Abbildung 4 ist eine exemplarische Darstellung eines einzelnen Szenarios mit optimierter Pumpplanung über einen Zeithorizont von vier Tage abgebildet. In der ersten Zeile des Diagramms werden die zugrundeliegende Strompreisverläufe dargestellt. Die zweite Zeile visualisiert das tideähnliche Verhalten des Außenpegels. Danach folgt der durch eine Optimierung resultierende Verlauf des Innenpegelstands des Sammelbeckens, der

zum Startzeitpunkt den Mittelwert der Kapazitätsgrenzen (untere Grenze: 0,6 m und obere Grenze: 0,8 m) annimmt. In Zeile 4 wird das angewendete Zuflussmuster dargestellt und in den nachfolgenden drei Zeilen werden die optimalen Verläufe der Pumpenkenndaten (Durchfluss, Drehzahl und Leistung) gezeigt.

Aus den genannten Rahmenbedingungen ergeben sich, wie in Tabelle 1 dargestellt, sechs unterschiedliche Untersuchungsszenarien, die jeweils einen der drei Strompreisverläufe und eine der beiden Zuflussstärken verwenden. Innerhalb dieser Szenarien wird der Gewichtungsfaktor  $w$  in äquidistanten Schritten zwischen 0 und 1 variiert, wodurch entweder ausschließlich die Energiekosten bzw. der Energieverbrauch oder eine gewichtete Kombination aus beiden Größen betrachtet und optimiert wird.

	Schwacher Zufluss	Starker Zufluss
<b>Strompreis 1</b>	Relative starke Preisschwankungen bei mäßigem Niederschlag	Relative starke Preisschwankungen bei stärkerem Niederschlag
<b>Strompreis 2</b>	Mittlere Preisschwankungen bei mäßigem Niederschlag	Mittlere Preisschwankungen bei stärkerem Niederschlag
<b>Strompreis 3</b>	Relative schwache Preisschwankungen bei mäßigem Niederschlag	Relative schwache Preisschwankungen bei stärkerem Niederschlag

**Tabelle 1:** Darstellung der Untersuchungsszenarien für unterschiedliche Strompreisverläufe und verschiedenen Zuflussstärken.

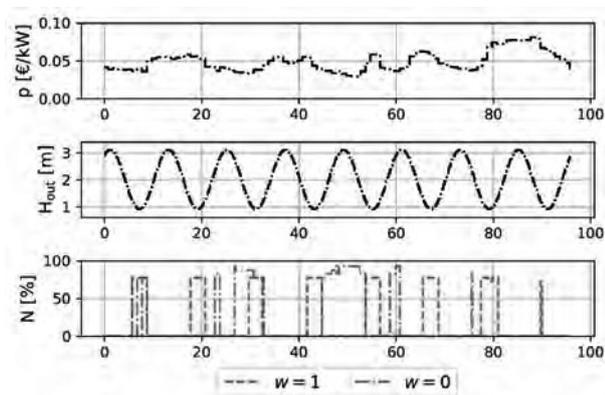
Um eine Vergleichbarkeit zwischen den Ergebnissen unter Anwendung der verschiedenen Preisverläufe herzustellen, wird eine Simulation durchgeführt, bei der die mathematische Optimierung jeweils auf 38 zyklische Verschiebungen des Zuflussmusters angewendet wird. Hierbei wird die Datenreihe vor jedem Optimierungsdurchlauf um zehn Zeitschritte verschoben. Durch diese Verschiebung wird gewährleistet, dass unterschiedliche, zufällige Parameterzustände zwischen aktuellem Preis und Zufluss eintreten.

Ausgewertet werden die resultierenden durchschnittlichen Energieverbräuche und die durchschnittlichen Energiekosten. Die Basis für die sechs Szenarien aus Tabelle 1 werden jeweils durch die Optimierungsergebnisse

bestimmt, die sich bei Annahme eines konstanten Durchschnittspreises für den zugrundeliegenden Strompreisverlauf ergeben. Diese Annahme steht stellvertretend für eine Optimierung bei tarifgebundenem Strombezug.

