

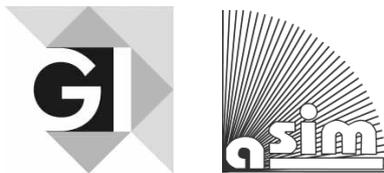
ASIM-Treffen STS/GMMS 2016

Workshop der ASIM/GI Fachgruppen STS und GMMS

10./11. März 2016 in Lippstadt

Tagungsband

Dmitrij Tikhomirov, Heinz-Theo Mammen, Thorsten Pawletta
(Hrsg.)



Arbeitsgemeinschaft Simulation ASIM
in der Gesellschaft für Informatik GI

Fachgruppe Simulation technischer Systeme STS

Fachgruppe Grundlagen und Methoden in
Modellbildung und Simulation GMMS



Hochschule Hamm-Lippstadt

Hochschule Hamm-Lippstadt
Marker Allee 76-78, 59063 Hamm
Tel: +49 2381 8789 217
www.hshl.de

Das Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Funksendung, der Wiedergabe auf photomechanischem oder ähnlichem Weg und der Speicherung in Datenverarbeitungsanlagen bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten.

ARGESIM Report 51

ASIM Mitteilung AM 158

ISBN print 978-3-901608-48-3

ISBN ebook 978-3-901608-88-9, DOI 10.11128/arep.51

©ARGESIM Verlag Wien, Hochschule Hamm-Lippstadt 2016

ARGE Simulation News (ARGESIM)
c/o F. Breitenecker, Div. Simulation, Vienna Univ. of Technology
Wiedner Hauptstrasse 8-10, A-1040 Vienna, Austria
Tel.: +43-1-58801-11452, Fax: +43-1-58801-42098
E-Mail: info@argesim.org; www.argesim.org

Vorwort

Auf Einladung der ASIM/GI Fachgruppen STS und GMMS und der Hochschule Hamm-Lippstadt findet am 10. und 11. März 2016 der Workshop "Simulation technischer Systeme inklusive der Grundlagen und Methoden in Modellbildung und Simulation" auf dem Campus Lippstadt statt.

Der Workshop der ASIM/GI Fachgruppen hat bereits eine lange Tradition. Die Workshops finden jährlich an verschiedenen Standorten statt. Die Veranstaltungen dieser Reihe tragen dazu bei, die Diskussion, den Informations- und Erfahrungsaustausch zwischen Fachleuten auf den Gebieten der Simulation technischer Systeme sowie der Grundlagen und Methoden der Modellbildung und Simulation zu fördern.

In zwei Tutorien, drei Plenarvorträgen und insgesamt mehr als vierzig Fachvorträgen aus Industrie und Forschung in vier parallelen Sessions werden aktuelle Themen aus dem Bereich der Simulation behandelt. Die angenommenen Vorträge wurden in die folgenden Sitzungen aufgeteilt:

- Thermische Systeme
- Mechatronik
- Methoden und Tools I und II
- FEM und MKS
- Co-Simulation / Echtzeit
- Model-Based Design
- E-Learning und Ausbildung
- Elektrische Systeme
- Diskrete Simulation
- FMI
- Automotive

Das Treffen wird begleitet von einer Ausstellung, auf der die anerkannten Fachfirmen ihre Produktpalette präsentieren.

Wir freuen uns, die Vertreter aus Forschungsinstituten und Industrie zum diesjährigen Workshop in Lippstadt begrüßen zu können und freuen uns gemeinsam mit Ihnen auf interessante Diskussionen und einen regen Erfahrungsaustausch.

Der vorliegende Tagungsband enthält die Langfassungen der Beiträge, für welche die Autoren selbst verantwortlich sind. In Ausnahmefällen wurden die Kurzbeiträge aufgenommen. Als Veranstalter bedanken wir uns recht herzlich bei den Autoren für die Bereitstellung der Beiträge und bei der Firma Hella KGaA & Co. für die finanzielle Unterstützung des Workshops.

Dmitrij Tikhomirov, Hochschule Hamm-Lippstadt

Heinz-Theo Mammen, FG Simulation technischer Systeme (STS)

Thorsten Pawletta, FG Grundlagen und Methoden in Modellbildung und Simulation (GMMS)

Lippstadt, im März 2016

Tagungsleitung:

Dmitrij Tikhomirov, Hochschule Hamm-Lippstadt
Heinz-Theo Mammen, ASIM/GI-STS
Thorsten Pawletta, ASIM/GI-GMMS

Organisationsteam:

Heinz-Theo Mammen
Klaus Panreck
Walter Commerell
Joachim Haase
Leo Gall
Thorsten Pawletta
Michael Striebel
Daniel Lückerath
Dmitrij Tikhomirov

Tagungsorganisation:

Hochschule Hamm-Lippstadt
Marker Allee 76-78
59063 Hamm
Tel. +49 2381 8789217
www.hshl.de

Tagungsort:

Hochschule Hamm-Lippstadt
Campus Lippstadt
Dr.-Arnold-Hueck-Str. 3
59557 Lippstadt

Veranstalter:

ASIM/GI-Fachgruppe Simulation technischer Systeme

Die Fachgruppe Simulation technischer Systeme (STS) befasst sich innerhalb der Arbeitsgemeinschaft Simulation (ASIM) mit der Modellbildung und Modellstudien für die Simulation neu zu entwickelnder oder zu verbessernder technischer Geräte und Bauteile. In der Fachgruppe finden diejenigen einen Ansprechpartner, die sich mit der Bereitstellung und Anwendung von Werkzeugen und Programmen zur Modellerstellung bei der Simulation der genannten Systeme beschäftigen. Dazu werden Fachgruppentreffen und Fachgespräche zu aktuellen Themen organisiert. Themen sind u. a. Modellierung und Simulation mechatronischer und thermischer Systeme, Simulation in der Elektronikentwicklung, in der Medizintechnik und im Automobilbau, Echtzeitsimulation, neue Methoden der Regelungstechnik, Modellierungssprachen wie Modelica und VHDL-AMS, Modellaustausch - Stichwort Functional Mockup Interface (FMI) sowie auch der Einsatz von Simulationsverfahren in der Ingenieursausbildung.

www.asim-gi.org/sts

ASIM/GI-Fachgruppe Grundlagen und Methoden in Modellbildung und Simulation

Die Fachgruppe Grundlagen und Methoden in Modellbildung und Simulation (GMMS) befasst sich in enger Zusammenarbeit von Industrie und Forschungseinrichtungen mit methodischen Entwicklungen zu Modellierungsansätzen, numerischen und softwaretechnischen Verfahren, Algorithmen, Simulationswerkzeugen sowie simulationsgestützter Experimentiermethoden (z.B. Optimierung). Weitere Schwerpunkte bilden: die Kombination von Methoden der Simulation und KI, Methoden der Verkehrssimulation und die Anwendung von Methoden der Modellbildung und Simulation in der Ausbildung. Letztere werden teilweise in speziellen Arbeitsgruppen behandelt. Generell sind von besonderem Interesse Methoden und Werkzeuge, die über mehrere Anwendungsdomänen hinweg eingesetzt werden.

www.asim-gi.org/gmms/

Hochschule Hamm-Lippstadt

Mit der Entscheidung der nordrhein-westfälischen Landesregierung, eine neue staatliche Fachhochschule in Hamm und Lippstadt aufzubauen, wurde 2009 der Grundstein für eine Hochschule neuen Profils gelegt. Präsident Prof. Dr. Klaus Zeppenfeld und Kanzler Karl-Heinz Sandknop haben gemeinsam mit ihrem Team praxiserfahrener Professorinnen und Professoren ein innovatives Studienangebot mit Fokus auf Ingenieurwissenschaften entwickelt.

Aktuell sind rund 4.500 Studierende in 14 Bachelor- sowie fünf Masterstudiengängen eingeschrieben. Besonderen Wert legt die Hochschule auf interdisziplinäre Ausrichtung, Marktorientierung und hohen Praxisbezug in der Lehre.

Mit den 2014 bezogenen Neubauten in Hamm und Lippstadt bietet die Hochschule modernste Gebäude und Einrichtungen und damit ideale Bedingungen für die Ausbildung kreativer Fachkräfte.

www.hshl.de

Inhaltsverzeichnis

	Seite
Plenarvortrag	
Flight simulation for tomorrow's aviation <i>Umut Durak</i> <i>Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Braunschweig</i>	6
Plenarvortrag	
Virtuelle Produktentwicklung bei Hella <i>Ulrich Kertscher</i> <i>Hella KGaA Hueck & Co., Lippstadt</i>	7
Plenarvortrag	
Neue Herausforderungen und Lösungen der System Synthese <i>Achim Rettberg</i> <i>Hella KGaA Hueck & Co., Lippstadt</i> (nicht im Tagungsband enthalten)	
Thermische Systeme	
Modellierung und Simulation des thermischen Runaways in zylindrischen Li-Ionen Batterien <i>Andreas Melcher, Carlos Ziebert, Boxia Lei, Wen Jiao Zhao, Magnus Rohde, Hans Jürgen Seifert</i> <i>Karlsruher Institut für Technologie (KIT), Karlsruhe</i>	8
Ein dynamisches Prozesssimulationsmodell für die energetische Betrachtung von Aluminium-Schmelzöfen in einer betriebsumfassenden Materialflusssimulation <i>Jörg Schmidt, Wolfgang Schlüter</i> <i>Hochschule Ansbach</i>	29
Parametrisierung von Lithium-Ionen-Zellen auf Basis von CIT-Messungen <i>Matthias Gulbins, Andreas Melcher, Carlos Ziebert, Boxia Lei, Thomas Markwirth, Joachim Haase</i> <i>Fraunhofer-IIS – Institutsteil Entwurfsautomatisierung EAS, Dresden</i> <i>Karlsruher Institut für Technologie (KIT) – IAM-AWP, Eggenstein-Leopoldshafen</i>	37
Vergleichende Untersuchung eines Rohrschlangen-Wärmeübertragers im Umgebungsmedium Wasser mittels numerischer Strömungsberechnung und dynamischer Prozesssimulation <i>Matthias Henninger, Wolfgang Schlüter</i> <i>Technologiezentrum Feuchtwangen</i> <i>Kompetenzzentrum Strömungssimulation, Hochschule Ansbach</i>	45
Mechatronik	
Nichtlineare Simulation zur Funktionsauslegung einer aktiven Fahrzeugfederung <i>Sven Jacobitz, Xiaobo Liu-Henke</i> <i>Fakultät Maschinenbau, Institut für Mechatronik</i> <i>Ostfalia Hochschule Wolfenbüttel</i>	54
Dynamics of the mechanical parts in mechatronics systems with VHDL AMS <i>Sergey Petkun</i> <i>Brose Fahrzeugteile GmbH & Co. Kommanditgesellschaft, Hallstadt</i>	63

Eine hierarchische Regelstruktur für das S-Mobile mit sphärischem Elektroantrieb <i>Marian Göllner, Haoqi Tao, Xiaobo Liu-Henke</i> <i>Fakultät Maschinenbau, Institut für Mechatronik</i> <i>Ostfalia Hochschule Wolfenbüttel</i>	71
Modellbasierte Diagnoseverfahren für nichtlineare DAE-Systeme <i>Dirk Weidemann, Ilja Alkov</i> <i>FH Bielefeld - Institut für Systemdynamik und Mechatronik</i>	82
Methoden und Tools I	
Towards Extending the Parallel-DEVS Formalism to Improve Component Modularity <i>Franz Preyser, Bernhard Heinzl, Philipp Raich, Wolfgang Kastner</i> <i>Institute of Computer Aided Automation, Vienna University of Technology, Vienna</i>	83
Pitfalls using discrete event blocks in Simulink and Modelica <i>Peter Junglas</i> <i>PHWT Vechta/Diepholz/Oldenburg</i>	90
Mathematisches Framework für Simulationsmodelle von State Event Modellen <i>Andreas Körner</i> <i>Technische Universität Wien, Institut für Analysis und Scientific Computing, Wien</i>	98
Entwurf und Betrieb aufgabenorientierter Multi-Robotersteuerungen mit MATLAB/Stateflow <i>Birger Freymann, Sven Pawletta, Artur Schmidt, Thorsten Pawletta</i> <i>Hochschule Wismar - University of Applied Sciences - Research Group CEA, Wismar</i>	100
FEM und MKS	
Trajectory optimization of a hypersonic plane s.t. heat limits <i>Kurt Chudej, Armin Rund, Markus Wächter</i> <i>Lehrstuhl für Ingenieurmathematik, Universität Bayreuth</i> <i>Institut für Mathematik und Wissenschaftliches Rechnen, Karl-Franzens-Universität Graz</i> <i>German Institute of Science and Technology, TUM Asia Pte Ltd, Singapore</i>	101
Experimentelle und numerische Untersuchung der Dämpfungsparameter von unverstärkten und glasfaserverstärkten Kunststoffen <i>Daniela Wurm, Jürgen Krome, Domingo Guerrero</i> <i>Hochschule Hamm-Lippstadt, Lippstadt</i> <i>Hella KG Hueck & Co. KGaA, Lippstadt</i>	109
Modellierung, Simulation und Validierung eines Betonpumpenmastes unter Berücksichtigung von Stahlelastizität und Hydraulikzylindern <i>Sophie Zorn, Thomas Breitenbach</i> <i>Fakultät Maschinenwesen</i> <i>Technische Universität Dresden</i>	113
Modelle zur Beschreibung des Schwingungsverhaltens piezoelektrischer Aktoren am Beispiel des Wanderwellenmotors <i>Jürgen Krome</i> <i>Angewandte Mechatronik</i> <i>Hochschule Hamm-Lippstadt</i>	120

Co-Simulation / Echtzeit

- On the Methodology of Cooperative and Multirate Simulation 127
Irene Hafner, Niki Popper, Felix Breitenecker
dwh GmbH
Technische Universität Wien
- Komfortuntersuchungen an Gebäudeanlagen mittels gekoppelter Coarse-Grid-CFD- und Systemsimulation 131
Stefan Wischhusen, Timo Tumforde, Johannes Brunnemann
XRG Simulation GmbH, Hamburg
- Anbindung eines aktiven Steuerkraftsystems an eine Echtzeitsimulation 142
Henrik Wüstenberg, Jürgen Gotschlich, Umut Durak
Hochschule Ostfalia
DLR Institut für Flugsystemtechnik, Braunschweig

Methoden und Tools II

- Towards a Distributed Simulation Toolbox for Scilab 148
Awad Mukbil, Peter Stroganov, Umut Durak, Sven Hartmann
Clausthal University of Technology
Department of Informatics
DLR Institute of Flight Systems, Braunschweig
- A layered structure for modeling manufacturing processes with the inclusion of energy consumption 155
Artur Schmidt, Thorsten Pawletta, Peter Junglas
Hochschule Wismar, Forschungsgruppe CEA
PHWT Vechta/Diepholz/Oldenburg
- Exploring the Advantages of Multi-Method Modelling in the Use Case of a Large Socio-Technical Infrastructure System – The Airport City 165
Barbara Glock, Niki Popper, Felix Breitenecker
dwh GmbH, dwh Simulation Services, Vienna
TU Wien, Institute for Analysis and Scientific Computing, Vienna

Model-Based Design

- Von den Requirements bis zum Seriencode - Ein durchgängiger Entwicklungsprozess mit Model-Based Design 173
Jens Lerche, Stephan Sieben
The MathWorks GmbH, Aachen
- Modellbildung und Simulation im Kontext des Systems Engineering 174
Jan Michael, Jens Holtkötter, Christian Henke, Ansgar Trächtler
Fraunhofer- Einrichtung Entwurfstechnik Mechatronik IEM, Paderborn
- Effiziente und durchgängige Entwicklung von Fahrfunktionen in der Lehre durch Simulation 179
Mirek Göbel
Hochschule Hamm-Lippstadt

E-Learning und Ausbildung

Entwicklung eines Simulators für autonome Roboter zur Ausbildung von Studierenden in der Mechatronik und Informatik 183

Axel Thümmler

Hochschule Hamm-Lippstadt

Brandschutzausbildung mittels e-learning und graphischer Simulation 195

Dieter Wloka

Fachgebiet Technische Informatik, FB 16 Elektrotechnik/Informatik

Universität Kassel

Blended Learning Methoden der Visualisierung, Demonstration, Selbstüberprüfung und Beurteilung in Mathematik, Modellbildung und Simulation 202

Andreas Körner, Stefanie Winkler, Martin Bicher, Felix Breitenecker

Technische Universität Wien, Institut für Analysis und Scientific Computing

Elektrische Systeme

Simulation von Smart Fusing-Konzepten für die dezentrale Leitungsabsicherung in zukünftigen Kfz-Bordnetzen 204

Thomas Reinders

Delphi Deutschland GmbH

Modellierung digitaler Schaltungsanteile in gemischten analog/digital Schaltkreisen für die analoge Simulation mittels Verilog-A 205

Michael Karagounis

Hochschule Hamm-Lippstadt

VDA FAT Arbeitskreis „Entwurfsmethodik und Simulation elektrischer und mechatronischer Systeme im Kfz“ 206

Thomas Lang

Robert Bosch GmbH, Reutlingen

Diskrete Simulation

Simulation eines Realfahrzyklus mit einer Lithium-Schwefel Batterie unter Berücksichtigung des "Shuttle-Effekts" 207

Christian Maurer, Walter Commerell, Claus Kröger, Ulrich Schrade

University of Applied Science Ulm

End-to-End Kliniksimulation – Absicherung der Klinik-Planung durch Simulation vollständiger Prozessketten 214

Carsten Matysczok, Markus Kühn, Moritz Schele

UNITY AG, Büren

Durchsatzoptimierung von Kommissionierrobotern durch Simulationsstudie 217

Martin Lucas

Lehrgebiet Technische Logistik

Hochschule Hamm-Lippstadt

A Simple Model of Cruising for Garage Parking 223

Oliver Ullrich, Benjamin Brandt, Daniel Lückerrath, Naphtali Rische

School of Computing and Information Sciences, Florida International University

Institut für Informatik, Universität zu Köln

FMI

Generische Visualisierung von FMU-basierten Modellen für die interaktive Simulation 230
Volker Waurich, Martin Großer, Sebastian Voigt
Professur für Baumaschinen- und Fördertechnik, TU Dresden

Verschaltung von Functional-Mockup-Units (FMUs) mit physikalischen Konnektoren am 237
Beispiel der thermischen Simulation eines Supermarktes
Wilhelm Tegethoff, Jan-Niklas Jäschke, Christian Schulze1, Nicolas Fidorra
TLK-Thermo GmbH, TU Braunschweig, Institut für Thermodynamik, Braunschweig

Simulation von mechatronischen Systemen basierend auf Modelica und FMI auf einer
kollaborativen Plattform
Michael Seibt, Dassault Systems München
(nicht im Tagungsband enthalten)

Algorithmen zur Simulatorkopplung 245
Christoph Clauß
Fraunhofer IIS EAS Dresden

Automotive

Auslegung einer erweiterten Betriebsstrategie für ein energieoptimales Elektrofahrzeug 246
Sören Scherler, Xiaobo Liu-Henke
Fakultät Maschinenbau, Institut für Mechatronik
Ostfalia Hochschule Wolfenbüttel

Cross-boundary Simulations of Fuel Cells 255
Szilárd Dombi, Walter Commerell, Hubert Mantz, Ferenc Lezsovits
Hochschule Ulm
Budapest University of Technology and Economics

Eine intelligente Fahrerassistenzfunktion zum energieoptimalen Fahrbetrieb für 258
Elektrofahrzeuge
Matthias Fritsch, Sören Scherler, Xiaobo Liu-Henke
Ostfalia Hochschule Wolfenbüttel

PC-basierte Simulation heterogener Modelle in der automobilen Software-Entwicklung 267
Gernot Deutenberg, Martin Rühl
dSPACE GmbH, Paderborn

Tutorien

Einführung zu Modelica und Dymola
Leo Gall
LTX Simulation GmbH, München
(nicht im Tagungsband enthalten)

Schneller ans Ziel mit Model-Based Design
Jens Lerche
MathWorks GmbH, Aachen
(nicht im Tagungsband enthalten)

Autorenverzeichnis 273

Sponsor- und Ausstellerverzeichnis 276

Flight Simulation for Tomorrow's Aviation

Umut Durak

DLR Institute of Flight Systems

umut.durak@dlr.de

Flight simulators have been operated within the aeronautics community for human factor studies and flight systems development for the last half century. They are virtual test beds, to evaluate concepts, conduct pilot-in-the-loop experiments and collect valuable user experience data. The German Aerospace Center (DLR) Institute of Flight Systems is involved in developing and employing research flight simulators for more than 40 years. The first generation ground based simulator of DLR was built for the HFB 320 FLISI in 1970s. The ground based simulators for ATTAS and ACT/FHS EC135 were later introduced as the second generation. The development of these simulators started at 1980s and they have been used until 2010s. In 2009, DLR Institute of Flight Systems commissioned a next generation reconfigurable research flight simulator, called the Air Vehicle Simulator (AVES) in order to address the research requirements for tomorrow's aviation. AVES has been operational since 2013 with its two cockpit infrastructures: an Airbus A320 and an Airbus Helicopters EC135. To date, various research projects have been carried out in AVES. Two of them that will be introduced in this talk as examples are Manual Operation of 4th Generation Airliners (MAN4GEN) and Enabling Technologies for Personal Air Transportation System (myCopter). While MAN4GEN aimed at identifying current deficiencies of situational awareness and manual control of modern flight decks with simulator experimentation, myCopter employed simulation experimentation for evaluating a steering wheel type control for personal air vehicles.

As an important difference from commercial flight simulators, research flight simulators require high flexibility and adaptability in almost all aspects from flight systems and flight dynamics models to cockpit displays or control loading systems. This was the key requirement that derived the AVES system design towards providing interchangeable roll-on/roll-off (RoRo) cockpits of rotorcraft (EC135) and airplanes (A320) that can be operated on motion and fixed-base platforms according to research objectives. Further, flexibility and adaptability inspired the overall simulator software design. Accordingly, 2Simulate, the real-time distributed simulation infrastructure of AVES, is being empowered by recent simulation engineering research activities that target at enhanced composability and interoperability. After a gentle introduction of 2Simulate, this talk will present a collection of these recent efforts that enable model integration at various levels. The 2Simulate Modeling Language effort proposes an Application Programming Interface for C++ model development. With Simulink Coder™ 2Simulate Generic Real-time Target; code generation for MATLAB/Simulink models is tailored for AVES. Moreover, the MATLAB/Simulink 2Simulate Toolbox that is currently being developed intends to accommodate AVES users with model-in-the-loop testing capabilities. Through 2Simulate Functional Mock-up Interface (FMI) Support, we are aiming at a FMI integration pipeline in AVES. Finally, 2Simulate High Level Architecture (HLA) Support endeavors providing capabilities for creating and joining HLA federations based on AviationSimNet Federation Object Model (FOM) in AVES.

Virtuelle Produktentwicklung bei Hella

Ulrich Kertscher

Hella KGaA Hueck & Co.

Ulrich.Kertscher@hella.com

Der Vortrag gibt Einblicke in die Entwicklung von Scheinwerfern bei Hella. Hochmodern, technisch komplex und virtuell: Die Entwicklung von Scheinwerfern – und vielen anderen Produkten auch – erfolgt heutzutage zu einem großen Teil digital. Es werden praktische Anwendung erläutert, wie mittels CAD komplexe und innovative Scheinwerfer entwickelt werden, die in Fahrzeugen in der ganzen Welt zum Einsatz kommen. Zudem wird aufgezeigt, welche Abläufe in der Praxis dahinterstecken, worin Besonderheiten liegen und welche Herausforderungen die virtuelle Produktentwicklung beinhaltet. Dabei wird speziell auf folgende Themenstellungen eingegangen:

- Vergangene, aktuelle und zukünftige Lichtprodukte bei Hella
- Aufbau eines Scheinwerfers
- Innovative Lichtverteilungen für die Scheinwerfer der Zukunft
- Hella Elektronikprodukte
- Virtuelle Produktentwicklung
- Prozesse, Tools (CATIA V4, V5, V6)
- Methodenentwicklungen bei Hella

Die beschriebenen Themenstellungen sollen einen Einblick in die Produktwelt und die Entwicklungsmethoden bei Hella geben.

Modellierung und Simulation des thermischen Runaways in zylindrischen Li-Ionen Batterien

Andreas Melcher¹, Carlos Ziebert¹, Boxia Lei¹,
WenJiao Zhao¹, Magnus Rohde¹, Hans Jürgen Seifert¹

¹ Karlsruher Institut für Technologie (KIT)
Institut für Angewandte Materialien – Angewandte Werkstoffphysik (IAM-AWP)
Hermann-von-Helmholtz-Platz 1
76344 Eggenstein-Leopoldshafen

andreas.melcher@kit.edu

Der thermische Runaway einzelner Lithium-Ionen Batterien (LIB) in größeren Batteriepacks ist ein Worst Case Szenario das unter allen Umständen vermieden werden muss. In dieser Arbeit wird das elektrochemisch-thermische Modell basierend auf der porösen Elektroden-Theorie von Newman [1] mit Beiträgen von exothermen Seitenreaktionen erweitert. Diese Reaktionen spielen die entscheidende Rolle, wenn sich durch zu starke Temperaturerhöhung innerhalb einer LIB ein thermischer Runaway ausbildet. Die Modellierung und Simulation wird am Beispiel einer einzelnen zylindrischen 18650 Zelle mit einer $LiCoO_2$ Kathode exemplarisch betrachtet. Zuerst wird das elektrochemisch-thermische Modell eingeführt und mit zusätzlichen Wärmequellen, die von den unterschiedlichen exothermen Reaktionen in der Zelle kommen, erweitert. Diese Wärmequellen lassen sich durch ein *Solid Fuel Model* bzw. *Constant Fuel Model* [2, 3, 4] modellieren.

Danach werden Simulationen des erweiterten Modells für ausgewählte Temperatur-Profile und Strom-Profile unter Variation bestimmter Systemparameter diskutiert. Das Ziel der Simulationen ist es, kritische Systemparameter zu identifizieren, sowie die räumlich-zeitliche Temperaturentwicklung der Zelle zu betrachten. Daraus wird eine Klassifizierung des thermischen Runaways unter Berücksichtigung der exothermen Reaktionen in der LIB vorgenommen. Dazu werden die Temperatur T , die Heizrate dT/dt , sowie die zweite zeitliche Ableitung der Temperatur d^2T/dt^2 betrachtet.

Im abschließenden Teil werden Simulationsergebnisse mit experimentellen Messungen verglichen sowie ein Ausblick auf zukünftige Arbeiten gegeben.

1 Einleitung

Lithium-Ionen Batterien (LIB) haben in den letzten 30 Jahren einen großen Anwendungsbereich, wie Mobilfunk-Telefone, Tablet- oder Laptop-Computer bis hin zu voll- (EV) oder hybrid-elektrischen (HEV) Fahrzeugen, gefunden. Die Größe der Batterien, sowie ihre Kapazität differiert mit der Anwendung. Die Anwendung in der Elektro-Mobilität erfordert andere Batterien und Batterie-Konfigurationen als die Anwendung im Bereich des Mobilfunks. Daraus resultiert, dass in LIB's physikalische Phänomene betrachtet werden müssen, die auf einem weiten Bereich der Zeit- und der

Längenskala auftreten, vom atomistischen Level bis zum makroskopischen Level. Insbesondere das thermische Verhalten ist eng verbunden mit der Sicherheit von Batterie-Systemen. Während des Betriebes einer LIB können verschiedene exotherme chemische Reaktionen in der Zelle stattfinden, wenn nur die innere Zelltemperatur groß genug wird. Dies geschieht insbesondere in dem Fall, wenn die in der Zelle erzeugte Wärmemenge größer ist als die Wärmemenge, die nach außen abgeführt wird. Dabei staut sich die Wärme auf, was zu einer Temperaturerhöhung in der Zelle führt. Durch diese Temperaturerhöhung werden die chemischen Reaktionen in der Zelle beschleunigt, was zu einer weiteren

Wärmeproduktion und Temperaturerhöhung führt, bis der sogenannte *Thermal Runaway* eintritt. Ein *Thermal Runaway* ist die Beschreibung für einen schnellen Temperaturanstieg in einem sehr kurzen Zeitintervall. Typische Werte hierfür sind Temperaturanstiege von größer 100K/min [5].

Ein Ansatz beruhend auf mathematischer Modellierung und Simulation verspricht einen tieferen Einblick in die Physik einer Batterie und daraus resultierend ein tieferes Verständnis der in einer LIB auftretenden Phänomene. Die Entwicklung geeigneter elektrochemischer Modelle begann mit den Arbeiten von Newman [6, 7, 8], basierend auf der Theorie poröser Elektroden [1]. Mit dieser Theorie wird die Diffusionsdynamik und der Ladungstransport in der Batterie beschrieben und man kann deshalb das elektrische Verhalten der Zelle vorhersagen. Sie ist anwendbar für kleine Batterie-Systeme, wo man die Temperaturverteilung innerhalb der Zelle als konstant annehmen kann. In größeren Batteriesystemen ist diese Annahme aufgrund der Inhomogenität des elektrischen Potential entlang der Stromkollektoren nicht mehr gültig. Deshalb muss das elektrochemische Modell mit entsprechenden thermischen Beiträgen und mit Beiträgen, die die verschiedenen Zeit- und Längenskalen mitberücksichtigen, erweitert werden. Erste thermische Modelle für LIB's wurden von Newman und Pals [9, 10, 11] eingeführt. Sie beruhen auf einer Wärmeleitungsgleichung mit den verschiedenen Wärmequellen als Inhomogenität. Dieser Ansatz wurde sowohl auf einfache Zellen, als auch auf ganze Batterien-Pakete [12, 13, 14, 15, 16] angewandt. Parallel zu dieser Entwicklung führte der Multiskalen-Charakter des elektrochemischen Modells [17, 18] zu einem neuen Beschreibungsansatz, der auf der mathematischen Homogenisierungstheorie und dem *Spatial Averaging Theorem* beruht. Neuere Ansätze in dieser Richtung kann man in den Arbeiten [20, 21, 22, 23] finden.

Die Koppelung des mehrskaligen elektrochemischen Modells und des thermischen Modells wurden in den Arbeiten [24, 25, 26] eingeführt und ist bis heute unter stetiger Weiterentwicklung [27, 28, 29, 30, 31]. Die Absicht hinter diesem Ansatz ist, ein besseres Verständnis des Verhaltens großer LIB's auf verschiedenen Längenskalen zu erhalten. Der Hauptnachteil dieses Ansatzes ist, dass man mit diesen Modellen das Phänomen des *Thermal Runaways* nicht nachbilden kann, da die dafür verantwortlichen exothermen chemischen Reaktionen in dem Modell nicht berücksichtigt sind. Einen ersten Ansatz diesen Nachteil zu beseitigen, findet man in [32]. Die Wärmeleitungsgleichung wird mit einem System gewöhnlicher Differentialgleichungen gekoppelt, die

die zeitliche Entwicklung der exothermen chemischen Reaktionen beschreiben. Das erste auf partiellen Differentialgleichungen beruhende Modell zur Beschreibung des *Thermal Runaway* bei LIB's wurde von Spatznitz et. al [33, 34] entwickelt. In den Arbeiten [35, 36, 37] wurde das elektrochemisch-thermische Modell mit einer Beschreibung der exothermen Reaktionen beruhend auf gewöhnlichen Differentialgleichungen erweitert. Aktuelle Forschungsergebnisse können in den Arbeiten [38, 39, 40] gefunden werden. In [38] wird eine LIB durch externe Erwärmung in den *Thermal Runaway* gebracht (*Ofenversuch*), während die Arbeit [39] den *Thermal Runaway* bei Lithium-Titanat Batterien untersucht, mit einem Modell ähnlich wie in [38] beschrieben. In der Dissertation von Tanaka [40] wird eine detaillierte Übersicht über die Modellierung und Simulation des *Thermal Runaways* bei LIB's unter Berücksichtigung verschiedener Zellchemien gegeben.

Diese Arbeit ist wie folgt gegliedert: Im nächsten Abschnitt 2 wird eine kurze allgemeine Übersicht über die elektrochemisch-thermische Modellierung einer LIB zur Beschreibung eines *Thermal Runaway* gegeben. Im Abschnitt 2.1 wird die allgemeine thermische Modellierung beruhend auf einer inhomogenen Wärmeleitungsgleichung beschrieben, gefolgt von dem multiskaligen elektrochemischen porösen Elektroden-Modell in Abschnitt 2.2. Abschließend wird in 2.3 die exotherme Reaktionskinetik modelliert. In Abschnitt 3 werden Ergebnisse verschiedener Simulationen gezeigt, und zwar wird der Ofenversuch in Abschnitt 3.1 beschrieben, gefolgt von Simulationen mit einem elektrischen Lastprofil in 3.2. Im Abschnitt 4 werden aus den Simulationen mit elektrischem Lastprofil kritische Systemparameter abgeleitet, während im Abschnitt 5 Simulationen mit experimentellen Messungen qualitativ verglichen werden. Im abschließenden Abschnitt 6 werden eine Zusammenfassung und ein Ausblick gegeben.

2 Thermische Modellierung von Lithium-Ionen Batterien

Zur Modellierung des thermischen Verhaltens von LIB's liegt im Folgenden der Hauptfokus in der Betrachtung einer einzelnen Zelle. Es wird angenommen, dass das Innere der Zelle von der umgebenden Luft durch das Batteriebehältnis getrennt ist, so dass man die Zelle als abgeschlossenes System betrachten kann.

2.1 Thermische Modellierung von Lithium-Ionen Batterien mit der Wärmeleitungsgleichung

Unter allgemeinen Arbeitsbedingungen wird eine LIB elektrisch und/oder thermisch belastet. Daraus resultiert, dass im Inneren der Zelle Wärme erzeugt wird durch verschiedene elektrochemische und chemische Prozesse. Wenn die erzeugte Wärme in der Zelle geringer ist als die Möglichkeit, Wärme nach außen abzuführen, dann arbeitet die Zelle in einem stabilen thermischen Zustand. Wenn nun aber mehr Wärme im Inneren der Zelle erzeugt wird, als an die Umgebung abgegeben werden kann, dann ist die Zelle in einem instabilen thermischen Zustand. In diesem Zustand ist das Worst Case Szenario das Auftreten eines *Thermal Runaway* mit einer anschließenden Zerstörung der Zelle. Der kritische Aspekt im Falle des *Thermal Runaway* ist seine Beziehung zur Energieerhaltung in der Zelle. Mit Hilfe der Energieerhaltung ist man in der Lage, die thermischen Charakteristiken einer Zelle, wie die Wärmeerzeugung und Wärmeabführung zu beschreiben. Für die Betrachtungen im Folgenden wird angenommen, dass die Zellgeometrie bzw. das Zellvolumen konstant ist, d.h. elastische Prozesse und Gaserzeugung in der LIB werden nicht betrachtet. Dann kann die allgemeine mathematische Gleichung für die Energieerhaltung als Anfangs-Randwert-Problem mit der Wärmeleitungsgleichung beschrieben werden [2, 3, 4].

$$\rho c_p \frac{\partial T}{\partial t}(\mathbf{x}, t) = \kappa \Delta T(\mathbf{x}, t) + Q(\mathbf{x}, t) \quad (1)$$

für das Temperaturfeld $T : (\mathbf{x}, t) \in \Omega \times \mathbb{R}^+ \mapsto \mathbb{R}$ der Zelle in K. Die dazugehörigen Anfangs- und Randbedingungen sind

$$\begin{aligned} T(\mathbf{x}, 0) &= T_0(\mathbf{x}), \quad \forall \mathbf{x} \in \bar{\Omega}, \\ \mathbf{n} \cdot (\kappa \nabla T) &= -h(T - T_{env}) - \varepsilon \sigma (T^4 - T_{env}^4), \quad \forall \mathbf{x} \in \partial\Omega. \end{aligned} \quad (2)$$

In diesem Zusammenhang bezeichnet $\mathbf{x} \in \bar{\Omega} \subset \mathbb{R}^3$ einen räumlichen inneren Punkt oder Oberflächenpunkt der Zelle und $t \in \mathbb{R}^+ := \{t \in \mathbb{R}, t \geq 0\text{s}\}$ die Zeit. T_0 bezeichnet das anfängliche Temperaturfeld in der Zelle zum Zeitpunkt $t = 0\text{s}$. O.B.d.A. wird angenommen, dass die Anfangstemperatur auf der Oberfläche der Zelle identisch mit der Umgebungstemperatur ist. Ω bezeichnet das Innere der Zelle, $\partial\Omega$ den Rand und der Abschluss $\bar{\Omega} = \Omega \cup \partial\Omega$ die komplette Zelle. Weiter ist $\mathbf{n} \in \mathbb{R}^3$ der nach außen gerichtete Einheitsnormalenvektor, h der Wärmeübertragungskoeffizient in $\text{W}/\text{K} \cdot \text{m}^2$, ε die einheitenlose Emissivität, $\sigma = 5.670373 \cdot 10^{-8} \text{W}/\text{m}^2 \cdot \text{K}^4$

die Stefan-Boltzmann Konstante, ρ ist die effektive Dichte der LIB in kg/m^3 , c_p die Wärmekapazität in J/K und κ die Wärmeleitfähigkeit in $\text{W}/\text{m} \cdot \text{K}$.

Die Lösung von Gleichung (1) beschreibt die Temperaturverteilung in der Zelle für Zeiten $t > 0\text{s}$ und alle räumlichen Punkte $\mathbf{x} \in \Omega$. Die erste Gleichung in (2) beschreibt die initiale Temperaturverteilung zum Zeitpunkt $t = 0\text{s}$, während die zweite Gleichung in (2) die Wärmeabfuhr nach außen beschreibt. Der erste Term auf der rechten Seite dieser Gleichung beschreibt den Wärmeabfluss nach außen durch Konvektion, während der zweite Term den Wärmeabfluss durch Strahlung beschreibt. In der Inhomogenität Q von Gleichung (1) sind all die verschiedenen Wärmequellen beinhaltet. Diese Beiträge kommen von exothermen Seitenreaktionen in der Zelle, jouscher und ohmscher Wärme, sowie von reversiblen und irreversiblen thermodynamischen Effekten:

$$Q(\mathbf{x}, t) = Q_{el-chem}(\mathbf{x}, t) + Q_{exotherm}(\mathbf{x}, t) \quad (3)$$

mit

$$Q_{el-chem}(\mathbf{x}, t) = Q_{Joule} + Q_{rev} + Q_{irrev} + Q_{Ohm}. \quad (4)$$

Dabei sind die elektrochemischen Wärmequellen wie folgt gegeben:

$$\begin{aligned} Q_{Joule} &= U \cdot I, \\ Q_{rev}(\mathbf{x}, t) &= I \cdot T(\mathbf{x}, t) \cdot \frac{\partial U_{eq}}{\partial T}, \\ Q_{irrev} &= (U_{eq} - U) \cdot I, \\ Q_{Ohm}(\mathbf{x}, t) &= \sigma^{eff} \|\nabla \phi_s(\mathbf{x}, t)\|^2 + \kappa^{eff} \|\nabla \phi_e(\mathbf{x}, t)\|^2 \\ &\quad + \kappa_D^{eff} \nabla \ln c_e(\mathbf{x}, t) \cdot \nabla \phi_e(\mathbf{x}, t). \end{aligned} \quad (5)$$

I ist das angelegte äußere Stromprofil in A, U die Batteriespannung und U_{eq} die Gleichgewichtsspannung der Zelle, jeweils in V. $\phi_{s,e}$ sind die elektrischen Potentiale im Fest- und Flüssiganteil der Zelle in V. c_e bezeichnet die Konzentration der Lithium-Ionen im Elektrolyt in mol/m^3 . Die Konstanten σ^{eff} , κ^{eff} und κ_D^{eff} sind effektive elektrische Leitfähigkeiten in S/m .

Die kritischen Variablen in der gesamten elektrochemischen Wärmequelle sind die Terme U , U_{eq} , $\partial U_{eq}/\partial T$ sowie die Potentiale $\phi_{s,e}$ und die Konzentration c_e . Diese Variablen müssen mit dem porösen Elektroden-Modell berechnet werden. Das angelegte Stromprofil I kann entweder direkt in die obigen Gleichungen eingehen oder muss ebenfalls aus dem porösen Elektroden-Modell berechnet werden mit Hilfe des *Spatial Averaging Theorem* [19] aus den entsprechenden Stromdichten. Die Zellspannung U muss aus den elektrischen Potentialen der Fest- und Flüssigphase innerhalb der Zelle bestimmt

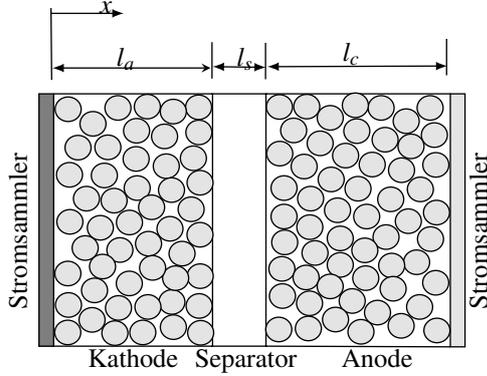


Abbildung 1: Schematischer Aufbau einer LIB.

werden. Ein kurzer Abriss über das poröse Elektroden-Modell wird in Abschnitt 2.2 gegeben.

Für die exotherme Wärmequelle wird angenommen, dass in der LIB n unabhängige exotherme chemische Reaktionen stattfinden. Jede diese Reaktionen gehorcht einem einfachen Arrhenius-Gesetz. Dann ist exotherme Wärmequelle

$$Q_{exotherm}(\mathbf{x}, t) = \sum_{i=1}^n Q_i(\mathbf{x}, t) \quad (6)$$

mit

$$\begin{aligned} Q_i(\mathbf{x}, t) &= q_i R_i(\mathbf{x}, t), \\ R_i(\mathbf{x}, t) &= A_i c_i^{m_i}(\mathbf{x}, t) \exp\left(-\frac{E_{a,i}}{RT(\mathbf{x}, t)}\right), \end{aligned} \quad (7)$$

dabei ist q_i die abgegebene Wärme in J/g, R_i die Reaktionsrate in 1/s, A_i der Frequenzfaktor in 1/s, m_i die dimensionlose Reaktionsordnung, $E_{a,i}$ die Aktivierungsenergie in J/mol und $R = 8.314462 \text{ J/mol K}$ die universelle Gaskonstante.

Die Konzentrationen $c_i(\mathbf{x}, t)$ müssen aus einem passenden Reaktions-Diffusions-Modell bestimmt werden, das in Abschnitt 2.3 kurz skizziert wird.

2.2 Das elektrochemische poröse Elektroden Modell

In Abbildung 1 ist der schematische Aufbau einer Zelle ist nach Cai et.al [27] dargestellt. Im Allgemeinen besteht eine Lithium-Ionen Batterie aus zwei Stromsammlern, der positiven und der negativen Elektrode (Kathode, Anode) und einem Separator zwischen den beiden Elektroden. Die Elektroden bestehen aus einer porösen Struktur gebildet aus den aktiven Partikeln und Poren, die mit

dem Elektrolyt gefüllt sind. Zur Bestimmung der elektrochemischen Wärmequelle muss man bei diesem Modell in Verbindung mit dem Multiskalenansatz das Gebiet betrachten, das von den aktiven Partikeln eingenommen wird, sowie die Zelle bestehend aus den drei Gebieten, die von der positiven und negativen Elektrode sowie dem Separator eingenommen werden.

2.2.1 Partikel-Modellierung

Im Gebiet der aktiven Partikel wird die Materialbilanz der Lithium Ionen durch das zweite Fick'sche Gesetz beschrieben. Weiter wird zur Vereinfachung der Betrachtungen bezüglich der aktiven Partikel in den beiden Elektroden angenommen, dass die Partikel alle aus idealen Sphären mit den Radien $R_{s,i}$, $i \in \{a, c\}$, für die negative bzw. positive Elektrode, bestehen. Des weiteren wird die Betrachtung soweit vereinfacht, dass für die Elektroden die entsprechenden Gleichungen jeweils nur für ein Partikel repräsentativ gelöst werden. Die Wärme- und Stromdichten aus der Gesamtheit der aktiven Partikel werden durch eine geeignete Homogenisierung bestimmt. Die Konzentration der Lithium-Ionen in einem solchen Partikel ist die Lösung der ersten Gleichung aus (8), welche ein parabolisches Anfangs-Randwert-Problem beschreiben:

$$\begin{aligned} \frac{\partial c_{s,\{a,c\}}}{\partial t} &= \frac{1}{r^2} D_{s,\{a,c\}} \frac{\partial}{\partial r} \left(r^2 \frac{\partial c_{s,\{a,c\}}}{\partial r} \right), \\ c_{s,\{a,c\}} \Big|_{t=0} &= c_{s,\{a,c\},0}(r), \\ \frac{\partial c_{s,\{a,c\}}}{\partial r} \Big|_{r=0} &= 0, \\ \frac{\partial c_{s,\{a,c\}}}{\partial r} \Big|_{r=R_s} &= -\frac{1}{D_{s,\{a,c\}} F} J_{\{a,c\}}. \end{aligned} \quad (8)$$

In den obigen Gleichungen sind die Konzentrationen $c_{s,\{a,c\}}$ in mol/m^3 gegeben. r bezeichnet die radiale Koordinate in m, $c_{s,\{a,c\},0}(r)$ in der zweiten Gleichung von (8) ist die initiale Konzentrationsverteilung entlang der radialen Koordinate zum Zeitpunkt $t = 0$ s. $D_{s,\{a,c\}}$ ist der Diffusionskoeffizient in der negativen/positiven Elektrode in m^2/s und $F = 96485.336 \text{ C/mol}$ ist die Faraday Konstante. Weiter findet im Zentrum des sphärischen Partikels kein Ionenfluss statt, dies wird durch die dritte Gleichung aus (8) beschrieben. Die letzte Gleichung in (8) beschreibt den Fluss der Lithium-Ionen. Dieser Fluss ist gleich der Produktionsrate der dazu gehörigen elektrochemischen Reaktion am *Solid-Electrolyte-Interface* (SEI). $J_{\{a,c\}}$ in mol/m^2 ist der Fluss der Lithium Ionen weg von der Oberfläche des Partikels.

2.2.2 Elektroden-Modellierung

Da die Zelle sich aus drei Teilgebieten zusammensetzt, muss man die Material- und Ladungsbilanz für die Lithium-Ionen in jedem dieser Teilgebiet separat bestimmen.

Der Transport der Lithium Ionen durch den Elektrolyt wird stark beeinflusst durch die Porenstruktur der Elektroden und des Separators und der Befeuchtung der Porenoberfläche. Unter der Annahme der statistischen Homogenität kann dies durch eine vereinfachte eindimensionale Diffusionsgleichung entlang der Richtung zwischen den Stromsammlern beschrieben werden. Deshalb ist die Konzentration die Lösung von

$$\frac{\partial \varepsilon_{e,\{a,s,c\}} c_{e,\{a,s,c\}}}{\partial t} = \frac{\partial}{\partial x} \left(D_{e,\{a,s,c\}}^{eff} \frac{\partial c_{e,\{a,s,c\}}}{\partial x} \right) + (1 - t_+^0) S_{\{a,s,c\}} J_{\{a,s,c\}} \quad (9)$$

mit den Anfangs- und Randbedingungen

$$\begin{aligned} c_{e,\{a,s,c\}} \Big|_{t=0} &= c_{e,\{a,s,c\},0}(x), \\ \frac{\partial c_{e,\{c\}}}{\partial x} \Big|_{x=0} &= 0, \\ \frac{\partial c_{e,\{a\}}}{\partial x} \Big|_{x=l_a+l_s+l_c} &= 0. \end{aligned} \quad (10)$$

Dabei bezeichnet $c_{e,\{a,s,c\},0}(x)$ die Anfangsverteilung der Lithium Ionen im Elektrolyt, $\varepsilon_{e,\{a,s,c\}}$ die Volumenfraktion des Elektrolyten in der negativen/positiven Elektrode sowie dem Separator, $S_{\{a,s,c\}}$ ist die Oberfläche der Elektroden per Einheitsvolumen in 1/m. Die effektive Diffusivität der Elektroden und des Separators ist $D_{e,\{a,s,c\}}^{eff}$ in m^2/s . $J_{\{a,s,c\}}$ ist der Stromfluss, wobei man o.B.d.A annehmen kann, dass im Separator kein Stromfluss auftritt, also $J_s = 0$. Weiter bezeichnet t_+^0 die dimensionslose Transportzahl. Die zweite und dritte Gleichung in (10) bedeuten das am Stromsammler kein Ionenfluss mehr stattfindet.

Darüber hinaus müssen an den Übergängen von den Elektroden zum Separator noch Stetigkeitsbedingungen für die Konzentration der Lithium Ionen und deren Fluss erfüllt sein:

$$\begin{aligned} c_{e,a} \Big|_{x=l_a^-} &= c_{e,s} \Big|_{x=l_a^+}, \\ c_{e,s} \Big|_{x=(l_a+l_s)^-} &= c_{e,c} \Big|_{x=(l_a+l_s)^+} \end{aligned} \quad (11)$$

und

$$\begin{aligned} D_{e,a}^{eff} \frac{\partial c_{e,a}}{\partial x} \Big|_{x=l_a^-} &= D_{e,s}^{eff} \frac{\partial c_{e,s}}{\partial x} \Big|_{x=l_a^+}, \\ D_{e,s}^{eff} \frac{\partial c_{e,s}}{\partial x} \Big|_{x=(l_a+l_s)^-} &= D_{e,c}^{eff} \frac{\partial c_{e,c}}{\partial x} \Big|_{x=(l_a+l_s)^+} \end{aligned} \quad (12)$$

Die Ladungsbilanz der Festphase in den beiden Elektroden wird durch das Ohm'sche Gesetz bestimmt, d.h.

$$\sigma_{\{a,c\}}^{eff} \frac{\partial^2 \phi_{s,\{a,c\}}}{\partial x^2} = S_{\{a,c\}} F J_{\{a,c\}}. \quad (13)$$

$\sigma_{\{a,c\}}^{eff}$ ist die effektive elektrische Leitfähigkeit der festen Phase in den Elektroden, $\phi_{s,\{a,c\}}$ das elektrische Potential in der festen Phase der Elektroden und $S_{\{a,c\}} F J_{\{a,c\}}$ repräsentiert eine volumetrische Stromdichte in A/m^3 . Weiter müssen noch Randbedingungen an der Grenzfläche zwischen dem Stromsammler und den Elektroden formuliert werden, sowie Übergangsbedingungen an der Grenzfläche zwischen Elektroden und den Separator. An der Grenzfläche zwischen dem Stromsammler und der positiven Elektrode ist der Ladungsfluss gleich der angelegte Stromdichte I_{app} in A/m^2 , also

$$-\sigma_p^{eff} \frac{\partial \phi_{s,c}}{\partial x} \Big|_{x=0} = I_{app}. \quad (14)$$

An der Grenzfläche zwischen Stromsammler und negativer Elektrode verschwindet das elektrische Potential, d.h.

$$\phi_{s,a} \Big|_{x=l_c+l_s+l_c} = 0. \quad (15)$$

An der Grenzfläche zwischen den Elektroden und dem Separator verschwindet der Ladungsfluss, was

$$\frac{\partial \phi_{s,c}}{\partial x} \Big|_{x=l_c} = \frac{\partial \phi_{s,c}}{\partial x} \Big|_{x=l_c+l_s} = 0 \quad (16)$$

bedeutet.

In der flüssigen Phase wird die Ladungsbilanz ebenfalls durch das Ohm'sche Gesetz bestimmt. Unter der Annahme, dass die effektiven Leitfähigkeiten räumlich konstant sind, ergibt dies

$$-\kappa_{\{a,s,c\}}^{eff} \frac{\partial^2 \phi_{e,\{a,s,c\}}}{\partial x^2} + \kappa_{D,\{a,s,c\}}^{eff} \frac{\partial^2 \ln c_{e,\{a,s,c\}}}{\partial x^2} = a_{\{a,s,c\}} F J_{\{a,s,c\}} \quad (17)$$

mit $\kappa_{D,\{a,s,c\}}^{eff} = \frac{2RT(1-t_+^0)\kappa_{\{a,s,c\}}^{eff}}{F}$. Die Randbedingungen zwischen der flüssigen Phase der Elektroden und dem

Stromsammler sind:

$$\left. \frac{\partial \phi_{e,c}}{\partial x} \right|_{x=0} = \left. \frac{\partial \phi_{e,a}}{\partial x} \right|_{x=l_a+l_s+l_c} = 0. \quad (18)$$

Des weiteren muss man beachten, dass das Potential und sein Fluss in der Flüssigphase stetig an den Grenzflächen zwischen den Elektroden und dem Separator sind.

2.2.3 Modellierung der elektrochemischen Wärmequellen

In den obigen Gleichungen kann der Fluss an den Porenwänden aus einer Butler-Vollmer Beziehung bestimmt werden:

$$J_{\{a,c\}} = \frac{k_i(c_{e,\{a,c\}})^{\alpha_1} (c_{s,\{a,c\}max} - c_{s,\{a,c\},surf})^{\alpha_1}}{(c_{s,\{a,c\},surf})^{\alpha_2}} \left[\exp\left(\frac{\alpha_1 F}{RT} \eta\right) - \exp\left(-\frac{\alpha_2 F}{RT} \eta\right) \right]. \quad (19)$$

Dabei bezeichnen $\alpha_{\{1,2\}}$ dimensionslose Übertragungskoeffizienten, $R = 8.314462 \text{ J/molK}$ ist die universelle Gas Konstante. T ist die Temperatur in K, $c_{s,e}$ ist die Konzentration der Lithium Ionen in der festen/flüssigen Phase der Elektrodenpartikel in mol/m^3 , $c_{s,\{a,c\},max}$, $c_{s,\{a,c\},surf}$ sind die maximal mögliche Konzentration und die Konzentration an der Partikeloberfläche. η bezeichnet das Überpotential

$$\eta_{\{a,c\}} = \phi_{s,\{a,c\}} - \phi_{e,\{a,c\}} - U_{eq} \quad (20)$$

in V. U_{eq} ist die Gleichgewichtsspannung und $\phi_{s,e}$ das elektrische Potential in der festen und flüssigen Phase der Zelle in V.

Dann sind die verschiedenen Wärmequellen wie folgt gegeben: Die Joule'sche Wärme ist definiert als Strom multipliziert mit der Zellspannung, d.h.

$$Q_{Joule} = U \cdot I \quad (21)$$

mit dem Strom $I = SFJ$ in A, dabei ist S das totale spezifische Grenzflächengebiet der Zelle in $1/\text{m}$, J ist der totale Fluss der Lithium Ionen, der mit dem *Spatial Average Theorem* berechnet werden muss aus $J_{\{a,c\}}$ über die gesamte Geometrie der Zelle in $\text{mol} \cdot \text{m/s}$. U ist die Spannung der Zelle definiert als die Differenz des Potentials zwischen der festen und flüssigen Phase der Zelle, also $U = \phi_s - \phi_e$. Daraus ergibt sich dann der irreversible Wärmeanteil der elektrochemischen Reaktionen zu

$$Q_{irrev} = I \cdot (U_{eq} - U) \quad (22)$$

und der reversible Wärmeanteil ist

$$Q_{rev} = I \cdot T \cdot \frac{\partial U_{eq}}{\partial T}. \quad (23)$$

Letztendlich ist der Ohmsche Wärmeanteil

$$Q_{Ohm} = \sigma^{eff} \left(\frac{\partial \phi_s}{\partial x} \right)^2 + \kappa^{eff} \left(\frac{\partial \phi_e}{\partial x} \right)^2 + \kappa_D^{eff} \frac{\partial \ln c_e}{\partial x} \cdot \frac{\partial \phi_e}{\partial x} \quad (24)$$

$$\text{mit } \kappa_D^{eff} = \frac{2RT(1-t_+^0)\kappa^{eff}}{F}.$$

2.3 Modellierung des thermischen Missbrauchs und des *Thermal Runaway* bei Lithium-Ionen Batterien

Exotherme Reaktionskinetik ist eng verbunden mit den thermischen Missbrauchsmechanismen in einer LIB. Verschiedene exotherme chemische Reaktionen können in einer Batterie auftreten, wenn die Temperatur derselben ansteigt. Diese Reaktionen erzeugen zusätzliche Wärmebeiträge, die die exothermen chemischen Reaktionen noch weiter beschleunigen, wenn diese Wärmebeiträge nicht nach außen abgeführt werden können [41]. Dies bedeutet, dass die Rate der Wärmeerzeugung größer ist als die Rate der Wärmeabfuhr und führt deshalb zur einer Erhöhung der Temperatur innerhalb der LIB. Äußere Bedingungen für eine solchen Temperaturanstieg können die Erwärmung der LIB von außen sein, Über- oder Unterladen der Zelle, Laden der Zelle mit hohen Ladeströmen, mechanische Beschädigung der Zelle von außen, z.B. Nagelpenetration, äußerer Kurzschluss u.v.m.. In all diesen Fällen kann ein *Thermal Runaway* auftreten mit Leckage der Zelle, Rauch- und Gasentwicklung, Feuer etc. und führt dann zu einer Zerstörung der Zelle. Zur Beschreibung des Missbrauchsverhaltens und des *Thermal Runaway* haben verschiedene Autoren diverse Modelle vorgeschlagen. Ein erster Ansatz die exotherme Reaktionskinetik bei LIB's zu beschreiben, findet man in [32]. Ein vereinfachtes Reaktions-Diffusionsmodell wird in den Arbeiten [35, 36, 37] beschrieben. Eine Modellierung der Nagelpenetration bei LIB's findet man in [42]. Ein analytischer Ansatz beruhend auf der Katastrophentheorie wird in [43, 44] gegeben.

Um das Missbrauchsverhalten bzw. den *Thermal Runaway* bei LIB's zu beschreiben, muss man die hauptsächlichen exothermen chemischen Reaktion identifizieren. Der allgemeine Mechanismus, der zu einem *Thermal Runaway* führt, kann in vier Schritten beschrieben werden [5, 32, 36, 37, 45]:

Exotherme Reaktion	Temperatur T_{start}
Anodenreaktionen	363.15K
SEI-Zersetzung	393.15K
Kathodenreaktionen	435.15K
Elektrolytzersetzung	465.15K

Tabelle 1: Anfangstemperaturen für die exothermen Reaktionen in einer LIB mit $LiCoCO_2$ -Elektrode

- (1) Bei Temperaturen $T > T_1$ wird die Grenzfläche der aktiven festen Partikel zum Elektrolyt in der negativen Elektrode in einer exothermen Reaktion zerstört.
- (2) Bei Temperaturen $T > T_2$ findet eine exotherme Reaktion zwischen den im aktiven Partikel der negativen Elektrode eingelagerten Lithium-Ionen und dem Elektrolyt statt.
- (3) Bei Temperaturen $T > T_3$ findet eine exotherme Reaktion zwischen dem aktiven Material in der positiven Elektrode und dem Elektrolyt unter Freisetzung von Sauerstoff statt.
- (4) Bei Temperaturen $T > T_4$ findet in einer abschließenden exothermen Reaktion die Zersetzung des Elektrolyts statt.

Für eine negative Elektrode mit einer $LiCoCO_2$ -Chemie sind nach [5] die Anfangstemperaturen in Tabelle 1 gegeben.

2.3.1 Das Solid-Fuel Model

Eine passende Theorie zur Beschreibung exothermer chemischer Reaktionen ist die mathematische Verbrennungstheorie [2, 3]. In dieser Theorie, ist wenn man hauptsächlich an der zeitlichen Entwicklung der Temperaturverteilung interessiert ist, die Benutzung des sogenannten *Solid Fuel Models* ein möglicher Ansatz.

Ein *Solid Fuel Model* ist ein gekoppeltes System parabolischer Differentialgleichungen bestehend aus den Gleichungen in (1) und (2) für das Temperaturfeld T sowie einem System zusätzlicher parabolischer Differentialgleichungen mit entsprechenden Anfangs- und Randbedingungen, die die zeitliche-räumliche Evolution der Konzentrationen der chemische Reaktionsprodukte der exothermen kinetischen Reaktionen beschreiben.

Im Folgenden wird angenommen, dass die verschiedenen

exothermen Reaktionen unabhängig voneinander sind. Mit Ω_i , $i \in \{sei, ne, pe, el\} \subset \Omega \subset \mathbb{R}^3$ und $\bigcup_i \Omega_i \subset \Omega$ wird das räumliche Gebiet bezeichnet in dem die exotherme Reaktion stattfindet, $\partial\Omega_i$ ist der zugehörige Rand und $\bar{\Omega}_i = \Omega_i \cup \partial\Omega_i$ der Abschluss. Dann gehorcht die i -te exotherme Reaktion dem Anfangs-Randwert-Problem

$$\begin{aligned} \frac{\partial c_i}{\partial t}(\mathbf{x}, t) &= d_i \Delta c_i(\mathbf{x}, t) + \gamma_i R_i(\mathbf{x}, t) \\ c_i(\mathbf{x}, 0) &= c_{i,0}(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_i, \\ c_i(\mathbf{x}, t) &= 0, \quad \forall \mathbf{x} \in \partial\Omega_i. \end{aligned} \quad (25)$$

Hier ist $c_{i,0}(\mathbf{x})$ die anfängliche Konzentration, d_i ist der Diffusionskoeffizient, γ_i der stöchiometrische Koeffizient. Die zugehörige exotherme Wärmequelle kann dann gemäß der Gleichungen (6) und (7) bestimmt werden.

2.3.2 Vereinfachung: Das Constant Fuel Model

Für die Simulationen in Abschnitt 3 wird das *Solid Fuel Model* weiter vereinfacht zu einem sogenannten *Constant Fuel Model* [4, 46]. Dabei wird die Dynamik der Konzentrationen c_i in Raum und Zeit vernachlässigt, wenn man annimmt, dass für alle Zeitpunkte $t > 0$ s und räumliche Punkte \mathbf{x} gilt

$$\begin{aligned} c_i(\mathbf{x}, t) &= c_{i,0}, \quad \forall t > 0, \quad x \in \Omega_i \\ m_i &= 1. \end{aligned} \quad (26)$$

Das reduziert das obige *Solid Fuel Model* auf das Anfangs-Randwert-Problem für die Wärmeleitungsgleichung (1) und (2) und die exothermen Wärmequellen sind dann wie folgt definiert:

$$\begin{aligned} Q_i(\mathbf{x}, t) &= q_i R_i(\mathbf{x}, t) \\ R_i(\mathbf{x}, t) &= A_i c_{i,0} \exp\left(-\frac{E_{a,i}}{RT(\mathbf{x}, t)}\right). \end{aligned} \quad (27)$$

Die exothermen Wärmequellen sind dann nur noch von der Temperatur T abhängig. Die Menge $i \in \{sei, pe, ne, ele\}$ bezeichnet die vier oben beschriebenen exothermen Reaktionen für die Zersetzung der SEI-Schicht, die Reaktionen in den positiven und negativen Elektroden mit dem Elektrolyt, und die Zersetzung des Elektrolyts.

3 Simulationen

Das mathematische Modell, das im Abschnitt 2 skizziert wurde, stellt einen multiskaligen Mehrgebietsansatz dar.

Um dieses Modell zu implementieren und das räumlich-zeitliche Verhalten einer LIB zu simulieren, wurde die kommerzielle Software *COMSOL Multiphysics* gewählt. Diese Software beinhaltet ein sogenanntes *Battery and Fuel Cell Module*, das es erlaubt in vereinfachter Art und Weise Batterien und Brennstoffzellen zu simulieren mit Hilfe vordefinierter Templates.

Dazu muss der Benutzer das poröse Elektroden Modell, das *Solid Fuel Model* bzw. *Constant Fuel Model* und das thermische Modell entsprechend in der Software definieren mit der dazugehörigen Koppelung, sowie der Zellgeometrie, den Simulationsparametern und der zeitlichen und räumlichen Auflösung.

Für die Simulationen wurde eine zylindrische 18650 Lithium-Ionen Batterie gewählt. Aus der Materialdatenbank von *COMSOL Multiphysics* wurde als negative Elektrode eine Li_xC_6 Anode, als positive Elektrode eine Li_xCoO_2 Kathode und als Elektrolyt ein 1:1 EC:DEC mit $LiPF_6$ als Leitsalz gewählt. Die internen spiralförmigen Windungen der LIB wurden nicht aufgelöst. In Bezug auf die exothermen Reaktionen in der LIB wurde das *Constant Fuel Model* verwendet.

In einer zylindrischen Geometrie, bleiben die Gleichungen für das Partikel-Modell und die Gleichungen zur Beschreibung der Dynamik der Elektroden unverändert, nur die Wärmeleitungsgleichung muss in Zylinderkoordinaten $\mathbf{X} = (r, \theta, z)$ neu formuliert werden:

$$\begin{aligned} \frac{\partial(\rho_{batt}c_{p,batt}T)}{\partial t} = & - \left(\frac{1}{r} \frac{\partial}{\partial r} \left(\kappa_{T,rad,batt} r \frac{\partial T}{\partial r} \right) \right. \\ & + \frac{1}{r^2} \kappa_{T,ang,batt} \frac{\partial^2 T}{\partial \theta^2} \\ & \left. + \kappa_{T,ang,batt} \frac{\partial^2 T}{\partial z^2} \right) \\ & + Q. \end{aligned} \quad (28)$$

$\kappa_{T,\{rad,ang\}}$ ist die effektive thermische Leitfähigkeit in radialer und angularer Richtung, ρ_{batt} ist die effektive Batteriedichte und $c_{p,batt}$ die effektive Wärmekapazität. Nach [35] sind diese effektiven Parameter wie folgt gegeben:

$$\begin{aligned} \kappa_{T,ang,batt} &= \frac{\sum_{i \in \mathcal{I}} \kappa_{T,i} L_i}{L_{batt}}, \\ \kappa_{T,rad,batt} &= \frac{\sum_{i \in \mathcal{I}} \frac{L_i}{\kappa_{T,i}}}{L_{batt}}, \\ \rho_{batt} &= \frac{\sum_{i \in \mathcal{I}} \rho_i L_i}{L_{batt}}, \\ c_{p,batt} &= \frac{\sum_{i \in \mathcal{I}} c_{p,i} L_i}{L_{batt}}. \end{aligned} \quad (29)$$

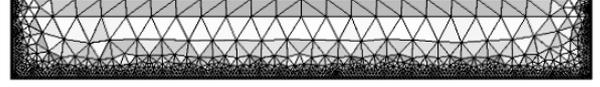


Abbildung 2: Vernetzung einer zylindrischen 18650 LIB in der $r - z$ -Ebene.

Dabei bezeichnet die Indexmenge $\mathcal{I} = \{pos., neg., cc_{pos}, cc_{neg}, sep\}$ die Beiträge der positiven/negativen Elektrode, des Separators und des Stromsammlers.

Mit dem oben beschriebenen Modell wurden nach der Implementierung verschiedene Simulationen durchgeführt. Im ersten Fall ein sogenannter Ofenversuch, bei dem die LIB von außen her durch die Erhöhung der Umgebungstemperatur erwärmt wird. Im zweiten Fall wird die Zelle von innen erwärmt durch ein Stromprofil. Die bei den Simulationen benutzten Parameter sind in den Tabellen 7 und 8 im Anhang A zu finden.

Die Simulationen wurden mit den Versionen 4.4 und 5.2 von *COMSOL Multiphysics* durchgeführt. In den Simulationen wurde für die Zeitintegration ein BDF-Verfahren (Backward Differentiation Formulas) mit einer minimalen Ordnung von 1 und einer maximalen Ordnung von 5, sowie einer variablen Steuerung der Schrittweite verwendet. Aus Gründen der numerischen Genauigkeit wurde der maximale Zeitschritt $\Delta t = 1s$ gewählt. Da die räumliche Diskretisierung in *COMSOL Multiphysics* auf der Finiten Element Methode (FEM) beruht, wurde eine adaptive räumliche FEM-Diskretisierung der Teilmodelle für die LIB benutzt. Das Modell für die Partikel wird in *COMSOL Multiphysics* automatisch gelöst, so dass man nur das eindimensionale Elektroden-Modell und die gesamte Zellgeometrie diskretisieren muss. Zur Diskretisierung des Elektroden-Gebietes wurde eine maximale Größe der Finiten Elemente von $1\mu m$ mit insgesamt 168 Elementen mit quadratischen Basisfunktionen genommen. Die Auflösung der dreidimensionalen Zellgeometrie wurde unter Ausnutzung der Invarianz der Winkelkoordinate in der $r - z$ -Ebene durchgeführt mit 2266 Dreieckselementen, mit Elementgrößen im Intervall $[3.9 \cdot 10^{-4}, 8.45 \cdot 10^{-2}]m$, ebenfalls unter Benutzung quadratischer Ansatzfunktionen. Die FEM-Diskretisierung ist in Abbildung 2 gezeigt.

t [s]	$T_{rad,min}$ [K]	$T_{rad,max}$ [K]	\bar{T}_{rad} [K]
100.38	305.96	312.66	307.87
880,38	461.98	465.66	462.23
884.38	472.27	474.57	473.23
887.38	490.49	496.41	495.27
887.76	495.71	504.78	503.43
888.27	516.66	558.02	554.07
888.3	518.84	624.24	607.95

Tabelle 2: Minimale,maximale und mittlere Zelltemperatur entlang der radialen Achse

3.1 Simulation eines Ofenversuchs

Bei der Simulation des Ofenversuchs wurde die Umgebungstemperatur T_{env} linear mit einer konstanten Heizrate von $dT_{env}/dt = 5 \cdot 10^{-2} \text{ K/s}$ ausgehend von der Anfangstemperatur T_0 gesteigert. Diese Heizrate wurde in eine Oberflächenwärmequelle Q_{surf} umgewandelt und als zusätzliche Randbedingung gemäß der Formel

$$\mathbf{n} \cdot (\kappa \nabla T) = Q_{surf}, \forall \mathbf{x} \in \partial\Omega \quad (30)$$

in das thermische Modell integriert. Weiter wurde der konvektive Wärmeübertragungskoeffizient h auf 0 gesetzt um adiabatische Bedingungen zu realisieren.

Die Simulation wurde zur Zeit $t = 888.3 \text{ s}$ mit einer mittleren Zelltemperatur $\bar{T} = 1053 \text{ K}$ und einer mittleren Oberflächentemperatur $\bar{T}_{surf} = 598 \text{ K}$ beendet. In Anhang B, Abbildung 3 ist die mittlere Zelltemperatur, mittlere Oberflächentemperatur und die Temperatur eines inneren Punktes im aktiven Batteriematerial gezeigt.

Zur weiteren Auswertung wurden in einem ersten Schritt die räumliche Entwicklung der Zelltemperatur bei dem horizontalen Querschnitt von $z = 0.0325 \text{ m}$ zu verschiedenen Zeitpunkten $t \in \{880.38, 887.38, 888.27\} \text{ s}$ betrachtet. Die zugehörigen Contour-Plots und radiale Temperaturprofile sind im Anhang B in der Abbildung 4 wiedergegeben. Die in Abbildung 4 gezeigten radialen Temperatur-Profile sind repräsentativ für einen Ofenversuch. Solange die exothermen Reaktionen in der Zelle eine untergeordnete Rolle spielen, wird die Zelle von Außen erwärmt, so dass sich ein räumlicher Temperaturgradient von außen nach innen einstellt mit dem Temperaturmaximum am Rand ($t = 880.38 \text{ s}$). Wenn nun das aktive Material in der Zelle soweit erwärmt wurde, dass die exothermen Reaktionen einen nicht zu vernachlässigen-

t [s]	$\Delta T_{rad,max}$ [K]	$\Delta T_{rad,ext}$ [K]	$\Delta T_{rad,int}$ [K]
100.38	6.69	0	6.69
880,38	3.67	0	3.67
884.38	2.30	0	2.30
887.38	5.92	1.56	5.92
887.76	9.07	1.67	9.07
888.27	41.96	41.96	8.97
888.3	105.41	105.41	47.17

Tabelle 3: Abweichungen der Zelltemperaturen im Zentrum und am Rand von der maximalen radialen Zelltemperatur

den Wärmebeitrag liefern, ändert sich das radiale Temperaturprofil derart, dass das Temperaturmaximum innerhalb des aktiven Batteriematerials angenommen wird mit zwei räumlichen Temperaturgradienten, einem Größeren zum Rand hin und einem Kleineren zum radialen Zentrum der Zelle hin ($t = 887.38 \text{ s}$). Befindet sich die Zelle schließlich im *Thermal Runaway* ($t = 888.27 \text{ s}$) so bleiben die radialen Temperaturgradienten erhalten, wobei diese Gradienten noch weiter zunehmen. Das radiale Temperaturprofil im aktiven Batteriematerial bildet ein Plateau bei maximaler Temperatur aus. In Tabelle 2 sind die maximale sowie minimale radiale Temperatur $T_{rad,max}$, $T_{rad,min}$ sowie die mittlere radiale Temperatur \bar{T}_{rad} zu verschiedenen Zeitpunkten angeben. In Tabelle 3 sind die Abweichungen zwischen minimaler und maximaler Zelltemperatur $\Delta T_{rad,max}$, sowie die Abweichungen von der maximalen radialen Temperatur zum Zentrum $\Delta T_{rad,int}$ und zum Rand der Zelle $\Delta T_{rad,ext}$ angeben.

Man sieht aus der letzten Tabelle, dass bei den letzten vier Zeitpunkten, d.h. zu Beginn und während des *Thermal Runaway* die Änderung in der maximalen Temperaturerhöhung d.h. $\Delta^2 T_{rad,max} = \Delta \Delta T_{rad,max}$ zunimmt.

Deshalb wurde in einem zweiten Schritt die zeitliche Dynamik der Zelle mit Hilfe der mittleren Zelltemperatur

$$\bar{T}(t) = \frac{1}{vol(\Omega)} \int_{\Omega} T(\mathbf{x}, t) d\Omega \quad (31)$$

und deren erster und zweiter zeitlichen Ableitung auf einem äquidistanten Zeitgitter mit $\Delta t = 0.1 \text{ s}$ betrachtet. Die zeitlichen Ableitungen $d\bar{T}/dt$ und $d^2\bar{T}/dt^2$ wurden aus der mittleren Zelltemperatur \bar{T} numerisch mit Hilfe eines Savitzky-Golay-Algorithmus zur Regularisation der Schlechtgestellttheit der numerischen Ableitung bestimmt. In Anhang B in der Abbildung 5 (a)-(c) ist der

t [s]	$T_{rad,min}$ [K]	$T_{rad,max}$ [K]	\bar{T}_{rad} [K]
200.96	306.09	306.36	306.28
1800.8	385.15	385.54	385.44
1830.1	470.83	476.95	475.72
1832.8	493.89	512.54	510.03
1833.1	504.13	536.37	532.91
1833.2	512.99	660.15	588.32

Tabelle 4: Minimale, maximale und mittlere Zelltemperatur entlang der radialen Achse

zeitliche Verlauf der mittleren Zelltemperatur \bar{T} und der ersten und zweiten Ableitung wiedergegeben. In Abbildung 5 (d) und (e) ist der funktionale Zusammenhang von $d\bar{T}/dt$ und $d^2\bar{T}/dt^2$ als Funktion von \bar{T} in einem einfach-logarithmischen Maßstab dargestellt. In Abbildung 5 (f) ist $d^2\bar{T}/dt^2$ als Funktion von $d\bar{T}/dt$ in einem zweifach-logarithmischen Maßstab dargestellt. Aus den beiden Abbildungen 5 (d) und (e) ersieht man, dass für hinreichend große Temperaturen \bar{T} die beiden ersten Ableitungen in dem einfach-logarithmischen Maßstab einen linearen Zusammenhang haben, was bedeutet:

$$\frac{d^i \bar{T}}{dt^i} \sim A_i \exp(B_i \cdot \bar{T}), \quad i = 1, 2. \quad (32)$$

Darüber hinaus zeigt sich in Abbildung 5 (f) für hinreichend große Heizraten $d\bar{T}/dt$ ein linearer Zusammenhang zur zweiten Ableitung $d^2\bar{T}/dt^2$ im zweifach-logarithmischen Maßstab, d.h.

$$\frac{d^2 \bar{T}}{dt^2} \sim a \cdot \left(\frac{d\bar{T}}{dt} \right)^b. \quad (33)$$

In Abbildung 5 (a)-(f) sind des Weiteren noch zusätzliche Linien gezeichnet. Die gestrichelten Linien bezeichnen die Temperaturen für das Auftreten der exothermen Reaktion nach [5] (s.Tabelle 1) und die zugehörigen Heizraten $d\bar{T}/dt$ und $d^2\bar{T}/dt^2$. Die gepunkteten Linien repräsentieren die Temperaturen \bar{T} und zweiten Ableitungen $d^2\bar{T}/dt^2$ bei Heizraten $d\bar{T}/dt \in \{10, 100\}$ K/min. Mit Hilfe dieser Linien wird eine noch detaillierte Interpretation des *Thermal Runaway* möglich sein.

t [s]	$\Delta T_{rad,max}$ [K]	$\Delta T_{rad,ext}$ [K]	$\Delta T_{rad,int}$ [K]
200.96	0.27	0.27	0.0028
1800.8	0.35	0.35	0.0024
1830.1	6.13	6.13	0.08
1832.8	18.66	18.66	0.82
1833.1	32.44	32.44	3.09
1833.2	87.15	87.15	31.25

Tabelle 5: Abweichungen der Zelltemperaturen im Zentrum und am Rand von der maximalen radialen Zelltemperatur

3.2 Simulation eines elektrischen Lastprofils unter adiabatischen Bedingungen

Analog zum Vorgehen im letzten Abschnitt 3.1 wurden adiabatische Simulationen für die LIB durchgeführt mit einem einfachen Stromprofil in Rechteckform mit einer Amplitude in der Stromdichte von $I_{app} = 80 \text{ A/m}^2$ und einem Lade- bzw. Entladeintervall von jeweils $T_{ch/disch} = 250 \text{ s}$ ohne Relaxationspausen dazwischen.

In Anhang B, Abbildung 6 sind die Contour-Plots und zugehörigen radialen Temperaturprofile zu drei Zeitpunkten gezeigt. In diesem Fall wird die Zelle von Innen zuerst durch Wärme der elektrochemischen Reaktionen und später auch durch die exothermen Reaktionen erhitzt. Deshalb prägt sich in diesem Fall nur ein generelles radiales Temperaturprofil aus, das ein Maximum im aktiven Batteriematerial hat und zum Rand und Zentrum der Zelle einen radialen Temperaturgradient aufweist, wobei der Gradient zum Rand stets größer ist als der Gradient zum Zentrum.

In Tabelle 4 sind analog zu Tabelle 2 die maximale sowie minimale radiale Temperatur $T_{rad,max}$, $T_{rad,min}$ sowie die mittlere radiale Temperatur \bar{T}_{rad} zu verschiedenen Zeitpunkten angegeben. In Tabelle 5 sind analog zu Tabelle 3 die Abweichungen zwischen minimaler und maximaler Zelltemperatur $\Delta T_{rad,max}$, sowie die Abweichungen von der maximalen radialen Temperatur zum Zentrum $\Delta T_{rad,int}$ und zum Rand der Zelle $\Delta T_{rad,ext}$ nochmals angegeben. Auch in diesem Fall ist $\Delta^2 T_{rad,max}$ im *Thermal Runaway* stark wachsend.

In Abbildung 7 (a)-(c) ist der zeitliche Verlauf der mittleren Zelltemperatur \bar{T} und der zugehörigen ersten und zweiten Ableitungen gezeigt. Die Unstetigkeiten in den Plots für $d\bar{T}/dt$ und $d^2\bar{T}/dt^2$ resultieren aus den Sprüngen im Stromprofil, wenn zwischen Lade- und Entladehalbzzyklus gewechselt wird. In den Bildern (d)-(e) wird

	(a ₁)	(a ₂)
$T_{env} =$	273.15 K	293.15 K
$h \in$	[7.4725, 7.48] W/m ² K	[8.945, 8.95] W/m ² K
$\Delta h_{krit} =$	0.0075 W/m ² K	0.005 W/m ² K
	(b ₁)	(b ₂)
$h =$	8 W/m ² K	10 W/m ² K
$T_{env,krit} \in$	[281.05, 281.15] K	[304, 304.0625] K
$\Delta T_{env} =$	0.1 K	0.0625 K

Tabelle 6: Kritische Parameterintervalle für den konvektiven Wärmeübertragungskoeffizient h und Umgebungstemperatur T_{env}

wieder der funktionale Zusammenhang von $d\bar{T}/dt$ und $d^2\bar{T}/dt^2$ als Funktion von \bar{T} im halb-logarithmischen Maßstab dargestellt. In Bild (f) ist erneut $d^2\bar{T}/dt^2$ als Funktion von $d\bar{T}/dt$ im doppel-logarithmischen Maßstab dargestellt. Für hinreichend große Temperaturen \bar{T} und Heizraten $d\bar{T}/dt$ gelten auch hier die Zusammenhänge aus den Gleichungen (32) und (33).

4 Bestimmung kritischer Systemparameter

Mit Hilfe der Gleichung (31) kann man aus den Simulationen aus Abschnitt 3.2 kritische Parameter bestimmen. Dafür werden in einem ersten Fall (a) für verschiedene festgelegte Umgebungstemperaturen T_{env} der konvektive Wärmetransfer h mittels eines Bisektionsverfahrens variiert. In einem zweiten Fall (b) ist der konvektive Wärmetransfer h fixiert und die Umgebungstemperaturen T_{env} variabel. Man ist dann in der Lage, die mittlere kritische Zelltemperatur \bar{T}_{krit} , oberhalb derer ein *Thermal Runaway* auftritt, zu bestimmen. Darüber hinaus kann man approximativ den kritischen konvektiven Wärmetransferkoeffizient h_{krit} bzw. die kritische Umgebungstemperatur $T_{env,krit}$ bestimmen. Abbildung 8 (a) zeigt das Ergebnis dieses Vorgehens für eine Umgebungstemperatur $T_{env} = 273.15$ K und den konvektiven Wärmeübertragungskoeffizient $h \in [0, 24]$ W/m² · K, sowie in Abbildung 8 (b) für $h = 8$ W/m² · K und $T_{env} \in [273.15, 373.15]$ K.

In Tabelle 6 sind die entsprechenden Intervalle für die kritischen Parameter jeweils für zwei Umgebungstemperaturen und für zwei konvektive Wärmeübertragungskoeffizienten dargestellt.

Aus den Simulationen, die in Abbildung 8 gezeigt werden, erhält man approximativ die mittlere kritische Zelltemperatur als

$$\bar{T}_{krit} \approx 400 \text{ K.} \quad (34)$$

Diese Temperatur ist nach [5] ungefähr die Temperatur bei der die Zersetzung der SEI-Schicht einsetzt, und ist als $T_{sei,decomp} \approx 393.15$ K angeben.

5 Qualitativer Vergleich von experimentellen Messungen mit Simulationen

Im Anhang B in Abbildung 9 oberes Bild ist der zeitliche Verlauf eines *Thermal Runaway*-Experimentes gezeigt. Dieser Versuch wurde mit einer LIB mit einer *LiCoO₂*-Kathode in einen ARC (Accelerating Rate Calorimeter) durchgeführt. Die Zelltemperatur wurde mit einem Thermoelement vom Typ N, das auf der Zelloberfläche befestigt, war gemessen. Der zeitliche Verlauf der Temperaturkurve zeigt den vollständigen Verlauf eines *Thermal Runaway* inklusive der Temperaturabnahme nach Zerstörung der Zelle. Qualitativ zeigt der funktionale Zusammenhang der ersten und zweiten zeitlichen Ableitung als Funktion der Temperatur in Abbildung 9 unteres Bild analoges Verhalten wie in Abbildung 5 (d)-(f). Für hinreichend große Temperaturen und Heizraten ist der funktionale Zusammenhang gemäß den Gleichungen (32) und (33) gegeben, obwohl die Kurven, was bei experimentellen Messungen zu erwarten ist, verrauscht sind.

6 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein einfaches gekoppeltes Modell zur elektrochemisch-thermischen Simulation eines *Thermal Runaway* bei Lithium-Ionen Batterien eingeführt. Das thermische Grundmodell ist dabei mit dem elektrochemischen und exothermen Modell gekoppelt. In den Simulationen wurde die raumzeitliche Entwicklung horizontaler Querschnitte betrachtet und das radiale Temperaturprofil bei einem Ofenversuch und unter elektrischer Belastung bestimmt. Mit Hilfe der mittleren Zelltemperatur und deren ersten und zweiten Ableitung wurde ein funktionaler Zusammenhang zwischen diesen Größen hergestellt. Daraus könnte man in Zukunft ein Kriterium für den Eintritt des *Thermal Runaway* ableiten. Weiter

wurden die Simulationsergebnisse qualitativ mit einer experimentellen Messung verglichen und gezeigt, dass für hinreichend große Temperaturen und Heizraten der funktionale Zusammenhang zwischen Temperatur und den ersten beiden zeitlichen Ableitungen auch bei Messungen gefunden werden kann. Das Modell in dieser Arbeit ist so formuliert, dass es sich leicht erweitern lässt, z.B. durch detaillierte räumliche Modellierung des elektrochemischen Modells oder des exothermen Modells. Ziel zukünftiger Arbeiten wird es sein, die räumliche Dynamik während eines *Thermal Runaway* besser aufzulösen und Kriterien zu bestimmen, so dass ein *Thermal Runaway* rechtzeitig erkannt werden kann und gegebenenfalls durch entsprechendes Eingreifen auch verhindert werden kann.

Danksagung

Diese Arbeit wurde finanziell unterstützt vom BMBF im Rahmen des Verbundprojektes *IKEBA-integrierte Komponenten und integrierter Entwurf energieeffizienter Batteriesysteme* (Zuwendung 16N12515) unter Betreuung des Projektträger VDI/VDE Innovation + Technik GmbH.

A Simulationsparameter

Anfangswerte	
Li-Konz. feste Phase (neg.)	$c_{s,neg,0} = 7917 \frac{\text{mol}}{\text{m}^3}$
Li-Konz. feste Phase (pos.)	$c_{s,pos,0} = 16002 \frac{\text{mol}}{\text{m}^3}$
Li-Konz. flüssige Phase	$c_{e,0} = 2000 \frac{\text{mol}}{\text{m}^3}$
Anfangstemperatur	$T_0 \in [273.15, 373, 15] \text{ K}$
Diffusionskonstanten	
Neg. Festphase	$D_{s,neg} = 3.9 \cdot 10^{-14} \frac{\text{m}^2}{\text{s}}$
Pos. Festphase	$D_{s,pos} = 1 \cdot 10^{-13} \frac{\text{m}^2}{\text{s}}$
Volumenfraktionen	
Neg. Festphase	$\epsilon_{s,neg} = 0.471$
Pos. Festphase	$\epsilon_{s,pos} = 0.297$
Neg. Flüssigphase	$\epsilon_{e,neg} = 0.357$
Pos. Flüssigphase	$\epsilon_{e,pos} = 0.444$

Maximale Li-Konzentration	
Neg. Festphase	$c_{s,max,neg} = 26390 \frac{\text{mol}}{\text{m}^3}$
Pos. Festphase	$c_{s,max,pos} = 22860 \frac{\text{mol}}{\text{m}^3}$
Wärmeleitfähigkeiten	
Neg. Elektrode	$\kappa_{T,neg} = 1.04 \frac{\text{W}}{\text{m}\cdot\text{K}}$
Pos. Elektrode	$\kappa_{T,pos} = 1.58 \frac{\text{W}}{\text{m}\cdot\text{K}}$
Neg. Stromsammler	$\kappa_{T,cc,neg} = 298.15 \frac{\text{W}}{\text{m}\cdot\text{K}}$
Pos. Stromsammler	$\kappa_{T,cc,pos} = 170 \frac{\text{W}}{\text{m}\cdot\text{K}}$
Dichten	
Neg. Elektrode	$\rho_{neg} = 1347.33 \frac{\text{kg}}{\text{m}^3}$
Pos. Elektrode	$\rho_{pos} = 2328.5 \frac{\text{kg}}{\text{m}^3}$
Neg. Stromsammler	$\rho_{neg,cc} = 8933 \frac{\text{kg}}{\text{m}^3}$
Pos. Stromsammler	$\rho_{pos,cc} = 2770 \frac{\text{kg}}{\text{m}^3}$
Partikelradien	
Neg. Elektrode	$r_{p,neg} = 12.5 \cdot 10^{-6} \text{ m}$
Pos. Elektrode	$r_{p,pos} = 8 \cdot 10^{-6} \text{ m}$
Elektrochemische Reaktionsratenkoeffizienten	
Neg. Elektrode	$k_{neg} = 2 \cdot 10^{-11} \frac{\text{m}}{\text{s}}$
Pos. Elektrode	$k_{pos} = 2 \cdot 10^{-11} \frac{\text{m}}{\text{s}}$
Wärmekapazität	
Neg. Elektrode	$c_{p,neg} = 1437.4 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
Pos. Elektrode	$c_{p,pos} = 1269.21 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
Neg. Stromsammler	$c_{p,cc,neg} = 385 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
Pos. Stromsammler	$c_{p,cc,pos} = 875 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
Separator	
Dichte	$\rho_{sep} = 1008.98 \frac{\text{kg}}{\text{m}^3}$
Wärmeleitfähigkeit	$\kappa_{T,sep} = 0.344 \frac{\text{W}}{\text{m}\cdot\text{K}}$
Wärmekapazität	$c_{p,sep} = 1978.16 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
Batteriegeometrie	
Radius	$r_{batt} = 9 \text{ mm}$
Höhe	$h_{batt} = 65 \text{ mm}$
Mandrel Radius	$r_{mand} = 2, \text{ mm}$
Zelldicke	$L_{batt} = 157 \mu\text{m}$
Dicke Batteriebehältnis	$d_{can} = 0.25 \text{ mm}$
Dicke neg. Stromsammler	$L_{neg,cc} = 7 \mu\text{m}$
Dicke pos. Stromsammler	$L_{pos,cc} = 10 \mu\text{m}$
Länge pos./neg. Elektrode	$L_{pos/neg} = 55 \mu\text{m}$
Länge Separator	$L_{sep} = 30 \mu\text{m}$

Tabelle 7: Simulationsparameter.

$x \in \{sei, pe, ne, e\},$ $y \in \{c, p, e\}$	Reaktionswärme $H_x / \left[\frac{\text{J}}{\text{kg}} \right]$	Frequenz- faktor $A_x / \left[\frac{1}{\text{s}} \right]$	Aktivierungsenergie $E_A / \left[\frac{\text{J}}{\text{mol}} \right]$	Volumenanteil $W_y / \left[\frac{\text{kg}}{\text{m}^3} \right]$
SEI reaction	$2.57 \cdot 10^5$	$1.667 \cdot 10^{15}$	$1.3508 \cdot 10^5$	$1.39 \cdot 10^3$
Neg. solvent reaction	$1.714 \cdot 10^6$	$2.5 \cdot 10^{13}$	$1.3508 \cdot 10^5$	$1.39 \cdot 10^3$
Pos. solvent reaction	$3.14 \cdot 10^5$	$6.667 \cdot 10^{13}$	$1.396 \cdot 10^5$	$1.3 \cdot 10^3$
Electrolyte decomp.	$1.55 \cdot 10^5$	$5.14 \cdot 10^{25}$	$2.74 \cdot 10^5$	$5 \cdot 10^2$

Tabelle 8: Exotherme Simulationsparameter.