## 4 Ergebnisse der Studie

Die Ergebnisse in Abbildung 5 machen deutlich, dass sich das Pumpmuster durch die Variation des Gewichtungsfaktors verändert. Der Basisansatz zur Minimierung des Energieverbrauchs mit  $w = 1$  führt dazu, dass die notwendigen Pumpaktivitäten zu relativ niedrigen Tidenphasen geplant werden. Im Umkehrschluss wird ebenfalls verdeutlicht, dass durch die Minimierung der Energiekosten mit  $w = 0$  eine Anpassung der Pumpaktivitäten an die Niedrigpreisphasen stattfindet.

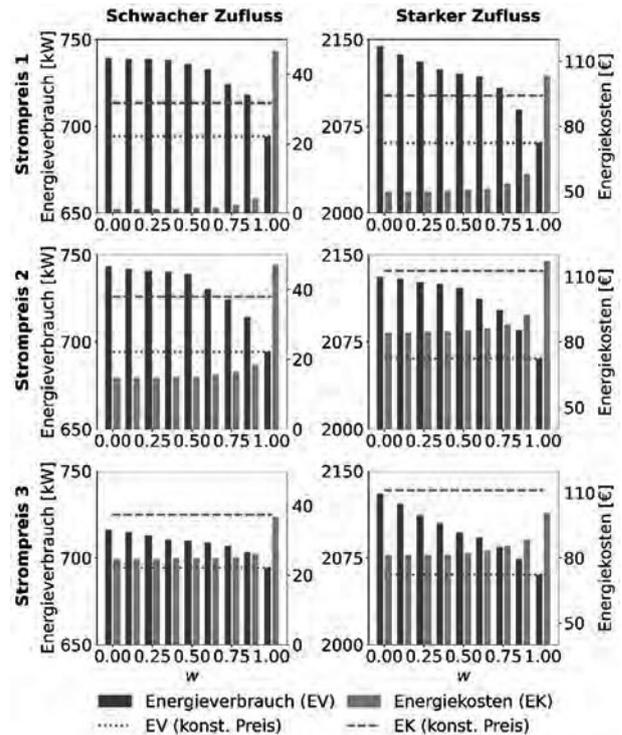


**Abbildung 5:** Vergleich für Energieverbrauchs- und Kostenoptimierung anhand von exemplarisch optimierten Pumpszenarien für vier Tage bei zufälligem Zuflussmuster (Strompreis 3).

Anhand von Abbildung 6 kann dies für eine detailliertere Betrachtung dargestellt werden. Es wird ersichtlich, dass für alle Szenarien aus Tabelle 1, durch steigende Werte für  $w$ , im Durchschnitt geringere Energieverbräuche bzw. höhere Energiekosten resultieren. Insbesondere die Kosten weisen in diesem Zusammenhang ein relativ starkes exponentielles Wachstum auf. Von oben nach unten betrachtet, nimmt dieser Effekt mit sinkender Schwankungsbreite im Strompreisverlauf rasant ab. Mit der Betrachtung von links nach rechts schwächt auch die Zunahme der Zuflussstärke diesen Effekt ab.

Unter Verwendung konstanter Durchschnittspreise ergeben sich für  $w > 0$  konstante Durchschnittswerte für den Energieverbrauch bzw. die Energiekosten, da das

Kriterium der Kostenoptimierung aufgrund der fehlenden Strompreisschwankung keine Vorteile birgt. Der resultierende Energieverbrauch entspricht demnach stets dem Energieverbrauch für  $w = 1$ . Für die Energiekosten zeigt sich ein anderes Bild. Die Kosten bei konstanten Strompreisstrukturen weisen für  $w < 1$  im dargestellten Bereich eine hohe negative Diskrepanz auf, da keine Niedrigpreisphasen für Pumpaktivitäten genutzt werden können.



**Abbildung 6:** Simulationsergebnisse zur mathematischen Optimierung der Zielkriterien (Energieverbrauch und Energiekosten) unter Variation des Gewichtungsfaktors  $w$  für unterschiedliche Zuflussszenarien und Strompreisverläufe.