B Simulationsergebnisse für den Ofenversuch und elektrisches Belastungsprofil

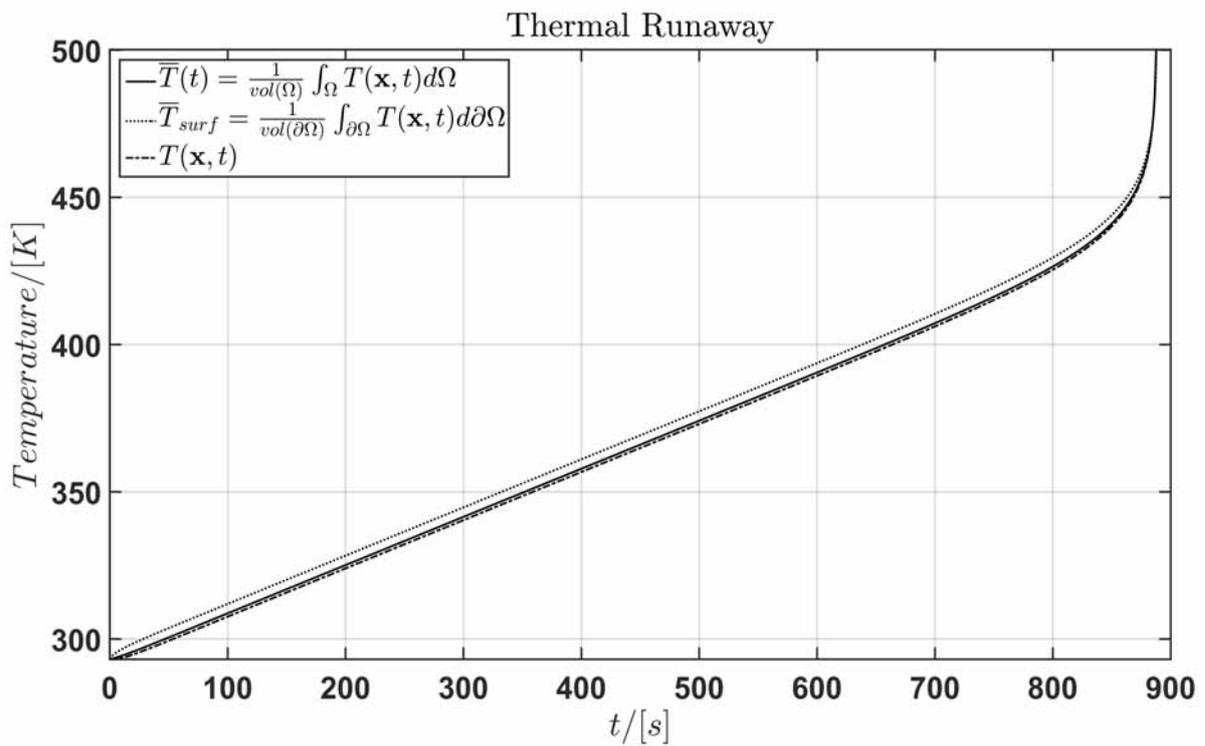


Abbildung 3: Ofenversuch: Zeitliche Temperaturentwicklung der mittleren Zelltemperatur, mittleren Oberflächentemperatur, der Temperatur eines inneren Punktes im aktiven Batteriematerial

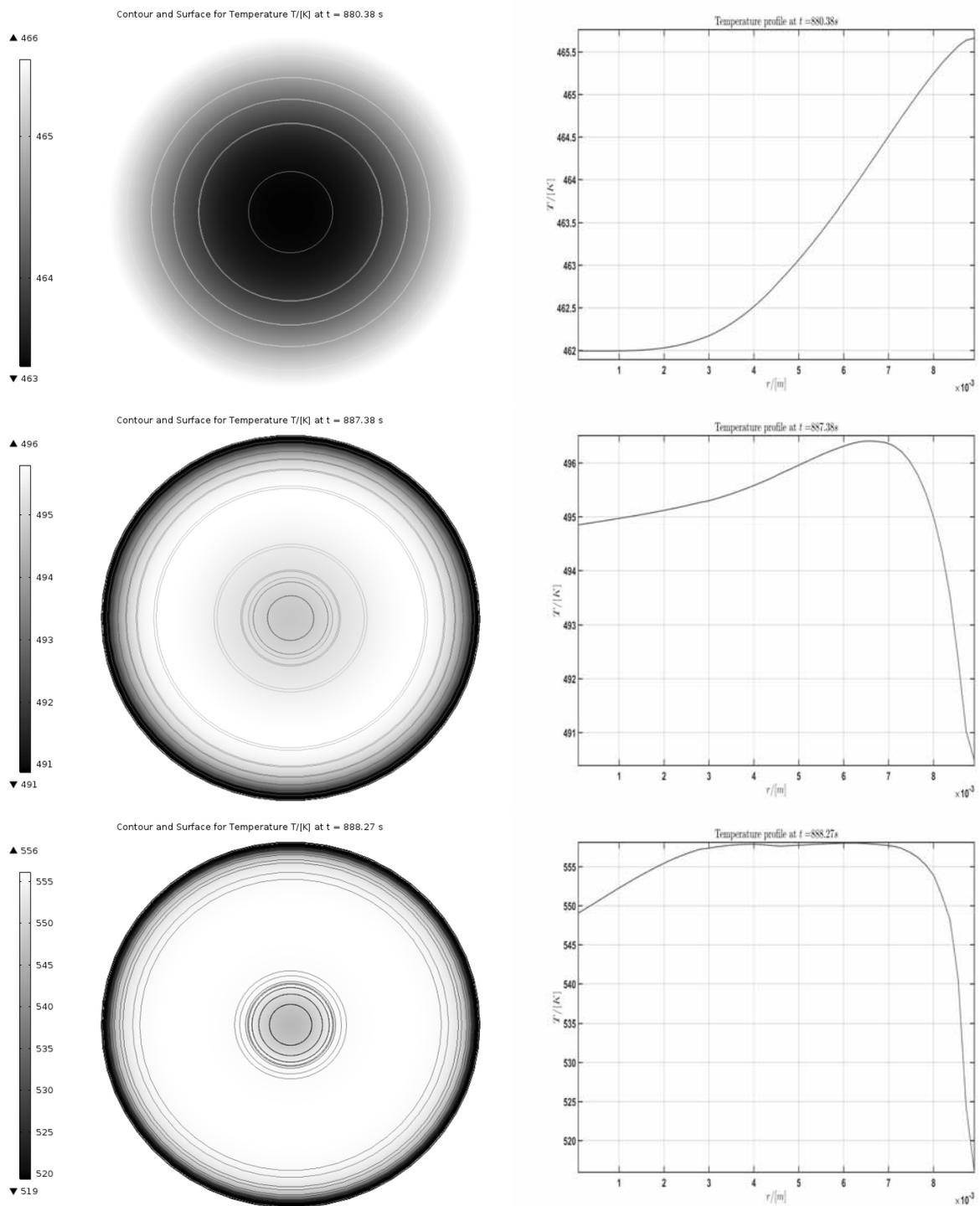


Abbildung 4: Ofenversuch: Contour-Plot und zugehöriges radiales Temperaturprofil in der xy-Ebene bei $z = 0.00325\text{m}$ zu den Zeitpunkten $t = 880.38\text{ s}$, $t = 887.38\text{ s}$ und $t = 888.27\text{ s}$

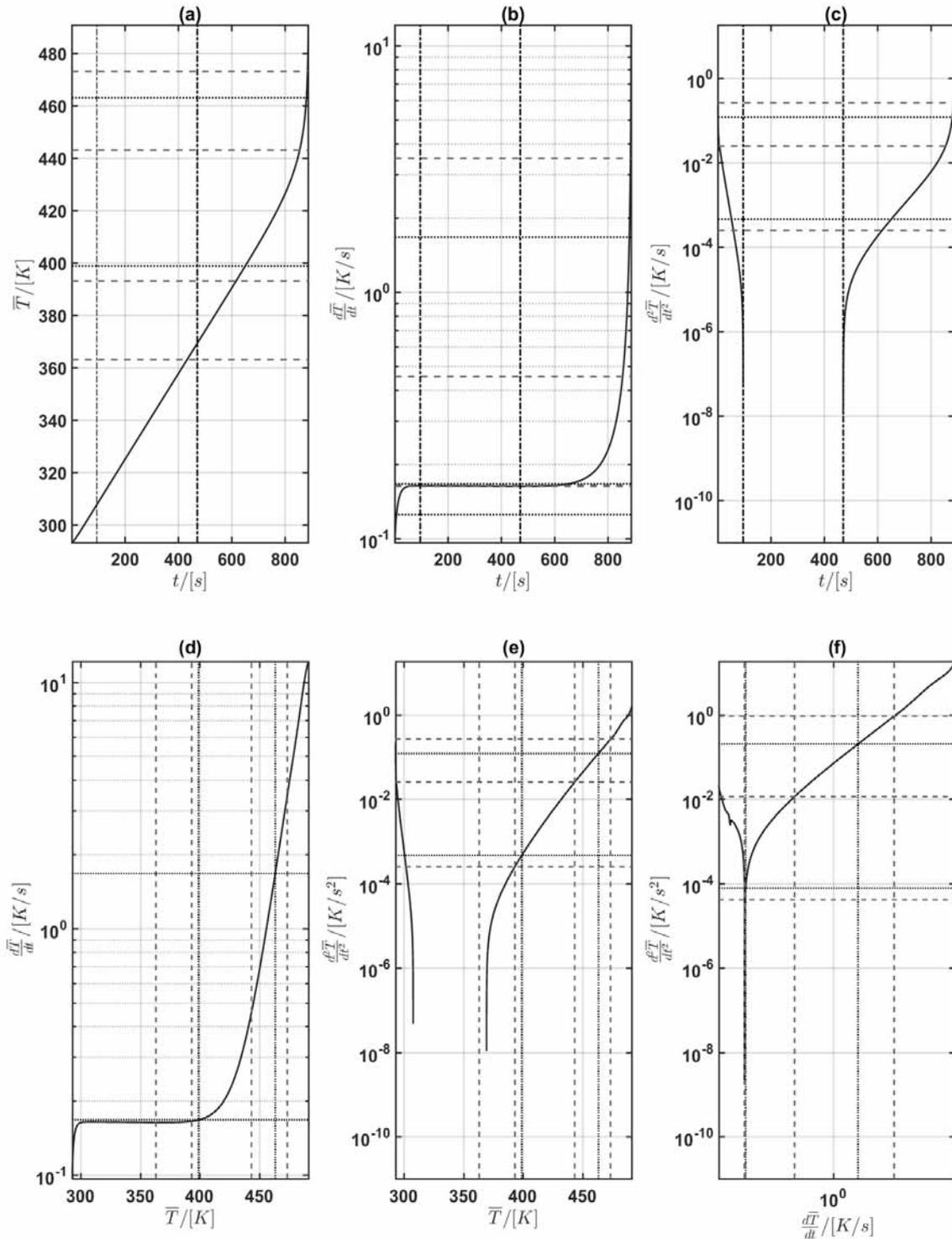


Abbildung 5: Zeitliche Entwicklung während des Ofenversuches: (a)-(c) Zeitentwicklung der mittleren Zelltemperatur \bar{T} , zugehörige Heizrate $d\bar{T}/dt$ und Temperaturbeschleunigung $d^2\bar{T}/dt^2$. (d)-(e) Temperatur-Heizraten-, Temperatur-Temperaturbeschleunigungs-, Heizraten-Temperaturbeschleunigungs-Plot.

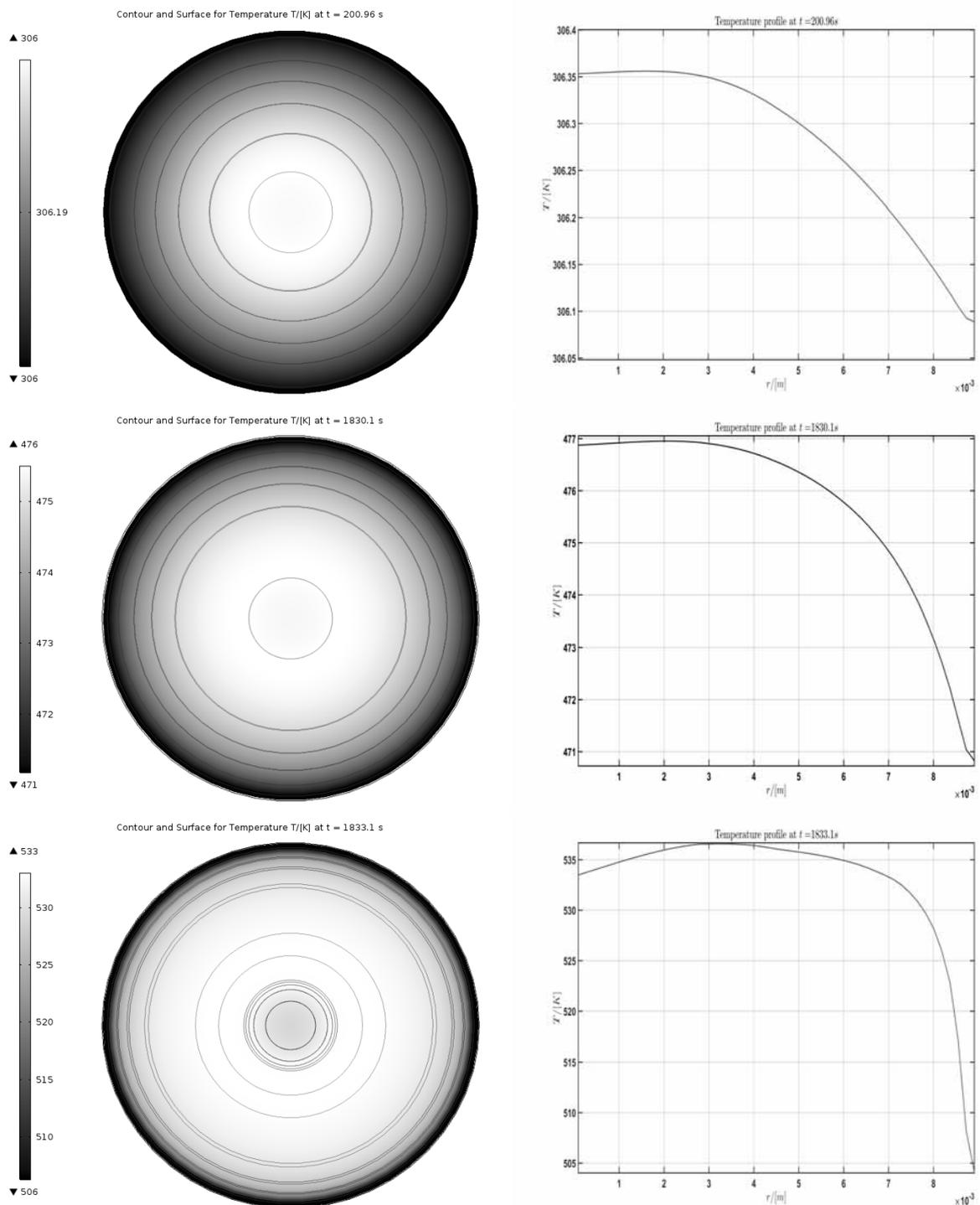


Abbildung 6: Zyklierungsversuch: Contour-Plot und zugehöriges radiales Temperaturprofil in der xy -Ebene bei $z = 0.00325$ m zu den Zeitpunkten $t = 200.96$ s, $t = 1830.1$ s und $t = 1833.1$ s

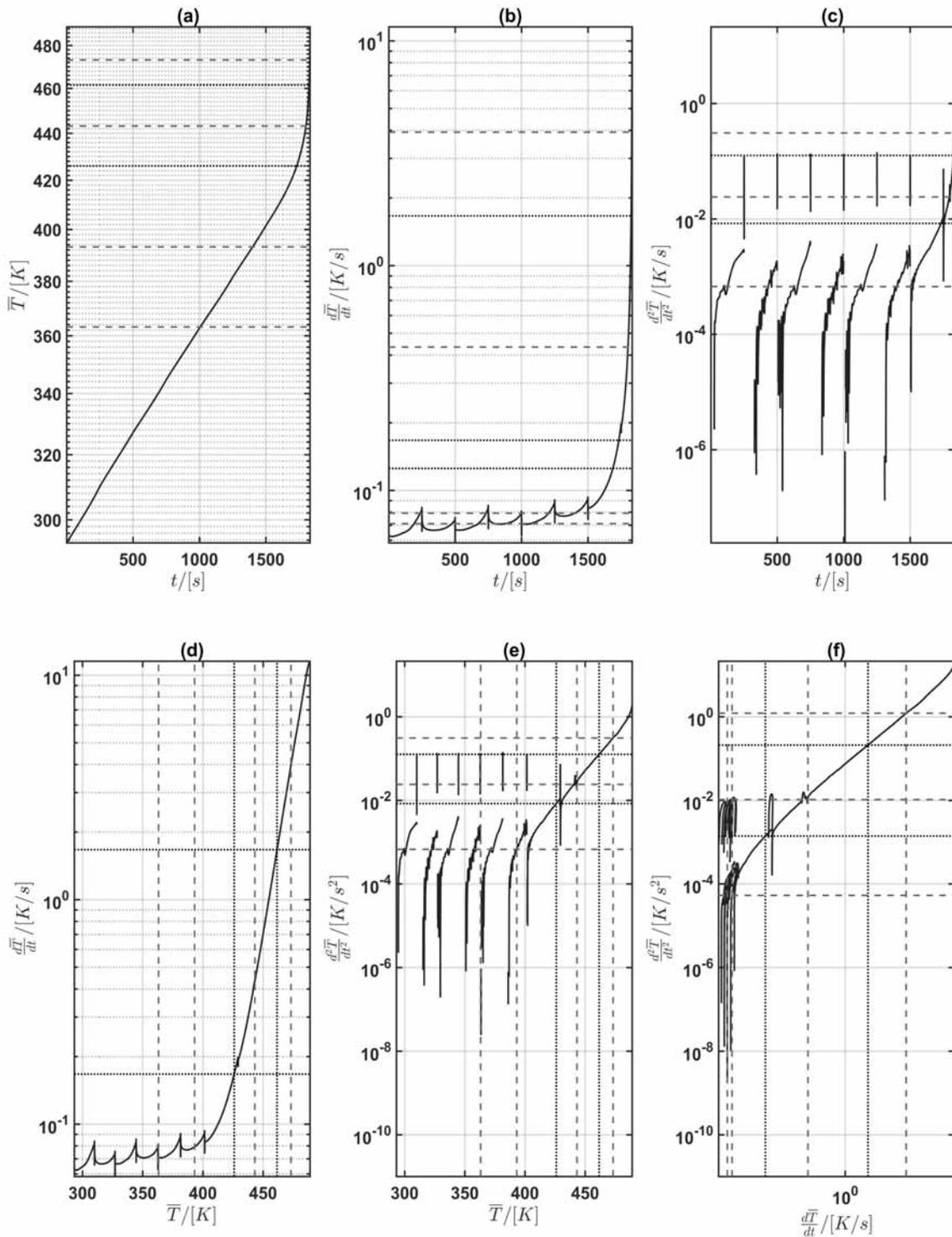


Abbildung 7: Zeitliche Entwicklung während des Zyklierungsversuches: (a)-(c) Zeitentwicklung der mittleren Zelltemperatur \bar{T} , zugehörige Heizrate $d\bar{T}/dt$ und Temperaturbeschleunigung $d^2\bar{T}/dt^2$. (d)-(e) Temperatur-Heizraten-, Temperatur-Temperaturbeschleunigungs-, Heizraten-Temperaturbeschleunigungs-Plot.

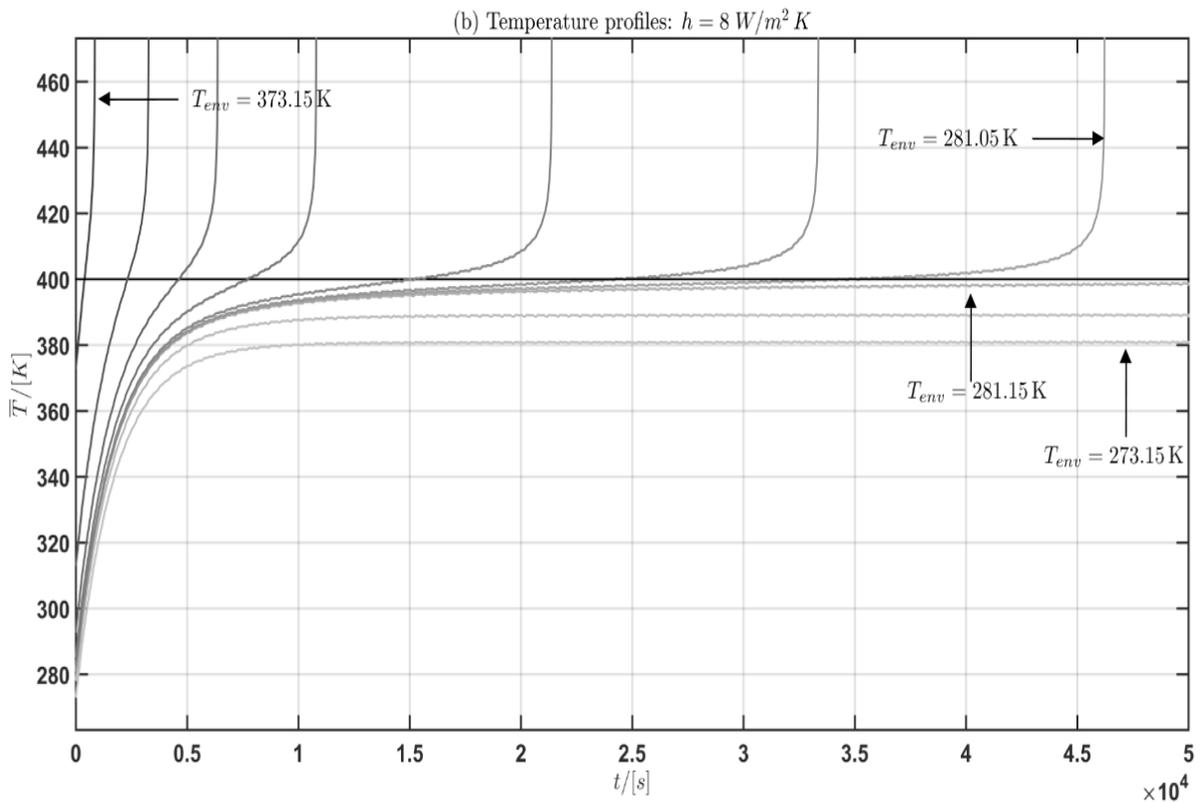
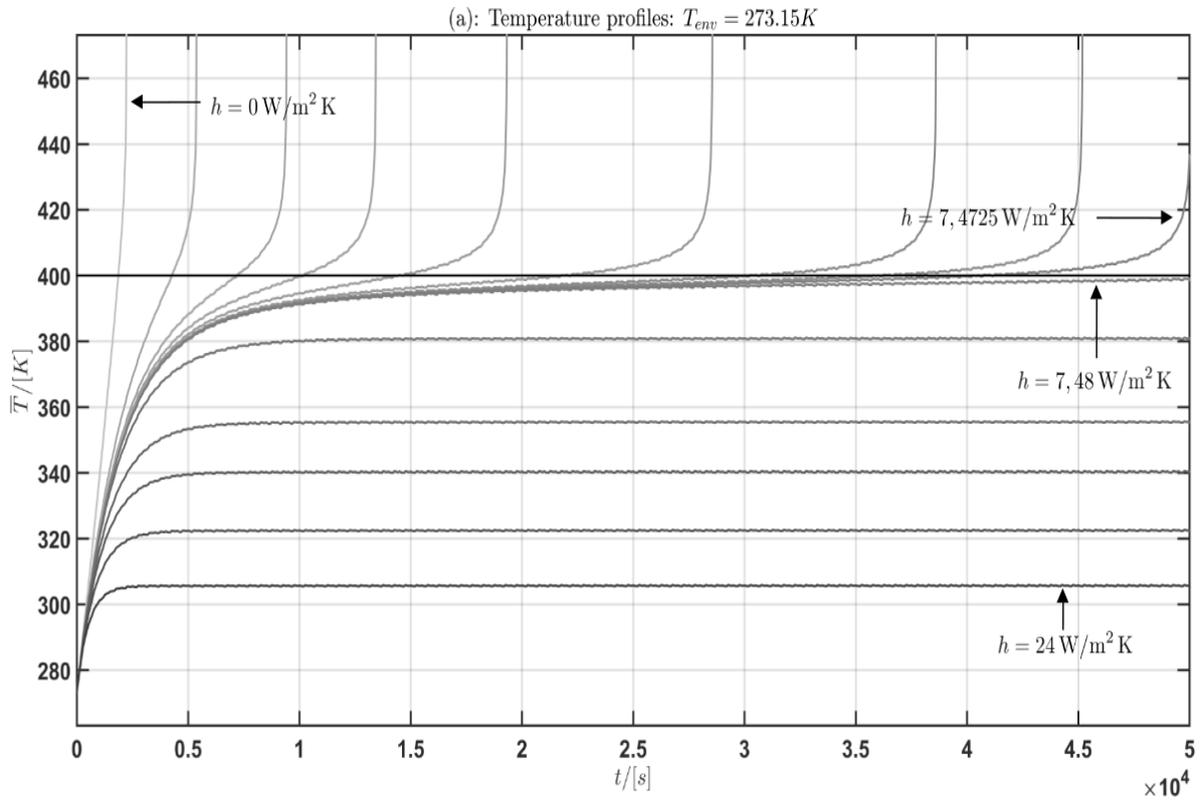


Abbildung 8: Temperaturprofil der mittleren Zelltemperatur \bar{T} :

(a) Umgebungstemperatur T_{env} fix, konvektiver Wärmetransferkoeffizient h variiert.

(b) Konvektiver Wärmetransferkoeffizient h fix, T_{env} variiert.

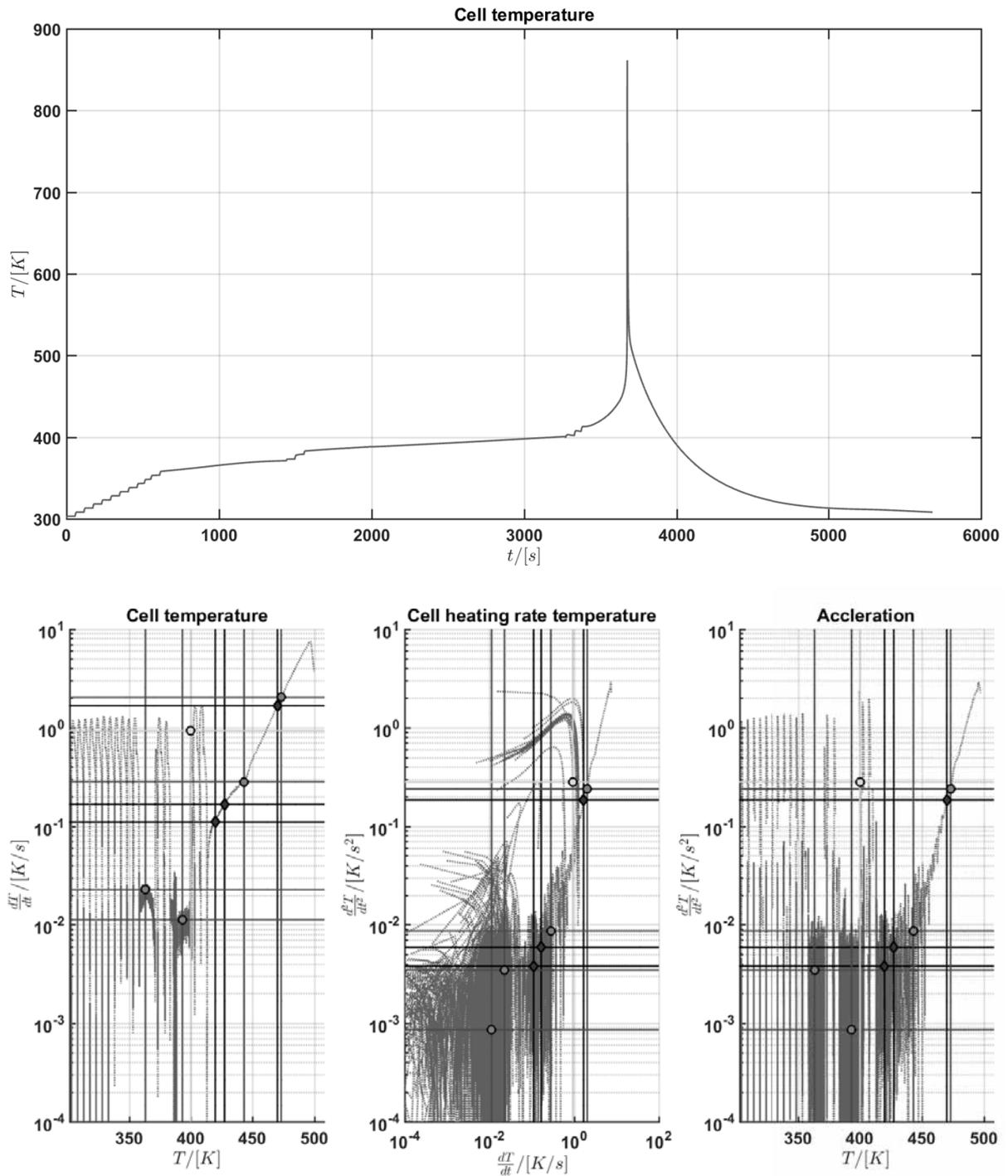


Abbildung 9: Experimenteller *Thermal Runaway* bei einem Ofen-Versuch

Literaturverzeichnis

- [1] Newman, J., Tiedemann, W., *Porous-Electrode Theory with Battery Applications*, AIChE Journal, 21, (1), 25-41, 1975.
- [2] Bebernes J., Eberly D., *Mathematical Problems from Combustion Theory*, Applied Mathematical Sciences Volume 83, Springer, 1989.
- [3] Volpert, V., *Elliptic Partial Differential Equations, Volume 2: Reactions-Diffusion Equations*, Birkhäuser, 2014.
- [4] Frank-Kamenetskij, D.A., *Diffusion And Heat Transfer in Chemical Kinetics*, Plenum Press, New York 1969
- [5] Abraham, D.P., Roth, E.P., Kostecky, R., McCarthy, K., MacLaren, S., Doughty, D.H. *Diagnostic examination of thermally abused high-power lithium-ion cells*, J. Power Sources, 161, 648-657, 2006.
- [6] Doyle, M., Fuller, T.F., Newman J., *Modeling Galvanostatic Charge and Discharge of the Lithium/Polymer/Insertion Cell*, J. Electrochem. Soc., 140 (6), 1526-1533, 1993.
- [7] Fuller, T.F., Doyle, M., Newman, J., *Simulation and Optimization of the Dual Lithium Ion Insertion Cell*, J. Electrochem. Soc., 141 (1), 1-10, 1994.
- [8] Doyle, M., Newman J., *Comparison of the Modeling Predictions with Experimental Data from Plastic Lithium Ion Cells*, J. Electrochem. Soc., 143 (6), 1890-1903, 1996 .
- [9] Newman, J., Tiedemann, W., *Temperature Rise in a Battery Module with Constant Heat Generation*, J. Electrochem. Soc., 142 (4), 1054-1057, 1995.
- [10] Pals, C.R., Newman, J., *Thermal modeling of Lithium/Polymer Battery, I. Discharge Behavior of a Single Cell*, J. Electrochem. Soc., 142 (10), 3274-3281, 1995.
- [11] Pals, C.R., Newman, J., *Thermal modeling of Lithium/Polymer Battery, II. Temperature Profiles in a Cell Stack*, J. Electrochem. Soc., 142 (10), 3282-3288, 1995
- [12] Al Hallaj S., Maleki, H., Hong, J.S., Selman, J.R., *Thermal modeling and design considerations of lithium-ion batteries*, J. Power Sources 83, 1-8, 1999.
- [13] Chen, S.C., Wan, C.C., Wang, Y.Y., *Thermal analysis of lithium-ion batteries*, J. Power Sources 140, 111-124, 2005.
- [14] Inui, Y., Kobayashi, Y., Watanabe, Y., Watase, Y., Kitamura, Y., *Simulation of temperature distribution in cylindrical and prismatic lithium ion secondary batteries*, Energy Conversion and Management, 48, 2103-2109, 2007.
- [15] Zhu, C., Li, X., Song, L., Xiang, L., *Development of the theoretically based thermal model for lithium ion battery pack*, J. Power Sources, 223, 155-164, 2013.
- [16] Hu, J., Tao, L., Jing, L., *Temperature Field Analysis and Thermal Dissipation Structure Optimization of Lithium-ion Battery Pack in PEVs*, Research Journal of Applied Sciences, Engineering and Technology, 7 (1), 62-71, 2014.
- [17] Wang, C.Y., Gu, W.B., Liaw, B.Y., *Micro-Macroscopic Coupled Modeling of Batteries and Fuel Cells. I. Model Development*, J. Electrochem. Soc., 145 (10), 3407-3417, 1998 .
- [18] Gu, W.B., Wang, C.Y., Liaw, B.Y., *Micro-Macroscopic Coupled Modeling of Batteries and Fuel Cells. II. Application to Nickel Cadmium and Nickel-Metal Hybrid Cells*, J. Electrochem. Soc., 145 (10), 3418-3427, 1998 .
- [19] Howes, F.A., Whitaker, S., *The Spatial Averaging Theorem Revisited*, Chemical Engineering Science, 40, (8), 1387-1392, 1985.
- [20] Latz, A., Zausch, J., Iliev, O., *Modeling of Species and Charge Transport in Li-Ion Batteries Based on Non-equilibrium Thermodynamics*, in Dimov, I., Dimova, S., Kolkovska, N. (Eds.): NMA 2010, LNCS 6046, 329-337, Springer 2011.
- [21] Richardson, G., Denault, G., Please, C.P., *Multiscale modeling and analysis of lithium-ion battery charge and discharge*, J. Eng. Math., 72, 41-72, 2012.
- [22] Landstorfer, M., Jacob, T., *Mathematical modeling of intercalation batteries at the cell level and beyond*, Chem. Soc. Rev., 42, 3234-3252, 2013.
- [23] Latz, A., Zausch, J., *Multiscale modeling of lithium ion batteries: thermal aspects*, Beilstein J. Nanotechnol., 6, 987-1007, 2015.

- [24] Gu, W.B., Wang, C.Y., *Thermal-Electrochemical Modeling of Battery Systems*, J. Electrochem. Soc., 147 (8), 2910-2922, 2000.
- [25] Wang, C.Y., Srinivasan, V., *Computational battery dynamics (CBD)-electrochemical/thermal coupled modeling and multi-scale modeling*, J. Power Sources, 110, 364-376, 2002.
- [26] Kumaresan, K., Sikha, G., White, R.E., *Thermal Model for Lion-Ion Cell*, J. Electrochem. Soc., 155 (2), A164-A171, 2008.
- [27] Cai, L., White, R.E., *Mathematical modeling of lithium ion battery with thermal effects in COMSOL Inc. Multiphysics (MP) software*, J. Power Sources, 196, 5985-5989, 2011.
- [28] Kim, G.H., Smith, K., Lee, K.J., Santhanagopalan, A.Pesaran, *Multi-Domain Modeling of Lithium-Ion Batteries Encompassing Multi-Physiscs in Varied Lenght Scales*, J. Electrochem. Soc., 158 (8), A955-A969, 2011.
- [29] Guo, M., Kim, G.H., White, R.E., *A three-dimensional multi-physics model for a Li-ion battery*, J. Power Sources, 240, 80-94, 2013.
- [30] Lee, K.J., Smith, K., Pesaran, A., Kim, G.H., *Three dimensional thermal-, electrical, and electrochemical-coupled model for cylindrical wound large format lithium-ion batteries*, J. Power Sources, 241, 20-32, 2013.
- [31] Latz, A., Zausch, J., *Thermal-Electrochemical Lithium-Ion Battery Simulations on Microstructure and Porous Electrode Scale*, ECS Transactions, 69, (1), 75-81, 2015.
- [32] Hatchard, T.D., MacNeil, D.D., Basu, A., Dahn, J.R., *Thermal Model of Cylindrical and Prismatic Lithium-Ion Cells*, J. Electrochem. Soc., 148 (7), A755-A761, 2001.
- [33] Spotnitz, R., Franklin, J., *buse behavior of high-power, lithium-ion cells*, J. Power Sources, 113, 81-100, 2003.
- [34] Spotnitz, R.M. Weaver, J., Yeduvaka, G., Doughty, D.H., Roth, E.P., *Simulation of abuse tolerance of lithium-ion battery packs*, J. Power Sources, 163, 2, 1 1080-1086, 2007.
- [35] Kim, G.H., Pesaran, A., Spotnitz, R., *A three-dimensional thermal abuse model for lithium-ion cells*, J. Power Sources, 170, 476-489, 2007.
- [36] Peng, P., Sun, Y., Jiang, F., *Thermal analyses of LiCoO₂ lithium-ion battery during oven tests*, Heat Mass Transfer, 50, 1405-1416, 2014
- [37] Peng, P., Sun, Y., Jiang, F., *Numerical simulations and thermal behavior analysis for oven thermal abusing of LiCoO₂ lithium-ion battery*, CIESC Journal, 65, 2, 647-657, 2014.
- [38] Lopez, C.F., Jeervarajan, J.A., Mukherjee, P.P., *Characterizarion of Lithium-Ion Battery Thermal Abuse Behavior Using Experimental and Computational Analysis*, J. Electrochem. Soc., 162 (10), A2163-A2173, 2015.
- [39] Chen, M., Sun, Q., Wu, K., Liu, B., Peng, P., Wang, Q., *A Thermal Runaway Simulation on a Lithium Titanate Battery and Battery Module*, Energies, 8, 490-500, 2015.
- [40] Tanaka, N., *Modeling and simulation of thermo-electrochemistry of thermal runaway in lithium-ion batteries*, PhD-Thesis, University of Stuttgart, 2015, <http://elib.uni-stuttgart.de/opus/volltexte/2015/10112/>, Last access: 18.10.2015.
- [41] Guo, G., Long, B., Cheng, B., Zhou, S., Cao, B., *Three-dimensional thermal finite element modeling of lithium-ion battery in thermal abuse application*, J. Power Sources 195, 2393-2398, 2010.
- [42] Chiu, K.-Ch., Lin, C.-H., Yeh, S.-F., Lin, Y.-H., Chen, K.-C., *An electrochemical modeling of lithium-ion battery nail penetration*, J. Power Sources, 251, 254-263, 2014.
- [43] Wang, Q., Ping, P., Sun, J., *Catastrophe analysis of cylindrical lithium ion battery*, Nonlinear Dyn., 61, 763-772, 2010.
- [44] Wang, Q., Ping, P., Zhao, X., Chu, G., Sun, J., Chen, C., *Thermal runaway caused fire and explosion of lithium ion battery*, J. Power Sources, 208, 210-224, 2012.
- [45] Lisbona, D., Snee, T., *A review of hazards associated with lithium and lithium-ion batteries*, Process Safety and Environmental Protection, 89, 434-442, 2011.
- [46] Semenov, N.N., *Zur Theorie des Verbrennungsprozesses*, Z.Phys., 48, 571, 1928.

Ein dynamisches Prozesssimulationsmodell für die energetische Betrachtung von Aluminium-Schmelzöfen in einer betriebsumfassenden Materialflusssimulation

Jörg Schmidt¹, Wolfgang Schlüter¹

¹Hochschule Ansbach

jo.schmidt@hs-ansbach.de

Ein weit verbreitetes Problem bei gasbetriebenen Schmelzöfen in der Druckgussindustrie ist eine ineffiziente, diskontinuierliche Betriebsweise, hervorgerufen durch Nachfrageschwankungen im laufenden Betrieb. Mit einer simulationsgestützten Optimierung soll eine produktivere und energieeffizientere Betriebsführung erreicht werden. Dazu wird ein dynamisches Simulationsmodell entwickelt, welches die Berechnung von Prozessparametern, wie Energieverbrauch oder Temperaturen im Schmelzofen, erlaubt. Das Ofenmodell teilt sich dabei in einen Bereich mit festem Aluminium (Schmelzschacht) und einen Bereich mit flüssigem Aluminium (Ofenwanne) auf. Neben einer Brennersteuerung werden im Modell auch eine Reihe von Verlustmechanismen berücksichtigt. Das so gebildete „Energiemodell“ des Schmelzofens wird mittels Messdaten validiert und in ein bestehendes Materialflussmodell integriert, um den Energieverbrauch im Kontext des Gesamtbetriebes zu erfassen. Die Validierung des Modelles weist eine gute Übereinstimmung zwischen Simulationsdaten und Messdaten aus dem laufenden Betrieb auf. Weiterhin wird eine Aussage über den Energieverbrauch bei verschiedenen Steuerungsszenarien des Materialflusses getroffen.

1 Einleitung

In der energieintensiven Industrie ist das Energieeinsparpotential besonders groß. Hier können schon prozentual kleine Einsparungen zu absolut gesehen großen Reduzierungen des Energieverbrauchs führen. Die Nicht-Eisen-Metalle sind neben der Chemie und der Stahlindustrie der dritte große umsatzstarke Bereich in der energieintensiven Industrie. In der vorliegenden Studie steht die Nicht-Eisen-Gussindustrie im Fokus.

In Deutschland werden jährlich 5,9 Mio. t Metall eingeschmolzen und vergossen, davon 1,1 Mio. t Nicht-Eisen-Werkstoffe. Geht man von dem Energieeinsatz von 2000 kWh pro t guter Guss aus, ergibt sich ein jährlicher Energieverbrauch von 2,2 Mrd. kWh in der Nicht-Eisen-Gussindustrie [1]. Nach einer offiziellen Auswertung des Statistischen Bundesamtes beträgt der Energiekostenanteil an der Bruttowertschöpfung der Gießerei-Industrie über 25 %. Dies zeigt, dass der Energieverbrauch für Gewinn und Wettbewerbsvorteile entscheidend ist und geeignete Maßnahmen für mehr Energieeffizienz in den in starkem internationalen Wettbewerb stehenden Unternehmen unabdingbar sind.

Das oberste Ziel eines Gussbetriebes ist die Sicherung der Produktion der Gussmaschinen durch ausreichende Versorgung mit flüssigem Metall. Überle-

gungen zur Energieeffizienz spielen aktuell bei der Steuerung der Flüssigmetallversorgung keine Rolle. Der Prozessschritt Schmelzen birgt jedoch ein hohes Einsparpotential (bis zu 25 %), da rund 50 % der in Druckgießereien eingesetzten Energie auf diesen zurückfällt [2]. Schritte wurden unternommen, um Schmelzöfen mit höherer Energieeffizienz zu produzieren [3] und die Effizienz von Gießereien zu erhöhen [4]. Nicht zuletzt aufgrund dieser Bedingungen und Maßnahmen hat sich die Energieeffizienz in der Nichteisenmetall-Produktion über die letzten beiden Jahrzehnte positiv entwickelt (rund 30 % weniger Energieeinsatz bei gesteigerter Produktion) [5].

Im Zuge dessen werden im Forschungsprojekt „Smart-Melting“, einem Teilprojekt des bayerischen Forschungsverbundes „Green Factory Bavaria“, neue Wege zur Steigerung der Energieeffizienz, in der Aluminium-Druckgussindustrie, untersucht. Probleme in diesem Zusammenhang sind u. a. eine ineffiziente Betriebsweise von gasbetriebenen Schmelzöfen, bedingt durch diskontinuierliche Nachfragefluktuationen von nachgeschalteten Druckgussmaschinen, sowie eine große Zahl von Parametern, welche den Prozess beeinflussen und deren Interdependenzen ein hohes Maß an Komplexität aufweisen. Solche Parameter sind z. B. Brennerstufe, Vorwärmtemperatur und die Art des verwendeten Aluminiummaterials.

Simulationsgestützte Untersuchungen auf diesem Gebiet liegen vor, allerdings beruhen diese i. d. R. nur auf CFD-Verfahren [6] [7] [8] [9] oder auf einer kleinen Größe des betrachteten Betriebs [1].

Ziel der vorliegenden Untersuchung ist die Entwicklung eines dynamischen Simulationsmodelles vom Schmelzofen, welches in eine betriebsumfassende Materialflusssimulation integriert ist und Daten bezüglich des Energieverbrauchs und weiteren Betriebsparametern berechnet. Mit diesen Erkenntnissen soll eine produktivere und energieeffizientere Betriebsführung erreicht werden.

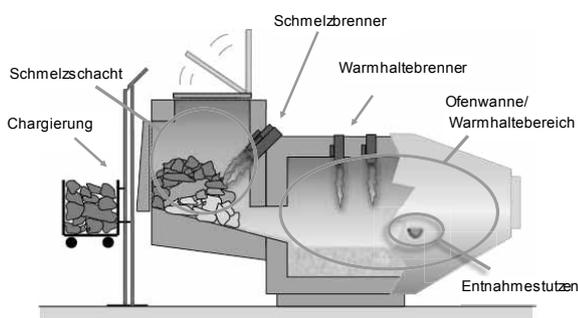


Abbildung 1. Schema Aluminium-Schmelzöfen

2 Aluminium-Schmelzöfen in der Druckgussindustrie

Diese Arbeit beschäftigt sich mit Aluminium-Schmelzöfen, welche genutzt werden, um das feste Aluminiummaterial zu erwärmen, zu schmelzen und auf einem bestimmten Temperaturniveau warmzuhalten. Die beteiligten Prozesse für den betreffenden Ofen zeigt Abbildung 1. Dabei wird der Ofen mit dem Aluminium in verschiedenen Materialformen beschickt, d. h. mit einem Behälter in den Schacht geladen (Chargierung). Im Schacht wird das beschickte Material durch Schmelzbrenner erwärmt und beginnt bei rund 600 °C zu Schmelzen (hier: Aluminiumlegierung namens EN AC-46000, auch bekannt als EN AC-AlSi9Cu3(Fe) oder Al226). Das flüssige Material fließt von der Schmelzkammer in die Ofenwanne, wo es mit Warmhaltebrennern auf einer Temperatur von rund 750 °C gehalten wird. Von dort wird das Flüssigmaterial mittels Entnahmestutzen und Kippen des Ofens entnommen und von Staplern zu den weiterverarbeitenden Druckgussmaschinen transportiert.

Beim vorliegenden Ofentyp handelt es sich um einen Schmelzofen mit kurzer Ausführung des Schachts.

Das heiße Abgas des Schmelzschachtes wird dabei über das Abgasrohr in der Ofenwanne abgezogen und dient somit der Erwärmung des Schmelzbades. Neben den erläuterten Prozessen Beschicken, Schmelzen, Warmhalten und Entnehmen finden auch die Prozesse Reinigen und Freischmelzen Anwendung. Beim Freischmelzen wird der Ofen mithilfe der Schmelzbrenner von Aluminium-Ablagerungen befreit und das verbleibende Schmelzgut im Schacht abgeschmolzen. Im Anschluss an das Freischmelzen wird der Ofen gereinigt. Bei diesem Vorgang werden die gelösten Verschmutzungen und Rückstände, die beim Einschmelzen von Ausschussmaterial zurückbleiben, aus dem Schmelzofen entfernt.

Der Gasverbrauch der Brenner wird beeinflusst durch eine Reihe von prozessbedingten und betriebsbedingten Einflüssen. Prozessbedingte Einflussfaktoren sind z. B. die Brennerführung und die Menge des beladenen Aluminiums im Schacht (Schachtfüllstand). Es existieren Zusammenhänge zwischen diesen Faktoren, welche teilweise im realen Betrieb unbekannt oder schwierig zu erfassen sind und mit der Simulation des Schmelzofens aufgezeigt werden sollen. Der Einfluss von betriebsbedingten Faktoren, wie Produktionsplan, Maschinenauslastung und Mitarbeiter, sind hingegen noch schwieriger zu erfassen. Die genannten Einflussfaktoren sorgen dafür, dass der spezifische Energieverbrauch im realen Betrieb deutlich vom Idealen abweicht. So fallen anstelle von ca. 600 kWh/t im idealen Betrieb, rund 800 bis 1300 kWh/t an. Eine Maßnahme zur Energieeinsparung ist z. B. die Anpassung des Schmelzbetrieb auf den Materialfluss [10]. Dies ist auch eines der Ziele, welche mit der Materialflusssimulation des Betriebes erreicht werden soll.

3 Dynamisches Simulationsmodell Schmelzöfen

3.1 Aufbau

Zur Umsetzung der Schmelzofensimulation dient ein dynamisches Simulationsmodell, basierend auf einer experimentellen Untersuchung [11]. Dabei werden mittels dynamischer Prozesssimulation (DPS) die Erwärmungs-, Schmelz- und Warmhalteprozesse des realen Ofens abgebildet (Abbildung 2). Informationen aus einer CAD-Zeichnung des Ofens, wie z. B. Oberflächen, werden in der DPS genutzt, um Wärmeverluste zu ermitteln. Eine CFD-Simulation wird genutzt, um Anhaltspunkte für die physikalischen zeit-

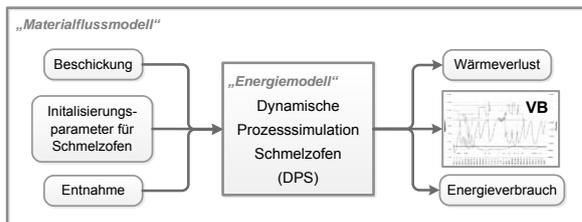


Abbildung 2. Schema Materialflussmodell

temperatur- und ortsabhängigen Variablen, wie bspw. den Wärmeübergangskoeffizienten, bzw. α -Wert (vom Rauchgas an das Aluminium), zu liefern und die Strömungsprozesse zu visualisieren. Vor Simulationsbeginn werden Initialisierungswerte vorgegeben, welche den Ofen, die Brennerdaten, das Aluminiummaterial und das Rauchgas betreffen. Diese Werte werden aus den theoretischen Eigenschaften, sowie praktischen Messungen am Ofen, angesetzt. Als Ergebnis liefert das Simulationsmodell eine Darstellung des Betriebsablaufes (VB) vom Ofen (Verlauf von Temperaturen, Massen und Gasverbrauch) und der Wärmeverluste.

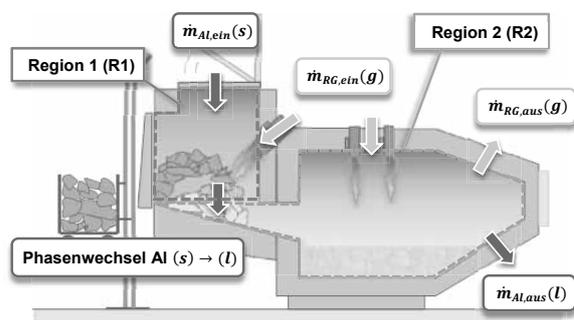


Abbildung 3. Gebietsaufteilung Ofenmodell

Das dynamische Simulationsmodell des Ofens, in diesem Kontext auch als „Energiemodell“ bezeichnet, ist mit einem „Materialflussmodell“ gekoppelt, welches den Materialfluss (Aluminium-Distribution) in der gesamten Gießerei erfasst. Neben logistischen Aspekten (Stapler für Transport von festem und flüssigem Aluminium) werden dabei auch die Aspekte der Druckgussmaschinen, wie Dosierofenspeicher, berücksichtigt. Die Schnittstellen der beiden Simulationsmodelle betreffen den Materialfluss (Beschickung und Entnahme). Dies bedeutet, dass der Energieverbrauch und die Betriebsablaufdaten der Ofen getrennt im Energiemodell berechnet werden. Mit diesem Simulationsmodell lassen sich Energieverbrauchsdaten, bei verschiedenen Steuerungsstrategien des Materialflusses, bestimmen.

3.2 Physikalisches Modell

Um den Energie- und Materialfluss zu berechnen, wird der Ofen in zwei Gebiete aufgeteilt (Abbildung 3). Region 1 (R1) umfasst den Schmelzschacht (kurz: SS) mit den Phasen für Festmaterial ($s = \text{solid}$) und dem Rauchgas der Schmelzbrenner ($g = \text{gaseous}$). Region 2 (R2) dagegen umfasst die Ofenwanne (kurz: OW) mit den Phasen für flüssiges Aluminium ($l = \text{liquid}$) und dem Rauchgas der Warmhaltebrenner (g). Zwischen den beiden Regionen findet der Phasenwechsel des Aluminiums, vom festen in den flüssigen Zustand, statt. Der Materialfluss wird dabei durch die in das System ein- und austretenden Massenströme \dot{m} des Rauchgases (Index RG) und des Aluminiums (Index Al) beschrieben. In R1 beschreibt $\dot{m}_{Al, ein}$ das durch die Chargiereinrichtung zugeführte Aluminium-Festmaterial und $\dot{m}_{Al, aus}$ das in einer Zeiteinheit geschmolzene Aluminium, was auch als „Schmelzleistung“ bezeichnet wird. Die Gleichung für den Materialfluss \dot{m}_{Al} des Aluminiums in R1 ergibt sich aus der Massenerhaltung nach (1).

$$\frac{dm_{Al}}{dt} = \dot{m}_{Al} = \dot{m}_{Al, ein} - \dot{m}_{Al, aus} \quad (1)$$

Die weiteren Materialflüsse für das RG und R2 sind analog zu (1).

Für das physikalische Modell wurden folgende Annahmen und Vereinfachungen getroffen:

- einheitliche Temperaturverteilung im Schmelzbad, aufgrund hoher Wärmeleitfähigkeit
- vereinfachte Temperatur- und Materialverteilung des beschickten Aluminiums.

Für die energetische Modellierung von R1 wird der Schmelzschacht als Gegenstromwärmeübertrager betrachtet, wie in einem vergleichbaren Ansatz [12]. In diesem Modellansatz wird das beschickte Aluminium durch das aufströmende Rauchgas erwärmt und entspricht damit dem Wirkungsprinzip des realen Schmelzschachts (Abbildung 4). Dabei kommt die

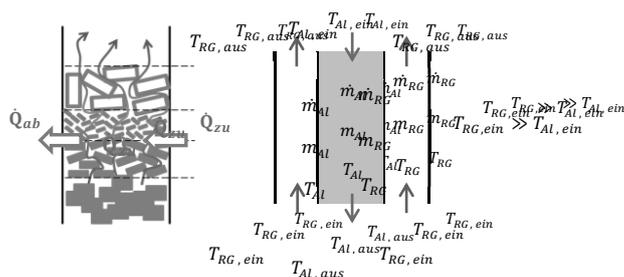


Abbildung 4. Modell Schmelzschacht

mittlere logarithmische Temperaturdifferenz zum Tragen, welche eine Verbindung zwischen den Temperaturen des Aluminiums und des Rauchgases herstellt. Dazu wird ein mittleres Temperaturprofil über den Schacht gebildet.

Mithilfe der Formel für die Änderung der inneren Energie U eines geschlossenen Systems, bzw. einer Form des 1. Hauptsatzes der Thermodynamik, lassen sich die jeweiligen Temperaturänderungen (Al, RG) am Austritt berechnen (2).

$$\frac{dU}{dt} = \dot{Q}_{zu} - \dot{Q}_{ab} = \frac{d(c_p \cdot m \cdot T)}{dt} \quad (2)$$

\dot{Q}_{zu} und \dot{Q}_{ab} entsprechen dabei der zu- und abgeführten Wärmeleistung im Schmelzschacht. Die Zuführung der Wärme erfolgt durch die Brenner, die Abführung durch Wärmeübertragung an das Aluminium, Verluste über die Ofenwand oder Wärmestrahlung. Der Parameter T entspricht der Temperatur, c_p der spezifischen Wärmekapazität.

Der Modellierungsansatz für die Ofenwanne, bzw. den Warmhaltebereich unterscheidet sich zu der des Schmelzschachtes (Abbildung 5). Im Gegensatz zu den Strömungsbedingungen im Schmelzschacht, in welchem die Aluminiumschüttung vom Rauchgas durchströmt wird, erwärmt das Rauchgas in der Ofenwanne lediglich die offenliegende Oberfläche des Schmelzbades (einfache Temperaturdifferenz anstelle logarithmischer Temperaturverteilung).

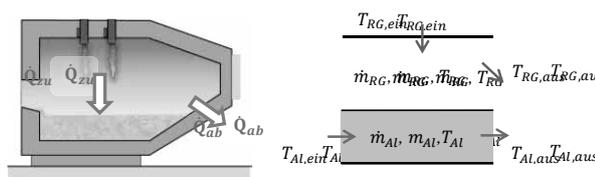


Abbildung 5. Modell Ofenwanne

3.3 Brennersteuerung

Die realen Öfen sind mit einer Reihe von Steuerungsmechanismen für die Brenner ausgestattet. In Anlehnung an die realen Steuerungsmechanismen wurden Algorithmen in der Simulation implementiert, welche eine Regelung der Brenner erlauben.

Die Steuerungsmechanismen für das Abschalten der Schmelzbrenner (kurz: SB) sind nachfolgend beschrieben. Wenn eine oder mehrere der folgenden Kriterien erfüllt sind, werden die SB abgeschaltet:

- maximale Abgastemperatur am Schachtaustritt erreicht
- Schmelzschacht leer
- Ofenwanne fast voll (Überlaufen soll verhindert werden).

Die Brenner werden eingeschaltet, wenn alle nachfolgend aufgeführten Kriterien zutreffen:

- maximale Abgastemperatur am Schachtaustritt unterschritten
- Schmelzschacht gefüllt
- Ofenwanne nicht voll.

Um die strömungsmechanischen Gegebenheiten der Realität besser abbilden zu können, wurde ein Übergangsbereich zwischen An- und Abschaltvorgang implementiert, in dem der Rauchgasmassenstrom nicht sprunghaft sondern allmählich abnimmt.

Wie auch die Schmelzbrenner, sind die Warmhaltebrenner (kurz: WHB) mit einer Steuerung ausgestattet. Für das An- und Abschalten der Warmhaltebrenner wird in diesem Fall nur ein Steuerungsmechanismus verwendet. Dabei werden die Brenner automatisch ausgeschaltet, wenn die maximale Schmelzbadtemperatur überschritten wird, um ein übermäßiges Überhitzen der Schmelze zu vermeiden. Beim Unterschreiten der Grenztemperatur schalten die WHB wiederum ein.

3.4 Oberflächen und Volumina

Die Kenntnis von Oberflächen und Volumina in Bezug auf den Ofen und dessen Komponenten ist wichtig, um Größenordnungen der wärmeübertragenden Flächen zu bestimmen. Die Flächen betreffen dabei die Ofenwände (Wärmeverluste) und die wärmetauschenden Oberflächen des festen Aluminiummaterials im Schacht, sowie des flüssigen Aluminiums in der Ofenwanne.

Für das Flüssigaluminium in der Ofenwanne wurde die Annahme getroffen, dass nur die Oberfläche des „Schmelzbades“ als wärmeübertragende Fläche dient. Die Oberfläche ist aufgrund der Ofenwannenform (ellipsenähnlich, mit einer konischen Verjüngung am Ende) stark abhängig vom Füllstand der Ofenwanne. Abbildung 6 zeigt eine Kennlinie der Oberfläche in Abhängigkeit der Badmasse. Der Graph wurde bei einer durchschnittlichen Schmelzbadtemperatur von 700 °C und einer Dichte von 2350 kg/m³ aufgenommen. Für abweichende Temperaturen hat die Dichte-

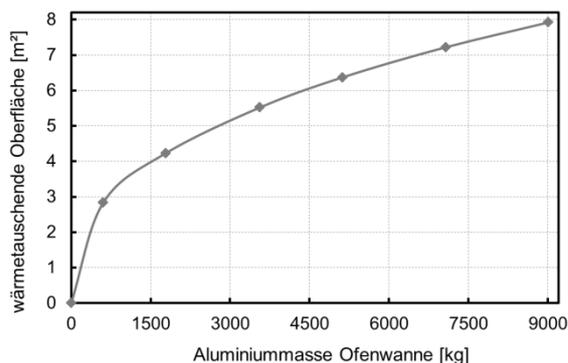


Abbildung 6. Kennlinie Schmelzbadoberfläche in Abhängigkeit der Badmasse

änderung nur einen geringen Einfluss (Fehler < 1 %) und ist damit ebenfalls anwendbar.

Beim festen Aluminiummaterial spielt die Form des beschickten Materials eine tragende Rolle. Das Aluminium wird dabei in Form von

- Ausschussmaterial (gegossenes Teil, welches durch Qualitätskontrolle gefallen ist)
- Blockmaterial (Masseln)
- Rücklaufmaterial (Restmaterial der Druckgussmaschinen)

zugeführt. Grundsätzlich gilt: Je höher die spezifische Oberfläche der Materialform, desto höher die wärmeübertragende Fläche und desto geringer die Einschmelzzeit (und damit auch geringerer Energieaufwand). Die spez. Oberflächen variieren von 0,0148 m²/kg (Masseln) bis 0,068 m²/kg (Ausschuss).

4 Validierung Simulationsmodell

Um die praxistaugliche Gültigkeit des Simulationsmodelles zu überprüfen, wurde eine Validierung des Simulationsmodelles, mit Messungen aus dem laufenden Betrieb, durchgeführt. Bei der Validierung dienen die Messdaten der Beschickung und Entnahme als Simulationsinput. Die gemessenen Daten

- Badmasse (Parameter $m_{Al,OW}$)
- Abgastemperatur Schacht ($T_{RG,aus}$)
- Abgastemperatur Baddecke (T_{RG})
- Schmelzbadtemperatur (T_{OW})
- Gasverbrauch (V_{EG})

dienen dagegen zum Vergleich mit den simulierten Daten. Da durch den großen Beschickungsbehälter

beim vorliegenden Ofentyp Mischungen der Materialform entstehen können, werden mittlere, spezifische Oberflächen berechnet und in den Beschickungsdaten integriert. Die Messdaten für die Aluminiummasse im Schmelzschacht liegen nicht in direkter Form vor, sondern leiten sich aus der Differenz des Ofengewichts (Gesamtgewicht Aluminium im Ofen) und der Schmelzbadmasse ab. Eine Schwierigkeit der Differenzenbildung stellt die unterschiedliche, zeitliche Auflösung der Messwerte von Ofengewicht und Badmasse dar. Dieses Problem wird mittels Erstellung eines gemeinsamen Zeitrasters gelöst, bei welchem Zwischenwerte interpoliert werden.

Zur Minimierung der Abweichungen zwischen Simulations- und Messdaten, wurden Parameter in einer iterativen Vorgehensweise angepasst. Dies betrifft u. a. die Wärmeübergangskoeffizienten, welche aufgrund der vorliegenden Strömungsverhältnisse stark variieren.

Um die Validierung quantitativ zu bewerten, wurden diverse Kennwerte berechnet. Die Messungen beziehen sich dabei auf 3 Messreihen mit einer Messdauer von rund 6 h. Die Ergebnisse der Validierung zeigt Tabelle 1. Die dargestellte Abweichung bezieht sich auf die mittlere Abweichung der Simulation zur Messung. Es ist zu erkennen, dass mit dem Simulationsmodell Abweichungen kleiner 20 % erreicht werden können. Über alle Abweichungen gemittelt ergibt sich eine geringere Abweichung von 7,5 %, d. h. es besteht eine gute Übereinstimmung zwischen simulierten und gemessenen Werten. Auf die Abweichungen im Einzelnen wird im Folgenden näher eingegangen.

Parameter	Abweichung zur Messung
Masse Aluminium Ofenwanne	3,5 %
Abgastemperatur Schmelzschacht	19,7 %
Temperatur Schmelzbad	2,2 %
Temperatur Baddecke	7,0 %
Gasverbrauch	7,8 %
Schmelzleistung	4,6 %

Tabelle 1. Ergebnisse Validierung

Die Aluminiummasse des Schmelzschachtes (nach Messreihe 2) ist in Abbildung 7 dargestellt. Die Gegenüberstellung der Kurven zeigt, dass der Verlauf

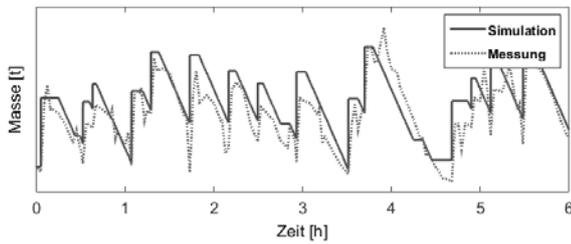


Abbildung 7. Verlauf Aluminium Schmelzschacht

der Messung gut nachempfunden werden kann. Sprünge im Verlauf deuten auf Beschickungen hin, lineare Abnahmen dagegen auf ein Abschmelzen des Aluminiums und damit Übergehen in die Ofenwanne. Die Steigungen und Endwerte, welche besonders abhängig von der maximalen Schmelzleistung, der Vorwärmtemperatur und dem Wärmeübergangskoeffizienten des Schmelzschachtes sind, verlaufen dabei relativ nahe den Messungen. Durch diese Faktoren wird auch der Zeitpunkt beeinflusst, wann die Schmelztemperatur erreicht ist und der Schmelzvorgang beginnen kann. Bedingt durch das grobe Modell des Schmelzvorganges kommt es bei der Simulation zu einer Abflachung der Spitzen im Vergleich mit den Messungen. Auch die Messungen unterliegen Ungenauigkeiten, wodurch weitere Abweichungen entstehen.

Aufgrund der beschriebenen Ableitung der Aluminiummasse im Schmelzschacht, aus der Masse in der Ofenwanne, zeigt sich ein ähnlicher Verlauf (Abbildung 8) wie in Abbildung 7. Da die Entnahmen im realen Fall nicht sprunghaft stattfinden, sondern durch das allmähliche Entnehmen einen stetigen Verlauf beim Kippvorgang haben, kommt es zu Abweichungen bei den Verläufen. Die Steigungen bei den Zunahmen des Ofengewichts, durch das Zufließen des Aluminiums aus dem Schacht, liegen aber nahe an den gemessenen Werten.

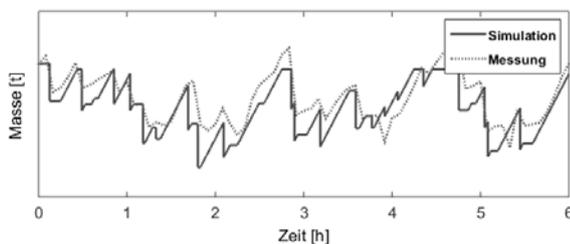


Abbildung 8. Verlauf Aluminium Ofenwanne

Der berechnete Gasverbrauch setzt sich zusammen aus dem Verbrauch der Schmelz- und Warmhaltebrenner. Zu Abweichungen kommt es hier durch die

in Abschnitt 3.3 erwähnten Steuerungsmechanismen der Brenner, d. h. es liegt ein abweichendes An- und Abschaltverhalten vor, bedingt durch den simulierten Prozessverlauf. Trotz vieler Interdependenzen erreicht man mit einer Abweichung von 7,8 % aber eine gute Näherung der realen Verhältnisse.

Höhere Abweichungen treten dagegen bei der Abgastemperatur im Schmelzschacht auf (rund 20 %). Diese Abweichung entsteht u. a. durch die physikalische Modellierung der Schmelzschachts als Gegenstromwärmeübertrager (vgl. Abschnitt 3.2). Problematisch ist das „kalte Ende“ des Wärmeübertragers (Annahme, dass Aluminium auf der Eingangsseite eine konstante Temperatur besitzt). Somit wird dauerhaft eine höhere Wärmemenge an das Aluminium übertragen, was zu einer höheren Abkühlung der Abgastemperatur führt. Weitere Abweichungen ergeben sich durch die Steuerung der Schmelzbrenner und eine grobe Abbildung der strömungstechnischen Gegebenheiten, z. B. bei der Wärmeübertragung. Ein Optimierungsansatz ist die Einführung von einer Reihe von Blockkapazitäten, mit unterschiedlichen Temperaturschichten, anstelle eines Blockes mit einer logarithmischen Temperaturverteilung.

Die Temperaturen in der Ofenwanne (Schmelzbad rund 2 %, Baddecke 7 %) können im Vergleich mit dem Schmelzschacht genauer abgebildet werden. Ein Grund dafür ist, dass ein einfacherer Modellansatz verwendet wird (vgl. Punkt 3.2). Weiterhin treten bzgl. des Schmelzbadtemperatur-Verlaufs keine hohen Temperaturschwankungen auf, was den trägen Erwärmungsvorgängen entgegenkommt.

Das gebildete Energiemodell ist aufgrund seiner modularen Struktur gut geeignet, um spezifische Verbesserungen durchzuführen und das System weiter der Realität anzupassen. Allerdings muss das Aufwand/Nutzen-Verhältnis berücksichtigt werden, mit dem Anpassungen durchgeführt werden sollen.

5 Ergebnisse und Potenziale

Das validierte Modell des Schmelzofens wird in die Materialflusssimulation integriert, um den Aluminiuminhalt der Ofenwanne darstellen und den Energieverbrauch, bei verschiedenen Steuerungsstrategien des Materialflusses, berechnen zu können.

Eine Validierung der betriebsumfassenden Materialflusssimulation wurde durchgeführt, um die beiden Modelle mit den realen Gegebenheiten zu vergleichen. Als Datensatz dienen Messungen einer Refe-

renzwoche (17 Schichten à 8 Stunden). Zur Überprüfung der Energiemodell-Funktion spielt der spezifische Energieverbrauch \dot{q}_{Al} eine besondere Rolle. Dieser berechnet sich nach (3) und setzt sich zusammen aus den Komponenten Gasverbrauch V_{EG} , Heizwert H_i des Brennstoffes und insgesamt geschmolzener Aluminiummasse $m_{Al,aus}$.

$$\dot{q}_{Al} = \frac{V_{EG} \cdot H_i}{m_{Al,aus}} \quad (3)$$

Mit einer Abweichung von 5,7 % der Simulation (1.039 kWh/t) zur Messung (983 kWh/t) kann die Simulation den realen Energieverbrauch sehr nahe wiedergeben.

Mittels Variation von Steuerungsstrategien des Materialflusses, soll der Produktionsablauf effizienter gestaltet und der Energieverbrauch gesenkt werden. Durch eine Optimierung des Betriebslaufes können z. B. die Fehlzeiten von Druckgussmaschinen (ungeplante Stillstände) reduziert werden. Dadurch leitet sich eine höhere Auslastung der Schmelzöfen ab und es wird ein kontinuierlicherer, effizienterer Betrieb erreicht. Berechnungen mit dem Simulationsmodell ergeben eine Reduzierung von bis zu 10 % bzgl. des spezifischen Energieverbrauchs. Allerdings erweist sich eine Abbildung der realen Ausfallzeiten von Druckgussmaschinen und Schmelzöfen, aufgrund von zufällig eintretenden Ereignissen, schwierig.

6 Zusammenfassung

Die vorgestellte Untersuchung befasst sich mit dem Erreichen einer produktiveren und energieeffizienteren Betriebsführung von Schmelzöfen in der Nicht-Eisen-Gussindustrie. Dazu wird ein dynamisches Simulationsmodell des Schmelzofens erstellt und in eine betriebsumfassende Materialflusssimulation integriert, um bspw. den Energieverbrauch bei verschiedenen Steuerungsstrategien zu berechnen.

Die Validierung des sogenannten Energiemodelles zeigt eine gute Übereinstimmung mit den Messwerten (im Mittel eine Abweichung von 7,5 % der betrachteten Werte). Durch spezifische Verbesserungen kann das Modell weiter optimiert werden.

Die Validierung des Materialflussmodells, mit integriertem Energiemodell, weist ebenfalls eine gute Näherung an die realen Verhältnisse auf (5,7 % Abweichung). Somit kann das Modell für weitere Untersuchungen genutzt werden.

Zukünftig sollen weitere Wärmeverlustmechanismen berücksichtigt und Steuerungsstrategien für den Schmelzofen generiert werden. Durch den Einsatz von CFD-Simulation sollen die Strömungsverhältnisse visualisiert und Aussagen, in Bezug auf die konvektive Wärmeübertragung im Ofen, getroffen werden. Weiterhin können verschiedene Erwärmungsszenarien des Aluminiums im Schmelzschacht und der Ofenwanne mittels CFD simuliert werden (z. B. Variation der Beschickung oder Rauchgaswerte). Dabei können z. B. Wärmeübergangskoeffizienten generiert und in das Energiemodell integriert werden.

7 References

- [1] C. Herrmann, H. Pries und G. Hartmann. *Energie- und ressourceneffiziente Produktion von Aluminiumdruckguss*. Springer Vieweg, Berlin, 2013.
- [2] Oskar Frech GmbH + Co. KG. *Den richtigen Hebel finden – Verbraucheranalyse in der Druckgießerei*. Schorndorf-Weiler, 2015.
- [3] R. Riedel und R. Hillen. *Energiesparwunder. Neue Effizienzklasse bei Aluminium-Schmelzöfen*. Giesserei-Rundschau, S. 18–19, 2012.
- [4] A. Dentel, W. Schlüter, A. Ringleb, D. Schneider, W. Stephan und G. Pröbstle. *Rationeller Energieeinsatz in der aluminiumverarbeitenden Industrie. Abwärmenutzung und verbesserte Isolierung von Transportbehältern für flüssiges Aluminium*. In: FORETA: Ergebnisse des Forschungsverbundes "Energieeffiziente Technologien und Anwendungen", Herausgeber: W. A. Mayer, Attenkofer, Straubing, S. C1-C24, 2013.
- [5] Wirtschaftsvereinigung Metalle. *Metallstatistik 2014*. Berlin, 2015.
- [6] T. Wittenschläger, D. Degen, V. Uhlig, D. Trimis, T. Reimann, K. Eigenfeld, Z. Mohammadifard, T. Vieregge und B.-A. Behrens. *Optimierung eines Aluminiumschmelzofens mittels numerischer Simulationen*. Gas Wärme International, S. 91–96, 2013.
- [7] N.-j. Zhou, S.-h. Zhou, J.-q. Zhang und Q.-l. Pan. *Numerical simulation of aluminum holding furnace with fluid-solid coupled heat transfer*. J. Cent. South Univ. Technol, S. 1389–1394, 2010.

- [8] M. Brune, A. Giese und D. Mackenstedt. *Optimierung der thermischen Prozessführung von gasbeheizten Schmelzöfen zur Produktion von Sekundäraluminium*. Gaswärme-Institut, Essen, 2003.
- [9] A. O. Nieckele, M. F. Naccache und M. Gomes. *Combustion performance of an aluminum melting furnace operating with liquid fuel*. J. Braz. Soc. Mech. Sci. & Eng, S. 437–444, 2010.
- [10] A. Pithan. *Betrachtungen zum Energieverbrauch in einem Aluminium-Schmelzwerk*. Euroguss, Nürnberg, 2014.
- [11] J. Schmidt, W. Schlüter und A. Ringleb. *Kopplung von Prozess- und CFD-Simulation zur Untersuchung des Wärmeverlusts beim Flüssigaluminiumtransport*. Tagungsband Konferenz ASIM/GI-Fachgruppen: Simulation technischer Systeme, in Stralsund, S. 209–217, 2015.
- [12] H. E. Scherf. *Modellbildung und Simulation dynamischer Systeme: Eine Sammlung von Simulink-Beispielen*. Oldenbourg, München, 2004.

Parametrisierung von Lithium-Ionen-Zellen auf Basis von CIT-Messungen

Matthias Gulbins¹, Andreas Melcher², Carlos Ziebert², Boxia Lei², Thomas Markwirth¹, Joachim Haase¹

¹Fraunhofer-IIS – Institutsteil Entwurfsautomatisierung EAS, Zeunerstr. 38, 01069 Dresden

²Karlsruher Institut für Technologie (KIT) – IAM-AWP, Hermann-von-Helmholtz-Platz 1,

76344 Eggenstein-Leopoldshafen

matthias.gulbins@eas.iis.fraunhofer.de

Für eine Simulationsumgebung zur Entwicklung von Batteriemanagementsystemen (BMS) sind Modelle für Li-Ion-Batteriezellen entwickelt und erprobt worden. Die Parametrisierung der Zellen erfolgte auf der Grundlage von CIT- (Current Interruption Technique) Messungen bei unterschiedlichen Temperaturen. Die Modelle wurden in der Modellierungssprache SystemC AMS unter Verwendung von Lookup-Tabellen implementiert. Eine Validierung erfolgte mit einem Stromprofil für ein Elektrofahrzeug auf Basis eines NEFZ-Geschwindigkeitsprofils.

1 Einleitung

Im Vorhaben IKEBA „Integrierte Komponenten und integrierter Entwurf energie-effizienter Batteriemanagementsysteme“ [1, 2] ist eine Simulationsumgebung zur Entwicklung von Batteriemanagementsystemen erarbeitet worden. Sie ermöglicht die Analyse der Wechselwirkungen von Batteriezellen, integrierten Schaltkreisen zur Überwachung von Zellspannungen und –temperaturen, Ladungsausgleichschaltungen und der Batteriemanagementsystemsoftware für unterschiedliche Lastprofile. Dabei können sowohl der Nominalfall untersucht als auch das fehlerhafte Verhalten einzelner Komponenten berücksichtigt werden. Als Modellierungssprachen werden SystemC [3] und SystemC AMS [4, 5] verwendet. Ziel ist es dabei, durch eine hohe Simulationsgeschwindigkeit eine möglichst große Anzahl von Testfällen untersuchen zu können.

Eine zentrale Rolle spielt in diesem Zusammenhang die Bereitstellung genauer und schnell auszuwertender Modelle der Batteriezellen. Die in diesem Zusammenhang verfolgten Ansätze sollen im Vortrag vorgestellt werden.

2 Modell für die Li-Ionen-Zelle

2.1 Anforderungen an das Batteriezellmodell

Mit den Modellen der Batteriezellen werden Batteriepacks modelliert, die Teil der Beschreibung von Batteriemanagementsystemen für die Entwurfsumgebung sind. Für Lösungen im Hochvoltbereich, die für elektrische Antriebe verwendet werden, ist eine große Anzahl von Zellen in der Simulation zu berücksichtigen. Das erfordert einerseits genaue, andererseits aber

auch schnell auswertbare Modelle. Basis für die Modellbildung ist das in Bild 1 dargestellte Ersatzschaltbild, bei dem die Abhängigkeit der Komponenten von Ladezustand SoC und Temperatur T berücksichtigt wird und das im aktuellen Zeitschritt bei der Simulation durch ein lineares Differentialgleichungssystem beschrieben wird, weil vereinfachend davon ausgegangen wird, die Werte von SoC und T aus dem jeweils letzten Simulationszeitschritt zu verwenden. Ein zweiter Aspekt ist die einfache Parametrisierbarkeit der Modelle. Diese wurde auf der Basis der CIT-Methode (Current Interruption Technique) bei jeweils konstanter Temperatur durchgeführt. Aus der Vorgabe von kurzen Stromimpulsen bei Ladung und Entladung der Batteriezelle ergibt sich der SoC. Aus den Spannungsverläufen an den Klemmen der Batteriezelle können dann die Modellparameter für bekannte Werte von SoC und T ermittelt werden. Diese Werte bilden dann die Grundlage für die erstellten Tabellenmodelle. Erweitert wurde dieses Vorgehen durch einen zweiten Schritt, bei der die den Tabellen zugrundeliegenden Werte nachträglich auf der Basis von Simulationsexperimenten korrigiert werden. Dadurch wird bei gleichem Charakterisierungsaufwand und gleicher Komplexität des Modells eine verbesserte Genauigkeit erreicht.

Selbsterwärmungseffekte und die thermische Kopplung zwischen Zellen in einem Batteriepack in Abhängigkeit von der Umgebungstemperatur sind zu berücksichtigen.

Weitere Anforderungen an Modelle von Batteriezellen sind die Berücksichtigung von Alterungseffekten und Fertigungsschwankungen. Beide Anforderungen

können im vorgestellten Ansatz durch Änderungen der Kapazität der Batteriezellen berücksichtigt werden. Da bedarf es aber sicherlich weiterer Untersuchungen zur Verbesserung der Modellqualität. Wünschenswert ist es auch fehlerhaftes Verhalten in die Modellierung einzubeziehen. Dazu gibt es die Möglichkeit der gezielten Injektion von Fehlern, wodurch Batterieparameter definiert geändert werden. Dieses Vorgehen gestattet es, die Maßnahmen zum Erkennen und zur Reaktion auf Fehler durch die Batteriemangement-Software im Rahmen der Entwurfsplattform zu untersuchen. Nicht einbezogen sind gegenwärtig in das Modell kompliziertere Fehlerfälle, die sich durch „schleichende“ Änderungen von Eigenschaften ergeben, die dann plötzlich zum Ausfall einzelner Zellen führen. Dazu gibt es Überlegungen, die aber noch nicht Eingang in die Simulationsumgebung der Entwurfsplattform gefunden haben [6].

2.2 Modellierungsansatz

Das elektrische Ersatzschaltbild der Einzelzelle (Bild 1) besteht üblicherweise aus einer Gleichstromspannungsquelle, die die Leerlaufspannung abbildet, den generellen Innenwiderstand R_{IN} , sowie ein oder mehrere RC-Glieder für das dynamische Verhalten. Die Zahl der RC-Glieder bestimmt die Genauigkeit, aber auch den Simulationsaufwand. Als guter Kompromiss hat sich die Verwendung von zwei RC-Gliedern ergeben. Dabei beschreibt das RC-Glied $R_1||C_1$ kurzzeitige Effekte im Sekundenbereich und das RC-Glied $R_2||C_2$ Effekte im Bereich von 100 bis 1000 Sekunden.

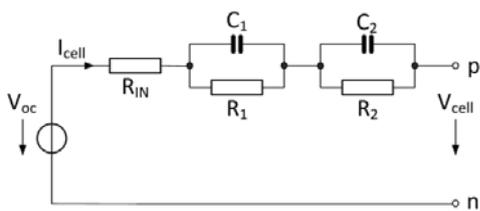


Bild 1: Ersatzschaltbild einer Batteriezelle

Die Gleichungen für die Analyse der in Bild 1 beschriebenen Ersatzschaltung lassen sich aus den Kirchhoffschen Gesetzen, dem Ohmschen Gesetz und den anderen Strom-Spannungs-Beziehungen der elektrischen Komponenten herleiten [7]. Als zusätzliche zentrale Größe aller Batteriemodelle ist der Ladezustand (State of Charge SoC) zu berücksichtigen. Er wird auf die Ladekapazität bezogen und meist in Prozent angegeben. Auch die komplementäre Größe (State of Discharge SoD = 1-SoC) findet gelegentlich Verwendung. Er ist selbst nicht direkt messbar. Aus-

gehend von einem bekannten Ladezustand, der notfalls durch vollständiges Laden erzielt wird, kann der Ladezustand durch Integration des Zellstroms i_{cell} berechnet werden Gleichung (1). Dabei muss beachtet werden, dass die Korrekturfaktoren α und β die Abhängigkeit von Lade- und Entladestrom bzw. Temperatur beschreiben. Leerlaufspannung V_{oc} und Zellspannung V_{cell} wiederum sind direkt von SoC, und damit auch von Zellstrom i_{cell} und Temperatur T abhängig, s. Gleichungen (2) und (3). Ebenso sind die Widerstände R_{IN} , R_1 und R_2 sowie die Kapazitäten C_1 und C_2 von SoC und Temperatur abhängig. Der Einfachheit halber wurde das in den Gleichungen weggelassen. Auch das Altern der Zelle wird zunächst nicht berücksichtigt. Dafür liegen derzeit noch nicht genügend Messungen vor.

Damit ergeben sich folgende Formeln:

- (1) $SoC(i_{cell}(t), T(t), t) = SoC_0 - \frac{1}{C_N} \cdot \int_0^t \alpha(i_{cell}(t)) \cdot \beta(T(t)) \cdot i_{cell}(t) \cdot dt$
- (2) $V_{oc}(i_{cell}(t), T(t), t) = \sum_{k=0}^n c_k \cdot SoC^k(i_{cell}(t), T(t), t) + \Delta V_{oc}(T)$
- (3) $V_{cell}(i_{cell}(t), T(t), t) = V_{oc}(i_{cell}(t), T(t), t) + R_{int} \cdot i_{cell}(t)$

- SoC ... State of Charge,
- C_N ... Nennkapazität,
- i_{cell} ... Entladestrom,
- α ... Korrekturfaktor Strom,
- T ... Temperatur,
- β ... Korrekturfaktor Temperatur,
- V_{oc} ... Leerlaufspannung,
- V_{cell} ... äußere Spannung der Zelle,
- R_{int} ... interner (Gesamt-)Widerstand der Zelle,
- $c_0 - c_n$... Koeffizienten der Funktion $V_{oc} = f(\text{SoC})$

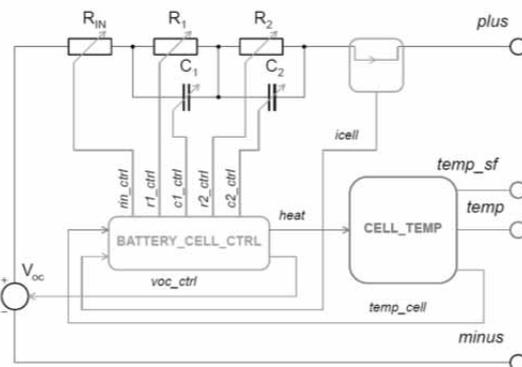


Bild 2: Struktur des Modells einer Batteriezelle in SystemC AMS

Das Modell der Batterie ist als lineares elektrisch-thermisches Netzwerk (ELN) in SystemC AMS im-

plementiert worden. Bild 2 zeigt dessen Struktur. Bei der Modellierung wurde ausgenutzt, dass sich SoC und Temperatur innerhalb kurzer Zeiten nur unwesentlich ändern. Im aktuellen Zeitschritt werden für die Auswertung SoC und Temperatur aus dem vorangegangenen Zeitschritt verwendet. Damit ist zur Ermittlung der Zellspannungen im aktuellen Zeitschritt ein lineares anstelle eines nichtlinearen Problems zu lösen.

Die Formeln (1) – (3) wurden zunächst entsprechend den Vorgaben aus [7] als Funktionen implementiert. Zur besseren Anpassung an die Messergebnisse wurde zum Tabellenmodell übergegangen.

2.3 Erweiterung des Modells für Temperatur

Zur Einbeziehung der Temperatur gehört nicht nur deren Berücksichtigung in den Gleichungen von SoC, V_{oc} und den Ersatzelementen, sondern auch die Entstehung und der Transport von Wärme während der Lade- und Entladevorgänge sowie die Wärmeübertragung aus und in die Umgebung.

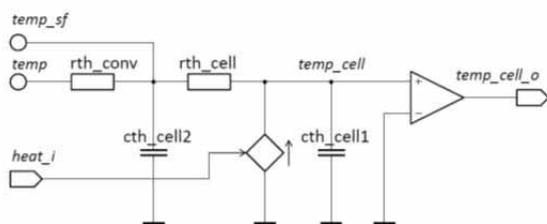


Bild 3: Nachbildung des Temperaturnetzwerkes der Batteriezelle in SystemC AMS

Im Temperaturnetzwerk in Bild 3 ist das berücksichtigt. Der Wärmeeintrag durch die elektrische Leistung in der Zelle ist unidirektional, ebenso der Einfluss der Zellinnentemperatur auf die Formeln zur Berechnung der internen Größen des Ersatzschaltbildes. Die Kommunikation mit der Zelloberfläche und der Umgebung geht in beide Richtungen. In der Beschreibung mit SystemC AMS wird das durch die Terminal-Ports verdeutlicht.

3 Parametrisierung des Modells

3.1 Elektrochemisch-kalorimetrische Messungen am KIT

Am Karlsruher Institut für Technologie (KIT) – Institut für Angewandte Materialien – Angewandte Werkstoffphysik (IAM-AWP) wurden verschiedene elektrochemisch-kalorimetrische Messungen an der Einzelzelle und an einem aus sechs in Serie geschalteten Zellen bestehenden Batteriepack zur Parametrisierung durchgeführt. Zum einen wurden wichtige Parameter

für das Temperaturnetzwerk, wie spezifische Wärmekapazität und Wärmetransferkoeffizient der Zelle in einem Accelerating Rate Calorimeter (ES-ARC, Fa. Thermal Hazard Technology) bestimmt. Zum anderen erfolgten statische und dynamische elektrochemische Messungen in einer Temperatorkammer (Voetsch VT³ 4018) mit dem Zyklierer BaSyTec HPS.

Der Ladezustand SoC ist von Zellstrom und Temperatur abhängig. Die Leerlaufspannung V_{oc} , sowie die Bauelemente der RC-Glieder des Ersatzschaltbildes sind von SoC und Temperatur abhängig. Diese Abhängigkeiten werden mit sogenannten Lookup-Tabellen beschrieben. Um sie besser zu erfassen, ist es sinnvoll, nur eine unabhängige Variable zu verändern. In unserem Falle wird deshalb die Temperatur konstant gehalten, d. h., die Messung findet unter isothermischen Bedingungen statt.

3.2 Statische Messungen

Statische Messungen sind sinnvoll, um die Abhängigkeit der Änderung des Ladezustands vom Lade- und Entladestrom zu bestimmen.

In den Ladephasen wird die Zelle vollständig aufgeladen. Das wird dadurch erreicht, dass nach Erreichen der Ladeschlussspannung mit fallendem Ladestrom weitergeladen wird. Man erreicht damit einen Ladezustand SoC von annähernd 100%.

Anschließend erfolgt die Entladung mit unterschiedlichen konstanten Strömen. Sie muss beendet werden, wenn die minimal zulässige Klemmenspannung erreicht wird. Das bedeutet aber nicht, dass die Zelle damit vollständig entladen wurde. Da über den Innenwiderstand der Zelle eine Spannung abfällt, liegt die Leerlaufspannung über der minimal zulässigen Klemmenspannung. Diese Spannungsdifferenz ist umso kleiner je kleiner der Entladestrom ist.

Nach Abschalten des Entladestroms stellt sich an den Klemmen der Zelle die Leerlaufspannung ein. Sie ist ein Maß für die Entladung der Zelle.

3.3 CIT-Messungen

Zur Bestimmung der Werte der Widerstände und Kapazitäten sind dynamische Messungen notwendig. Das kann mit elektrochemischer Impedanzspektroskopie (EIS) oder mittels Strompulsen geschehen. Im vorliegenden Artikel kam letztere Methode zur Anwendung.

Bei der auch Current Interruption Technique (CIT) genannten Methode wird die Zelle mit definierten Strompulsen entladen. In unserem Falle betrug die Stromstärke der Pulse 1 C, d. h., sie entsprach in A

etwa der in Ah gemessenen Nennkapazität. Die Dauer der Pulse war so bemessen, dass die Zelle um etwa 1% entladen wurde, d. h., etwa 36 s lang. Zwischen den Pulsen lag eine Relaxationszeit von etwa einer Stunde. In dieser Zeit hatte sich an den Klemmen die Leerlaufspannung wieder eingestellt. Damit konnte jeder Entladepuls von der Leerlaufspannung starten.

Zu Beginn der Messungen war die Zelle vollständig aufgeladen. Bei den Messungen machte das sich dadurch bemerkbar, dass die Zellspannung leicht sank, ohne dass Entladestrom floss.

4 Algorithmen zur Bestimmung der Parameter

4.1 Funktionen vs. Tabellen

Die Abhängigkeiten der internen Größen wie SoC, V_{oc} , R_{IN} , R_1 , C_1 , R_2 und C_2 können als analytische Funktionen oder Tabellen formuliert sein.

- $SoC = f_1(i_{cell}, T)$ oder $SoC = tab_1(i_{cell}, T)$
- $V_{oc} = f_2(SoC, T)$ oder $V_{oc} = tab_2(SoC, T)$
- $R_{IN} = f_3(SoC, T)$ oder $R_{IN} = tab_3(SoC, T)$
- $R_1 = f_4(SoC, T)$ oder $R_1 = tab_4(SoC, T)$
- $C_1 = f_5(SoC, T)$ oder $C_1 = tab_5(SoC, T)$
- $R_2 = f_6(SoC, T)$ oder $R_2 = tab_6(SoC, T)$
- $C_2 = f_7(SoC, T)$ oder $C_2 = tab_7(SoC, T)$

Analytische Funktionen haben den Vorteil, dass die Modelle bei der Simulation sehr schnell sind. Sie müssen den gesamten Wertebereich der Argumente erfassen. Das geht oft nur auf Kosten der Genauigkeit. Tabellen sind hingegen leichter zu erstellen, da sie einen lokalen Bezug zu den Messergebnissen haben. Oft sind Tabellen auch ein Zwischenergebnis, aus dem die analytischen Funktionen gewonnen werden. Tabellen sind damit flexibler und können leichter neuen Messergebnissen angepasst werden. Ein besonderer Vorteil besteht bei mehrdimensionalen Tabellen, wie sie beim Modell der Li-Ionen-Zelle auftreten. Die Formulierung voneinander unabhängiger Funktionen der Argumente Strom und Temperatur oder Ladezustand und Temperatur wie sie das Modell von Gao [7] vorsieht, gelingt nicht immer. Mehrdimensionale Tabellen umgehen dieses Problem; die Interpolation kann in jeder Dimension erfolgen.

Wir haben uns für ein Tabellen-Modell entschieden. Es umfasst im Wesentlichen die oben definierten Tabellen tab_1 bis tab_7 .

Sowohl bei den statischen als auch bei den CIT-Messungen erreicht man den Punkt $SoC = 0\%$ nicht wirklich. Das ist auch gut so, da sonst die Zelle irreparabel geschädigt wird.

Beim kleinsten Entladestrom z. B. 0,1 C kommt man diesem Punkt als nächsten. Durch Vergleich der Leerlaufspannungen am Ende der statischen Entladungen mit 1 C und 0,1 C sowie der (letzten) CIT-Entladung mit 1 C kann der Ladezustand am Ende dieser Entladungen abgeschätzt werden.

Nun kann der Verlauf des Ladezustands SoC über der aus den CIT-Messungen ermittelten Kurve bestimmt werden. Für die Tabelle $V_{oc} = tab_2(SoC)$ braucht man noch die den Ladezuständen zugeordnete Leerlaufspannung. Die Relaxationszeit, d. h. die Erholungsphase nach jedem Entladepuls, wurde so gewählt, dass sich an deren Ende die Leerlaufspannung V_{oc} als Klemmenspannung einstellt. Diese kann direkt in die Tabelle übernommen werden.

Mit der Tabelle $V_{oc} = tab_2(SoC)$ können jetzt die Ladezustände zum Ende der statischen Entladungen, gleichzeitig Beginn der statischen Ladungen, bestimmt werden. Damit ist der Verlauf von SoC für diese Kurven bekannt, und aus dem Verhältnis der entnommenen Ladungsmenge zur Nennkapazität der Zelle kann der Ratenfaktor α für den jeweiligen Lade- und Entladestrom berechnet werden. Diese Ratenfaktoren werden zu der Tabelle $SoC = tab_1(i_{cell})$ zusammengefasst.

Die eindimensionalen Tabellen werden für verschiedene Temperaturen, z. B. $0^\circ C$, $25^\circ C$ und $40^\circ C$ aufgestellt und aus ihnen zweidimensionale Tabellen $SoC = tab_1(i_{cell}, T)$ und $V_{oc} = tab_2(SoC, T)$ gebildet.

4.2 Fitting mit lsqcurvefit

Da die RC-Glieder dynamisches Verhalten haben, kann ihre Dimensionierung gut durch Beobachten der Sprungantwort bestimmt werden. Genau das passiert bei den CIT-Messungen.

Wir betrachten einen Teil der CIT-Entladekurve bestehend aus Entladepuls und Relaxationsphase.

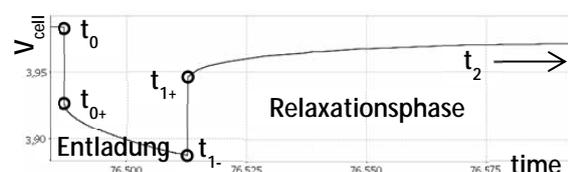


Bild 4: betrachteter Teil der CIT-Entladekurve

In Bild 4 ist solch ein Teil der CIT-Kurve mit den relevanten Zeitpunkten dargestellt. Zum Zeitpunkt t_0 wird der Entladestrom schlagartig eingeschaltet und die Zellspannung als Sprungantwort gemessen. Zum Zeitpunkt t_1 wird die Entladung wieder schlagartig beendet und eine erneute Sprungantwort gemessen. Danach erfolgt eine relativ lange Relaxationsphase (bis t_2), an deren Ende die Zellspannung der Leerlaufspannung entspricht. Die Zeitpunkte „kurz vor“ werden mit „-“, die Zeitpunkte „kurz danach“ mit „+“ gekennzeichnet. Damit lassen sich die links- bzw. rechtsseitigen Grenzwerte der Spannungen an den Sprungzeitpunkten beschreiben. Aus dem Ersatzschaltbild kann man die Sprungantworten für die Stromimpulse der Größe I_{CIT} berechnen und mit den Messungen vergleichen.

Für den Beginn der Entladung $t = t_0$ gilt:

$$(4) \quad V_{cell}(t_0) = V_{oc}(SoC_0)$$

$$(5) \quad V_{cell}(t_{0+}) = V_{oc}(SoC_0) - i_{cell} \cdot R_{IN}$$

Im Verlauf des Entladepulses $t_0 < t < t_1$ laden sich die beiden Kapazitäten C_1 und C_2 entsprechend der jeweiligen Zeitkonstante $\tau_1 = R_1 \cdot C_1$ und $\tau_2 = R_2 \cdot C_2$ auf.

$$(6) \quad V_{cell}(t) = V_{oc}(SoC(t)) - i_{cell} \cdot R_{IN} \\ - i_{cell} \cdot R_1 \cdot \left(1 - e^{-\frac{(t-t_0)}{(R_1 \cdot C_1)}}\right) \\ - i_{cell} \cdot R_2 \cdot \left(1 - e^{-\frac{(t-t_0)}{(R_2 \cdot C_2)}}\right)$$

Die Zellspannung beträgt zum Ende des Entladepulses $t = t_1$:

$$(7) \quad V_{cell}(t_{1-}) = V_{oc}(SoC_1) - i_{cell} \cdot R_{IN} \\ - i_{cell} \cdot R_1 \cdot \left(1 - e^{-\frac{(t_1-t_0)}{(R_1 \cdot C_1)}}\right) \\ - i_{cell} \cdot R_2 \cdot \left(1 - e^{-\frac{(t_1-t_0)}{(R_2 \cdot C_2)}}\right)$$

Der Sprung des Ladestroms erfolgt zum Zeitpunkt $t = t_1$. Auch hier gibt es einen Spannungssprung über R_{IN} , so dass nach dem Sprung des Stromes von I_{CIT} auf 0 gilt:

$$(8) \quad V_{cell}(t_{1+}) = V_{oc}(SoC_1) - i_{cell} \cdot R_1 \cdot \left(1 - e^{-\frac{(t_1-t_0)}{(R_1 \cdot C_1)}}\right) \\ - i_{cell} \cdot R_2 \cdot \left(1 - e^{-\frac{(t_1-t_0)}{(R_2 \cdot C_2)}}\right)$$

Die beiden Kapazitäten C_1 und C_2 sind am Ende des Entladepulses geladen und können sich in der Relaxationsphase entladen:

$$(9) \quad V_{cell}(t) = V_{oc}(SoC_1)$$

$$- i_{cell} \cdot R_1 \cdot \left(1 - e^{-\frac{(t_1-t_0)}{(R_1 \cdot C_1)}}\right) \cdot e^{-\frac{t-t_1}{R_1 \cdot C_1}} \\ - i_{cell} \cdot R_2 \cdot \left(1 - e^{-\frac{(t_1-t_0)}{(R_2 \cdot C_2)}}\right) \cdot e^{-\frac{t-t_1}{R_2 \cdot C_2}}$$

Am Ende der Relaxationsphase sind die Kapazitäten wieder entladen und $V_{cell}(t_2) = V_{oc}(SoC_1)$.

Die Zelle wird während des Entladepulses gleichmäßig entladen, d. h., der SoC sinkt linear von SoC_0 zu Beginn auf SoC_1 am Ende des Pulses, $t_0 < t < t_1$:

$$(10) \quad SoC(t) = -\frac{SoC_0 - SoC_1}{t_1 - t_0} \cdot t + \frac{SoC_0 \cdot t_1 - SoC_1 \cdot t_0}{t_1 - t_0}$$

Man beachte, dass in den Gleichungen R_{IN} , R_1 , C_1 , R_2 und C_2 ebenfalls von SoC abhängig sind. Da sich der SoC nur um 1% ändert, kann man für diese Variablen den SoC als konstant betrachten.

Der Wert von R_{IN} wird aus der der Sprunghöhe der jeweiligen Pulsantwort bestimmt. Mit Hilfe der Routine *lsqcurvefit* in MATLAB können die Werte für R_1 , C_1 , R_2 und C_2 aus der Entladekurve und aus der Relaxationskurve bestimmt werden. Die beiden RC-Glieder sind in Reihe geschaltet und keines zeichnet sich gegenüber dem anderen aus. Deshalb können die beiden RC-Glieder getauscht werden, ohne dass sich etwas an der Lösung ändert. Um unnötige Sprünge im Verlauf $R_i = f_i(SoC)$ und $C_i = f_k(SoC)$ zu vermeiden, wird $R_1 \cdot C_1 < R_2 \cdot C_2$ festgelegt.

Man erhält auf diese Weise zunächst Werte von R_{IN} , R_1 , C_1 , R_2 und C_2 für jeden Entladepuls und jede Relaxationsphase. Diese werden zu eindimensionalen Tabellen für R_{IN} , R_1 , C_1 , R_2 und C_2 abhängig vom SoC bei einer bestimmten Temperatur zusammengefasst. Dabei zeigt sich, dass sich die Werte von Puls zu Puls relativ stark ändern. Das liegt daran, dass schon kleine Messfehler und das Rauschen zu großen Abweichungen führen können.

4.3 Simulative Optimierung der Lookup-Tabellen für die Modellbeschreibung

Für die Simulation werden Tabellen gebraucht, in denen die Werte einigermaßen stetig verlaufen. Das wird dadurch erreicht, dass für die Tabellen 5%-SoC-Schritte vorgegeben werden. Diese Werte erhält man durch Mittelwertbildung und Glätten der durch das Fitting gewonnenen Tabellen. Die in Abschnitt 4.1 genannten Tabellen $tab_3 - tab_7$ kann man zu einer großen Tabelle zusammenfassen, in der für jeden SoC-Wert Werte für alle fünf Ersatzschaltungselemente vorliegen. Im Folgenden wird diese Tabelle punktuell verbessert, indem der Wert eines Elements

an einer oder zwei benachbarten Stellen in der Tabelle verändert wird.

Diese Tabelle bildet die Grundlage für die Simulation der Ersatzschaltung mit MATLAB.

Durch Vergleich der Ergebnisse einer ersten Simulation mit den Vorgaben aus den Messungen findet man die Stelle mit der größten Abweichung. Dieser Stelle ist ein SoC-Wert zugeordnet, der im Allgemeinen zwischen zwei Werten der 5%-Skala der SoC-Werte liegt. Bei der Simulation werden die Werte für die Ersatzschaltbildelemente R_{IN} , R_1 , C_1 , R_2 und C_2 durch Interpolation der beiden Nachbarwerte berechnet. Wird einer (oder beide) Nachbarwerte geändert, so ändert sich das entsprechende Ersatzschaltbildelement und damit auch der Funktionswert an dieser Stelle und der Nachbarstelle.

Durch systematisches Ändern der Nachbarwerte werden variierte Tabellen erzeugt, Ursprungstabelle und Varianten simuliert und nach Vergleich mit den Messwerten die Variante ermittelt, die die geringste Abweichung von den Messwerten erwarten lässt.

	SoC	R_{IN}	R_1	C_1	R_2	C_2
Ursp
	70%	0,04	0,045	110	0,032	660
	75%	0,03	0,044	100	0,033	650
Var1	...	0,044
	70%	0,03	0,045	110	0,032	660
	75%	0,03	0,044	100	0,033	650
Var2	...	0,04
	70%	0,033	0,045	110	0,032	660
	75%	0,033	0,044	100	0,033	650
Var3	...	0,044
	70%	0,033	0,045	110	0,032	660
	75%	0,033	0,044	100	0,033	650
Var4	...	0,036
	70%	0,03	0,045	110	0,032	660
	75%	0,03	0,044	100	0,033	650
Var5	...	0,04
	70%	0,027	0,045	110	0,032	660
	75%	0,027	0,044	100	0,033	650
Var6	...	0,036
	70%	0,027	0,045	110	0,032	660
	75%	0,027	0,044	100	0,033	650

Tabelle 1 Ausschnitt aus der Ursprungstabelle Ursp zwischen SoC=70% und 75% sowie Ausschnitte aus den variierten Tabellen Var1 ... Var6 mit Änderungen von R_{IN} um $\pm 10\%$.

Dabei wird jedes Element unabhängig von den anderen verändert. Jeder Nachbarwert wird um einen bestimmten Prozentsatz erhöht und erniedrigt, aber auch beide Nachbarwerte erhöht und erniedrigt. Für jedes Element ergeben sich so sechs verschiedene Änderungen. Bei fünf verschiedenen Ersatzschalt-

bildelementen sind das 30 variierte Tabellen. Zusammen mit der Ursprungstabelle gibt es also 31 verschiedene Tabellen für die Ersatzschaltung. Die Gleichungen in Abschnitt 4.2 wurden in MATLAB programmiert. MATLAB erlaubt es, dass alle 31 verschiedenen Tabellen parallel simuliert werden. Nach der Simulation liegen dann 31 verschiedene Kurven vor, aus denen die mit der geringsten Abweichung ausgewählt wird. Die zugehörige Tabellenvariante ist Ursprungstabelle für die nächste Iteration. Tabelle 1 zeigt beispielhaft einen Ausschnitt der Ursprungstabelle sowie jeweils die beiden relevanten Zeilen der ersten sechs von insgesamt 30 variierten Tabellen.

CIT-Messungen sind allgemein sehr zeitaufwendig. Dementsprechend dauern auch die Simulationen sehr lange. Ein lokales Vorgehen ist deshalb sinnvoll. Liegt die Stelle mit der größten Abweichung z. B. bei SoC=73%, so wird die Tabelle bei SoC=70% und SoC=75% geändert. Durch die Interpolation kann das Auswirkungen auf SoC-Werte zwischen 65% und 80% haben. Damit kann als Anfangszustand SoC=80% bzw. der nächstgrößere Wert von SoC in der Messkurve verwendet werden, und die Simulation kann bei Unterschreiten von SoC=65% beendet werden.

Die auf diese Weise optimierten Tabellen wurden für die Modellbeschreibung in SystemC AMS verwendet und die Simulation mit COSIDE ausgeführt. Bild 5 zeigt den Vergleich der Zellspannung von CIT-Messung und COSIDE-Simulation an drei Ladepulsen, Bild 6 detaillierter für einen Puls.

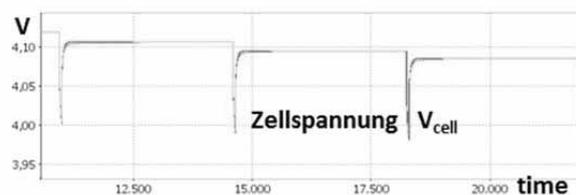


Bild 5: Vergleich Messung (grün) – Simulation (rot) bei Pulsentladung (CIT) für drei Entladepulse

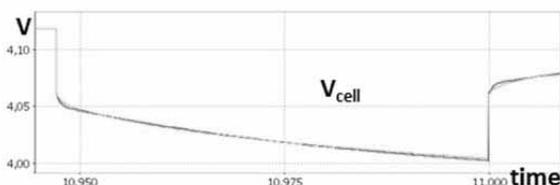


Bild 6: Vergleich Messung (grün) – Simulation (rot) bei Pulsentladung (CIT) für einen Entladepuls

5 Validierung

5.1 Vergleich Simulation-Messung für Einzelzelle beim NEFZ

Zur Validierung wurde ein Stromprofil für einen elektrischen Antrieb auf Basis des Neuen Europäischen Fahrzyklus‘ NEFZ verwendet. Dieser Zyklus besteht aus – sowohl in Amplitude als auch Anstieg - unterschiedlich starken Entladeimpulsen für unterschiedlich starke und lange Beschleunigungen als auch Ladephasen wie sie beim Bremsen auftreten. Während der 500 s dauernden Zyklen werden die Zellen um ca. 15% entladen. Um das Stromprofil in unterschiedlichen Ladezuständen zu testen, wurden fünf Messungen und Simulationen mit 100%, 80%, 60%, 40% und 20% durchgeführt. In einer 6. Messung wurde die vollständig geladene Zelle durch Ausführung mehrerer Zyklen hintereinander fast vollständig entladen.

Die Messungen wurden als elektrochemisch- kalorimetrische Messungen am KIT durchgeführt. Der Vergleich von Messungen und Simulationsergebnissen ergab eine gute Übereinstimmung der Zellspannung. Bild 7 zeigt die Kurven zu diesem Vergleich bei einem Anfangsladezustand von 60%.

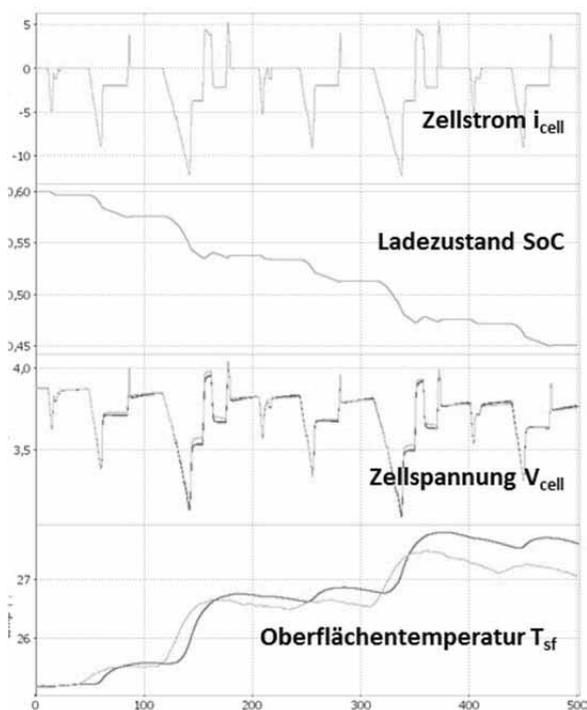


Bild 7: Vergleich Messung (grün) – Simulation (rot) der Einzelzelle für den NEFZ bei SoC=60%

Beim Vergleich der Temperaturkurven ergab sich eine Abweichung von 0,5 K. Am Fahrzeug wird mit einer Genauigkeit von 1 K gemessen. Dabei wurde vom

Modell sowohl die Wärmeerzeugung als auch der Abfluss der Wärme in die Umgebung erfasst. Die weiteren Untersuchungen sollten sich auf das Batteriepack beziehen und längere Messungen / Simulationen berücksichtigen.

5.2 Vergleich Simulation-Messung für Batteriepack beim NEFZ

Für den Betrieb im Fahrzeug werden viele Zellen benötigt. Dabei bilden jeweils sechs Zellen ein Batteriepack. Ein Batteriepack ist die kleinste Einheit, die mit dem Batteriemanagementsystem zusammenarbeitet. Die Zellen des Packs sind elektrisch und thermisch zusammengeschaltet. Das Modell und die Parameter sind für alle Zellen eines Packs gleich; sie unterscheiden sich nur (geringfügig) in ihrer Nennkapazität und im Anfangsladezustand. Damit haben alle Zellen des Packs einen individuellen Anfangs-SoC. Durch wiederholtes Laden und Entladen des Packs können die Unterschiede des Ladezustands der Zellen größer werden. Einen Ausweg bietet das Cell Balancing, bei dem Zellen gezielt entladen oder weniger geladen werden, um die Ladezustände anzugleichen.

Die Messungen wurden am KIT auch für ein Batteriepack durchgeführt. Neben dem Batteriestrom und der Gesamtspannung des Packs wurden die Spannungen der einzelnen Zellen sowie deren Oberflächentemperaturen gemessen.

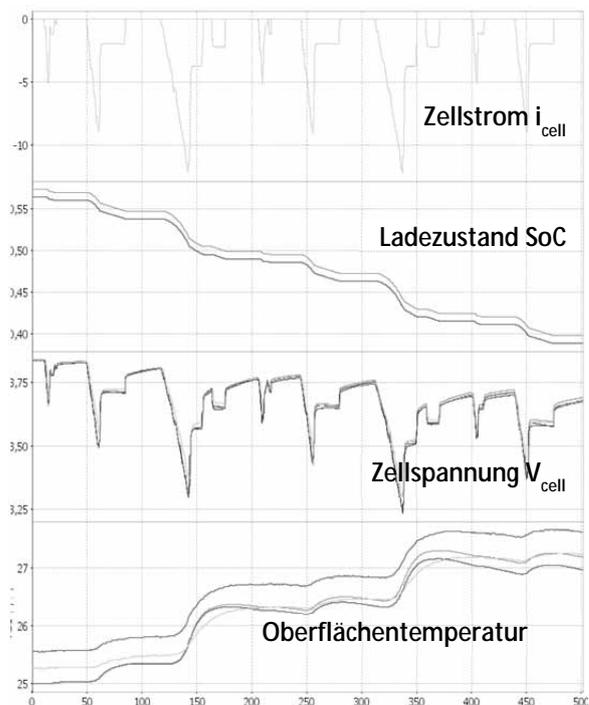


Bild 8: Messung (dunkel- + hellgrün) – Simulation (rot+orange) von 2 Zellen des Batteriepacks für den NEFZ

Der Vergleich von Mess- und Simulationsergebnissen kann hier wegen der großen Zahl der Signale nur beispielhaft für zwei Zellen erfolgen. Es wurden eine Randzelle und eine innere Zelle ausgewählt, da sich deren Temperaturen signifikant unterscheiden. In Bild 8 werden die Ergebnisse gegenübergestellt. Um Vergleiche mit dem Demonstrator des IKEBA-Projekts zu ermöglichen, wurde als Stimulus der NEFZ ohne Ladezyklen verwendet.

Bei den Zellspannungen sind die Abweichungen kleiner als 10 mV, bei den Temperaturen kleiner als 0,5 K. Bei den Temperaturen kommt es nicht nur auf die kurzzeitige Wärmeentwicklung, sondern auch auf die Wärmespeicherung und den Wärmetransfer an. Deshalb ist es notwendig, in einer weiteren Messung das vollständig geladene Batteriepack mit aneinandergereihten Fahrzyklen in knapp einer Stunde vollständig zu entladen. Neben dem Zellstrom und den Zellspannungen der einzelnen Zellen werden die Oberflächentemperaturen der jeweils beiden inneren und äußeren Zellen des Packs ausgewertet. Das erlaubt eine Anpassung des Modells an die reale Umgebung und eine genauere Beschreibung der thermischen Verkopplung der Batteriezellen.

6 Zusammenfassung und Ausblick

Leistungsfähige Batteriemodelle sind zentrale Komponenten von Simulationsumgebungen für den Entwurf von Batteriemanagementsystemen. Positive Erfahrungen wurden mit der Parametrisierung derartiger Modelle auf der Basis von CIT-Messungen bei unterschiedlichen konstanten Temperaturen gesammelt. Bei der Parametrisierung wurde in zwei Stufen vorgegangen. Zunächst wurden abschnittsweise Parameter auf der Grundlage von Klemmenströmen und -spannungen einer Batteriezelle mit einem Algorithmus zur Anpassung von Kurvenverläufen abgeleitet. Die so ermittelten Werte dienen als Stützstellen für Tabellenmodelle. In einem zweiten Schritt wurden die Funktionswerte an den Stützstellen modifiziert, um eine bessere Übereinstimmung von Simulations- und Messergebnissen für längere Messabläufe zu erreichen. Diese Modifikation erfolgte durch Reduzierung der Abweichung als Folge von vorgegebenen Variationen der Funktionswerte an den Stützstellen. Die so ermittelten Tabellenmodelle weisen auch für andere Stromprofile als die bei den CIT-Messungen verwendeten eine gute Übereinstimmung von Mess- und Simulationsergebnissen auf.

Um eine ähnlich gute Übereinstimmung zwischen Mess- und Simulationsergebnissen wie bei der Einzelzelle auch für ein Batteriepack zu erreichen, sind weitere Überlegungen erforderlich. Insbesondere sind dabei nach gegenwärtigem Erkenntnisstand die Anforderungen an die Genauigkeit der Beschreibung der thermischen Verkopplung der Batteriezellen zu untersuchen.

Förderhinweis

Die diesem Beitrag zugrundeliegenden Arbeiten wurden mit Mitteln des Bundesministeriums für Bildung und Forschung unter den Förderkennzeichen FKZ 16N12440 und 16N12515 im Rahmen des Verbundprojektes „IKEBA-integrierte Komponenten und integrierter Entwurf energieeffizienter Batteriesysteme“ gefördert unter Betreuung des Projektträgers VDI/VDE Innovation + Technik GmbH. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Literatur

- [1] Vorhaben IKEBA. URL: <http://www.iam.kit.edu/awp/ikeba/>
- [2] Markwirth, T.; Gulbins, M.; Einwich, K.; Haase, J.: Efficient Modelling of Heterogeneous Battery Management Systems (BMS), *SNE Simulation Notes Europe*, Volume 25, Issue 2. Vienna, ARGESIM / ASIM / EUROSIM 2015, pp. 93-98.
- [3] *IEEE Std 1666-2011 for Standard SystemC Language Reference Manual*. Jan 2012. doi: 10.1109/IEEESTD.2012.6134619
- [4] *IEEE P1666.1/D4 Draft SystemC Analog/Mixed-Signal (AMS) extensions Language Reference Manual*, October 2015. URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7293581&isnumber=7293580>
- [5] Banerjee, A.; Sur, B.: *SystemC and SystemC-AMS in Practice: SystemC 2.3, 2.2 and SystemC-AMS 1.0*, Springer 2014.
- [6] Melcher, A.; Zieber, C.; Lei, B.; Rohde, M.; Seifert, H.-J.: Modellierung und Simulation des thermischen Runaways in zylindrischen Li-Ionen Batterien. *Unterlagen zu dieser Veranstaltung*
- [7] Gao, L.; Liu, S.; Dougal, R. A.: Dynamic Lithium-Ion Battery Model for System Simulation, *IEEE Trans. Components and Packaging Technologies*, vol. 25, no. 3, Sept. 2002, pp. 495 – 505

Vergleichende Untersuchung eines Rohrschlangen-Wärmeübertragers im Umgebungsmedium Wasser mittels numerischer Strömungsrechnung und dynamischer Prozesssimulation

Matthias Henninger¹, Wolfgang Schlüter²

¹Technologiezentrum Feuchtwangen

²Kompetenzzentrum Strömungssimulation, Hochschule Ansbach

matthias.henninger@hs-ansbach.de

Im Rahmen des Forschungsprojektes Energieversorgung Schwimmender Bauten (EnAqua) wurde die Beheizung schwimmender Bauten mithilfe einer Kombination aus Rohrschlangen-Wärmeübertrager und Wärmepumpe als umweltschonende Wärmequelle untersucht. Um die Dimensionierung zu erleichtern und ein besseres Prozessverständnis zu erhalten, wurden ein dynamisches Prozess- und Strömungssimulationsmodell für den Wärmeübertrager entwickelt.

Als Grundlage der Modellbildung werden der geometrische Aufbau und die Funktionsweise des Wärmeübertragers erläutert. Für die Entwicklung des dynamischen Prozesssimulationsmodells wird die verwendete Abstraktion des Problems und die daraus resultierende mathematische Modellierung beschrieben. Die mit Star CCM+ durchgeführte numerische Strömungsberechnung basiert auf der geometrischen Modellierung des Wärmeübertragers, welche mithilfe eines Makros flexibel gestaltet wurde, und dessen Diskretisierung. Um die Genauigkeit der realisierten Simulationsmodelle überprüfen zu können, wird ein Validierungskonzept erarbeitet, welches auf Messungen an einem Laborversuchsstand basiert.

Nach erfolgter Validierung wird eine Vergleichsstudie zwischen den Modellen ausgeführt, anhand welcher die Vor- und Nachteile der Modelle dargelegt werden. Anhand dieser Ergebnisse kann abgeschätzt werden, welches Modell unter welchen Rahmenbedingungen präferiert werden sollte, um eine ausreichende Ergebnisqualität bei möglichst geringem Ressourceneinsatz zu erhalten. Im Rahmen der Vergleichsstudie konnte gezeigt werden, dass die dynamische Prozesssimulation trotz eines hohen Abstraktionslevels gute Ergebnisse erzielt.

1 Einleitung

Die Energieverbraucher Deutschlands können grundsätzlich in vier Bereiche eingeteilt werden. Diese sind Industrie, Verkehr, Gewerbe/Handel und private Haushalte. Der Endenergieverbrauch der privaten Haushalte betrug im Jahr 2013 723 TWh [1]. Dies entspricht einem Anteil von 28 % am gesamten Endenergieverbrauch. Eine genauere Analyse dieses Verbrauchs zeigt, dass von diesem 69 % für die Raumwärme, also die Heizung des Hauses verwendet wird. Von diesen werden jedoch lediglich 14,5 % durch erneuerbare Energien erzeugt [2]. In Bezug auf die CO₂ Problematik, den Klimawandel und die Erreichung des 2 °C Zieles ist hier großes Potenzial vorhanden.

Im Forschungsprojekt EnAqua, welches sich mit der Energieversorgung schwimmender Bauten (Abb. 1) beschäftigte, wurde deshalb auch das Ziel einer umweltschonenden, erneuerbaren Wärmeversorgung der

Gebäude in die Agenda mitaufgenommen. Eine geprüfte Alternative, welche sich aufgrund der einfachen Zugänglichkeit der Wärmequelle "Seewasser" anbot, war der Einsatz eines Rohrschlangen-Wärmeübertragers in Kombination mit einer Wärmepumpe. Der Wärmeübertrager ist unter dem Gebäude angebracht und direkt in das Seewasser integriert.

Abbildung 1: Schwimmendes Gebäude ar-che-aqua [3]

α_a	Wärmeübergangskoeffizient Rohraußenseite	[W/(m ² *K)]	Nu	Nußelt-Zahl	---
α_i	Wärmeübergangskoeffizient Rohrinneiseite	[W/(m ² *K)]	Nu _l	Laminare Nußelt-Zahl	---
ρ_{Seew}	Dichte Seewasser	[kg/m ³]	Nu _t	Turbulente Nußelt-Zahl	---
λ_R	Wärmeleitfähigkeit Rohr	[W/(m*K)]	\dot{Q}	Wärmestrom	[W]
D_a	Außendurchmesser der Spirale	[m]	s	Abstand zwischen 2 Windungen	[m]
r_a	Außenradius Rohr	[m]	T_m	Mittlere log. Temperaturdifferenz	[K]
r_i	Innenradius Rohr	[m]	T_{Seew}	Temperatur Seewasser	[K]
l	Länge des „gestreckten“ Rohres	[m]	$T_{\text{WTM,aus}}$	Temperatur WTM am Austritt	[K]
L	Axialer Platzbedarf	[m]	$T_{\text{WTM,ein}}$	Temperatur WTM am Eintritt	[K]
L_c	Charakteristische Länge	[m]	v_{Seew}	Strömungsgeschwindigkeit Seewasser	[m/s]
\dot{m}_{WTM}	Massenstrom WTM, kg/s	[kg/s]	v_{WTM}	Strömungsgeschwindigkeit WTM	[m/s]
Mat	Rohrmaterial	---	WTM	Wärmeträgermedium	---
n	Anzahl der Windungen	---	WÜ	Wärmeübertrager	

Um die Leistungsfähigkeit des Wärmetauschers untersuchen und abschätzen zu können, wurden ein dynamisches Prozesssimulationsmodell sowie ein CFD-Modell erarbeitet.

In der vorliegenden Arbeit werden konkret diese Ansätze der dynamischen Prozess- und CFD-Simulation für die Untersuchung eines Rohrschlangen-Wärmeübertragers betrachtet und miteinander verglichen. Um die Komplexität des dynamischen Prozesssimulationsmodells und des dazugehörigen mathematischen Modells möglichst gering zu halten, ist es ein wesentliches Ziel eine geeignete Abstraktion des Problems zu finden. Die Umsetzung des Prozesssimulationsmodells erfolgt mit MATLAB und die numerische Strömungsberechnung mit der Software Star CCM+ von CD-adapco. In der Arbeit sollen die Anwendbarkeit und Übereinstimmung der Ansätze mit realen Messwerten zum konkreten Wärmeübertragungsproblem anhand von Messungen an einem Laborprüfstand untersucht werden. Nach Bestätigung der Anwendbarkeit der Modelle werden die Modelle unter unterschiedlichen Rahmenbedingungen miteinander verglichen. Diese Gegenüberstellung zielt darauf ab zu ermitteln, welches Modell bei welchen Rahmenbedingungen zu präferieren ist und warum dies der Fall ist. Die Anwendung des passenden Modells soll eine ausreichende Qualität bei möglichst geringem Ressourceneinsatz ermöglichen. Ebenfalls soll gezeigt werden, dass mithilfe der dynamischen Prozesssimulation qualitativ gute Ergebnisse bei geringem Aufwand möglich sind und es somit nicht zielführend ist generell auf CFD-Programme zurückzugreifen.

2 Systembetrachtung

Als Basis für die Modellerstellung müssen der Aufbau und die Funktionsweise des Systems bekannt sein. Dabei gibt es grundsätzlich verschiedene Varianten an Rohrschlangen-Wärmeübertragern. Die verwendete Variante ist in Abbildung 2 dargestellt. Es ist zu erkennen, dass der Wärmeübertrager aus einem gewundenen Rohr besteht, welches die beteiligten Medien voneinander trennt. Im inneren des Rohres strömt das Wärmeträgermedium (WTM/Sole), welches einer 34 Vol. %-igen Glykol N/Wasser Mischung entspricht. Außerhalb des Rohres befindet sich das Seewasser, welches als Wärmequelle genutzt wird.

Abbildung 2: Aufbau des Spiralwärmetauschers

Die Funktionsweise des Wärmetauschers wird anhand von Abbildung 3, die ein auf 4,5 Windungen verkürztes Modell zeigt, erklärt werden. Für die Beschreibung wird davon ausgegangen, dass die Temperatur des Seewassers höher ist, als die Temperatur des Wärmeträgermediums und somit ein Erwärmungsvorgang von diesem auftritt. Das WTM tritt mit minimaler Temperatur am Eintritt ein und wird bei

Durchströmen des Wärmetauschers erwärmt. Am Austritt ist die Temperatur des WTMs infolge des Wärmestroms vom Seewasser in Richtung Sole maximal. Die Größe des Wärmestroms ist vor allem abhängig von:

- der Temperaturdifferenz zwischen Seewasser und WTM
- der Größe der Übertragungsfläche
- der Wärmeleitfähigkeit des Rohrmaterials
- dem konvektiven Wärmeübergang innerhalb und außerhalb der Rohrwand

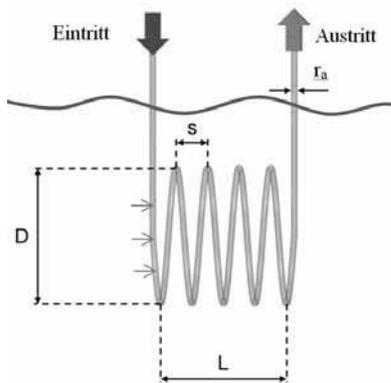


Abbildung 3: Funktionsweise des Wärmeübertragers

Die bestimmenden geometrischen Größen des Wärmeübertragers sind in Tabelle 1 angegeben.

Tabelle 1: Geometriedaten des Spiralwärmetauschers

Spezifikation	Symbol	Wert	Einheit
Ausdehnung der Spirale in x-Richtung	L	8,780	m
Außendurchmesser der Spirale	D _a	0,685	m
Außendurchmesser des Rohres	r _a	0,016	m
Innendurchmesser des Rohres	r _i	0,013	m
Länge des Rohres	l	119,930	m
Anzahl der Windungen	n	58	---
Abstand der Windungen	s	0,150	m
Material des Rohres	Mat	PE-80	---

3 Modell der dynamischen Prozesssimulation

Grundlage der dynamischen Prozesssimulation ist die Modellbildung und die damit verbundene mathematische Beschreibung des Problems [4]. Um den Rohrschlangen-Wärmeübertrager mathematisch einfach erfassen zu können wird die Rohrschlange, wie in Abbildung 4 symbolisch angedeutet „auseinander“ gezogen bis schließlich ein gerades Rohr vorliegt. Dies führt zu einer signifikanten Vereinfachung der physikalischen Beschreibung. Der Anströmwinkel zwischen Seewasser und Rohr wird auf 90 ° festgelegt, das heißt das Rohr wird wie in Abbildung 4 angedeutet quer angeströmt.

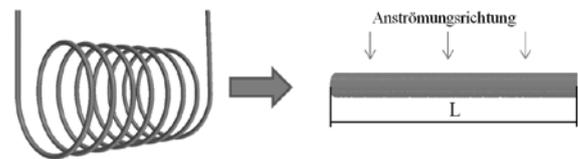


Abbildung 4: Vereinfachung der Geometrie

Die übertragene Wärme kann nun aufgrund der Vereinfachung durch die Peclet-Gleichung [5] für Zylinder berechnet werden (1). Die Gleichung bildet die Abhängigkeiten des auftretenden Wärmestroms \dot{Q} zwischen Seewasser und WTM ab. In (1) entspricht T_m der logarithmischen Temperaturdifferenz und nicht wie in der Literatur angegeben der adiabaten Mischtemperatur.

$$\dot{Q} = \frac{2 \cdot \pi \cdot l \cdot T_m}{\underbrace{\frac{1}{\alpha_i \cdot r_i}}_{\text{Konvektion}} + \underbrace{\frac{1}{\lambda_R \cdot \ln\left(\frac{r_a}{r_i}\right)}}_{\text{Leitung}} + \underbrace{\frac{1}{\alpha_a \cdot r_a}}_{\text{Konvektion}}} \quad (1)$$

Konvektion Leitung Konvektion

Die Wärmeübergangskoeffizienten α_i und α_a können nach Ansätzen des VDI-Wärmeatlas berechnet werden [6]. Grundlage dieser Berechnungsansätze ist die Ähnlichkeitstheorie des Wärmeübergangs, welche auf empirischen Formeln basiert. Durch die Ähnlichkeitstheorie erfolgt die Berechnung von α über die dimensionslose Kennzahl von Nußelt (2).

$$\alpha = \frac{Nu \cdot \lambda}{L_c} \quad (2)$$

Für die Berechnung des Wärmeübergangskoeffizienten α_a auf der Rohraußenseite basiert das Grundmodell auf einem querangeströmten, umströmten Zylinder. Für diesen konkreten Fall setzt sich die Nußelt-

Zahl, wie in (3) angegeben aus einem laminaren Anteil Nu_l und einem turbulenten Anteil Nu_t zusammen. Die Berechnungsansätze der genannten Nußelt-Zahlen sind im Unterkapitel 2 von Hauptkapitel G6 (S.817) im VDI-Wärmeatlas zu finden [6]. Die Berechnungsansätze von α_i sind in der gleichen Literatur im Unterkapitel 4.1 von Hauptkapitel G1 (S.788) zu finden.

$$Nu = 0,3 + \sqrt{Nu_l^2 + Nu_t^2} \quad (3)$$

Eine relevante Größe für die Auslegung und Validierung des Wärmeübertragers ist die Temperatur am Austritt, welche auch als Rücklauf­temperatur bezeichnet wird. Anhand dieser kann eine Aussage über die Leistungsfähigkeit des Wärmeübertragers gemacht werden. Mit (1) und (4) kann die Änderung der Austrittstemperatur berechnet werden. Diese ist abhängig von der Differenz zwischen der Eintrittstemperatur des WTMs $T_{WTM, ein}$ und dessen Austrittstemperatur $T_{WTM, aus}$, sowie der Größe des Wärmestroms.

$$\frac{dT_{WTM, aus}}{dt} = T_{WTM, ein} - T_{WTM, aus} + \frac{\dot{Q}}{c_{WTM} \cdot \dot{m}_{WTM}} \quad (4)$$

4 Modellerstellung der numerischen Strömungssimulation

4.1 Geometrische Modellierung

Um den Wärmeübertragungsvorgang mit numerischer Strömungssimulation simulieren zu können wird das geometrische Modell, wie in Abbildung 5 gezeigt in die drei Bereiche Fluid 1 (WTM), Fluid 2 (Seewasser) und Rohr eingeteilt. Die Größe von Bereich Fluid 2 wird bewusst groß gewählt, um den Einfluss der Simulationsränder zu minimieren.

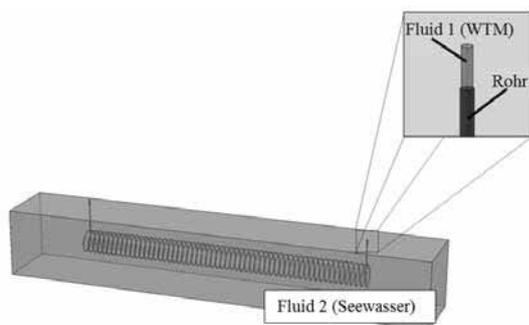


Abbildung 5: CFD-Aufbau des geometrischen Modells

Die geometrische Modellierung wurde durch ein Makro automatisiert, welches durch die Verwendung von drei Grundbausteinen, die aneinander gereiht werden können, eine flexible Anzahl an Windungen ermöglicht. Wie in Abbildung 6 zu erkennen werden die Bausteine Anfang, Mitte und Ende verwendet, welche wiederum die vorher beschriebene Dreiteilung in Sole, Rohr und Seewasser aufweisen. Der Teil Anfang besteht aus einem Einlassstutzen, einer Windung und einem großen Fernfeld, welches relevant ist um vor der ersten Windung ein homogenes Geschwindigkeitsfeld des Seewassers sicherzustellen. Der Aufbau des Teils Ende entspricht einer Spiegelung von Teil Anfang. Die Mitte entspricht exakt einer Windung, welches entsprechend der Anzahl an gewünschten Windungen dupliziert und aneinander gereiht werden kann. Neben dem Aufbau ist in der Abbildung zu erkennen, dass der Wärmeübertrager senkrecht von vorne angeströmt wird.

Abbildung 6: Grundbausteine des CFD-Modells

4.2 Vernetzung des Rechengebiets

Die zweckmäßige Vernetzung ist entscheidend für eine schnelle Simulation bei hinreichender Genauigkeit. Für die numerische Strömungsberechnung muss das im vorherigen Abschnitt erarbeitete geometrische Modell diskretisiert (vernetzt) werden. Für die Vernetzung werden die Rechengebiete Fluid 1 (WTM) und Fluid 2 (Seewasser) mit unstrukturierten Polyeder-Netz und das Rohr mit einem speziell für dünne Materialien entwickelten Netztyp vernetzt. Um eine konvergierende Lösung zu erzeugen, werden die Grenzflächen durch eine feinere Vernetzung genauer aufgelöst. Für eine geringe Rechenzeit und gutes Konvergenzverhalten ist zusätzlich die Zellqualität und Konformität des Netzes entscheidend. Unter Zellqualität wird die Abweichung einer Zelle zu einer ideal gleichmäßigen Zelle verstanden. Diese wird in mehreren Iterationsschritten verbessert, bis ein aus-

reichend hohes Qualitätslevel erreicht ist. Probleme mit der Konformität treten insbesondere zwischen den einzelnen Bereichen und Grundbausteinen auf. In Abbildung 7 links ist eine schlechte Konformität an der Grenzfläche von zwei Grundbausteinen dargestellt. Durch eine Verbesserung der Konformität wie in Abbildung 7 rechts gezeigt konnte die Rechenzeit drastisch verringert werden. Um die Unabhängigkeit der Simulation von der Vernetzung zu bestätigen, wurde eine Netzstudie durchgeführt. Es zeigt sich, dass ein Netz mit 3,5 Millionen nötig ist, um eine ausreichend genaue Auflösung zu erzielen, welches jedoch zu einem hohen Rechenaufwand führt.

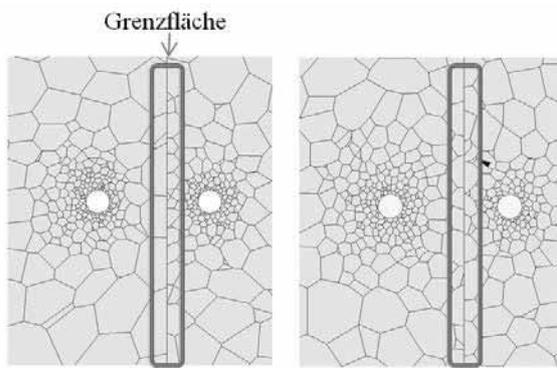


Abbildung 7: Schlechte und gute Konformität

4.2.1 Physikalisches Modell

Für die Modellierung von Turbulenzen wird das $k-\varepsilon$ -Modell verwendet, da dieses einen guten Kompromiss darstellt aus Robustheit, Rechenaufwand und Genauigkeit [7]. Es werden zwei verschiedene physikalische Modelle erstellt. In Modell eins wird die Dichte des Seewassers als konstant angenommen. Im zweiten Modell wird die natürliche Konvektion berücksichtigt. Dafür wird die Dichte des Seewassers als variabel eingestellt und das Modell der Gravitation hinzugefügt.

5 Validierung

5.1 Validierungskonzept

Um die Qualität der Simulationsergebnisse überprüfen zu können, wird ein Validierungskonzept erarbeitet. Dieses ist in Abbildung 8 symbolisch dargestellt. Grundlage der Validierung ist ein Laborversuchsprüfstand mit einem Versuchswärmeübertrager der Brandenburgischen Technischen Universität Cottbus-Senftenberg (BTUCS). Dieser Übertrager unterscheidet sich in den geometrischen Abmessungen wie Anzahl der Windungen, Länge des Rohres und In-

nendurchmesser des Rohres von dem unter Tabelle 1 spezifizierten Wärmetauscher. In der dynamischen Prozesssimulation wird ebenfalls der Laborversuchswärmeübertrager implementiert. Anschließend wird die gemessene mit der anhand der Prozesssimulation simulierten Rücklaufftemperatur verglichen. Dieser Vergleich wird unter Teilkapitel 5.2 ausgeführt und fällt positiv aus, daher wird angenommen, dass das Modell die Problematik gut abbildet und die Übereinstimmung zwischen Simulation und Realität, aufgrund des starken Abstraktionsgrades auch mit einem anderen Rohrschlangen Wärmeübertrager ausreichend ist. Aufgrund dieser Behauptung kann der reale Rohrschlangen-Wärmeübertrager in das dynamische Prozesssimulationsmodell implementiert werden. Die daraus resultierenden Ergebnisse können dann für die Validierung der CFD-Modelle herangezogen werden.

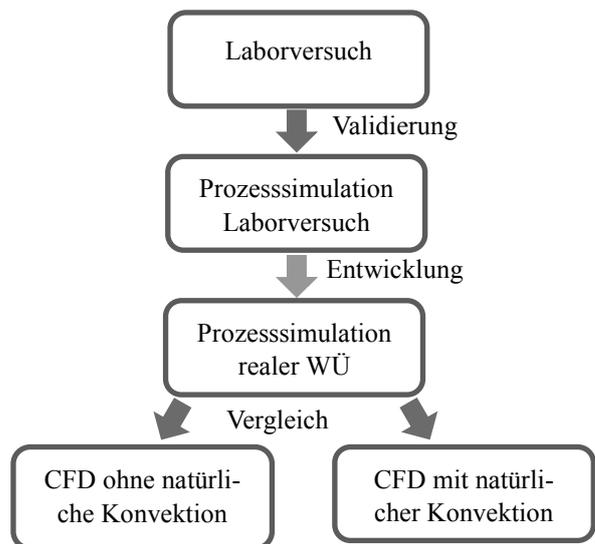


Abbildung 8: Validierungskonzept

5.2 Vergleich Messung und Prozesssimulation

Um die Strömung des Seewassers nachzubilden wird eine Tauchpumpe verwendet. In das dynamische Prozesssimulationsmodell werden die geometrischen Daten des Versuchswärmeübertragers und die Rahmenbedingungen aus Tabelle 2 implementiert und simuliert. Die Ergebnisse dieser Simulation werden mit den Messwerten aus dem Versuch anhand der Rücklaufftemperatur miteinander verglichen. Zwischen der Rücklaufftemperatur der Messung und der Simulation ist wie in Abbildung 10 zu sehen, grundsätzlich eine gute Übereinstimmung vorhanden. Lediglich in den hervorgehobenen Bereichen ist eine leicht erhöhte Abweichung erkennbar. Die statistische

Auswertung des Vergleichs ergibt einen relativen Mittelwertfehler von 0,6 %.

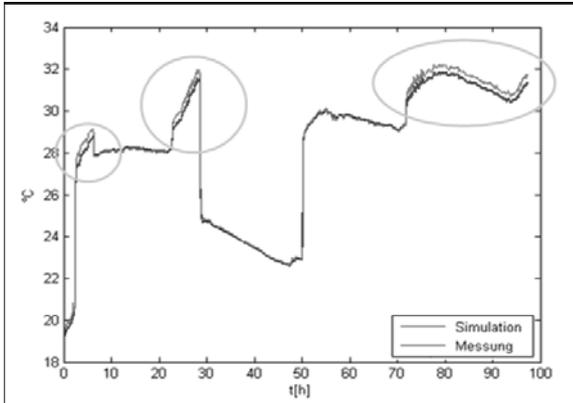


Abbildung 10: Vergleich Messung - Prozesssimulation
Anhand dieses Vergleichs kann gezeigt werden, dass das dynamische Prozesssimulationsmodell, trotz der in Kapitel 3 beschriebenen Modellvereinfachungen eine sehr gute Übereinstimmung mit den Messergebnissen des Laborprüfstands erzielt. Durch die Validierung kann jedoch nicht die allgemeine Anwendbarkeit der Modelle garantiert werden. Dies liegt an verschiedenen Schwächen der Laborversuche. So wurde der WÜ im Versuch nur ideal von vorne angeströmt. Ebenfalls konnte der Fall des stehenden Gewässers aufgrund von Zeitgründen nicht anhand eines Laborversuches bestätigt werden. Diese Einschränkungen führen dazu, dass bei den genannten Rahmenbedingungen eine Anwendbarkeit vermutet, jedoch nicht sichergestellt werden kann.

Tabelle 2: Rahmenbedingungen der Validierungssimulation

Spezifikation Fluid 1 (WTM)	Symbol	Wert	Einheit
Massenstrom	\dot{m}_{WTM}	0,43	kg/s
Fließgeschwindigkeit	v_{WTM}	0,76	m/s
Temperatur am Eintritt	$T_{WTM,ein}$	279,00	K
Spezifikation Fluid 2 (Seewasser)			
Fließgeschwindigkeit	v_{Seew}	0,02	m/s
Temperatur am Eintritt	T_{Seew}	285,50	K
Dichte Seewasser	ρ_{Seew}	999,60	Kg/m ³

6 Ergebnisse

6.1 Erwärmungsverlauf

Mit den erarbeiteten CFD-Modellen kann die Temperaturzunahme des WTMs in jeder einzelnen Windung des Wärmetauschers ermittelt werden. Die Temperaturzunahme in Abhängigkeit der Anzahl an durchströmten Windungen, mit den in Tabelle 2 genannten Rahmenbedingungen, ist in Abbildung 11 dargestellt.

Insgesamt steigt die Temperatur des WTMs um 3,2 K, was einem Anteil von 48,8 % der Temperaturdifferenz zwischen WTM und Seewasser entspricht. Bis zur Mitte, dies entspricht Windung 29, wird eine Erhöhung der Temperatur um 1,9 K erreicht. In der zweiten Hälfte liegt die Zunahme bei 1,30 K. Dies entspricht einer um 30 % verlangsamteten Zunahme. Dies zeigt deutlich, dass eine Erhöhung der Windungszahlen nur begrenzt wirkungsvoll ist und ein Optimum zwischen Materialaufwand und Temperaturerhöhung gefunden werden muss.

Abbildung 11: Temperaturverlauf WTM

6.2 Auswirkungen der Naturkonvektion bei niedrigen Fließgeschwindigkeiten

Um die Auswirkungen der Naturkonvektion des Seewassers betrachten zu können, wurde in einem zweiten Modell die konstante Dichte durch eine temperaturabhängige Dichte ersetzt und die Gravitation berücksichtigt. Mit beiden Modellen wurde anschließend eine Simulation mit einer Fließgeschwindigkeit des Seewassers von 0,001 m/s ausgeführt. Im Fall der konstanten Dichte (Abb. 12) bildet sich um die Rohrschlange ein Bereich mit reduzierter Temperatur, insbesondere auf der strömungsabgewandten Seite ist eine deutliche Verringerung der Temperatur bis auf 282,5 K erkennbar. Dies lässt sich erklären durch das fehlende Absinken des abgekühlten Mediums, denn

Abbildung 12: Temperaturfeld konstante Dichte
daraus resultiert, dass die Bewegung des Seewassers lediglich auf der Fließgeschwindigkeit von diesem basiert. Daraus resultiert ein sehr langsames Fortschreiten, insbesondere auf der strömungsabgewandten Seite der Rohre. Es ergibt sich in diesen Bereichen somit eine lange Verweilzeit, was aufgrund der Wärmeübertragung an das WTM die niedrigen Temperaturen erklärt. In diesen Bereichen ist die treibende Temperaturdifferenz zwischen den Medien mit ca. 3 K gering. Dies führt zu einem signifikant geringeren Wärmestrom im Vergleich zu einer Temperaturdifferenz von 6 K, welche bei guter Durchmischung vorliegen würde.

Im Modell mit variabler Dichte, welcher in Abbildung 13 dargestellt ist, tritt der gerade beschriebene Effekt der Abkühlung nur in deutlich verringerter Form auf. Es ist lediglich auf der strömungsabgewandten Seite eine leicht verminderte Temperatur erkennbar. Generell ist eine Ausbreitung des Bereichs mit vermindelter Temperatur nach unten erkennbar. Dieses Absinken erfolgt aufgrund der verminderten Temperatur und damit Dichte des Wassers. Somit erfolgt ein ständiger Abtransport des abgekühlten Seewassers, was zu einer höheren Temperaturdifferenz von ca. 6 K zwischen den Arbeitsmedien führt. Daraus resultiert ein höherer Wärmestrom im Vergleich zum Modell der konstanten Dichte.

Abbildung 13: Temperaturfeld variable Dichte

6.3 Vergleich der Simulationsmodelle

Um die Auswirkungen der Modellgenauigkeit und Art der Modellierung auf das Simulationsergebnis bei unterschiedlichen Fließgeschwindigkeiten des Seewassers bewerten zu können, wird mit den Modellen eine Testreihe durchgeführt und in Abbildung 14

dargestellt. In der Testreihe, wird die Fließgeschwindigkeit des Seewassers v_{Seew} von 0,001 m/s in sieben Schritten auf 0,1 m/s erhöht.

In Bereich höherer Fließgeschw. gilt $v_{See} \geq 0,2$ m/s. Die Verläufe der Simulationsansätze liefern hier nahezu identische Werte. Für diese Fließgeschwindigkeiten, liegt nur eine geringe Verweilzeit in direkter Rohrumbgebung vor, was lediglich zu einer geringen Temperaturabnahme führt. Dies und die bessere Durchmischung führen dazu, dass die Auswirkungen der Naturkonvektion sehr gering sind und vernachlässigt werden können. Die Abweichung zwischen den Strömungssimulationsmodellen und der Prozesssimulation ist innerhalb des kompletten Bereichs nahezu konstant und beträgt zwischen vier und fünf Prozent. Als Konsequenz kann daraus abgeleitet werden, dass in diesem Bereich die dynamische Prozesssimulation trotz des hohen Abstraktionsgrades sehr gute Übereinstimmung erreicht und aufgrund der erheblich niedrigeren Rechenzeit zu bevorzugen ist.

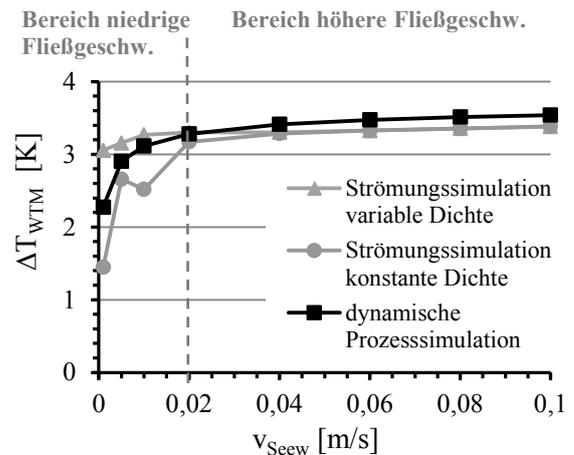


Abbildung 14: Vergleich der Simulationsmodelle für unterschiedliche Fließgeschwindigkeiten v_{Seew}

In Bereich niedriger Fließgeschw. hingegen differieren die Ergebnisse zwischen den Simulationen stark. Grundsätzlich ist eine Abnahme der Temperaturänderung mit abnehmender Fließgeschwindigkeit erkennbar. Diese ist jedoch in den Modellen unterschiedlich stark ausgeprägt. Die Unterschiede zwischen den CFD-Modellen können anhand der Naturkonvektion, wie in Unterkapitel 6.2 beschrieben, erklärt werden, wobei die Dominanz des Effekts mit abnehmender Fließgeschwindigkeit zunimmt. Auch die Abweichung der dynamischen Prozesssimulation ergibt sich aus der Vernachlässigung der Naturkonvektion. So ergibt sich bei der Berechnung ein zu geringer Wert

für die Nußelt-Zahl und den daraus resultierenden konvektiven Wärmeübergangskoeffizienten.

Zur besseren Analyse zeigt Tabelle 3 die absoluten und relativen Abweichungen zwischen den Simulationsmodellen. Die CFD-Simulation mit variabler Dichte wird als Referenz verwendet, da mit dieser die Realität am besten abgebildet werden kann. Betrachtet man die relativen Abweichungen ist zu erkennen, dass diese für das CFD-Modell mit konstanter Dichte am größten sind. Mit steigender Fließgeschwindigkeit wird die Abweichung für beide Modelle signifikant geringer. Das Prozesssimulationsmodell erreicht bei einer Fließgeschwindigkeit von 0,010 m/s eine Abweichung < 5 %.

Tabelle 3: Vergleich der Modelle in Bereich 1

v_{Seew} [m/s]	0,001	0,005	0,010	0,020
1. ΔT_{WTM} mit CFD variable Dichte	3,055	3,155	3,267	3,301
2. ΔT_{WTM} mit CFD konstante Dichte	1,446	2,661	2,517	3,172
Relative Abweichung zu 1 [%]	52,7	25,7	23,0	3,9
3. ΔT_{WTM} mit Prozesssimulation	2,274	2,906	3,115	3,282
Relative Abweichung zu 1 [%]	25,6	7,9	4,7	0,6

Der Vergleich der Modelle für unterschiedliche Strömungsgeschwindigkeiten des umgebenden Gewässers zeigt, dass für $v_{\text{Seew}} < 0,010$ m/s ausschließlich das CFD-Modell mit variabler Dichte anwendbar ist. Für $v_{\text{Seew}} \geq 0,010$ m/s ist die relative Abweichung zwischen CFD mit variabler Dichte und der Prozesssimulation kleiner als 5 Prozent, so dass aufgrund der geringeren Rechenzeit die dynamische Prozesssimulation zu bevorzugen ist.

Als Einschränkung ist zu nennen, dass nur die Anströmung senkrecht von vorne untersucht wurde und die Gültigkeit der Ergebnisse für andere Anströmrichtungen nur vermutet werden kann. So kann vermutet werden, dass die Auswirkungen der Naturkonvektion für $v_{\text{Seew}} < 0,010$ m/s bei seitlicher Anströmung geringer sind, da hier weniger nachfolgende Windungen vorhanden sind, welche beeinflusst werden.

7 Zusammenfassung und Ausblick

Im vorliegenden Beitrag ist die Modellierung für die dynamische Prozesssimulation und CFD-Simulation eines Rohrschlangen-Wärmeübertragers, welcher direkt in einen See integriert ist, beschrieben. Die Anwendbarkeit der Modelle wird anhand einer Gegenüberstellung mit Messergebnissen eines Laborversuches untersucht. Diese Gegenüberstellung zeigt unter den gegebenen Rahmenbedingungen, dass die Modellbildung und die daraus resultierenden Modelle korrekt sind. Um die Auswirkungen der Naturkonvektion betrachten zu können, wurden Simulationen mit unterschiedlichen Fließgeschwindigkeiten des Seewassers ausgeführt und die Ergebnisse gegenübergestellt. Im stehenden Gewässer ($v_{\text{Seew}} < 0,010$ m/s) ist der Einfluss der Naturkonvektion groß, was in diesem Bereich nur die Verwendung der CFD mit variabler Dichte erlaubt. Für größere Fließgeschwindigkeiten sind die Auswirkungen der Naturkonvektion gering und die dynamische Prozesssimulation ist aufgrund der sehr niedrigen Rechenzeit zu bevorzugen.

Mit der Bestätigung der dynamischen Prozesssimulation konnte gezeigt werden, dass diese auch bei starker Abstraktion des Problems anwendbar ist und dabei signifikante Einsparungen in Bezug auf Rechenzeit möglich sind. Daraus kann gefolgert werden, dass es nicht immer zielführend ist, die Realität mit einem sehr aufwändigen Modell möglichst exakt abzubilden. Bei der Untersuchung von Wärmeübertragungsproblemen kann auch, frei nach dem Sprichwort „Nicht mit Kanonen auf Spatzen schießen“, auch eine „einfache“ Lösung zum Erfolg führen.

8 References

- [1] Umweltbundesamt. *Energieverbrauch nach Energieträgern und Sektoren*. Berlin 2015. Link: <http://www.umweltbundesamt.de/daten/energiebereitstellung-verbrauch/energieverbrauch-nach-energetraegern-sektoren>
- [2] Bundesministerium für Wirtschaft und Energie (BMWi). *Endenergieverbrauch nach Anwendungsbereichen*. Berlin 2015. Link: <http://www.bmwi.de/DE/Themen/Energie/Energiedaten-und-analysen/Energiedaten/energiegewinnung-energieverbrauch.html>

- [3] H. Stopp und P. Strangfeld. *Schwimmende Wohnbauten*. Beuth Verlag, Deutschland, 2012.
- [4] P. Junglas. *Praxis der Simulationstechnik*. Haan-Gruiten Verlag, 2014 .
- [5] W. Polifke und J. Kopitz. *Wärmeübertragung*. Pearson Studium, München, 2009.
- [6] Verein Deutscher Ingenieure. *VDI-Wärmeatlas*. Springer, Deutschland, 2013
- [7] USER GUIDE Star CCM+ Version 9.04. CD-adapco, 2014.

Nichtlineare Simulation zur Funktionsauslegung einer aktiven Fahrzeugfederung

Sven Jacobitz¹, Xiaobo Liu-Henke²

Ostfalia Hochschule für angewandte Wissenschaften, Fakultät Maschinenbau,
Institut für Mechatronik, Salzdhumer Str. 46/48, 38302 Wolfenbüttel

¹*sve.jacobitz@ostfalia.de*, ²*x.liu-henke@ostfalia.de*

Zur Lösung des Zielkonflikts zwischen Fahrkomfort und -sicherheit wird eine aktive Fahrzeugfederung eingesetzt. Der vorliegende Beitrag fokussiert die Konzeption der aktiven Fahrzeugfederung für ein Elektrofahrzeug. Zur Entwicklung eines solchen technischen Systems wird die modellbasierte Methode herangezogen und bildet den Schwerpunkt dieser Arbeit. Ein nichtlineares MKS-Fahrzeugmodell mit sieben Freiheitsgraden zur Repräsentation der dazugehörigen Kinematik- und Dynamikfunktion des Fahrzeugs wird hergeleitet. Nach der Modellvalidierung wird die hierarchisch angeordnete Reglerstruktur beschrieben werden. Nach der Analyse des geregelten Systemverhaltens wird das Verbesserungspotenzial der aktiven Federung mittels der nichtlinearen Simulation unter Einsatz des entwickelten nichtlinearen MKS-Modells aufgezeigt.

1 Einleitung

Der Einsatz erneuerbarer Energien in der Mobilität ist eines der wichtigsten Elemente zur Erreichung der Klimaschutzziele. In diesem Zusammenhang steigt die Anzahl der neu zugelassenen Elektrofahrzeuge exponentiell [1]. Besonders die Technologie der Direktantriebe bietet einen hohen Wirkungsgrad sowie eine Vielzahl an regelungstechnischen Möglichkeiten. Um diese methodisch und effizient zu nutzen, wird an der Ostfalia Hochschule der Funktionsträger für regenerative Elektromobilität und Fahrdynamik (FREDY) mittels eines ganzheitlichen, durchgängig modellbasierten Entwurfsprozesses entwickelt [2].

Insbesondere bei Fahrzeugen mit Direktantrieben erweist sich die Auslegung eines passiven Fahrwerks aufgrund der erhöhten ungefederten Massen problematisch. Hierbei verschärft sich der Zielkonflikt zwischen Fahrkomfort und -sicherheit [3]. Durch Einsatz einer aktiven Federung lässt sich dieser Zielkonflikt lösen.

Der vorliegende Beitrag fokussiert die Modellierung eines gesamten Fahrzeugs zur Auslegung einer Mehrgrößenregelung, welche die Freiheitsgrade Huben (z), Wanken (κ) und Nicken (φ) des Fahrzeugaufbaus aktiv beeinflusst.

2 Methodik und Lösungsansatz

Moderne Produkte stellen aufgrund steigender Anforderungen mechatronische Systeme aus verschiedenen Fachdomänen der Mechanik, Elektrotechnik und Informationstechnik dar. Um die hieraus entstehende Komplexität zu bewältigen, ist es notwendig eine ganzheitliche Betrachtung durchzuführen [4]. Zur Minderung zeitintensiver Iterationsschleifen finden immer mehr modellbasierte Entwurfsverfahren Anwendung, da hierbei bereits in frühen Phasen des Prozesses die Wechselwirkungen zwischen den Systemkomponenten berücksichtigt werden.

Um diese Anforderungen zu erfüllen, wird der mechatronische Entwicklungsprozess nach [5] angewendet. Hierbei erfolgt zunächst die hierarchische Strukturierung des Gesamtsystems in die vier Funktionsebenen vernetztes mechatronisches System (VMS), autonomes mechatronisches System (AMS), mechatronische Funktionsgruppe (MFG) und mechatronisches Funktionsmodul (MFM) mittels top-down Verfahren. Anschließend wird das System sukzessive von unten nach oben mittels mechatronischer Komposition entworfen. Dieses Vorgehen ermöglicht die Entwicklung einzelner Funktionen und Funktionsgruppen unabhängig vom Gesamtsystem. Für die Auslegung dieser Module wird ein verifikationsorientierter Kreislauf, bestehend aus den Prozessen Model-in-the-Loop (MiL), Software-in-the-Loop (SiL) und Hardware-in-the-Loop (HiL), eingesetzt (siehe Abbildung 1).

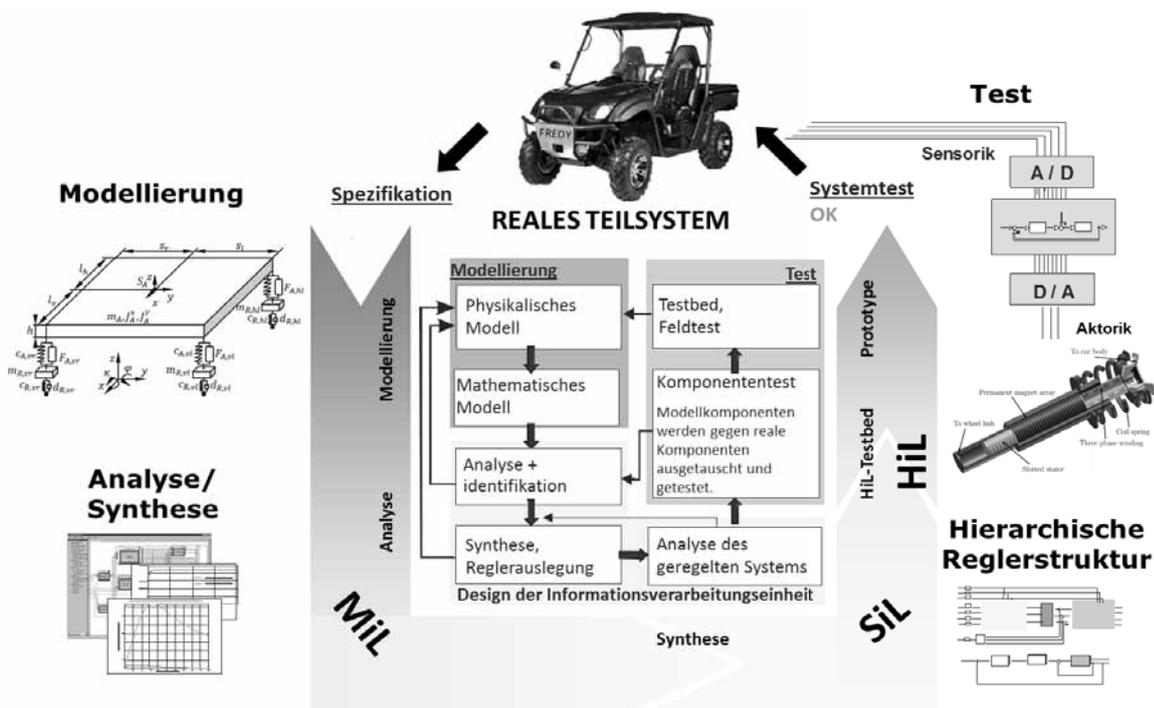


Abbildung 1. Mechatronischer Entwicklungskreislauf nach [5]

2.1 Strukturierung des Systems

Die aktive Fahrzeugfederung ist ein Bestandteil des elektronischen Fahrwerksmanagements. Abbildung 2 stellt die mechatronische Funktionsstruktur des aktiven Federungsmoduls mit den verwendeten Schnittstellen dar. Vom Fahrzeugmanagement werden die Sollgrößen der Aufbaubewegung bereitgestellt, welche durch die globale Aufbauregelung in Sollkräfte der lokalen Regelungen der Federbeine verarbeitet werden. Diese werden durch die untergeordneten lokalen Regelkreise eingeregelt.

2.2 Konzept der aktiven Fahrzeugfederung

Die Federung ist eines der wichtigsten Elemente eines modernen Kraftfahrzeugs [6]. Durch eine weiche Federung wird der Fahrzeugaufbau und deren Insassen von den Straßenebenenheiten isoliert und somit der gewünschte Fahrkomfort sichergestellt. Aber eine weiche Radaufhängung stellt große Radlastschwankungen dar, welche zu einer Minderung der Fahrsicherheit führen. Eine hohe Fahrsicherheit kann zwar durch eine hohe Dämpfung erreicht werden. Diese Maßnahme führt jedoch zu einem verschlechterten Fahrkomfort. Der Fahrkomfort und die Fahrsicherheit sind somit nicht unabhängig voneinander beeinflussbar und bilden einen Zielkonflikt, welcher durch die Optimierung passiver Systeme nicht gelöst werden kann.

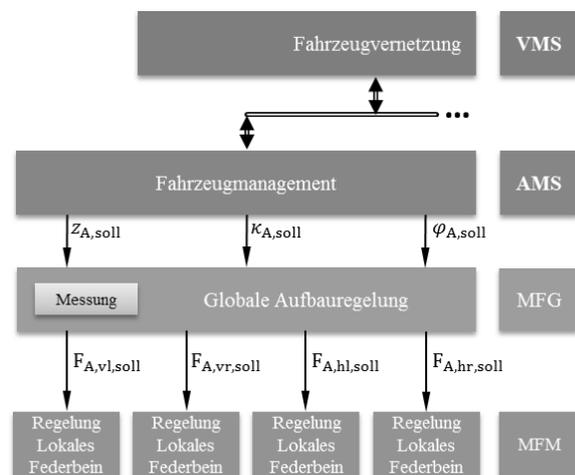


Abbildung 2. Mechatronische Hierarchisierung der aktiven Federung

Um diesen Zielkonflikt zu lösen, werden vermehrt aktive Federungssysteme eingesetzt [8], welche sich unter Einsatz von Regelung, Aktorik und Sensorik realisieren lassen. Solch aktive Systeme sind aufgrund der hohen Kosten und des hohen Energiebedarfs jedoch selten in Serienfahrzeugen eingesetzt. Durch Weiterentwicklung entsprechender Aktorik und der Möglichkeit der Energierückgewinnung kann sich diese Situation ändern [9].

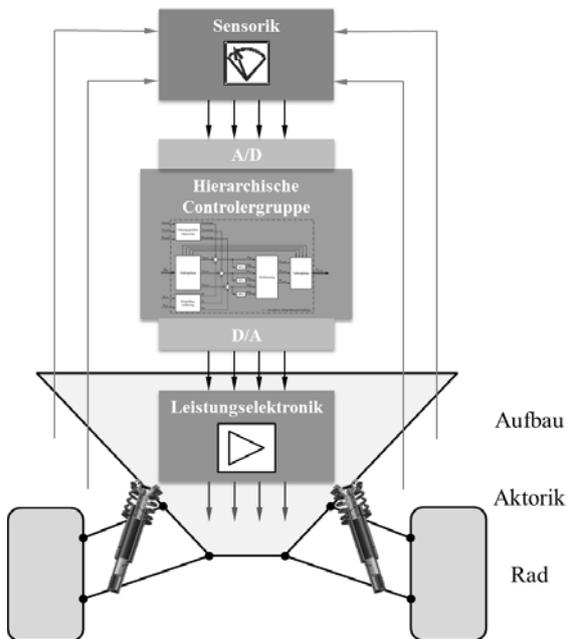


Abbildung 3. Prinzipieller Aufbau des aktiven Fahrwerks

Abbildung 3 zeigt das Konzept einer aktiven Federung für den Funktionsträger FREDY. Um einen hohen Fahrkomfort bei gleichzeitig hoher Fahrsicherheit zu erzielen, wird das konventionelle Federbein, welches in der Regel aus passiver Feder und passivem Dämpfer besteht, durch einen integrierten elektrischen Aktor ersetzt. Hierbei handelt es sich um eine Parallelschaltung aus passiver Feder und elektrischer Synchronmaschine. Durch diese Anordnung können zusätzliche Kräfte zwischen Aufbau und Rad eingebracht werden,

um die gewünschte Feder- und Dämpfungscharakteristik aktiv einzustellen. Der Kern des aktiven Systems ist eine hierarchisch angeordnete Reglerstruktur, welche die Ansteuerungssignale der Aktoren bestimmt. Durch diese aktive Federungstechnik kann die physikalische Grenze verschoben und damit ein besseres und neues Systemverhalten erreicht werden, da die gewünschte Feder- / Dämpfungskraft im entsprechenden Frequenzbereich durch aktive Eingriffe erzielt werden kann. Somit können ein hoher Fahrkomfort und gleichzeitig eine hohe Fahrsicherheit durch diese aktive Stabilisierung aller Aufbaubewegungen gewährleistet werden.

3 Modellbildung

Für den beschriebenen Entwurfsprozess wird das Fahrzeug als Fünfmassenschwinger betrachtet, welcher aus einer Aufbaumasse und vier ungefederten Massen (Rädern) besteht (Abbildung 4a). Die Räder besitzen jeweils einen Freiheitsgrad in Vertikalrichtung, während der Aufbau die Freiheitsgrade Huben (z), Wanken (κ) und Nicken (φ) aufweist. Die Räder sind jeweils durch eine Parallelschaltung bestehend aus einem nichtlinearen Federelement, einem passiven Dämpfer und einem elektrischen Linearaktor mit der Aufbaumasse verknüpft.

Bei der Modellbildung kann von einem symmetrischen Fahrzeug ausgegangen werden, sodass die Wankachse mittig des Fahrzeugs liegt [10]. Die Wank- und Nickachse sind in das Koordinatensystem des Aufbaus definiert.

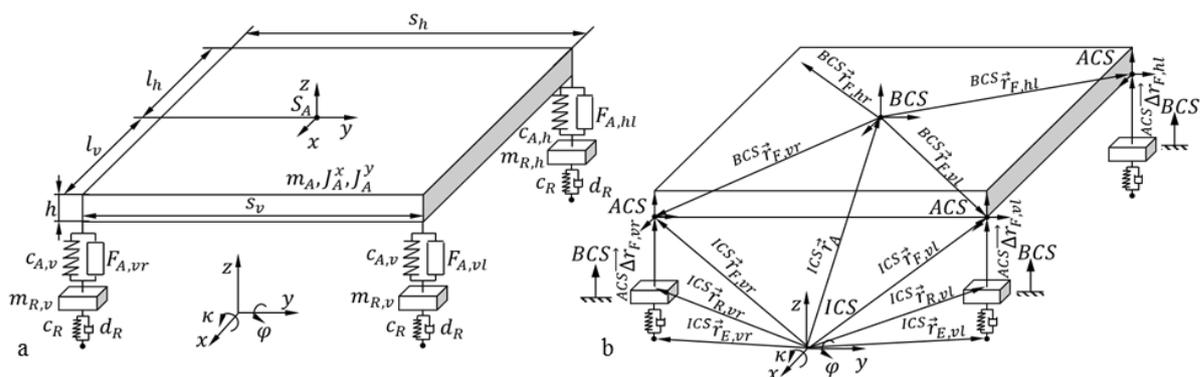


Abbildung 4. Ersatzmodell zur Herleitung der Kinematik- und Dynamikfunktion

3.1 Kinematikfunktion

Die Kinematikfunktion des Fahrzeugs wird für die Ent- und Verkopplung der Globalregelung benötigt. Zur Herleitung müssen die verwendeten Koordinatensysteme definiert werden. Hierbei wird zwischen dem

Inertialkoordinatensystem (ICS), den körperfesten Koordinatensystemen (BCS) und Kontaktpunktkoordinatensystemen (ACS) unterschieden. Die ACS werden jeweils an den Kontaktpunkten der Federbeine mit

der Aufbaumasse und der jeweiligen Radmasse definiert. Die BCS werden jeweils im Schwerpunkt der Massen angenommen. Für die Definition des ICS wird zunächst das Straßenprofil näher betrachtet. Dies kann in ein sich langsam änderndes Grundprofil, welchem das Fahrzeug folgen soll und ein sich schneller änderndes Störprofil, gegen welches das Fahrzeug isoliert werden soll, eingeteilt werden. Wird ein Koordinatensystem auf die Oberfläche des Grundprofils gelegt, ändert sich dessen Position nur langsam und es kann als quasiinertial bezeichnet werden [11]. Das ICS wird somit direkt unter den Schwerpunkt der Aufbaumasse auf der Oberfläche des Grundprofils definiert. Abbildung 4b stellt das physikalische Modell mit den kinematischen Beziehungen dar.

Die direkte Kinematik bzw. die inverse Kinematik des Systems wird unter Verwendung von Drehmatrizen \underline{T} durch Koordinatentransformation berechnet und später in die globalen Regelkreise eingebettet.

3.2 Dynamikfunktion

Die Dynamikfunktion beschreibt die Bewegung der Aufbau- und Radmassen aufgrund der wirkenden Kräfte.

Abbildung 5 illustriert die Kräftebilanz.

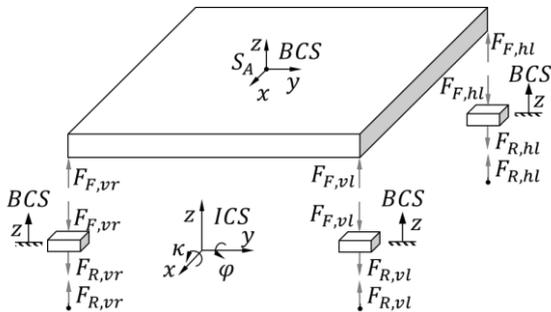


Abbildung 5. Freigeschnittenes Fahrzeugmodell

Aus dem physikalischen Ersatzmodell werden Bewegungsgleichungen nach Newtons Ansatz für beide symbolisch abgeleitet. Dabei wird von den Beschleunigungen und Winkelbeschleunigungen des Aufbaus und jeder Radmasse im Inertialkoordinatensystem

$$\begin{aligned} {}^{ICS}\vec{r}_A &= [z_A \quad \kappa_A \quad \varphi_A]^{-T} \\ {}^{ICS}\vec{r}_R &= [z_{R,vr} \quad z_{R,vl} \quad z_{R,hr} \quad z_{R,hl}]^{-T} \end{aligned} \quad (1)$$

ausgegangen. Die Differenzialgleichungen ergeben sich durch Gleichsetzen von Massen- und dem Trägheitstensor \underline{J} mit den von den Federn und Dämpfern am Schwerpunkt der jeweiligen Massen ausgeübten Kräften und dem Moment:

$$\begin{aligned} \begin{bmatrix} m_A & 0 & 0 & 0 \\ 0 & J_A^x & 0 & 0 \\ 0 & 0 & J_A^y & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{z}_A \\ \ddot{\kappa}_A \\ \ddot{\varphi}_A \end{bmatrix} &= \underline{T}_F(\kappa_A, \varphi_A) \vec{F}_F \\ \begin{bmatrix} m_{R,v} & 0 & 0 & 0 \\ 0 & m_{R,v} & 0 & 0 \\ 0 & 0 & m_{R,h} & 0 \\ 0 & 0 & 0 & m_{R,h} \end{bmatrix} \begin{bmatrix} \ddot{z}_{R,vr} \\ \ddot{z}_{R,vl} \\ \ddot{z}_{R,hr} \\ \ddot{z}_{R,hl} \end{bmatrix} &= -\vec{F}_F - \vec{F}_R \end{aligned} \quad (2)$$

Die Matrix $\underline{T}_F(\kappa_A, \varphi_A)$ beschreibt hierbei die Transformation der Federbeinkraftbeträge \vec{F}_F in Kräfte und Momente im Aufbau BCS.

3.3 Systemanalyse mittels Simulation

Die Analyse des Modells erfolgt im Frequenz- und Zeitbereich. Hierbei wird zusätzlich eine Plausibilitätsauswertung mittels eines vorhandenen Gesamtfahrzeugs [12] als Referenzmodell durchgeführt. Zur Darstellung im Frequenzbereich wird Gl. (2) linearisiert und als Zustandsraummodell mit 14 Zustandsgrößen dargestellt:

$$\begin{aligned} \dot{\vec{x}} &= \underline{A}\vec{x} + \underline{B}\vec{u} \\ \vec{y} &= \underline{C}\vec{x} + \underline{D}\vec{u} \end{aligned} \quad (3)$$

Als Zustandsgrößen \vec{x} werden die Freiheitsgrade der Aufbau- und Radmassen sowie deren Geschwindigkeiten gewählt. Eingangsgröße des Systems ist die Straßenunebenheit, welche sich als Vertikalanregung z_E , Wankanregung κ_E und Nickanregung φ_E darstellen lässt.

Aus den Zustandsgleichungen kann nun die Übertragungsmatrix des Systems \underline{G} bestimmt werden:

$$\underline{G} = \underline{C}(\underline{sI} + \underline{A})^{-1} \underline{B} + \underline{D} \quad (4)$$

wobei \underline{I} die Einheitsmatrix darstellt. Aus dieser Übertragungsmatrix lassen sich die Frequenzgänge des Systems errechnen.

Abbildung 6 stellt die berechneten Frequenzgänge des linearisierten Modelles denen eines passiven Gesamtfahrzeugmodells (Referenzmodell) gegenüber. Im Vergleich zu einem konventionellen Fahrwerk ist hierbei die Dämpfung schwächer ausgelegt. Die Frequenzgänge des linearisierten Modells sind plausibel und stimmen gut mit denen des Referenzmodells überein. Die Eigenfrequenz des Aufbaus liegt in Vertikalrichtung bei ca. 1,3 Hz und ist mäßig gedämpft. In Wank- und Nickrichtung liegt die Eigenfrequenz bei ca. 1 Hz und ist wie in Vertikalrichtung mäßig gedämpft. Die Radeigenfrequenzen sind bei ca. 15 Hz zu erkennen. Nach [10] liegen diese Werte in einem für Pkw normalen Bereich.

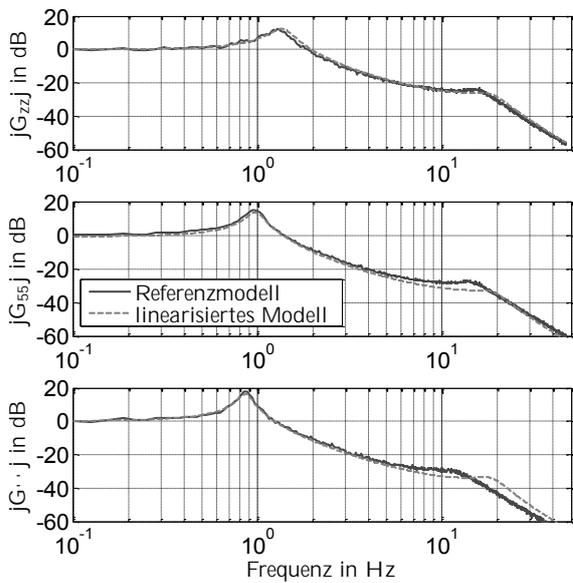


Abbildung 6. Vergleich der Frequenzgänge von Referenzmodell und linearisiertem Modell

Abbildung 7 stellt die Sprungantworten der nichtlinearen Simulation im Zeitbereich anhand des aufgestellten nichtlinearen MKS-Modells dar. Die Sprungantworten bei Straßenanregung in Vertikal-, Wank- und Nickrichtung zeigen ebenfalls eine gute Übereinstimmung. Die Verkopplung zwischen den betrachteten Freiheitsgraden des Aufbaus aufgrund der physikalischen Gegebenheiten ist klar ersichtlich und im Modell sehr gut abgebildet.

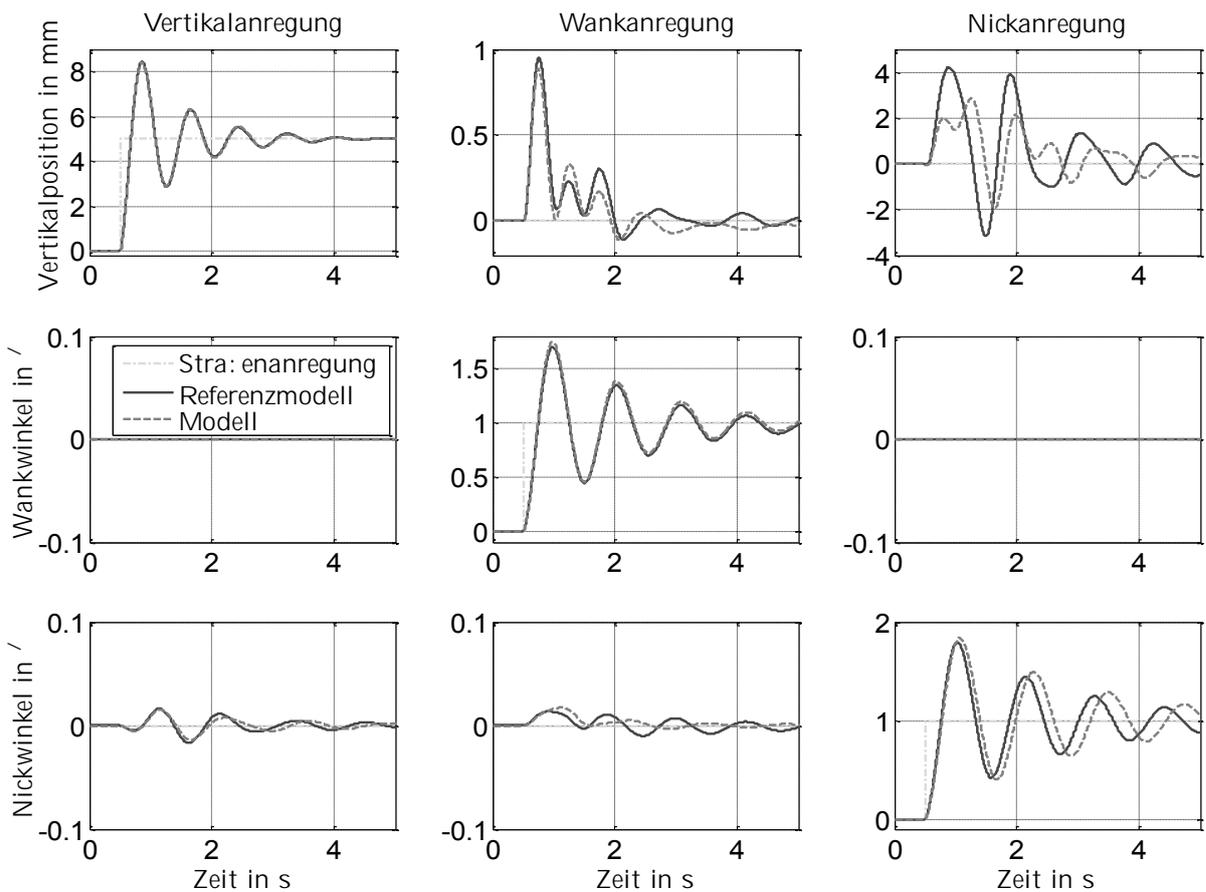


Abbildung 7. Sprungantworten bei Anregung durch Straßenunebenheiten

4 Modellbasierte Reglersynthese

Nach Analyse des passiven Systems kann die Reglerstruktur so ausgelegt werden, dass das geforderte dynamische Verhalten des Systems erzielt wird. Abbildung 8 illustriert die hierarchische Reglerstruktur. Hierfür wird das verallgemeinerte Kaskadenprinzip angewendet [5]. Diese Mehrgrößenregelung lässt sich in eine globale Aufbauregelung und eine lokale Regelung einteilen. Im Folgenden wird im weiteren Verlauf näher erläutert.