## 5 Zusammenfassung und Ausblick

Im Kontext dieses Beitrags wurde ein vereinfachtes Mathematisches Modell (MILP) dargestellt, das einige der wichtigen Zusammenhänge eines Entwässerungssystems durch Linearisierungs- und Diskretisierungstechniken komplexreduziert beschreibt. Das MILP wurde genutzt, um verschiedene Szenarien im Hinblick auf Energieverbräuche und -kosten zu vergleichen. Es wurde dargelegt,

in welchem Maße die Kosten unter bestimmten Bedingungen reduziert werden können, wenn eine Betrachtung der Strompreisschwankungen bei der Planung der Pumpaktivitäten berücksichtigt wird. Durch Simulation wurden die Ergebnisse anhand einer größeren Bandbreite von äußeren Bedingungen getestet. Die Ergebnisse zeigen auf, dass bereits durch eine geringe Priorisierung der Energiekosten bei der Optimierung große kostensenkende Effekte entstehen. Um dieses Forschungsgebiet weiter zu untersuchen, bietet es sich an mehrstufige Entwässerungssysteme zu analysieren. Außerdem können durch die Betrachtung von stochastischen Schwankungen für Strompreise und Niederschlag bzw. Zuflüssen weitere wichtige Erkenntnisse erlangt werden. Z.B. kann hierdurch die Abweichung der Lösungsgüte vom berechneten Optimum untersucht werden.

### Danksagung

Dieser Beitrag wird durch das Verbundvorhaben „SCHOEPFWERK40 – Reduktion des Energieverbrauchs von Schöpfwerken durch intelligente Pumpsteuerung und -regelung“ unter der Nummer 03EN2008 A-C im Rahmen des 7. Energieforschungsprogramm der Bundesregierung durch das Bundesministerium für Wirtschaft und Klimaschutz (BMWK) gefördert.

Des Weiteren bedanken wir uns für die Unterstützung bei unseren Projektpartnern, Herr Nicolai, Geschäftsführer und Technische Leitung des Unterhaltungsverbands Kehdingen und Herr Flathmann, Leitung Technisches Büro der Vetterkind GmbH.

### Literatur

- [1] Bachmann, A.; Bank, L.; Bark, C.; Bauer, D.; Blöchl, B.; Brugger, M.; Buhl, H.U.; Dietz, B.; Donnelly, J.; Friedl, T.; Halbrügge, S.; Hauck, H.; Heil, J.; Hieronymus, A.; Hinck, T.; Ilieva-König, S.; Johnzén, C.; Koch, C.; Köberlein, J.; Köse, E.; Lochner, S.; Lindner, M.; Mayer, T.; Mitsos, A.; Roth, S.; Sauer, A.; Scheil, C.; Schilp, J.; Schimmelpfennig, J.; Schulz, J.; Schulze, J.; Sossenheimer, J.; Strobel, N.; Tristan, A.; Vernim, S.; Wagner, J.; Wagon, F.; Weibelzahl, M.; Weigold, M.; Weissflog, J.; Wenninger, S.; Wöhl, M.; Zacharias, J.; Zäh, M.F.: *Energieflexibel in die Zukunft - Wie Fabriken zum Gelingen der Energiewende beitragen können*. VDI Verlag 2021.
- [2] Sauer, A.; Abele, E.; Buhl, H.U.: *Energieflexibilität in der deutschen Industrie: Ergebnisse aus dem Kopernikus-Projekt-Synchronisierte und energieadaptive Produktionstechnik zur flexiblen Ausrichtung von Industrieprozessen auf eine fluktuierende Energieversorgung (SynErgie)*. Fraunhofer Verlag 2019.
- [3] Reinhart, G.; Reinhardt, S.; Graßl, M.: *Energieflexible Produktionssysteme. Einführungen zur Bewertung der Energieeffizienz von Produktionssystemen*. wt Werkstattstechnik Online 102 2012, S. 622–628.
- [4] Schulz, J.; Lütkes, F.; Szabo, A.; Zaeh, M.F.: *Energy-orientated material flow simulation with stochastic optimisation for peak load management*. Procedia CIRP 107, 2022, S. 399–404.
- [5] Uhlig, B.; Kloock, M.; Mennenga, M.; Herrmann, C.: *Simulation-based energy flexibility analysis of manufacturing process chains: heat treatment in a foundry*. Procedia CIRP 107, 2022, S. 1379–1384.
- [6] Fecarotta, O.; Carravetta, A.; Morani, M.C.; Padulano, R.: *Optimal Pump Scheduling for Urban Drainage under Variable Flow Conditions*. Resources, 2018, 7(4): S. 73.
- [7] Luna, T.; Ribau, J.; Figueiredo, D.; Alves, R.: *Improving energy efficiency in water supply systems with pump scheduling optimization*. Journal of cleaner production, 2019, S. 342–356.
- [8] Fichter, C.; Müller, M.: *Intelligente Steuerung von Schöpfwerken zur Reduktion des Energieeinsatzes und der CO<sub>2</sub>-Emissionen im Projekt Demand Side Management Sielentwässerung-DSMS*. Zeitschrift für Energiewirtschaft 2021, 45(4): S. 287–294.
- [9] Heger, J.; Voß, T.: *Simulationsbasierte Optimierung zur Energieersparnis und Verbrauchsflexibilisierung in der Hinterlandentwässerung*. 18. ASIM-Fachtagung Simulation in Produktion und Logistik, Chemnitz, 2019.
- [10] Eckhardt, K.: *Hydrologische Modellierung - Ein Einstieg mithilfe von Excel*. Heidelberg: Springer Spektrum 2014.
- [11] Bundesnetzagentur, 2017: *SMARD: Strommarktdaten*. Online verfügbar unter <https://www.smard.de/home/downloadcenter/download-marktdaten/>, zuletzt geprüft am 06.01.2023.