4.1 Globale Reglerstruktur

Wie in Kapitel 3.3 gezeigt wurde, stellt die Dynamik des Aufbaus ein Mehrgrößenproblem dar, bei dem die drei betrachteten Freiheitsgrade durch die physikalische Beschaffenheit stark miteinander verkoppelt sind. Daher liegt die Grundidee für die Auslegung der Mehrgrößenregelung der Aufbaudynamik darin, die miteinander verkoppelten Bewegungen zunächst durch eine Entkopplung in die jeweiligen quasimodalen Bewegungen zu separieren. Damit kann die Auslegung der Mehrgrößenregelung auf die einfache Behandlung der Eingrößensysteme zurückgeführt werden. Jede Bewegung kann separat betrachtet und ein Regler für jede Bewegung ohne Rücksicht auf die anderen optimal ausgelegt werden. Darauf aufbauend wird ein Zustandsregler in Kombination mit dem Skyhook-Dämpfer-Ansatz zur Regelung jeder einzelnen Bewegung eingesetzt.

Abbildung 8 zeigt die Struktur der Globalregelung für die Aufbaudynamik. Die Regelstruktur besteht hauptsächlich aus drei Blöcken: Entkopplung, Abstimmung und Verkopplung.

Die Entkopplung hat die Aufgabe, die quasimodalen Vertikalwege, Wank- und Nickwinkel des Aufbaus basierend auf der Systemkinematik (Kapitel 3.1) aus den gemessenen Positionen der Federbeine abzuleiten. Daraus werden die gewünschten Sollkräfte für die aktiven Federbeine mit vorgegebenen Vertikal-, Wank- und Nickfedern bzw. -dämpfungen im Block Abstimmung unter Verwendung des Skyhook-Prinzips generiert. Anschließend werden die Stellsignale für die lokalen Federbeine im Block Verkopplung, welcher ebenfalls auf der Systemkinematik basiert, für die unterlagerten Regelkreise generiert.

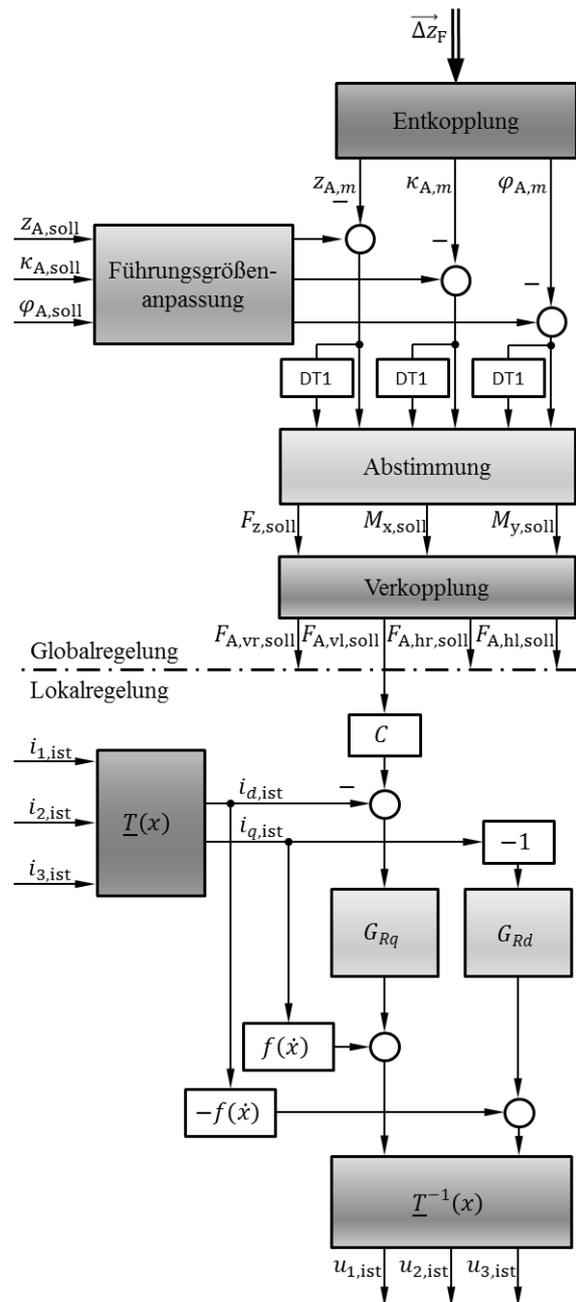


Abbildung 8. Hierarchische Reglerstruktur der aktiven Fahrzeugfederung

Die Reglerparameter dieser Globalregelung werden durch Polvorgabe in einem Optimierungsprozess ausgelegt. Abbildung 9 zeigt die Optimierungsergebnisse. Die Sprungantworten des Systems in allen betrachteten Bewegungsrichtungen weichen nur mit geringem Fehler von den Referenzsignalen ab. Die daraus gelieferten Regelparameter führen zu den gewünschten Modalkräften, womit sowohl hoher Fahrkomfort als auch hohe Fahrsicherheit erreicht werden können.

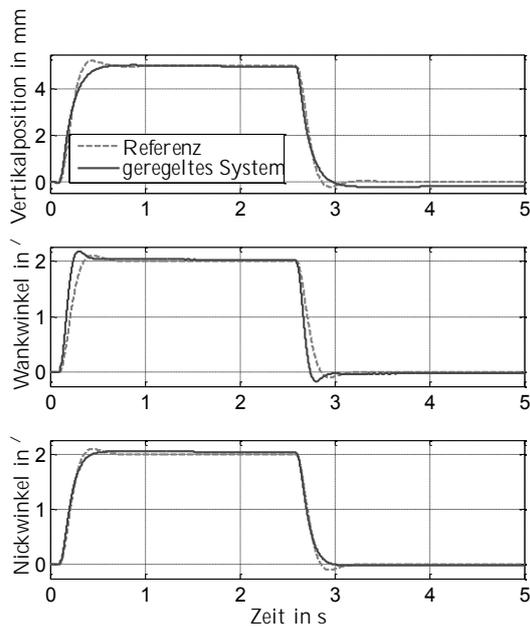


Abbildung 9. Ergebnisse der Optimierung

4.2 Lokale Reglerstruktur

Eine Aufgabe der Lokalregelung ist es, die Kraft exakt zu dosieren. Dies kann dank Charakteristik der linearen Synchronmaschine ohne großen Aufwand in eine

Stromregelung realisiert werden. Ähnlich wie bei der feldorientierten Regelung einer rotativen Maschine wird auch hier eine Clarke- / Park-Transformation $T(x)$ durchgeführt und das dreiphasige System so auf zwei gekoppelte Gleichstrommaschinen reduziert. Durch diese Transformation kann die Auslegung der Regelung nach den bekannten Verfahren erfolgen.

5 Analyse des geregelten Systems

Zur Funktionsabsicherung werden zahlreiche nichtlineare Simulationen im Zeitbereich durchgeführt. Die Systemantworten bei sprungförmiger Anregung durch Straßenunebenheiten werden durch Abbildung 10 illustriert. Als Vergleich wird die Antwort eines konventionellen Fahrzeugs auf dasselbe Anregungssignal unter gleichen Betriebsbedingungen herangezogen.

Das geregelte System zeigt im Vergleich zum konventionellen Fahrwerk ein deutlich verbessertes dynamisches Verhalten. Die Verkopplung der Freiheitsgrade wurde durch die Regelung fast vollständig entkoppelt, was insbesondere bei Betrachtung der Vertikalposition bei Wank- und Nickanregung sichtbar ist.

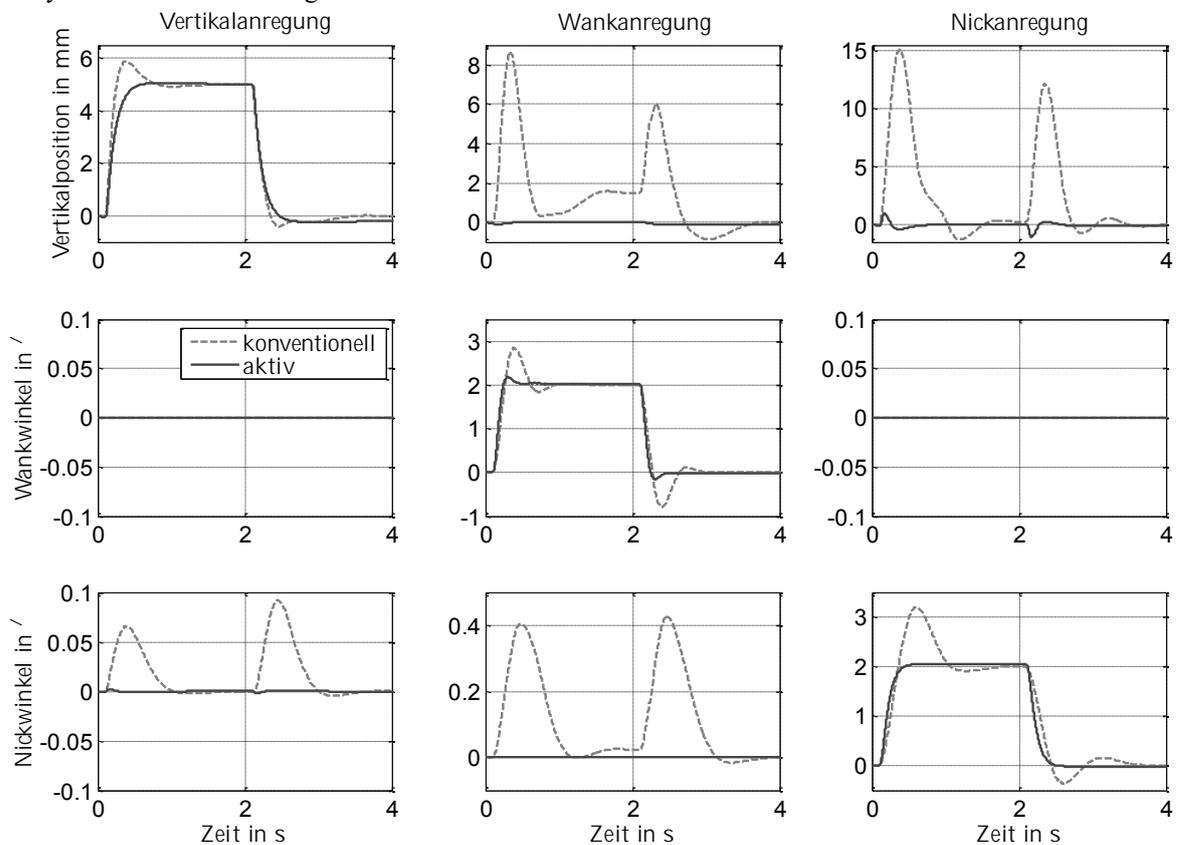


Abbildung 10. Störsprungantworten bei Vertikal-, Wank- und Nickanregung durch Straßenunebenheiten

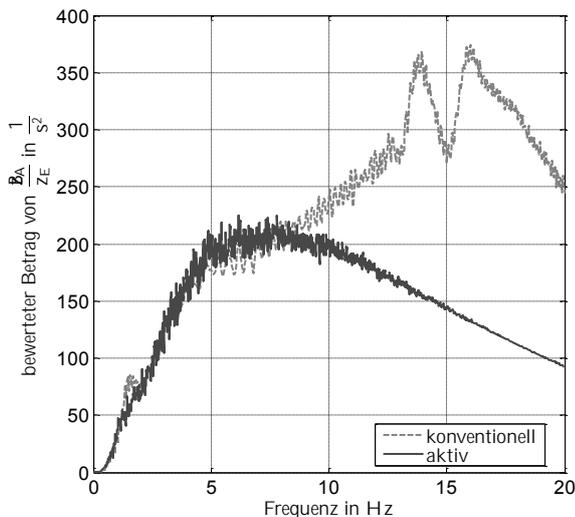


Abbildung 11. frequenzabhängige Aufbaubeschleunigung des konventionellen und des aktiven Fahrwerks

Abschließend wird die Steigerung des Fahrkomforts durch die aktive Federung analysiert. Hierzu ist in Abbildung 11 die frequenzabhängige, nach DIN EN ISO 8041 bewertete, Aufbauvertikalbeschleunigung dargestellt. Je geringer die Fläche unter der Beschleunigungskurve, desto höher ist der Fahrkomfort. Der Fahrkomfort wird um ca. 54% gesteigert. Gleichzeitig kann festgestellt werden, dass der quadratische Mittelwert der dynamischen Radlastschwankung durch die aktive Federung im Vergleich zu Konventionellen Fahrwerken um ca. 65% gesenkt wurde, was eine deutlich erhöhte Fahrsicherheit gewährleistet.

6 Zusammenfassung

Im vorliegenden Beitrag wurde ein nichtlineares Fahrzeugmodell zum Entwurf einer aktiven Fahrzeugfederung mittels mechatronischem Entwicklungsschema hergeleitet. Anhand eines Referenzmodells erfolgte eine Plausibilitätsprüfung im Frequenz- und Zeitbereich. Aufgrund der Modelleigenschaften konnte eine geeignete Reglerstruktur abgeleitet und hinsichtlich einer Verbesserung des Fahrkomforts sowie der Fahrsicherheit ausgelegt werden. Die Analyse mittels nichtlinearer Simulation zeigte, dass der Fahrkomfort und die Fahrsicherheit des Elektrofahrzeugs durch die ausgelegte aktive Fahrzeugfederung im Gegensatz zu einem vergleichbaren konventionellen Fahrwerk wesentlich verbessert wurden. Anhand dieser Erkenntnisse können in Zukunft weitere Analysen hinsichtlich realisierbarem Aktorkonzept und Energierückgewinnung sowie die Erprobung an einem geeigneten Prüfstand durchgeführt werden.

Danksagung

Diese Arbeit wurde im Rahmen eines Forschungsprojekts des Niedersächsischen Ministeriums für Wissenschaft und Kultur durchgeführt. Für die Förderung bedanken sich die Autoren herzlichst.

Gefördert durch:



**Niedersächsisches Ministerium
für Wissenschaft und Kultur**

7 References

- [1] Schott, B. et al.: *Entwicklung der Elektromobilität in Deutschland im internationalen Vergleich und Analysen zum Stromverbrauch*, Fortschrittsbericht, Stuttgart 2013
- [2] Liu-Henke, X. et al.: *FreDy - An electric vehicle with intelligent chassis-control systems*. 10th International Conference on Ecological Vehicles and Renewable Energies (EVER), S. 1–8, 2015
- [3] Gysen, B. L. et al.: *Active electromagnetic suspension system for improved vehicle dynamics*. Vehicle Power and Propulsion Conference (VPPC), S. 1156–1163, 2008
- [4] Koch, T.: *Integration von Konstruktion und mechatronischer Komposition während des Entwurfs mechatronischer Systeme am Beispiel eines integrierten Radmoduls*. VDI Fortschritt-Berichte Reihe 20, Rechnerunterstützte Verfahren, Nr. 401. Düsseldorf: VDI-Verlag 2005
- [5] Liu-Henke, X.: *Mechatronische Entwicklung der aktiven Feder-Neigetchnik für das Schienenfahrzeug RailCab*. VDI Fortschritt-Berichte Reihe 12, Verkehrstechnik / Fahrzeugtechnik, Nr. 589. Düsseldorf: VDI-Verlag 2005
- [6] Le Nguyen, H. et al.: *Modified Skyhook Control of a Suspension System with Hydraulic Strut Mount*. ICROS-SICE International Joint Conference 2009
- [7] Jacobitz, S.: *Modellbasierter Entwurf einer aktiven Federung für ein Elektrofahrzeug*. Masterarbeit, Ostfalia Hochschule, Wolfenbüttel, Deutschland 2015
- [8] Pfeffer, P. E.: *Aktive Fahrwerksysteme*. ATZ - Automobiltechnische Zeitschrift Nr. 06, S. 444–451, 2011

- [9] Dragon, L.: *Fahrzeugdynamik. Wohin fahren wir?* Forschung für das Auto von morgen, S. 239–260, 2008
- [10] Mitschke, M. und Wallentowitz, H.: *Dynamik der Kraftfahrzeuge*. Springer-Verlag, Deutschland, 2014
- [11] Hestermeyer, T.: *Entwurf, Implementierung und Realisierung einer modellgestützten Regelung für ein Fahrzeug mit volltragendem aktivem Fahrwerk und Motorpumpeneinheit*. Diplomarbeit, Universität Paderborn, Deutschland 2000
- [12] Buchta, R.: *Mechatronische Entwicklung eines Forschungselektrofahrzeugs zur Erprobung von Fahrdynamikregelungen und Fahrerassistenzsystemen*. Berichte aus der Fahrzeugtechnik. Aachen: Shaker Verlag, 2005

Dynamics of the mechanical parts in mechatronics systems with VHDL AMS

Dr.Sergey Petkun

Brose Fahrzeugteile GmbH & Co. Kommanditgesellschaft, Hallstadt

Sergey.Petkun@Brose.com

The modelling of mechatronic system includes always some executive mechanisms which are controlled by the “intelligent” electronics with energy drive. For mechanical part of mechatronic system is vital essential dynamical behaviour: the mechatronics doesn't exist without movement. Up today there is a big gap in the thinking between the domains of mechatronics what has many different reasons for it like educational background, structural membership, and available tools and so on. Every domain has own proper tool which is optimized for domain specific task. At the system level the focus of analysis is in properly functionality the whole system and the high abstraction is asked to be able don't fall down too deep in details of specific domains. The mechatronics lives for movement, which from mechanical point of view means the dynamics. The high abstraction level in mechanics leads intuitively to behaviour modelling. For description the dynamical behaviour of mechanical system should be taken into account the most important mechanic properties: inertia, elasticity and forces. It shouldn't be forgotten about the different kinds of movement: translational and rotational. With such rough approach it is clear which mechanical elements (models) are needed first of all for the modelling of dynamics with VHDL AMS: masses, springs (elasticity), friction (losses and transfer from translation to rotational). On some simple example dynamical behaviour of friction is more detailed studied.

1 Introduction

Mechatronics systems consist as a rule from four common parts: mechanics, drives, electronics and software. Every domain has its own development philosophy, tools, and focus points, which are dependent from requirements to system. These requirements, usually summarized in specification, are not always precise enough defined, because of there are the many-sided possibilities in life situations. At present there are two contrary trends in the industrial development: on one side universalization and on the other side specialization. The first trend is forced with growing process the globalisation of the world. It is direct connected with development of mechatronic system. The specialization means the concentration at single domain like mechanics or electronics. The electronics is often already merging with software. There are very seldom the companies having all four domain under own development and production. Today development is very

often separated from its manufactures. The structured boundaries between the domains on organization level exist even in company having all domains. The assembling of the domain specific component occurs at first on the management level where more business aspects are laying usual in focus than technique. All it makes more difficult to organize of reasonable assembly the components in some mechatronic system. From technical point of view, the assembly corresponds to system level. Besides the structural complexity on the development the mechatronic system, the dynamics is a key word for such systems. That means that the system is changing its state during the time. The mechatronic system exists only due to dynamics. The view at the movement of mechatronics device depends from domain and level. For mechanics, the most important is the positions of different part of mechanical component and acting forces on it. For electronics is in interests the position only of controlled parts of the same system and changing of currents and volt-

ages in its electrical components and so on. At system level a proper functionality is in focus analysis. That included corresponding position and reasonable behaviour of electrical part with smart functional logic in all range of working conditions and by all component tolerances. The system level view cannot allow going down into all details which are interest for domain. There is a danger to lose the overview at system as whole one. It leads to simplify of system to essentials for analysis. To reduce means not to avoid the most details but rather to substitute the complex precise description of component through simplified behaviour, reflecting the interesting points of it. We assume the rest of part (all exclusive mechanics) is already available in VHDL AMS. Now the task is to add in system mechanical part to complete the system.

2 Behaviour of mechanical parts on dynamics

In mechanics there are not many basic terms to describe the dynamics behaviour. They are masses and forces. The whole our studies are valid for Newton's mechanics. The description of system behaviour the mechatronic device should give answer on the question where, when and how. For it is necessary to introduce space and time. The whole system can be reduced up to material point with concentrated mass in it. From point view the classical Newton's mechanics, to get answer on the question where and when is necessary to give 6 initial values (3D-position and -forces). During the movement, in all points of time should be given the 3D-acting forces. There is more information about system for mechatronic devices a rule, because they are not unknown objects and are developed for some required purposes. That is a reason why its analysis of system behaviour can be simplified. For example, the trajectory of movement the mechanical parts is known (defined) but the current position is in the interest for every point of time with acting force on it. In this case the movement can be described only single position variable like its position onto the known trajectory. It is clear that an origin for trajectory should be defined. The description of mechanical state in VHDL AMS has principal behaviour nature. It is connected with modern trend to simplify the complexities of whole process to

essentials for analysed situation. The behaviour description of dynamics deals with not with complex geometry of system but rather with their properties. The most common properties of mechanical system, which are important for mechanics, are inertia, elasticity and friction. These properties are the main characteristics for every mechanical system for dynamical analysis at the system level. The transmission and transformation and distribution of energy are important derivations of it. The inertia is the only possibility to bring the time delay in some dynamical process of the mechanics. The elasticity presents the capability to accumulate mechanical energy in component and to exchange it between them. The friction describes the losses of energy and yet essential more and plays in dynamical process some special role. The friction helps also to save mechanical state thanks interconnection between components without energy losses. Historically, VHDL was developed for analysis of electrical systems. It was developed originally for simulation only pure digital systems. The extension AMS (Analog-Mixed-Signal) gave possibility to extend the system description on analogue systems where the mechanics belongs. The similarity of both domains is united under the method of its description: behavioural. The difference between the domains begins from necessary "connecting wires". Electronics needs the only electrical "wire" as connecting element. In the mechanics at least, two necessary connection kinds are demanded: translation position and rotational angle. These set may be extent at velocity translational and so on. The analogy between domains is again at new level demanded [1]. It should be not forget the temperature dependences of these properties. Some properties have strong dependability from temperature like stiffness. Other properties are undependable at all like inertia. An electrical network is obeyed the current and voltage Kirchhoff's laws. For mechanics the analogy of current law is the D' Alembert principle: "The sum of all forces inclusive inertial force, acting of the body is equal zero" [2]. It is equal the law of Newton for dynamics, but formal with introduction of fictitious force of inertia has other mathematical formulation. For second Kirchhoff's law for voltage can be taken analogy in mechanics like Leibniz's principle of continuity the velocities [3]. This law analogies were given only for electro-mechanical analogy force \leftrightarrow current and voltage \leftrightarrow position. For analogy force \leftrightarrow voltage, current \leftrightarrow position can be

found other physically suitable analogies of laws or principles.

2.1 Inertia

Mass for translational and moment of inertia for rotational movement belong to property of inertia. In reference to dynamics, the inertia allows to bring the time delay between action and reaction of the component. An implementation the equation for inertia using VHDL AMS, mass model can be presented as follows:

```
entity mass_t is
  generic (m : real :1.0);
  port (terminal pos : translational); -!/?
end entity mass_t;
architecture basic of mass_t is
  quantity x across F
    through pos to translational_ref;
  quantity v : velocity:=0.0;
begin
  v==x'dot;
  F==± m· v'dot; -!/?
end architecture basic;
```

Despite of simplicity of mathematical presentation of mass model, already

$$F(t) = \pm m \cdot \frac{dx^2}{dt^2} \quad (1)$$

two questions appear in it (marked with !?). The first one is the number of connections of model. This question is connected with time delay concept. Which delay is meant: relative or absolute (what is meant under x)? The all analogue elements have both pins. It is very interesting (or may be important) to understand the difference between single-pin mass and two-pin inverter model [4]. Some explanation of sign choice in equation before mass constant model of the mass the simulator SABER[®] (Synopsys) sounds like: "Note the (-) sign in the force equation. This is consistent with the convention that a positive force causes the position (pos) to increase. In this case, the reaction force tends to resist a positive translational acceleration" [5]. At the end, it doesn't matter how to set it, but if the sign is set in mass model, it should be consistent followed the proper force direction in all mechanical models.

In following, our sign before mass will be set in plus (+) like in VDA model libraries [6]. The VHDL AMS simulator can simulate the both directed models, but there is not known method to define automatically the force direction mixture in simulation.

```
entity inverter_t is
  generic (m : real :1.0);
  port(terminal pos_1 : translational;
    terminal pos_2 : translational);
end entity inverter_t;

architecture basic of inverter_t is
  quantity x across F
    through pos_1 to pos_2;
  quantity v : velocity:=0.0;
begin
  v==x'dot;
  F==+m· v'dot;
end architecture basic;
```

If the second pin pos_2 is connected to the translation reference, the inverter model is transformed into classic mass model. The second question deals with choice what is the sign before the mass constant m in model equation. Unfortunately, standard for extension AMS of the language VHDL gives freedom of this choice in language. The tool manufacture supplies the simulator usually with the generic libraries. In the supplied libraries is already sign chosen according to provided components in it. The sign choice of the inertia force determines the sign in all mechanical models dealing with forces. The choice of sign for inertia force is not prescribed also in standard of language Modelica what leads to similar problems. Practically, that means: without knowing the force orientation in mechanical model is not recommended to use this component in your own device model. The mixed usage is not tested from tool, is permitted from language construction and syntax. A mixture usage of such models leads to false simulation results, which sometimes in complex models are not at once visible. For example, in model the mass of SABER[®] (Synopsys) the sign in equation is minus, in open source library VDA is plus. As consequence of it is non-compatibility of mechanical part the VDA libraries with SABER[®] mechanical one. It concerns all components where the forces are presented. The energy transformer like motors are also involve in it.

2.2 Elasticity

The term solid (rigid) body was introduced as some absolute for simplification of analysis in classic mechanics. If the stiffness of connected parts is relatively great in comparison with other connections/components, it can be neglected in form non-deformable connections/components. In reality there isn't any absolute rigid body. The main question is how much elastic energy should be taken into account. If not whole of elastic energy will be saved, it is spoken about plastic effects. The simplest model of elasticity is elastic spring. The equation for some lineal spring looks like as

$$F = +K \cdot X \quad (2)$$

where K - stiffness of component (connection), X - difference between both pins of the model. Physically X means degree of deformation the spring. By such definition of model, it is suitable for symmetrical usage in common model. The sign cannot select as arbitrary. The force sign (before stiffness constant) depends from sign in model inertia. This simple elastic spring describes the situation of linear dependence between relative position difference and applied force. The constant K means that in whole working range only linear dependence was assumed. The elastic spring presents more property as mechanical construction. It should not mix with mechanical construction spring. The first one is only properties, second is some material construction element. Very important to understand that elastic spring places the role of energy saver or force transmitter. The force is transmitted by means of the spring without any delay (duration of force transmission is equal zero!). The spring can be used as component model for energy accumulation or connection element with elastic interconnection. According to definition the spring is not defined in which mechanical state (stress or strain) work this imaginary spring. It is only information that the reaction force is proportional relative displacement of the both ends.

The next interesting aspect in reference to elasticity is initial state of system with preloaded spring. The typical introduction of initial length of spring look like as following expression

$$F = +K \cdot (X - L_0), \quad (3)$$

where L_0 is initial length of unstressed elastic spring. As consequence of it is introduction some preloaded

force in all mechanical chain from one hand. From another hand, it is the end of symmetry of such components by their usage if the $L_0 \neq 0$. The state of spring stress or strain is dependable from its orientation. State analysis belongs more to construction as to properties. An appearance of element the length L_0 means some transformation from normalized kinematic structure (sum of initial length differences are equal to zero) to dimensional one. This physically is equal to some additional initial force embedded in system. It should be not forgotten that all are only idealized presentation of real movement which describes the dynamics of mechatronic system good enough for analysis system level.

2.3 Friction

There is usual misunderstanding to treat the friction as only losses. But the friction is essential more as mechanical losses. This is more some object for transformation and stabilization. It can transform the mechanical energy in heat. It is transformer for translation movement in rotational one. A transition from dynamics to statics is impossible without friction. There are many physical phenomena dealing with friction. For example, the damping is nothing else as velocity dependable friction.

At system level it is concentrated not the complex physical phenomena of friction but rather than its consequences for dynamics. The basic for such description of friction is the Coulomb's law. The peculiarity of this law is the formulation not like some equality in equation but estimation of boundary condition

$$F_{fric} \leq \mu_{fric} \cdot N \quad (4)$$

where F_{fric} - maximal friction force before the movement is starting, μ_{fric} - friction coefficient, dependent on the contacting surfaces, N -normal force between both bodies.

The greatest problem by usage of friction model is nonlinearity. The friction force is always directed against the velocity or forces, trying to start a movement. And very often there is space anisotropy, i.e. the friction forward is not equal backward one. This transmission from movement to staying is very difficult point of time for every simulator of dynamics because at this moment there is some jump from some

small velocity V_{min}^{\pm} to zero. Such discontinuity is some stumbling block for overcoming it is used simple trick. It is assumed linear behaviour force-velocity in range $[-V_{min}^-, +V_{min}^+]$, where V_{min}^{\pm} are small enough, see Figure 1. This is typical model which is used by

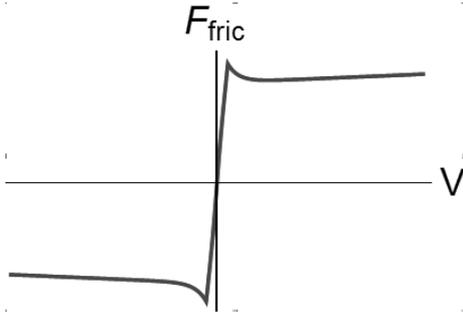


Figure 1: Friction Coulomb-Stribeck (defined in zero)

numerical calculation. The main disadvantage is by that behaviour by velocity is equal zero. Using such modelling the movement begins under acting arbitrary force what doesn't reflect the reality. The paradox of such modelling consists in the ignorance of the important part of dynamics. The dynamics means the changing of state. During this transformation of movements is happen the most interesting from dynamical point of view events. Some stationary state is very similar to static case. The more real model should look like as following at the Figure 2.

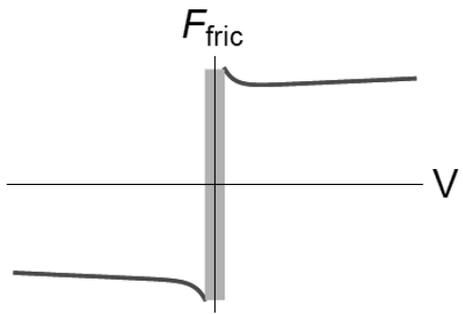


Figure 2: Friction Coulomb-Stribeck (undefined in the vicinity of zero)

The basic model of friction can be extended in different directions like temperature dependability, nonlinear or anisotropic behaviour and so on.

The most important is proper to describe the slack zone, mathematically - the inequality of Coulomb's

law. Some example of realization that concept in VHDL AMS is below

```
entity dry_fric is
    generic(
        fric_coul: real :=0.1;
        fric_brkw: real :=0.2;
        c_w: real :=1.0;
        vel_threshold: real :=1.0E-4);
    port(
        terminal pos_1 : translational;
        terminal pos_2 : translational);
end entity dry_fric;
architecture basic of dry_fric is
    quantity pos across fric_frc
        through pos_1 to pos_2 ;
    quantity vel: velocity ;
begin
    vel_eq: vel == pos'dot;
    if abs(vel)> vel_threshold use
        fric_frc ==sign(vel)*(fric_coul+
            (fric_brkw-fric_coul)*exp(-c_w*abs(vel)));
    else
        if abs(vel)/= 0.0 use
            if abs(fric_frc)<fric_brkw use
                vel==0.0;
            else
                fric_frc == sign(vel)*fric_brkw;
            end use;
        else
            fric_frc ==0.0;
        end use;
    end use;
end architecture basic;
```

Into the range $[-v_threshold, +v_threshold]$ (grey region on Figure 2) velocity remains equal to zero and the magnitude of friction force can be "arbitrary" from colour range, i.e. according to Coulomb's law. The "arbitrary" force means the statically equilibrium state should be valid in this velocity range. When the magnitude friction reaches the value $\mu_{fric} \cdot N$, after that the movement begins. For proper practical application this model should be extended to appropriate initialization (for DC-analysis). The anisotropy can be taken into account with different parameters for forward and backwards movements. Also the "staying zone" should be corrected with introduction unsymmetrical range $[-v_threshold^-, +v_threshold^+]$ if it is necessary.

2.4 Forces and mechanisms

The forces are main terms which are used in dynamics analysis. The acting effort of arbitrary mechanism can be replaced through equivalent force. That is the most important feature of behavioural modelling. The best example of usage such analysis is electronics. They can control the billion transistors despite of such complex structure single one. There are a number of analogies between mechanical and electrical components [1]. The most known are mass like capacitor or mass like inductance. In first case the electrical energy is analogue of mechanical one, in second magnetic field is similar mechanical. Corresponding the force is analogue either current or voltage. Going out these analogies it can be already seen which elements are most important for mechanics. They are as different sources like force or position, an amplifier and so on. Like analogue of such component in mechanics can be taken a mechanical lever. Additional complexity arises on mechanics modelling because of existence two kinds of movements as translational and rotational. For transferring from translation movement to rotational one, rack and pinion is mechanism suitable for that.

Every domain despite of similarity and analogies has own specialties. For mechanics this is e.g. definition which force is internal, which - external. So definition is direct connected with sign choice in mass model. On system analysis the new thinking is asked because the other questions are to answer as by component analysis. The euphoria with using FEM-methods onto system level is damped at present. The reduction of FEM-models to simplified are intensively developed at nowadays. Next special feature of mechanics is usage of big range around working point. The real springs works in stressed block state as good as linear elastic range also. Block state is nothing else like only essential greater stiffness.

The standardized set of mechanisms should be yet more clearly defined in the future. The question is which set is necessary to cover of most mechanical application in the mechatronic device. It should be found reasonable compromise between practical usage and theoretical requirement.

3 Oscillation of spring-mass system under dry friction force

A study was performed on some simple example of oscillation the translation spring-mass under friction. All models used by that are described above. To get the oscillation in system the force source was used. The purpose of study is test the friction model. Similar investigation with simplified friction model was supposed in [7]. The classical friction model, which is used very often in education is viscous friction, i.e. friction force is assumed proportional velocity. The advantage of this model consists in existing analytical solution for it. The disadvantage is the infinite time up to full stop according to the solution.

Already the verbal description of system behaviour brings some parameters in model of dry friction: the effort by starting of movement a little bit greater as by self-movement and at some finite time the mass will be stop. The introducing parameters in above model describe this behaviour. In the example, the parameters were taken the same like in the [7] to compare the simulation results. The analytical solution does not exist for spring-mass free oscillation with Stribeck friction. The some approximation of number of stops was derived in [7] for simplified model. The simplification concerns of transition from breakaway friction onto Coulomb's friction. The spring-mass system with Stribeck friction is shown an Figure 3 The force

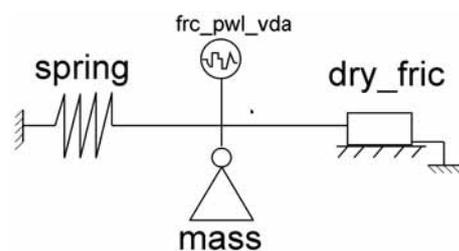


Figure 3: Spring-mass system with friction

source (frc_pwl_vda) gives initial impact to set up at the beginning the initial amplitude. The number of stop point by oscillation up to full stop n_{stops} can be calculate for simplified model [7] as following

$$n_{stops} = \text{int}\left(\frac{A_0 \cdot K}{2 \cdot \mu_k mg} - \frac{\mu_s - \mu_k}{2 \cdot \mu_k}\right), \quad (5)$$

where $\text{int}()$ - integer part of expression, A_0 - initial

amplitude, μ_k - kinetic friction coefficient, μ_s - static friction coefficient, g - gravity acceleration, m - mass, K - stiffness of spring. For our case the formula looks like that

$$n_{stops} = \text{int}\left(\frac{A_0 \cdot K}{2 \cdot fric_coul} - \frac{fric_brkw - fric_coul}{2 \cdot fric_coul}\right) \quad (6)$$

Formula (6) is the same like (5) only for friction parameters used above. Table 1 shows the results of calculation for estimation n_{stops} . The graphical result of

n_{stop}	$fric_brkw$ (μ_k)	$fric_coul$ (μ_s)
11	7.84 (0.4)	1.96 (0.10)
4	7.84 (0.4)	6.86 (0.35)

Table 1: Number of stops according to formula

simulation with SABER[®] simulator for both cases is presented on Figure 4. Friction forces are shown in above region of graphs and the mass positions is below of it. The system was simulated with two kinetic

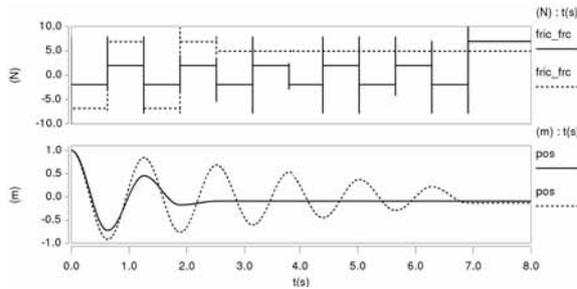


Figure 4: Friction forces (above) and mass positions (below) for $m = 2kg$, $\mu_k = (0.1/0.35)$, $\mu_s = 0.4$, $K = 50 \frac{N}{m}$, $A_0 = 1m$

frictions $\mu_k = 0.1$ (solid) and $\mu_k = 0.35$ (dashed) what corresponds $fric_coul = 1.96$ and $fric_coul = 6.86$. The influence of c_w , which characterizes of transformation static to kinetic friction, is have not seen if $c_w > 50$. The value of magnitude $vel_threshold$, if $vel_threshold < 0.1$, is not important for number of stops. The transient curves and amount of stops are the same as in [7].

The friction model can be extended for viscous part. The simulation results should be compared, if it is possible, with reality. The big difficulties have place by separation the viscous part from dry one.

4 Conclusion and recommendations

The modelling language VHDL AMS is very suitable for dynamic description the mechatronic system. The creating of mechanical part of mechatronic system requires the new methodical approaches and more essential physical knowledge's about physic behaviour the system. Only essentials for dynamics should be taken into account. The essentials are strong dependable from questions to simulation. What is an important for one simulation, can be neglect in another case for the same system. However, the dynamical behaviour should be reflected real one undependable on questions to simulation,.

The careful analysis of energy distribution in the system helps to make proper syntheses of it. The most important mass and stiffness structuring of system can be made properly and easy after that. It should to be cautious in using foreign mechanical models. The proof of the chosen force direction is necessary by that. The simulator does not test the whole model on consistent usage of force direction in constitute parts.

The behaviour modelling is process of gaining of the understanding the system. The abstraction from a constructive realization and concentrating on essentials is building the know-how on dynamics the system. The simulation generic models reflect as rule not constructive specialties of system bat rather the dynamical properties of it. It is great difference the system level from component level where the constructive realization is in focus of study.

What is most important for dynamics depends on questions asked to simulation. The proper formulated question on dynamical behaviour is already a half of solution. The complex dynamical process [8] can be described with relative simple set of behaviour models.

It doesn't matter which tool (simulator) is used by the proper description of dynamical behaviour of mechatronic system. It should no forgotten that high dynamics in mechanics has place by changing of movement state but rather than in some stationary one.

References

- [1] P. Le Corbeiller, Ying-Wa Yeung *Duality in Mechanics*. The journal of the acoustical society of Am., Vol.25, No.7,pp.643-648, 1952
- [2] Encyclopedia Britannica. *D'Alembert's principle*.
<http://www.britannica.com/science/dAlemberts-principle>
- [3] Larry M. Jorgensen *The principle of continuity and Leibniz's theory of consciousness*. Journal of the History of Philosophy 47 (2):pp. 223-248 (2009)
- [4] Malcolm C. Smith *Synthesis of mechanical networks: the inerter*. Automatic Control, IEEE Transactions on, Vol. 47, No. 10, pp.1648-1662, Oct., 2002
- [5] SABER®(Synopsys) *Mass model (in MAST)*.
<https://www.synopsys.com/Company/Pages/default.aspx>
- [6] VDA FAT AK30 *VHDL AMS Model libraries*.
http://fat-ak30.eas.iis.fhg.de/mb/index_en.html
- [7] Avi Marchewka, David S. Abbott, Robert J. Beichner *Oscillator damped by a constant-magnitude friction force*. Am. J. Phys., Vol. 72, No. 4, pp.477-483, 2004
- [8] Sergey Petkun, Xin Zhao *BroSAnT – an Example of the Mechatronical Way of Thinking*. SNE 22(3-4), pp.181-186, 2012

Eine hierarchische Regelstruktur für das S-Mobile mit sphärischem Elektroantrieb

Marian Göllner^{1,a}, Haoqi Tao^{1,b}, Xiaobo Liu-Henke^{1,c}

¹Ostfalia Hochschule für angewandte Wissenschaften, Fakultät Maschinenbau,
Institut für Mechatronik, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel

^amar.goellner@ostfalia.de, ^bha.tao@ostfalia.de, ^cx.liu-henke@ostfalia.de

In diesem Beitrag wird die mechatronische Komposition eines hochdynamischen Systems mit sphärischem Elektroantrieb namens S(phere)-Mobile vorgestellt, welches Gegenstand der aktuellen Forschung ist. Zur Repräsentierung des Systemverhaltens werden die expliziten Kinematik- und Dynamikfunktionen des gesamten mechatronischen Systems symbolisch hergeleitet und in einem detaillierten MKS-Modell eingebettet. Darauf basierend wird die nichtlineare Mehrgrößenregelung, hierarchisch aufgebaut mit globalen, zentralen und lokalen, dezentralen Controllergruppen, ausgelegt und anhand ausgewählter Anwendungsszenarien validiert. Mittels Model-in-the-Loop-Simulation (MiL) werden unter Einsatz verschiedener Anwendungsszenarien sowohl die hohe Regelgüte, als auch das erreichbare sehr gute Systemverhalten bewiesen.

1 Einleitung

Ziel eines Forschungsvorhabens der Fachgruppe Regelungstechnik der Ostfalia HaW ist die modellbasierte, rechnergestützte Entwicklung eines hochdynamischen Elektrotransportsystems S-Mobile, welches mit omnidirektionalen, sphärischen Elektroantrieben als Stelleinrichtung ausgerüstet ist. Dieses S-Mobile soll z.B. als Basis eines Serviceroboters für die Interaktion mit Menschen in typischen humanoiden Umgebungen oder als Stellrichtung einer intelligenten Mobilitätshilfe sowie als autonomes Fahrzeug zum Einsatz in Fabriken oder Logistikzentren dienen.

Abbildung 1 zeigt das neuartige Konzept des S-Mobile, bestehend aus der Antriebskugel und dem Aufbau.

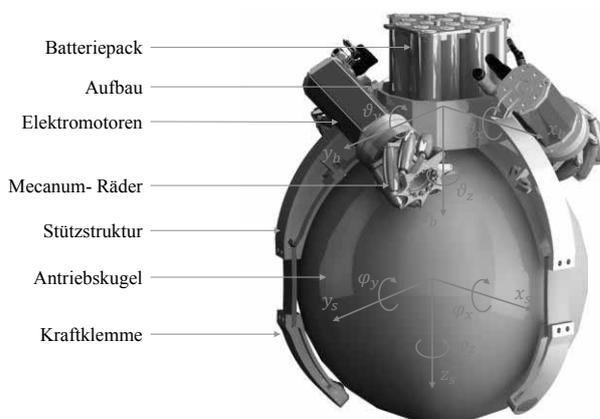


Abbildung 1. Neues Konzept des S-Mobile

Der Aufbau ist untergliedert in die mechanische Tragstruktur sowie den Aktuatoren, der Sensorik und der Informationsverarbeitung, welche mit der Antriebskugel das Kernmodul bilden.

Die Tragstruktur verfügt über Kraftstützen die zum einen die Gewichtskräfte des Aufbaus auf die Antriebskugel übertragen und zum anderen über eine Blattfeder eine Klemmwirkung erzeugen. Die Klemmkraft greift genau gegenüber der Auflagepunkte der Antriebsräder an und ermöglichen so, dass die Normalkräfte auf die Radreibflächen konstant gehalten werden können. Dementsprechend lässt sich mit der Vorspannung der Blattfeder eine gezielte Beeinflussung der Normalkräfte zwischen Räder und Antriebskugel und damit eine Einstellung der Reibkraft bei konstantem Reibbeiwert erzielen.

Zur Messwerterfassung wurden ein linearer Beschleunigungssensor sowie Gyroskope zur Winkelgeschwindigkeitserfassung und ein Magnetometer zur Kompensation der Magnetfeldorientierung integriert. Die Bewegung der Antriebskugel lässt sich über Odometrie erfassen. Das heißt, dass über eine Drehzahlerfassung der Antriebsmotoren, sowie eine Koordinatentransformation dieser in das Ursprungskordinatensystem, auf die Bewegung der Antriebskugel geschlossen wird. Die relative Position und die absoluten Winkel des S-Mobile im Raum können durch Sensorsignalverarbeitung aus den Daten der Sensoren ermittelt werden. Die absolute Position soll durch die spätere Integration eines GPS Moduls ergänzt werden.

Mittels einer hierarchischen Regelstruktur, bestehend aus einem globalen nichtlinearen Gain-Scheduling Regler und lokalen zeitdiskreten Dead-Beat-Reglern, wird eine kontrollierte Bewegung des S-Mobile in fünf vorgesehenen Freiheitsgraden, der Längsbewegung in x- und y-Ebene, der Drehbewegung um die Hochachse (gieren), wanken und nicken, ermöglicht.

Als Aktuatoren werden Elektromotoren eingesetzt, da die geforderte hohe Systemdynamik mit anderen Übertragungsmedien nicht gewährleistet ist. Hinsichtlich der Anforderungen an die Schnelligkeit und Bandbreite der Aktoren, aus Sicht des zu entwerfenden Regelkreises, kommen hier nur elektrische Motoren in Frage. Zudem müssen kleine Winkel mit konstanten Drehmomenten ausgeregelt werden, weshalb hier Synchronstrommotoren verwendet werden sollen. Da für die Realisierung dieses Systems Aktuatoren mit hohen Drehmomenten und relativ geringen Drehzahlen benötigt werden, wird zwischen den Antriebsmotoren und den Mecanum-Rädern ein spielreies Getriebe vorgesehen.

Die Anordnung der Aktoren spielt für die Realisierung eine große Rolle. Abbildung 1 zeigt die gewählte Konfiguration mit einem sphärisch angeordneten Antrieb mit drei um $\gamma_1 - \gamma_2 = 120$ Sphärengrad versetzten Antriebsrädern. Die Räder sind zu dem Aufbau um $\varepsilon = 45$ Grad geneigt und zusätzlich um $\eta = 45$ Grad um die eigene Längsachse verdreht (siehe Abbildung 2). Durch diese geeignete Anordnung können jegliche weitere Stützstrukturen eingespart, ein auf-

rechtes Drehen des Aufbaus um seine Hochachse ermöglicht und eine energieeffiziente Nutzung der notwendigen Mecanum-Räder gewährleistet werden. Diese Räder ermöglichen durch ihren zusätzlichen Freiheitsgrad erst das abrollen auf der Antriebskugel.

2 Modellbildung

Die Entwicklung erfordert eine systematische und interdisziplinäre Vorgehensweise, die eine optimale Auslegung des Gesamtsystems erlaubt. Diese mechatronische Komposition erfolgt in einem iterativen Prozess, bestehend aus Modellbildung, Analyse und der modellbasierten Auslegung. Durch die modellbasierte Auslegung können z.B. benötigte Stellkräfte gefunden und zum Auslegen von Aktoren genutzt werden [3], [4].

Um das Verhalten eines solchen dynamischen Systems zu untersuchen und anschließend eine hierarchische Regelstruktur zu entwerfen, müssen sowohl die Kinematik-, als auch die Dynamikfunktionen berücksichtigt werden.

Abbildung 2 zeigt ein vereinfachtes physikalisches Ersatzmodell, welches die Kinematik und Dynamik des Systems repräsentiert.

In der vorherigen Forschungsarbeiten [1] wurde bereits ein dreidimensionales Modell hergeleitet, welches zunächst zur Auslegung der Entkopplungs- und Verkopplungsstruktur sowie zur Validierung des ersten Entwurfs des S-Mobile diente. Die dabei gefundenen Gleichungen können nun in der vorliegenden

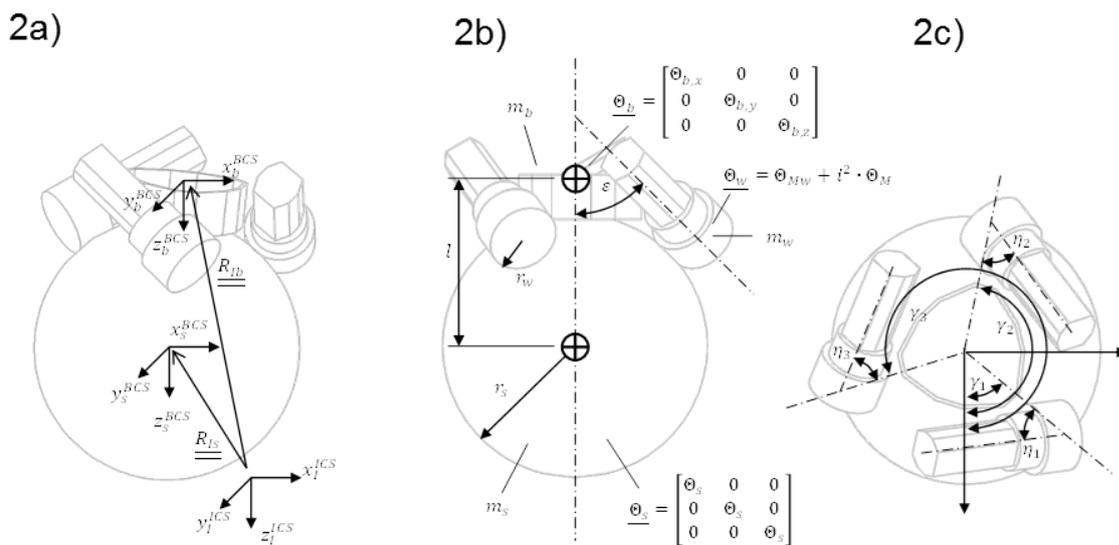


Abbildung 2. Physikalisches Streckenmodell

Arbeit so erweitert werden, das ein Modell der neuartigen Antriebskonfiguration mit den räumlich versetzten Motoren und dem Verhalten der Mecanum Räder erstellt werden kann. Zudem basiert der optimierte Reglerentwurf auf der Linearisierung dieses nichtlineareren Modells um verschiedene Arbeitspunkte um so den gewünschten Gain-Scheduling Regler zu finden.

Zudem soll mit dem nichtlinearen Modell untersucht werden, inwiefern die getroffenen Vereinfachungen in den physikalischen Extrembereichen des Systems noch Gültigkeit haben und ob entsprechenden Abweichungen toleriert werden können. Durch dieses Vorgehen kann validiert werden, inwieweit die hier hergeleitete Reglerstruktur für die geforderten Systemcharakteristika ausreichend ist.

Im Folgenden werden Kinematik und Dynamik im Detail dargestellt.

2.1 Kinematikfunktionen

Die Kinematikfunktionen dienen als Grundlage der dynamischen Funktionen, um Lage, Orientierung, Winkel und Winkelgeschwindigkeiten zu ermitteln.

Die kinematischen Zusammenhänge werden aus den Freiheitsgraden und der Geometrie des S-Mobile bestimmt. Die einzelnen Teilkomponenten werden mit geeigneten Koordinatensystemen nach Abbildung 2a versehen. Das Inertialkoordinatensystem (ICS) mit den Komponenten (x_I, y_I, z_I) liegt auf der Fahrbahn. Die jeweiligen Körperkoordinatensysteme (BCS) liegen in jedem Starrkörper. Die Kopplungsbedingungen werden hier nicht wie üblich über Verbindungskoordinatensysteme (ACS) dargestellt, sondern über vereinfachte kinematische Beziehungen. Durch die konstruktive Gestaltung der Mecanum-Räder [5] sind die Kopplungsbedingungen unstetig, es wurden geometrische Beziehungen angewandt. Die Kinematik des Gesamtsystems wird durch Koordinatentransformation mittels Transformationsmatrizen symbolisch hergeleitet:

$$\underline{R}_z = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$\underline{R}_y = \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \quad (2)$$

$$\underline{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix} \quad (3)$$

Die Transformation in die einzelnen Koordinatensysteme erfolgt durch die Multiplikation dieser grundlegenden Matrizen. Die Transformationsmatrizen dienen hauptsächlich zur Transformation der Bewegungen der einzelnen Körperkoordinaten in das ortsfeste Koordinatensystem:

$$\underline{R}_{Ib} = \underline{R}_x \cdot \underline{R}_y \cdot \underline{R}_z \quad \text{Transformation von ICS zu b} \quad (4)$$

$$\underline{R}_{bI} = \underline{R}_{Ib}^T = \underline{R}_x^T \cdot \underline{R}_y^T \cdot \underline{R}_z^T \quad \text{Transformation von b zu ICS} \quad (5)$$

Weiterhin können die Winkelgeschwindigkeiten der Antriebskugel, der Räder und des Aufbaus durch diese Koordinatentransformationen berechnet werden. Die Winkelgeschwindigkeit der Antriebskugel soll durch die Größen $\dot{\phi}_x$ und $\dot{\phi}_y$ beschrieben werden. Eine Drehung der Kugel um die eigene Hochachse z wird hier aufgrund des dafür erforderlichen Bohrmomentes nicht berücksichtigt. Somit ergibt sich für die absolute Rotationsgeschwindigkeit der Kugel:

$${}^I\Omega_{Is} = \underline{R}_{Is} \cdot \begin{bmatrix} \dot{\phi}_x \\ \dot{\phi}_y \\ 0 \end{bmatrix} = \left(\underline{R}_{Is} \cdot \begin{bmatrix} \dot{\phi}_x \\ \dot{\phi}_y \\ 0 \end{bmatrix} \right) \times \begin{bmatrix} 0 \\ 0 \\ R_s \end{bmatrix} \quad (6)$$

Die relative Winkelgeschwindigkeit der Antriebskugel bezogen auf den Aufbau im BCS^b ist identisch mit der Differenzgeschwindigkeit der absoluten Winkelgeschwindigkeiten der Antriebskugel und des Aufbaus jeweils im BCS^b. Somit folgt:

$${}^b\Omega_{bs} = \underline{R}_{bI} \cdot {}^I\Omega_{Is} - {}^b\Omega_{Ib} \quad (7)$$

Die zur Umrechnung nötige absoluten Winkelgeschwindigkeiten des Aufbaus ${}^b\Omega_{Ib}$ ergeben sich durch die Multiplikation der Tait-Bryan-Winkel $\underline{\hat{\theta}}$ mit dem totalen Differential der entsprechenden Transformationsmatrix, dargestellt als Jakobi-Matrix \underline{J}_{bI} :

$${}^b\Omega_{Ib} = \underline{J}_{bI} \cdot \underline{\hat{\theta}} = \begin{bmatrix} \dot{\theta}_x - \sin(\theta_y) \cdot \dot{\theta}_z \\ \cos(\theta_x) \cdot \dot{\theta}_y + \cos(\theta_y) \cdot \sin(\theta_x) \cdot \dot{\theta}_z \\ -\sin(\theta_x) \cdot \dot{\theta}_y + \cos(\theta_x) \cdot \cos(\theta_y) \cdot \dot{\theta}_z \end{bmatrix} \quad (8)$$

Die relativen Geschwindigkeiten der Antriebsräder sind im Koordinatensystem BCS^b mit einer Übersetzung von \ddot{u}_c an die Winkelgeschwindigkeit $\dot{\psi}_i$ des Motors gekoppelt. Somit ergibt sich für alle Räder i mit $i = 1, 2, 3$:

$${}^i\Omega_{Iw} = \frac{1}{r_c} \cdot \dot{\psi}_i \quad (9)$$

Die absoluten Geschwindigkeiten der Räder können durch Bindungsgleichung in Abhängigkeit der jeweiligen Positionen der einzelnen Koordinatensysteme gefunden werden. Nach Abbildung 2b ergibt sich die Bedienung, dass kein Schlupf zwischen Kugel und

Rad vorhanden ist, sodass folgende allgemeine Gleichung für alle Räder i mit $i = 1,2,3$ gilt:

$$\left(\begin{array}{c} R_s \cdot \frac{\sin(\varepsilon)}{\sin(\eta_i)} \cdot \cos(\gamma_i) \\ R_s \cdot \frac{\sin(\varepsilon)}{\sin(\eta_i)} \cdot \sin(\gamma_i) \\ -R_s \cdot \frac{\cos(\varepsilon)}{\sin(\eta_i)} \end{array} \right) \cdot \begin{bmatrix} -\sin(\gamma_i) \\ \cos(\gamma_i) \\ 0 \end{bmatrix} = {}^i\Omega_{Iw} \cdot R_w \quad (10)$$

Auf Grundlage dieser Kopplungsgleichung kann nun durch auflösen nach ${}^i\Omega_{Iw}$ eine Beziehung zwischen der Winkelgeschwindigkeit der Antriebskugel $\dot{\varphi}_x, \dot{\varphi}_y$ und der des Aufbaus hergestellt werden. Die absolute Geschwindigkeit der Räder folgt dann durch einsetzen in die folgende Gleichung:

$${}^i\Omega_{bw} = {}^i\Omega_{Iw} + \begin{bmatrix} \frac{\sin(\varepsilon)}{\sin(\eta_i)} \cdot \cos(\gamma_i) \\ \frac{\sin(\varepsilon)}{\sin(\eta_i)} \cdot \sin(\gamma_i) \\ -\frac{\cos(\varepsilon)}{\sin(\eta_i)} \end{bmatrix}^T \cdot {}^b\Omega_{Ib} \quad (11)$$

Somit sind nun die Kinematikfunktionen sämtlicher Komponenten des Systems bekannt und die gefundenen Gleichungen können zur Berechnung der Dynamikfunktionen genutzt werden.

2.2 Dynamikfunktionen

Die Dynamikfunktionen beschreiben das dynamische Verhalten unter Wirkung von Kräften und Momenten. Unter Zuhilfenahme der physikalischen Gesetze kann eine mathematisch formulierte Aussage über das Systemverhalten gemacht werden [3]. Die Dynamik des S-Mobile setzt sich aus der Dynamik der Tragstruktur (Aufbau, Räder, Antriebskugel), der Aktoren und Sensoren zusammen. Mit Hilfe geeigneter Vereinfachungen wird ein räumliches Modell des S-Mobile erstellt.

Dynamik der mechanischen Tragstruktur

Hier sollen die Differentialgleichungen, die die dynamischen Vorgänge im System beschreiben, mit Hilfe des Lagrange Formalismus hergeleitet werden. Diese Methode verwendet die Erhaltung der Energie im System zur Herleitung der kraft- und momentenabhängigen Bewegungsgleichungen. Es muss folgender Gleichung gelöst werden:

$$\frac{d}{dt} \left(\frac{\partial T - V}{\partial \dot{q}} \right) - \left(\frac{\partial T - V}{\partial q} \right) = Q_{nk}^e + Q_{nk}^r \quad q \in \mathbb{R}^5 \quad (12)$$

Es werden Terme der kinetischen T und der potenziellen Energie V jeweils partiell nach den Minimalkoordinaten \vec{q} abgeleitet. Als Minimalkoordinaten wurden festgelegt:

$$\vec{q} = [\vartheta_x \quad \vartheta_y \quad \vartheta_z \quad \varphi_x \quad \varphi_y]^T \quad (13)$$

Die kinetische Energie der Antriebskugel ergibt sich aus dem folgenden Zusammenhang aus Abbildung 3b:

$$T_s = \frac{1}{2} \cdot m_s \cdot \dot{I}_s^T \cdot \dot{I}_s + \frac{1}{2} \cdot {}^b\Omega_{bs}^T \cdot \underline{\Theta}_s \cdot {}^b\Omega_{bs} \quad (14)$$

Für die kinetische Energiegleichung des Aufbaus wurde als Referenzpunkt die Schwerpunktlage der Antriebskugel gewählt. Durch diesen Umstand können zwar Transformationen entfallen, es muss aber ein Koppelterm T_{coup} in die Gleichung eingefügt werden:

$$T_b = \frac{1}{2} \cdot m_b \cdot \dot{I}_b^T \cdot \dot{I}_b + \frac{1}{2} \cdot {}^b\Omega_{Ib}^T \cdot \underline{\Theta}_b \cdot {}^b\Omega_{Ib} + T_{coup} \quad (15)$$

$$T_{coup} = m_b \cdot \left(\underline{R}_{bI} \cdot \dot{I}_b \right) \cdot \left({}^b\Omega_{Ib} \times \begin{bmatrix} 0 \\ 0 \\ l \end{bmatrix} \right) \quad (16)$$

Die kinetische Energie der Antriebsräder ergibt sich rein aus rotatorischen Termen, da sie fest mit dem Motor und somit translatorisch mit dem Aufbau verbunden sind:

$$T_{w,i} = \frac{1}{2} \cdot \underline{\Theta}_{w,i} \cdot {}^i\Omega_{bw}^2 \quad \text{mit } i = 1,2,3 \quad (17)$$

Die gesamte kinetische Energie wird durch Summation der einzelnen kinetischen Energien jeder Komponente gebildet:

$$T = T_s + T_b + T_{w,1} + T_{w,2} + T_{w,3} \quad (18)$$

Da die Antriebskugel nicht in der Hochachse verfahren kann und somit keine potentielle Energie aufbauen kann, ist die potentielle Energie im Schwerpunkt der Kugel gleich null. Da die Antriebsräder mit dem Aufbau verbunden sind, ist die potentielle Energie des Aufbaus auch gleich der gesamten potentiellen Energie des Systems. Die potentielle Energie des Aufbaus errechnet sich nach folgender Gleichung:

$$V = V_b = m_b \cdot \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \cdot \underline{R}_{bI} \cdot \begin{bmatrix} 0 \\ 0 \\ l \end{bmatrix} \quad (19)$$

Die nicht konservativen Kräfte bestehen aus den durch die Motoren erzeugten Drehmomenten:

$$\vec{\lambda} = [\tau_1 \quad \tau_2 \quad \tau_3]^T \quad (20)$$

Der Multiplikator $\vec{\lambda}$ lässt sich hier als Koordinate der verallgemeinerten Drehmomente verwenden. Um diese an die gewählten verallgemeinerten Koordinaten anzupassen, sollen die verallgemeinerten Drehmomente Q_{nk}^e durch die Multiplikation mit der Jacobi-Matrix J_{nk}^e umgerechnet werden. Die Jacobi-Matrix kann durch die Projektion (bzw. partielle Ableitung) der Winkelgeschwindigkeiten der Allseitenrädern ${}^b\Omega_M$ mit den verallgemeinerten Koordinaten \vec{q} ermittelt

werden [6]. Dabei müssen die Winkelgeschwindigkeiten der Allseitenräder relative zur Motorachse eingesetzt werden. Dies liegt daran, dass der Drehmoment-Multiplikator $\vec{\lambda}$ auch relativ zur Motorwellen-Achse definiert wird. Somit ergibt sich das verallgemeinerte Drehmoment \underline{Q}_{nk}^e , welches an den drei Allseitenrädern wirkt zu:

$$\underline{Q}_{nk}^e = \left(\underline{J}_{nk}^e \right)^T \cdot \vec{\lambda} \in \mathbb{R}^5 \quad (21)$$

Mit der Jacobi-Matrix:

$$\underline{J}_{nk}^e = \frac{\delta \underline{b}\Omega_W}{\delta \vec{q}} = \begin{bmatrix} \frac{\delta \underline{b}\Omega_{W1}}{\delta \theta_x} & \frac{\delta \underline{b}\Omega_{W1}}{\delta \theta_y} & \frac{\delta \underline{b}\Omega_{W1}}{\delta \theta_z} & \frac{\delta \underline{b}\Omega_{W1}}{\delta \varphi_x} & \frac{\delta \underline{b}\Omega_{W1}}{\delta \varphi_y} \\ \frac{\delta \underline{b}\Omega_{W2}}{\delta \theta_x} & \frac{\delta \underline{b}\Omega_{W2}}{\delta \theta_y} & \frac{\delta \underline{b}\Omega_{W2}}{\delta \theta_z} & \frac{\delta \underline{b}\Omega_{W2}}{\delta \varphi_x} & \frac{\delta \underline{b}\Omega_{W2}}{\delta \varphi_y} \\ \frac{\delta \underline{b}\Omega_{W3}}{\delta \theta_x} & \frac{\delta \underline{b}\Omega_{W3}}{\delta \theta_y} & \frac{\delta \underline{b}\Omega_{W3}}{\delta \theta_z} & \frac{\delta \underline{b}\Omega_{W3}}{\delta \varphi_x} & \frac{\delta \underline{b}\Omega_{W3}}{\delta \varphi_y} \end{bmatrix} \quad (22)$$

Analog dazu werden die auf den Aufbau eingepprägten Reaktionsdrehmomente \underline{Q}_{nk}^r nach Abbildung 2c berechnet mit den Gleichungen:

$$\underline{Q}_{nk}^r = \left(\underline{J}_{nk}^r \right)^T \cdot \left(-\underline{b}\underline{E}_W \cdot \vec{\lambda} \right) \in \mathbb{R}^5 \quad (23)$$

Die entsprechende Jacobi-Matrix zur Umrechnung der Motordrehmomente lässt sowohl nach der Erweiterung mit ${}^{b_l}J$ durch polynomiale Separation erhalten, als auch durch die partielle Ableitung der absoluten Winkelgeschwindigkeit des Aufbaus ${}_{b_l}\Omega_b$ nach \vec{q} .

$$\underline{J}_{nk}^r = \begin{bmatrix} 1 & 0 & -\sin(\theta_y) & 0 & 0 \\ 0 & \cos(\theta_x) & \cos(\theta_y) \cdot \sin(\theta_x) & 0 & 0 \\ 0 & -\sin(\theta_x) & \cos(\theta_y) \cdot \cos(\theta_x) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \frac{\delta \underline{b}\Omega_b}{\delta \vec{q}} \quad (24)$$

Der Multiplikator $\vec{\lambda}$ muss wegen Richtungsänderung der Drehmomente mit der Matrix ${}_{b_l}\underline{E}_W$ umgerechnet werden, welche aus den für die drei Motorwellen-Achsen entsprechenden Einheitsrichtungsvektoren, zusammengesetzt wird:

$$\underline{b}\underline{E}_W = R_s \cdot \begin{bmatrix} \cos \gamma_1 & \cos \gamma_2 & \cos \gamma_3 \\ \sin \gamma_1 & \sin \gamma_2 & \sin \gamma_3 \\ -1 & -1 & -1 \end{bmatrix} \quad (25)$$

Mit diesen Werten lässt sich der Lagrange Formalismus lösen und ein dreidimensionales, nicht-lineares Modell der mechanischen Tragstruktur entsteht.

Dynamik des Aktuators

Die Aktuatoren des Systems sind Synchronmotoren. Das maximale Drehmoment liegt bei dieser Motorenbauart an, wenn ein 90° vorausseilendes Drehfeld generiert wird (Abbildung 3). Zur Auslegung der Regelung wird das Dreiphasensystem mit den Strömen i_1 , i_2 und i_3 in das vom Polradwinkel

abhängige Zweiphasensystem mit den Strömen i_d und i_q überführt.

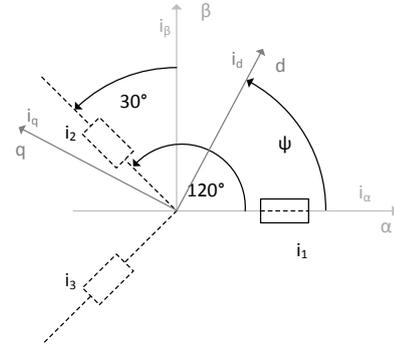


Abbildung 3. Park- und Clarke- Transformation

Zunächst wird die Clarke-Transformation durchgeführt, welche die Ströme in die ortsfesten Komponenten i_α und i_β transformiert.

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} \quad (26)$$

Anschließend werden die Komponenten längs und quer zur Rotorachse über den Verdrehwinkel des Rotors berechnet.

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \cdot \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (27)$$

Nach durchgeführter Clarke- und Park-Transformation lässt sich der Synchronmotor wie ein Gleichstrommotor behandeln. Die folgende Abbildung 4 zeigt das elektrische, einsträngige Ersatzmodell des Motors

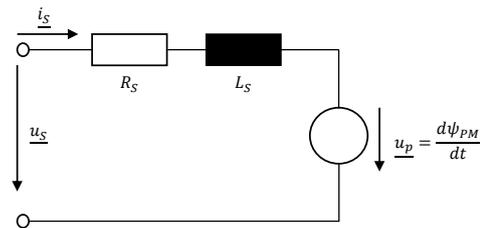


Abbildung 4. Elektrisches Ersatzmodell

Dieses ist im Wesentlichen ein elektrischer Gleichstromkreis bestehend aus der Motorinduktivität L_s und dem Strangwiderstand R_s . Die Ständerspannung u_s ist die Spannungsquelle aus der der Strangstrom i_s resultiert. Die Polradspannung u_p wirkt als Gegeninduktionsspannung.

Die Spannungen in d- und q-Richtung ergeben sich hiernach zu:

$$u_d = R_s \cdot i_d + L_d \cdot \frac{di}{dt} - \psi_y \cdot L_q \cdot i_q \quad (28)$$

$$u_q = R_s \cdot i_q + L_q \cdot \frac{di}{dt} + \psi_y \cdot (L_d \cdot i_d + \phi_0) \quad (29)$$

Zur Auslegung des Dead-Beat Reglers ist eine Darstellung der Systemgleichungen in der z-Ebene notwendig. Dazu werden zunächst die Übertragungsfunktionen für das Verhalten des drehmomentproportionalen Stromes i_d und des Quer-Stromes i_q zu ihren Stellspannungen U_d und U_q abgeleitet:

$$G_{sd}(s) = \frac{i_d(s)}{u_d(s)} = \frac{\frac{1}{R_s}}{1 + \frac{L_d}{R_s} s} \quad (30)$$

$$G_{sq}(s) = \frac{i_q(s)}{u_q(s)} = \frac{\frac{1}{R_s}}{1 + \frac{L_q}{R_s} s} \quad (31)$$

Nach der Transformation in die z-Ebene und Multiplikation mit der Übertragungsfunktion des zeitdiskreten Haltegliedes H_0 ergeben sich so zwei Übertragungsfunktionen für die Strecke:

$$H_0 G_{sd}(z) = \frac{\frac{1}{R_s} \left(1 - e^{-\frac{R_s}{L_d} T} \right)}{\left(z - e^{-\frac{R_s}{L_d} T} \right)} = G_{HSd}(z) \quad (32)$$

$$H_0 G_{sq}(z) = \frac{\frac{1}{R_s} \left(1 - e^{-\frac{R_s}{L_q} T} \right)}{\left(z - e^{-\frac{R_s}{L_q} T} \right)} = G_{HSq}(z) \quad (33)$$

Somit wird eine Entkopplung der beiden elektrischen Teilsysteme möglich.

Zur Bildung des Momentes der Synchronmaschine gilt nach [6] folgender Zusammenhang:

$$M_{1,2,3} = \frac{3}{2} \cdot p \cdot \dot{u}_G \cdot (\phi_0 \cdot i_q + (L_d - L_q) \cdot i_d \cdot i_q) \quad (34)$$

Abschließend werden die Spannungen durch inverse Park- und Clarke-Transformationen in die Spannungen U_1 , U_2 und U_3 zurückgeführt und über die Leistungselektronik gestellt.

Dynamik der Sensoren und des Reglers

Die Dynamik der Sensorik zur Erfassung der Winkelgeschwindigkeiten, der Wegbeschleunigungen, des Magnetfeldes und der Aktuatordrehzahl wird über ein PT1-Systemverhalten in der Simulation berücksichtigt.

Die digitale Realisierung eines Reglers führt zu einer Phasenabsenkung. Diese resultiert aus den auftreten-

den Totzeiten, die durch den Abtast-Haltvorgang und die zur Berechnung des Regelalgorithmus notwendige Bearbeitungszeit verursacht werden. Dies kann durch ein System mit endlich vielen Zustandsgleichungen approximiert werden. Zur Nachbildung dieser Effekte wird eine Padé-Approximation 2. Ordnung im Modell implementiert [4].

3 Modellbasierte Reglerauslegung

Abbildung 5 zeigt die hierarchische Reglerstruktur. Sie besteht aus einem zentralen, globalem Regelsystem und dezentralen, lokalen Reglern die im Folgenden im Detail dargestellt werden sollen.

Das globale Regelsystem arbeitet mit den ermittelten Sensorwerten, die erst in dem Block Sensordatenverarbeitung ausgewertet, umgerechnet und gefiltert werden müssen. Der globale Regler ist ein dreidimensionaler LQ-Regler mit Gain Scheduling. Er wird nachfolgend aus dem nicht-linearen dreidimensionalen Streckenmodell durch Linearisierung und finden des quadratischen Optimums gebildet. Durch die Linearisierung an unterschiedlichen Arbeitspunkten und das detektieren und umschalten zwischen diesen im Betrieb wird ein Gain Scheduling realisiert.

Die lokalen Regler werden als zeitdiskrete Dead-Beat Regler ausgeführt. Der Einsatz eines zeitdiskreten Reglers hat dabei den Vorteil, dass die Regelalgorithmen schnell auf dezentralen Microcontrollern lauffähig gemacht werden können. Zudem wird durch den Charakter des Dead-Beat Reglers stationäre Genauigkeit bei geringem Berechnungsaufwand und hoher Dynamik erreicht.

3.1 Globaler Regler

Der dreidimensionale LQ-Regler mit Gain Scheduling wird an mehreren Arbeitspunkten durch die Linearisierung des nicht-linearen Streckenmodells und dem Umschalten zwischen diesen Arbeitspunkten während des Betriebes realisiert. Es werden zu jedem linearisierten Modell ein LQ-Regler ausgelegt und die Ergebnisse der einzelnen Regelmatrizen in einem Look-Up Table abgespeichert, welches durch die detektierten Systemzustände auf den aktuellen Betriebspunkt umgeschaltet wird.

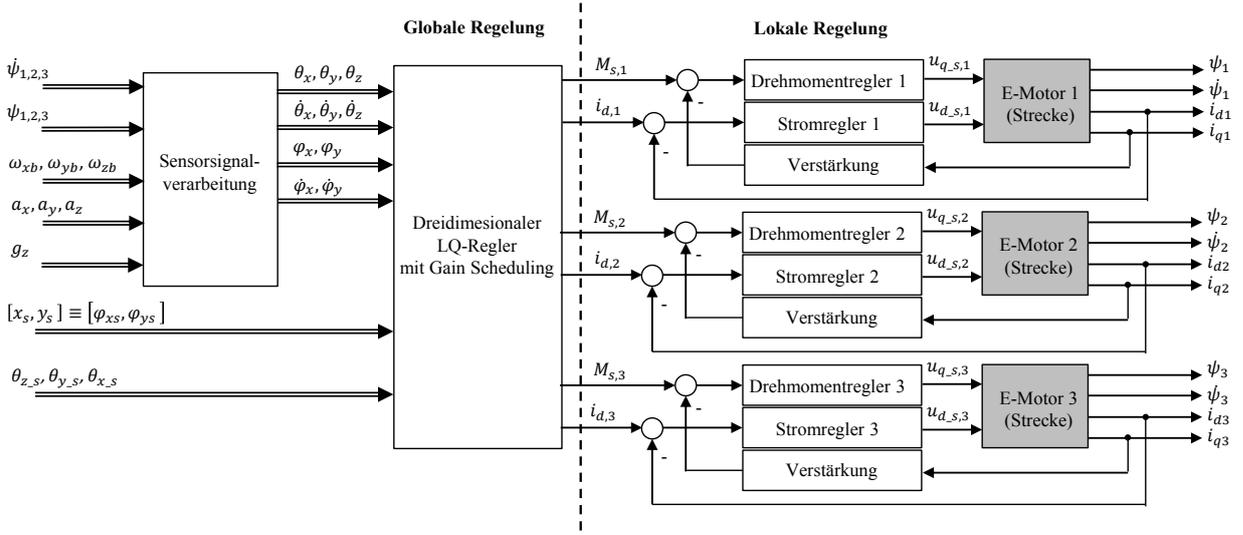


Abbildung 5. Zustandsregelung mit Aufschaltung z_i

Linaerisierung an beliebigem Arbeitspunkt

Um die parametrisierte Linearisierungsfamilie zu erhalten, soll die Linearisierung um beliebige Arbeitspunkte durchgeführt werden. Die lineare Bewegungsgleichung wird mittels partieller Ableitungen nach dem verallgemeinerten Koordinatenvektor und dem Eingangsvektor bestimmt:

$$\Delta \underline{y}_{lin} = f|_{APi} + \underline{J}_u(i) \cdot \Delta \underline{u} + \underline{J}_q(i) \cdot \Delta \underline{q} + \underline{J}_{\dot{q}}(i) \cdot \Delta \dot{\underline{q}} + \underline{J}_{\ddot{q}}(i) \cdot \Delta \ddot{\underline{q}} = 0 \in \mathbb{R}^5 \quad (35)$$

Dazu werden die Übergabeparameter zur symbolischen Herleitung der Jacobi-Matrizen wie folgt gewählt:

$$\underline{J}_u(i) = \left. \frac{\delta f}{\delta \underline{u}} \right|_{APi}; \underline{J}_q(i) = \left. \frac{\delta f}{\delta \underline{q}} \right|_{APi}; \underline{J}_{\dot{q}}(i) = \left. \frac{\delta f}{\delta \dot{\underline{q}}} \right|_{APi}; \underline{J}_{\ddot{q}}(i) = \left. \frac{\delta f}{\delta \ddot{\underline{q}}} \right|_{APi} \quad (36)$$

Analog entsteht die Bewegungsgleichung durch umformen der Gleichung (33).

$$\Delta \ddot{\underline{q}} = \underbrace{\left(-\underline{J}_{\ddot{q}}(i)^{-1} \cdot \underline{J}_q(i) \right) \cdot \Delta \underline{q} + \left(-\underline{J}_{\ddot{q}}(i)^{-1} \cdot \underline{J}_{\dot{q}}(i) \right) \cdot \Delta \dot{\underline{q}}}_{\text{für } \underline{A}_i \cdot \underline{x}} + \underbrace{\left(-\underline{J}_{\ddot{q}}(i)^{-1} \cdot \underline{J}_u(i) \right) \cdot \Delta \underline{u}}_{\text{für } \underline{B}_i \cdot \underline{u}} + \underbrace{\left(-\underline{J}_{\ddot{q}}(i)^{-1} \right) \cdot f|_{APi}}_{\text{für } \underline{E}_i \cdot \underline{z}_i} \quad (37)$$

Darauf basierend wird der i . beliebige Zustandsraum erstellt. Dabei sind der Zustandsvektor \underline{x} , Eingangsvektor \underline{u} , Ausgangsvektor \underline{y} , Ausgangsmatrix \underline{C} und die Durchgangsmatrix \underline{D} an jedem Arbeitspunkt gleich. Das nichtlineare System kann somit als Kombination mehrerer linearer Zustandsmodelle bzw. einer linearen Zustandsmodellfamilie verstanden werden.

$$\dot{\underline{x}} = \underline{A}_i \cdot \underline{x} + \underline{B}_i \cdot \underline{u} + \underline{E}_i \cdot \underline{z}_i$$

$$\underline{y} = \underline{C} \cdot \underline{x} + \underline{D} \cdot \underline{u} \quad (38)$$

Mit dem Eingangsvektor \underline{u} :

$$\underline{u} = \vec{\lambda} = [\tau_1 \quad \tau_2 \quad \tau_3]^T \quad (39)$$

Und dem Ausgangsvektor \underline{x} :

$$\underline{x} = [\vartheta_x \quad \dot{\vartheta}_x \quad \vartheta_y \quad \dot{\vartheta}_y \quad \vartheta_z \quad \dot{\vartheta}_z \quad \varphi_x \quad \dot{\varphi}_x \quad \varphi_y \quad \dot{\varphi}_y]^T \quad (40)$$

Die Systemmatrix \underline{A}_i und Eingangsmatrix \underline{B}_i können aus den verschiedenen Jacobi-Matrizen $\underline{J}_{\ddot{q}}$, $\underline{J}_{\dot{q}}$, \underline{J}_q und \underline{J}_u nach der linearisierten Bewegungsgleichung (37) berechnet und kombiniert werden. Der bei der Linearisierung erzeugte Grundterm kann als messbare Störgrößen im Zustandsraum verwendet werden, da der Grundterm anhand des Arbeitspunkts berechnet werden kann. Bei einem bestimmten i . Arbeitspunkt ist dieser Störvektor \underline{z}_i eine Konstante. Aus diesem Grund wird die Störmatrix $\underline{E}_i = \underline{I}$ eingesetzt.

LQ-Regler an beliebigem Arbeitspunkt

Die folgende Abbildung 6 zeigt die Regelstruktur des Nichtlinearen LQ-Reglers. Zu der Linearen Zustandsrückführung über K_i und der Vorsteuerung N_i wird hier eine Störgrößenaufschaltung vorgenommen, welche die nicht-linearitäten, die durch den Störvektor \underline{z}_i erfasst werden, kompensieren soll. Diese Aufschaltung, sowie die Matrizen N_i und K_i sind jeweils nur für eine Arbeitspunkt der nicht-linearen Strecke und somit nur für ein i . beliebiges Zustandsraummodell der Zustandsmodellfamilie gültig. Es wird dementsprechend für jedes Mitglied der Modellfamilie ein LQ-Regler ausgelegt.

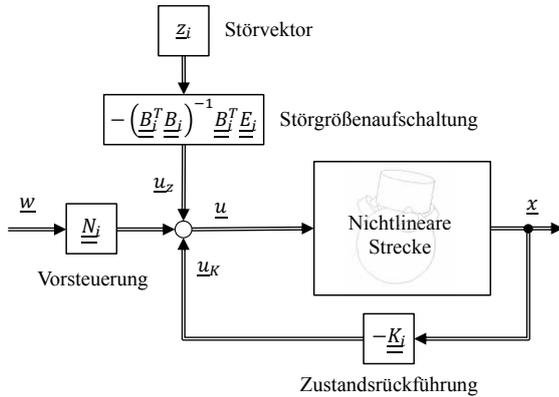


Abbildung 6. Zustandsregelung mit Aufschaltung z_i

Allgemein gilt es nun das Gütekriterium nach Ricatti zu optimieren. Zu diesem Zweck soll der Regelfehler \underline{e} , durch Betrachtung des Zustandsvektors \underline{x} (mit $\underline{y} = \underline{C} \cdot \underline{x}$ und $\underline{e} = \underline{w} - \underline{y}$ bei $\underline{w} = 0$), und der Stellaufwand durch den Eingangsvektor \underline{u} berücksichtigt werden. Es gilt die folgende Gleichung zu minimieren [7]:

$$J_{LQ} = \int_{t_0}^{t_e} \left(\underline{x}^T(t) \cdot \underline{Q} \cdot \underline{x}(t) + \underline{u}^T(t) \cdot \underline{R} \cdot \underline{u}(t) \right) dt \quad (41)$$

Die Faktoren Q und R dienen dabei einer Gewichtung der einzelnen Zustände des Zustandsvektors \underline{x} und des Eingangsvektors \underline{u} . \underline{Q} ist eine symmetrische, quadratische $l \times l$ Matrix, \underline{R} eine symmetrische, quadratische $m \times m$ Matrix. Grundsätzlich kann die Gewichtung und somit Auslegung der Faktoren \underline{Q} und \underline{R} nach Bryson's Regel [8] erfolgen:

$$Q_{jj} = \frac{1}{\text{maximal akzeptierbare Größe } x_j^2} \text{ mit } j \in \{1, 2, \dots, l\} \quad (42)$$

$$R_{kk} = \frac{1}{\text{maximal akzeptierbare Größe } u_k^2} \text{ mit } k \in \{1, 2, \dots, m\} \quad (43)$$

Was dem folgenden Kriterium entspricht:

$$J_{LQR} = \int_0^\infty \left(\sum_{j=1}^l Q_{jj} \cdot x_j(t)^2 + \sum_{k=1}^m R_{kk} \cdot u_k(t)^2 \right) dt \quad (44)$$

Diese Regel dient als Startpunkt einer experimentellen Anpassung der Gewichtungsmatrizen auf Basis von Simulationsergebnissen.

Die optimale Zustandsrückführung wird durch den Faktor K definiert, welcher das Gütekriterium entsprechend minimiert. In Abhängigkeit der gewählten Matrizen Q und R sowie der Matrix P, die die positive Lösung der Riccati-Gleichung repräsentiert, kann nach folgendem Algorithmus die Zustandsrückführung berechnet werden.

$$\underline{K}_j = \underline{R}^{-1} \cdot \underline{B}_j^T \cdot \underline{P} \quad (45)$$

Mit der Matrix P als Lösung der Riccati-Gleichung:

$$0 = \underline{A}_j \cdot \underline{P} - \underline{P} \cdot \underline{B}_j \cdot \underline{R}^{-1} \cdot \underline{B}_j^T \cdot \underline{P} + \underline{Q} + \underline{A}_j \cdot \underline{P} \quad (46)$$

Daraus ergibt sich das Stellgesetz:

$$\underline{u}^* = -\underline{K}_j \cdot \underline{x} \quad (47)$$

Da diese Gleichungen ein Optimierungsproblem darstellen, müssen diese mit Hilfe numerischer Methoden berechnet werden. Zur Verbesserung des stationären Verhaltens muss eine Vorsteuerung mit der Verstärkung N ausgelegt werden:

$$\underline{N}_j = - \left(\underline{C} \left(\underline{A}_j - \underline{B}_j \cdot \underline{K}_j \right)^{-1} \underline{B}_j \right)^{-1} \quad (48)$$

Da die aufgeschaltete Matrix der Inversen der stationären Verstärkung entspricht, wird durch sie nun der Zustands-Regler stationär genau. [8]

Um die Zustandsrückführung \underline{K}_j und Vorsteuerung \underline{N}_j wie gezeigt zu bestimmen muss hierbei der Zustandsregler durch die Addition des Anteils

$$\underline{u}_z = - \left(\underline{B}_j^T \cdot \underline{B}_j \right)^{-1} \underline{B}_j^T \cdot \underline{E}_j \cdot \underline{z}_i \quad (49)$$

beim Eingangsvektor \underline{u} nach [9] erweitert werden. Infolgedessen werden die Störgrößen \underline{z} kompensiert. Dabei gilt:

$$\underline{B}_j \cdot \underline{u}_z + \underline{E}_j \cdot \underline{z}_i = 0 \quad (50)$$

In der Nähe des quasisationären Arbeitspunkts ist die Störgrößenaufschaltung nicht notwendig. Außerhalb dessen soll der Stellgrößenanteil \underline{u}_z hingegen die Störungen kompensieren, welche durch den aktuellen Betriebspunkt des nichtlinearen Systems verursacht werden. Alle Zustandsregler haben einen Gültigkeitsbereich in der Nähe des jeweiligen Arbeitspunktes.

Abbildung 7 zeigt die Regelstruktur mit Gain-Scheduling welche sich aus der Auswahl der Schedulingparameter und Berechnung der Stellgröße zusammensetzt.

Es wird ein n-dimensionaler Lookup-Table erstellt, in dem alle Regelmatrizen für beliebige Arbeitspunkte durch Interpolation (vgl. [10]) gespeichert werden. Dadurch werden Sprünge innerhalb der Ansteuerung bei Umschaltung zwischen den verschiedenen Regelungsparametern vermieden. Realisiert wird die Ablaufplanung der Zustandsregler abhängig vom aktuellen Zustand des Systems.

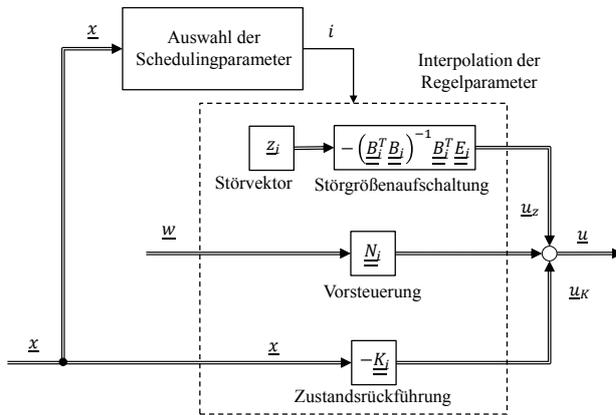


Abbildung 7. Gain Scheduling Prinzip

3.2 Lokaler Regler

Die Sollwertgenerierung des globalen Reglers gibt Drehmomente für die einzelnen Motoren des Systems aus. Als lokaler Regler wird dementsprechend für jeden Aktor ein zeitdiskreter Dead-Beat-Regler in Kombination mit einer Störgrößenaufschaltung ausgelegt, der das Drehmoment stellt.

Die Betrachtung der Synchronmotoren über das in Abschnitt 2.2 ermittelte Park/Clarke transformierte Gleichstrommodell erlaubt eine feldorientierte Regelung. Über diesen Ansatz kann die Feldorientierung durch das Regeln des rotorbezogenen Stromes i_d beeinflusst werden. Das Drehmoment des Motors ist direkt proportional zum Strom i_q . Somit ist durch Regelung dieser beiden Ströme und Transformation der Stellgrößen über inverse Park- und Clarke-Transformation eine Drehmomentregelung möglich.

Dead-Beat-Regler haben die Eigenschaft mit einer endlichen Anzahl von n -Schritten mit einer sprungförmigen Vorgabe der Führungsgröße den Istwert auf den Sollwert zu regeln, somit ergibt sich für die Ausregelzeit:

$$T_{aus} = n \cdot T \quad (51)$$

Hierbei entspricht T der Abtastzeit des Systems. Nach der Ausregelzeit besteht keine Regeldifferenz, weshalb keine weiteren Kompensationsglieder im Regelkreis notwendig sind (vgl. [2]). Das Regelverhalten kann nun durch Vorgabe einer Führungsübertragungsfunktion gestellt werden, die Übertragungsfunktion des Reglers lautet:

$$G_R = \frac{1}{G_{HS}(z)} \cdot \frac{G(z)}{1-G(z)} \quad (52)$$

Nach umformung der Gleichung (52) erhält man für die Regler Übertragungsfunktion:

$$G_R = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2} + \dots + q_n z^{-n}}{1 - p_1 z^{-1} - p_2 z^{-2} - \dots - p_n z^{-n}} \quad (53)$$

Die Übertragungsfunktion $G_R(z)$ lässt sich somit aus der bekannten Übertragungsfunktion $G_{HS}(z)$ und der gewünschten Führungsübertragungsfunktion $G(z)$ beschreiben. [4]

$$G_{HS}(z) = \frac{z^{-1}}{z} \cdot \mathfrak{L} \left\{ \frac{1}{s} \cdot G(s) \right\} = \frac{b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}} \quad (54)$$

Die Koeffizienten des Reglers lassen sich, nach [4] bestimmen wodurch sich für die bekannte Übertragungsfunktion $G_{HS,d,q}(z)$ ergibt:

$$G_{HS}(z) = \frac{\frac{1}{R_s} \left(1 - e^{-\frac{R_s}{L_{d,q}} T} \right) z^{-1}}{1 - e^{-\frac{R_s}{L_{d,q}} T} z^{-1}} = \frac{b_1 z^{-1}}{1 + a_1 z^{-1}} \quad (55)$$

Somit ergibt sich für einen Dead-Beat-Regler ohne Stellgrößenvorgabe:

$$G_R = \frac{q_0 + q_1 z^{-1}}{1 - p_1 z^{-1}} = \frac{\frac{1}{R_s} \left(1 - e^{-\frac{R_s}{L_{d,q}} T} \right) z^{-1} e^{-\frac{R_s}{L_{d,q}} T}}{1 - e^{-\frac{R_s}{L_{d,q}} T} z^{-1}} \cdot \frac{1}{1 - e^{-\frac{R_s}{L_{d,q}} T} z^{-1}} \quad (56)$$

Um diesen Regler als Drehmomentregler einsetzen zu können, wird der Rückführgröße eine Verstärkung vorgeschaltet:

$$G_c = \frac{3}{2} \cdot p \cdot \phi_0 \cdot \ddot{u}_G \quad (57)$$

Mit dieser konstanten Größe kann der gemessene Strom proportional in das gestellte Drehmoment umgerechnet werden.