# Hauptvorträge

## Invited Lectures

Saisonaler Energiespeicher auf Basis von Kalk – Experimentelle Entwicklung und Systemsimulation (Abstract)

*Marc Linder, DLR Institut für Technische Thermodynamik, Stuttgart;*

*Michael Böhm, Institut für Systemdynamik, Universität Stuttgart*

Digital Marbling: Simulation of a Traditional Bookbinder's Craft (Abstract)

*Graham Horton, Institut für Simulation und Grafik,*

*Otto-von-Guericke-Universität Magdeburg*

Langbeiträge (L) sind zu finden im **ARGESIM Report 21**

**Proceedings Langbeiträge ASIM Workshop 2023 STS/GMMS/EDU**

ISBN ebook 978-3-903347-61-8, DOI 10.11128/arep.21, ASIM Mitteilung AM 185

Kurzbeiträge (K) und Abstract-Beiträge (A) sind zu finden im **ARGESIM Report 22**

**Kurzbeiträge & Abstract-Beiträge ASIM Workshop 2023 STS/GMMS/EDU**

ISBN ebook 978-3-903347-62-5, DOI 10.11128/arep.22, ASIM Mitteilung AM 186

Abstracts der Hauptvorträge sind im **ARGESIM Report 21** und im **ARGESIM Report 22** zu finden.



# Saisonaler Energiespeicher auf Basis von Kalk

## – experimentelle Entwicklung und Systemsimulation

PD Dr.-Ing. Marc Linder <sup>1</sup>, Dr.-Ing. Michael Böhm <sup>2</sup>

<sup>1</sup> Deutsches Zentrum für Luft- und Raumfahrt (DLR e.V.)

<sup>2</sup>Institut für Systemdynamik, Universität Stuttgart

### Hintergrund

Das Heizen unserer Innenräume ist für ein Viertel des Endenergiebedarfs in Deutschland verantwortlich. Gleichzeitig beträgt der Anteil an Erneuerbaren Energien zur Deckung dieses Bedarfs derzeit ca. 15 % und liegt damit nur minimal über dem Wert von vor 10 Jahren (ca. 13 %). Die ganzjährige Versorgung mit erneuerbarer thermischer Energie stellt daher eine zentrale Herausforderung eines vollständig de-karbonisierten Energiesystems dar.

Die wesentlichen Gründe hierfür sind die saisonalen Zyklen, die akkumulierte Energiemenge sowie die sehr eingeschränkte Möglichkeit eines Transports der thermischen Energie. Dem gegenüber stehen Erfolge und ambitionierte Ausbauziele unterschiedlicher Technologie zur Gewinnung erneuerbarer elektrischer Energie die sich grundsätzlich über sehr lange Distanzen ergänzen und ausgleichen können

### 1 Technologieentwicklung

Am Deutschen Zentrum für Luft- und Raumfahrt wurde in den letzten Jahren ein Energiespeicher entwickelt der diese Problemstellung adressiert und große Mengen an Energie verlustfrei in günstigen Behältern dezentral speichern kann. Die Grundlage hierfür bildet die bekannte reversible Reaktion von gebranntem Kalk und Wasser, sodass die Speichermaterialien denkbar günstig sind und v.a. ressourcenschonend bereitgestellt und später auch entsorgt werden können. Das Speichermaterial wird unter Aufnahme von elektrischer Energie beladen. Dabei wird Wasserdampf freigesetzt und das zurückbleibende Pulver in Behältern bei Raumtemperatur zwischengelagert.

Die bedarfsorientierte Entladung findet durch die Rückreaktion mit flüssigem Wasser statt, wobei den Behältern nur so viel Material entnommen wird wie zur Deckung des aktuellen Bedarfs notwendig ist. Die erreichbaren Temperaturen betragen dabei ca. 90°C und sind somit auch für die Heizungssysteme in Bestandsgebäuden ausreichend hoch.

### 2 Systemsimulation und Integration

Aktuelles Ziel ist die Entwicklung eines Prototyps der erstmals außerhalb des Labors am Demonstrator-Hochhaus des SFB 1244 an der Universität Stuttgart installiert und anschließend unter realen Umgebungsbedingungen getestet wird. Ergänzend dazu werden zusammen an der Universität Stuttgart prädiktive Betriebs- und Regelungsansätze entwickelt, sodass diese Speichertechnologie optimal als Bindeglied zwischen dezentralem thermischem Bedarf, dezentraler erneuerbarer Stromerzeugung und dem Stromnetz agieren kann.

Der Speicher wird zu diesem Zweck zunächst rein bilanziell betrachtet, um in Kombination mit weiteren Komponenten als Teil eines thermischen Gebäudenetzwerks eingesetzt zu werden. Auf Basis dieses Netzwerkmodells wird zunächst eine prädiktive saisonale Betriebsplanung gemacht, um festzulegen, welche Komponente zu welchem Zeitpunkt die nötige Heizleistung bereitstellen soll. Damit der numerische Aufwand bei diesem langen Horizont vertretbar bleibt, wird eine hohe Abtastzeit von 1 Tag gewählt. Die so erhaltenen Werte dienen als Referenz für eine unterlagerte Optimierung mit kürzerem Horizont – typischerweise einige Tage – und kürzerer Abtastzeit von 1h. Diese Planung berücksichtigt dann die kurzfristigen

Einflüsse wie Wetter und konkrete Nutzeranforderungen.

Damit der Speicher die durch diese Planung vorgegebenen Heizleistungen robust und zuverlässig bereitstellen kann, wird am Institut für Systemdynamik ein detailliertes dynamisches Modell des Speichers entwickelt und dessen Parameter identifiziert. Im Vortrag wird ausführlich auf die einzelnen Aspekte dieses Modells eingegangen. Das Modell dient später als Basis für den Entwurf einer Regelung, die zunächst in der Simulation und später am erwähnten Prototyp auch zusammen mit der optimierungsbasierten Betriebsplanung getestet wird.

# Digital Marbling – Simulation of a Traditional Bookbinder's Craft

Graham Horton

Computer Science Department, University of Magdeburg, 39016 Magdeburg, Germany. [graham.horton@ovgu.de](mailto:graham.horton@ovgu.de)