4 Analyse des Systemverhaltens

Nach der Auslegung der hierarchischen Regelstruktur sollen die Algorithmen mittels Model-in-the-Loop-Simulation validiert und das Systemverhalten des geregelten S-Mobiles analysiert werden.

Ziel der Aufbauregelung ist primär ein Verfahren des S-Mobile bei gleichzeitiger Stabilisierung des Aufbaus zu ermöglichen. Zu diesem Zweck wurde eine Positionsvorgabe als Sollwert auf den Regler gegeben. Abbildung 8 zeigt die Antwort des Systems auf diesen Verlauf.

Die Positionsänderung erfolgt kontinuierlich mit einer konstanten Steigung ab 0.5s bis zu einem Vorgabenwert von $x_s = 0.4m$. Die Maschine folgt diesem Verlauf verzögert innerhalb von 4.5s nach. Dabei verfährt die Maschine zunächst rückwärts, um den Schwerpunkt des Aufbaus in Fahrtrichtung vorzuneigen. Dadurch wird eine gewollte Instabilität des Aufbaus erzeugt, welche mit den Beschleunigungskräften kompensiert wird und somit eine hohe Systemdynamik

mik ermöglicht. Dies ist auch am Drehwinkel des Aufbaus zu erkennen. Während der Beschleunigung wird dieser zunächst um bis zu 3.9° in Fahrtrichtung geneigt um dann durch die entstehenden Beschleunigungskräfte bis auf -1.6° zurückgedreht zu werden. Die Reglerstruktur ermöglicht danach wieder ein Ausregeln des Aufbaus bei 0° Aufbauwinkel, also dem quasistationären Punkt. Bei 9s wird das S-Mobile angeregt die Position $x_s=0m$ anzufahren, was durch die Symmetrie der Maschine equivalent verläuft.

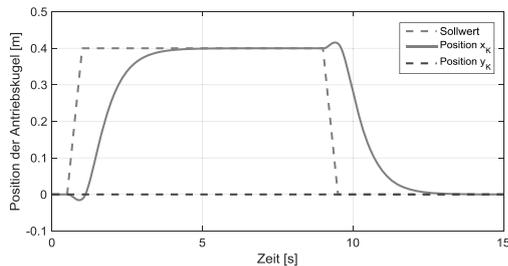
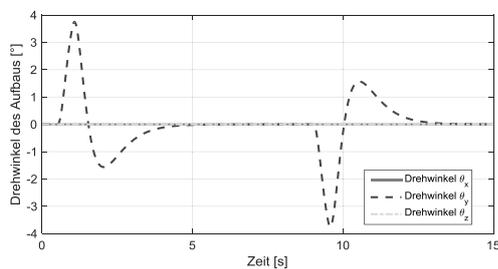


Abbildung 8. Positionsvorgabe

Weiterhin wurde eine Vorgabe der Geschwindigkeit der Aufbaubewegung in eine bestimmte Richtung vorgegeben. Die Nachfolgende Abbildung 9 zeigt den Positions- und Drehwinkelverlauf bei einer konstanten Geschwindigkeit.

Bei 1s wird eine kontinuierliche Geschwindigkeitsvorgabe als Rampe auf das Regelsystem geschaltet. Das System reagiert abermals mit dem Kippen des Aufbaus in Fahrtrichtung durch eine Fahrbewegung in die Gegenrichtung. Anschließend folgt das System mit konstanter Geschwindigkeit der Vorgabe. Der Aufbauwinkel wird währenddessen wieder auf den quasistationären Punkt bei 0° geregelt. Da bei einer konstanten Bewegung keine Beschleunigungskräfte auftreten, kann der Aufbauwinkel während der gesamten Bewegung konstant gehalten werden. Das Stoppen des S-Mobile erfolgt in gleicher Weise bedingt durch die Symmetrie der Konstruktion.

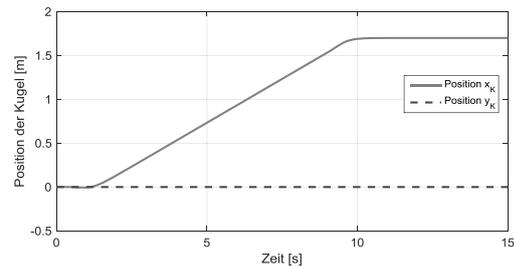
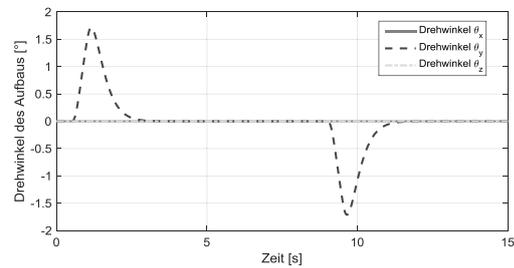


Abbildung 9. Geschwindigkeitsvorgabe

5 Zusammenfassung und Ausblick

Im vorliegenden Beitrag wurde das neue Konzept des S-Mobiles vorgestellt. Anhand des erweiterten MKS-Modells mit der expliziten Kinematik- und Dynamikfunktion des Systems wurde eine hierarchische Regelstruktur ausgelegt. Die Regelstruktur besteht aus einem globalen und lokalen Reglern, welche vor allem durch ihre leichte Adaptierbarkeit auf einem autarken Microcontroller lauffähig ist. Die gefundenen Ergebnisse wurden anhand diverser MiL-Simulationen validiert. In der weiteren Forschungsarbeit soll nun ein realer Forschungsträger auf Basis der gefunden Erkenntnisse umgesetzt und die Ergebnisse weitgehend validiert werden.

6 Referenzen

- [1] M. Göllner und X. Liu-Henke. *Konzeption eines hochdynamischen Systems mit sphärischem Elektroantrieb*. ASIM STS/GMMS Workshop 2014, Reutlingen, 2014
- [2] X. Liu-Henke, J. Lückel und K.-P. Jäker. *Ganzheitlicher mechatronischer Entwurf eines aktiven Feder-/Neigemoduls*. VDI-Tagung Mechatronik – Mechanisch/Elektrische Antriebstechnik, in Wiesloch, Deutschland, 2000
- [3] X. Liu-Henke, R. Buchta und F. Quantmeyer. *Simulation eines mechatronischen Lenkungsmoduls für ein Elektrofahrzeug mit dezentra-*

- len Direktantrieben*. ASIM, in Krefeld, Deutschland, 2011
- [4] M. Göllner und X. Liu-Henke. *Mathematical derivation of the geometry of a Mecanum-wheel for a size exact roll off on a spherical surface*. MSM. Opole, Polen, 2014
- [5] E. Pellegrini, K. Diepold, R. Dessort, und H. Panzer. *3D-Modeling of a Robot Balancing on a Ball*. München, 2.08.2011. url: <https://mediatum.ub.tum.de/doc/1081938/1081938.pdf>.
- [6] J. Blath. *Zustandsregelungen*. TU Clausthal, Deutschland, 2012
- [7] T. Holzhütter. *Zustandsregelung*. Fachhochschule Hamburg, Deutschland, 2009
- [8] O. Föllinger und F. Dörrscheidt. *Regelungstechnik: Einführung in die Methoden und ihre Anwendung*. 10. durchges. Auflage, Nachdr. der 8., überarb. Auflage. 1994
- [9] J. Adamy. *Nichtlineare Systeme und Regelungen*. Berlin, Springer Vieweg, 2014.
- [10] J. Wurster. *Entwurf einer zeitdiskreten Stromregelung für einen Permanentmagnet-Synchronmotor*. Bachelorarbeit, Ulm, 2011

Modellbasierte Diagnoseverfahren für nichtlineare DAE-Systeme

Dirk Weidemann¹, Ilja Alkov¹

¹Institut für Systemdynamik und Mechatronik
FH Bielefeld

dirk.weidemann@fh-bielefeld.de

Dem Erkennen von Fehlern, die während des Betriebs eines technischen Systems auftreten, kommt sowohl aus sicherheitstechnischen wie auch aus wirtschaftlichen Aspekten eine hohe Bedeutung zu. Typische sicherheitstechnische Aspekte sind bspw. der Schutz des Betreibers bzw. Benutzers des technischen Systems aber auch der Umwelt. Demgegenüber stehen bei den wirtschaftlichen Aspekten die Reduktion der Stillstandszeiten und Wartungskosten im Vordergrund. Während bei einigen technischen Systemen der Betreiber bzw. Benutzer die sichtbaren Symptome den zugrundeliegenden Fehlern zuordnen und somit die Fehlererkennung eigenständig durchführen kann, gelingt dies insbesondere bei komplexeren Systemen im Allgemeinen nicht mehr. Vor diesem Hintergrund kommen bei komplexen Systemen zunehmend modellbasierte Verfahren zum Einsatz die eine automatische Fehlererkennung ermöglichen. Da der bei komplexen Systemen vielfach angewendete objektorientierte Modellierungsansatz auf nichtlineare differentialalgebraische Gleichungen führt, werden im vorliegenden Beitrag modellbasierte Diagnoseverfahren für nichtlineare DAE-Systeme vorgestellt. Die Diagnoseverfahren basieren auf einer hybriden Zustandsschätzung und gestatten neben der reinen Fehlererkennung auch die Fehlerlokalisierung und -identifikation. Zusätzlich zum *Autonomous-Multiple-Model*-Ansatz und dem *Interacting-Multiple-Model*-Ansatz wird darüber hinaus der *Generalized-Pseudo-Bayesian*-Ansatz 1. und 2. Ordnung für nichtlineare DAE-Systeme beschrieben.

Towards Extending the Parallel-DEVS Formalism to Improve Component Modularity

Franz Preyser^{*}, Bernhard Heinzl, Philipp Raich, Wolfgang Kastner
Institute of Computer Aided Automation, Vienna University of Technology,
Treitlstraße 1-3, 1040 Vienna, Austria

^{*} *franz.preyser@tuwien.ac.at*

When applying classic DEVS modelling in the context of industrial facility simulation, we experienced the well-known drawbacks of DEVS concerning the treatment of simultaneous events. Switching to Parallel DEVS, some of those problems could be avoided, however other, similar problems in connection with mealy components and transient states occurred. In the following work, we try to locate the root of these problems and give a basic example to demonstrate them. Further, possible remedies in the form of modelling work-arounds inside the PDEVS formalism are described, although none of them seems to solve the problems sustainably. Eventually extensions of the formalism itself are proposed with the goal of easing the design of models including mealy components. Finally the consequences of such modifications for the simulation algorithms are discussed.

1 Introduction

In the course of the research project BaMa (Balanced Manufacturing), in which simulation is used for optimizing energy efficiency in industrial facilities [1], we had to deal with the problem of how to model and simulate hybrid systems. We decided on an integrated approach, using a hybrid formalism based on DEVS (Discrete Event System Specification).

DEVS was introduced by Zeigler as a modular, hierarchical, and very powerful modelling formalism [2]. However, modeling systems of higher complexity can be difficult and error-prone. One reason for this lies in the way of how concurrent events are resolved. To combat those difficulties, Chow and Zeigler extended the DEVS formalism in 1994 [3] and introduced PDEVS.

Still, we experienced significant difficulties during the modelling process with both formalisms. We already documented some of our problems with DEVS in [4], and switching to PDEVS instead unfortunately caused a different set of problems, which we will describe in this paper.

Already in 2007 Traoré [5] mentioned shortcomings of PDEVS. Traoré also criticised the necessity of transient states for modelling mealy-behaviour and even proposed a formalism extension named *EasyDEVS*. Although *EasyDEVS* supports mealy-behaviour without transient states, there are still ways to produce them implicitly, and therefore, it does not improve modularity. The use of transient states in practice of DEVS-modelling seems to be so common that in the graphical DEVS description language *DDML* [6] they even introduced a special symbol for *zero states*, as they call them.

Another work dealing with PDEVS modeling difficulties is [7]. Cicirelli et al. mention problems arising when using PDEVS for modelling *Timed Petri Nets* and locate the root of these problems in the input-independent calculation of output messages. They point out that it is not possible for the confluent transition function to behave exactly like the external transition function, as in the first case the output function is executed and there is no way to prevent that.

In this work, we locate a common root for the modelling problems described in [5] and [7] and eventually suggest a modification of PDEVS to solve them.

2 DEVS, PDEVS Recapitulation

2.1 Classic DEVS

The most important property of the classic DEVS formalism, which is passed on to all its extensions and variations, is its hierarchical and modular structure. The basic component is an *Atomic*, which can be visualised as block with input and output port(s). *atomic blocks* can be coupled resulting in a *coupled DEVS* that again has input and output ports and can be treated like an *atomic DEVS*. An *atomic DEVS* is specified by the following 7-tuple [8]:

$$\langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta \rangle$$

X ... set of possible inputs (e.g. \mathbb{R}^n)

Y ... set of possible outputs (e.g. $\mathbb{R}^+ \times \mathbb{N} \times \mathbb{R}^m$)

S ... set of possible states (=state space)

$$Q = \{(s, e) | s \in S, e \in [0, ta(s)]\}$$

$ta : S \rightarrow \mathbb{R}_0^+ \cup \{\infty\}$... *time advance function*

$\lambda : S \rightarrow Y$... *output function*

$\delta_{ext} : Q \times X \rightarrow S$... *external state transition function*

$\delta_{int} : S \rightarrow S$... *internal state transition function*

Regarding a detailed description of the single components of the formalism we refer to [8]. What we want to point out though, is that the output function λ depends only on the inner state of the system.

A coupled DEVS N is described by the following [8]:

$$\langle X, Y, D, \{M_d\}_{d \in D}, \{I_d\}_{d \in D \cup \{N\}}, \{Z_{i,d}\}_{i,d \in D \cup \{N\}}, Select \rangle$$

X ... set of possible inputs

Y ... set of possible outputs

D ... index set

M_d ... child DEVS of N for each $d \in D$

$I_d \subset D \cup \{N\}$... influencer set of d , $d \notin I_d$

$Z_{i,d}$... output translation function

$Select : 2^{D \cup \{N\}} \rightarrow D \cup \{N\}$... tie breaking function

Again, we refer to [8] for a more detailed explanations.

When processing an internal event, first λ is computed which leads to an instantaneous execution of δ_{ext} at every receiving block (which must not be the sending block itself). Then δ_{int} of the original block is executed.

The *Select* function specifies a strict order in which concurrent internal state transitions inside a coupling are performed. As external events only occur while another block is executing an internal event, no atomic block will ever be confronted with external events during executing an internal event. However, this also means that the order in which concurrent internal and external events at a block are processed entirely depends on the coupling specification. This introduces big difficulties for modellers trying to design reusable atomic blocks, because they do not know in which context the block will be used. Moreover, it puts a strictly sequential execution order constraint onto the simulation algorithm prohibiting certain model structures as shown in figure 1.

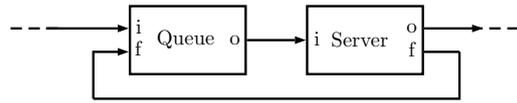


Figure 1: A server-queue model with a feedback path.

The problem with the example in figure 1 occurs when the *Queue* wants to send an entity to the *Server*, while the *Server* sends an entity to the queue at the exact same time. It is assumed that the *Server* cannot accommodate more than one entity at a time. If the maximal capacity of the *Queue* is reached, there will be a deadlock or an entity will be lost, depending on the particular model design. No matter who is chosen by the *Select* function to execute its internal event first, *Server* or *Queue*, the respective counterpart will not be able to store the transmitted entity.

2.2 Parallel DEVS

Parallel-DEVS (PDEVS) is supposed to overcome the problems known from classic DEVS. The idea is to relocate the resolution of concurrencies from the coupling level to the atomic level. As a consequence, all atomic blocks scheduled for an internal event execute their output function in parallel, potentially leading to multiple concurrent input messages at receiving blocks. These concurrent input messages are collected in a *bag* denoted with x^b , which is then processed in one single call of an appropriate state transition function. To treat the case of an simultaneous internal event, the *confluent transition function* δ_{conf} is introduced, enabling the modeller to resolve con-

currencies at the atomic level. So, the main difference from PDEVS to classic DEVS is the additional transition function δ_{conf} replacing the *Select* function. For a detailed specification see [3], [8].

A simulation step of a coupled PDEVS model can be divided into two sub-steps. In the first sub-step (λ -step) all blocks scheduled for an internal event (*imminent* blocks) calculate their λ , producing output messages collected in input bags at the receiving blocks. In the second sub-step (δ -step), the appropriate state transition function (δ_{ext} , δ_{int} or δ_{conf}) is executed in every imminent and in every receiving block consuming the content of the input bags. The possibility to buffer the messages in bags enables the modelling of the *Server-Queue* example shown in figure 1, without running into the troubles present in classic DEVS.

However, λ still solely depends on the state of the system and not on the current input bag. As a consequence, mealy behaviour, that is an instantaneous reaction of a block to an input event with an output event, can only be realized indirectly by going through a *transient state*. Such a transient state stores all the necessary input bag information and immediately triggers an internal event to cause an execution of λ . Concerning the PDEVS simulation, this means that output messages of mealy blocks triggered by an input event cannot be calculated during the first λ -step. It first takes a δ -step to change into a transient state and then a second λ -step to create the desired output message. Therefore, the two steps (λ and δ) are repeated until there is no transient state left in the system, before simulation time can advance.

Figure 2 shows an example with n mealy blocks coupled in a row. When the first mealy block experiences an input event, it takes two λ -step – δ -step iterations for the event to arrive at the second block. Consequently, it takes $n + 1$ such iterations until the event leaves the n -th mealy block. Although simulation time is not advancing during this process, events are delayed when passing mealy blocks regarding the treatment order at the final destination block (block $n + 1$). Due to the delay caused by the mealy blocks, the actually concurrent messages are not processed in the same δ -step, but in several consecutive δ -steps. Thus, we are ending up with a partially sequential processing order, which again depends on the coupling structure of the interacting blocks. This reintroduces the modelling difficulties known from DEVS.

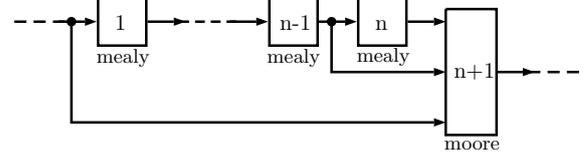


Figure 2: A cascade of mealy blocks concluded by a moore block.

3 Example

For a modelling paradigm to be broadly accepted in practice, it is important to be able to reuse models and components formulated with it. In the following example we will demonstrate how the reuse of two very basic, independently designed blocks may produce unintended behaviour in PDEVS.

3.1 Formal Description

The first introduced block (*Comparator*) simply compares a real input value with a given constant C and outputs 1 if the input value is greater than C and 0 otherwise. Therefore, a corresponding PDEVS model looks like the following:

$$\begin{aligned} X &= \mathbb{R}, Y = \{0, 1\} \\ S &= \mathbb{R} \times (\mathbb{R}_0^+ \cup \{\infty\}) \\ ta(s) &= ta(v, \sigma) = \sigma \\ \lambda(s) &= \lambda(v, \sigma) = \begin{cases} 1 & \text{if } v > C \\ 0 & \text{if } v \leq C \end{cases} \\ \delta_{int}(s) &= \delta_{int}(v, \sigma) = (v, \infty) \\ \delta_{ext}(s, e, x^b) &= \delta_{ext}(v, \sigma, e, x^b) = (v_{in}, 0) \quad x^b = \{v_{in}\} \\ \delta_{conf}(s, x^b) &= \delta_{conf}(v, \sigma, x^b) = (v_{in}, 0) \quad x^b = \{v_{in}\} \end{aligned}$$

The second block (*Switch*) represents a switch passing on input values to one of its two output ports. The active output port is selected by a separate control input port. The corresponding PDEVS model looks like the following:

$$\begin{aligned} X &= (\mathbb{R} \times \{0\}) \cup (\{0, 1\} \times \{1\}), Y = \mathbb{R} \times \{0, 1\} \\ S &= \mathbb{R} \times \{0, 1\} \times (\mathbb{R}_0^+ \cup \{\infty\}) \\ ta(s) &= ta(v, c, \sigma) = \sigma \\ \lambda(s) &= \lambda(v, c, \sigma) = (v, c) \\ \delta_{int}(s) &= \delta_{int}(v, c, \sigma) = (v, c, \infty) \end{aligned}$$

$$\begin{aligned} \delta_{ext}(s, e, x^b) &= \delta_{ext}(v, c, \sigma, e, x^b) \\ &= \begin{cases} (v_{in}, c, 0) & \text{if } x^b = \{(v_{in}, 0)\} \\ (v, c_{in}, \infty) & \text{if } x^b = \{(c_{in}, 1)\} \\ (v_{in}, c_{in}, 0) & \text{if } x^b = \{(v_{in}, 0), (c_{in}, 1)\} \end{cases} \\ \delta_{conf}(s, x^b) &= \delta_{ext}(s, 0, x^b) \end{aligned}$$

The second entry of each pair in the input bag x^b indicates the input port (0 or 1) at which this pair arrived. Now consider the two blocks coupled as shown in figure 3. Obviously the intention is to route values bigger than C to the second output port of the switch and other values to the first output port of the switch.

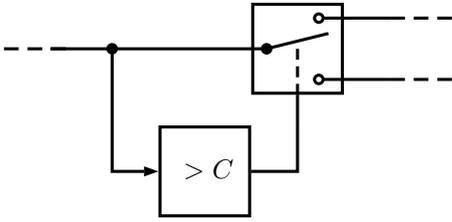


Figure 3: *Comparator-Switch* model. A *Switch* controlling the flow-path of messages depending on their values.

Since the *Comparator* is of type mealy, the result of the comparison of an input message reaches the *Switch* only in the second iteration of the two simulation steps (λ -step and δ -step). However, the *Switch* already sends the received input value during the second λ -step at the output port that is active at that time. But the control input is processed not until the second δ -step, which is too late.

3.2 Example Implementation

To demonstrate the described behaviour of the *Comparator-Switch* model we implemented it in MATLAB using the MatlabDEVS-Toolbox ([9], [10]). Figure 4 shows a diagram visualising the simulation results.

The behaviour we would expect is that all messages with value greater than 3 are routed to output port 1 and that all smaller values are routed to output port 0. However, as we can see, the first value routed to output port 1 is 6, confirming the behaviour predicted at the end of section 3.1.

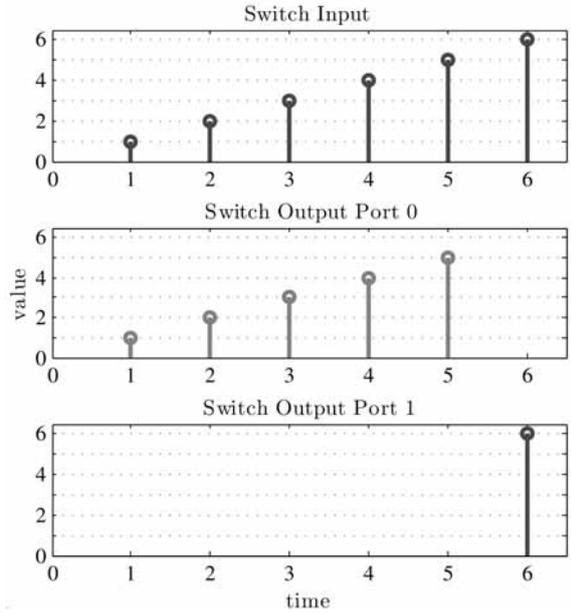


Figure 4: Simulation results of the *Comparator-Switch* model implemented with the MatlabDEVS-Toolbox. The *Comparator* parameter C is set to 3.

3.3 Potential Remedies

Now the question arises of how to modify the *Comparator-Switch* model to achieve the desired behaviour. One possibility would be to merge the two blocks, *Comparator*, and *Switch* into a single atomic PDEVs. Then the switch could be set correctly before any value is passed on to the output ports. This solution though would have a negative impact on the reusability of the *Comparator-Switch* block. For example if the logic controlling the position of the switch needs to be altered, a redesign of the whole block would be necessary, whereas otherwise only the *Comparator* would need to be replaced.

Another solution would be to redesign the *Switch* to wait for the control signal before passing on an input message to an output port. This could be accomplished by modelling two consecutive transient states for the *Switch* when an input message arrives. This way, passing on the input message to an output port can be delayed until the control input arrives. However, if the *Switch* is reused in another model, where more than one mealy-block in a row calculate the control signal, the *Switch* again would show the unwanted

behaviour. Furthermore, the *Switch* itself is a mealy block and when used in a bigger model, further delays would also influence the subsequent model parts.

A third way could be to introduce a real delay. That is, to modify the *Switch* in such a way that it enters a non-transient state after each reception of an input message. Only after the very short life-time of that non-transient state, the input message would be passed on to the output. In this way, it is ensured that the control signal arrives before producing any output, no matter how many mealy blocks are involved in producing the control signal. However, the meaning of “short” concerning the life-time of the non-transient state strongly depends on the surrounding system and could change if the *Switch* is used in a different context. Additionally the real delay may have a negative impact on the outcomes of the simulation.

4 Tackling the Formalism

Another way out of the dilemma pointed out in the sections above could be a alteration of the formalism itself, which would also require modifications in the simulation algorithms.

4.1 Possible Extensions of the Formalism

As pointed out in preceding sections, both in classic DEVS and in Parallel DEVS, the output function λ only has access to the state of the system when calculating an output message. Therefore, if a input message is to affect or cause an immediate output message, it has to be stored in the state of the system during δ_{ext} respectively δ_{conf} . This state has to be a transient one in order to trigger an execution of λ , which then calculates an output message from it. Exactly these circumstances are causing the problematic delays and make it impossible to collect all messages produced in a model at a certain simulation time *before* executing any state transition.

The goal is to enable mealy blocks to calculate output messages in reaction to input messages without having to undergo a state transition in between. For this purpose we propose to change the definition of λ from:

$$\lambda : S \rightarrow Y^b, s \mapsto \lambda(s) = y^b$$

to:

$$\lambda : S \times X^b \rightarrow Y^b, (s, x^b) \mapsto \lambda(s, x^b) = y^b$$

where also $x^b = \emptyset$ is allowed. Concerning the simulation algorithm, the λ -step would be split into three sequential sub-steps resulting in the following scheme:

λ -step :

moore-step : execute λ of all currently imminent blocks (blocks with an internal event) filling up input bags x^b at all receiving blocks.

mealy-step 1 : if no input bag x^b has been altered in the previous step, go to δ -step.

mealy-step 2 : (re-)execute λ of all mealy blocks whose input bag x^b changed in the previous step and go back to **mealy-step 1**.

δ -step : calculate δ_{int} at each imminent block with empty input bag, δ_{ext} at each non-imminent block with not empty input bag, and δ_{conf} at each imminent block with not empty input bag.

Since λ now depends on the input bag it is necessary to recalculate it every time the bag is altered due to the calculation of λ of an influencing block. As circular dependencies (algebraic loops / illegitimate models) are not allowed in DEVS/PDEVS, the loop **mealy-step 1 – mealy-step 2** has to terminate at some point. After termination, a *stable* set of input bags for all blocks should be the result. That is, in the entire model there is no mealy block left whose input bag differs from the one used for its last calculation of λ . In this way it is assured that every single block already received all input messages before it conducts a state transition. Transient states should be obsolete in this modified formalism. At least, they are no longer necessary to model mealy behaviour.

There are two issues we want to discuss in more detail. First, the simulation algorithm must be able to distinguish between moore and mealy type blocks. Therefore, each atomic block needs to tell the simulation engine somehow whether it is of type mealy or not. However, as we can see at the *Switch* example, it may depend on the content of the input bag whether a block will react with an output message or not. A control signal alone will not trigger any output reaction at the *Switch* block. Furthermore, it is also possible that the mealy property depends on the current system state

as well. Therefore we propose to introduce a *mealy function*

$$\mu : S \times X^b \rightarrow \{true, false\}, (s, x^b) \mapsto \mu(s, x^b),$$

which needs to be defined by the modeller for each atomic component. This function can be called by the simulation engine in **mealy-step 2** to determine whether a block is currently of type mealy or not.

The second issue to be explained in more detail is what exactly can be understood by an alteration of an input bag in **mealy-step 1** and which special cases are possible. Due to **mealy-step** iterations, for some blocks λ is possibly calculated several times. Therefore, it is necessary that the result of such a calculation always replaces a previously calculated value and is not appended to the previously computed outputs. As a consequence, it is possible that, due to a change in the input bag, μ suddenly returns *false* although it returned *true* previously. However, if μ was true also an output message has been produced already. If the concerned block is not currently imminent, this former output message has to be retrieved. In some cases this may lead to a completely empty input bag at one of the receiving blocks. If such a block as well already produced an output message and is not imminent, its output messages have to be retrieved as well and so on. Thus, when recalculating λ , previous messages not only have to be replaced, but some of them may also have to be retrieved.

It can be seen that there are some special cases to be considered and a rigorous definition of a simulation algorithm and its implementation are still work in progress. Furthermore, it can not be ruled out that the formalism modification creates new hidden weaknesses. Therefore, extensive tests need to be conducted.

4.2 Consequences for the previous examples

The modified PDEVS formalism still enables buffering messages in bags. Therefore, the *Server-Queue* model from section 2.1 can be implemented.

Applying the new formalism to the *Comparator* model, it would simplify to:

$$X = \mathbb{R}, Y = \{0, 1\}$$

$$\begin{aligned} S &= \emptyset \\ ta(s) &= \infty \\ \lambda(s, x^b) &= \lambda(\{v\}) = \begin{cases} 1 & \text{if } v > C \\ 0 & \text{if } v \leq C \end{cases} \\ \delta_{int}(s) &= s \dots \text{ will never be triggered} \\ \delta_{ext}(s, e, x^b) &= s \dots \text{ does nothing} \\ \delta_{conf}(s, x^b) &= s \dots \text{ will never be triggered} \\ \mu(s, x^b) &= true \end{aligned}$$

and the *Switch* model would change to:

$$\begin{aligned} X &= (\mathbb{R} \times \{0\}) \cup (\{0, 1\} \times \{1\}), Y = \mathbb{R} \times \{0, 1\} \\ S &= \{0, 1\} \\ ta(s) &= \infty \\ \lambda(s, x^b) &= \lambda(c, x^b) \\ &= \begin{cases} (v_{in}, c) & \text{if } x^b = \{(v_{in}, 0)\} \\ (v, c_{in}) & \text{if } x^b = \{(c_{in}, 1)\} \\ (v_{in}, c_{in}) & \text{if } x^b = \{(v_{in}, 0), (c_{in}, 1)\} \end{cases} \\ \delta_{int}(s) &= \delta_{int}(c) = c \dots \text{ will never be triggered} \\ \delta_{ext}(s, e, x^b) &= \delta_{ext}(c, e, x^b) \\ &= \begin{cases} c_{in} & \text{if } x^b = \{(c_{in}, 1)\} \\ c_{in} & \text{if } x^b = \{(v_{in}, 0), (c_{in}, 1)\} \\ c & \text{else} \end{cases} \\ \delta_{conf}(s, x^b) &= \delta_{ext}(c, 0, x^b) \dots \text{ will never be triggered} \\ \mu(s, x^b) &= \mu(c, x^b) \\ &= \begin{cases} false & \text{if } x^b = \{(c_{in}, 1)\} \\ true & \text{else} \end{cases} \end{aligned}$$

5 Discussion

The overall goal of the proposed modifications to PDEVS is to make it easier for a modeller to design reusable models and thus, improving modularity of the formalism as a whole. Although a complete simulation engine still needs to be designed, it can already be seen that such an engine will most likely be more complex than established PDEVS engines. This is due to the iterations taking place in the λ -step which require a more complex bag management. Moreover, the complexity of the λ functions increases because of the dependency on the input bag. However, each λ -step iteration in the modified PDEVS

formalism replaces a λ -step – δ -step iteration in the original PDEVS formalism. Therefore, the number of δ -steps would be reduced. Furthermore, the expected improvements on reusability of models should shorten and simplify the model design and implementation phase.

6 Conclusion and Future Work

In this paper we pointed out modelling difficulties with PDEVS occurring when using mealy blocks in a model. We gave an example demonstrating these difficulties by showing how two independently designed components produce incorrect behaviour when used together in a coupled model. This example was then implemented and simulated using the MatlabDEVS-Toolbox to demonstrate the predicted behaviour.

Further, we described three ways of solving the problems by altering the PDEVS model. However, with regard to the reusability of the model components, all three solutions seem to be unsatisfying. Thus, we proposed modifications of the PDEVS formalism itself with the goal to improve its modularity. We described what requirements the revised formalism would impose on the simulation algorithm and how the specification of the example would change.

Finally, we discussed possible advantages and disadvantages of the proposed formalism modifications and its consequences for the simulation algorithm. The next necessary step will be a profound and formal definition of a corresponding abstract simulator and a corresponding implementation to prove its validity.

Acknowledgement We want to thank all partners of the project BaMa for their contributions. The research presented is funded by the Austrian Climate and Energy Funds within the program e!MISSION.at Energy Mission Austria, project number 840746.

References

- [1] B. Heinzl, I. Hafner, P. Smolek, *et al.*, “Towards a Common Description of Interdisciplinary Aspects Relevant for Holistic Energy Analysis of Production Facilities”, in *Beiträge zum Workshop der ASIM/GI-Fachgruppen STS und GMMS*, A. L. Claudia Rahn, Franz Jendersie, Ed., Wien: ARGESIM/ASIM Pub, 2015.
- [2] B. P. Zeigler, *Theory of Modeling and Simulation*. Krieger Pub Co, 1976, ISBN: 978-0898748086.
- [3] A. C. H. Chow and B. P. Zeigler, “Parallel DEVS: a parallel, hierarchical, modular, modeling formalism”, in *Proceedings of Winter Simulation Conference*, Society for Computer Simulation International, Dec. 1994, pp. 716–722, ISBN: 0-7803-2109-X. DOI: 10.1109/WSC.1994.717419.
- [4] F. Preyser, I. Hafner, and M. Rößler, “Implementation of Hybrid Systems Described by DEV&DESS in the QSS Based Simulator PowerDEVS”, in *Beiträge zum Workshop der ASIM/GI-Fachgruppen STS und GMMS*, A. L. Claudia Rahn, Franz Jendersie, Ed., Stralsund: ARGESIM/ASIM Pub Wien, 2015, p. 8.
- [5] M. K. Traoré, “Easy DEVS”, *Proceedings of the 2007 spring simulation*, vol. 1, pp. 214–216, 2007.
- [6] M. K. Traoré, “A graphical notation for DEVS”, in *SpringSim '09 Proceedings of the 2009 Spring Simulation Multiconference*, Society for Computer Simulation International, Mar. 2009, p. 7.
- [7] F. Cicirelli, A. Furfaro, and L. Nigro, “Conflict management in PDEVS: an experience in modelling and simulation of time petri nets”, pp. 349–356, Jul. 2007.
- [8] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, 2000, ISBN: 9780127784557.
- [9] T. Pawletta, C. Deatcu, O. Hagendorf, *et al.*, “DEVS-Based Modeling and Simulation in Scientific and Technical Computing Environments”, in *DEVS Integrative M&S Symposium (DEVS'06)*, D. Hamilton, Ed., Huntsville/AL, USA, 2006, pp. 151–158.
- [10] C. Deatcu and T. Pawletta, “A Qualitative Comparison of Two Hybrid DEVS Approaches”, *SNE - Simulation Notes Europe*, vol. 22, no. 1, pp. 15–24, 2012. DOI: 10.11128/sne.22.tn.10107.

Pitfalls using discrete event blocks in Simulink and Modelica

Peter Junglas
PHWT Vechta/Diepholz/Oldenburg
peter@peter-junglas.de

Though Simulink as well as Modelica are basically tools for the modeling of continuous systems, they both contain several elements that allow for discrete and hybrid modeling. Comparing the respective block libraries one finds almost identically looking components, but this similarity is deceptive. Using simple digital circuits with flip-flops as examples, one finds subtle differences in the handling of events, which can lead to an unexpected behaviour of a model. A good understanding of the underlying mechanisms, especially of the Modelica standard description, is therefore essential for the successful modeling of discrete systems. But even then it can be difficult to create models which have the required behaviour.

1 Introduction

Modern simulation programs with their graphical user interfaces and large block libraries have simplified the task of creating complex models considerably. Superficially the basic procedure is almost identical even for tools with very different philosophies like the signal-oriented Simulink and Modelica-based physical modeling programs [1] like Dymola. This can lead to a concentration on the apparent block structure of a model, disregarding details behind the scene like solver properties or the defining language. That this negligence can result in modeling errors that are hard to understand will be shown in the following.

While Simulink and Modelica have their roots strongly in continuous modeling, they both have basic discrete features, which can be enhanced with additional packages for state machines [2, 3] or process based models [4, 5]. Already the basic libraries contain blocks like `Memory` and `Unit Delay` in Simulink resp. `Pre` and `UnitDelay` in Modelica, which look very similar – but their underlying mechanisms are quite different.

In the following a few examples of simple digital circuits containing flip-flops will demonstrate that seemingly identical models can lead to different results. After a close look at the Modelica language specification a detailed analysis of the models will explain, why they behave differently from their corresponding

Simulink counterparts. Finally it will be shown how to cope with the subtleties and create robust models of flip-flops in Modelica.

Though all ideas presented here surely are well-known to experts, the consequences of "well-known facts" are not always self-evident. Therefore this study may be useful not only for students (and teachers), but for practitioners as well, especially when they are employing different simulation tools.

2 Simulating flip-flops in Simulink und Modelica

Trying to model digital circuits containing flip-flops is a rewarding endeavor, because it illustrates the intricacies of discrete modeling in a nutshell. Though the models presented in this section are quite simple, some of them show a behaviour that probably may come as a surprise.

2.1 The static RS flip-flop

The simplest flip-flop is the static RS flip-flop, which is usually shown as two nor gates with crosswise connected outputs. This can not be modeled directly, because it leads to an algebraic loop over discrete variables. To break it, one can simply add a "very short"

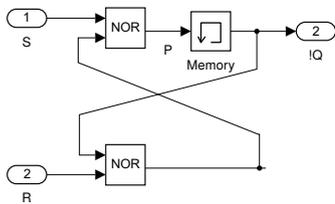


Figure 1: Static RS flip-flop (A)

The results in fig. 2 (version A) show for all possible input combinations $S = 1$ at $t = 10$, which is called *forbidden* corresponding outputs Q and $!Q$ are not inverse to each other. The delay is clearly, at $t = 7$ they lead to another lag of $!Q$ behind Q .

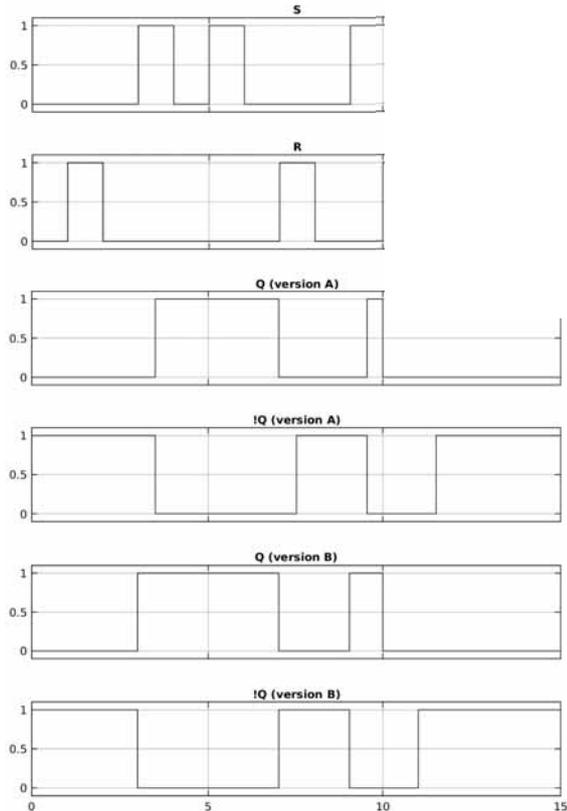


Figure 2: Simulation results of static RS flip-flops

Actually the delays are astonishingly large: The fun-

damental time period of the Simulink solver is set to 0.1, but the delay is 0.5. This value is inherited from the blocks defining the input values S and R . With some extra effort one can reduce it, but defining a very small time step leads to long execution times, while the inconsistency still remains – however shortly.

Therefore the Simulink block library contains a differ-

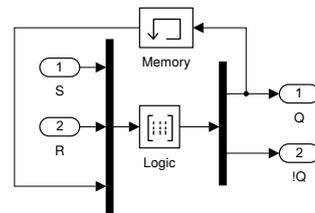


Figure 3: Static RS flip-flop (version B)

In Modelica one can reproduce the simple version A using the `Pre` block to implement the delay – actually this `RSFlipFlop` is already included in the basic `Blocks` library. It directly leads to the desired results as in version B with no output delay at all! Apparently, Simulink and Modelica work differently here. With a little effort one can replicate version B as well, but it simply reproduces the former correct results.

2.2 The triggered RS flip-flop

To cope with timing problems the *triggered* flip-flops have an additional input, often denoted `CLK`, for synchronisation purposes. The state of such a flip-flop changes only, when this input changes in a predefined way, e.g. from `true` to `false` (*negative edge*). Fig. 4 shows example signals for the inputs S , R and `CLK` and the corresponding outputs Q and $!Q$. Basically the flip-flop behaves like before, but the changes take place only at the negative edges of the `CLK` input.

In Simulink such a flip-flop can be easily modeled by adding a `Trigger` block to the subsystem of the static RS flip-flop. This creates an additional *trigger* input

for the block, which leads exactly to the required behaviour.

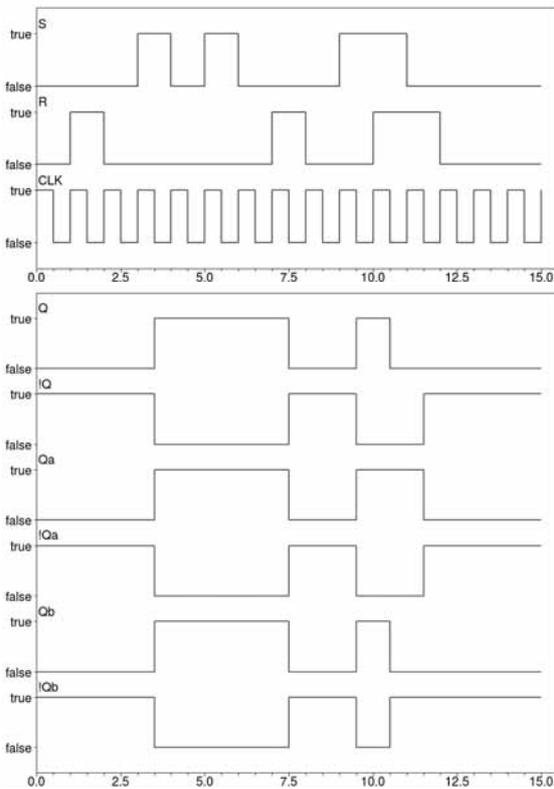


Figure 4: Simulation results of triggered RS flip-flops

In Modelica one can use the FallingEdge block to replicate the Simulink model (fig. 5). At a negative edge of its input signal it creates an “infinitesimally short” output impulse, which unfortunately doesn’t show up in result plots. That it is really working can be seen in fig. 4, where its output Qa and !Qa show the correct results – up to $t = 10.5$. At this point the CLK triggers while the inputs have the “forbidden” combination. Both outputs should be false now, but Qa stays at true mysteriously.

Since the reason, why the simple static RS flip-flop works, was puzzling anyhow, one could come up with the idea of replacing it by the elaborated version B, which even leads to a closer correspondence with Simulink. This produces the results shown as Qb and !Qb in fig. 4: It works up to $t = 10.5$, but then instead of the correct results (false/false) or the previous ones (true/false) it surprisingly goes to false/true. What is going on here?

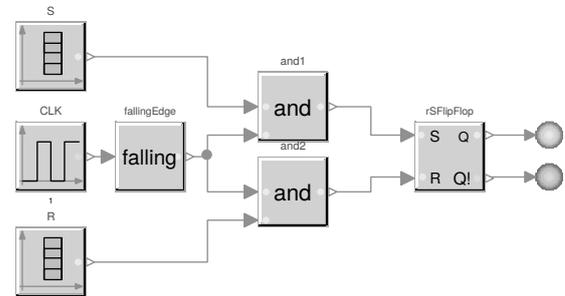


Figure 5: Triggered RS flip-flop (Modelica)

The point is not that it is particularly important to model the RS flip-flop at the forbidden state – in fact in reality one ensures that this state can not appear, e.g. with additional logic as in the JK flip-flop. The real problem is that one would (naively) expect both Modelica models to work, but they don’t. However, one has no chance to understand these results looking at the models only from the block level.

2.3 The shift register

The basic memory block is the D flip-flop, which has only one data input D and a CLK input and stores the data value at the negative edge of its CLK input. It can be created easily from a triggered RS flip-flop by identifying S with D and connecting its R input via a NOT gate to D (fig. 6). Since the “forbidden” state is ruled out here, one can hope that the Modelica version will work as expected.

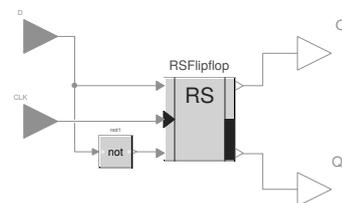


Figure 6: D flip-flop

A simple shift register (fig. 7) will be used to test the D flip-flop. The Simulink model shows the correct behaviour: Each flip-flop delays its input signal by one clock period, so that the incoming signal is “shifted” through the register.

The corresponding Modelica model however doesn’t

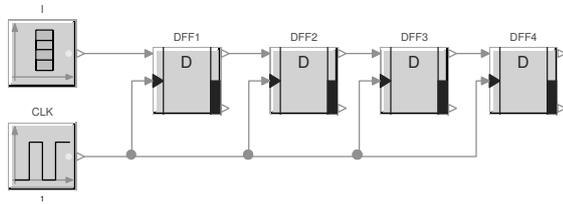


Figure 7: Shift register

work as requested (fig. 8): While the output of the first flip-flop DFF1 is actually the delayed input, the later flip-flops seem to “absorb” a single signal, so that only longer blocks survive.

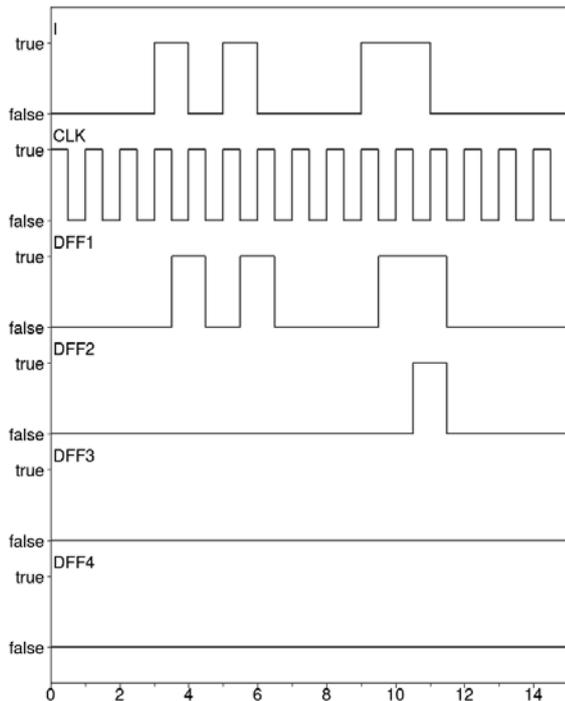


Figure 8: Simulation results of a shift register (Modelica)

Using version B of the static RS flip-flop inside the model, one again gets different, but still wrong results: Though the output of DFF1 is still correctly delayed, all other outputs coincide with DFF1 – the signal runs through the register in one step! The same thing happens incidentally, if one uses the D flip-flop that is hidden in the Modelica library `Electrical.Digital.Examples`.

In contrast to the triggered RS flip-flop, here one can find an easy ad-hoc workaround: If one adds a Pre

block before each D input (optionally including the first one, for symmetry) the shift register works perfectly.

3 Detailed analysis of the models

The Modelica results shown above seem to be proper nonsense and the result of grave bugs of the simulation program. But quite the contrary: They are completely in accordance with the Modelica language specification [6], as will be shown in the following.

3.1 A look at the Modelica language specification

The handling of events in Modelica is quite complicated, the relevant descriptions are scattered throughout the standard document [6]. The basic definition of an *event* is given in ch. 8.5 [6, p.92], but the most relevant point for the current discussion is the clarification of the pre operator:

`pre(y)`
 Returns the “left limit” $y(t^{pre})$ of variable $y(t)$ at a time instant t . At an event instant, $y(t^{pre})$ is the value of y after the last event iteration at time instant t (...).
 ...
 A new event is triggered if at least for one variable v “`pre(v) <> v`” after the active model equations are evaluated at an event instant. In this case the model is at once reevaluated. This evaluation sequence is called “*event iteration*”. The integration is restarted, if for all v used in pre-operators the following condition holds:
 “`pre(v) == v`”. [6, p.29f]

The exact procedure is described with more detail in an appendix [6, p.262f]. It has two important consequences:

1. Discrete variables can run through several values at a fixed time instant, if the event loop consists of more than one iteration.

2. $\text{pre}(v)$ is the value of v at the preceeding iteration, not the one before the first iteration (i. e. at t^-).

This is in contrast to the much simpler Simulink definition, where the *previous* value is related to time, i.e. it is $y(t^-)$.

3.2 Understanding the static RS flip-flop

To thoroughly understand the behaviour of a discrete model in Modelica, one has to go through the event iterations manually and compute the values of all variables. For version A of the static RS flip-flop (fig. 1) these are the two inputs S, R , the two outputs $Q, !Q$ and the input P of the Pre block (shown as Memory block in the figure).

To understand, why the model has no time delay, it suffices to look at the values around $t = 3$. To compute them one starts with the known inputs S, R and the value of $!Q$, which is given as the P of the “last” step. This immediately gives Q and finally P . Table 1 displays all values, where $t = 3^-$ is before the event, $t = 3.x$ during the event loop and $t = 3^+$ at the end.

t	S	R	P	Q	!Q
3^-	0	0	1	0	1
$3.a$	1	0	0	0	1
$3.b$	1	0	0	1	0
3^+	1	0	0	1	0

Table 1: Event iteration for the static RS flip-flop

The first line is given by the previous state of the flip-flop, the next line reflects the change of the input S . Comparing the two lines shows that P has changed, i.e. $P \neq \text{pre}(P)$. This leads to a new event iteration with the result shown in the third line. The change in Q and $!Q$ triggers another iteration giving the result in the last line. This time nothing has changed, the iteration stops and the simulation time continuous with these values.

This analysis clarifies the behaviour of the static flip-flop: It works without output delays because of the immediate re-evaluation after a change. In a similar way one can make sure that version B works as well.

3.3 Understanding the triggered RS flip-flop

For a detailed walk-through one starts with a labelling of all signals, including those which are hidden inside subsystems. The result for the triggered RS flip-flop (version A) is shown in fig. 9.

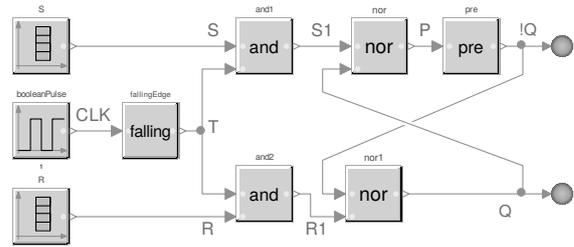


Figure 9: Detailed view of the triggered RS flip-flop

The interesting point here is the wrong result for $R = S = 1$, which happens at the negative edge of the CLK input at $t = 10.5$ (cf. fig. 4). The computation starts with the known inputs S, R, CLK and the value of $!Q$, which is the last value of P . Using them one immediately gets $T, S1, R1$ and Q and finally P . The result is shown in table 2.

t	S	R	CLK	T	S1	R1	P	Q	!Q
10.5^-	1	1	1	0	0	0	0	1	0
$10.5.a$	1	1	0	1	1	1	0	0	0
$10.5.b$	1	1	0	0	0	0	0	1	0
10.5^+	1	1	0	0	0	0	0	1	0

Table 2: Event iteration for the triggered RS flip-flop

Before $t = 10.5$, the flip-flop is set, i.e. $Q = 1$ and $P = 0$, which fixes the first line. The next line shows the change of the CLK input to 0, which sets $T = 1$ and all other signals accordingly. The following iteration has $T = 0$, cutting of S and R again and leading to another iteration. Finally nothing changes, so the event loop stops and the simulation time progresses again.

At $10.5a$ the “correct” output $!Q = Q = 0$ appears, but before it gets stable the T signal is already 0 again. The “falling edge” detection doesn’t survive the event iteration, its signal is too short to work as expected. To understand why the flip-flop works properly for other times, table 3 provides the computations for $t = 7.5$.

One easily assures oneself of the correct working of the flip-flop at all other times and of the – strange, but correct – behaviour of version B.

t	S	R	CLK	T	S1	R1	P	Q	!Q
7.5 ⁻	0	1	1	0	0	0	0	1	0
7.5.a	0	1	0	1	0	1	1	0	0
7.5.b	0	1	0	0	0	0	1	0	1
7.5 ⁺	0	1	0	0	0	0	1	0	1

Table 3: Event iteration for the triggered RS flip-flop, 2nd example

3.4 Explaining the shift register

To understand the strange behaviour of the shift register it suffices to examine a short version with two D flip-flops in a row. Fig. 10 shows the detailed model and all signal names.

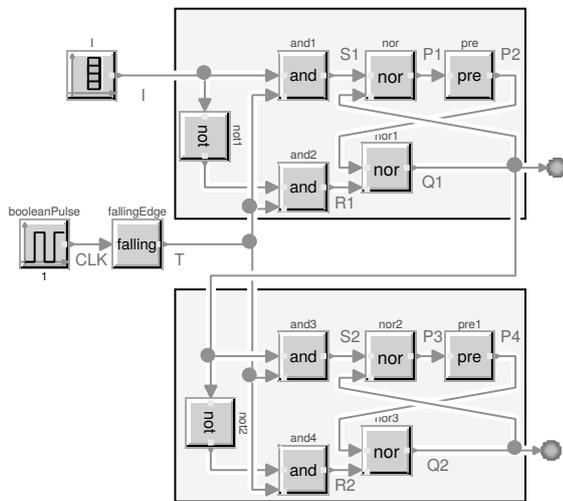


Figure 10: Detailed view of a short shift register

As should be clear by now, one starts with I, CLK, P2 (last value of P1) and P4 (last value of P3), computes T, S1, R1, S2, R2, Q1, Q2 and finally P1, P3. Using the input values of fig. 8 leads to the results shown in table 4.

It is now obvious, why the shift register doesn't work as intended: At event iteration 4.5.a the first flip-flop is already reset, i. e. Q1 has the value 0. But also at this

t	I	CLK	T	S1	R1	P2	Q1	P1	S2	R2	P4	Q2	P3
3.5 ⁻	1	1	0	0	0	1	0	1	0	0	1	0	1
3.5.a	1	0	1	1	0	1	0	0	0	1	1	0	1
3.5.b	1	0	0	0	0	0	1	0	0	0	1	0	1
3.5 ⁺	1	0	0	0	0	0	1	0	0	0	1	0	1
4.5 ⁻	0	1	0	0	0	0	1	0	0	0	1	0	1
4.5.a	0	0	1	0	1	0	0	1	0	1	1	0	1
4.5.b	0	0	0	0	0	1	0	1	0	0	1	0	1
4.5 ⁺	0	0	0	0	0	1	0	1	0	0	1	0	1

Table 4: Event iteration for the shift register

point the flip-flops are unlocked, i. e. T is 1. Therefore the second flip-flop gets the new value of Q1, not the old one. This makes plausible that an additional Pre block might cure the problem. To make sure though, one has to go through another manual computation.

4 Solution of the problems

The last section has shown, how to explain the strange behaviour of the flip-flop example models in the context of Modelica's event system. But this doesn't help with the task of creating reliable, easy to use flip-flop components, because there is apparently no way to implement a triggered system with the semantics that is needed here – at least using Modelica's standard events.

Similar problems have been found in a completely different application area, namely the modeling of reliable state machines [3]. They have been solved there by extending the Modelica language itself: Version 3.3 contains features for synchronous signals that are tied to discrete clocks [7]. A corresponding block library has been created that allows to include these features easily in a graphical environment [8].

Fortunately the new possibilities are exactly what is needed to create a properly working triggered flip-flop. Fig. 11 shows an implementation based on the new Synchronous library. The Sample and Hold blocks transfer standard signals to synchronous ones and vice versa. The EventClock creates a clock signal that is triggered by a positive edge of its input.

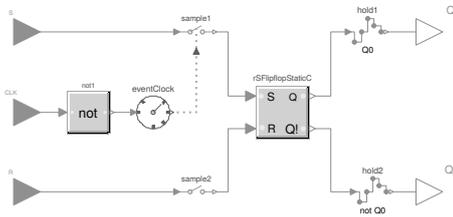


Figure 11: Triggered RS flip-flop with clock

But the main point is hidden inside the static RS flip-flop (fig. 12): Though it looks almost exactly like the version B of the static flip-flop (fig. 3), instead of a Pre block it now contains the new UnitDelay block. It returns the value of its clocked input signal at the previous time step, not at some mysterious event loop iteration.

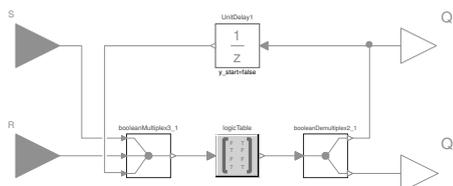


Figure 12: Static RS flip-flop with clock

Since the semantics of the blocks and signals are now identical to their Simulink versions, so are the results: The flip-flop works as expected, even at the “forbidden” state, and the shift register built from it properly shifts its input.

5 Conclusions

Modelica’s event system is complicated – to say the least. This can lead to models that behave very differently to what one might expect, especially if one has worked with Simulink before. The main culprit is the pre operator – or corresponding block – that is not defined in a strictly temporal sense, but relates to a previous iteration of the event loop.

The good news is that the Synchronous library and its underlying Modelica constructs add just the kind of temporal semantics that one needs e. g. for the modeling of flip-flops or state machines. This could be used to complement the Digital library with still missing

standard flip-flop components.

But the crucial point to bear in mind is: Even with today’s sophisticated graphical simulation tools modeling is more than connecting blocks from a library! Without a thorough understanding of the underlying mechanisms one is on murky ground, and simulation results may be correct just by pure luck.

Acknowledgements

The author is thankful to Fabian Köslin from Dassault Systemes for helpful discussions about the intricacies of the Modelica event system.

The author is grateful for the hospitality extended to him by Tom Schramm and his colleagues at the Department of Geomatics, HCU Hamburg.

References

- [1] P. A. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3*. Wiley & Sons, New York, 2015.
- [2] The MathWorks. *Stateflow: Model and simulate decision logic using state machines and flow charts*. Online: <http://www.mathworks.de/products/stateflow/> (called 2015-11-25).
- [3] H. Elmqvist, F. Gaucher, S. E. Mattsson, F. Dupont. *State Machines in Modelica*. Proc. 9th Int. Modelica Conference, Munich, p. 37-46, 2012.
- [4] The MathWorks. *SimEvents: Model and simulate discrete-event systems*. Online: <http://www.mathworks.de/products/simevents/> (called 2015-11-25).
- [5] V. Sanz, A. Urquia, S. Dormido. *Parallel DEVS and Process-Oriented Modeling in Modelica*. Proc. 7th Int. Modelica Conference, Como, p. 96–107, 2009.
- [6] Modelica Association. *The Modelica Language Specification Version 3.3 Revision 1, July 11, 2014*. Online: <https://modelica.org/documents/>

ModelicaSpec33Revision1.pdf (called 2015-11-27).

- [7] H. Elmqvist, M. Otter, S. E. Mattsson. *Fundamentals of Synchronous Control in Modelica*. Proc. 9th Int. Modelica Conference, Munich, p. 15-26, 2012.
- [8] M. Otter, B. Thiele, H. Elmqvist. *A Library for Synchronous Control Systems in Modelica*. Proc. 9th Int. Modelica Conference, Munich, p. 27-36, 2012.

Mathematisches Framework für Simulationsmodelle von State Event Modellen

Andreas Körner¹

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

Der Beitrag befasst sich mit der Problematik rund um State Event Modelling. Ein Grundproblem hierbei stellt die mathematische Formulierung der Modelbeschreibung dar. Für gewisse Spezialfälle wie beispielsweise Variable Structure Systems (VSS), gibt es entsprechende Beschreibungen in den Anwendungen, doch fehlt hier eine Struktur um es auf andere Typen von State Event Modelle abstrahieren zu können. Ein Grund hierfür ist, dass die Modellbeschreibung oft mit der die Simulation betreffenden Numerik vermennt wird. Das Auffinden der State Events, welches entweder durch mathematische Gleichungen formuliert wird, oder durch einen entsprechenden Algorithmus abgebildet wird, wird als Modellbeschreibung aufgefasst. Eine klare mathematische Beschreibung fehlt hier oft.

In diesem Artikel werden diese beiden Ebenen der mathematischen Beschreibung und der Numerik betreffenden Simulation getrennt betrachtet. Dies ermöglicht, ein mathematisches Modell und die an den entsprechenden Simulator gekoppelte Formulierung eines sogenannten Simulationsmodells als Implementierung in einer Simulationsumgebung zu unterscheiden. Das mathematische Modell bietet die Möglichkeit eines abstrakten Vergleiches diverser State Event Modelle unabhängig von Einschränkungen, welche in den Simulationsmodellen berücksichtigt werden müssen. Die beiden Modellbeschreibungen, das mathematische Modell und das Simulationsmodell werden in Beziehung zueinander gesetzt und unterschiedliche Vergleichsmöglichkeiten werden erörtert um Möglichkeiten von Benchmarks und Comparisons korrekt definieren zu können.

1 Einleitung

State Event Modelling als Modellierungstechnik ist in den letzten Jahren und Jahrzehnten immer bedeutsamer geworden. Der Ansatz lockt durch einen simples Optimalitätsargument, dass die unterschiedlichen Zustände eines Systems getrennt betrachtet werden und die Übergänge von einem Zustand in den anderen zu definieren. Implizit wird erwartet, dass in einem Zustand die entsprechende Modellbeschreibung hinsichtlich eines Attributs optimal gewählt wurde. Das kann sich manifestieren, dass lediglich eine Koordinatentransformation notwendig ist, oder aber auch dass einmal eine gewöhnliche Differentialgleichung als Beschreibung verwendet wird, einmal aber eine Differential-Algebraische Gleichung notwendig ist.

2 Framework für State Event Modelle

Damit ergibt sich eine strukturelle Beschreibung, die sich der Bezeichnungen Zustände (States) und der zugehörigen Übergänge (Transitions) bedient. Als In jedem Zustand wird die Modellbeschreibung herangezogen um in einem Simulationsenvironment gerechnet zu werden und ergibt damit eine entsprechenden zeitlichen Verlauf der Lösungen, welche hintereinander angeordnet den zeitlichen Verlauf der Trajektorie des gesamten Systems ergibt. Dabei gibt es Zeitpunkte, wo eine Modellbeschreibung durch einen Transition in eine andere Modellbeschreibung übergeführt wird, dies wird als Event bezeichnet. Es werden also Modellbeschreibungen durchlaufen, welche nicht als zeitliche Abfolge den Simulationslauf abbilden. Abbildung 1 veranschaulicht diesen Zusammenhang mit Hilfe einer Auswahl von 3 Modellen.

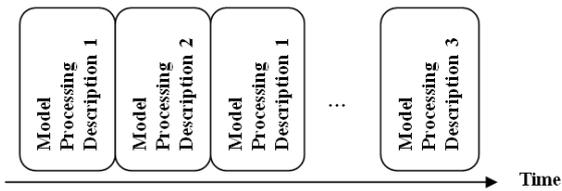


Abbildung 1. Zeitliche Durchlaufung von Teilmodellbeschreibungen eines State Event Modells.

Wichtig ist hierbei die Möglichkeit der Durchlaufung einzelner Teilbeschreibungen, da diese nicht abgearbeitet sind, wenn diese einmal durchlaufen wurden, diese können mit den entsprechenden Transitions weitere male durchlaufen werden. Diese Eigenschaft einer Beschreibung ist auf der topologischen Ebene zu finden, diese muss in einer prozeduralen Darstellung erfasst werden. Eine mögliche Repräsentation ist jene, welche die Zustände und Übergänge in einem Graphen, wie es für das in Abbildung 1 angeführte Beispiel in Abbildung 2 dargestellt ist.

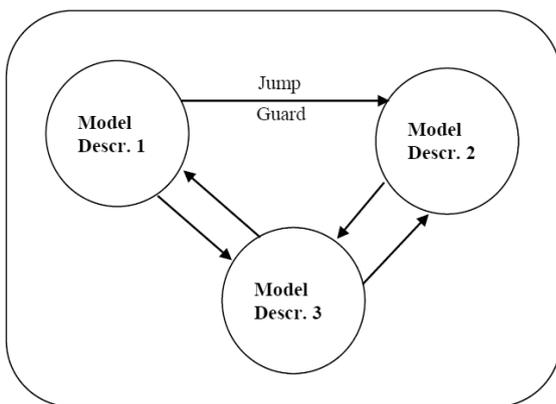


Abbildung 2. Strukturelle Darstellung von Teilmodellbeschreibungen eines State Event Modells.

Grundlegendes Problem in dieser Darstellung ist jedoch, dass diese Beschreibung lediglich eine mathematische Beschreibung ist und keinen Simulationsaspekt beinhaltet. Bei Modellen, welche die Simulationsumgebung berücksichtigen – sogenannten Simulationsmodellen – ist jedoch das Problem, dass oftmals eine derart reine mathematische Beschreibung nicht vorliegt. Es muss also eine Beschreibung gefunden werden, welche sowohl das mathematische Modell, als auch die Aspekte einer Simulationsumgebung beinhaltet. Diese strukturelle Trennung von unterschiedlichen Beschreibungsebenen ermöglicht eine saubere Unterscheidung zwischen Modell und Simulationsmodell, wie in Abbildung 3 illustriert.

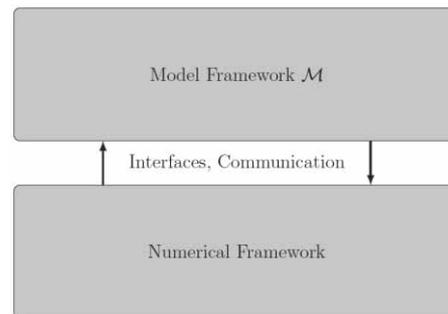


Abbildung 3. Strukturelle Trennung von Mathematischer Beschreibung und numerischen Betrachtungen der Simulationsumgebung.

Die Komponente, welche für die mathematische Beschreibung zuständig ist, wird als Model Framework bezeichnet und ist für die abstrakte Beschreibung zuständig. Das Numerical Framework beschreibt die gesamten numerischen und simulativen Attribute.

3 Zusammenfassung und Ausblick

Dieser Beitrag ist als Work in Progress Beitrag zu verstehen, wo die bisherigen Überlegungen ausgeführt werden. Es kann als Fortsetzung des Beitrages [1] sein und eine Formalisierung der Gedanken und Beispiele aus [2]. Maßgeblich ist dieser Framework Begriff für Vergleiche von Modellierungsansätzen und Simulationsumgebungen. Hiermit kann eine Fähigkeit bewertet, zugeordnet und eindeutig verglichen werden. Weitere Arbeitsaufgaben werden im Bereich der mathematischen Formalisierung und Untersuchung des Mathematical Frameworks liegen.

4 References

- [1] A. Körner und F. Breiteneker. *Approaches for State Event Handling by Simulation Algorithm and via Model Description*. In: Proceedings ASIM 2014 22 Symposium Simulationstechnik, Wittmann J., Deatcu C. (Hrsg.); ARGESIM/ASIM, 43/2/Vienna, ISBN: 978-3-901608-44-5; ARGESIM/ASIM, Wien, 2014
- [2] A. Körner, H. Ecker und F. Breiteneker. *Anforderungen an State Event Charakterisierung für die Simulation Hybrider Modelle*. In: ASIM-Workshop der STS/GMMS 2014", J. Scheible, I. Bausch-Gall, C. Deatcu (Hrsg.); ARGESIM, Wien

Entwurf und Betrieb aufgabenorientierter Multi-Robotersteuerungen mit MATLAB/Stateflow

Birger Freymann, Sven Pawletta, Artur Schmidt, Thorsten Pawletta

Hochschule Wismar - University of Applied Sciences - Research Group CEA, Wismar, Germany

{birger.freymann, sven.pawletta, artur.schmidt, thorsten.pawletta}@hs-wismar.de

In der industriellen Produktion gewinnen Multirobotersysteme (MRS) zunehmend an Bedeutung. Ein MRS ist ein Roboterteam bestehend aus mehreren Industrierobotern mit einem zentralen Steuerungsrechner. Hierbei können die einzelnen Industrieroboter von verschiedenen Herstellern und unterschiedlichen Typs sein. Wie schon der Begriff Team impliziert, vereinen die einzelnen Roboter ihre Fähigkeiten, um Aufträge effizienter zu lösen, als es durch Einzelrobotersysteme möglich ist. Ein solches Roboterteam weist gegenüber einzelnen Robotern mit unabhängigen, lokalen Steuerungsrechnern verschiedene Vorteile auf. Beispielsweise können Sensorinformationen untereinander ausgetauscht und zur Erstellung eines gemeinsamen Umweltmodells verwendet werden. Weiterhin können die Roboter miteinander in gegenseitige Wechselwirkung, sogenannte Interaktion, treten. Dadurch kann eine gezielte Zusammenarbeit der Teammitglieder erfolgen und die Performanz des gesamten Systems erhöht werden.

Die Entwicklung von Steuerungen für MRS erfolgt gegenwärtig mit proprietären Entwicklungsumgebungen der einzelnen Roboterhersteller. Dadurch ist es schwierig, MRS mit Robotern unterschiedlicher Hersteller zu realisieren. Im Gegensatz dazu bietet der vorgestellte Ansatz die Möglichkeit zum Aufbau von Teams aus Robotern verschiedener Hersteller.

Eine Herausforderung bei der Steuerung von MRS sind die Interaktionen zwischen den Industrierobotern. Zur Lösung dieses Problems werden aufbauend auf einer Klassifikation nach Lüth [1] unterschiedliche Interaktionsklassen eingeführt. Weiterhin wird von einer modellbasierten und aufgabenorientierten Steuerungsentwicklung (task-oriented control (TOC)) ausgegangen. Die TOC basiert auf der Spezifikation von einzelnen, wiederverwendbaren Aufgaben, wie zum Beispiel *GreifeTeil* oder *TransportiereTeil*. Steuerungsaufträge werden durch Komposition von Aufgaben in Form von Sequenzen, Iterationen oder Verzweigungen abgebildet. Eine Aufgabe ist eine implizite Beschreibung und muss in explizite Steuerungsbefehle umgesetzt werden. Aufbauend auf der TOC wird gezeigt, wie die eingeführten Interaktionsklassen als Aufgaben abgebildet werden können. Neben der Aufgabenorientierung soll die Steuerungsentwicklung modellbasiert gemäß dem Ansatz des Rapid Control Prototypings (RCP) erfolgen. Dazu wird auf dem Simulation Based Control (SBC) Ansatz aufgebaut. Der SBC definiert ein Vorgehensmodell gemäß dem RCP und ein Framework zur Implementierung. Das Framework unterstützt die durchgängige Weiterentwicklung eines Steuerungsmodells von der Entwurfsphase bis zum operativen Steuerungsbetrieb. Es basiert auf einem Schichtenmodell und einer Modellbibliothek. Die Aufgaben sind als wiederverwendbare Komponenten in der Modellbibliothek organisiert.

Im Vortrag wird anhand eines Anwendungsbeispiels die prinzipielle prototypische Umsetzung des Konzepts mit MATLAB/Stateflow aufgezeigt. Die Steuerungsentwicklung erfolgt robotertyp- und herstellerunabhängig. Die Prozessanbindung, erfolgt mittels der Robotic Control and Visualization (RCV) Toolbox for MATLAB von der Research Group CEA.

[1] Lüth, Tim C. *Technische Multi-Agentensysteme: verteilte autonome Roboter- und Fertigungssysteme*, Carl Hanser Verlag, München Wien, 1998

Trajectory optimization of a hypersonic plane s.t. heat limits

Kurt Chudej¹, Armin Rund², Markus Wächter³

¹Lehrstuhl für Ingenieurmathematik, Universität Bayreuth

²Institut für Mathematik und Wissenschaftliches Rechnen, Karl-Franzens-Universität Graz

³German Institute of Science and Technology, TUM Asia Pte Ltd, Singapore

kurt.chudej@uni-bayreuth.de

Extreme thermal loads appear during ascent and reentry of future hypersonic planes. Besides of the availability of a hypersonic propulsion system also an efficient lightweight thermal protection system is mandatory for the feasibility of the entire mission. We consider a trajectory optimization problem enhanced with a PDE modelling the heat load at the stagnation point of the plane. The thermal protection system heats up depending on the flight path. A maximal thermal load is prescribed, which is mathematically modeled as state constraints. These state constraints couple the PDE to the ODE resulting in a fully coupled system. The thermal protection system is discretized in space into several layers using a finite volume scheme. Considering the problem after semidiscretization as a classical ODE constrained optimal control problem, the orders of the state constraints of the different layers are investigated. It turns out that the order is rising with the number of layers starting from two (for the outer layer). The influence of the state constraints is analyzed in numerical experiments. The numerical results are obtained by a direct solution method.

1 Introduction

The design of new hypersonic airplanes is a difficult task. Flying in the hypersonic flight regime yields considerable aerothermic loads. Therefore innovative lightweight materials and a sophisticated thermal protection system (TPS) are mandatory for the airplane. Budget and weight restrictions for the new hypersonic airplane are directly connected with the design of the TPS. One solution approach is to monitor the aerothermic loads during flights. A further step is to find an optimal trajectory which stays below certain thresholds on the different layers of the TPS. This allows to use a not so demanding TPS. Intensive research in designing hypersonic transportation systems was conducted in the 1990s, see e.g. [1–10], and is currently again intensified.

We consider a specific version of the above described trajectory optimization problem. It is an optimal control problem with ordinary (ODE) and partial differential equations (PDE). In particular, the flight path optimization problem is enhanced with a PDE describing the thermal load at the stagnation point, see e.g.

[11, 12, 13]. The problem is fully coupled. On the one hand the position and velocity of the ODE-constraint (equations of motion) enter the heat equation of the thermal load. On the other hand a state constraint on the temperature in the TPS establishes an influence of the PDE on the optimal flight path. Therefore, we have to solve the full optimal control problem s.t. to the ODE, the PDE and the state constraint.

For the numerical solution at first a finite volume approach is used in order to transform the PDE into a system of ODEs. The resulting ODE-constrained optimal control problem s.t. a state constraint is then discretized into a (finite dimensional) nonlinear optimization problem, see e.g. [14] Chap. 4.5.

The aim of the paper is to describe in a detailed mathematical way the coupling between a complicated but standard trajectory optimization problem with an instationary heat equation together with state constraints. We will enhance the previous mathematical description of [13]. The new contribution of this paper lies especially in the detailed analysis of the order of the state constraints of the different layers of the TPS.

Moreover in the meantime the theory and numerics of optimal control s.t. PDE constraints has matured, see e.g. the books [15, 16, 17]. Nevertheless today only a few results for coupled ODE-PDE constrained optimal control problems are available. A simplified version of the hypersonic airplane – a rocket car – was analyzed in [18, 19, 20, 21]. For the rocket car detailed theoretical (and numerical) investigations are available, in particular about the regularity of the Lagrange multipliers and about the nonlocal effects of the state constraint via the heat equation. We compare these effects with those observed in the numerical solution of the much more challenging model of the hypersonic plane.

The article is organized as follows. In Section 2 the trajectory optimization problem is introduced successively. Emphasis is laid on the inequality constraints, the PDE and its discretization. In Section 3 the state constraints are analyzed and a link to DAE-constrained optimal control problems is established. Section 4 reports on a numerical test case of a minimal fuel flight over a given range. The influence of the heating constraint on the optimal flight path is investigated in a comparative study.

2 Optimal control problem

Optimizing the hypersonic flight while considering the thermal protection system gives rise to a complex optimal control problem. Therefore, we introduce the problem step by step. The model of the hypersonic space vehicle together with the TPS is based on [12], see also [11, 13].

We consider a minimum fuel — resp. maximum final mass — range flight over a spherical rotating Earth in the equator plane with free final time t_f . The trajectory optimization problem is modeled as an optimal control problem using five state variables, the velocity V [m/s], the flight path angle γ [–], the altitude H [m], the range ζ [m], and the total mass m [kg]. Control variables are the angle of attack α [–] and the equivalence ratio Φ_L [–] (needed for the computation of the fuel consumption). Based on wellknown equations of motion for flight mechanics, see e.g. [22], the optimal control problem is given as

$$\max_{\alpha(t), \Phi_L(t), t_f} m(t_f) \quad (1)$$

s.t.

$$\begin{aligned} \dot{V}(t) &= f_1(V, \gamma, H, m; \alpha, \Phi_L) \\ &= [F(V, H; \alpha, \Phi_L) \cos(\alpha) - D(V, H; \alpha)]/m \\ &\quad + \sin \gamma \left(\omega_E^2 (r_E + H) - g(H) \right), \end{aligned} \quad (2)$$

$$\begin{aligned} \dot{\gamma}(t) &= f_2(V, \gamma, H, m; \alpha, \Phi_L) \\ &= [F(V, H; \alpha, \Phi_L) \sin(\alpha) + L(V, H; \alpha)]/(mV) \\ &\quad + \cos \gamma \left(\frac{V}{r_E + H} + \frac{\omega_E^2 (r_E + H) - g(H)}{V} \right) + 2 \omega_E, \end{aligned} \quad (3)$$

$$\dot{H}(t) = V \sin \gamma, \quad \dot{\zeta}(t) = V \cos \gamma, \quad (4)$$

$$\dot{m}(t) = -\dot{m}_F(V, H; \alpha, \Phi_L), \quad (5)$$

with the gravitational acceleration $g(H) = g_0 r_E^2 / (r_E + H)^2$ [m/s²], the gravitational constant g_0 [m/s²], the radius of the Earth r_E [m] and the angular velocity of the Earth ω_E [rad/s]. The expressions for the thrust $F(V, H; \alpha, \Phi_L)$ [N], the drag $D(V, H; \alpha)$ [N], the lift $L(V, H; \alpha)$ [N] and the fuel consumption $\dot{m}_F(V, H; \alpha, \Phi_L)$ [kg/s] are given in [12, 13].

The control and state variables have to fulfill inequalities motivated by technical reasons and model validity. These inequalities consist of state constraints

$$V > 0 \text{ [m/s]}, \quad -\pi \text{ [-]} \leq \gamma \leq \pi \text{ [-]}, \quad H \geq 500 \text{ [m]}, \quad (6)$$

$$10\,000 \text{ [Pa]} \leq q(V, H) \leq 50\,000 \text{ [Pa]}, \quad (7)$$

with the dynamic pressure $q(V, H)$ and mixed constraints

$$0 \text{ [-]} \leq n(V, H, m; \alpha) \leq 2 \text{ [-]}, \quad (8)$$

with the load factor $n = L(V, H; \alpha) / (mg_0)$. Further, there are technical restrictions to the controls

$$\begin{aligned} -1.5\pi/180 \text{ [-]} \leq \alpha \leq 20\pi/180 \text{ [-]}, \\ \Phi_{L, \min}(V, H) \leq \Phi_L \leq \Phi_{L, \max}(V, H). \end{aligned} \quad (9)$$

To find an adequate trajectory boundary conditions are added, especially for the range ζ . The numerical example below is a minimal fuel flight over a given range. Therefore the initial and boundary conditions are set to

$$\begin{aligned} V(0) &= 150 \text{ [m/s]}, \quad V(t_f) = 150 \text{ [m/s]}, \\ \gamma(0) &= 0 \text{ [-]}, \quad \gamma(t_f) = 0 \text{ [-]}, \\ H(0) &= 500 \text{ [m]}, \quad H(t_f) = 500 \text{ [m]}, \\ \zeta(0) &= 0 \text{ [m]}, \quad \zeta(t_f) = 9 \cdot 10^6 \text{ [m]}, \\ m(0) &= 244\,000 \text{ [kg]}. \end{aligned} \quad (10)$$

Up to now, the trajectory optimization problem constitutes a classical ODE control problem with control and state constraints. In the following, the thermal

protection system is accounted for. The heating is modelled using a nonlinear heat equation. Constraining the maximal temperature finally couples the PDE to the ODE, resulting in a fully coupled ODE-PDE control problem.

In hypersonic flight the heating of the plane is heavily influenced by two layers around the plane, the bow shock, and the boundary layer around the plane, see Fig. 1 and 2. While the atmospheric temperature T_∞ can be directly obtained from the altitude H of the vehicle, the temperature after the bow shock T_1 and the temperature in the boundary layer T_c are solutions to a system of nonlinear equations. Therefore it is necessary to compute all atmospheric variables after the bow shock and inside the boundary layer.

Hypersonic flight causes a substantial heating of the aircraft wall, therefore a sophisticated thermal protection system is needed. The thermal protection system consists of several insulated layers of suitable materials and thicknesses. For simplicity we consider here only the most critical stagnation point. Further thermal models for other important parts of the surface, e.g. near the fuel tank, are described e.g. in [11, 12]. At the stagnation point a $d_{\text{TPS}} = 4 \cdot 10^{-3}$ [m] thick layer of carbon fiber reinforced silicon carbide material C/SiC is assumed. The density of the material is $\rho_{\text{TPS}} = 2,100$ [kg/m³]. The heat conductivity λ_{TPS} and heat capacity $c_{p,\text{TPS}}$ of the material are modeled by temperature dependent polynomials of degree three resp. one.

The temperature $\Theta(t, \vec{x})$ in the thermal protection system is dependent on time t and position $\vec{x} \in \Omega$. Ω

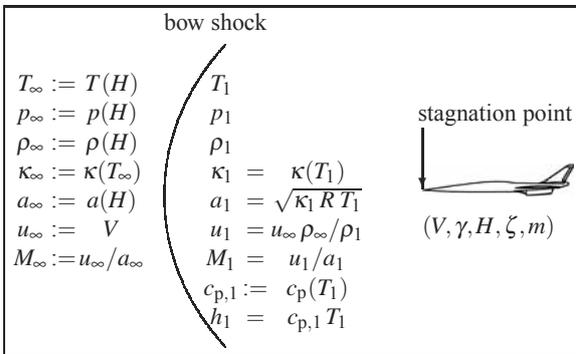


Figure 1: Bow shock in front of airplane (schematic diagram).

represents the spatial neighborhood of the stagnation point. The temperature distribution is modelled with the instationary nonlinear heat equation

$$\rho_{\text{TPS}} c_{p,\text{TPS}}(\Theta) \frac{\partial \Theta}{\partial t} = \text{div}[\lambda_{\text{TPS}}(\Theta) \text{grad} \Theta], \quad (11)$$

see [13]. We consider in the following for simplicity only a 1D model at the stagnation point. The 3D coordinate \vec{x} is therefore substituted by the 1D spatial axis x (interior normal of the wall), i.e. $x \in [0, d_{\text{TPS}}]$, see Fig. 2. We complete the parabolic PDE with initial and boundary conditions:

$$\begin{aligned} \Theta(0, x) &= 300 \text{ [K]}, \quad x \in [0, d_{\text{TPS}}], \\ \frac{\partial \Theta}{\partial x}(t, 0) &= \dot{q}_{\text{aero}}(V(t), H(t), \Theta(t, 0)) \\ &\quad - \varepsilon \sigma (\Theta(t, 0)^4 - T_\infty(H)^4), \quad t > 0, \\ \frac{\partial \Theta}{\partial x}(t, d_{\text{TPS}}) &= \alpha_q (\Theta(t, d_{\text{TPS}}) - T_{\text{in}}) \\ &\quad + \varepsilon \sigma (\Theta(t, d_{\text{TPS}})^4 - T_{\text{in}}^4), \quad t > 0. \end{aligned} \quad (12)$$

Therein, ε [-] resp. α_q [J/(m² · s · K)] denote the emissivity resp. the heat transfer coefficient between the thermal protection system and the air. Both boundary conditions consider heat conduction and radiation. The heat flux $\dot{q}_{\text{aero}}(V, H, \Theta_1)$ on the outer boundary depends on the atmospheric variables in the boundary layer. The formulas for the heat flux $\dot{q}_{\text{aero}}(V, H, \Theta_1)$ can be found in [12, 13].

With the heat equation we monitor the heat distribution in the stagnation point. Additional state constraints $\Theta(t, x) \leq \Theta_{\text{max}}(x)$ are imposed. They are subject to material constraints (or budget constraints and might even allow to use less expensive material).

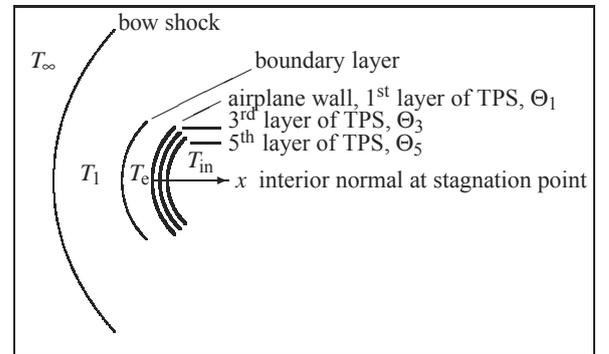


Figure 2: Zoom of Fig. 1, schematic diagram at stagnation point after finite volume discretization.

In the following we apply a semidiscretization. This allows to transform the ODE-PDE optimal control problem to a pure ODE constrained control problem. Accordingly, we can analyze the state constraint using the concept of an order of a state constraint, which is well-established in optimal control s.t. ODEs.

Therefore, we apply a standard finite volume discretization to the different layers of the thermal protection system. In particular, we consider N layers resulting in

$$\begin{aligned}\dot{\Theta}_1(t) &= g_1(\Theta_1, \Theta_2, V, H) = \\ &= [\mathcal{R}_{\text{out}} - \psi_2]/[d\rho_{\text{TPSCp,TPS}}(\Theta_1)],\end{aligned}\quad (13)$$

$$\begin{aligned}\dot{\Theta}_i(t) &= g_i(\Theta_{i-1}, \Theta_i, \Theta_{i+1}) = \\ &= [\psi_i - \psi_{i+1}]/[d\rho_{\text{TPSCp,TPS}}(\Theta_i)],\end{aligned}\quad i = 2, 3, N-1, \quad (14)$$

$$\begin{aligned}\dot{\Theta}_N(t) &= g_N(\Theta_{N-1}, \Theta_N) = \\ &= [\psi_N - \mathcal{R}_{\text{in}}]/[d\rho_{\text{TPSCp,TPS}}(\Theta_N)],\end{aligned}\quad (15)$$

with $\psi_i = \frac{\Theta_{i-1} - \Theta_i}{d} \lambda_{\text{TPS}} \left(\frac{\Theta_{i-1} + \Theta_i}{2} \right)$ and $d = d_{\text{TPS}}/N$ and $T_{\text{in}} = 300$ [K]. The boundary conditions of the heat equation are transformed by the discretization to

$$\begin{aligned}\mathcal{R}_{\text{out}} &:= \dot{q}_{\text{aero}}(V, H, \Theta_1) - \varepsilon \sigma (\Theta_1^4 - T_{\infty}(H)^4), \\ \mathcal{R}_{\text{in}} &:= \alpha_{\text{q}} (\Theta_N - T_{\text{in}}) + \varepsilon \sigma (\Theta_N^4 - T_{\text{in}}^4).\end{aligned}\quad (16)$$

Initial conditions for the temperatures in the layers of the thermal protection system are

$$\Theta_i(0) = 300 \text{ [K]}, \quad i = 1, \dots, N. \quad (17)$$

The state constraints transform to

$$\Theta_i(t) \leq \Theta_{i,\text{max}}, \quad i = 1, \dots, N. \quad (18)$$

3 Analysis of the state constraints

In principle, PDE control problems with additional state constraints are significantly more involved, since the Lagrange multiplier of a pointwise state constraint is a measure only. Therefore, methods that are not formulated in the function space setting normally exhibit mesh dependence, i.e. their iteration count increases with finer meshes. An important aspect for adjoint-based methods is the discretization of the measure. Large errors might occur if it is not resolved properly on the temporal or spatial grid. Also direct methods, even if they are not based on a discretization of the Lagrange multiplier, are usually slower and sometimes might fail to converge. For ODE control problems these difficulties can be analyzed using the order

of a state constraint. In the following we will apply this concept for the ODE-PDE control problem after semidiscretization.

For ODE control problems with *scalar* control the order of a state constraint is defined as follows [23]: The expression of a the state constraint is successively differentiated w.r.t. the time while time derivatives of state variables are replaced using the ODE. The order is the minimal number of such differentiation steps until a control variable appears in the expression. It is well-known that the numerical difficulties increase with rising order. It can be explained by linking the problem to a system of differential-algebraic equations (DAE) [24]. In particular, if one considers the active inequalities as equalities, the state-constrained ODE control problem can be interpreted as an optimization problem with a semi-explicit DAE. Accordingly, the differential index of this DAE equals the order of the state constraint increased by one ([25] p. 21, [26]). Consequently, it is well-known that state constraints of order one can be solved accurately, while state constraints of order two lead to instabilities comparable to semi-explicit DAEs of differential index three.

For flight path optimization problems with *multidimensional* controls this has to be generalized to: "a state constraint of order p has differential index $i_d \geq p + 1$ " [27, 28]. Since the differential index is connected with the perturbation index [29] we also get a glimpse on the numerical difficulties which have to be expected.

In the following we consider the ODE-PDE optimal control problem after semidiscretization in space given by (1–10, 13–18). Next we will analyze the order of the state constraints in the context of ODE control.