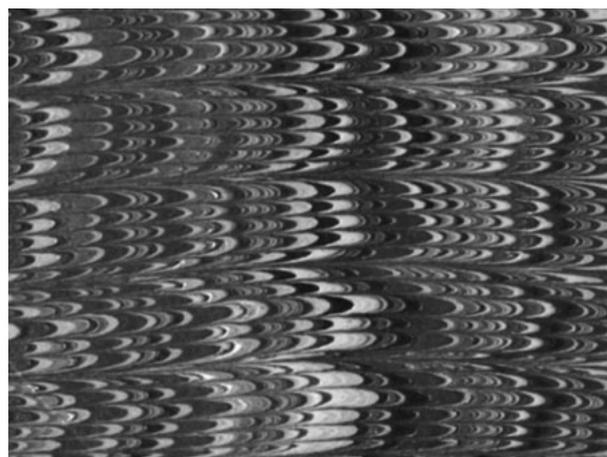
Up until the era of mass production, books were bound manually, often using elaborately decorated paper for the endpapers and covers. In the technique known as marbling, pigments were distributed on the surface of a water bath, and a pattern was induced in them with a needle or a comb. This pattern was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



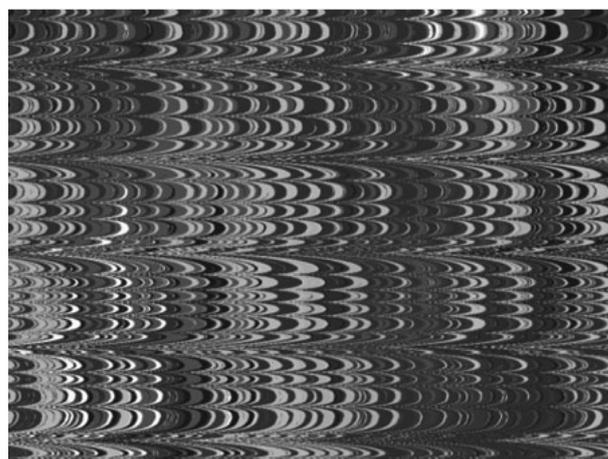
**Figure 1:** Making spirals in the floating pigments using a needle. (Photo: Barbara Kelnhofer)

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

The focus of the presentation will be to show various patterns generated by the simulator. These will include comparisons with the endpapers of various 19th and 20th century books that show how accurate even the simple mathematical model can be. Finally, original, digitally marbled patterns will be presented which - although physically feasible - have yet to be created by traditional craftspeople.



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



**Figure 3:** Simulation of the marbled pattern



## AUTORENINDEX – INDEX OF AUTHORS

### ASIM WORKSHOP 2023 STS/GMMS/EDU – LANGBEITRÄGE

- A. Ahlbrecht 95  
K. Akgün 87  
A. Apostolatos (K-A: 21)  
C. Assmann (K-A: 23) J.  
Bartelt 125  
G. Battenberg (K-A: 23)  
M. Bicher (K-A: 27)  
M. Böhm 153 (K-A: 3)  
F. Borgmann (K-A: 9)  
F. Breitenecker (K-A: 27)  
B. Buchholz 55  
U. Durak 77, 87, 95, 103  
C. Fink 55  
D. Garth (K-A: 23)  
H. Ghezali 103  
B. Glock (K-A: 27)  
M. Göllner 63, 71  
J. Görs (K-A: 29)  
S. Groß (K-A: 21)  
P. Grzona (K-A: 15)  
S. Gupta 87, 95, 103  
I. Hafner (K-A: 27)  
S. Hartmann 77  
M. Hempel 143  
J. Heger 143  
G. Horton 155 (K-A: 5, 29)  
S. Jacobitz 37, 63, 71  
D. Jörss 55  
P. Junglas 3  
N. Kalbe (K-A: 9)  
P. Krenckel 117  
C. Krull 117, (K-A: 25)  
T. Li 63  
M. Linder 153 (K-A: 3)  
X. Liu-Henke 37, 45, 63, 71  
N. Lohse (K-A: 23)  
B. Lukic 95  
N. Marten 19  
M. Münnich (K-A: 15)  
M. Mujica Mota (K-A: 27)  
G. Mušic (K-A: 27)  
N. Popper (K-A: 27)  
T. Pawletta 125, (K-A: 27)  
B. Prell 27  
J. Reiff-Stephan 27  
J. Riemenschneider (K-A: 23)  
M. Ringel 55  
C. Rippinger (K-A: 27)  
M. Rössler (K-A: 27)  
O. Rose 11  
G. Schneckenreither (K-A: 27)  
L. Schneebeili (K-A: 23)  
C. Schwarz (K-A: 31)  
J. Sprockhoff 95  
H. Stein 11  
S. Straßburger 133  
K. Thiele 19  
T. Turmaz (K-A: 9)  
T. Uhlig 11  
C. Urach (K-A: 27)  
T. Vietor 45  
M. Wastian (K-A: 27)  
F. Widdascheck (K-A: 23)  
B. Wörrlein 133  
G. Zauner (K-A: 27)  
M. Zechmeister (K-A: 27)  
J. Zhang 37, 45, 71  
F. Zumpe (K-A: 15)

Langbeiträge (L) sind zu finden im

**ARGESIM Report 21 Proceedings Langbeiträge ASIM Workshop 2023 STS/GMMS/EDU**

ISBN ebook 978-3-903347-61-8, DOI 10.11128/arep.21, ARGESIM Report 21, ASIM Mitteilung 185

Kurzbeiträge (K) und Abstract-Beiträge (A) sind zu finden im

**ARGESIM Report 22 Kurzbeiträge & Abstract-Beiträge ASIM Workshop 2023 STS/GMMS/EDU**

ISBN ebook 978-3-903347-62-5, DOI 10.11128/arep.22, ARGESIM Report 22, ASIM Mitteilung 186

Abstracts der Hauptvorträge sind im **ARGESIM Report 21** und im **ARGESIM Report 22** zu finden



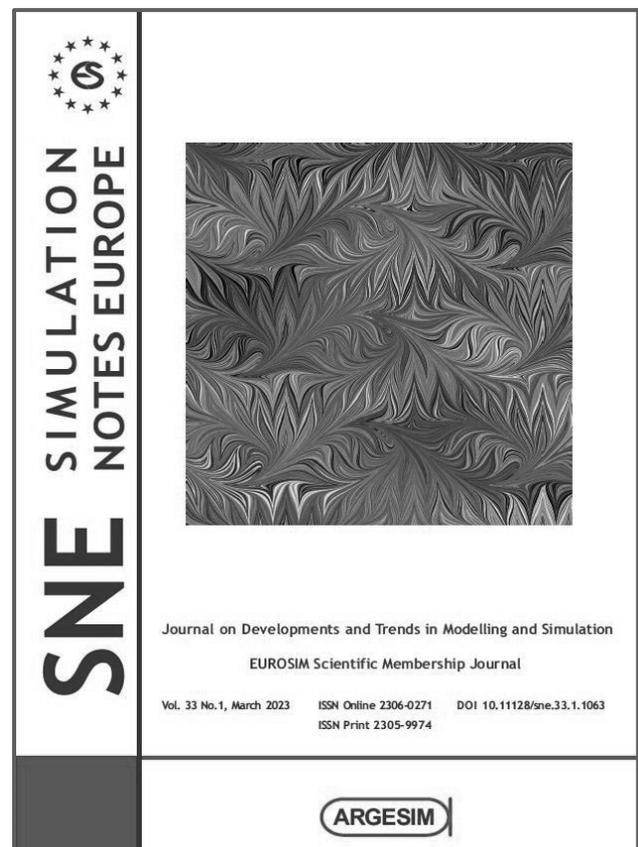
# SNE SIMULATION NOTES EUROPE

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, and the scientific membership journal of **ASIM**, the German simulation society. **SNE** is open for post-conference publications and for special issues organized by **EUROSIM** societies, e.g. **ASIM** thematic special issues or **ASIM** post-conference special issues.

**SNE** is primarily an electronic journal and follows an open access strategy, with free download in basic layout. Members of **EUROSIM** societies, as **ASIM**, **SIMS**, e.g. are entitled to download **SNE** in an elaborate and extended layout. Print **SNE** is available for specific groups of **EUROSIM** societies.



[www.sne-journal.org](http://www.sne-journal.org)

ISBN ebook 978-3-903347-61-8  
ARGESIM Report 21  
[www.argesim.org](http://www.argesim.org)

DOI 10.11128/arep.21  
ASIM Mitteilung 185  
[www.asim-gi.org](http://www.asim-gi.org)