Theorem: *The dynamic pressure limit (7) is a state constraint of order one. The temperature limit on Θ_1 is a state constraint of order two. The temperature limit on Θ_i is a state constraint of order $i + 1$, $i = 2, \dots, N$.*

Proof: Since \dot{q} depends explicitly on both controls α and Φ_L , the dynamic pressure inequality constraint is a state constraint of order one. On the other hand $\dot{\Theta}_1 = g_1(\Theta_1, \Theta_2, V, H)$ does not contain any control explicitly. We have to differentiate it with respect to time. Obviously $\ddot{\Theta}_1 = \frac{\partial g_1}{\partial V} \dot{V} +$

$\dots = \frac{\partial g_1}{\partial V} f_1(V, \gamma, H, m; \alpha, \Phi_L) + \phi(V, \gamma, H, m)$ depends on the controls α and Φ_L . Note that in general $\frac{\partial g_1}{\partial V} \neq 0$. Therefore the temperature inequality constraint $\Theta_1 \leq \Theta_{1,\max}$ is a state constraint of order two. Analogous reasoning using (13) proves the rest. We skip the other state constraints (6), since they are not active in the numerical studies below.

The Theorem shows that if the state constraint on the inner layers Θ_i , $i \geq 2$ are not active at all at the optimal solution, then only state constraints of order one or two play a role. This is the case if the maximum of the temperature is always attained at the outer boundary (at the place of the heat source). The numerical results below show this behavior. A proof of such a statement was obtained for a simpler setting in [20] Thm 4.12. Therefore the differential index of the semi-explicit DAE is $i_d \geq 3$, which has already severe numerical consequences.

4 Results

In the following we solve the trajectory optimization problem with $N = 5$ layers and temperature constraints $\Theta_{i,\max} = 1010$, $i = 1, \dots, N$. The main focus is on the influence of the temperature constraint on the solution. Therefore, the solution is compared to the solution for $\Theta_{i,\max} = \infty$, which will be called the "unconstrained problem" despite the fact that the other inequality constraints (6-10) are included in both cases. In the unconstrained case the heat equation decouples and can be simulated after the optimization. In contrast, an active temperature constraint leads to a fully coupled ODE-PDE control problem.

The presented numerical results are based on a revised Fortran code [13] and were computed using the direct collocation software DIRCOL [30] for the solution of the ODE constrained optimal control problem. The resulting nonlinear program was solved by the SQP-solver SNOPT [31].

The constrained solution exhibits an active temperature constraint $\Theta_1 = \Theta_{1,\max}$ on $t \in [t_{\text{on}}, t_{\text{off}}]$ with $t_{\text{on}} \doteq 1390.57$ [s], $t_{\text{off}} \doteq 4866.99$ [s]. Fig. 4 shows the optimized velocities of the constrained and unconstrained case. At the beginning both curves coincide. At around $t \approx 1000$ [s] the velocity is reduced in order to fulfill the temperature constraint later at t_{on} . From this we infer that the state constraint on the PDE has a non-

local time effect on the ODE control variables. This is typical for ODE-PDE control problems, and it is in coherence with the observations in the rocket car problem [18, 19, 20, 21]. During the active arc $[t_{\text{on}}, t_{\text{off}}]$ the velocity is not exactly constant but slightly increasing. Since the velocity in the constrained case is reduced for a long time, the total flight time t_f is increased to meet the boundary condition on the range. The optimal altitude is displayed in Fig. 4. It is slightly increasing during the active arc.

The temperature constraint causes an increase of 6 % in the total flight time, see Tab. 1. While the fuel consumption per time interval is lower due to the lower velocity, the total fuel consumption $m(0) - m(t_f)$ is higher due to the longer flight time t_f . Ascent and descent are rather similar in both cases, but descent starts about 400 [s] later in the constrained case, see Fig. 4.

The controls angle of attack α and equivalence ratio Φ_L are depicted in Fig. 4.

The difference between the unconstrained and the constrained case is mainly because of the active state constraint on the outer layer of the TPS, see Fig. 3. Depicted are the temperatures Θ_i in the i -th layer of the thermal protection system for the reference case and the constrained case. In particular, Θ_1 is active for a long arc $[t_{\text{on}}, t_{\text{off}}]$. The transition at t_{on} into $\Theta_1 = \Theta_{1,\max}$ for $t \in [t_{\text{on}}, t_{\text{off}}]$ is smooth, see the zoom in Fig. 3. For the prototype problem of the rocket car, this transition was proven to be differentiable in [21] Thm. 3.3.4. The temperatures in the other layers are nearly constant on approximately the same time interval $[t_{\text{on}}, t_{\text{off}}]$. They fastly approximate a constant value, their stationary values for $\Theta_1 = \Theta_{\max}$. Due to $T_{\text{in}} \ll \Theta_{\max}$, these stationary values drop monotonously in

optimal control problem	unconstrained (1–10,13–17)	constrained (1–10,13–18)
flight time t_f	6 961.5 [s]	7 378.78 [s]
min. total fuel consumption $m(0) - m(t_f)$	62 486.6 [kg]	62 787.7 [kg]
maximum temperature in TPS: Θ_1	1 124.0 [K]	1 010.0 [K]

Table 1: Results of optimal control

direction to the interior of the plane. Consequently, we observe that the state constraints of order three or more are not active at the optimum.

Alltogether we observe just an active arc (a boundary arc) for one layer. In the continuous description this corresponds to a line $[t_{\text{on}}, t_{\text{off}}] \times x_0$ in the space-time cylinder of the heat equation with $x_0 = 0$. In the case of the rocket car problem there is a full analysis for such active sets and the corresponding line measure, see [21]. In that case, it was proven that the solution to the adjoint PDE is just unbounded at (t_{on}, x_0) and (t_{off}, x_0) and possesses high regularity elsewhere. Consequently, for precise numerical results one specifically has to resolve these locations accurately. Since $x_0 = 0$ is given here, it suffices to resolve the times $t_{\text{on}}, t_{\text{off}}$ accurately via using a fine time grid.

5 Conclusion and Outlook

A complete mathematical description of the coupling of the heat equations including the boundary conditions of the TPS together with the standard trajectory optimization problem is now available. This allows a more detailed mathematical investigation of the (numerical) solution properties. In particular, its semi-discretized version – considered as an ODE-constrained optimal control problem – can be analyzed using the concept of an order of a state constraint. It turns out that the arising state constraints show orders of one or upto $N + 1$ for N layers in the TPS. However, the numerical results show that only two state-constraints of order one and two are active, the dynamic pressure limit and the limit on the outer wall temperature Θ_1 . These two state constraints already yield an involved optimal control problem. The proposed direct solution method is, however, able to produce the optimal solution.

Since it is out of scope to establish deep theoretical aspects for such a complicated and nonlinear dynamical system describing the hypersonic plane, a simpler prototype problem was set up in former works, the hypersonic rocket car [18, 19, 20, 21]. However, the numerical results for the hypersonic plane are in agreement with the facts that were proven for the rocket car. In particular, the state constraint on the PDE variable is active only on a line in the space time cylinder.

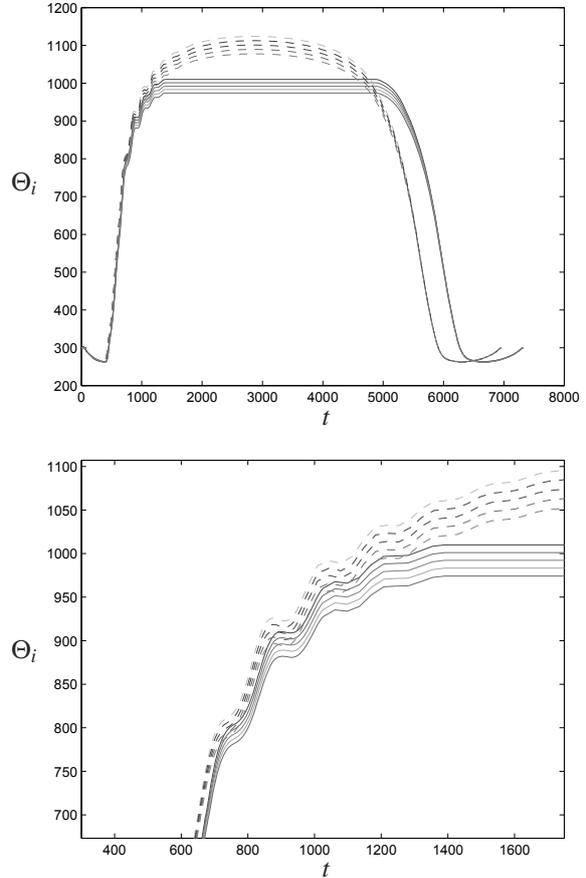


Figure 3: Temperatures Θ_i [K] in the i -th layer of the TPS ($i = 1, \dots, 5$ from top to bottom) (solid line: constrained, dashed line: unconstrained) and zoom around t_{on} .

This is typical for ODE-PDE control problems, where the controls enter just the ODE. Moreover, the spatial position of the active set is fixed also for the hypersonic plane. Accordingly, the state constraint can be reduced to this spatial position, which frees computational effort and memory for the solution method. Additionally, this reduction might also speed up the convergence of the optimizer.

Coupled ODE-PDE control problems with state constraints, as the one investigated here, frequently arise in applications. However, their analytical and numerical treatment deserves a closer investigation in the future.

References

- [1] Högenauer, E.: Raumtransporter. *Zeitschrift für Flugwissenschaften und Weltraumforschung*, 11, pp. 309–316, 1987.
- [2] Kuczera, H., Krammer, P., Sacher, P.: *Sänger and the German Hypersonics Technology Programme – Status Report 1991*. 42nd IAF-Congress, Montreal, Canada, Paper-No. IAF-91-198, 1991.
- [3] Buhl, W., Ebert, K., Herbst, H.: *Optimal Ascent Trajectories for Advanced Launch Vehicles*. 4th AIAA International Aerospace Planes Conference, Orlando, 1992.
- [4] Bayer, R., Sachs, G.: Optimal Return-to-Base Cruise of Hypersonic Carrier Vehicles. *Zeitschrift für Flugwissenschaften und Weltraumforschung*, 19, pp. 47–54, 1995.
- [5] Sachs, G., Dinkelmann, M.: Reduction of Coolant Fuel Losses in Hypersonic Flight by Optimal Trajectory Control. *J. Guid. Control Dynam.*, 19, pp. 1278–1284, 1996.
- [6] Miele, A., Mancus, S.: Optimal Ascent Trajectories and Feasibility of Next-Generation Orbital Spacecraft. *J. Optim. Theory App.*, 95, pp. 467–499, 1997.
- [7] Windhorst, R., Ardema, M.D., Bowles, J.V.: *Minimum Heating Reentry Trajectories for Advanced Hypersonic Launch Vehicles*. Proc. AIAA Guidance, Navigation, and Control Conference, New Orleans, USA, AIAA-97-3535, 1997.
- [8] Meese, E., Norstrud, H.: Simulation of Convective Heat Flux and Heat Penetration for a Spacecraft at Re-entry. *Aerosp. Sci. and Technol.*, 6, pp. 185–194, 2002.
- [9] Bertin, J.J., Cummings, R.M.: Fifty years of hypersonics: where we’ve been, where we’re going. *Progress in Aerospace Sciences*, 39, pp. 511–536, 2003.
- [10] DFG, Deutsche Forschungsgemeinschaft: *Basic Research and Technologies for Two-Stage-to-Orbit Vehicles*. Wiley-VCH Verlag, Weinheim, 2005.
- [11] Dinkelmann, M., Wächter, M., Sachs, G.: Modelling and simulation of unsteady heat transfer effects on trajectory optimization of aerospace vehicles. *Mathematics and Computers in Simulation*, 53, pp. 389–394, 2002.
- [12] Wächter M.: *Optimalflugbahnen im Hyperschallflug unter Berücksichtigung der instationären Aufheizung*. Dissertation, Lehrstuhl für Flugmechanik und Flugregelung, Technische Universität München, 2004.
- [13] Witzgall, M., Chudej, K.: *Flight Path Optimization subject to Instationary Heat Constraints*. In: Troch, I., Breitenecker, F. (eds.): 7th Vienna International Conference on Mathematical Modelling (MATHMOD 2012). - Wien: International Federation of Automatic Control, pp. 1141–1146, 2012.
- [14] Betts, J.: *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM, Philadelphia, 2001.
- [15] Tröltzsch, F.: *Optimale Steuerung partieller Differentialgleichungen*. Vieweg+Teubner, Wiesbaden, 2. Auflage, 2009.
- [16] Hinze, M., Pinnau, R., Ulbrich, M., Ulbrich, S.: *Optimization with PDE Constraints*. Springer Netherlands, 2009.
- [17] Borzi A., Schultz, V.: *Computational Optimization of Systems Governed by Partial Differential Equations*. SIAM, Philadelphia, 2012.
- [18] Pesch, H.J., Rund, A., von Wahl, W., Wendl, S.: On Some New Phenomena in State-constrained Optimal Control if ODE as well as PDE are Involved. *Control and Cybernetics*, 39, 3, pp. 647–660, 2010.
- [19] Wendl, S., Pesch, H.J., Rund, A.: *On a State-Constrained PDE Optimal Control Problem arising from ODE-PDE Optimal Control*. In: M. Diehl, F. Glineur, and W. Michiels (eds.): Recent Advances in Optimization and its Applications in Engineering. Springer, Heidelberg, pp. 429–438, 2010.
- [20] Pesch, H.J., Rund, A., von Wahl, W., Wendl, S.: *On a Prototype Class of ODE-PDE State-constrained Optimal Control Problems. Part 1: Analysis of the State-unconstrained Problems*. Preprint, Universität Bayreuth, 19 pages, 2011; published as appendix C.2 in [21].

- [21] Rund, A.: *Beiträge zur Optimalen Steuerung partiell-differential algebraischer Gleichungen*. Dissertation, Fakultät für Mathematik, Physik und Informatik, Universität Bayreuth, 2012.
- [22] Miele, A.: *Flight Mechanics. Volume 1: Theory of Flight Paths*. Pergamon Press, 1962.
- [23] Hamilton, W.E.: On Nonexistence of Boundary Arcs in Control Problems with Bounded State Variables. *IEEE Trans. Automatic Control*, AC-17, pp. 338–343, 1972.
- [24] Eich-Soellner, E., Führer, C.: *Numerical Methods in Multibody Dynamics*. Teubner, Stuttgart, 1998.
- [25] Eich, E.: *Projizierende Mehrschrittverfahren zur numerischen Lösung von Bewegungsgleichungen technischer Mehrkörpersysteme mit Zwangsbedingungen und Unstetigkeiten*. VDI Reihe 18, Nr. 109, Düsseldorf, 1992.
- [26] Schulz, V.H., Bock, H.G., Steinbach, M.C.: Exploiting Invariants in the Numerical Solution of Multipoint Boundary Value Problems for DAE. *SIAM J. Sci. Comput.*, 19, 2, pp. 440–467, 1998.
- [27] Chudej, K., Günther, M.: Global State Space Approach for the Efficient Numerical Solution of State-Constrained Trajectory Optimization Problems. *J. Optim. Theory App.*, 103, 1, pp. 75–93, 1999.
- [28] Chudej, K.: *Effiziente Lösung zustandsbeschränkter Optimalsteuerungsaufgaben*. Habilitation, Fakultät für Mathematik und Physik, Universität Bayreuth, 2001.
- [29] Campbell, S.L., Gear, C.W.: The Index of General Nonlinear DAEs. *Numerische Mathematik*, 72, pp. 173–196, 1995.
- [30] von Stryk, O.: *Numerische Lösung optimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der adjungierten Variablen*. Dissertation, Mathematisches Institut, Technische Universität München, 1994.
- [31] Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. *SIAM Review*, 47, 1, pp. 99–131, 2005.

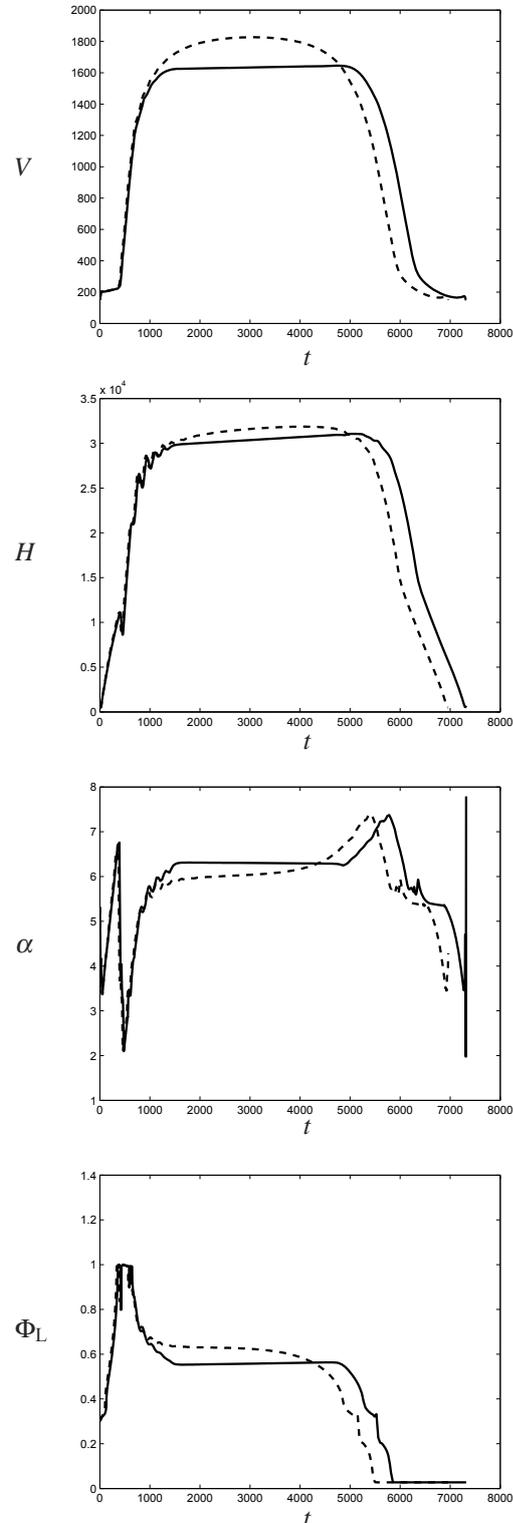


Figure 4: Velocity V [m/s], altitude H [m], angle of attack α , and equivalence ratio Φ_L (solid line: constrained, dashed line: unconstrained)

Experimentelle und numerische Untersuchung der Dämpfungsparameter von unverstärkten und glasfaserverstärkten Kunststoffen

Daniela Wurm¹, Jürgen Krome¹, Domingo Guerrero²

¹Hochschule Hamm-Lippstadt, Lippstadt

²Hella KG Hueck & Co. KGaA, Lippstadt

juergen.krome@hshl.de

Aufgrund ihrer multifunktionalen Eigenschaften kommt Kunststoffen heutzutage eine herausragende Bedeutung bei vielen industriellen Anwendungen zu. Die Anforderungen, die dabei an die statischen und dynamischen Eigenschaften der Kunststoffbauteile gestellt werden, sind hoch. Die dynamischen Eigenschaften der Bauteile werden wesentlich durch das Dämpfungsverhalten der verarbeiteten Polymere beeinflusst. In der Literatur und in Materialdatenbanken finden sich jedoch nur sehr eingeschränkt Angaben zu diesen Materialeigenschaften. Im Rahmen der Entwicklung neuer Produkte wird das dynamische Verhalten von Einzelteilen und/oder Baugruppen häufig mit Hilfe von Finite-Elemente-Simulationen (FEM-Simulationen) analysiert. Sollen mit solchen dynamischen FE-Simulationen nicht nur die Eigenfrequenzen der Systeme korrekt ermittelt werden, sondern auch die Höhen der Schwingungsamplituden richtig bestimmt werden, so muss das Dämpfungsverhalten der Kunststoffe bzw. der Kunststoffbauteile in den FE-Simulationen berücksichtigt werden. Hierfür sind Kenntnisse über das Dämpfungsverhalten der projektrelevanten Polymere sowie der verwendeten Kunststofffügeverbindungen erforderlich, welche oft experimentell ermittelt werden müssen. In diesem Beitrag wird eine Methode vorgestellt, mit der die Dämpfungsparameter von Kunststoffen, die bei der ersten Biegeeigenfrequenz auftreten, experimentell mit Hilfe von Abklingversuchen und einem Laser-Vibrometer ermittelt werden können. Ferner wird dargestellt, wie die ermittelten Dämpfungsparameter mit Hilfe eines Strukturdämpfungsmodells numerisch abgebildet werden können und es wird auf die Nachteile eingegangen, die mit diesem Strukturdämpfungsmodell verbunden sind.

1 Einleitung

Während der Produktentwicklung erfordern Freigabeprozesse und/oder Optimierungsschleifen häufig eine Analyse des dynamischen Verhaltens gegenwärtiger Produktentwicklungsstände. Diese Analysen können wahlweise mit experimentellen Versuchen oder numerisch mit Hilfe von dynamischen FEM-Simulationen durchgeführt werden. Die experimentellen Untersuchungen stellen hier die wesentlich zeitintensivere Alternative dar, sodass den FEM-Simulationen in diesem Bereich ein sehr hoher Stellenwert zukommt. Sie erfordern keine Probenkörper und Einspannvorrichtungen und es bedarf insbesondere keiner freien Maschinenkapazitäten an den Prüfständen.

Die Finite Elemente Methode stellt grundsätzlich eine Reihe von dynamischen Simulationsansätzen zur Verfügung, mit denen das dynamische Verhalten von Bauteilen analysiert werden kann. Eine effektive Anwendung dieser Verfahren setzt allerdings voraus, dass die Werkstoffkennwerte, insbesondere der Elastizitätsmodul und die Werkstoffdämpfungsparameter, der im System verarbeiteten Materialien mit hinrei-

chender Genauigkeit bekannt sind. [1] Sind in dem zu analysierenden Bauteil Kunststoffe beliebiger Art verarbeitet, gestalten sich die FEM-Simulationen auf Grund dieser Voraussetzungen schwierig. Dies begründet sich damit, dass die Werkstoffdämpfungsparameter von Kunststoffen häufig nicht oder nicht ausreichend genau bekannt sind. Insbesondere in einschlägigen Materialdatenbanken finden sich häufig keine Angaben zu Dämpfungsparametern. Die Begründung hierfür liegt in den zahlreichen Faktoren, die einen Einfluss auf die Werkstoffdämpfung nehmen / nehmen können und eine generische Definition der Materialdämpfung häufig nicht ermöglichen. [2]

2 Experimentelle Ermittlung der Dämpfungsparameter

Um das dynamische Verhalten der zu untersuchenden Systeme dennoch mit Hilfe der FEM-Simulation analysieren, bewerten und optimieren zu können, müssen die Werkstoffdämpfungsparameter der relevanten Kunststoffe dann für das jeweils vorliegende Projekt experimentell wahlweise aus Resonanzkurven oder aus Abklingkurven ermittelt werden. Erstge-

nannte können ausschließlich durch experimentelle Versuche auf einem dynamischen Prüfstand (Shaker) ermittelt werden. Steht ein solcher nicht zur Verfügung, können die Dämpfungsparameter alternativ mit Hilfe von Abklingversuchen ermittelt werden. Hierfür muss das Abklingverhalten der Systeme mit einem Laservibrometer (berührungslos!) oder mit Hilfe von einem Beschleunigungsaufnehmer und einem Oszilloskop in Form von Abklingkurven aufgezeichnet werden. Aus diesen Abklingkurven kann dann die Werkstoffdämpfung ermittelt werden. [3]

Die Durchführung von Abklingversuchen ist in der DIN EN ISO 6721-1 genormt. [4] Grundsätzlich wird bei den Versuchen ein zu untersuchender Probenkörper fest in eine Einspannvorrichtung eingespannt und einmalig durch ein Störsignal aus seiner Gleichgewichtslage ausgelenkt. Das Störsignal wird dabei in der Regel mit Hilfe von einem Impulshammer appliziert. Anschließend führt das System freie (gedämpfte) Schwingungen durch, die mit den bereits genannten Verfahren aufgezeichnet werden können. Auf diese Weise wird das Abklingverhalten der Systeme ermittelt. Dieses kann anschließend in Form von einem Amplituden-Zeit-Diagramm dargestellt werden. Abbildung 1 zeigt exemplarisch den Versuchsaufbau eines Abklingversuches, der mit Hilfe eines Laservibrometers aufgezeichnet werden soll. Abbildung 2 zeigt eine in einem Amplituden-Zeit-Diagramm dargestellte Abklingkurve eines experimentell untersuchten Systems.

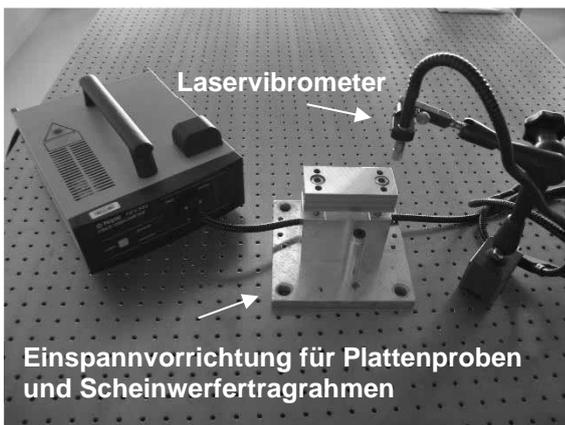


Abbildung 1: Aufbau eines Ausschwingversuchs

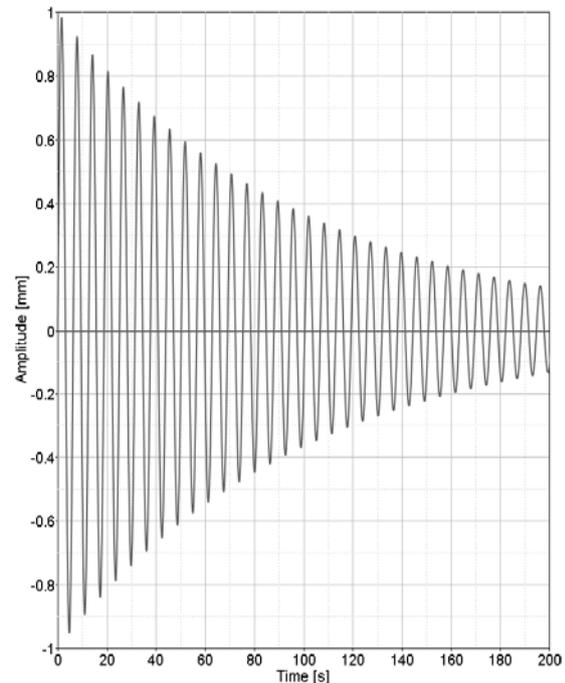


Abbildung 2: Gemessene Abklingkurve

Aus dem Amplituden-Zeit-Diagramm kann das Lehrsche Dämpfungsmaß als charakteristische Größe für die Werkstoffdämpfung über die Auswertungsmethode des logarithmischen Dekrements ermittelt werden. Das logarithmische Dekrement ist definiert als der natürliche Logarithmus des Verhältnisses von zwei aufeinanderfolgenden Schwingungsamplituden, die in die gleiche Richtung zeigen [3]:

$$\Lambda = \ln \frac{x_m}{x_n} = \frac{2\pi\omega_0 D}{\sqrt{\omega_0^2 - \delta^2}} = \omega_0 D t \quad (1)$$

Soll das logarithmische Dekrement auf Basis von mehr als zwei Schwingungsamplituden betrachtet werden, so ist die nachfolgende Formel anzuwenden [3]:

$$\Lambda = \ln \frac{u(t_i)}{u(t_{i+n})} = \frac{1}{n} \ln \frac{u(t_i)}{u(t_{i+n})} \quad (2)$$

mit $n = 1, 2, \dots, n$.

Wird das Lehrsche Dämpfungsmaß über diese Auswertungsmethode ermittelt, ist zwingend zu berücksichtigen, dass alle Berechnungen ausschließlich auf Basis der Schwingungsamplituden erfolgen, die im System im eingeschwungenen Zustand auftreten. Die Notwendigkeit hierfür ergibt sich aus Gleichung 1. Mit der Eigenfrequenz schwingt ein frei schwingendes System ausschließlich im eingeschwungenen Zustand.

Gemäß DIN EN ISO 6721-1 eignet sich die Methode des logarithmischen Dekrements besonders für Dämpfungsanalysen bei schwach gedämpften Systemen. Die Norm rät außerdem dazu, Dämpfungsparameter von einem Material mit mindestens drei voneinander unabhängigen Versuchen zu ermitteln, um eine minimale statistische Sicherheit der Versuchsergebnisse zu gewährleisten. In diesem Zusammenhang ist es außerdem ratsam, die Streuung der für einen Werkstoff ermittelten Dämpfungsparameter durch die Berechnung der den Messwerten zu Grunde liegenden Standardabweichung zu analysieren. [4] Tabelle 1 zeigt die experimentell ermittelte Materialdämpfung für die untersuchten Werkstoffe jeweils gültig für einen definierten Frequenzbereich und den ersten Bieegeigenmode.

Tabelle 1: Gemessene Dämpfungsparameter

Werkstoff	Materialdämpfung	Relative Standardabweichung
1	0,0318	3 %
2	0,00321	2 %
3	0,0022	5 %
4	0,035	3 %

Im Anschluss an die Ermittlung der Dämpfungskennwerte liegt es in den Händen des Simulationsexperten, ein Dämpfungsmodell auszuwählen, mit dem die experimentell ermittelten Materialdämpfungsparameter in den dynamischen FEM-Simulationen abgebildet werden können. Für Werkstoffdämpfungsparameter, die über einen endlichen Frequenzbereich konstant sind, wie es bei den in Tabelle 1 dargestellten Parametern der Fall ist, wird auf Basis zahlreicher nach diesem Schema durchgeführter Dämpfungsuntersuchungen die Anwendung der Strukturdämpfung empfohlen.

Bei der Strukturdämpfung handelt es sich um eine frequenzunabhängige Dämpfung, die im Vergleich zur modalen Dämpfung und zur Rayleigh-Dämpfung zunächst nicht auf Basis des allgemeinen viskosen Dämpfungsansatzes beschrieben werden kann. Vielmehr basiert das Modell der Strukturdämpfung auf einer Dämpfungsmatrix der Form [5]:

$$D(\omega) = \frac{\Psi}{\omega} K \quad (3)$$

Der Koeffizient Ψ beschreibt hier den Werkstoffverlustfaktor und ω die Erregerkreisfrequenz. Für den Werkstoffverlustfaktor Ψ gilt gemäß [6]:

$$\Psi = \frac{d\omega}{k} = \frac{c}{m\omega} \quad (4)$$

Da in der Praxis aufgrund der Linearität der Differentialgleichungen überwiegend mit der viskosen Dämpfung gearbeitet wird, sind viele Softwareanwendungen (auch Simulationssoftware) auf die Abbildung viskoser Dämpfungsparameter ausgelegt. Um dennoch mit dem Modell der Strukturdämpfung arbeiten zu können, muss die Strukturdämpfung in eine äquivalente viskose Dämpfung umgerechnet d.h. in Form eines Lehrschen Dämpfungsmaßes dargestellt werden. Gemäß [5] gilt für einen viskos gedämpften Schwinger der Zusammenhang:

$$(-\omega^2 + 2i\omega D\omega_0 + \omega_0^2)U = \frac{F_0}{m}, \quad (5)$$

während für einen Schwinger, der mit der Strukturdämpfung gedämpft ist, die nachfolgende Beziehung gilt:

$$(-\omega^2 + i\Psi\omega_0^2 + \omega_0^2)U = \frac{F_0}{m}. \quad (6)$$

Diese beiden Gleichungen sind identisch, wenn gilt:

$$\Psi\omega_0 = 2\omega D. \quad (7)$$

Diese Bedingung wird nur für $\omega = \omega_0$ erfüllt. Für diese Frequenz kann die Strukturdämpfung daher mit Hilfe der nachfolgenden Formel exakt in eine viskose Dämpfung umgerechnet werden. [5]

$$\Psi = 2D \quad (8)$$

Für alle weiteren Frequenzen, für die $\omega \neq \omega_0$ gilt, beschreibt dieser Ansatz das reale Dämpfungsverhalten nur näherungsweise. Gleichung 7 wurde für den Fall $\omega = \omega_0$ entwickelt, da das dynamische Verhalten des Systems für diesen Fall am stärksten durch die Werkstoffdämpfung beeinflusst wird.

Die Anwendung des Strukturdämpfungsmodells in dynamischen FE-Simulationen unterliegt allerdings einer wesentlichen Beschränkung: Sie ist nur möglich, wenn ausschließlich das Resonanzverhalten des Systems untersucht wird. Dies begründet sich damit, dass das Modell das reale Dämpfungsverhalten wie bereits dargestellt nur für den Resonanzfall korrekt widerspiegelt. Ein experimentell durchgeführter Frequency-Sweep kann somit beispielsweise nicht mit Hilfe des Strukturdämpfungsmodells simulativ nachgebildet werden, da der Ansatz in diesem Fall das reale Systemverhalten in dem größten Teil des untersuchten Frequenzbereiches nicht exakt widerspiegelt, sondern nur für eine Resonanzfrequenz das Systemverhalten richtig abbildet. Eine Modalanalyse kann

hingegen problemlos mit diesem Dämpfungsansatz durchgeführt werden.

3 Zusammenfassung

Kunststoffbauteile sind häufig dynamischen Belastungen ausgesetzt, die zu unerwünschten Schwingungen in der Anwendung bis hin zum Versagen der Bauteile führen können. Wie stark die auftretenden Schwingungen eines Systems sind, hängt wesentlich von der Materialdämpfung im genutzten Frequenzbereich ab. Gerade bei Kunststoffen sind diese Dämpfungsparameter oft nicht bekannt. Sie müssen experimentell für den jeweiligen Anwendungsfall ermittelt werden. Für verschiedene Kunststoffproben wurde mithilfe von Abklingkurven die Dämpfungsparameter ermittelt und für anschließende FEM Simulationen aufbereitet. Mit den anschließenden Simulationen kann nun unter Nutzung der ermittelten Parameter das dynamische Verhalten realer Bauteile voraus berechnet werden, so dass eine zuverlässige Auslegung der Strukturen möglich ist.

4 References

- [1] UNBEKANNTER AUTOR: Abaqus 6.13 Documentation, [online], [o. Jahr], <http://129.97.46.200:2080/v6.13/E>. Beispiel und S. Exempel. *Titel der V. Zeitschrift*, S. 111–123, 2012.
- [2] Littmann, W.: Identifikation der Materialparameter von Kunststoffen – Messung von Elastizitätsmodul und Dämpfung (Verlustfaktor, Schwinggüte, Absorptionskoeffizient), ATHENA Technologie Beratung GmbH, Paderborn 2013
- [3] UNBEKANNTER AUTOR: Experimentelle Schwingungsanalyse – Allgemeines Messtechnisches Labor, Version 2.3, 30. April 2004, [online], http://webcache.googleusercontent.com/search?q=cache:cqRMYDgmURcJ:www.ids.uni-hannover.de/fileadmin/IDS/ids_lehre/AML/AML,Mechanik.pdf+%&cd=2&hl=de&ct=clnk&gl=us, (Zugriff am 22.09.2015, 08:04 Uhr)
- [4] DIN Deutsches Institut für Normung e.V.: DIN EN ISO 6721-1: Kunststoffe – Bestimmung dynamisch-mechanischer Eigenschaften – Teil 1: Allgemeine Grundlagen (ISO 6721-1:2011); Deutsche Fassung EN ISO 6721-1:2011, Beuth Verlag GmbH, Berlin 2011
- [5] Wandinger, J.: Dämpfungsmodelle, [online], [o.Jahr], wandinger.userweb.mwn.de/LA_Elastodynamik_1/v3_4.pdf, (Zugriff am 17.09.2015, 15:26 Uhr)
- [6] UNBEKANNTER AUTOR: Materialdämpfung, [online], [o.Jahr], <http://www.esocaet.com/wikiplus/index.php/Materialdaempfung>, (Zugriff am 20.09.2015, 14:23 Uhr)

Modellierung, Simulation und Validierung eines Betonpumpenmastes unter Berücksichtigung von Stahlelastizität und Hydraulikzylindern

Sophie Zorn, Thomas Breitenbach

Fakultät Maschinenwesen
Technische Universität Dresden
01602 Dresden, Germany

sophie.zorn@tu-dresden.de

Zur Systemanalyse eines Betonpumpenmastes wurde ein Mehrkörpermodell aus starren und elastischen Körpern, sowie den notwendigen Hydraulikzylindern erstellt und durch Messungen verifiziert. Für die Modellbildung wurde die kommerzielle Simulationssoftware *Dymola* von *Dassault Systems* verwendet, welche auf der objektorientierten Sprache *Modelica* basiert. Die Besonderheit des Modells liegt in der Darstellung der einzelnen Armsegmente als flexible Körper, da diese einen maßgeblichen Einfluss auf das Schwingungsverhalten des Systems haben. Hierfür wird der Ansatz des *Floating-Frame-of-Reference* verwendet, wobei jeder Körper mittels einer SID-Datei (*Standard Input Data*) beschrieben wird. Am Beispiel eines 42m langen Verteilermastes wird das Mehrkörpermodell erläutert, die durchgeführten Messungen vorgestellt, sowie die Übereinstimmung zwischen Simulationsergebnissen und Messdaten aufgezeigt. Untersuchungen am validierten Modell zeigen die unterschiedlichen Einflüsse der Stahlelastizität, sowie der Hydraulik auf das Schwingungsverhalten, wobei die Vernachlässigung einer der beiden Einflüsse zu stark abweichenden Ergebnissen führt.

1 Einleitung

Die Aufgabe einer Autobetonpumpe ist es, den vom Fahrmischer gelieferten Frischbeton an den gewünschten Ort auf der Baustelle zu transportieren. Zur Aufnahme des Betons ist ein Trichter vorhanden aus dem der Frischbeton mithilfe der Kernpumpe in die Förderleitung gepumpt wird. Zur flexiblen Positionierung des Betons auf der Baustelle wird die Förderleitung entlang eines zusammenfaltbaren und hydraulisch betätigten Mastes geführt. Je nach Typ ist der Mast zwischen 20 und 70 Metern lang und besteht aus drei bis sechs Segmenten. Über ein Drehgelenk ist der Mast mit dem Unterbau, der aus Abstützung und Fahrzeug besteht, verbunden. Die einzelnen Segmente sind jeweils über Drehgelenke gekoppelt. Angetrieben werden sie über Hydraulikzylinder, wobei die Linearbewegungen über Koppelgetriebe (z.B. Watt'sche oder Stephenson'sche Ketten) in Rotationen um die Gelenkpunkte übersetzt werden.

Der Trend zum Leichtbau, wie er aus der Luft- und Raumfahrt oder der Automobilindustrie bekannt ist, ist auch in der Baumaschinenbranche zu spüren. Leichtere Maschinen haben einen geringeren Kraftstoffverbrauch, können schneller bewegt werden und

benötigen für die Bewegung weniger Energie, folglich sind kleinere Antriebe ausreichend. Bei Maschinen, die auf ein Fahrzeug montiert werden, kommt hinzu, dass leichtere Maschinen Fahrgestelle mit weniger Achsen verwenden können und damit die Gesamtkosten gesenkt werden. Aufgrund der langen schmalen Armkonstruktionen fällt bei Autobetonpumpen der Trend zum Leichtbau bezüglich der Dynamik besonders ins Gewicht. Vor allem die rhythmischen Förderstöße regen den Verteilermast zu Schwingungen an, sodass eine modellbasierte Untersuchung im Entwicklungsprozess sinnvoll wird.

In [1] wird ein Betonpumpenmast modelliert und Simulationen des Ein- und Ausfaltens durchgeführt. Bei dem vorgestellten Modell handelt es sich um eine Mehrkörpersimulation, welche flexible Körper enthält und auch die hydraulischen Aktoren abbildet. Eine Validierung der Ergebnisse wird jedoch nicht präsentiert. [2] simuliert ebenfalls einen Betonpumpenmast aus fünf flexiblen Elementen in der Softwareumgebung von ADAMS. Bei einer Anregung am Drehkranz, welche die Arbeit der Kernpumpe darstellen soll, wird in horizontaler Konfiguration die Frequenz des Schwingens der Mastspitze gemessen. Diese stimmt mit Messwerten überein, die Amplitude

wird jedoch nicht überprüft. Auf die hydraulischen Komponenten der Maschine wird nicht weiter eingegangen und insgesamt nur eine Maststellung überprüft. Weitere Veröffentlichungen zur Modellbildung von Verteilmasten können in [3],[4],[5] gefunden werden.

2 Modellbildung

Für die Modellbildung wurde die kommerzielle Simulationssoftware *Dymola* von *Dassault Systems* verwendet, welche auf der objektorientierten Sprache *Modelica* basiert. Prinzipiell wird die mechanische Struktur vom Mast mittels starrer und flexibler Körper, Drehgelenken und translatorischen Antrieben abgebildet. Den Antrieben werden, mit Hilfe der *Modelon Hydraulics Library*, die Eigenschaften der entsprechenden Hydraulikzylinder zugewiesen. Über einen Hydraulikschaltplan können diese angesteuert und somit der Mast in jede gewünschte Konfiguration gefahren werden. Eine Besonderheit des Modells sind die als flexible Körper angenommenen einzelnen Armsegmente, wodurch die Stahlelastizität der Bauteile berücksichtigt wird und eine realistische Abbildung des Maschinenverhaltens resultiert.

Je nach Typ besteht der Verteilmast aus drei bis sechs einzelnen Segmenten welche als Arm 1 bis Arm 6 bezeichnet werden. Jeder Arm ist über ein Drehgelenk an seinen Vorgänger gekoppelt, wobei Arm 1 am Drehkopf der Maschine befestigt ist. Um unterschiedliche Maststellungen zu realisieren, werden die Arme mittels Hydraulikzylindern und entsprechenden Umlenkhebeln verfahren. Die Mastzylinder werden ausgehend vom Drehkopf alphabetisch bezeichnet, bei 5 Armen gibt es demnach den A- bis E-Zylinder.

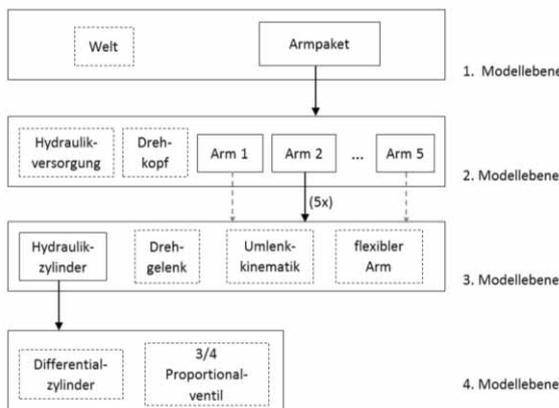


Abbildung 1. Übersicht der einzelnen Modellebenen

Der Aufbau des Simulationsmodells ist in Abb. 1 schematisch dargestellt. Er ist in vier Modellebenen gegliedert, wobei mit steigender Modellebene die Modellierung immer detaillierter wird. In der ersten Modellebene befindet sich die Welt und das komplette Armpaket, welchem über Parameter bestimmte Eigenschaften zugeordnet werden können. Innerhalb des Armpakets (2. Modellebene) sind die fünf Armsegmente enthalten sowie die Hydraulikversorgung der Aktuatoren. Die einzelnen Armsegmente (3. Modellebene) wiederum sind alle gleich aufgebaut, jedes besitzt den Hydraulikzylinder und das Drehgelenk zum Vorgänger, die Umlenkinematik sowie das Armsegment als flexibler Körper. Die Hydraulikzylinder sind in Modellebene 4 separat aufgebaut.

2.1 Theorie der flexiblen Körper

Flexible Körper verformen sich unter Einwirkung von äußeren Kräften und es kann zu Durchbiegungen und Schwingungen in den Elementen kommen. Im Gegensatz zum starren Körper bewegt sich somit ein beliebiger Punkt auf dem Körper bezüglich eines körperfesten Koordinatensystems. Der Abstand zweier Punkte auf dem Körper bleibt folglich nicht mehr konstant. Die Verschiebung eines Punktes bezüglich seiner Ausgangslage wird als Verformung oder Deformation bezeichnet. Die Erweiterung eines Mehrkörpersystems um flexible Körper hat den Vorteil, dass einem Körper gleichzeitig Masse und Steifigkeit zugeordnet werden können und eine Trennung nicht erforderlich ist. Um einen flexiblen Körper zu beschreiben, wurden unterschiedliche Formulierungen entwickelt, wobei zwischen großen und kleinen Deformationen unterschieden wird. Im Folgenden wird die häufig eingesetzte Formulierung des *Floating Frame of Reference* näher erläutert, welche auch bei *Dymola* zur Berechnung verwendet wird und für kleine Deformationen gültig ist [6].

Die Idee dieses Ansatzes ist, die Bewegung des flexiblen Körpers aufzuteilen in eine Referenzbewegung im globalen Koordinatensystem und eine Relativbewegung (Verformung) bezüglich eines körperfesten Koordinatensystems. Somit können kleine Verformungen, auch wenn eine große Referenzbewegung vorliegt, in den Verformungskoodinaten linearisiert werden. Für die Beschreibung von verformbaren Körpern wird der Ansatz des starren Körpers um einen flexiblen Anteil erweitert. Die erwähnte Referenzbewegung entspricht dem Starrkörperverhalten. In Abbildung 2 sind diese Beziehungen dargestellt.

Bei einem starren Körper i befindet sich der Punkt K bezüglich eines körperfesten Bezugssystems ${}_iO$ zu jedem Zeitpunkt am Ort r^i . Wenn p^i den Ursprungsort des Körperkoordinatensystems ${}_iO$ angibt, wird die Position von K im Inertialsystem durch

$$k^i(t) = p^i(t) + r^i \quad (1)$$

beschrieben.

Um die Deformation des flexiblen Körpers abzubilden, wird davon ausgegangen, dass sich K um einen Vektor u^i von seiner ursprünglichen Position entfernt ($|u^i| \ll |r^i|$), es ergibt sich ein neuer Positionsvektor

$$k_{flex}^i(t) = p^i(t) + r^i + u^i(t) \quad (2)$$

wobei u^i zeitabhängig ist.

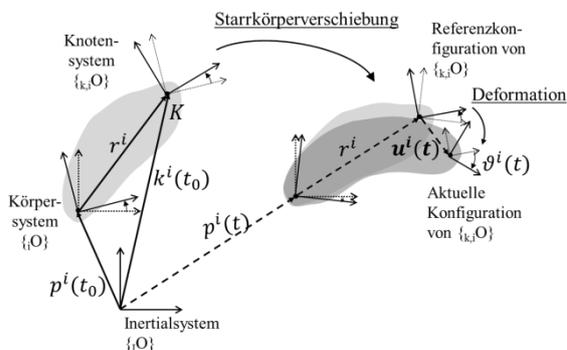


Abbildung 2. Beziehungen und Bezeichnungen bei Verwendung des Floating Frame of Reference

2.2 Verformungsansatz

Die Verschiebung der Knotenkoordinatensysteme um den Vektor u wird als Verformung bezeichnet und muss, da sie in den allermeisten Fällen unbekannt ist, über geeignete Funktionen approximiert werden. Hierbei werden translatorische und rotatorische Verschiebungen getrennt beschrieben, sodass zwei Ansatzfunktionen notwendig sind. Die translatorischen Bewegungen werden weiterhin mit u und die rotatorischen mit ϑ bezeichnet. Mit Hilfe des Ritz'schen Verfahrens werden beide durch eine nur von der Zeit abhängige generalisierte Koordinate $q(t)$ und einer ortsabhängigen Ansatzfunktion beschrieben

$$u(R, t) = \Phi(R)q(t) \quad (3)$$

$$\vartheta(R, t) = \Psi(R)q(t). \quad (4)$$

Die Ansatzfunktionen Φ und Ψ müssen beim Ritz'schen Verfahren den geometrischen Randbedingungen genügen und ihre Wahl legt die Qualität der Näherung zur Darstellung der Verformung fest. Häu-

fig werden zur Darstellung Eigenschwingformen herangezogen, wobei jedoch nicht alle Eigenformen Verwendung finden [7]. Zu hohen Eigenfrequenzen gehörende Eigenformen leisten nur einen geringen Beitrag zur Darstellung der Verformung, daher können diese im Vorfeld herausgenommen werden. Bei der Wahl der Eigenformen ist darauf zu achten, dass sie die gewünschten oder erwarteten Verformungen darstellen können.



Abbildung 3. Eigenschwingformen eines Armsegmentes

Eine Sichtung der zu den niedrigen Eigenfrequenzen gehörenden Eigenfunktionen ist notwendig, weil nicht immer zwangsläufig die niedrigsten Frequenzen zur gewünschten Verformung gehören. Abbildung 3 zeigt am Beispiel eines Armsegmentes der Betonpumpe die ersten drei Eigenformen.

2.3 Einbeziehen flexibler Körper in die MKS

Um einen Körper als flexiblen Körper in *Dymola* einzubinden, muss er entsprechend vorbereitet werden. Flexible MKS bilden eine Verknüpfung der Finite-Elemente-Methode und der herkömmlichen MKS mit starren Körpern. Allgemein kann die Einbindung eines flexiblen Körpers in fünf Schritte gegliedert werden:

1. Erzeugen der CAD-Geometrie
2. Vernetzung der Geometrie im FE-Programm
3. Modellordnungsreduktion (MOR) und Modalanalyse
4. Aufbereitung der Daten für MKS
5. Einbinden in MKS Modell

Die Einbindung eines FE-Körpers mit seinen mehreren tausend Freiheitsgraden würde den Rahmen einer MKS sprengen, sodass der Zwischenschritt der Modellordnungsreduktion notwendig ist. Hierbei wird das diskrete Modell auf ein System kleinerer Dimension reduziert. Dafür werden Knoten ausgewählt die

3.2 Auswertung

In Abbildung 6 bis Abbildung 8 sind die Messdaten den Simulationsdaten zu den Versuchen *M1*, *M2* und *M3* gegenübergestellt. Für die Auswertung wurden die Marker an den Positionen der Gelenke verwendet. Die mithilfe der Kraftmessdose ermittelten Vorspannkraften aus den Versuchen, werden in der Simulation durch eine externe Kraft auf die Mastspitze dargestellt. Aus Vertraulichkeitsgründen sind die Messwerte nur qualitativ angegeben, sodass keine Aussagen über die absoluten Abweichungen gemacht werden können.

An den Ergebnissen aus Messung *M1* wird die hohe Übereinstimmung zwischen Simulations- und Messergebnissen sichtbar. In Abbildung 6 sind die vertikalen Positionen des D- und E-Gelenks, sowie der Mastspitze (Tool Center Point) gegenübergestellt. Eine Frequenzauswertung ergibt eine Abweichung von $\Delta f_1 < 1\%$ und auch das Abklingverhalten stimmt gut überein. Lediglich die Amplituden der Simulation fallen etwas geringer aus und erreichen 85% der Messwerte.

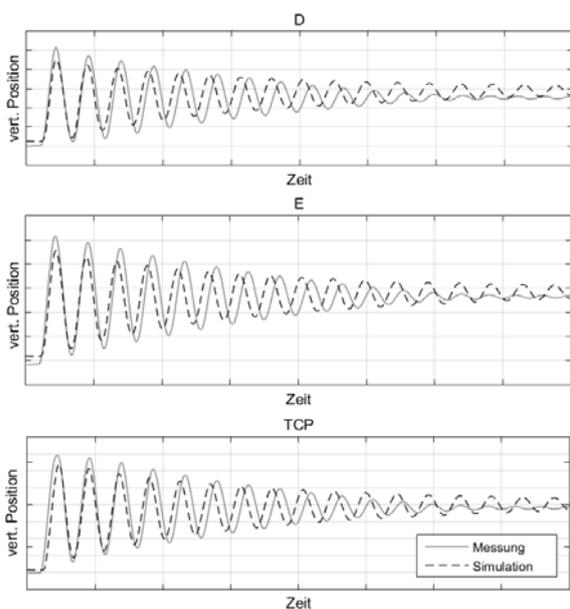


Abbildung 6. Simulations- und Messergebnisse für Bogenstellung (*M1*)

Die Auswertung der gestreckten Stellung *M2* weist ähnlich hohe Übereinstimmungen auf (s. Abb. 7).

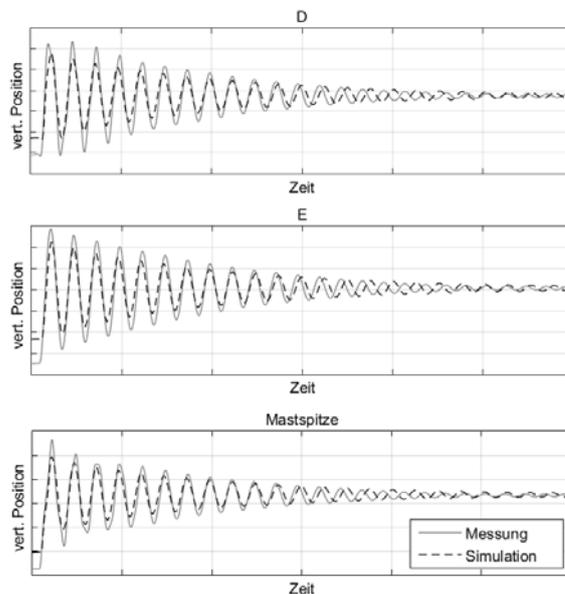


Abbildung 7. Simulations- und Messergebnisse für gestreckte Stellung (*M2*)

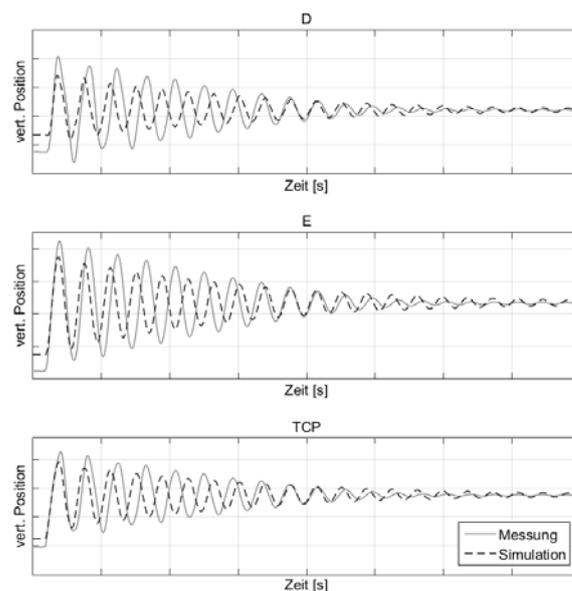


Abbildung 8. Simulations- und Messergebnisse für Roll-Z Stellung (*M3*)

Die stärksten Abweichungen finden sich in der Roll-Z Stellung. Auch hier stimmt das Abklingverhalten wieder gut überein, jedoch schwingt der Mast in der Simulation mit einer leicht erhöhten Frequenz. Die Abweichung liegt bei $\Delta f_3 \approx 9\%$. Die Amplituden fallen in der Simulation wieder geringer aus und erreichen 82% der Messwerte.

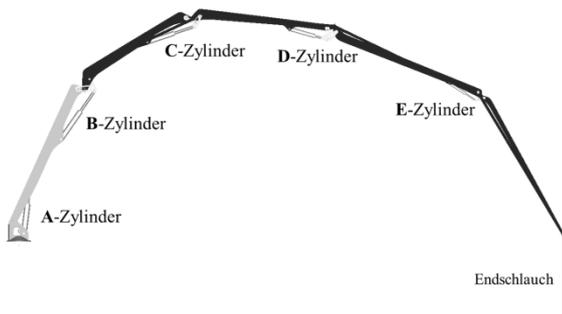


Abbildung 9. Bogenstellung als Startposition für die Messungen *M4-M8*

Zur Überprüfung der Systemdynamik sind die Messungen *M4-M8* durchgeführt worden. Jeder Zylinder wird ausgehend von der Bogenstellung einzeln angesteuert und die Auswirkung auf die Mastspitze wird untersucht (Abbildung 9). Durch das abrupte Abstoppen werden Schwingungen im System induziert, welche auch im Modell abgebildet werden. Die Messdaten sind den Simulationsergebnissen in Abbildung 10 gegenübergestellt.

Für das Vorgehen am A-,B-,D- und E-Zylinder ergeben sich hohe Übereinstimmungen zwischen Simulations- und Messdaten. Amplituden als auch Frequenzen stimmen überein. Abweichungen bis zu 15% treten zwischen Messung und Simulation beim C-Zylinder auf.

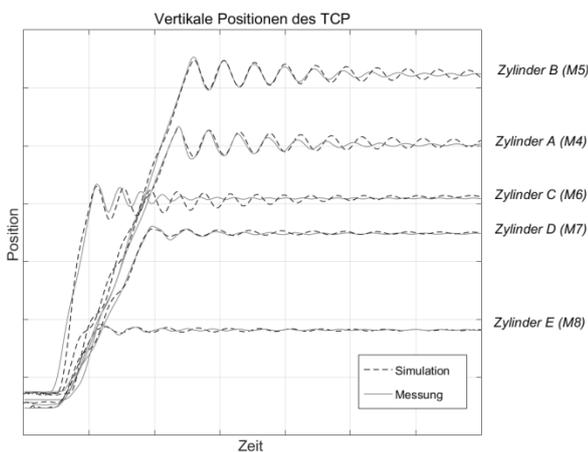


Abbildung 10. Schwingungsverhalten der Mastspitze bei den Versuchen *M4 - M8*

Die Auswertung der Mess- und Simulationsergebnisse zeigt auf, dass das erstellte flexible Mehrkörpersystem das Systemverhalten des Verteilermastes realistisch abbildet.

4 Simulationsstudien

Das entwickelte und verifizierte Mastmodell wird zur Systemanalyse eingesetzt. Von Interesse ist der Einfluss der Stahlelastizität sowie der Hydraulikzylinder auf das Schwingungsverhalten. Hierfür werden Simulationen an unterschiedlichen Modellen ohne flexible Körper und ohne Kompressibilität der Hydraulik durchgeführt und dem verifizierten Modell gegenübergestellt.

Am verifizierten Modell aus Abschnitt 2 und 3 wird in einer Simulation untersucht, wie sich der Betonpumpenmast verhält, wenn keine Stahlelastizität berücksichtigt wird, die Arme also aus starren Körpern modelliert werden. Die Hydraulikzylinder bleiben dabei unverändert im Modell enthalten. Dem gegenübergestellt wird ein Mastmodell, welches die Stahlelastizität berücksichtigt, bei dem jedoch die Hydraulikzylinder als starre Elemente eingebunden sind. An beiden Modellen wird Versuch *M2*, der in Abschnitt 3.1 erläutert wurde, durchgeführt. Hierfür ist der Mast in der gestreckten Konfiguration und startet bei beiden Simulationen mit denselben Gelenkwinkelvorgaben. Zum Vergleich der Simulationen wird die vertikale Position der Mastspitze verwendet sowie die Daten aus Versuch *M2*, bei dem Stahlelastizität sowie Hydraulik abgebildet sind. In Abbildung 11 sind die Simulationsergebnisse dargestellt. Als Referenz soll das flexible Modell mit Hydraulik dienen, welches den Betonpumpenmast realitätsnah abbildet.

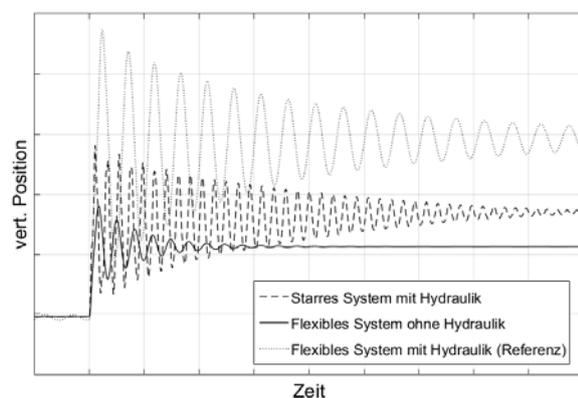


Abbildung 11. Vertikale Position der Mastspitze in gestreckter Stellung beim Ausschwingen von Versuch *M2* mit unterschiedlichen Mastmodellen

Zu Beginn wirkt an der Mastspitze jeweils eine Kraft, welche die Spitze nach unten drückt. Nachdem die Kraft gelöst wird, kann der Arm frei schwingen. Im

Verlauf der drei Ausschwingvorgänge ist zu erkennen, dass die Mastspitze unterschiedliche vertikale Ruhepositionen einnimmt, sie durch die angreifende Kraft demnach unterschiedlich stark herunter gezogen wurde. Das System aus flexiblen Körpern ohne Berücksichtigung der Hydraulikzylinder lässt sich am wenigsten verformen und die Schwingungen klingen schnell ab, was an der gewählten Dämpfung von 3 pro Schwingform liegt. Weitaus stärkere Schwingungen, höherer Amplitude und Frequenz sowie geringerer Dämpfung bilden sich im System aus starren Körpern und Hydraulikzylindern. Aus diesen Ergebnissen wird ersichtlich, dass ein wesentlicher Anteil der Gesamtdurchbiegung des Mastes aus den Hydraulikkomponenten resultiert. Im Vergleich zu den Ergebnissen mit flexiblen Körpern und Hydraulikzylindern wird die Durchbiegung von keinem anderen Modell realitätsnah abgebildet. Auch die Eigenfrequenzen liegen bei beiden Modellen höher. Mit den durchgeführten Simulationen kann gezeigt werden, dass weder die flexiblen Körper noch die Hydraulikzylinder alleine eine realitätsgetreue Abbildung des Betonpumpenmastes darstellen können. Die einzelnen Armsegmente als flexible Körper anzunehmen ist daher berechtigt und führt im Zusammenspiel mit der Hydraulikmodellierung zu sehr guten Simulationsergebnissen.

5 Zusammenfassung

Das zur Systemanalyse eines Betonpumpenmastes entwickelte Mehrkörpermodell aus starren und elastischen Körpern sowie den notwendigen Hydraulikzylindern bildet das reale System realitätsnah ab, wie die Übereinstimmung zwischen Simulationsergebnissen und Messdaten aufzeigt. Untersuchungen am validierten Modell zeigen die unterschiedlichen Einflüsse der Stahlelastizität und der Hydraulik auf das Schwingungsverhalten. Die Vernachlässigung einer der beiden Einflüsse führt zu einem stark abweichenden Schwingungsverhalten.

6 References

- [1] X. Gang, Z. Yan, Z. Fanli und C. Liping: *Modelling and Simulation of the Coupled Rigid-flexible Multibody Systems in MWorks*. In: Proceedings of the 9th International Modelica Conference, Seiten 405-416, 2012.
- [2] Y. Ping, W. Rui, S. Yuhong und L. Pengmin: *Dynamic Simulation of Concrete Pump Truck's Boom System Based on Rigid-Flexible Coupling Model*. Applied Mechanics and Materials, Seiten 989-993, 2012.
- [3] M. Schneider: *Modellbildung, Simulation und nichtlineare Regelung elastischer, hydraulisch angetriebener Großmanipulatoren*. Doktorarbeit, Gerhard-Mercator-Universität - GH Duisburg, 1999.
- [4] D. Nissing: *Identifikation, Regelung und Beobachterauslegung für elastische Großhandhabungssysteme*. Fortschritt-Berichte VDI: Reihe 8, Mess-, Steuerungs- und Regelungstechnik. VDI-Verlag, 2002.
- [5] D. Schramm, D. Franitza und W. Lalo: *Virtual Prototyping and Real-Time Simulation of Heavy Equipment Manipulators including Elastic Deformations and Hydraulics*. ISARC, Ferrara, Italy 2005.
- [6] R. Schwertassek und O. Wallrapp: *Dynamik flexibler Mehrkörpersysteme*. Vieweg, 1999.
- [7] R.R.Craig und M.C.C. Bampton: *Coupling of Substructures for Dynamic Analyses*. AIAA Journal, 6:1313-1319, 1968.
- [8] O. Wallrapp: *Standardization of Flexible Body Modeling in Multibody System Codes, Part I: Definition of Standard Input Data*. Mechanics of Structures and Machines, 22, 1994.
- [9] Topometric GmbH: *Simulation des Bauteilverhaltens*, QZ Carl Hanser Verlag, München, 2015.

Modelle zur Beschreibung des Schwingungsverhaltens piezoelektrischer Aktoren am Beispiel des Wanderwellenmotors

Jürgen Krome

Angewandte Mechatronik, Hochschule Hamm-Lippstadt

juergen.krome@hshl.de

Piezoelektrische Materialien werden heute in vielen Anwendungen als Sensoren und Aktoren eingesetzt. Man nutzt dabei den physikalischen Zusammenhang zwischen einer von außen aufgebracht mechanischen Kraft und der elektrischen Ladungsverschiebung innerhalb des Kristallgitters. Mit diesem direkten piezoelektrischen Effekt lassen sich Sensoren höchster Genauigkeit erstellen. Auch der inverse piezoelektrische Effekt lässt sich nutzen, in dem man durch Anlegen einer äußeren elektrischen Spannung eine Verformung des Bauteils erhält. Dieser Effekt wird bei Aktoren und Ultraschallwandlern genutzt. Analytisch lässt sich der inverse und direkte piezoelektrische Effekt durch lineare Zustandsgleichungen beschreiben. Dabei sind die elektrische Flussdichte D , die elektrische Feldstärke E , die mechanische Dehnung S und die mechanische Spannung T miteinander verknüpft. Es handelt sich also um gekoppelte Feldprobleme, die auch mit modernen FE-Systemen beschrieben werden können. Im Rahmen der Auslegung neuartiger Sensoren und Aktoren können so komplexe Modelle genutzt werden, um das statische und dynamische Verhalten der Systeme zu untersuchen. In diesem Beitrag wird die analytische Beschreibung des piezoelektrischen Effekts und deren Abbildung mithilfe der Methode der finiten Elemente vorgestellt. Am Beispiel des piezoelektrischen Wanderwellenmotors werden Simulationen des Schwingungsverhaltens der Statoren vorgestellt. Aus den Ergebnissen der kontinuumsmechanischen Modelle werden Parameter abgeleitet, die anschließend in einem diskreten Modell zur Beschreibung des Stators genutzt werden.

1 Einleitung

Neben pneumatischen und hydraulischen Aktoren haben sich elektromechanische Aktoren stark verbreitet. Dabei wird stets elektrische Energie in mechanische Energie umgewandelt, indem man ausgehend von einer elektrischen Spannung oder einem elektrischem Strom das entstehende magnetische Feld, das entstehende elektrische Feld oder auch ein Wärme-feld nutzt, um eine mechanische Ausgangsgröße zu erhalten.

Man kann nun die Aktoren weiter unterscheiden nach Wirkprinzipien, bei denen bewegliche Bauteile vorhanden sind (z.B. Elektromotor, Elektromagnet und Tauchspule) und sogenannten Festkörperaktoren, bei denen es nur zur Verformung eines Körpers kommt. Zu den Festkörperaktoren zählen z.B. der thermische Aktor, bei dem man die Wärmedehnung nutzt, die Formgedächtnislegierungen, bei denen man eine Phasenumwandlung im Kristallgitter nutzt und die piezoelektrischen Kristalle.

Piezoelektrische Werkstoffe werden oft als Schallwandler (z.B. Akustik, Medizintechnik) aber auch als mechanischer Antrieb eingesetzt. Da die Ausdehnung der Werkstoffe und damit ein möglicher Stellweg sehr

gering sind, werden oft Übersetzungen genutzt oder es werden resonante Schwingungen erzeugt, welche durch Reibkräfte auf ein bewegliches Bauteil übertragen werden.

Eine Möglichkeit zum Bau eines rotatorischen Antriebs besteht in der resonanten Anregung einer symmetrischen Kreisplatte. Entstehende Mikroschwingungen werden dann in geeigneter Weise auf einen Rotor übertragen. Der sogenannten Wanderwellenmotor ist sowohl von seiner Mechanik als von der piezoelektrischen Schwingungsanregung interessant und soll daher als Beispiel für die Modellierung eines piezoelektrischen Aktors dienen.

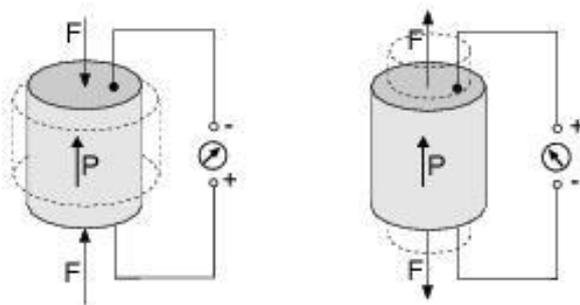
2 Der Piezoelektrische Effekt

Verformt man piezoelektrische Kristalle durch äußere mechanische Kräfte kommt es im Inneren des Kristallgitters zu einer Ladungsverschiebung, was wiederum zu einer Ladungskonzentration an den Oberflächen der Körper führt. Sind die auf den Oberflächen angebrachten Elektroden isoliert, entsteht im Inneren des Kristalls ein elektrisches Feld. Man nennt diesen von den Brüdern Curie entdeckten Effekt den direkten Piezoeffekt. Er lässt sich nutzen, um mit einem Sensor ein elektrisches Ausgangssignal in Abhängig-

keit einer Verformung oder Kraft zu messen (z.B. Beschleunigungssensor).

Die Umkehrung des Effekts, bei dem durch Anlegen eines äußeren elektrischen Feldes in Abhängigkeit der Feldrichtung eine mechanische Verformung des Kristalls entsteht, nennt man den indirekten Piezoeffekt. Man nutzt ihn für Aktoren in Lautsprechersystemen, Mikrodosierventilen, Einspritzventilen und Schwingungsantrieben. [1]

a) direkter Piezoeffekt



b) indirekter Piezoeffekt

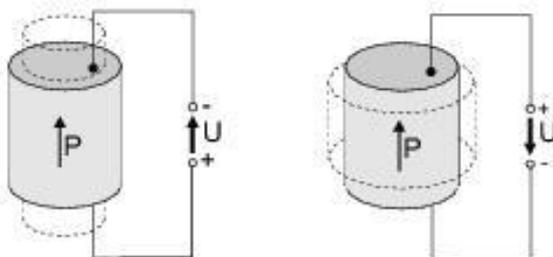


Abbildung 1: Direkter und indirekter Piezoeffekt

Begründet ist der piezoelektrische Effekt in dem Aufbau des Kristallgitters der Materialien. Unterhalb der sogenannten Curie-Temperatur geht das Kristallgitter aus energetischen Gründen vom kubischen in den tetragonalen Zustand über. Da nun die Ladungsschwerpunkte der Ionen nicht mehr übereinander liegen, entsteht in der einzelnen Elementarzelle ein elektrischer Dipol. Die einzelnen Kristalle sind jedoch beliebig in der Struktur verteilt, somit entsteht nach außen hin kein Dipol. Erst durch die künstliche Ausrichtung der Dipole mithilfe eines starken elektrischen Feldes in gewünschter Richtung lässt sich der gewünschte Effekt erzeugen.

Wird nun nach der Polarisierung des piezoelektrischen Materials erneut ein elektrisches Feld angelegt, kommt es zu einer reversiblen Ausrichtung der entstandenen Dipole, wodurch sich der Körper verformt.

(indirekter Piezoeffekt). Eine mechanische Verformung von außen führt dagegen zu der zuvor genannten Ladungsverschiebung und damit zur Ladungskonzentration an der Oberfläche.

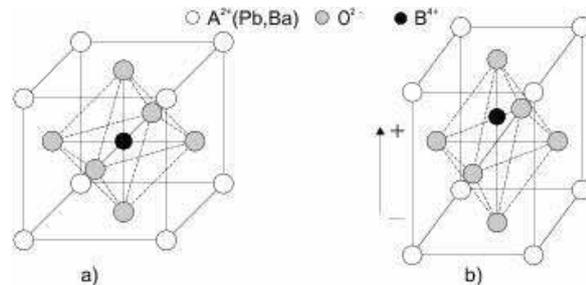


Abbildung 2: Elementarzelle piezoelektrischer Materialien oberhalb (a, kubisch) und unterhalb (b, tetragonal) der Curie-Temperatur

Obwohl das Verhalten piezoelektrischer Materialien oft nicht linear ist, werden zur Beschreibung überwiegend linearisierte Materialgesetze genutzt. Sie verknüpfen die elektrischen und mechanischen Größen und beschreiben das Verhalten der Werkstoffe für kleine Änderungen um einen nominalen Betriebspunkt.

Für den eindimensionalen Fall lassen sich die Zusammenhänge mit den folgenden Gleichungen beschreiben.

$$T = c^E S - e E \tag{1}$$

$$D = e S + \epsilon^S E \tag{2}$$

T ist dabei die mechanische Spannung (Tension), S die mechanische Dehnung (Strain), E die elektrische Feldstärke und D die dielektrische Verschiebung. Bekannt aus dem Hookeschen Gesetz ist der E-Modul, der sich hier für ein konstantes elektrisches Feld E in dem Parameter c^E wiederfindet. Weitere Materialparameter sind die piezoelektrische Spannungskonstante e und die Dielektrizitätskonstante ϵ^S bei konstanter mechanischer Dehnung S .

3 Der Ultraschall-Wanderwellenmotor

Kernstück des Ultraschall-Wanderwellenmotors ist eine kreissymmetrische Platte, an deren Unterseite piezoelektrische Keramiken aufgeklebt sind. Die Keramiken sind derart polarisiert und angeordnet, dass sich zwei orthogonale Eigenschwingungsformen resonant anregen lassen. Diese Schwingung führt zu elliptischen Bewegungen an der Oberfläche der Scheibe, mit welchen ein Rotor reibschlüssig angetrieben wird. Derartige Motoren sind z.B. in Autofocus-Systemen im Einsatz.

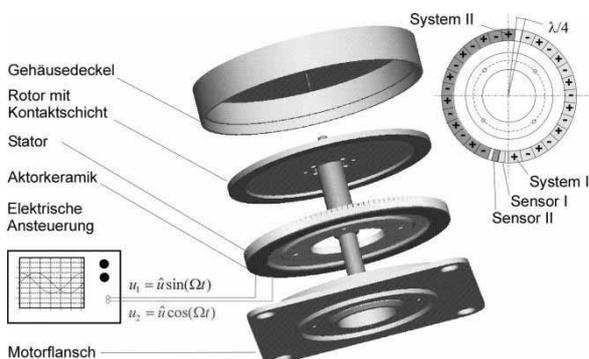


Abbildung 3: Aufbau des Wanderwellenmotors

3.1 Eigenschwingungsformen der Kreisplatte und die Entstehung der Wanderwelle

Das dynamische Verhalten der Kreisplatte lässt sich mit der Kirchhoffschen Plattentheorie näherungsweise beschreiben, vorausgesetzt das Verhältnis der Plattendicke zur Wellenlänge der Schwingungsform ist klein, die Auslenkung w der Mittelebene ist klein gegenüber der als konstant angenommenen Dicke und die Ebenen senkrecht zur Mittelebene bleiben auch im verformten Zustand senkrecht zur Mittelebene.

Bezeichnet man die Auslenkung der Plattenmittelebene mit $w(r, \varphi, t)$ an den Punkten r und φ zurzeit t , so lautet die Bewegungsgleichung der Platte:

$$D\nabla^4 w(r, \varphi, t) + \rho h \ddot{w}(r, \varphi, t) = q(r, \varphi, t), \quad (3)$$

wobei D die Biegesteifigkeit der Platte und ∇^2 der Laplace-Operator sind. Für den Fall der freien Schwingung ist die rechte Seite der Bewegungsgleichung identisch Null und es ergeben sich Lösungen der Form:

$$w(r, \varphi, t) = R(r) \cos(m\varphi) \cos(\omega t), \quad (4)$$

mit $m = 0, 1, 2, \dots$. Die Eigenkreisfrequenzen ω und die Amplitudenfunktion $R(r)$ ergeben sich aus dem entsprechenden Eigenwertproblem. Die Eigenschwingungsformen der Kreisplatte ergeben sich zu

$$W(r, \varphi) = R(r) \cos(m\varphi) \quad (5)$$

Die in den Eigenschwingungsformen auftretenden Linien, an denen die Auslenkung verschwindet, heißen Knotenkreise, wenn sie auf einem konstanten Radius liegen und Knotengeraden, wenn Sie unter einem konstanten Winkel liegen, (Abbildung 4).

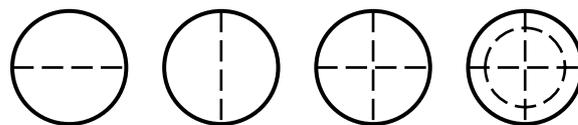


Abbildung 4: Knotengeraden und Knotenkreise

Aufgrund der Kreissymmetrie treten alle Eigenschwingungsformen (außer die für $m = 0$) doppelt auf. D.h. für jede Eigenkreisfrequenz ω existieren zwei orthogonale Eigenschwingungsformen. Abbildung 3 zeigt die Eigenschwingungsform des Stators mit 11 Wellenlängen am Umfang bei 46,6 kHz. Beim Wanderwellenmotor nutzt man nun gerade diese Eigenschaft der Kreisplatte, indem man durch die Überlagerung von zwei stehenden Wellen eine umlaufende Welle erzeugt.

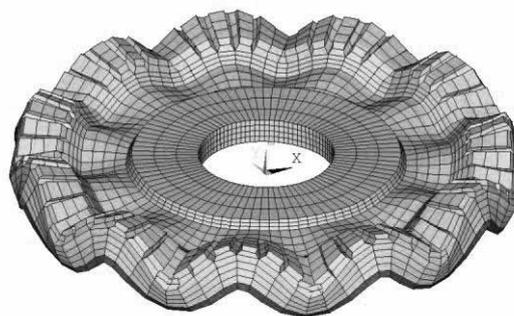


Abbildung 5: Eigenschwingungsform des Stators

Es lassen sich für die Bewegungsgleichung der Kreisplatte also zwei Lösungen der Form

$$w_1(r, \varphi, t) = AR(r) \cos(m\varphi) \cos(\omega t) \quad (6)$$

$$w_2(r, \varphi, t) = BR(r) \sin(m\varphi) \sin(\omega t + \alpha) \quad (7)$$

angeben. Es handelt sich dabei um zwei orthogonale linear unabhängige Stehwellen gleicher Frequenz. Aber auch die Überlagerung dieser beiden Wellen ist eine Lösung der Plattengleichung. Für den Fall $\alpha = 0$ und $|A| = |B|$ entsteht eine Wanderwelle der Form

$$w(r, \varphi, t) = AR(r) \cos(m\varphi - \omega t) \quad (8)$$

welche mit der Phasengeschwindigkeit ω/m in positive φ -Richtung umläuft.

Die Funktion $w(r, \varphi, t)$ beschreibt nun die Auslenkung der Plattenmittelebene senkrecht zur Mittelebene. Betrachtet man nun einen Punkt der Platte außerhalb der Ebene und nutzt wieder die Kirchhoffsche Plattentheorie, so kann man zeigen, dass die Punkte an der Oberfläche eine elliptische Bewegung ausführen. Eine zweite Kreisplatte die man mit entsprechender Vorspannung an die mit einer Resonanzfrequenz

angeregte Statorplatte presst, wird nun durch die auftretende tangentielle Bewegungskomponente der Oberflächenpunkte in Rotation versetzt.

Die resonante Anregung der beiden Stehwellen erfolgt über piezoelektrische Aktoren, die an der Unterseite der Kreisplatte aufgeklebt sind. Sie sind entsprechend der gewünschten Eigenschwingungsform polarisiert und um die erforderliche räumliche Phasenverschiebung verschoben angeordnet.

Eine genauere Herleitung der Wanderwelle und des Funktionsprinzips des Wanderwellenmotors findet sich in [1].

4 Theorie zur Finite-Elemente-Methode bei Piezoelektrischen Materialien

Wie in Abschnitt 2 bereits erläutert lässt sich das Materialverhalten piezoelektrischer Materialien näherungsweise durch die linearen konstitutiven Grundgleichungen beschreiben. Um komplexe Strukturen numerisch zu berechnen, teilt man sie in endlich viele Gebiete, die sogenannten Finiten Elemente ein. Diese Elemente werden an definierten Knotenpunkten miteinander verknüpft. Innerhalb der Elemente wird nun das Materialverhalten näherungsweise durch so genannte Formfunktionen beschrieben. An den diskreten Knotenpunkten dagegen erfolgt eine exakte Berechnung der elektrischen und mechanischen Größen.

Die Anwendung der Newtonschen Grundgleichung

$$\operatorname{div}(\mathbf{T}) = \rho \frac{\partial^2 \mathbf{u}}{\partial t^2} \quad (9)$$

und der Maxwell'schen Grundgleichungen

$$\operatorname{div}(\mathbf{D}) = \mathbf{0} \quad (10)$$

auf linearen konstitutiven Grundgleichungen und die zuvor genannte Diskretisierung des Gebietes führt dann schließlich auf das numerisch zu lösenden Differentialgleichungssystem.

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}} \\ \ddot{\boldsymbol{\varphi}} \end{bmatrix} + \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\boldsymbol{\varphi}} \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{uu} & \mathbf{K}_{u\varphi} \\ \mathbf{K}_{\varphi u} & \mathbf{K}_{\varphi\varphi} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\varphi} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{Q} \end{bmatrix} \quad (11)$$

Die Voraussetzung $\operatorname{div}(\mathbf{D}) = \mathbf{0}$ bedeutet dabei, dass das piezoelektrische Material isolierend ist, d.h. dass keine freien Volumenladungen vorhanden sind. In der Steifigkeitsmatrix \mathbf{K} wird die Kopplung zwischen den elektrischen und mechanischen Größen sichtbar. \mathbf{K}_{uu} ist die mechanische Steifigkeitsmatrix, $\mathbf{K}_{\varphi\varphi}$ ist die dielektrische Steifigkeitsmatrix und $\mathbf{K}_{u\varphi}$ die piezoelek-

trische Kopplungsmatrix. \mathbf{M} ist die Massenmatrix und \mathbf{C} die mechanische Dämpfungsmatrix des Systems. \mathbf{F} enthält die von außen wirkenden mechanischen Kräfte und \mathbf{Q} die aufgebrachten Ladungsver-schiebungen. An den Gleichungen ist sichtbar, dass die piezoelektrischen Finiten Elemente zusätzlich zu den mechanischen Verschiebungsfreiheitsgraden \mathbf{u} noch das elektrische Potential $\boldsymbol{\varphi}$ als Knotenvariable besitzen. [2]

Bei Vorgabe der elektrischen und mechanischen Rand- und Anfangsbedingungen können für diese Gleichungen numerische Lösungen gefunden werden. So kann z.B. eine statische Berechnung oder eine numerische Modalanalyse durchgeführt werden. Auch die Systemantwort auf eine harmonische Anregung lässt sich berechnen.

Im Rahmen der Auslegung des Stators eines Wanderwellenmotors wurden für verschiedene Geometrien und auch verschiedene elektrische und mechanische Randbedingungen die dynamischen Eigenschaften berechnet.

Der Stator des Wanderwellenmotors wird aus einer Messinglegierung mit geringer Dämpfung gefertigt. Für die Funktion benötigt man nun einen Außenring mit ausreichender Ausdehnung in Dickenrichtung, um eine genügend große Tangentialkomponente der Schwingung an der Oberfläche zu erreichen. Zur Reduktion der Biegesteifigkeit wird dieser Ring auf der Seite des Rotors häufig noch mit Nuten versehen. Zur mechanischen Befestigung und auch Aufnahme des entstehenden Drehmoments wird der Außenring über eine biege-weiche Entkopplungszone mit einem massiven Innenring verbunden. Abbildung 6 zeigt den Querschnitt einer Statorgeometrie.

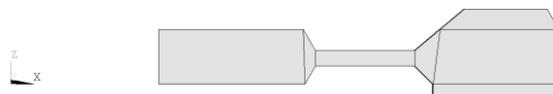
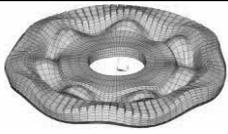
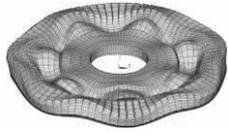
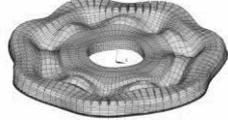
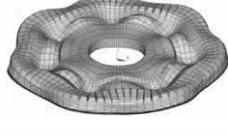
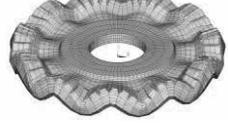
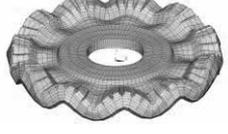
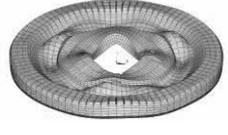


Abbildung 6: Eigenschwingungsform des Stators

An der dem Rotor gegenüberliegenden Seite werden schließlich die Aktorkeramiken verklebt. Sie sind entsprechend der gewünschten Schwingungsform abschnittsweise abwechselnd polarisiert und über eine gemeinsame Elektrode elektrisch miteinander verbunden. Da nun zwei Eigenschwingungsformen angeregt werden sollen, steht für jede Eigenschwingungsform nur der halbe Umfang zur Verfügung. Entsprechend der Orthogonalität der Eigenschwingungsformen, müssen die beiden Anregesysteme um

eine viertel Wellenlänge gegeneinander verschoben werden. In der zwischen den Anregesystemen frei bleibenden Zone werden noch piezoelektrische Sensoren appliziert, um die Amplitude der beiden Stehwellen zu messen und damit die Wanderwelle zu regeln. Abbildung 3 zeigt die Anordnung der Piezokeramiken am Stator.

Tabelle 1. Ergebnisse der Modalanalyse

Nr.	Frequenz	Mode Shape
1	42702 Hz	
2	42782 Hz	
3	43479 Hz	
4	44561 Hz	
5	44773	
6	46604	
7	46920	
8	47885	

Ist die Geometrie des Stators definiert, wird der Stator am Innenring gelagert und eine numerische Modalanalyse durchgeführt. Zur Reduktion der Rechenzeit wird der zu untersuchende Frequenzbereich einge-

schränkt. Tabelle 1 zeigt die berechneten Eigenschwingungsformen und zugehörigen Eigenfrequenzen.

Sehr schön zu erkennen ist die Orthogonalität der Eigenschwingungsformen. Die Schwingungsformen 1 und 2, 4 und 5 sowie 6 und 7 sind jeweils um eine viertel Wellenlänge verschoben. Die Eigenfrequenzen der jeweiligen Moden weichen jedoch geringfügig voneinander ab. Dieses ist in der Symmetrieabweichung der Piezokeramiken begründet. Da sich die piezoelektrische Kopplung in den Keramiken auf die Biegesteifigkeit auswirkt, sind auch die resultierenden Eigenfrequenzen leicht unterschiedlich.

Wesentlich sind auch die elektrischen Randbedingungen an den Piezokeramiken. Es besteht die Möglichkeit, die durch Knotenkopplungen definierten Elektroden kurz zu schließen, also auf allen Elektroden das Potential 0 V zu definieren oder eine Elektrode nicht zu definieren. Abgebildet werden dadurch der Fall der elektrisch kurzgeschlossenen Elektroden oder eben die offenen Elektroden. In der Modalanalyse ergeben sich dadurch unterschiedliche Eigenfrequenzen des Systems. Für kurzgeschlossene Elektroden erhält man die sogenannte Serienresonanzfrequenz, sie entspricht der Resonanzfrequenz des gedämpften mechanischen Systems. Für offene Elektroden erhält man die Parallelresonanzfrequenz des Systems. Da die Differenz der Eigenfrequenzen proportional zur piezoelektrischen Kopplung des Systems ist, lässt sich daraus ein Kopplungsfaktor definieren, der ein Maß für die Wirksamkeit der Aktoren ist. Für Kopplungsfaktoren deutlich kleiner 1 ergibt er sich zu

$$k_{eff}^2 = 2 \frac{f_p - f_s}{f_s} \quad (12)$$

Ziel der numerischen Auslegung der Statoren ist nun ein möglichst großer Kopplungsfaktor, um möglichst viel elektrische Energie in mechanische Schwingungsenergie umzuwandeln.

5 Reduktion der kontinuumsmechanischen Modelle

Im Rahmen von Gesamtsystembetrachtungen des Wanderwellenmotors werden nun jedoch einfache Verhaltensmodelle benötigt, die das Klemmenverhalten des Stators als elektromechanisches Übertragungsglied abbilden. Notwendig ist also ein Ersatzmodell mit wenigen Freiheitsgraden, welches die beiden Schwingungsformen, die elektromechanische Kopplung und die auftretenden Symmetriestörungen

ausreichend abbildet. Die mechanischen Eigenschaften können durch idealisierte mechanische Elemente (Zweipole) und die elektrischen Eigenschaften entsprechend über elektrische Zweipole abgebildet werden. Unter Nutzung elektrisch-mechanischer Analogien ist es dann möglich, ein rein mechanisches Ersatzmodell aufzubauen. Dabei wird die Analogie genutzt, in der die elektrische Potentialdifferenz einer mechanischen Kraft und der elektrische Strom einer mechanischen Geschwindigkeit, bzw. die elektrische Ladung einer Verschiebung entspricht. Abbildung 7 zeigt das äquivalente mechanische Ersatzmodell eines piezoelektrischen Aktors in Resonanznähe. Das kapazitive Verhalten der Piezokeramik wird durch eine Feder mit dem Parameter $1/C_P$ und die elektrischen Verluste in der Keramik durch einen Dämpfer mit dem Parameter R_P beschrieben. Die mechanische Schwingung in Resonanznähe wird durch die Feder c_s , den Dämpfer d_s und die Masse m_s beschrieben. Die piezoelektrische Kopplung bildet hier ein idealer Übertrager in Form eines Hebels ab. Dasselbe Modell lässt sich unter Ausnutzung der elektromechanischen Analogien auch aus Kapazitäten für die Federn, aus ohmschen Widerständen für die Dämpfer, aus Induktivitäten für die Massen und aus einem idealen Transformator für den Hebel aufbauen. [3]

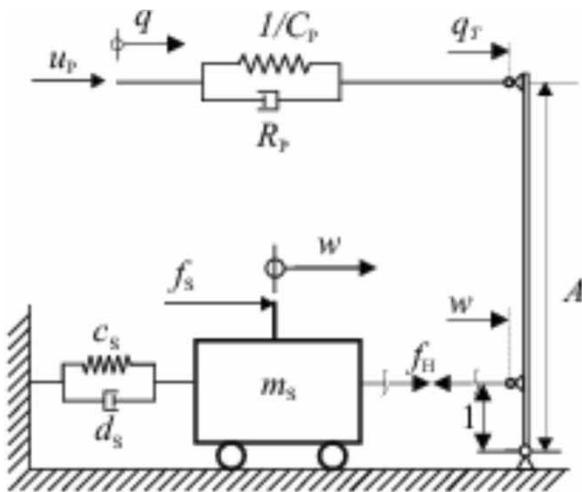


Abbildung 7: Ersatzmodell eines Piezoaktors

Soll nun das dynamische Verhalten des Stators in der Nähe der Resonanzfrequenz der für die Anregung der Wanderwelle genutzten Eigenschwingungsformen abgebildet werden, so müssen zwei dieser Modelle genutzt werden. Die Querkopplung zwischen den beiden Schwingungsmoden aufgrund der Asymmetrie in der Aktorik, muss dann durch einen weiteren idea-

len Übertrager (Hebel) abgebildet werden. Abbildung 8 zeigt das entsprechend entwickelte Modell.

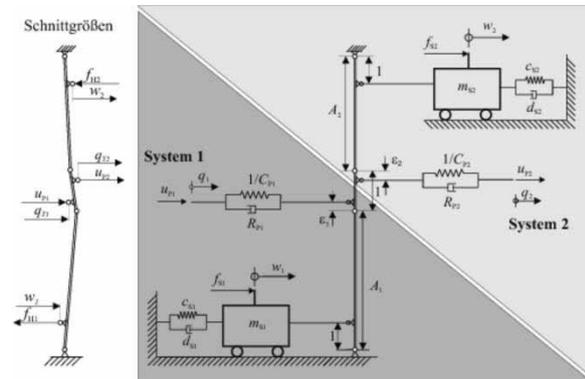


Abbildung 8: Ersatzmodell des Stators

Die zuvor erläuterten Parameter tauchen nun für jede Stehwelle auf. Die Querkopplung zwischen den Systemen ist durch die Hebellängen ε beschrieben. Für das gezeigte Modell lassen sich nun die Bewegungsgleichungen und damit auch die Übertragungsfunktionen aufstellen. Damit das Systemverhalten in dem geforderten Frequenzbereich genügend genau beschrieben wird, müssen die Modellparameter ausreichend genau bestimmt werden. Eine Möglichkeit ist die experimentelle Ermittlung der Parameter am schwingenden Stator. Dabei können aus den gemessenen Kurzschlussadmittanzen charakteristische Stützstellen ermittelt werden und anschließend die Modellparameter berechnet werden. Charakteristische Parameter sind z.B. die Serienresonanz- und die Parallelresonanzfrequenzen des Systems.

Eine weitere Möglichkeit die diskreten Ersatzparameter des Modells zu bestimmen besteht in der numerischen Simulation mit den kontinuumsmechanischen FE-Modellen. Dieses ist vor allem dann erforderlich, wenn im Rahmen der Auslegung des Motors noch keine Prototypen zur Verfügung stehen.

Die modale Masse des Statormodells m_s lässt sich z.B. aus der Lösung des Eigenwertproblems bestimmen. Die Massenmatrix des FE-Modells enthält die Massenverteilung des Bauteils, die Lösung des Eigenwertproblems liefert die Eigenfrequenz des Systems und die Eigenschwingungsform mit dem Verschiebungsfeld. Normiert man nun die Eigenvektoren so, dass die maximale Verschiebung identisch 1 ist und nimmt an, dass die Auslenkung w_i des diskreten Modells der Auslenkung der Eigenschwingungsform an dieser Stelle entspricht, so lässt sich durch Multi-

plikation der Eigenvektoren mit der Massenmatrix die effektive Masse bestimmen, Gleichung (13).

$$m_S = \Phi^T M \Phi \quad (13)$$

Die Ersatzsteifigkeiten c_S lassen sich nun aus den berechneten Eigenfrequenzen bestimmen, Gleichung (14)

$$c_S = m_S \omega_r^2 \quad (14)$$

Die Ermittlung der Koppelfaktoren zur Beschreibung der piezoelektrischen Kopplung kann schließlich durch entsprechende statische Berechnungen erfolgen.

Unter Nutzung des hier beschriebenen diskreten Ersatzmodells kann nun ein Gesamtmodell des Wanderwellenmotors erstellt werden. Ergänzt um ein Modell des Rotors und ein nichtlineares Kontaktmodell zur Beschreibung des Stator-Rotorkontakts, kann das Systemverhalten von der Ansteuerung bis zum Abtriebsdrehmoment simuliert werden.

6 Zusammenfassung

Piezoelektrische Aktoren werden in vielen Anwendungen als Sensoren und Aktoren eingesetzt. Im Rahmen der Auslegung und auch für die Regelung ist es erforderlich, ausreichend genaue Modelle zur Beschreibung des Systemverhaltens zu entwickeln. Der piezoelektrische Effekt beschreibt den Zusammenhang zwischen der Verformung eines Kristalls und der im Inneren auftretenden dielektrischen Verschiebung. Die Anwendung der Newtonschen und Maxwell'schen Grundgleichungen auf die konstitutiven Materialgleichungen führt auf ein kontinuumsmechanisches Ersatzmodell, welches im Rahmen der Finite-Elemente-Analyse gelöst werden kann. Durch dieses sehr detaillierte Modell lässt sich das statische und dynamische Verhalten des Stators eines Wanderwellenmotors einschließlich der piezoelektrischen Kopplung gut abbilden.

Will man jedoch Systemsimulationen für den gesamten Wanderwellenmotor machen, müssen einfache diskrete Ersatzmodelle entwickelt werden. Unter Ausnutzung elektrisch-mechanischer Analogien wurde ein mechanisches Ersatzmodell entwickelt, welches das mechanische und elektrische Verhalten des Stators in der Nähe seiner Resonanzfrequenz genügend genau beschreibt. Die piezoelektrische Kopplung und auch die Kopplung zwischen zwei Systemen können damit abgebildet werden. Die Modellparameter sind durch experimentelle Untersuchungen mit

einem Prototyp, aber auch mit Hilfe des kontinuumsmechanischen Modells bestimmt worden.

7 References

- [1] Krome, Jürgen: Modelle zur Untersuchung des Schwingungsverhaltens von Statoren für piezoelektrische Ultraschall-Wanderwellenmotoren, HNI-Verlagsschriftenreihe, Bd. 59; Mechatronik und Dynamik; Herausgeber: Wallaschek, Jörg, ISBN 3-931466-58-2, 1999
- [2] Allik, H., Hughes, J.: Finite Element for piezoelectric Vibration. International Journal for Numerical Methods of Engineering. Nr. 2, S. 151-157, 1970
- [3] Krome, Jürgen, Wallaschek, Jörg: Reduktion kontinuumsmechanischer Modelle im Rahmen der Optimierung mechatronischer Systeme am Beispiel des Wanderwellenmotors. VDI-Bericht 1315 „Mechatronik im Maschinen- und Fahrzeugbau“, S. 75-90, 1997

On the Methodology of Cooperative and Multirate Simulation

Irene Hafner¹, Niki Popper¹, Felix Breiteneker²

¹dwh GmbH, Neustiftgasse 57-59, 1070 Wien

²Technische Universität Wien, Wiedner Hauptstraße 8-10, 1040 Wien

irene.hafner@dwh.at

The term *co-simulation* has become more widely known in the last decade as the need for integrative simulations of complex systems is steadily increasing. This paper gives an overview of common co-simulation and multirate algorithms and tries to take up the advantages, risks and restrictions of certain methods as well as challenges to be expected when coupling simulations.

At first, the terms multirate simulation and cooperative simulation are explained in detail and several coupling methods are described. The next section covers approaches to examine and improve stability issues. Finally, reasonable fields of application for cooperative simulation are considered and ideas for further coupling methods are proposed.

Although this is a quite theoretical field of study, considerations show its practical usability for complex systems like for example production facilities and the increase of energy efficiency by the application of an integrative, predictive simulation.

1 Introduction

In times of increasing complexity of systems aimed to be modelled and simulated, it has become more and more important to be able to simulate models with partial models of different complexity and differing requirements regarding solver algorithms step sizes and other model-specific properties. To meet these requirements, models of such complexity are approached via cooperative or multirate simulation algorithms.

2 Co-Simulation - State of the Art

Cooperative simulation (abbreviated co-simulation), in literature also called modular simulation, distributed time integration or simulator coupling, stands for the coupled simulation of at least two partial systems. This can be realized via different simulators, solver algorithms or at least individual solver step sizes. Every co-simulation with more than one rate

of step sizes, i.e. at least two participating systems use different, individual step sizes for the calculation of internal values, is called multirate simulation. The crucial part of multirate co-simulation is the approximation of accuracy loss resulting from the extrapolation in between synchronization references.

In general, co-simulation can be divided into loose coupling and strong coupling methods depending on the interdependencies between the partial systems. For further inspections, let us consider a system divided into two partial systems described by (1):

$$\begin{aligned} \dot{y}_1 &= f_L(y_1, y_2, t), & y_1(t_0) &= y_{1,0} \\ \dot{y}_2 &= f_A(y_1, y_2, t), & y_2(t_0) &= y_{2,0} \end{aligned} \quad (1)$$

2.1 Strong Coupling Methods

If high accuracy is required but co-simulation is necessary not due to highly differing time constants but differing modelling approaches and requirements regarding the implementation, strong coupling will be considered. Using strongly coupled algorithms, solutions are iterated in every step until given error tolerances

for the overall system are fulfilled. This also means that all participating systems have to set the same time steps throughout the whole simulation. These itera-

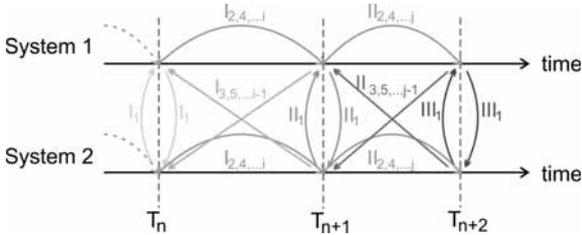


Figure 1: Illustration of the data exchange between two strongly coupled simulation algorithms.

tions allow high simulation accuracy but on the other hand lead to huge computational costs.

2.2 Loose Coupling Methods

Loose coupling co-simulation (in literature also referred to as weak coupling co-simulation or only co-simulation, see also [1]) means that the partial systems are integrated separately with individual time steps and synchronization at given macro time steps, which implies that values from other systems have to be extrapolated (and sometimes interpolated) between two synchronization references. Loose coupling co-simulation enables a faster simulation of distributed systems in comparison to strong coupling co-simulation, but it only makes sense if the systems are only loosely coupled themselves, meaning minor dependencies on variables from other systems (see equation (2)) as in contrast to strong coupling methods, loose coupling algorithms are prone to error accumulation due to the required extrapolation.

$$\left\| \frac{\partial f_1}{\partial y_2} \right\| \ll \left\| \frac{\partial f_1}{\partial y_1} \right\| \text{ and } \left\| \frac{\partial f_2}{\partial y_1} \right\| \ll \left\| \frac{\partial f_2}{\partial y_2} \right\| \quad (2)$$

Within algorithms for loose coupling, further distinctions can be made.

2.2.1 Gauß-Seidl Type Methods

Gauß-Seidl Type methods require sequential executions of partial systems. One system is integrated with its individual time step for one macro time step using

extrapolated values for variables from other systems. The next system to be executed can then apply interpolation to gain the values from the first system and so on until all systems have been simulated for the current macro time step. The same procedure is followed for all subsequent macro time steps. The stability of

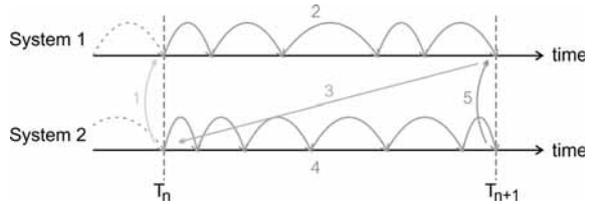


Figure 2: Sequential execution and data exchange for two Gauß-Seidl type coupled systems during one macro step.

Gauß-Seidl Type methods depends highly on the execution order of the participating subsystems.

2.2.2 Jacobi Type Methods

Applying algorithms of so-called Jacobi Type, all participating systems are simulated in parallel during each macro time step. At each synchronization reference, values needed from the respective other systems are exchanged and further extrapolated for usage in the next macro step. Between two synchronization references, every subsystem uses its own time step.

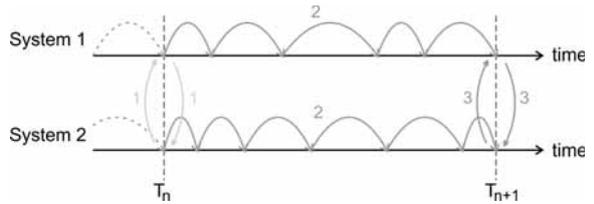


Figure 3: Communication between two systems loosely coupled according to the Jacobi type method.

2.2.3 Waveform Iteration Methods

So-called waveform iteration methods are an iterative enhancement of Gauß-Seidl Type methods. Again, all partial systems are simulated in parallel for one macro time step but before the next macro step follows, the partial systems are recalculated for the same step using values gained by interpolation from the other sys-

tems. This procedure is repeated until a given tolerance is obtained. Although in the first place this

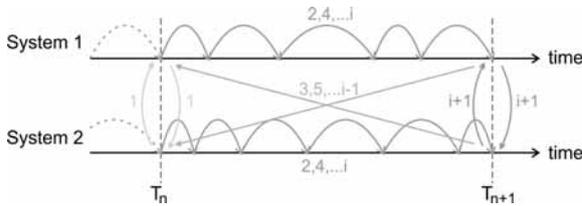


Figure 4: Waveform iteration method for two participating systems.

method seems to be very promising, it also holds a few difficulties. On the one hand, due to the iterations in every step, the execution takes a lot of computational time. On the other hand, in order to allow these iterations, not only does the overall algorithm (*master algorithm*) need to step back but the partial algorithms need to allow steps back and recalculations. The transfer of those newly calculated values to the master involves additional challenges, especially if different, maybe already prespecified software tools are combined.

2.2.4 Other

Furthermore, there are mixed methods not fitting into the structure above, like the asynchronous algorithm proposed by [2]. In this approach, subsystems do not necessarily have any time steps in common but are simulated sequentially. The next system to be simulated is determined by the smallest current simulation time. In the scenario above, both systems

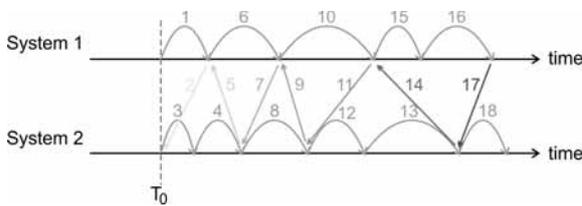


Figure 5: Illustration of an asynchronous coupling of two partial systems.

start at simulation time T_0 . Without loss of generality, system 1 is integrated first using interpolated values for variables from system 2. Now that system 1 has progressed one step and system 2 is still at T_0 ,

therefore at a smaller simulation time, system 2 is required to make a step, using interpolated (or, if required, extrapolated) values for variables from system 1. If, as in the case of the depicted scene, the simulation time in system 2 is still smaller than in system 1, the next step is calculated for system 2. As soon as the simulation time of system 2 has exceeded the simulation time of system 1, the next step is calculated for system 1 with its own chosen time step and so on. This method takes more time since there are no parallel executions and more data exchanges than in Jacobi or Gauß-Seidl type methods, but the results remain stable and more accurate.

3 Developments, Challenges and Risks

In addition to the definition of these coupling methods, they have to be investigated for the numerical effects they have on separately nicely working integration algorithms. Since the 1990s there has been a lot of research on the stability of multirate simulation. As the field of numerics of differential equations and differential algebraic equations itself is already a seemingly infinite area of research, the combination of different methods out of this area is even harder to investigate from a general point of view. Hence many studies on stability of coupling methods are done on systems with certain limitations (see for example [5]) and introducing methods for stabilization ([3, 4]).

In [3] it is shown that zero-stability can not be guaranteed for loose-coupling co-simulation if there exist algebraic loops between the participating subsystems. In addition, two iterative schemes are described which always guarantee stability. On the other hand, a method introducing a filter causing the elimination of algebraic loops and hence again warranting zero-stability is shown. However, since the second method modifies the system itself, it actually cannot be seen as a plain coupling method.

[5] shows that for sequential algorithms, the order in which the subsystems are executed is important so a contractivity condition necessary for stability is fulfilled.

There are also studies on the order of extrapolation

of values from other systems. To obtain higher accuracy, a higher extrapolation order may be used instead of constant extrapolation. However, this can lead to instabilities for loose coupling methods of Jacobi or Gauss-Seidl type, which can be coped with by stabilization as in [5] or combinative algorithms ([2]).

Apart from the fact that the restriction to cases with special requirements leaves a lot of unexploited methods for further investigations, the - reasonable - restriction to systems fulfilling certain requirements holds a few risks. There exist several software tools allowing the more or less easy coupling of certain simulators. Unfortunately, these are often used without further investigation on the consequences regarding numerical stability - like, for example, testing the system and used algorithms for the requirements necessary to guarantee stability.

4 Conclusion and Outlook

The present work has explained the basic principles of cooperative and multirate simulation with regard to methodology, consequences and risks. Although the research on simulator coupling goes back more than two decades, still a lot of questions remain unanswered and some of them even need to be posed.

One further aim of this work is the study of some of the open fields of research in this area such as coupling of highly contrastive simulation approaches, discrete and continuous partial models, further the processing of events, for example, the development of an algorithm which not only adapts the macro step size in consideration of previous steps but also allows steps back in time.

For systems with similar time constants and model properties (such as, for example, equal domains in physical models) but which might already modelled in different environments due to different people developing them, the integration of both systems in one tool should be considered for the sake of accuracy and speed instead of co-simulating them as a sometimes easier applicable, but more error-prone solution.

This implies that a means is intended to be established of determining applications where co-simulation - and which method exactly therein - is the most suited method but also scenarios where it may not be the best solution and other ways of satisfyingly simulat-

ing large systems have to be considered.

References

- [1] M. Busch. *Zur effizienten Kopplung von Simulationsprogrammen*, Kassel University Press, Germany, 2012.
- [2] S. Liang, H. Zhang and H. Wang. *Combinative Algorithms for the Multidisciplinary Collaborative Simulation of Complex Mechatronic Products Based on Major Step and Convergent Integration Step*. In: Chinese Journal of Mechanical Engineering, S. 355, 2011.
- [3] R. Kübler and W. Schiehlen. *Two Methods of Simulator Coupling*. In: Mathematical and Computer Modelling of Dynamical Systems, S. 93-113, 2000.
- [4] M. Arnold and M. Günther. *Preconditioned Dynamic Iteration for Coupled Differential-Algebraic Systems*. In: BIT Numerical Mathematics, S. 1-25, 2001.
- [5] M. Arnold. *Stability of Sequential Modular Time Integration Methods for Coupled Multibody System Models*. In: Journal of Computational and Nonlinear Dynamics, S. 031003, 2010.

Acknowledgement

This work is part of the *MeCoAMuCo* project funded by the Austrian Research Promotion Agency (FFG).

Komfortuntersuchungen an Gebäudeanlagen mittels gekoppelter Coarse-Grid-CFD- und Systemsimulation

Stefan Wischhusen¹, Timo Tumforde¹, Johannes Brunnemann¹
¹XRG Simulation GmbH, Harburger Schloßstraße 6-12, 21079 Hamburg

wischhusen@xrg-simulation.de

Der lokale Komfort in Gebäuden hängt zum Einen vom Aufbau der Räume und den verwendeten Materialien ab und zum Anderen von der verwendeten Anlagentechnik und deren Regelung. Eine direkt gekoppelte Simulation zwischen einem Coarse-Grid-CFD-Zonenmodell und einem numerisch effizienten Systemmodell der Anlagentechnik bietet die Möglichkeit, verschiedene Anlagenvarianten und deren Auswirkungen auf den Komfort innerhalb einer klimatisierten Zone zu untersuchen. Mit Hilfe von dynamischen Modelica-Modellen der HVAC Library [1] und dem neuen 3D Convection Modell aus der HumanComfort Library [2] können derartige Fragestellungen in einer einzigen Simulationsumgebung mit geringem Aufwand analysiert werden. Am Beispiel eines Großraumbüros wird ein konventionelles Klimatisierungssystem mit einem regenerativ betriebenen Kühlsystem und einem reinen Lüftungssystem verglichen und notwendige Änderungen zur Einhaltung des Komforts diskutiert.

1 Einleitung

Für die optimale Auslegung der Klimaanlage eines Gebäudes ist das ganzheitliche Verständnis von Anlage und auszurüstenden Räumlichkeiten erforderlich. Entsprechend der Nutzungsart, Fensterfläche, Sonnenausrichtung und Tageszeit fallen unterschiedliche Heiz- und Kühlbedarfe in den Räumen an, denen die Klimaanlage mit einem geeignet konditioniertem Luftmassenstrom begegnen muss. Während auf der Anlagenseite die für einen Ganzjahresbetrieb erforderlichen Leistungen und die geeignete Regelung bestimmt werden müssen, soll von Seiten der Nutzer die Behaglichkeit innerhalb des Raumes jederzeit sichergestellt sein. Dazu sind räumlich aufgelöste Daten über Temperaturen und Luftgeschwindigkeiten erforderlich, wie der vorliegende Artikel belegt. Eine Abbildung der klimatisierten Zone mit nur einem Kontrollvolumen und folglich einer gemittelten Temperatur erweist sich hingegen in Situationen mit hohen inneren Kühllasten als nicht ausreichend.

In diesem Beitrag wird das dynamische Modell

eines Gebäuderaumes und der dazugehörigen Klimaanlage vorgestellt. Dabei kommt ein von XRG Simulation neu entwickeltes Modelica-Raummodell zum Einsatz, das auf einer dreidimensionalen finiten Volumen-Diskretisierung der Navier-Stokes-Gleichungen basiert und thermische Strahlung, interne Wärmelasten sowie Festkörper im Raum berücksichtigt.

Dadurch wird es möglich, das gesamte System Raum-Luft-Klimaanlage-Wetter in einem einzigen Modelica Systemmodell abzubilden und dynamisch zu simulieren.

Anhand eines durch reale Wetterdaten vorgegebenen Beispielszenarios wird die dynamische Wechselwirkung der unterschiedlichen Systemteile veranschaulicht. Dabei sollen auch Vorteile und Limitationen des gewählten neuartigen Modellierungsansatzes diskutiert werden.

2 Coarse Grid CFD Modell

2.1 Eigenschaften

Das dem 3D-Convection-Modell zugrunde liegende Coarse Grid CFD Modell (siehe [9] für weitere Details) basiert auf einer finiten-Volumen-Diskretisierung der dreidimensionalen Navier-Stokes-Gleichungen auf einem kubischen Gitter [7, 8, 4] (siehe Abb. 1) und ist Teil der Human Comfort Library [2].

Das Modell ermöglicht die Simulation von Luftströmungen in komplexen Geometrien in räumlichen Auflösungen von typischerweise Dezimetern. Des Weiteren können symmetrische oder periodische Randbedingungen gesetzt werden.

Festkörper innerhalb der Geometrie können als physikalische Zellen oder als externe Zellen berücksichtigt werden: Physikalische Zellen berechnen ihre Innentemperatur aus einer Energiebilanz und wechselwirken mit der sie umgebenden Luft durch (konvektive) Wärmeströme, ihre Oberflächentemperaturen werden dynamisch aus einem Gleichgewicht verschiedener Wärmeströme bestimmt (siehe Abb. 3). Über externe Zellen können Temperaturen oder Wärmeströme als Randbedingungen gesetzt werden.

Im Coarse-Grid-CFD-Modell kann thermische Strahlung sowohl in vereinfachter Form, als auch detailliert (durch geometrische Sichtfaktoren) berücksichtigt werden.

Der Effekt turbulenter Strömung wird als effektive Modifikation der dynamischen Viskosität der Luft durch ein 0-Gleichungs-Modell abgebildet [10].

Massenströme können an jeder beliebige Festkörper-Luft-Grenzfläche in das Modell eingetragen werden, Wärmelasten in jede physikalische (Luft oder Festkörper) Zelle.

Neben trockener Luft können auch beliebige Multikomponenten-Gemische idealer Gase modelliert werden, z.B. feuchte Luft als eine Mischung aus trockener Luft und Wasserdampf. Die Massenanteile der einzelnen Bestandteile werden getrennt bilanziert. Zusammen mit einem durch Konzentrationsgradienten getriebenen Diffusionsmodell ermöglicht dies eine räumlich aufgelöste Analyse, z.B. des Wasser- und/oder des CO₂-Gehaltes der Luft.

Das Coarse-Grid-CFD-Modell ist vollständig in DY-MOLA [6] / Modelica [5] umgesetzt, wodurch es sich einfach in größere Systemmodelle einfügen lässt, die dann als Gesamtmodell simuliert werden können.

2.2 Diskretisierung

Jede Elementarzelle definiert ein räumliches Kontrollvolumen, über das die physikalischen Erhaltungsgrößen bilanziert werden. Die einer Erhaltungsgröße entsprechende dynamische Variable (Zustand) ergibt sich dabei aus der Integration bestimmter skalarer oder Vektorfelder über ein Kontrollvolumen und wird als in der Mitte des Kontrollvolumens lokalisiert angenommen. Die kubische Gestalt der Elementarzellen ermöglicht eine einfache Beschreibung der Zellgrenzen durch kartesische Koordinaten, siehe Abb. 1.

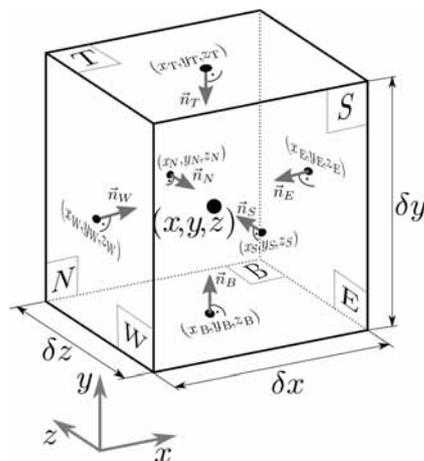


Abbildung 1: Diskretisierungsschema. Generische kubische Elementarzelle mit Nomenklatur. Die Oberflächen sind mit $\{\underline{East}, \underline{West}, \underline{Top}, \underline{Bottom}, \underline{North}, \underline{South}\}$ bezeichnet.

Zur Vermeidung bestimmter ansatzbedingter numerischer Probleme [7, 8] wurden 4 versetzte Gitter implementiert: ein Energie- und Massenbilanzgitter (e-cells), sowie jeweils ein Impulsbilanzgitter in (u,v,w)-Richtung (u,v,w-cells), siehe Abb. 2. Dementsprechend besitzt das Modell 4 Arten physikalischer Zellklassen. Zusätzlich gibt es Zellklassen zur Definition von Randbedingungen: adiabat, Temperatur, symmetrisch, periodisch.

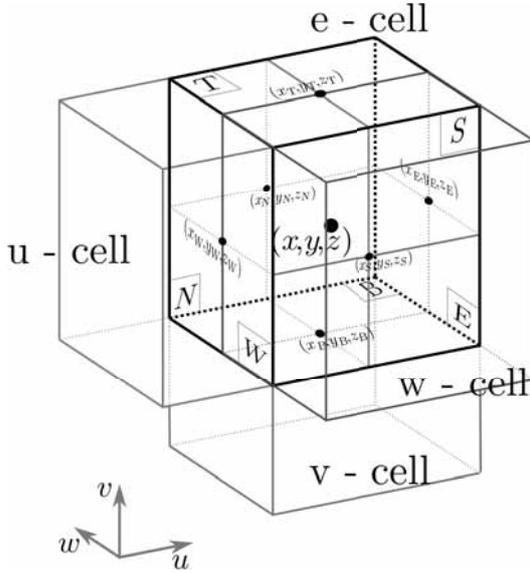


Abbildung 2: 4-fach versetztes Gitter Masse/Energie (e-cell), Impuls x -Richtung (u-cell), y -Richtung (v-cell), z -Richtung (w-cell).

2.3 Geometrie und modularer Modellaufbau

Die physikalische Modellierung ist modular aufgebaut: jeder Energiezelle wird ein Zelltyp zugeordnet. Die Zelle kann ein Luftvolumen, einen interner Festkörper (mit dynamischer Energiebilanz), ein externer Festkörper (mit von außen gesetzter Temperatur) oder ein adiabater Festkörper sein. Alle notwendigen Bilanzgleichungen, inklusive Mediendaten, sind im Energiezellenmodell enthalten und werden entsprechend des Zelltyps aktiviert. Komplexe Geometrien können so einfach durch ein Zelltypen-Parameterarray definiert werden, der globale Satz physikalischer Gleichungen für das Gesamtsystem wird vollautomatisch aus den lokalen Gleichungen der Energiezellen zusammengesetzt. Die Impulselementarzellen sind entsprechend konsistent aufgebaut: jede Impulszelle enthält einen universellen Satz von Impulserhaltungsgleichungen sowie Oberflächenwärmeübergangsmodelle (konvektiv, Wärmeleitung, Wärmestrahlung). Der Typ der Oberfläche (z.B. Luft-Luft, Luft-Festkörper, Festkörper-Festkörper) wird ebenfalls automatisch ermittelt und der passende Satz an Gleichungen ausgewählt.

2.4 Bilanzgleichungen

In der beschriebenen Diskretisierung ergibt sich die Massenbilanz als

$$\frac{d}{dt}m = \sum_{\alpha} \varepsilon_{\alpha} m_{flow,\alpha} \quad (1)$$

Dabei ist m die Masse des Kontrollvolumens, $\alpha \in \{E, W, T, B, N, S\}$ nummeriert die Oberflächen der Energie-Elementarzelle (siehe Abb. 1). Der Massenstrom durch die jeweilige Oberfläche ist mit $m_{flow,\alpha}$ bezeichnet. Der Faktor $\varepsilon_{\alpha} = \pm 1$ ist positiv/negativ, wenn der jeweilige Oberflächennormalenvektor parallel/antiparallel zu seiner entsprechenden Koordinatenrichtung ist. Für Festkörperzellen ist die Masse konstant.

Die Energiebilanz ergibt sich zu

$$\frac{d}{dt}(m c_V T) = \sum_{\alpha} \varepsilon_{\alpha} (H_{flow,\alpha} + Q_{flow,\alpha}) + Q_{flow,Source} \quad (2)$$

Dabei bezeichnen T die Zelltemperatur, c_V die spezifische Wärmekapazität bei konstantem Volumen, H_{flow} , Q_{flow} den konvektiven Enthalpiestrom bzw. den Wärmestrom durch Wärmeleitung. Durch $Q_{flow,Source}$ können Wärmelasten in das Kontrollvolumen eingebracht werden. Für Festkörperzellen ist $H_{flow,\alpha} = 0$, da keine Massenströme an den Grenzen dieser Zellen auftreten.

Das Coarse-Grid-CFD-Modell unterstützt beliebige Mehrkomponentengase. Für den Massenanteil $\Xi[i] = m[i]/m$ einer Gaskomponente i an der Gesamtgasmasse m einer Energiezelle ergibt sich:

$$\frac{d}{dt}\Xi[i] = \frac{1}{m} \left(\sum_{\alpha} \varepsilon_{\alpha} (\Xi[i]_{\alpha} - \Xi[i]) m_{flow,\alpha} + \sum_{\alpha} \varepsilon_{\alpha} (1 - \Xi[i]) m_{flow,diff,\alpha}[i] \right) \quad (3)$$

Hierbei ist $\Xi[i]_{\alpha}$ der Massenanteil an der Zellgrenze, $m_{flow,diff,\alpha}[i]$ ist der durch Diffusion verursachte Massenstrom der Komponente i an der Zelloberfläche α und wird in einem entsprechenden Diffusionsmodell in den Impulszellen aus den anliegenden Konzentrationsgradienten berechnet.

Die Finite-Volumen-Form der Impulsbilanz in den

Impulszellen ist gegeben durch:

$$\begin{aligned}
 \frac{d}{dt}(m c) = & m_{flow,Source} |c_{Source}| \\
 & + (p_{out} A_{out} - p_{in} A_{in}) + g_c m \\
 & + \sum_{\alpha} \varepsilon_{\alpha} (m_{flow,\alpha} c_{\alpha}) \\
 & + \mu_E \left. \frac{\partial c}{\partial x} \right|_E A_E - \mu_W \left. \frac{\partial c}{\partial x} \right|_W A_W \\
 & + \mu_T \left. \frac{\partial c}{\partial y} \right|_T A_T - \mu_B \left. \frac{\partial c}{\partial y} \right|_B A_B \\
 & + \mu_N \left. \frac{\partial c}{\partial z} \right|_N A_N - \mu_S \left. \frac{\partial c}{\partial z} \right|_S A_S ,
 \end{aligned} \quad (4)$$

dabei ist c der Platzhalter für eine der drei Geschwindigkeiten u, v, w in x, y, z -Richtung. Der erste Term auf der rechten Seite steht für eine Massenstromquelle. Die Flächeninhalt der Impulszellgrenzen ist durch A_{α} (z.B. A_E als Ost-Fläche) gegeben. Die Drücke in den angrenzenden Energiezellen, deren Zentrum genau auf den Außenflächen der jeweiligen Impulszelle liegen, sind p_{out}, p_{in} . Die positive Strömungsrichtung stimmt dabei mit der Koordinatenrichtung überein: "out" → "in" wie in Abb. 3 dargestellt. Die dynamische Viskosität ist durch $\mu = \mu_0 + \mu_{turb}(\rho, dr, c)$ gegeben, wobei μ_0 die dynamische Viskosität der ruhenden Luft ist und μ_{turb} eine Korrektur, die Turbulenzeffekte mittels eines 0-Gleichungs-Modelles berücksichtigt [10, 4, 9]. Der Anteil der Erdbeschleunigung in positiver Koordinatenrichtung der jeweiligen Geschwindigkeitskomponente $c = u, v, w$ ist durch g_c bezeichnet. An Festkörper-Luft Oberflächen sind alle Zeilen in (4) außer der ersten Zeile 0 gesetzt. An Festkörper-Festkörper - Grenzflächen ist die Strömungsgeschwindigkeit ebenfalls 0. Drücke werden in den Energiezellen aus der Zustandsgleichung idealer Gase berechnet:

$$p = \rho R_{spec} T . \quad (5)$$

Dabei ist ρ die Dichte des Gases in der Energiezelle und R_{spec} ist die spezifische Gaskonstante mit $R_{spec} = R/M$, wobei R die universelle Gaskonstante ist und M die molare Masse des Gasgemisches.

2.5 Wärmeübergang und Strahlung

An jeder Grenzfläche zwischen zwei Energiezellen wird die Oberflächentemperatur T_{surf} aus einer lokalen Energiebilanz errechnet (siehe Abb. 3), bei der Wärmeströme sowohl aus Wärmeleitung

($Q_{flow,out}, Q_{flow,in}$) als auch aus Wärmestrahlung ($Q_{flow,rad}$) berücksichtigt werden. Zur Berechnung

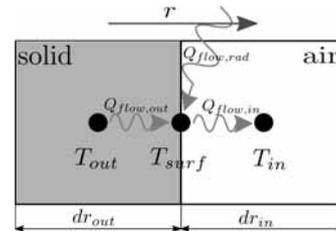


Abbildung 3: Wärmeübergang an Festkörper-Luft-Grenzflächen. Die Summe der Wärmeströme am Oberflächenknoten soll null sein. Für Luft-Luft Grenzflächen wird der durch Strahlung verursachte Wärmestrom $Q_{flow,rad}$ vernachlässigt. Hierbei ist r eine der drei Koordinatenrichtungen (x, y, z).

des Strahlungsaustausches zwischen zwei Festkörper-Luft-Grenzflächen werden geometrische Sichtfaktoren genutzt, um mögliche geometriebedingte Verschattungen von Oberflächen zu berücksichtigen. Der Einfachheit halber wird dabei die Strahlungsabsorption der Luft vernachlässigt.

2.6 Einbindung in Systemmodelle und Stand der Entwicklung

Die Zustandsgrößen $T, \rho, \Xi[i]$ für Energiezellen und u, v, w für Impulszellen gehen direkt in das Gleichungssystem des Gesamtsystemmodelles ein, in die das Coarse-Grid-CFD-Modell eingebettet ist. Für Berechnungen auf einzelnen CPU-Kernen ist gegenwärtig eine Größenordnung von ca. 15000 Zuständen das obere Limit für Modelica-Modelle in DYMOLA. Die Modelle der hier beschriebenen Szenarien haben rund 3300 Zustände, wovon rund 3100 Zustände auf das Coarse-Grid-CFD-Raummodell entfallen, das 520 Zellen besitzt und mit feuchter Luft als Gasmedium rechnet. Die durchschnittliche Rechenzeit für ein dynamisches 2-tägiges Szenario betrug rund 10h auf einem Kern eines Intel® Core™ i5 Prozessors mit 3.1 Ghz. Mit dem beschriebenen Coarse-Grid-CFD-Modell können Geometrien mit bis zu 2000 Zellen abgebildet werden. Dies erscheint wenig im Vergleich zu etablierten CFD-Programmen mit einigen Millionen Zellen. Das Coarse-Grid-CFD Modell ist daher kein Ersatz für im mm-cm-Bereich räumlich

aufgelöste CFD-Simulationen. Für viele Fragestellungen im Gebäudebereich ist die beschriebene räumliche Auflösung im dm-Bereich jedoch völlig ausreichend. Das Coarse-Grid-CFD Modell ermöglicht dazu kostengünstig die Erstellung räumlich aufgelöster, dynamischer Strömungssimulationen. Die erstellten Modelle lassen sich ohne großen Aufwand direkt in größere Gesamtsystemmodelle einbinden.

Die Verwendung eines kubischen Gitters ermöglicht eine einfache Geometriespezifikation, erfordert jedoch gegenwärtig die Rasterung bestehender Geometriestrukturen (Strukturen unterhalb der Auflösungsgrenze, schräge/runde Formen) auf kubische Grundformen. So sind beispielsweise in den Szenarien in Abschnitt 3 Personen und Sitze als gemeinsame Festkörperzellen mit gemittelten Materialeigenschaften modelliert.

Die automatisierte Erstellung grober Geometrien und die Erweiterung des Modelles auf schräge Oberflächen ist Gegenstand der gegenwärtigen Entwicklungsarbeit. Weiterhin steht die Anwendung von Techniken der Modellordnungsreduktion zur effektiven Erstellung größerer Gebäudemodelle im Fokus der Weiterentwicklung des Coarse-Grid-CFD-Modells.

3 Simulation eines Büroraumes

Das zuvor beschriebene CFD Coarse Grid Modell kann für verschiedenste Anwendungen eingesetzt werden. Neben der Simulation von klimatisierten Zonen im Gebäude ist selbstverständlich auch die Simulation von Fahrzeug- und Flugzeugkabinen [3] möglich.

In den nachfolgenden Abschnitten wird die Anwendung des CFD Modells für Komfortuntersuchungen im Gebäude näher betrachtet.

3.1 Beschreibung der Szenarien

Als Untersuchungsobjekt wurde ein Büroraum mit 6 Arbeitsplätzen und ca. 45 m² Fläche definiert (siehe Abbildung 4). Der zu kühlende Bereich des Gebäudes besteht aber aus 6 gleich großen Räumen, wobei nur ein Raum durch das CFD Modell abgebildet wird. Die Kühllast des Raumes wird durch einen Skalierungs-

faktor für den Luftstrom an das RLT-System weitergegeben. Somit ist die Kühlleistung der RLT-Anlage 6-mal so groß wie für den Einzelraum.

Der Einzelraum (siehe Abbildung 4) besitzt eine Südausrichtung mit einem 15 m² großen Fensteranteil. Das Büro ist tagsüber (8:00 - 17:00 Uhr) voll besetzt. Die inneren Wärmelasten werden in die entsprechenden Festkörperzellen eingetragen. Dabei wird die Wärmeabgabe einer jeden Person mit 80 W definiert sowie die Wärmeabgabe der PCs mit 150 W je Rechner. Wie am Ende von Abschnitt 2.6 beschrieben sind Personen und Sitze gemeinsam als effektive Festkörperzellen dargestellt, wobei die 80 W Wärmelast zu gleichen Teilen auf den oberen und den unteren Teil verteilt werden.

Die Lufteinlässe befinden sich auf Fußbodenniveau unterhalb des Fensters, der Luftauslass unterhalb der Decke rechts von der Tür.

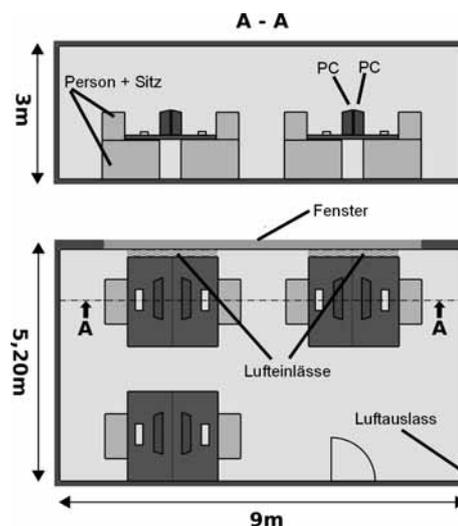


Abbildung 4: Schematische Abbildung des abgebildeten Büroraumes.

Als Randbedingungen werden zwei heiße Sommertage definiert, wobei der erste Tag lediglich zum Einschwingen der Gebäudetemperatur mit simuliert wird und nicht in die Auswertung einfließt. Die maximale Außentemperatur am betrachteten zweiten Tag beträgt ca. 32°C und die Einstrahlung beträgt maximal 865 W/m² horizontal (siehe Abbildung 5).

Es werden in den Simulationen nun verschiedene Systemkonfigurationen untersucht:

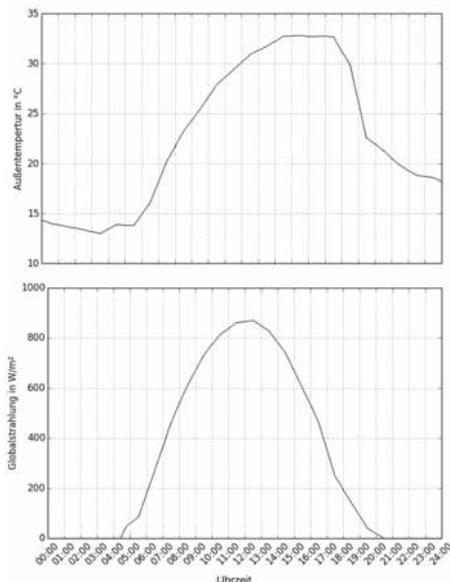


Abbildung 5: Temperatur- und Globalstrahlungsverlauf am betrachteten Tag

- **Szenario UOV:** Der ungekühlte aber belüftete Gebäudebereich **ohne** Verschattung
- **Szenario UMV:** Der ungekühlte aber belüftete Gebäudebereich **mit** Verschattung
- **Szenario EWS:** Der gekühlte und verschattete Gebäudebereich mit Erdwärmesondenkühlung
- **Szenario KONV:** Der konventionell gekühlte und verschattete Gebäudebereich

In jedem Fall wird angenommen, dass sich die Fenster nicht öffnen lassen, wie es etwa in vielen hohen Bürokomplexen der Fall ist.

Die Lüftung wird mit einem Volumenstrom entsprechend einer 3-fachen Luftwechselrate pro Stunde (ca. 420 m³/h) für den einzelnen Raum betrieben. Die Zulufttemperatur entspricht dabei entweder der veränderlichen Außentemperatur oder der Temperatur der konditionierten Zuluft. Das Lüftungssystem wird in jedem Szenario an das Raummodell angekoppelt und mit simuliert.

In den Szenarien mit Verschattung (**UMV**, **EWS**, **KONV**) wird an der Außenseite des Fensters eine automatisch schließende Jalousie mit bis zu 50%-iger

Verschattung angesetzt. Hierdurch verringert sich die Einstrahlung in den Raum erheblich.

Die Erdwärmesonde besteht aus zwei vertikalen Doppel-U-Rohren, welche Wärme über eine Wärmeträgerflüssigkeit an das Erdreich abgeben. Das Erdreich ist Teil des Modells und wurde mit einer ungestörten Temperatur von 12°C am Rande des betrachteten Erdvolumens versehen. Im sondennahen Bereich treten im Kühlbetrieb deutlich höhere Erdreichtemperaturen auf.

Im Szenario **KONV** ist anstelle der Erdwärmesonde ein Chillersystem mit Pufferspeicher integriert. Der Chiller hat eine Kühlleistung von 12 kW und arbeitet mit einer Vorlauftemperatur von 6°C.

3.2 Beschreibung der Simulationsmodelle

Das Modell des Großraumbüros ist in Abbildung 6 grafisch dargestellt.

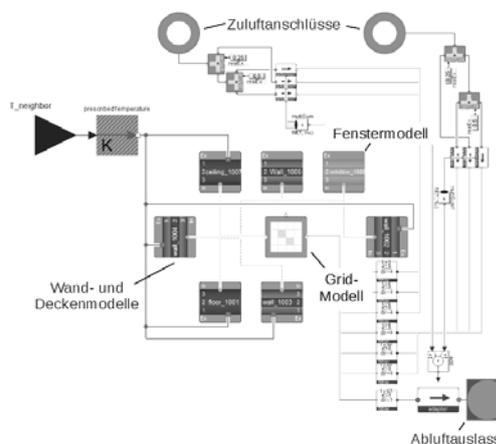


Abbildung 6: Grafische Darstellung des Büromodells

Das dreidimensional diskretisierte Raumgitter ist mit den entsprechenden Wand-, Fenster- und Boden- bzw. Deckenmodellen aus der HumanComfort Library verbunden. An den äußeren Flächen der innen liegenden Wände liegt eine frei wählbare Temperatur an, wodurch eine thermische Verbindung zu den Nachbarräumen des Büros abgebildet wird. Das Modell der südlich orientierten Außenwand sowie das Fenstermodell kommunizieren mit einem global verwen-

deten Wettermodell, welches die nötigen Informationen über die herrschenden Außenbedingungen bereit stellt. Die Zu- und Abluftanschlüsse des Raumes werden mit Komponenten der HVAC Library modelliert und über spezielle Adaptermodelle mit dem Raumgitter-Modell verbunden.

Das gezeigte Büromodell wird in den jeweiligen Systemmodellen der Anlagentechnik eingesetzt. In Abbildung 7 ist das Modell des Systems EWS dargestellt.

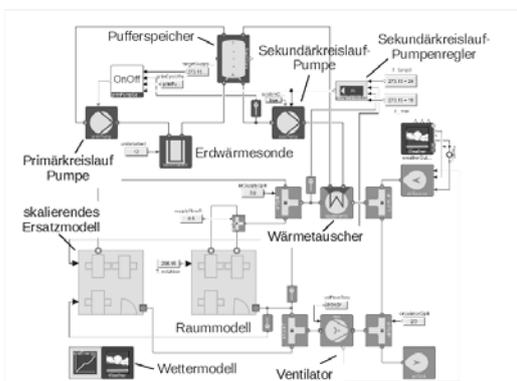


Abbildung 7: Grafische Darstellung des Anlagenmodells für das Szenario EWS

Die Kälteerzeugungsseite ist mit Modellen einer Erdwärmesonde, mit Primär- und Sekundärkreislauf-Pumpe und mit einem Pufferspeicher aus der HVAC Library aufgebaut. Die luftseitige RLT-Anlage mit Ventilator, Wärmetauscher, Luftverteilungssystem und Rezirkulationsleitung ist mit dem Grid verschaltet. Es ist zudem ein weiteres Modell im Luftpfad parallel zum dreidimensional aufgelösten Raummodell integriert, welches dessen Kühlbedarf entsprechend der gewählten Raumanzahl skaliert. Die Sekundärkreislauf-Pumpe wird entsprechend der angestrebten Zieltemperatur in der Abluft des Raumes und einer minimalen Zulufttemperatur geregelt. Die Primärkreislauf-Pumpe wird mit einer konstanten Drehzahl betrieben und über die Vorlauftemperatur des Sekundärkreislaufes An/Aus geregelt. Die Kälteerzeugungsseite ist mit Modellen einer Erdwärmesonde, mit Primär- und Sekundärkreislauf-Pumpe und mit einem Pufferspeichers aus der HVAC Library aufgebaut. Die luftseitige RLT-Anlage mit Ventilator, Wärmetauscher, Luftverteilungssystem und Rezirkulationsleitung ist mit dem Grid verschaltet. Es ist

zudem ein weiteres Modell im Luftpfad parallel zum dreidimensional aufgelösten Raummodell integriert, welches dessen Kühlbedarf entsprechend der gewählten Raumanzahl skaliert. Die Sekundärkreislauf-Pumpe wird entsprechend der angestrebten Zieltemperatur in der Abluft des Raumes und einer minimalen Zulufttemperatur geregelt. Die Primärkreislauf-Pumpe wird mit einer konstanten Drehzahl betrieben und über die Vorlauftemperatur des Sekundärkreislaufes An/Aus geregelt.

Im System KONV wird der Primärkreislauf der Kälteerzeugungsseite durch ein Modell eines konventionellen Chillers mit einer Zieltemperatur von 6°C ersetzt (siehe Abbildung 8). Der Chiller wird über die Vorlauftemperatur des Sekundärkreislaufes An/Aus geregelt.

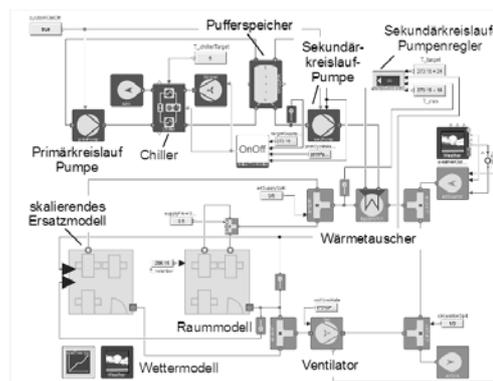


Abbildung 8: Grafische Darstellung des Anlagenmodells für das Szenario KONV

3.3 Auswertung und Diskussion

Im Folgenden werden die Simulationsergebnisse für die einzelnen Szenarien vorgestellt. Hierbei wird für die Betrachtung des Raumluftzustandes im Büroraum der Fokus auf die Verhältnisse um 14:00 Uhr des betrachteten Sommertages gelegt.

Zunächst soll das Szenario UOV (ungekühlte Raumlüftung ohne Verschattung) betrachtet werden. In Abbildung 9 sind die lokalen Raumlufttemperaturen für den bodennahen Bereich sowie für den vertikalen Schnitt an den fensternahen Arbeitsplätzen dargestellt.

Im gesamten Büro stellen sich Temperaturen von

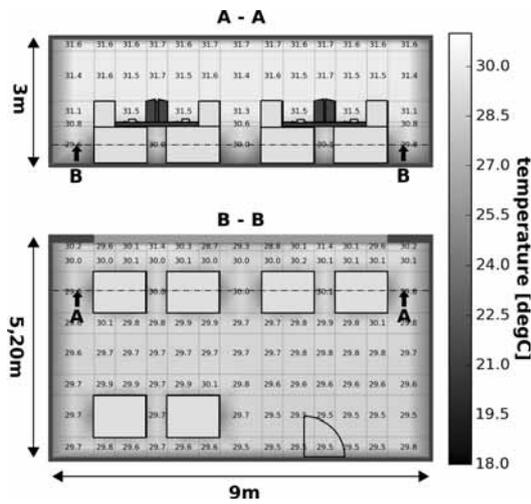


Abbildung 9: Lokale Raumlufttemperaturen für das Szenario UOV

über 30°C ein. Es wird also deutlich, dass zusätzliche Maßnahmen notwendig sind, um einen komfortablen Raumluftzustand zu erreichen. Eine Möglichkeit in dieser Hinsicht ist die Verwendung von Verschattungsmaßnahmen. Im Szenario UMV ist eine automatisch schließende Jalousie implementiert, die den direkten Einfall von Sonnenstrahlung in den Raum deutlich reduziert. Die mit ansonsten unveränderten Randbedingungen berechneten Raumtemperaturen sind in Abbildung 10 dargestellt.

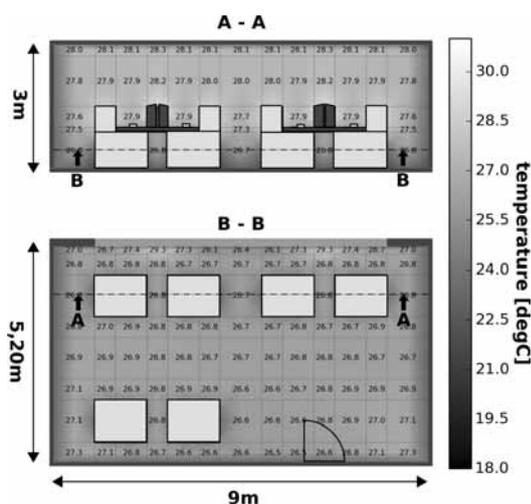


Abbildung 10: Lokale Raumlufttemperaturen für das Szenario UMV

Es zeigt sich, dass die Raumtemperaturen im Vergleich zum unverschatteten Fall auf einem deutlich niedrigerem Niveau angesiedelt sind. Allerdings handelt es sich noch nicht um ein Raumklima, das vom menschlichen Organismus als komfortabel betrachtet wird. Um einen solchen Zustand zu erreichen, sind aktive Kühlmaßnahmen nötig.

Im Szenario EWS wird die Zuluft des Raumes mithilfe eines freien Kühlsystems mit einer Erdwärmesonde gekühlt. Die zugehörige Temperaturverteilung ist in Abbildung 11 zu sehen.

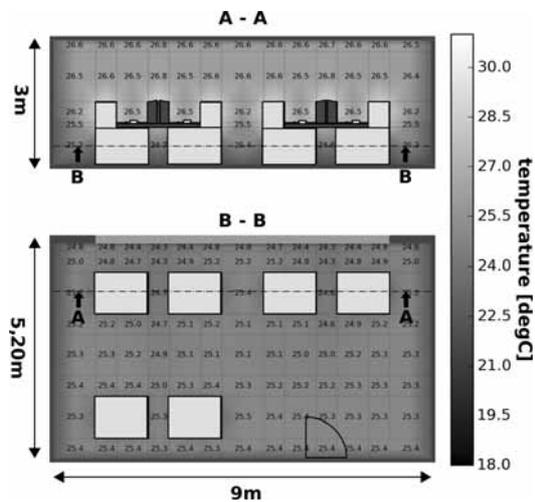


Abbildung 11: Lokale Raumlufttemperaturen für das Szenario EWS

Der Kühleffekt führt zu abermals niedrigeren Lufttemperaturen im Vergleich zu den vorherig untersuchten Szenarien. Allerdings zeigt sich, dass die Kapazität der Erdwärmesonde erreicht und überschritten wird und die Zulufttemperatur nicht hinlänglich gekühlt werden kann. Um eine weitere Absenkung der Raumluft zu realisieren, muss die Kühlleistung erhöht werden. Dies kann etwa durch das Einbringen einer zweiten Erdwärmesonde erlangt werden. Für diesen Fall ist in Abbildung 12 die dazugehörige Temperaturverteilung dargestellt.

Generell ist festzustellen, dass die zusätzlich bereitgestellte Kühlleistung dazu führt, dass die Raumluft durchschnittlich weiter abgesenkt werden kann. Da das Komfortbefinden eines Menschen allerdings nicht ausschließlich von der durchschnittlichen Raumluft abhängt, ist eine detailliertere Betrachtung nötig, um entscheiden zu können, ob auch eine Erhöhung des

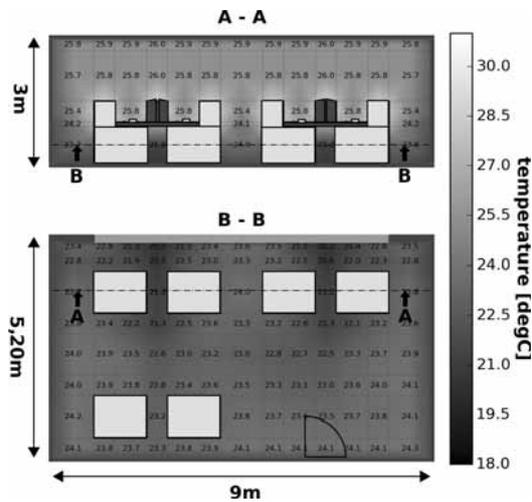


Abbildung 12: Lokale Raumlufttemperaturen für das Szenario EWS mit doppelter Sondenanzahl

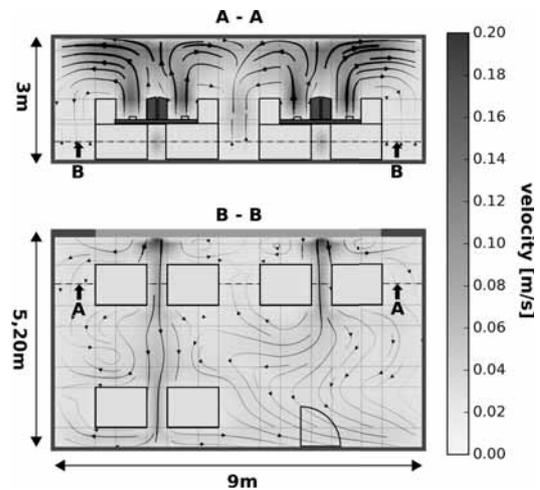


Abbildung 13: Lokale Luftgeschwindigkeiten für das Szenario EWS mit doppelter Sondenanzahl

Raumkomforts erzielt werden konnte. In diesem Zusammenhang zeigt sich der deutliche Mehrwert einer dreidimensionalen Auflösung im Vergleich zu einem herkömmlichen Einknoten-Ansatz, der nur eine Mischtemperatur für das gesamte Büro liefert. Ein relevanter Faktor im Komfortbefinden ist der vertikale Temperaturgradient. Dieser beträgt in der Nähe der fensternahen Arbeitsplätze in diesem Falle bis zu 5K, im vorherigen Falle mit einer deutlich höheren Temperatur in Fußbodennähe nur etwa 2K. Nach DIN EN ISO 7730 [15] führt eine vertikale Temperaturdifferenz von 5K zu einem Anteil von etwa 20 Prozent unzufriedener Individuen. Aus diesem Grunde gilt es zu hinterfragen, ob die weitere erzielte Temperaturabsenkung im Raum auch tatsächlich zu einer Erhöhung des Komforts führt. Eine solche Information lässt sich mit dem klassischen Einknoten-Ansatz nicht generieren.

Ein weiterer Mehrwert eines dreidimensionalen Ansatzes ist die Möglichkeit, Luftgeschwindigkeitsverhältnisse im berechneten Raum darzustellen. Abbildung 13 zeigt die lokale Geschwindigkeitsverteilung im Büro. Es zeigt sich, dass sich die Luftgeschwindigkeiten in einem Bereich befinden, der nicht als unangenehm wahrgenommen wird.

Abschließend soll Anhand des Szenarios KONV der Unterschied einer konventionellen Kühlmethode mit Chiller und einer freien Kühlung mit Erdwärmesonde erläutert werden. Im Szenario KONV wird die Erdwärmesonde durch einen konventionell arbeitenden

Chiller ersetzt. Die Temperaturverteilung für dieses Szenario ist in Abbildung 14 dargestellt.

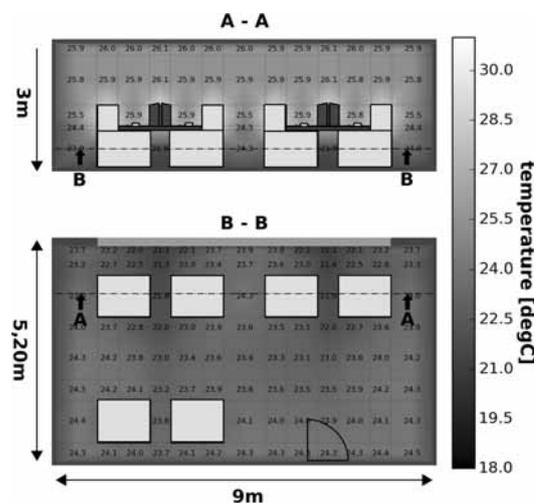


Abbildung 14: Lokale Raumlufttemperaturen für das Szenario KONV

Es zeigt sich, dass wie erwartet mit dem Chiller als Kältequelle die gleichen Ergebnisse erzielt werden können, wie mit den vorherig eingesetzten zwei Erdsonden. Ein Chiller kann mit einer deutlich kälteren Vorlauftemperatur betrieben werden als eine Erdwärmesonde, wodurch der benötigte Kältemittelmassenstrom zur Kühlung der Zuluft verringert oder auch der Luft-Kältemittelwärmetauscher kleiner dimensioniert

werden kann. Erkauft wird dieser Vorteil mit der benötigten elektrischen Leistungsaufnahme der Kompressoren. Hier zeigt sich der maßgebliche Unterschied im Betrieb zwischen einer freien und einer konventionellen Kühlung: Die elektrische Leistungsaufnahme der Primärkreispumpe in einem System mit Erdwärmesonde ist bei etwa gleicher Kühlleistung im Vergleich zum konventionellen Chiller-Kompressor maßgeblich kleiner. Die elektrischen Verbräuche und Kühlleistungen sind in Abbildung 15 im Tagesverlauf für die Szenarien KONV und EWS (mit zwei Erdsonden) abgebildet.

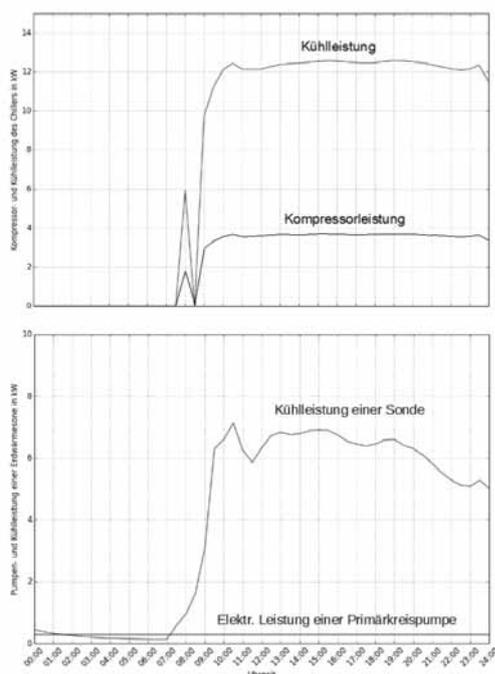


Abbildung 15: Elektrische Verbräuche und Kälteleistungen der verschiedenen Erzeuger

4 Zusammenfassung

Komfortuntersuchungen im Gebäude werden immer wichtiger, insbesondere in Bürogebäuden, die ebenfalls den sich verschärfenden Anforderungen der Energie-Einsparverordnung unterliegen. Hierdurch wird im Winter der Energiebedarf reduziert. Die erhöhte Isolierung führt jedoch zu einem immer größer werdenden Kühlbedarf im Sommer. Untersuchun-

gen zeigen, dass die Produktivität von Personen ab 15°C um 1-2% pro Kelvin Temperaturerhöhung abfällt [11], womit sich eine vorherige Untersuchung des Komforts mittels Simulationen auszahlen kann.

In diesem Beitrag wurden ein von XRG Simulation entwickelter Modellbaukasten, bestehend aus der HumanComfort LibraryTM und der HVAC Library, für Komfortuntersuchungen in Modelica vorgestellt und auf ein Gebäude mit mehreren Großraumbüros angewendet. Es wurde gezeigt, dass eine Simulation von Raummodell(en) und Anlagentechnik in einem Modell möglich ist. Im Unterschied zu anderen bekannten Ansätzen und Softwarepaketen (z.B., Energy+ [12], IDA ICE [13], Buildings Library [14]) erfolgt die Modellierung vollständig und transparent in Modelica. Die CFD-Simulation ist notwendig, da sogenannte 1-Knoten oder 1-Volumen-Ansätze zur Bilanzierung eines Raumes oder einer klimatisierten Zone keine Aussage über lokale Temperaturdifferenzen und lokale Luftgeschwindigkeiten erlauben.

Eine energetische Bilanzierung ist mit dem CFD-Modell ebenfalls möglich oder es kann aus dem 3D-Modell aufgrund der verwendeten Standardschnittstellen mit wenigen Arbeitsschritten ein numerisch hoch performantes 1D-Modell abgeleitet werden. Somit kann der Modellbaukasten für eine breite Palette von Aufgabenstellung in der Gebäudetechnik schnell und kostengünstig eingesetzt werden.

In der vorliegenden Arbeit konnte gezeigt werden, welche Auswirkungen die Charakteristika der verschiedenen Anlagentypen zur Kälteerzeugung auf das Raumklima haben. Des Weiteren wurde herausgestellt, dass auch lokale Aspekte wie die Anordnung von Zu- und Abluftgittern einen maßgebliche Einfluss auf den Komfort innerhalb eines Raumes besitzen, die nur mit einer dreidimensionalen Auflösung des Objektes erfasst werden können.

Die dabei auftretenden Fragestellungen lassen sich effektiv durch das hier vorgestellte Systemmodell untersuchen.

Literatur

- [1] HVAC Library 2.0.0. XRG Simulation GmbH, Deutschland, 2016.
<http://www.xrg-simulation.de/de/produkte/xrg-library/xrg-hvac-library>, zuletzt abgerufen am 05.02.2016.
- [2] HumanComfort Library™ 2.0.0. XRG Simulation GmbH, Deutschland, 2016.
<http://www.xrg-simulation.de/de/produkte/xrg-library/humancomfort>, zuletzt abgerufen am 05.02.2016.
- [3] B. Michaelsen, J. Brunnemann, J. Eiden, T. Giese, M. Schmid. *Dynamic Simulation of an Aircraft Compartment coupled to a Ventilation System* in Proceedings of the 5th International Workshop on Aircraft System Technologies, in Hamburg, Feb. 2015.
- [4] M. Bonvini and M. Popovac. “*Fluid flow modeling with Modelica*.” In 7th Vienna International Conference on Math. Modelling, Wien, Österreich, 2012.
- [5] Modelica Language Specification 3.3, Modelica Association, 2012.
<https://www.modelica.org/documents/ModelicaSpec33.pdf>, zuletzt abgerufen am 05.02.2016.
- [6] DYMOLA - Dynamic Modelling Laboratory, Version 2016 FD01 (64-bit), ©Dassault Systèmes, 2015.
<http://www.3ds.com/products-services/catia/capabilities/systems-engineering/modelica-systems-simulation/dymola/>, zuletzt abgerufen am 05.02.2016.
- [7] H.K. Versteeg and W. Malalasekera. “*An introduction to computational fluid dynamics, The finite volume method*”, Longman Scientific & Technical, 1995.
- [8] S.V. Patankar. “*Numerical Heat Transfer and Fluid Flow*”, Hemisphere Publishing Corporation, 1980.
- [9] A. Vasilj, J. Brunnemann, A. Wick. “*Zonal Modeling of Aircraft Compartments - Fluid Simulation in Modelica*”, conference contribution, Deutscher Luft- und Raumfahrtkongress Augsburg, 2014.
- [10] Q. Chen and W. Xu. “*A zero equation turbulence model for indoor airflow simulation*.” *Energy and Buildings*, 28(2): 137 - 144, 1998.
- [11] O. Seppänen, W. J. Fisk, Q. Lei. “*Effect of Temperature on Task Performance in Office Environment*”,
<https://indoor.lbl.gov/sites/all/files/lbnl-60946.pdf>, zuletzt abgerufen am 05.02.2016.
- [12] EnergyPlus™.
<https://energyplus.net>, zuletzt abgerufen am 05.02.2016.
- [13] IDA ICE 4.7. EQUA Simulation AB, Schweden, 2016.
- [14] Modelica Buildings Library 2.1.0. Lawrence Berkeley National Laboratory, USA, 2015.
- [15] DIN EN ISO 7730 (2003-10). *Ergonomie des Umgebungsklimas - Analytische Bestimmung und Interpretation der thermischen Behaglichkeit durch Berechnung des PMV- und des PPD-Indexes und der lokalen thermischen Behaglichkeit*. Berlin: Beuth

Anbindung eines aktiven Steuerkraftsystems an eine Echtzeitsimulation

Henrik Wüstenberg¹, Jürgen Gotschlich², Umut Durak²

¹Hochschule Ostfalia

²DLR Institut für Flugsystemtechnik

he.wuestenberg@ostfalia.de

{ juergen.gotschlich, umut.durak }@dlr.de

In der Flugsimulation werden aktive Steuerkraftsysteme häufig für Forschungszwecke genutzt. Ein Grund dafür ist die Verbesserung der realistischen Umgebung durch simulierte Steuerkräfte. Die aktiven Steuerorgane erzeugen diese Steuerkräfte, um dem Piloten das Gefühl eines realen Luftfahrzeugs zu vermitteln. Ein Steuerkraftsystem funktioniert grundsätzlich autonom. Jedoch bringt eine Anbindung an die Simulationsinfrastruktur Vorteile, indem Steuerkraftkonfigurationen flexibel geändert werden können.

Das Deutsche Zentrum für Luft- und Raumfahrt e. V. (DLR), Institut für Flugsystemtechnik, betreibt am Standort Braunschweig den Air VEHICLE Simulator (AVES). Es wird ein Airbus A320 Simulator und ein Eurocopter EC135 Simulator genutzt. Die Simulatoren werden über die im DLR entwickelte Simulationsinfrastruktur *2Simulate* verbunden, welche für eine deterministische Ausführung von Prozessen sorgt. Jeder Prozess erfüllt eine spezifische Aufgabe, wie z.B. Verbindungen zu externen Geräten oder auch die Berechnung dynamischer Modelle. Beide Simulatoren verwenden leistungsfähige aktive Steuerkraftsysteme der Fa. Wittenstein Aerospace & Simulation GmbH. Die jeweiligen Achsen werden nach einer Force-Feel Charakteristik geregelt. Im Umfang dieser Publikation wird die Schnittstelle zwischen Simulation und dem aktiven Steuerkraftsystem vorgestellt, die in *2Simulate* mittels eines WCLS-Tasks (Wittenstein Control Loading System) implementiert ist. Anhand einer Beispielanwendung werden Konfigurationsänderungen am Steuerkraftsystem demonstriert.

1 Einleitung

Aktive Steuerkraftsysteme werden in Flugsimulatoren häufig für Forschungszwecke genutzt. Eine steigende Anzahl Simulatoren wird mit diesen Systemen betrieben, da die taktile Wahrnehmung, als weiterer Kanal zum Piloten, ein großer Vorteil ist [1]. Sie verbessert das Bewusstsein des Piloten für eine Flugsituation und verstärkt die realistische Umgebung im Simulator.

Das DLR betreibt am Standort Braunschweig den AVES, einen Airbus A320 Simulator und einen Eurocopter EC135 Simulator. Diese Arbeit befasst sich mit dem aktiven Steuerkraftsystem im Hubschrauber Simulator [2]. Das Steuerkraftsystem trägt die Produktbezeichnung „Reconfigurable Helicopter Mission Trainer“ (RHMT). Dieser setzt sich hauptsächlich aus einer Regelungseinheit (SCM), mehreren Versorgungseinheiten (SPS) und den Aktuatoren für jede Achse zusammen [3]. Das Steuerkraftsystem wird im AVES grundsätzlich zur Forschung genutzt. Die Realitätsnähe der Flugsimulation wird hier durch Handling Quality Studien mit Piloten getestet. Dafür ist es entscheidend die hinterlegten Kraft-Positions-Kennlinien während der Simulation zu beeinflussen, um sie auf besondere Flugzustände anzupassen. Beispielsweise werden durch die Umströmung der Steuerflächen am Flugzeug Kräfte erzeugt, die bei einem mechanischen System entgegen den Steuereingaben des Piloten

wirken. Dafür simuliert das aktive Steuerkraftsystem diese Kräfte. Die Modifikationen an den Parametern der Regelung können durch eine graphische Benutzeroberfläche der Fa. Wittenstein Aerospace & Simulation GmbH, namentlich AktivToolkit, durchgeführt werden [4].

Diese Publikation befasst sich mit der Schnittstelle zwischen der Simulation und dem aktiven Steuerkraftsystem. Dafür wird die Simulationsinfrastruktur *2Simulate* des DLR verwendet, eine Software für verteilte Echtzeitsimulationsanwendungen. Diese Software stellt dem Anwender Prozesse zur Verfügung die spezifische Aufgaben erfüllen und bei der Programmentwicklung unterstützen. Die Schnittstelle zum Steuerkraftsystem wird durch einen *WCLS-Task* in *2Simulate* verwendet. Des Weiteren wird eine Beispielanwendung vorgestellt, die Veränderungen am Wittenstein System mithilfe der *WCLS-Task* Methoden durchführt. Die Funktionen des Prozesses übertragen Parameterwerte über ein spezifisches Daten-Transfer-Protokoll an das SCM des Steuerkraftsystems. Zur Programmentwicklung wird für die Beispielanwendung der Toolkit Protokoll Simulator der Fa. Wittenstein verwendet. Der Emulator verfügt über grundsätzliche Funktionen des aktiven Steuerkraftsystems, wie das Wechseln von Achsenzuständen [5].

Das zweite Kapitel umfasst die Idee aktiver Steuerkraftsysteme im Allgemeinen und geht auf Details des Wittenstein Systems ein. Im Anschluss be-

schreibt das dritte Kapitel den Aufbau des 2Simulate-Frameworks und erläutert den Wittenstein-Task. Das vierte Kapitel beschreibt die Beispielanwendung und erläutert ihre Funktionen. Ein Ausblick folgt im fünften Kapitel dieser Publikation.

2 Das aktive Steuerkraftsystem

Während ein Flugzeug fliegt, fließt eine Luftströmung über die Oberfläche und erzeugt einen Widerstand. Besonders an den Steuerflächen, also an den Rudern des Luftfahrzeugs, entsteht durch Reibung ein hoher nichtlinearer Widerstand. Dieser muss bei mechanischen Flugsteuerungen vom Piloten überwunden werden. Um diese Steuerkräfte in der Simulation zu realisieren, wird ein geregelter Antrieb benötigt. Dafür werden hydraulische oder elektronische Aktuatoren verwendet [7]. Das aktive Steuerkraftsystem der Wittenstein AG ist ein elektronisches System mit leistungsstarken Gleichstrommotoren. Diese Motoren geben eine Kraftrückmeldung in Abhängigkeit von einer Kraft-Positions-Kennlinie, wie in Abbildung 1 dargestellt. Somit wird die vom Piloten aufzubringende Kraft aus der Kennlinie und in Abhängigkeit von der Auslenkung des Steuerelements berechnet [3].

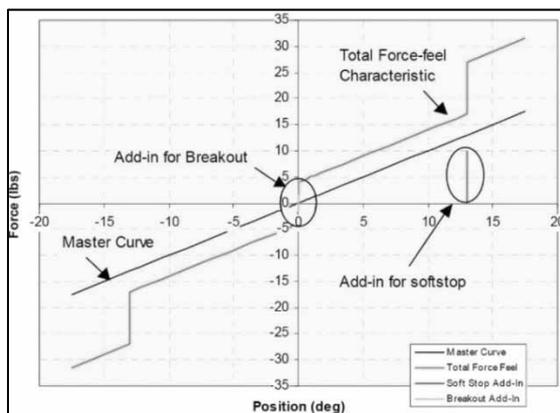


Abbildung 1 Kraft-Position Kennlinie [3]

Die flexiblen Einstellungen des Wittenstein Systems für die Kraft-Positions-Kennlinien werden durch zahlreiche Funktionen beim Konfigurieren umgesetzt. Beispielsweise kann die Grundkurve (Master-Curve) durch weitere zusätzliche Kurven (Addin-Curves) erweitert werden. Die Zusatzkurven ermöglichen Soft-Stop's, Breakout's und weitere besondere Rückmeldungen zum Piloten [4]. Abbildung 1 zeigt wie die Regelungssoftware des Steuerkraftsystems „Aktiv8“ die Kennlinie aus einigen Kurven zusammensetzt.

2.1 Die Regelungsstruktur

Die Entwicklung des RHMT basiert auf der Verwendung von „Commercial-Off-The-Shelf“ (COTS) Komponenten. Weswegen die Leistung der

einzelnen Geräte weitestgehend ausgeschöpft werden soll [7]. Dafür wurde von Wittenstein eine dezentrale Regelung des Systems entworfen, siehe Abbildung 2. Das SCM (System Control Module) dient neben der Regelung auch für die Kommunikation im Steuerkraftsystem. Dafür besitzt es mehrere Schnittstellen, wie Ethernet und CAN-Bus, zur Kommunikation mit Aktuatoren, den SPS oder anderen externen Geräten [3].

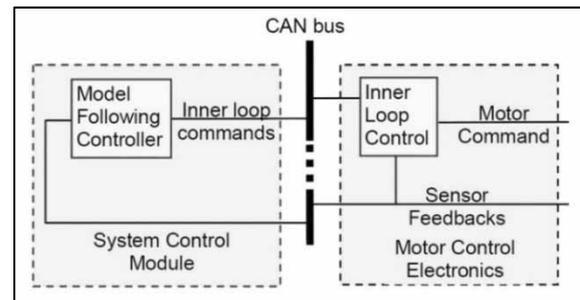


Abbildung 2 Regelungsstruktur [7]

Abbildung 2 zeigt im linken Bereich das SCM. Zur Regelung nimmt das SCM Positions- und Kraftmesswerte von der Motorregelungseinheit auf. Die Position entspricht der Auslenkung des Steuerorgans und die Kraft entspricht der aufgebrauchten Steuerkraft des Piloten. Der sogenannte Model-Following-Regler des SCM besitzt ein hinterlegtes dynamisches Modell über das das Antwortverhalten der einzelnen Aktuatoren berechnet wird. Eingangsgrößen der Regelung sind die Messwerte der Sensoren am Aktuator. Das Antwortverhalten wird folglich als Steuerbefehl über ein CAN-Bus Netzwerk an den entsprechenden Aktuator übertragen. Die innere Regelung dient hier als Steuerelektronik für den Gleichstrommotor und erzeugt entsprechend dem Antwortverhalten des Modells und der Kraft-Position Kennlinie ein Gegenmoment auf das Steuerelement [7].

2.2 Die Softwarestruktur

Das SCM betreibt die Regelungssoftware *Aktiv8*. Sie unterteilt sich in die drei Komponenten *Aktiv8 Control*, *Aktiv8 DLM* und *Configuration File Set*. *Aktiv8 Control* ist generell für die Regelung zuständig, siehe Kapitel 2.1. Die *Aktiv8 DLM*- und *Configuration File Set*-Komponenten speichern Dateien zwischen und stellen sie bei Bedarf der Regelung zur Verfügung [3].

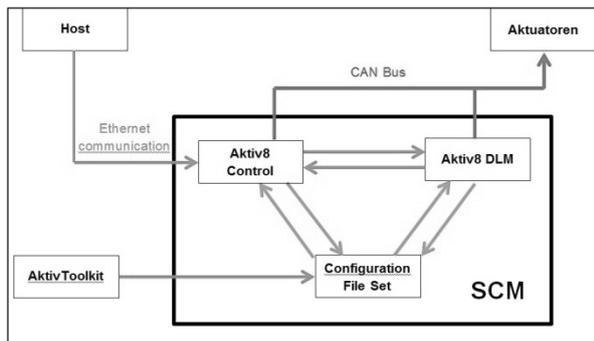


Abbildung 3 Die Wittenstein Softwarestruktur

Abbildung 3 zeigt das SCM mit seinen Software-Komponenten. Hier wird im oberen Bereich die Kommunikation des SCM mit dem Host dargestellt. Diese Ethernet Verbindung dient der Übertragung von Steuerbefehlen aus der laufenden Simulation an das Steuerkraftsystem [3]. Die Grundlagen zur Nutzung dieser Verbindung werden in den folgenden Kapiteln erläutert. Abgebildet ist ebenfalls die CAN-Bus-Verbindung zwischen dem SCM und den Aktuatoren.

Der untere Bereich von Abbildung 3 zeigt das AktivToolkit, eine Software die von einem Standard-Personal-Computer mit Microsoft Windows Betriebssystem verwendet wird. Sie dient als die bisherige Schnittstelle, um die Systemparameter des aktiven Steuerkraftsystems zu modifizieren. Diese Anwendung enthält alle Funktionen die im Umfang von Forschungsprojekten während der Simulation genutzt werden sollen, um die Kraft-Positions-Kennlinie und andere Parameter zu verändern. Zum Beispiel kann die Trimmungsposition, die Haft- und Gleitreibung, die Kopplung von Achsen, etc. eingestellt werden. Diese veränderten Konfigurationen am Steuerkraftsystem werden in dem *Configuration File Set* gespeichert. Weitere Funktionen des AktivToolkit umfassen die Messung der getätigten Steuereingaben während oder im Anschluss an die Simulation. Das *Aktiv8 DLM* speichert das dynamische Modell für die Regelung und stellt es *Aktiv8 Control* zur Verfügung [4].

3 2Simulate

Am DLR wird *2Simulate* als Framework für verteilte zyklisch geplante Echtzeitanwendungen entwickelt. Es wird in C++ programmiert und teilt sich in drei Komponenten auf, namentlich 2Simulate Real-Time-Framework (2SimRT), 2Simulate Model Control (2SimMC) und 2Simulate Control Center (2SimCC). Die erste Komponente 2SimRT stellt den Hauptbestandteil dar und sorgt für eine deterministische Planung und Regelung von Echtzeitprozessen. 2SimMC ist eine generische Modellschnittstelle für Modelle die aus einer Reihe von Modellierungssprachen entwickelt wurden. Beispielsweise MATLAB/Simulink oder 2Simulate Modelling Language (TSML).

Die letzte Komponente 2SimCC ist eine Bedienoberfläche für 2Simulate und wird zur Steuerung der laufenden Anwendungen verwendet [8].

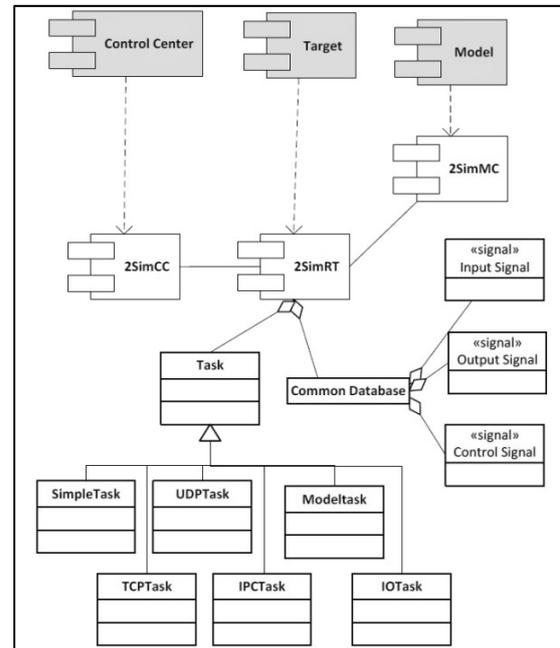


Abbildung 4 2Simulate Übersicht

Die Komponente 2SimRT stellt eine Anzahl an Prozessen in Form einer Klassenbibliothek zur Verfügung, siehe Abbildung 4. Der Nutzer kann die Prozesse durch eine Reihe von Callback Funktionen an seine Erfordernisse anpassen, die vor bzw. nach der Initialisierung sowie der Prozessierung ausgeführt werden. Beispielsweise kann ein Nutzer mit einem UDP-Task eine spezifische UDP-Verbindung betreiben oder mit einem TCP-Task eine spezifische TCP-Verbindung betreiben [8]. Die Verbindung zum Steuerkraftsystem wird über eine UDP Verbindung betrieben, siehe Kapitel 3.1. Ein Task wird durch die Instanziierung innerhalb einer 2Simulate-Anwendung zum Prozess und erfüllt für ein Nutzerprogramm eine spezifische Aufgabe.

Außerdem zählt eine generische Datenbank zu 2Simulate, die Ein-, Ausgangs- und Kontrollsignale für jede Anwendung in einer hierarchischen Struktur darstellt. Die Grundlage für die sogenannte *Common Database* sind SCD-Dateien, die in einem tabellarischen Format alle Signale auflisten. Beim Erstellen einer 2Simulate-Anwendung werden die Signale aus der SCD-Datei durch eine [externe] Anwendung in generische Strukturen umgewandelt. Die Strukturen werden über Headerdateien in eine Applikation eingebunden und können genutzt werden. Die Datenbank wird generell verwendet, um den Datenfluss durch interne und externe Schnittstellen handzuhaben [8].

3.1 Der WCLS-Task

Das aktive Steuerkraftsystem versendet Daten über eine UDP-Verbindung an die Simulationsinfrastruktur. Deswegen ist der *WCLS-Task* von dem UDP-Task abgeleitet und enthält entsprechende Methoden, um die UDP-Verbindung zu überprüfen, zu konfigurieren und zu verwenden. Zusätzlich enthält der *WCLS-Task* eigene Methoden zur Konfiguration des Steuerkraftsystems. Zum Beispiel eine Methode zum Hinzufügen von Achsen, die für jede zu verwendende Achse einmal aufgerufen werden muss. Dabei muss jeweils eine Struktur zum Empfangen und Versenden von Daten pro Achse übergeben werden.

4 WCLS-Beispielanwendung

Die Beispielanwendung wird mit einem Steuerkraftemulator der Fa. Wittenstein entwickelt. Er kann die Anbindung des Steuerkraftsystems an den Hubschraubensimulator nachbilden. Der Simulator stellt einen Großteil der Funktionen der realen Anlage zur Verfügung und ermöglicht es grundsätzliche Funktionen zu testen [5]. In den folgenden Kapiteln werden der Aufbau und die Funktionsweise der Beispielanwendung erklärt.

4.1 Die Hauptfunktion

Die Anwendung beginnt mit dem Hinzufügen von Headerdateien der verwendeten *2Simulate*-Prozesse und der *Common Database*. Darauf folgt die Hauptfunktion, sie wird teilweise in Listing 1 dargestellt. Die Funktion erstellt eine *2Simulate* Root-Applikation und initiiert die *Common Database*. Innerhalb der Funktion wird ebenfalls eine Instanz des *WCLS-Task* erzeugt. Anschließend wird das *UserCmd*-Signal, über die *setUserCtrl*-Funktion gesetzt. Das *UserCmd* Signal schaltet den Zustand der Achsen, siehe auch Kapitel 4.2. Weiterhin werden innerhalb der Funktion Funktionszeiger für Callback-Funktionen an die *2Simulate* Root-Applikation übergeben. Außerdem werden die Achsen des Steuerkraftsystems über eine *AddAxis*-Funktion hinzugefügt.

```
int main( int argc, char *argv[] )
{
//---- generate 2Simulate root ap-
plication ----
TSim* pTSim=new TSim("TUT06",
"2Simulate Tutorial6","1.0");

//---- setup common database ----
TSimulateTutorial6Database(
pTSim, &tSimulateTutorial6Common);
[...]
//---- WclsTask ----
TSimWclsTask* pWclsTask=new TSim-
WclsTask(pTSim,"WclsTask",
TASK_SCHED_RR, 30, 10*imSECTonSEC,
TRUE, TRUE, TRUE);
```

```
pWclsTask->setDesc("WclsTask for
CLS");
pWclsTask->setHost(
"127.0.0.1",3002,3001);
[...]
pWclsTask->setUserCtrl(
(int*)&com->c.wcls.UserCmd,
(int*)&com->c.wcls.State,
(int*)&com->c.wcls.AxisState);

//Callback implementation
pWclsTask->setPreProcCB(
(void*)(TSim*,TSimRtTask*))
&pWclsTask_PreProcCB);

pWclsTask->setIntermediateProcCB(
(void*)(TSim *,TSimUdpTask
*))&pWclsTask_IntermediateProcCB);

//Add axis
pWclsTask->addAxis(
(WCLSTASK_AXIS_DATA_RCV *)
&com->i.r.wcls.sp.PosPit,
(WCLSTASK_AXIS_DATA_SND *)
&com->o.r.wcls.sp.PosOffPit);
[...] }
```

Listing 1 Main-Funktion

In der Beispielanwendung werden ausschließlich die Callback-Funktion vor der Prozessierung (PreProcCB) und während der Prozessierung (IntermediateProcCB) genutzt.

4.2 Callback-Funktionen der Prozessierung

Innerhalb jedes Zyklus einer Prozessierung wird jede Callback-Funktionen aufgerufen [8]. Im Folgenden soll der wiederholte Ablauf der Prozessierung erläutert werden und anschließend die Funktionsweise der PreProcCB und der IntermediateProcCB in der Beispielanwendung erklärt werden.

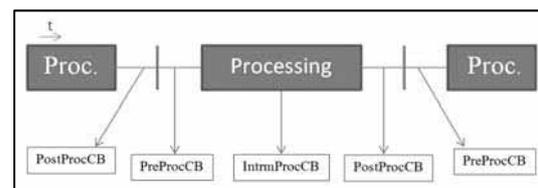


Abbildung 5 Abstrakter Prozessierungsablauf

Abbildung 5 zeigt den zeitlichen Verlauf mehrerer Prozessierungszyklen hintereinander. Dabei stellen die vertikalen Markierungen das Ende, bzw. den Anfang, eines Zyklus dar. Die PreProcCB wird vor der Prozessierung eines Zeitintervalls aufgerufen. Sie überprüft die UDP-Verbindung zwischen Simulator und aktivem Steuerkraftsystem, sowie den Betriebsstatus der Achsen. Während der Prozessierung wird der IntermediateProcCB aufgerufen. Dieser führt diverse Parameteränderungen durch, schaltet Zustände und verändert Kennlinien. Abbildung 5 stellt auch den PostProcCB im Prozessierungsablauf dar. Er wird nach der Prozessierung

eines Zyklus aufgerufen und kann zum Beispiel die berechneten Steuerkräfte kalibrieren.

Der PreProcessCB beginnt den Status der UPD-Verbindung zu überprüfen, sowie den Status der Achsen abzufragen. Listing 2 zeigt einen Ausschnitt aus der PreProcessCB-Funktion.

```
void pWclsTask_PreProcCB(
TSim *pAppl, TSimRtTask *pRtTask)
{
    [...]
    else if(
        iClsCmd==WCLSTASK_STATE_ACTIVE
        ||
        iClsCmd==WCLSTASK_STATE_ACTPEND
    )
    { com->c.WCLS.UserCmd =
      WCLSTASK_USERCMD_START; }

    else if(
        iClsCmd==WCLSTASK_STATE_STARTUP
    )
    {com->c.WCLS.UserCmd =
      WCLSTASK_USERCMD_INIT; }

    else if(
        iClsCmd==WCLSTASK_STATE_SETOFF
    )
    {com->c.WCLS.UserCmd =
      WCLSTASK_USERCMD_OFF; }
    [...]
}

```

Listing 2 PreProcess Callback-Funktion

Für die Achsenüberprüfung wird ein Signal namens UserCmd verwendet. Der Wert dieses Signals entscheidet über einen Wechsel der Achsenzustände, siehe Kapitel 4.1. Um das UserCmd zu beeinflussen, wird ein weiteres Signal aus der Datenbank genutzt, namentlich iClsCmd. Dieses wird durch eine bedingte Verzweigung in Listing 2 abgefragt und setzt dadurch den Wert der UserCmd-Variable.

Der IntermediateProcCB führt die Initialisierung von Parametern durch und vermeidet damit Fehler beim Ersten Programmdurchlauf. Eine Bedingung entscheidet, ob Veränderungen an einer Achse durchgeführt werden sollen. Dabei werden Integerwerte von 0 bis 7 angenommen, welche entsprechend der acht unterschiedlichen Achsen Parameter verändern. Die Bedingung wird mit dem Trim-Release-Status als Beispiel in Listing 3 dargestellt.

```
void pWclsTask_IntermediateProcCB(
TSim *pAppl, TSimUdpTask *pUdpTask)
{
    [...]
    if ( com->c.WCLS.EditAxis >= 0
    && com->c.WCLS.EditAxis <= 7 ) {
        [...]
    }
}

```

```
//--set trim release status--
fTrimRel[com->c.WCLS.EditAxis]
= lcast(&com->c.WCLS.TrmRel[
com->c.WCLS.EditAxis]);

pWCLSTask->setTrimRel(
com->c.WCLS.EditAxis,1,
&fTrimRel[com-
>c.WCLS.EditAxis]);
}
[...]
```

Listing 3 Parameteränderung aus IntermediateProcCB

Das Listing zeigt einen Ausschnitt aus dem verschachtelten Codeblock der oben genannten Bedingung. Die hier verwendete Funktion ändert den TrimRelease-Status indem sie der setTrimRel-Funktion einen Parameter übergibt, hier „0“ oder „1“. Für diese Funktion wird vorerst eine lokale Variable verwendet, um den Datentyp des Parameters zu casten. Dieser Cast ist notwendig, um die Variable als Argument zu übergeben. Anschließend wird die Speicheradresse des Parameters an die Funktion übergeben. Da die setTrimRel-Funktion Teil des WCLS-Tasks ist wird der Zugriffoperator „->“ verwendet.

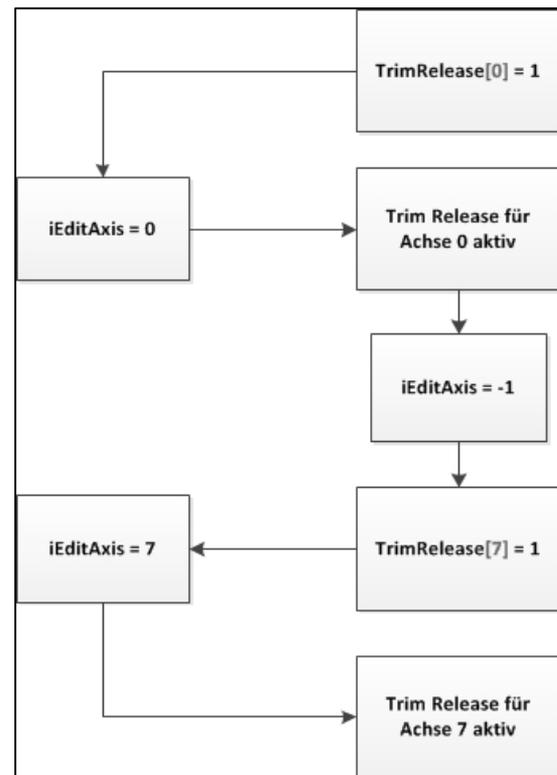


Abbildung 6 Beispiel der Parameterveränderung

Der oben beschriebene Funktionsablauf ist in Abbildung 6 als Ablaufschema dargestellt. Der genaue Ablauf beginnt damit eine achsenspezifische Variable eines Parameters auf einen gewünschten Wert zu setzen. Im Beispiel TrimRelease[0] = 1. Anschließend wählt man mit der iEditAxis-Variable

die zu verändernde Achse aus, wodurch die Bedienung aus Listing 3 verzweigt und die gewünschte Änderung ausgeführt wird. Abschließend wird die „iEditAxis“-Variable zurückgesetzt. Abbildung 6 verdeutlicht zusätzlich die Idee einer achsenspezifischen Variablen, indem der Ablauf der gleichen Änderung für eine andere Achse abgebildet ist.

Die Änderung der Kennlinien beinhaltet eine weitere Besonderheit. Bevor die Koordinaten einer Kennlinie übergeben werden, wird ein Programm zum Einlesen aufgerufen. Die eingelesenen Daten umfassen die Koordinaten, den Titel und weitere Bedingungen der Kennlinie. Für das Einlesen wird eine Textdatei verwendet, sie muss in einem definierten Format strukturiert sein.

5 Bewertung und Ausblick

Diese Publikation beschreibt das aktive Steuerkraftsystem der Fa. Wittenstein und geht auf die Fähigkeiten des 2Simulate-Frameworks ein. Anhand einer Beispielanwendung wird eine mögliche Variante gezeigt, um Veränderungen am Steuerkraftsystem während einer 2Simulate-Echtzeitanwendung durchzuführen.

Die graphische Benutzeroberfläche der Wittenstein Software, AktivToolkit, ist sehr umständlich bei Veränderungen an den Konfigurationen. Außerdem sind die Erklärungen der Handbücher nicht eindeutig und führen zu einer schweren Handhabung des Werkzeugs. Deswegen sollte der WCLS-Task eine graphische Benutzeroberfläche erhalten die die Übersichtlichkeit, besonders für Einsteiger, verbessert. Ebenso kann das Einlesen der Kennlinienkoordinaten vereinfacht werden, indem ein Anwenderfreundliches Programm wie Microsoft Excel zur Eingabe von Daten verwendet wird.

Im Umfang eines weiteren Projekts kann die erstellte Software an den Hubschraubersimulator angebunden werden. Damit können Nutzer von den Funktionen der Beispielanwendung in ihren eigenen Anwendungen Gebrauch machen.

6 Literaturverzeichnis

- [1] von Grünhagen, Wolfgang, Müllhäuser, Mario et. Al.; „Active Inceptors in FHS for Pilot Assistance Systems“; 36th European Rotorcraft Forum; Paris, France; 7.-9. September 2010
- [2] Duda, H., Gerlach, T., Advani S., Potter M.; „Design of the DLR AVES Research Flight Simulator“; AIAA Modeling and Simulation Technologies (MST) Conference, Boston, MA, 2013

- [3] “Wittenstein Control Loading Systems Manual”; Wittenstein aerospace & simulation Inc; 17. Juli 2007
- [4] “Wittenstein AktivToolkit User Manual”; Wittenstein aerospace & simulation Inc (2005)
- [5] “Wittenstein Control Loading Systems Protocol Simulator Instructions”; Wittenstein aerospace & simulation Inc (2007)
- [6] Allerton, David; “Principles of Flight Simulation”; John Wiley & Sons, United Kingdom; 2009
- [7] Cowling, David; “The Development of a new Range of Control Loading Systems”; Royal Aeronautical Society Flight Simulation Group Meeting; United Kingdom, London; 3.-4. November 2004
- [8] Gotschlich, J., Gerlach, T, Durak, U.; “2Simulate: A Distributed Real-Time Simulation Framework“; ASIM STS/GMMS Workshop 2014; Reutlingen, Germany; 2014

7 Abkürzungsverzeichnis

- RHMT – Reconfigurable Helicopter Mission Trainer
- AVES – Air VEHICLE Simulator
- DLR – Deutsches Zentrum für Luft- und Raumfahrt e. V.
- SCM – System Control Module
- SPS – System Power Supply
- CAN-Bus – Controller Area Network
- FbW – Fly-by-Wire
- COTS – Commercial Off The Shelf
- GUI – Graphische Benutzeroberfläche
- 2SimRT – 2Simulate Real-Time-Framework
- 2SimMC – 2Simulate Model Control
- 2SimCC – 2Simulate Control Center
- SCD – Simulation Common Database
- PreProcCB – Preprocessing Callback
- IntrmProcCB – Intermediateprocessing Callback
- PostProcCB – Postprocessing Callback

Towards a Distributed Simulation Toolbox for Scilab

Awad Mukbil¹, Peter Stroganov¹, Umut Durak^{1,2}, Sven Hartmann¹

¹ Clausthal University of Technology

Department of Informatics

² DLR Institute of Flight Systems

{awad.mukbil, peter.stroganov, sven.hartmann}@tu-clausthal.de, umut.durak@dlr.de

Scilab is an open-source cross-platform computing environment for engineering and scientific applications. It provides a high-level programming language with hundreds of built-in functions for numerical computation. Additionally, it encompasses Xcos, which is a Causal Block Diagrams-based graphical editor for modeling and simulation of dynamical systems. Thus, it supports model-based simulation. With emerging technologies like cyber-physical systems, ubiquitous computing, smart devices and ambient intelligence, the subject reality (simuland) involves multiple heterogeneous and distributed interacting entities. Hence, the modeling and simulation of these emerging systems is evolving towards constructing distributed simulations where individual models of distributed entities interact with each other via well-defined and agreed interfaces. Although distributed simulation has been widely utilized by mainly defense modeling and simulation community since the 1980s, the combination of distributed simulation with model-based simulation techniques for simulating technical systems poses new research challenges. To date, Scilab and Xcos do not contain built-in distributed simulation capabilities. This paper first introduces the requirement of distributed model-based simulation and then presents an implementation strategy aligned with the Scilab and the Xcos software architecture.

1 Introduction

Scilab is an open source, cross-platform computing environment [1]. It is widely employed for engineering and scientific applications such as signal processing or optimization [2].

Scilab encompasses a high-level programming language with hundreds of built-in functions for numerical computation and a graphical editor, called Xcos, for modeling and simulation of dynamical systems using Causal Block Diagrams.

Technical system is the term that is used for all man-made machines [3]. As the emerging category of technical systems, Cyber Physical Systems (CPS) are now addressing new capabilities and properties. CPS are real-time systems which are composed of distributed networked heterogeneous physical devices and computational components [4]. They were introduced as the integration of computing with the physical processes [5]. They are characterized by their networked interacting components.

Not only the technical systems, but their operation environment is also changing. Further ubiquitous computing concepts are integrating mobile computing capabilities with pervasive elements of the environment, including sensors, actuators and computing

nodes [6]. With ambient intelligence, the environment is getting smarter and reactive [7]. Simuland is defined as the real-world item of interest; the object, process, or phenomenon to be simulated [8]. The simuland in the modeling and simulation of technical systems is evolving rapidly towards involving multiple heterogeneous, and distributed interacting entities. Hence, distributed simulation tools and techniques are more required now for co-simulating loosely coupled individual models of distributed entities of emerging technical systems. These techniques and tools are required in model-based simulation environments, like Scilab, that are extensively used in the engineering of technical systems.

Distributed simulation is a technology that enables a simulation to be executed on multiple computing nodes, such as a set of networked personal computers [9]. It usually deals with the execution of simulations on loosely coupled systems. It includes execution on geographically distributed computers interconnected via a wide area network such as the Internet. While the traditional motivations were reducing execution time, enabling geographically distributed experimentation, integrating simulators from different manufacturers and fault tolerance, recently, the distributed nature of simulands and the corresponding models

has been declared the fundamental purpose of distributed simulation [10].

The communication among the entities in a distributed simulation can be conducted through pair-wise connections or by utilizing a shared bus (middleware) that all simulations can use collectively. High Level Architecture (HLA) proposes industry-wide accepted standardization in a shared bus by specifying the interfaces with the interconnected entities and the shared bus [11][12][13]. There are also some efforts that focus on providing HLA functionalities in model-based simulation environments. Pawletta et al. propose a HLA toolbox for MATLAB [14][15]. ForwardSim is providing HLA Toolbox for MATLAB [16] and HLA Blockset Simulink [17], while Theppaya et al. offer an approach to integrate HLA Runtime Infrastructure Services with Scilab [18].

(API) that is designed as an external Scilab module. Then, Section 3 introduces an approach for developing a networking block library as an Xcos palette.

2 Scilab Networking Module

Scilab supports several general purpose programming languages, such as C/C++, Java or FORTRAN. The Scilab Networking Module is proposed as an external Scilab module. It aims to constitute an abstraction layer for Scilab users. This abstraction layer will wrap BSD Sockets [19], a classical socket API that was developed in the 1980s.

Briefly, the Scilab Networking Module will give Scilab the ability to open a socket by choosing/reserving a Port number and the local IP address of the device. Then, based upon the protocol that will be used (TCP/UDP), Scilab will start communication in a Client/Server architecture.

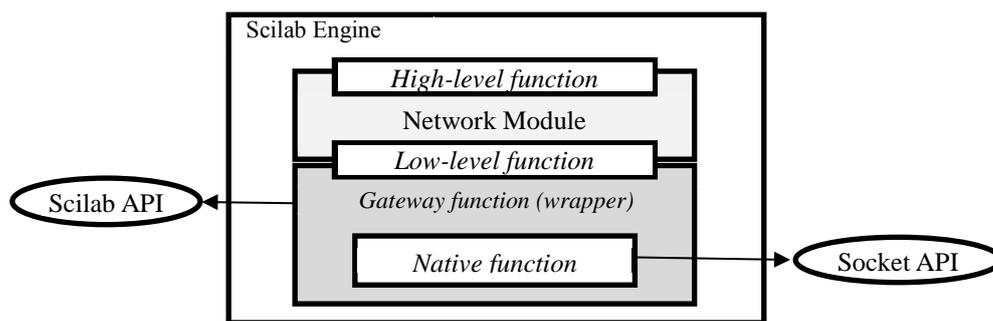


Figure 1 Extending native function to the Scilab engine using Gateway

These efforts constitute bases for constructing an HLA capability in a model-based simulation environment. While it is a legitimate requirement to support HLA for a distributed simulation toolbox, due to high complexity and a relatively gradual learning curve, simple and relatively easy pair-wise networking is regarded as the initial capability set towards a more capable distributed simulation toolbox. Therefore, this study presents an attempt at enabling pair-wise connections between simulation entities, particularly in Scilab and Xcos.

One of the widely employed techniques for pair-wise networking simulations is to use Transmission Control Protocol / Internet Protocol (TCP/IP) or User Datagram Protocol (UDP). Accordingly, this paper will present an implementation strategy for UDP and TCPI/IP networking features in Scilab and Xcos aligned with their software architecture. Section 2 presents an implementation strategy for a scripting level networking Application Programming Interface

Scilab software architecture promotes a methodology to create external modules. Our concern here is to embed the capabilities of an external library to the Scilab engine. In Scilab, this is called *interfacing* [20]. Interfacing is the linking process, by which Scilab can use native function that is developed using a general purpose programming language, as a primitive Scilab function.

Gateway is a term used in Scilab for a function, written in C language, that is responsible for converting the data to and from the native function, and provides a call interface to the native function. It is also called *wrapper function* because it wraps the native function, so it can be regarded as a Scilab primitive function (Figure 1).

The gateway function relies on some header files, such as *api_scilab.h*, *MALLOC.h* and *Scierror.h*. These headers support the gateway with all function prototypes needed to interact with the Scilab engine.

Below is an example of a native function that is a part of the Scilab Networking Module. *UDPsend_dblData* sends data (doubles) using the UDP protocol. It is a pure C function that is called by the gateway.

As given in the following code listing, it opens a UDP socket for the Scilab process in order to send the data to the specified address and the port number.

```
int UDPsend_dblData(char *_stServerName,
int _iPort, double _dblData)
{
    int sock, length, n;
    struct sockaddr_in server;
    struct addrinfo hints,*res;
    char str[INET_ADDRSTRLEN];

    memset(&hints,0,sizeof hints);
    hints.ai_family=AF_INET;
    hints.ai_socktype=SOCK_DGRAM;

    if((n=getaddrinfo(_stServerName, NULL,
&hints, &res))!=0)
        fprintf(stderr, "getaddrinfo error:
%s\n",gai_strerror(n));

    sock = socket(AF_INET,SOCK_DGRAM,0);
    if (sock < 0)

perror("socket failed!");

    server.sin_addr=((struct
sockaddr_in*)res->ai_addr)->sin_addr;
    server.sin_family = AF_INET;
    server.sin_port = htons(_iPort);
    length=sizeof(struct sockaddr_in);

    /* send data using UDP */
    n=sendto(sock, &_dblData,
sizeof(double), 0, (const struct
sockaddr*)&server, length);
    if (n < 0){
        perror("UDP-send failed!");

printf("%s:%d",inet_ntop(AF_INET,&server.
sin_addr,str,sizeof(str)),server.sin_port
);}
    else
        printf("UDP-sending %f, n=%d\n",
_dblData, n);

    shutdown(sock,SHUT_RDWR);
}
```

Code 1. Native C function to send data using UDP.

Gateways incorporate various logical sections or steps. These steps will be presented as code listings

for the example gateway that wraps the native function *UDPsend_dblData*. The first step, as presented in Code 2, is to declare the local variables that are required for the wrapper function. These local variables will be used by the Scilab engine for data exchange.

```
int sci_udpclient(char *fname)
{
    SciErr sciErr;
    int* portAddr = NULL;
    int* dataAddr = NULL;
    int* domainAddr = NULL;
    double port = 0;
    char *domain = NULL;
    double dblData = 0;
    char* strData = NULL;
```

Code 2. Define local variables

In Code 3, we let Scilab check the number of input and output arguments which will be provided by the Scilab engine. Then, these input and output arguments are assigned to the local variables that have already been defined.

```
CheckInputArgument(pvApiCtx, 3, 3);
CheckOutputArgument(pvApiCtx, 0, 1);

sciErr =
getVarAddressFromPosition(pvApiCtx, 1,
&domainAddr);
    if(sciErr.iErr)
    {
        printError(&sciErr, 0);
        return 0;
    }
    sciErr =
getVarAddressFromPosition(pvApiCtx, 2,
&portAddr);
    if(sciErr.iErr)
    {
        printError(&sciErr, 0);
        return 0;
    }
    sciErr =
getVarAddressFromPosition(pvApiCtx, 3,
&dataAddr);
    if(sciErr.iErr)
    {
        printError(&sciErr, 0);
        return 0;
    }
```

Code 3. Define variables and addresses pointers

Code 4 presents the validity check for each parameter. The Scilab API provides various functions for

each datatype validation; here, we present only two of these functions: *isStringType* and *isDoubleType*. Respectively, these functions check whether the value given is of *string* type or of *double* type. When the validation is successful, it executes the function that is responsible for taking the data from the Scilab engine. For the local variables, we use *getAllocatedSingleString* to copy the string value to a local variable, and *getScalarDouble* to copy the double value to a local variable.

Notice that the first string value is for the domain address, and the second double value is for the port number.

```

/* ===== check inputs ===== */
// check domain
if(!isStringType(pvApiCtx,
domainAddr))
{
    Scierror(999, _("%s: Wrong
type for input argument #d: A string
expected.\n"), fname, 1);
    return 0;
}
if (
getAllocatedSingleString(pvApiCtx,
domainAddr, &domain) )
{
    Scierror(999, _("%s: Wrong
size for input argument #d: A scalar
expected.\n"), fname, 1);
    return 0;
}
// check port
if(!isDoubleType(pvApiCtx, portAddr))
{
    Scierror(999, _("%s: Wrong
type for input argument #d: A Integer
expected.\n"), fname, 2);
    return 0;
}
if (getScalarDouble(pvApiCtx,
portAddr, &port) )
{
    Scierror(999, _("%s: Wrong
size for input argument #d: A scalar
expected.\n"), fname, 2);
    return 0;
}

```

Code 4. Check and get parameters value from Scilab Console

Finally, in Code 5, we check the last parameter, which holds the data that we want to send, then call

the native function that is responsible for sending the data using the address and the port number.

```

if (getScalarDouble(pvApiCtx, dataAddr,
&dblData))
{
    Scierror(999, _("%s: Wrong size for
input argument #d: A scalar
expected.\n"), fname, 3);
    return 0;
}
UDPsend_dblData(domain, (int)port,
dblData);
sciprint("Data sent (UDP): %f size:
%d\n", dblData, sizeof(dblData));

```

Code 5. Calling the native function

3 Xcos Networking Palette

In this section, we will introduce an approach for developing a networking block library as Xcos palette. Rather than employing a native API, our approach adopts Scilab's native support for the Tool Command Language (TCL). TCL is a general purpose scripting language that was designed in the 1980s [21]. Scilab offers a native mechanism to invoke TCL language code directly from Scilab code using the *TCL_EvalStr* function.

TCL core supports TCP sockets, but UDP implementations have been provided as extensions of the TLC core [22]. This section presents a TCL-based implementation strategy that utilizes core TCP sockets. In Code 6, the *SERVER_open* function, which is used to open a new TCP socket on the given port number, is presented. This function just wraps several TCL code lines into Scilab code. The great advantage of this approach with TCL is that it is cross-platform, as all Scilab distributions contain TCL by default.

```

function SERVER_open(port)
    TCL_EvalStr("set ::SERVER_handle [socket
-server SERVER_newClient
+string(port)+"]");
    TCL_EvalStr("fconfigure $::SERVER_handle
-blocking 0 -translation crlf;");
    printf("Server open for client Request at
Port%d\n ",port)
endfunction

```

Code 6. TCP server initialization

When developing external blocks for Xcos, *interface functions* are used to define the block appearance, the number of inputs and outputs and the behavior. Code 7 shows an interface function of the block that sends data to a remote client. The function has three argu-

ments: job, arg1 and arg2. Job specifies the mode of the Scilab block's interface function. There are three modes:

Plot: In this mode, data about the block are displayed on the plot.

Set: In this mode, the block has its parameters initialized using the *scicos_getvalue* function.

Define: In this mode, block appearance is initialized and also the inputs and outputs are checked.

```
function
[x,y,typ]=newcsi_block_m(job,arg1,arg2)

x=[];
y=[];
typ=[];

select job

case "plot" then
standard_draw(arg1)
case "getinputs" then
[x,y,typ]=standard_inputs(arg1)
case "getoutputs" then
[x,y,typ]=standard_outputs(arg1)
case "getorigin" then
[x,y]=standard_origin(arg1)
case "set" then

x=arg1
graphics=arg1.graphics;
model=arg1.model;
exprs=graphics.exprs

while %t do

[ok,ip,r,rv,exprs]=scicos_getvalue('Set TCP
server parameters',..
['TCP port';'no use'];,..
list('vec',1,'vec',1),..
exprs)

if ~ok then
break,
end
model.rpar = [ip;rv];
graphics.exprs = exprs
x.graphics = graphics
x.model = model
break
end

case "define" then

in=0
out=1
```

```
model=scicos_model()
model.sim=list("mybf",5)
model.out=1

model.blocktype="c"
model.dep_ut=[%t %t]
model.label="TCP R"
exprs=[string(["1234";"0"])]
gr_i=['txt=["TCP
RECEIVER"]'];

'xstringb(orig(1),orig(2),txt,sz(1),s
z(2),'fill')')
disp(gr_i);
x=standard_define([4 2],model,exprs,gr_i)
disp("ready to go")
End

endfunctionsd
```

Code 7. The block interface function

Computation functions are used to define the behavior of an Xcos block during a simulation. They have input parameters – block and flag. Block corresponds to the block which owns this function, and flag represents the simulation phase. Using the flag parameter, we can define the function behavior in different simulation phases of an Xcos model, such as initialization, simulation and termination.

In the following example, see Code 8, a computation function of the block which receives data over the network is given. Particular tasks are performed in different simulation phases. When flag has value 4 (initialization), the function starts the TCP server. If flag is 5 (termination), it stops the server. If flag is 1 (simulation), the computation function sets block output to what is received from the TCP server.

```
function block=mybf(block,flag)
if flag==1 then
block.outptr(1)(1)=retval;
end

if flag==4 then
global retval;
retval=0;

mode(0);

SERVER_close();
SERVER_open(block.rpar(1));

end

if flag==5 then
SERVER_close();
end
```

```

endfunction

function SERVER_readableEvent(str)

disp (str);
global retval;
retval=strtod(str);
endfunction

```

Code 8. A block computation function

4 Conclusion

After discussing the emerging requirements for having distributed simulation capabilities in model-based simulation environments for technical systems, the paper presents an implementation strategy for an initial peer-to-peer networking capability for Scilab and Xcos with the Scilab Networking Module and the Xcos Networking Palette.

The Scilab Networking Module exercises an implementation strategy that exploits the Scilab gateway, which enables interfacing with general purpose programming language APIs and the Xcos Networking Palette employs TCL as a general purpose scripting language. While utilizing general purpose programming language APIs is more complex, they provide flexibility for developers to extend the feature set of the API using general purpose programming languages. Running TCL scripts in Scilab, on the other hand, is easier to implement, but limits the user with the TCL capabilities.

This paper presents a first attempt towards a distributed simulation toolbox in Scilab. Future work includes developing a full capability implementation for peer-to-peer communication. Both TLC and Scilab gateway will be employed where appropriate. Further, we will be investigating the implementation strategies for shared bus architectures like HLA.

5 References

- [1] S. L. Campbell, J. P. Chancelier and R. Nikoukhah. *Modeling and Simulation in Scilab/Scicos*. Springer, USA, 2006.
- [2] C. Bunks, J-P. Chancelier, F. Delebecque, M. Goursat, R. Nikoukhah and S. Steer. *Engineering and scientific computing with Scilab*. Edited by Claude Gomez. Springer Science & Business Media, 2012.
- [3] V. Hubka and W. Eder. *Theory of Technical Systems: A Total Concept Theory for Engineering Design*. Springer-Verlag, Germany, 1988.
- [4] T. Xiao and W. Fan. *Modeling and Simulation Framework for Cyber Physical Systems*. Advanced Methods, Techniques, and Applications in Modeling and Simulation, Japan, Springer, pp. 105-115, 2012.
- [5] E. Lee. *Cyber Physical Systems: Design Challenges*. 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC), Orlando, FL, 2008
- [6] K. Lyytinen and Y. Yoo. *Issues and Challenges in Ubiquitous computing*. Communications of the ACM, 45(12), pp. 63-65, 2002.
- [7] J.C. Augusto and P. McCullagh. Ambient intelligence: Concepts and applications. *Computer Science and Information Systems*, 4(1), pp. 1-27, 2007.
- [8] M.D. Petty. *Verification, Validation, and Accreditation*. Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains, Eds. Sokolowski, J.A. and Banks, C.M. John Wiley & Sons, 2010.
- [9] R. M. Fujimoto. *Parallel and Distributed Simulation Systems*. Proceedings of 2001 Winter Simulation Conference, Arlington, VA, 2001.
- [10] O. Topcu, U. Durak, H. Oguztuzun and L. Yilmaz. *Distributed Simulation: Model-Driven Engineering Approach*. Springer, 2016.
- [11] IEEE Standard for Modeling and Simulation High Level Architecture (HLA) – Framework and Rules. New York, NY, 2010.
- [12] IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification. New York, NY, 2010.
- [13] IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template (OMT) Specification. New York, NY, 2010.
- [14] S. Pawletta, B.P. Lampe, W. Drewelow and T. Pawletta. *Eine HLA-Toolbox für Matlab*. Simulation und Visualisierung 2001 (SimVis 2001), Magdeburg, Germany, 2001.

- [15] S. Pawletta and C. Stenzel. *Matlab(R) High Level Architecture Toolbox*. [Online] Available at: <https://www.mb.hs-wismar.de/~stenzel/software/MatlabHLA.html> [Accessed on 26 January, 2016].
- [16] ForwardSim, *HLA Toolbox*. [Online] Available at: <http://www.forwardsim.com/products/hla-toolbox/> [Accessed on 26 January, 2016].
- [17] ForwardSim, *HLA Blockset*. [Online] Available at: <http://www.forwardsim.com/products/hla-blockset/> [Accessed on 26 January, 2016].
- [18] T. Theppaya, P.Tandayya and C. Jantaraprim. *Integrating the HLA RTI Services with Scilab*. 6. IEEE International Symposium on Cluster Computing and the Grid (CCGRID 06), Singapore, 2006.
- [19] M.T. Jones. *BSD sockets programming from a multi-language perspective*. Charles River Media, Inc., 2003.
- [20] *Scilab help*. [Online] Available at: https://help.scilab.org/docs/5.5.2/en_US/api_scilab.html [Accessed on 26 January, 2016]
- [21] P. Raines and J. Tranter. *TCL/TK in a Nutshell*. O'Reilly Media, Inc., 1999.
- [22] *UDP for Tcl*. [Online] Available at: <http://wiki.tcl.tk/16733> [Accessed on 26 January, 2016].

A layered structure for modeling manufacturing processes with the inclusion of energy consumption

Artur Schmidt¹, Thorsten Pawletta¹, Peter Junglas²

¹Hochschule Wismar, Forschungsgruppe CEA

²PHWT Vechta/Diepholz/Oldenburg

{artur.schmidt,thorsten.pawletta}@hs-wismar.de, peter@peter-junglas.de

In the design of manufacturing processes the consideration of resource usage, especially energy consumption, is getting more attention. However, the inclusion of the relevant physical processes in a unified modeling approach is often a non-trivial task. The commonly used simulation tools in this domain usually only support discrete event modeling techniques, while models of physical processes are often continuous in nature and are modeled using ordinary differential equations (ODEs) or differential algebraic equations (DAEs). Moreover, the investigation of such physical processes demands a more specific consideration of process control operations, which are favorably modeled using state machines. To combine the diverse paradigms in a clear-cut way, a layered structure is proposed, which encapsulates the different levels of the model and defines a precise interface between them. The example of an industrial furnace serves as a proof of concept and demonstrates its versatility by integrating a transaction-based material flow model with a state machine describing the process control and physical modeling of the energy flows.

1 Introduction

Models of dynamic systems are based on a simplification of reality. A basic problem is the allowable degree of simplification, which depends on the purpose for which a model is developed. The model must meet the accuracy requirements, but should also be as simple as possible. This means that: (i) the answers delivered by the model have to be true for the current purpose; and (ii) the model should be simple enough that working with it is easy [1].

The modeling and simulation of manufacturing process chains has been a subject of study for several decades. According to [2], the typical modeling approach is discrete event modeling in this domain. This fact is reflected in the popular simulation tools applied in this field today, which particularly support discrete event modeling methods. However, the situation has recently been changing, because of new aspects that have been taken into account and increasing requirements for accuracy.

One of these aspects is the time-dependent energy consumption of single process operations or of total process chains. Approaches for single process opera-

tions, such as in [3], are focused on the energy consumption of single machine operations and they are often based on differential algebraic equations. However, approaches for investigating several machines coupled in a process chain use more abstract models mostly based on discrete event methods, such as in [4] and [5]. Today most approaches related to energy processes in manufacturing are only focused on the simulation of energy consumption. In [6] it is emphasized that the energy consumption of a production line has to be considered in production planning and scheduling. However, the energy consumption then has to be examined together with all the other production performance indicators, such as through-put time, load factors etc.

At this point it is obvious that the decision regarding the necessary degree of simplification for a model takes on a new dimension, because the number of essential variables and evaluation parameters increases. Moreover, it is necessary to determine how finely grained the approximations for continuous energy consumption processes should be. So far, manufacturing models considering production planning problems are mostly discrete event-oriented and the popular simulation environments used in this domain do

not support continuous process modeling.

In [7] a simulator coupling is proposed to execute manufacturing models with mixed discrete event and continuous process behavior. However, the approach is a customized solution and it shows well-known problems of simulator couplings. A hybrid modeling approach based on the Discrete Event and Differential Equation System Specification (DEV&DESS) in [8] is discussed in [9]. It extends the DEV&DESS by an inline integration method that schedules the integration time like discrete events. Thus, continuous processes can be modeled using ordinary differential equations and are solved within a discrete event-oriented simulation environment. However, both approaches are limited according to the modularity and clear separation between model specification and simulation execution. A real modular and accurate problem solution requires a hybrid simulation environment. This paper introduces a novel modular, layered modeling approach for complex hybrid systems implemented in the MATLAB/Simulink [10] environment and using the toolboxes SimEvents [11], Stateflow [12] and Simscape [13]. It supports component-based modeling, exemplified by the example of manufacturing components. Each component consists of a layered structure with sub-components for mapping the internal: (i) material flow; (ii) control operations; and (iii) physical machine operations. Sub-components of a layer can be easily replaced by other sub-components if the interfaces are compatible. Hence, several sub-components with different modeling methods or levels of detail can be developed for a component and alternatively used in a model depending on the goal of the simulation study.

The paper introduces the different layers describing a production component and discusses typical modeling methods for each layer in a general way. Then the approach is illustrated using the application of an industrial furnace. For the industrial furnace a library with several sub-components for each layer is introduced. Finally, some simulation results of various furnace models with different modeling methods and levels of detail are presented.

2 Layers of abstraction for describing production facilities

The *multilayered architecture* is a common design pattern used in software development to cope with the

complexity of the system to be modeled [14]. Each layer represents a different aspect of the system with well-defined interaction between the layers. For the description of a production facility three layers immediately come to mind (Fig. 1):

- The *material flow* layer describes the flow of the parts into and out of the facility. It is the basic layer for connecting facilities to a production line or process chain.
- The *process control* layer maps the local process control of the facility. This is especially important for facilities with complicated manufacturing processes, where the parts are handled in several physical process phases, which may be iterated according to an internal control program or other internal conditions.
- The *process physics* layer implements the physical processes that are relevant in the different manufacturing phases.

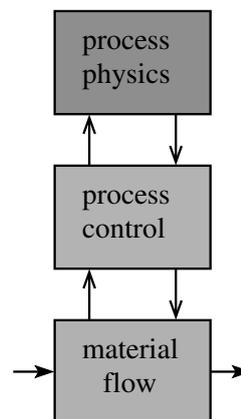


Figure 1: Layers describing a production facility

It is important to adapt the level of abstraction for each layer to the questions at hand and only include the processes that are needed to answer them. Each layer provides its own kind of information: The material flow is mainly concerned with logistic quantities such as waiting times and utilizations; the control layer gives the timing of the different manufacturing phases, which can be useful in the context of other information. The kind of output values of the physics layer can vary widely depending on the implemented details. For the investigation of energy-related questions that are studied and exemplified here, one is mainly interested in the total energy consumption. But

more detailed information can be used too, e.g. the energy used in single internal phases for process optimization or the exact timing of power requirements to take advantage of time varying energy costs.

3 Modeling approaches corresponding to the layers

The layers described above typically utilize very different kinds of abstraction, which demand diverse modeling approaches. A simulation platform for the complete model therefore needs to support a wide range of modeling paradigms.

The material flow layer is often described using the transaction-based modeling method, a special kind of a process-based approach, which defines abstract entities moving between stationary components and acting on them [8]. The entities are identified in manufacturing systems with workpieces or tools, components moved between the production facilities, thus making the material flow apparent.

A convenient method to describe the different manufacturing phases in a facility, which are the concern of the process control layer, are state graphs [15]. The phases directly correspond to the states and the transitions describe the internal process logic. Alternatively, one could again use a process-based approach, wherein the entities now do not denote physical parts, but abstract control tokens. Their flow through stationary components reflects the changes of the internal phases.

The physical manufacturing processes are often modeled as continuous systems, wherein the basic physical laws, e.g. from mechanics or thermodynamics, lead to differential equations (ODEs). If the description contains algebraic constraints or the equations are constructed automatically using a physical modeling approach [16], then the mathematical model is enlarged to a system of differential algebraic equations (DAEs).

For the simulation of the logistic and process-oriented aspects of a production line several discrete simulation programs exist and are in wide industrial use, such as Arena [17] from Rockwell Automation and Plant Simulation [18] from Siemens. Usually these programs lack components such as ODE or DAE solvers to cope with continuous system specifications. This is why different simulators are coupled to solve such problems, such as in [7]. The introduced layered structure

(Fig. 1) can support such simulator couplings, but it cannot solve the general numerical problems of a simulator coupling.

Sometimes a fine-grained physical description of a production facility is already available, e.g. a CAD model inside a finite element program. In such a case the layered structure also supports problem solving using simulator couplings. But this method usually has several severe drawbacks; besides being very cumbersome to handle and leading to very long execution times, the level of physical detail is often too high for the questions in which we are interested here. The extraction of the relevant results from a plethora of computed numbers is often a non-trivial task.

Instead, one should use a system that is capable of hybrid modeling to create a model with the appropriate level of detail. One widely used software is the Matlab/Simulink environment. Basically designed for the simulation of continuous systems using the signal flow paradigm, it can be extended using additional toolboxes and blocksets to include discrete and discrete event modeling features: SimEvents implements components for process-based modeling; Stateflow provides state charts. Both are fully integrated in the Simulink environment. To expand the continuous tool chest one can add physical modeling features with the Simscape blockset and language. These toolboxes/blocksets will be used in the following to validate the layered structure approach by implementing concrete examples.

An alternative choice could be to use a Modelica-based solution [16]. Though mainly designed for continuous modeling with the physical modeling paradigm, it contains basic hybrid features. They have been used to include an additional package for state charts [19] and the package DEVSLib [20] which allows for defining discrete event systems using Zeigler's basic DEVS formalism [8]. Based on this, one can implement a ready-to-use Modelica package similar to Matlab's SimEvents blockset. A corresponding Modelica library is in preparation [21].

4 Application to an industrial furnace

For the investigation of energy efficiency an industrial furnace is an especially rewarding example; it is widely used in metalworking processes for hardening,

annealing and tempering and is often one of the most extensive energy consumers. In addition, its internal operation is generally rather complex, making it an ideal example for the application of different modeling approaches. The concrete operation of the furnace is patterned in the following after the descriptions in [9] and [22].

When parts enter the furnace, they are collected until a given batch size is reached. A complete batch then enters the oven, is subjected to possibly several phases of heat treatment and leaves the furnace. This completes the description of the material flow.

The process control layer should map six phases (Fig. 2).

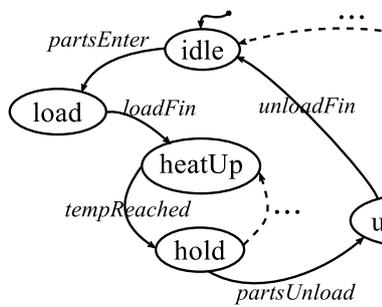


Figure 2: State graph describing the in cesses

The *idle phase* spans the times before and after all parts have left the *load phase* parts enter the oven phase the oven is heated until a *g* is reached; this is then held constant *phase*. According to the requirements: hold phases can be iterated several times at different temperatures. Finally, the parts leave the *unload phase*. An additional *cool-down* phase may be included either to make sure that the parts leave the oven with a moderate temperature or to describe a shutdown of the oven.

Since the focus of this investigation lies on the energy consumption, the process physics layer is concerned with the modeling of the heat flows in the furnace. The only source is the power supply of the actual heater. From here the heat flows mainly through convection and radiation processes to the parts and to the internal structures and the casing of the oven. During the heat-up and hold phases losses are mainly due to conduction through the casing into the environment, while in the load or unload phases additional losses are caused

by the open doors. Because of the complicated geometry, the physical details, especially of the convection processes, are also rather complicated. However, for the estimation of the total heat flows common approximate methods usually give quite accurate results.

5 Designing a library for the furnace

To show how to realize the abstract ideas in practice a set of layered models will be described using modeling methods supported by the Matlab/Simulink environment. The inclusion of a complete furnace component

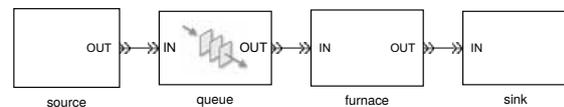


Figure 3: Simple discrete event-based production line model with a furnace component in SimEvents

Figure 4 shows the internal composition of the furnace component according to the introduced layered structure. It consists of the three subsystems: (i) MF for the material flow; (ii) PC for the process control; and (iii) PP for the process physics. The subsystems have to communicate in several ways:

- MF receives parts from the external input port and sends to the PC the number of parts that have entered the MF. When the batch size is reached the PC starts the physical process (PP) which models the different manufacturing phases during the operation of the furnace.
- When the last manufacturing phase (“unload”) is finished, the PC sends a “leaving” signal to the MF, which accordingly forwards the processed parts to its external output port.
- During the operation of the furnace the PC signals the current manufacturing phase to the PP,

which adapts the internal heat flows accordingly. This can mean changing the supplied heat between the heating and holding phases or increasing the losses due to the doors being open while loading or unloading.

- In return the PP sends the current temperature values of the furnace and the parts to the PC, which uses them to determine whether the heat-up phase or the optional final cool-down phase is complete.

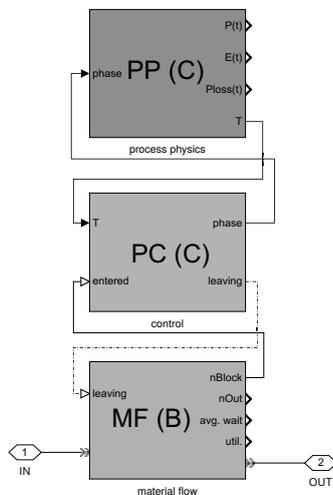


Figure 4: Layered structure of the furnace compo

Each layer has its specific set of parameters and quantities: The MF subsystem defines the batch and logistic properties such as average waiting and machine utilization; the PC subsystem gets heating program and outputs the time in the different phases. The PP system needs a lot of physical parameters for the calculation of the heat flows and provides the power requirements during the process as well as the total energy consumption.

To prove the concept and provide some concrete examples a basic library has been implemented (Fig. 5), containing several subsystems for each layer, which use different modeling methods or have varying levels of detail. They are labeled with the two letters denoting the layer and a third in brackets giving the complexity in ascending order, with (A) being the simplest model:

- The basic material flow subsystem MF(A) uses only a simple server, while MF(B) explicitly contains an input tray, where the batch is compiled.

- The simplest process control subsystem PC(A) uses a simple SimEvent model to describe one pass through the four basic phases (ignoring idle and cool-down states). PC(B) adds a loop to allow a repetition of heat-up and hold phases according to its heat program parameter, implemented as a SimEvents model. Additionally, PC(C) adds a cool-down phase, but because of the more complex control logic it is implemented using the state machine approach with Stateflow.
- The process physics subsystem PP(A) uses only the internal oven temperature and a simple formula for the global losses, while PP(B) adds the temperature of the parts, the heat transfer between oven and parts and additional losses during the load and unload phases. Both use standard Simulink methods to implement the corresponding differential equations. Finally, PP(C) employs physical modeling which is modeled using Simscape for the same physical processes as PP(B). This makes the physical model structure more transparent and easier for engineers to expand.

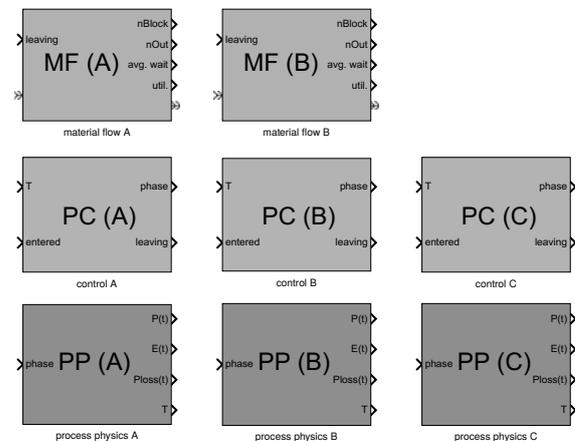


Figure 5: Library with subsystems to compose different variants of a furnace component

6 Implementation details

In the following, three different variants of a furnace component model will be described in more detail: (i) the very basic model ovenBAA – the letters stand for the composition of MF(B), PC(A) and PP(A) components; (ii) a medium complex model ovenBCA; and

(iii) the most complex variant ovenBCC. This section is devoted to the implementation of the components using the MATLAB/Simulink tools, and we will discuss some simulation results.

6.1 Material flow

All of the model examples considered here start with the subsystem MF(B) for mapping the internal material flow. The model structure of subsystem MF(B) (Fig. 6) consists of two simple servers: the first (N-Server) for the input tray; the second (N-Server) for the proper. Both hold incoming entities (i.e. parts) until the given batch size, until their succeeding gates open

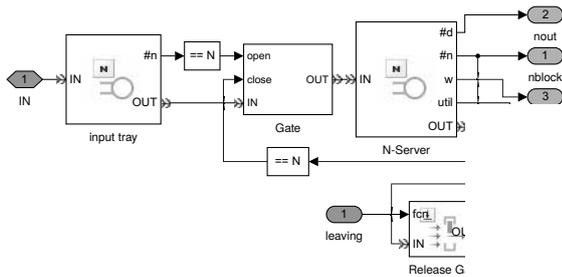


Figure 6: Discrete event-based model of MF(B) using SimEvents

6.2 Process control

The process control subsystem starts when the number of parts that have entered at the material flow layer is equal to the batch size. In subsystem PC(A) implemented using SimEvents, this leads to the creation of a control entity that passes through a sequence of servers denoting the different phases. In this model, all servers simply have a fixed processing time, only the heat-up phase is different. Here the entity is held in the server until the current oven temperature, which is computed in the upper physics layer, has reached the given temperature T_{set} . After the unload phase, the control entity is destroyed and the wake-up signal (trigger) is created, which opens the Release Gate at the material flow layer.

The much more elaborate subsystem PC(C) (Fig. 8)

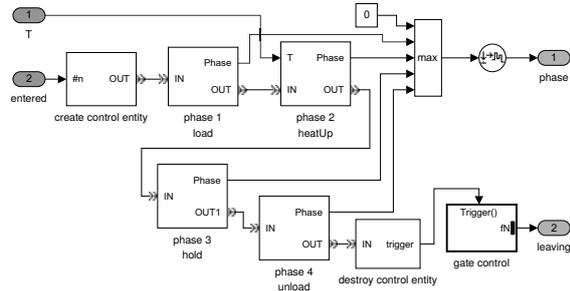


Figure 7: Discrete event-based model of PC(A) using SimEvents

incorporates all six phases, pictured in Figure 2, as well as possible replications of the heat-up and hold phases. It is a hybrid DEV&DESS model based on the state machine approach, but including continuous state event handling. It is mainly implemented using Stateflow. The two auxiliary subsystems create events when the waiting time has changed or the heat-up or cool-down temperature has been reached. The model PC(C) is essentially an adapted version of a model described in [9].

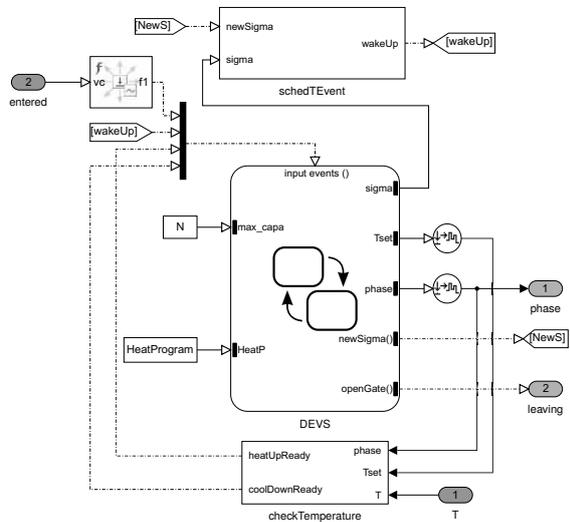


Figure 8: Discrete event-based model with continuous state event handling of PC(C) mainly using Stateflow

6.3 Process physics

The basic process physics subsystem PP(A) (Fig. 9) uses a simple power balance to compute the change of

the oven temperature T_o :

$$C_o \dot{T}_o = P_{heat} - P_{loss},$$

where C_o is the total heat capacity of the oven. power loss is computed with Newton's simple la cooling:

$$P_{loss} = k_a(T_o - T_e),$$

where T_e is the temperature of the surroundings k_a a constant that subsumes all convective and ductive processes. The heating power is assume be constant during the heat-up phase and to match losses during the hold phase:

$$P_{heat} = \begin{cases} P_H & \text{heat-up} \\ k_A(T_o - T_e) & \text{hold} \end{cases}$$

For the computation of the total power demand of the

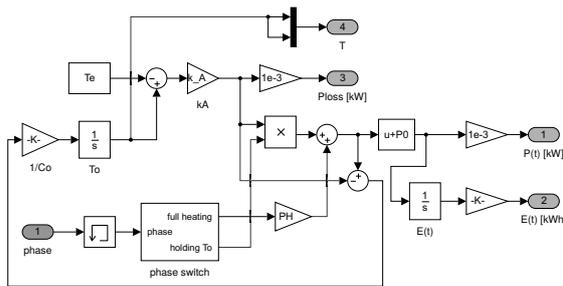


Figure 9: ODE-based model of PP(A) using Simulink

The subsystem PP(C) (Fig. 10) uses physical modeling based on Simscape to incorporate much more physical details, such as convection and radiation from the oven to the load and – during load and unload phases – to the environment. The governing differential equations are not built up explicitly here; instead, the physical components such as heat capacities and various kinds of heat flow are represented directly, which makes the physical structure of the model much clearer.

6.4 Implementation pitfalls

The extension of Simulink by several packages is necessary to make a multi-paradigm model possible, but it also leads to small inconsistencies, which have to be overcome. A minor nuisance here is the large number of different ways used to express values and signals in the packages used: Simple scalar values can be

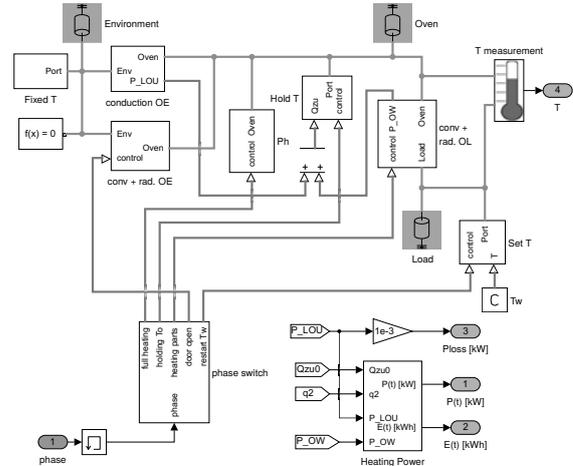


Figure 10: Physical model (DAE) of PP(C) using Simscape

time-based in Simulink, event-based in SimEvents or physical-valued in Simscape. An event can be defined by a sample time hit, a rapid value change (“edge”), a special trigger signal or a function call.

To clearly define a common interface for the layer components, their external values are fixed in the following way: The temperature and phase values are time-based; the number of parts is event-based; and the “leaving” signal is a function call. If the internal implementation of a block generates differing types, then one has to use one of the many converter blocks to get the proper kind of signal, e.g. the round Event to Timed Signal blocks that can be seen in Figures 7 and 8.

This problem is especially annoying in the physical modeling environment. The large number of necessary converters and reference points clutters the model and destroys the clear physical structure. Hiding them in subsystems is an obvious way to regain an ordered visible representation of the underlying physics model.

7 Simulation results

In a simple example experiment 24 parts were processed using a batch size of six. After the heat-up phase their temperatures were held first to 400 °C for 40 minutes, then to 800 °C for 30 minutes. Figure 11 shows some simulation results of the model variant ovenBCC: the temperatures of the furnace and parts, the power consumption of the furnace, the accumu-

lated used energy and the current internal manufacturing phase, each as functions over time.

The results are similar to those presented in [9], but ovenBCC incorporates more details, especially of the physical model layer. The other two example models ovenBAA and ovenBCA basically reproduce the results from [9], since they are based on the same physical model assumptions. For comparison, Table 1 shows the total energy use E_1 at the exit of the last part and E_2 at the end of the simulation; Figure 12 displays the power consumption for the three models.

Model	E_1 [kWh]	E_2 [kWh]
ovenBAA	71.3	108.8
ovenBCA	69.0	106.6
ovenBCC	82.6	117.6

Table 1: Comparison of the energy consumption

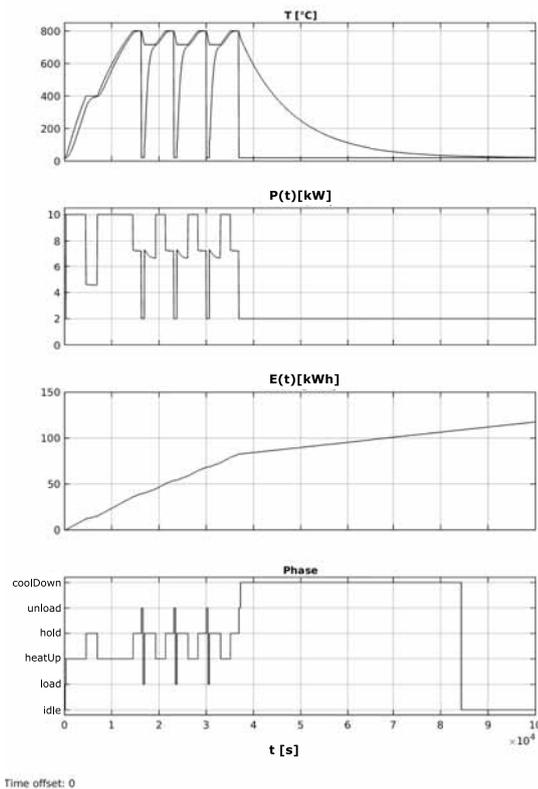


Figure 11: Simulation results of model variant ovenBCC

The higher energy needs in variant ovenBCC are due to the heating of the parts, which enter the oven with the low temperature of the environment – an effect that

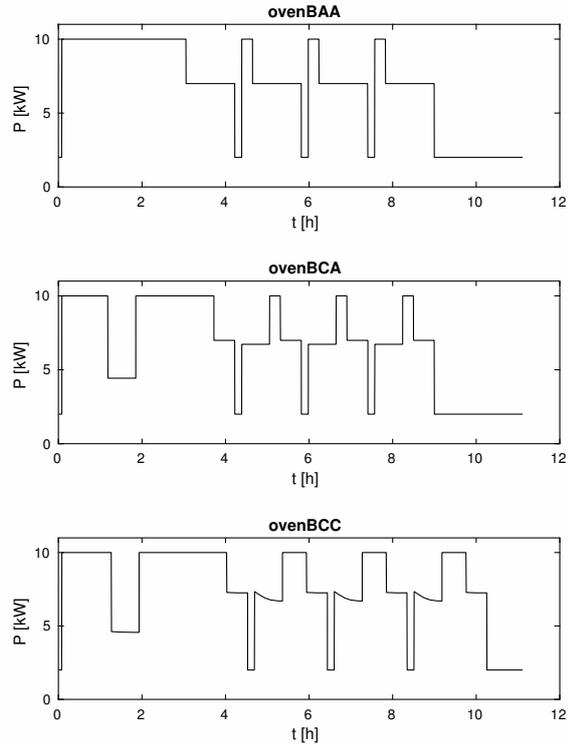


Figure 12: Comparison of the power results

has not been taken into account in the other models. The small difference between the two simpler model variants results from the different timing of the phases, as can be seen from Figure 12. It is interesting to note that the total energy results only differ by 10% - 15%. If this level of accuracy is sufficient for the question at hand, e.g. for a global assessment of a complex production line, then one can use one of the simple physics models. However, Figure 12 shows that for a detailed examination of the power needs during the individual phases one has to stick to the complexities of model variant ovenBCC.

How much do we pay for the additional level of detail in the complex model? For a comparison of run times, the number of parts was raised to 600 and three runs were performed for each model. To get rid of loading and compile times, the means of only the last two results were taken, leading to the values in Table 2.

The results show that the complex physics layer using Simscape costs a factor of two in execution time, which can be reduced to a factor of 1.6 by replacing the physical modeling with standard ODE methods using Simulink (ovenBCB). The small value for ovenBCA comes as a surprise, since its process layer

Model	Run time [s]
ovenBAA	25.1
ovenBCA	20.7
ovenBCC	49.8
ovenBCB	39.3
ovenBBA	27.7

Table 2: Comparison of run times

is definitely more complex than that of ovenBAA. The solution to this puzzle is given in the last line of the table: The subsystems PC(B) and PC(C) have nearly the same complexity, but PC(C) uses Stateflow instead of SimEvents. Apparently, the implementation of SimEvents has some potential for optimization.

8 Conclusion

The introduced layered model structure supports the modular modeling of complex components, such as production facilities, using different methods for specifying the dynamic behavior. Thereby, it provides a natural approach for implementing compatible models with different levels of detail.

Using the example of a model library for an industrial furnace it has been shown that model components of production facilities can be usefully structured in three layers. The first layer maps the internal material flow. It delivers an external input and output interface for connecting with other facilities in a manufacturing process chain model and an internal interface for communication with the second layer. The dynamic of this layer is described as transaction-oriented, a special kind of entity-based discrete event modeling. In the second layer local process control operations are modeled. These operations have mainly discrete or event-based dynamics and are usefully described using transaction or state-based methods. However, depending on the specific process operations, modeled in the third layer, it could be necessary to detect and handle state events of continuous values. Additionally, the process control layer provides event-based control inputs for the first and third layers. The dynamic behaviors of specific process operations can vary significantly and particularly depend on the level of detail. In the case of the industrial furnace the process operations have been described by (i) simple ordinary

differential equations and (ii) complex differential algebraic equations using physical modeling methods.

Using the model library, various models for the industrial furnace using different modeling techniques have been implemented and compared relating to their simulation results and runtimes. Until now the experiments have been focused on the furnace as a single production facility and its energy consumption. In the next step the furnace component should be examined in the more complex context of process chains and then further performance indicators should be considered.

Acknowledgements

A. S. and Th. P. express their sincere thanks to the German Research Foundation DFG for funding the research project no. PA 631/2-2

P.J. is grateful for the hospitality extended to him by Tom Schramm and his colleagues at the Department of Geomatics, HCU Hamburg.

References

- [1] D.P.F. Möller. *Mathematical and Computational Modeling and Simulation – Fundamentals and Case Studies*. Springer Pub., 2004.
- [2] E. Lefeber, J.E. Rooda. *Modeling and Analysis of Manufacturing Systems*. In: Handbook of Dynamic System Modeling, P.A. Fishwick, CRC Press, p. 34_1-34_20, 2007.
- [3] C. Eisele. *Simulation-based optimization of the electrical consumption of metal-cutting machine tools (in German)*. Dissertation Thesis, TU Darmstadt, Germany, 2014.
- [4] N. Weinert. *An approach for the planning and operation of energy-efficient production systems (in German)*. Dissertation Thesis, TU Berlin, Germany, 2010.
- [5] R. Larek. *Resource-efficient design of manufacturing process chains by simulation and numerical optimization (in German)*. Dissertation Thesis, Univ. Bremen, Germany, 2012.
- [6] S. Schrems. *A method for the model-based integration of machine-specific energy demand in*

- production planning (in German)*. Dissertation Thesis, TU Darmstadt, Germany, 2014.
- [7] T. Peter, S. Wenzel. *Simulation-based Planning and Evaluation of Energy Efficiency for Production Systems in Car Manufacturing (in German)*. Simulation in Production and Logistics, Fraunhofer Pub., Stuttgart, Germany, 535-544, 2015.
- [8] B. Zeigler, T. G. Kim, H. Praehoffer. *Theory of Modelling and Simulation*. Academic Press, San Diego, 2nd ed. 2000.
- [9] A. Schmidt, T. Pawletta. *Hybride Modellierung fertigungstechnischer Prozessketten mit Energieaspekten in einer ereignisorientierten Simulationsumgebung*. Proc. ASIM 2014 - 22. Symposium Simulationstechnik, Berlin, p. 109-116, 2014.
- [10] The MathWorks. *Simulink: Simulation and Model-Based Design*. Online: www.mathworks.de/products/simulink/ (called 2015-11-11).
- [11] The MathWorks. *SimEvents: Model and simulate discrete-event systems*. Online: www.mathworks.de/products/simevents/ (called 2015-11-11).
- [12] The MathWorks. *Stateflow: Model and simulate decision logic using state machines and flow charts*. Online: www.mathworks.de/products/stateflow/ (called 2015-11-11).
- [13] The MathWorks. *Simscape: Model and simulate multidomain physical systems*. Online: www.mathworks.de/products/simscape/ (called 2015-11-11).
- [14] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal. *Pattern-Oriented Software Architecture, Vol. 1: A System of Patterns*. John Wiley & Sons, New York, 1996.
- [15] D. Harel. *Statecharts: A Visual Formalism for Complex Systems*. Science of Computer Programming, 8, pp. 231-274, 1987.
- [16] P. A. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3*. Wiley & Sons, New York, 2015.
- [17] W. D. Kelton, R. Sadowski, N. Zupick. *Simulation with Arena*. McGraw-Hill, New York, 6. ed. 2014.
- [18] S. Bangsow. *Tecnomatix Plant Simulation: Modeling and Programming by Means of Examples*. Springer, Berlin, 2016.
- [19] H. Elmqvist, F. Gaucher, S. E. Mattsson, F. Dupont. *State Machines in Modelica*. Proc. 9th Int. Modelica Conference, Munich, p. 37-46, 2012.
- [20] V. Sanz, A. Urquia, F. E. Cellier, S. Dormido. *System modeling using the Parallel DEVS formalism and the Modelica language*. Simulation Modelling Practice and Theory, 18, 998-1018, 2010.
- [21] P. Junglas. *A Modelica library for transaction based modeling*. Work in progress.
- [22] V. Heuer, K. Löser. *Energetische Optimierung von thermochemischen Vakuumprozessen und Anlagen in der Großserie*. Elektrowärme International, 69, pp. 261-268, 2011.

Exploring the Advantages of Multi-Method Modelling in the Use Case of a Large Socio-Technical Infrastructure System – The Airport City

Barbara Glock¹, Niki Popper¹, Felix Breitenecker²

¹dwh GmbH, dwh Simulation Services, Neustiftgasse 57-59, Vienna 1070

²TU Wien, Institute for Analysis and Scientific Computing, Wiedner Hauptstrasse 8-10, Vienna 1040

barbara.glock@dwh.at

Modelling large socio-technical infrastructure systems tends to be tricky. In the past either one “best fitting” modelling technique was used to model the system or a small part of the system was modelled. This lead to many tradeoffs, where the chosen modelling method got to its limits: by modelling a small part, effects from outside were ignored and modelling the whole system with one method lead to either much detail - where micro-based methods were applied - where it was not required which further lead to high computation times. Using a macro-based approach lead to less detail where it would have been needed. Different parts of the Airport City are modelled with the best fitting modelling technique. These parts are researched for their coupling mechanisms to model the whole system and see how effects in one part evolve and pass to the next subsystem. Currently an agent based model of the landside with a modal split of passenger arrival, a Discrete Events terminal model, a multi-method agent-based model with an integrated System Dynamics model representing the retail area of an airport and an agent based model of the airside are in development and their advantages and disadvantages are being explored.

1 Introduction

The planning of big infrastructure developments is getting more challenging due to more complex structures and an increased number of construction standards. Large infrastructure systems can be decomposed into **subsystems** that are somehow interconnected and correspond on different levels with each other. This makes an analysis more difficult. Furthermore, there are **different views** on the system that need to be addressed and satisfied: stakeholders, planners, consumers, decision makers, etc. On the one side questions addressing profit arise, and on the other side consumers want a specific level of quality. Also ecologic aspects need to be kept in mind. In general this calls for modelling and simulation. These large systems, broken into pieces, consist of different subsystems with much more detail, each of them with its own dynamic effects. This gives an opportunity for **multi-method modelling (MMM)** to use all advantages of each method in combination.

1.1 Related Work

Applying multi-method modelling techniques to model large infrastructure systems hasn't been properly researched yet, since the focus of most modellers has been in increasing know-how on the singular methods themselves and additionally the computational power hasn't been that advanced in the past

then it is nowadays to meet this challenge. Most literature focuses on comparing the available modelling method in their research area to find the best fitting [1] or discuss what the advantages and disadvantages are [2], but also how a model in one paradigm can be transformed into a model of another paradigm [3]. Especially under what circumstances, mathematically described, this is allowed. But, as mentioned before, rather view publications try to combine the advantages of different modelling paradigms: **System Dynamics (SD)**, a rather old approach, developed in the 1950s by Jay W. Forrester [4] has been researched very well [5]. **Discrete Events Simulation (DES)** [6] has been researched very well too. Literature on so-called *hybrid* models, where these two modelling paradigms are being combined, coupling continuous and discrete models, can be found [7]. When it comes to **agent-based modelling (ABM)**, this being a rather new approach [8], modellers are still struggling with finding a “right” definition – different definitions exist! [9] -, than thinking of how to combine for example agent-based methods with other modelling paradigms like System Dynamics. Researching literature revealed some works on first attempts and implementations where agent-based models are coupled with System Dynamics models [9] and one paper on a *classification* attempt for combining SD and ABM was found [10].

1.2 Aim

The aim of this dissertation project is to research different multi-method modelling techniques for modelling large socio-technical infrastructure systems as airports, to **reveal their advantages and disadvantages** and to **give some guidance on how to apply these different modelling paradigms**. Macro-based and micro-based modelling paradigms are different in view and offer different advantages. *The one modelling method's advantage is the other ones disadvantage. So why not combine their advantages?* The focus will lie on ABM combined with SD and on DES combined with SD, since these two different modelling paradigms are a little bit trickier to combine. The use case where this will be tested is the Airport City, since this is a large, decomposable system and yields a lot of possibilities to apply multi-method modelling techniques.

1.3 Structure of the Paper

In section 2 an overview on the researched modelling techniques will be given. These are System Dynamics, agent-based modelling and Discrete Events Simulation. In section 3 the classification by Swinerd and McNaught of 2012 will be summarized as it is used as starting point for a possible further classification. In section 4 the modelled and partially implemented submodels of the Airport City will be described together with their coupling methods and further planned submodels. Finally in section 5 this paper will conclude with an outlook on further work within this project.

2 Modelling Methods used for MMM

Multi-method modelling with the following three modelling methods are researched and used in the application of airport planning, because these methods are, according to literature, used most often in this area and are also applicable to model the subsystem to answer specific research questions.

2.1 System Dynamics

In this macro-based modelling paradigm the point of view is from above, where only aggregated levels are looked at. It was developed in the 1950s by Jay W. Forrester, who applied it first in management systems [4]. He then transferred this methodology to social systems. Nowadays diverse literature on *System Dynamics* and *Systems Thinking* exists [5]. A SD model consists of six basic elements, as seen in table 1.

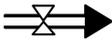
Element	Representation	Description
Stocks (Levels)		describe the state of the system at each time and represent aggregates
Flows		describe the changes of the stocks; are basically auxiliaries and are only allowed between stocks or between stocks and sinks/sources
Parameter		are constants and represent rates on which changes of stocks are dependent
Auxiliary		are helpful for a better understanding of the model and represent algebraic equations
Sink / Source		describe the boundaries of the system
Links		describe the causalities of other elements

Table 1. Basic elements of System Dynamics.

An SD model basically represents a set of differential and algebraic equations. A simple example can be seen in figure 1 and related equations in equations (1) and (2).

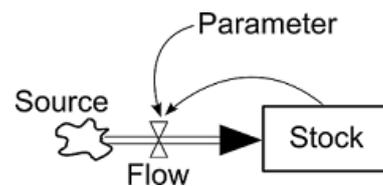


Figure 1. Simple stock and flow structure of SD.

Possible equations for time t :

$$\frac{d Stock(t)}{dt} = Flow(t) \tag{1}$$

$$Flow(t) = Stock(t) \cdot Parameter \tag{2}$$

The dynamics of the system emerge from **causal links** of the modelled variables that often form feedback loops. Such loops can be balancing or reinforcing driving the dynamics of the system.

Application areas are amongst others economics, health care and policy design.

2.2 Agent-based Modelling

Agent-based modelling is a rather new approach and several properties of agents can be found in literature [9]. A selection of those are:

- **Proactiveness, purposefulness:** ability to take the initiative in order to achieve goals
- **Situatedness:** an agent is embedded in its environment and senses and acts on it
- **Reactiveness, responsiveness:** ability to react in a timely fashion to changes in the environment
- **Autonomy:** ability to control own actions and internal state
- **Social ability:** ability to interaction and communication with other agent, sometimes even awareness of other agents
- **Anthromorphy:** having human-like attributes like beliefs and intentions
- **Learning:** ability to increase performance over time based on previous experience
- **Mobility:** ability to move around in the simulated physical space, sometimes even between different machines
- **Specific purpose:** designed to accomplish well-defined tasks

Not every agent-based system has all of those properties [9] and depending on the agent's purpose one should not speak of agents and non-agents, but speak of a *continuum of agency*: The more agent characteristics an entity possess and the more developed those are, the higher the degree of agency it has.

2.3 Discrete Events Simulation

Discrete Events Simulation models are similar to agents, but the entity modelled here is not like an agent autonomous. It is passively led through the system instead. Furthermore, changes in the state of the system happen due to events at discrete points in time [11]. Between two consecutive events the state remains unchanged.

Basic elements of DES are:

- **Entities:** have discrete properties and can be arranged in sets or lists
- **Events:** is an instantaneous happening that changes the state of the system. Events are arranged in an event list and are scheduled by using event notices that provide different information like type or time of event.

- The **clock:** is a global variable that represents the simulated time. There are activities (time spans of certain length already know by simulation start) and delays (time spans of uncertain length like waiting time of a passenger in a queue). The clock can be forwarded in different manners.
- **Scheduler:** handles the event list, forwards the events to the event processing routine, also re-schedules events if necessary and updates the clock.

This kind of modelling is mostly used in logistics and transportation.

3 Classification of MMM

Researching literature revealed that few attempts in multi-method modelling were done, but only one paper was found that tried to classify models that used ABM and SD modelling to model the whole system. There are three proposed categories [10].

Two coupled submodels are called **interfaced** if they have some point of interaction like communication between elements. The submodels run **alternating and independently** (see figure 2). There is no direct feedback between the submodels.

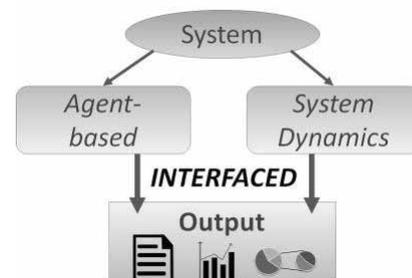


Figure 2. Interfaced multi-method model.

Two submodels are **sequential** if one submodel needs the output of another as input (see figure 3).

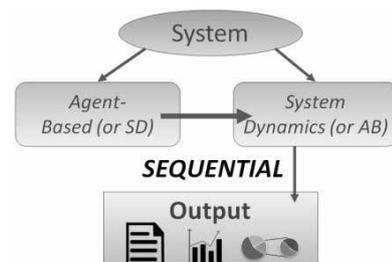


Figure 3. Sequential multi-method model.

In an **integrated** multi-method model the submodels interact with each other in some way, as can be seen in figure 4.

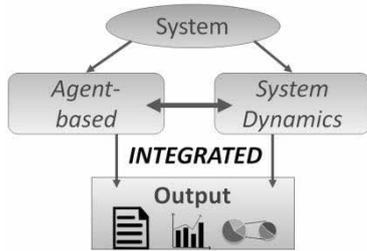


Figure 4. Integrated multi-method model.

According to Swinerd and McNaught [10] there are three ways to model an integrated design:

- **Agents with rich internal structure:** within each agent an SD model exists (see model on retail area). One can think of the SD model within an agent as the agent’s “brain” that “tells” him what to do in a dynamic way. Here, influences from both sides can be considered: ABM submodel passes information to the SD submodel and vice versa.
- **Stocked agents:** a level within an SD model is used to bind an aggregate measure of an agent-based model. This could be an SD submodel that calculates production costs on car sales and the influence of fuel price on consumer choice of vehicle technology where consumers are modelled in the AB submodel. Here only influences from SD to ABM are modelled, but not the other way round.
- **Parameters with emergent behaviour:** a parameter of an SD model is calculated by an agent-based model. An example is an agent-based submodel where demographic development is modelled with agents together with their individual properties (age, sex, maybe socio-economic factors). Out of these properties the value of a parameter for a coupled SD submodel is calculated.

There is a fine line between the classes of multi-method models. The modeller has to decide what fits best. It is also dependent on where the system boundaries lie. This classification approach is very useful as a starting point in researching multi-method modelling, since it is also applicable not only for SD and ABM modelling but also for other modelling methods.

4 Use Case: Airport City

The Airport City gives a good example of a **large socio-technical infrastructure system** to model, since on the one hand passenger demand forecasts predict an increase in the long term growth of transported passengers of avg. 4% per year [12] and on the other hand the system can be decomposed to smaller subsystems, that are somehow interconnected to each other on different levels. In figure 5 a selection of important areas of the Airport City can be seen.

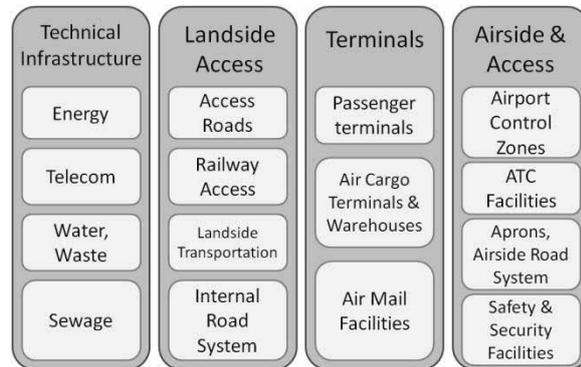


Figure 5. Selection of areas in the Airport City. (Source: adapted from company *AI-MS Aviation Infrastructure Management Systems*)

Also different views need to be satisfied here as well: planners, consumers, and stakeholders. Currently, a terminal model modelled with Discrete Events and a model of the retail area as an integrated ABM-SD model with *rich internal structure* have been developed. Furthermore, in development are an agent-based model of the landside and the integration of a network-based agent-based model of the airside. Still planned is a SD model to calculate costs/profit and CO2 emission of airplanes. The advantages and disadvantages together with the possible coupling mechanisms are being researched. The described models are implemented in AnyLogic 7 [13].

4.1 Terminal Model with Discrete Events

The terminal area is after the landside area the second area where departing passengers go to. The processes going on include check-in, security controls, passport controls and proceed to gate through retail area [14]. These processes are dependent on specific features of the passenger, like if he is business or tourist passenger (only hand luggage or not), or what his destination is (if within Schengen the passenger can proceed without passport control) or if he is handicapped or not. This submodel also includes transfer passengers,

meaning passengers arriving at the airport by plane, going through passport control if necessary and proceeding to gate after going through retail area. This circumstance shows on the one hand, that this sub-model gets input from the landside as well as from the airside and if some delays or other effects happen in the parts of the airport not represented in the terminal submodel it has an effect on the terminal sub-model.

The research question in this model is if resources like personnel and number of open counters is enough at each time to maintain the quality standards measured in waiting time of passengers. This being a simple server-queue question is modeled best using Discrete Events with counters and personnel being resources and passengers being entities.

A first version of a DES model has already been implemented, as seen in figure 6. This model includes the basic servers in such a process (check-in, security, passport control and transfer) and distinguishes between tourist and business passengers. In AnyLogic simple blocks for creating (sources), processing (server), and queuing (queue) entities are used. Resources are created via a (scheduled) resource pool. Here, sources and sinks (Exit) build the interface to an adjacent submodel where entities are led through the system.

An agent based model modelling the landside can be coupled by this rule: every time an agent (passenger) exits the landside model, an entity (it is possible to pass other information as well) in the DES model is generated. On the other side the exit can be seen as an interface as well. Passengers proceed to the gate through the retail area. Every time an entity enters the sink in the DES model an agent will be created in the ABM model of the retail area. This connection is from type interfaced.

4.2 Retail Area with Integrated ABM - SD Model

The retail area is economically seen a very important part of the airport since most of the profit is gained here. The retail area is a shopping area after having passed the controls in the terminal where passengers go through when they proceed to the gate to depart. One main research question in this area is to maximize profit by guaranteeing a specific level of quality for passengers like a short way to the gate or attractive sales. Here also spatial information is needed.

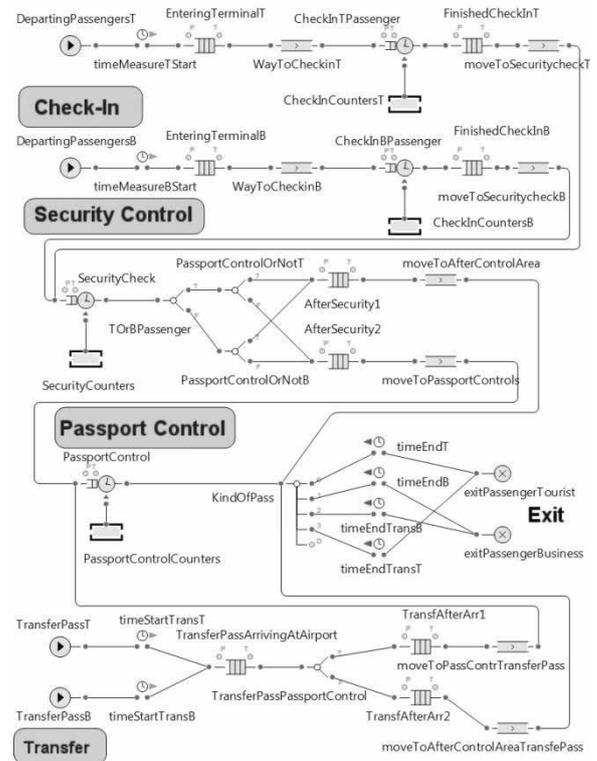


Figure 6. Discrete Event Simulation of the terminal.

The processes are dependent on an individual's behaviour: he/she reacts to the environment and other passengers; therefore an ABM submodel that includes spatial information (map of the shops) is used. The environment is the retail area with the shops. The inner parts of an agent should be dynamically changing, since this is more realistic: next to some individual parameters, the agent consists of an SD submodel that models the need to eat and the need to buy other things. In this case the retail area submodel is itself an **integrated model of agents with rich internal structure**. Each agent follows rules like:

- proceed to gate in time
- if hungry and still time to departure, then look for eating store and eat
- in dependence of attractiveness of store and in dependence of estimated income buy something if there is still time to departure and the need to buy something exceeds a specific threshold

These rules always take into account the by the SD model calculated *need* to buy something. This means there is communication from the SD module to the agent based module (tell him where to go). In return the need is dynamically calculated by the SD module

by using individual information from the agent (age, gender, time until departure), but also using information from the environment of the AB module, like the attractiveness of the store that has some “basic attractiveness” and furthermore is calculated by the number of people inside (if no people are inside it may be something wrong with it, if too much people are inside it is overcrowded).

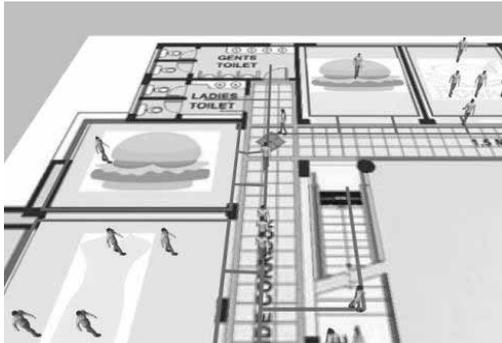


Figure 7. Integrated multi-method model of retail area - agents with rich internal structure (agents).

A first version of this submodel has been developed (by M. Obermair and B. Glock), as seen in the 3D version of the animation in figure 7. A network is applied to a ground floor and passenger agents interact on this plan with the shops trying to satisfy their goals that are modelled with SD, which can be seen in figure 8.

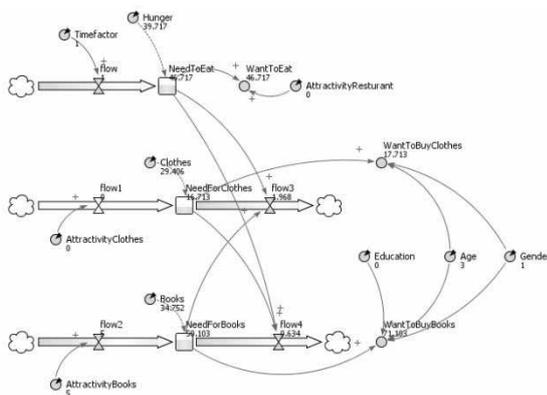


Figure 8. Integrated multi-method model of retail area - agents with rich internal structure (SD model within each agent).

Clearly, this multi-method approach allows agents to develop their needs dynamically and not by discrete rules, which gives a more realistic picture of the world.

4.3 Airside

The airside of the airport is the part where diverse processes take place that deal with outgoing and incoming planes, like it is shown in figure 9.

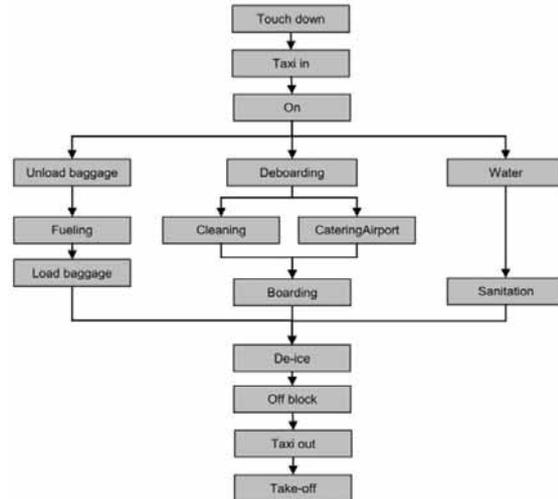


Figure 9: Ground handling process [15].

The so called **ground handling processes** include all processes around the plane. After touchdown (landing of the plane), the taxi arrives to get the passengers. After that the unloading and water refill starts. There are some limitations like cleaning and catering have to start after disembarking or fuelling after unloading luggage that need to be considered as well. In this submodel the spatial context plays an important role since travelling times contribute to quality measurements for passengers and the calculation of optimizations regarding the ground handling process itself. The amount of flights is increasing and space on the airside where passengers can board (directly through the gate via boarding bridges or on the apron) is limited. Therefore, in this case an **AB submodel** is used with a given **network** on which the agents can operate, as seen in figure 10.

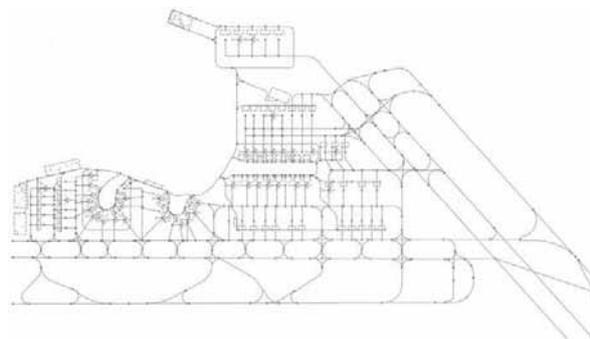


Figure 10: Example of an airside network of the Vienna International Airport.

Different types of agents interact with each other on an environment like the network (selection):

- Planes
- Mobile stairs
- Catering vehicle
- Belt loader
- Baggage cart
- Container/pallet dolly
- Container loader
- Tractor, ...

They have one overall goal to get the plane as soon as possible up in the air again.

4.4 Further Planned Submodels

Next a submodel modelling the landside with agents where the type of agents is different (cars, and not passengers) is planned and a submodel modelling the CO₂ emission (with SD) as an integrated design is planned, where the agent-based and Discrete Event submodels have an influence on parameters of the SD model (classification of parameters with emergent behaviour).

So, also questions addressing different types of agents and their coupling mechanisms can be addressed as well as questions regarding the coupling of those two modelling paradigms the other way round: in the retail area an overall agent-based model existed with an SD model integrated and here an SD model acts as overall model with agents having an influence on this parameters.

4.5 The Big Modelling Picture

Summarizing the proposed submodels as seen in figure 11 there is a terminal submodel with DES that interacts with the retail area submodel modelled with ABM, which itself is a multi-method model as the “brain” of each agent is a SD model. The retail area submodel will be connected to the airside submodel. A submodel of the landside will be integrated. Passengers proceed not online from landside to the airside, but also from airside to the landside, if arriving. Transfer passengers are included as well. So, we can see that those submodels are from integrated form all along, because these submodels have to exchange information (e.g. the agent or entity being passed on) as well. The further planned SD submodel, calculating profit and CO₂ emission requires information from the other submodels. Somehow, this SD model will give (delayed) some information back to the

submodels (dotted lines), since the effect of the CO₂ development will have an effect on the structures of the airport implemented with the submodels.

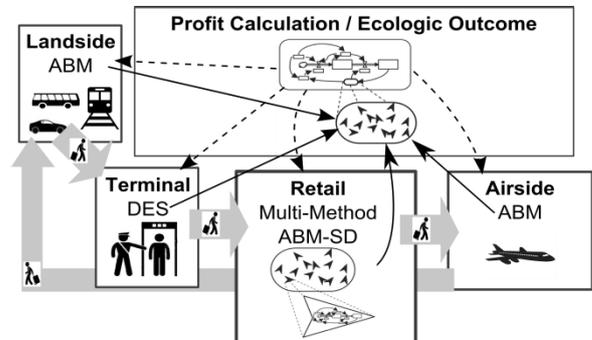


Figure 11: The coupling schema of the modelled and planned submodels

The advantages of this multi-method modelling technique clearly are that the model gets more realistic by also including individual’s properties. The disadvantage is that it gets more complex. On the other hand due to the modular architecture verification and validation will be, compared to a whole ABM model, a little bit easier.

5 Conclusion and Outlook

Using multi-method models is getting more and more important [16], since large infrastructure systems like airports get more complex and larger. Errors by using only one method for the large system can accumulate over time resulting in difficult decision making. By using different (best fitting) modelling methods for different subsystems and utilizing all their advantages a more realistic presentation of the model can be created. This also makes communication to decision makers easier and accumulating errors are eliminated to some extent. Furthermore, calculation times of the simulation models can be reduced as well.

Next (interesting) steps in this project include building and integrating a CO₂-emission submodel and a profit calculation model that interact with the former described submodels and an airside and a landside model.

6 References

- [1] H.J. Scholl. *Agent-Based and System Dynamics Modelling: A Call for Cross Study and Joint Research*. In Proceedings of the 34th Annual Hawaii International Conference on System Dynamics, Hawaii, USA, 2001.

- [2] T. Lorenz, A. Jost. *Towards an Orientation Framework in Multi-Paradigm Modelling*. In Proceedings of the 24th International Conference of the System Dynamics Society, Nijmegen, The Netherlands, 2006.
- [3] P. Einzinger. *A Comparative Analysis of System Dynamics and Agent-Based Modelling for Health Care Reimbursement Systems*. Dissertation, TU Wien, Austria, 2014.
- [4] J. Forrester. *Industrial Dynamics*. Productivity Press, Cambridge, MA, 1961.
- [5] J. Sterman. *Business Dynamics – Systems Thinking and Modelling for a Complex World*. McGraw-Hill Education Ltd, USA, 2000.
- [6] J. Banks, J. Carson. *Discrete-Event System Simulation*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1984.
- [7] B. Jovanoski, R. Minovski, S. Voessner, G. Lichtenegger. *Combining System Dynamics and Discrete Event Simulations – Overview of Hybrid Simulation Models*. 2012.
- [8] C. Macal, M. North. *Tutorial on Agent-Based Modelling and Simulation Part 2: How to Model with Agents*. In Proceedings of the 2006 Winter Simulation Conference, Monterey, California, p.73-83, 2006.
- [9] N. Schieritz, P. Milling. *Modelling the Forest of Modelling the Trees – A Comparison of System Dynamics and Agent-Based Simulation*. In Proceedings of the 21st International Conference of the System Dynamics Society, New York City, USA, p.1-15, 2003.
- [10] C. Swinerd, K. McNaught. Design Classes for Hybrid Simulations Involving Agent-Based and System Dynamics Models. Elsevier Simulation Modelling Practice and Theory (25), p.118-133, 2012.
- [11] B. Zeigler, H. Praehofer, T. Kim. Theory of Modelling and Simulation: *Integrating Discrete Event and Continuous Complex Dynamic Simulation*. Academic Press, San Diego, 2000.
- [12] R.De. Neufville, A. Odoni. *Airport Systems – Planning, Design and Management*. 2nd Edition, United States, The McGraw-Hill Education LLC, 2013.
- [13] AnyLogic. www.anylogic.com. 2015.
- [14] A. Schulz, S. Baumann, S. Wiedenmann. *Flughafen Management*. Oldenburg Verlag, München, 2010.
- [15] A. Norin. *Airport Logistics – A Case Study of the Turn-Around Process*. Elsevier Journal of Air Transport Management, p.31-34, 2012.
- [16] S. Brailsford, N. Mustafee, S. Diallo, J. Padilla, A. Tolk. *Hybrid Simulation Studies and Hybrid Simulation Systems: Definitions, Challenges, and Benefits*. In Proceedings of the 2015 Winter Simulation Conference, Huntington Beach, California, p.1678-1692, 2015.

ACKNOWLEDGMENTS

This project is a dissertation within *Talente* funded by BMVIT Ministry of Transport, Innovation and Technology and handled by FFG Austrian research Promotion Agency.

Von den Requirements bis zum SerieneCode - Ein durchgängiger Entwicklungsprozess mit Model-Based Design

Jens Lerche¹, Stephan Sieben¹

¹The MathWorks GmbH, Aachen

Jens.Lerche@mathworks.de, Stephan.Sieben@mathworks.de

Quer durch alle Industrien werden Produkte komplexer. Das Zusammenspiel von Mechanik, Elektronik und Software erfordert eine gesamtheitliche Betrachtung der Systeme schon während der Entwicklung.

Um die Kosten und den zeitlichen Entwicklungsaufwand bei hoher Produktqualität zu gewährleisten reicht eine reine Steckensimulation häufig nicht mehr aus. Verifikation und Validierung von Software auf Modellebene gewinnen zunehmend an Bedeutung.

In unserem Vortrag stellen wir den modellbasierten Ansatz zur Gesamtsystementwicklung vor. Dabei durchlaufen wir den Prozess von den Spezifikationen bis zur finalen Produktabnahme und gehen auf typische Herausforderungen ein

Modellbildung und Simulation im Kontext des Systems Engineering

M. Sc. Jan Michael, M.Sc. Jens Holtkötter, Dr.-Ing. Christian Henke, Prof. Dr.-Ing. Ansgar Trächtler

Fraunhofer- Einrichtung Entwurfstechnik Mechatronik IEM

jan.michael@iem.fraunhofer.de

In Zeiten von Industrie 4.0 und Cyber-physikalischen Systemen unterliegt die Komplexität dieser Systeme einer progressiven Zunahme [1]. Um dieser Zunahme zu begegnen und Produkte dennoch effizient und zeit-optimal entwickeln zu können, wird vermehrt auf innovative Methoden gesetzt, welche Entwicklungsprozesse beschleunigen und gleichzeitig eine ausreichende qualitative Absicherung garantieren. Im Kontext des Systems Engineering bildet die modellbasierte Entwicklung ein wesentliches Kernelement.

Vor diesem Hintergrund zeigt dieser Beitrag, wie anforderungsbasiert Simulationsmodelle zu entwickeln sind, um das Kosten-Nutzen Verhältnis effizient gestalten zu können. Es wird anhand eines realen Systems aus dem Smart Home Bereich beispielhaft gezeigt, wie nach [2] verschiedene Modellvarianten mit jeweils unterschiedlichen Anforderungen erstellt und genutzt werden. Insbesondere die Verknüpfung dieser Simulationsmodelle mit den aus frühen Phasen der Systemenspezifikation entstehenden abstrakten Systemmodellen wird gezeigt. Ebenso wird die Ableitung von Anforderungen an die Simulationsmodelle aus den Systemmodellen. Diese Anforderungen an die Simulationsmodelle gilt es dann möglichst genau zu erfüllen.

Ziel bei diesem Vorgehen ist, die Simulationsmodelle ebenso zu spezifizieren, wie es auf der Produktebene geschieht, um schlussendlich Mehr-/ oder Minderaufwand im Bereich der Modellbildung zu reduzieren respektive zu vermeiden. Als Resultat des durchgängigen Vorgehens kann zunächst Zeit durch iterative Optimierungsschleifen gespart werden, was final auch einen monetären Vorteil mit sich bringt und kongruent zu den Ansätzen des Systems Engineering ist.

1 Einleitung

Die Entwicklung technischer Systeme erfolgt heute weitestgehend unterstützt durch den Einsatz der Simulation. Dabei wird in unterschiedlichsten Branchen und Bereichen an Modellen gearbeitet, um die Effizienz der Entwicklung zu. Die 4. industrielle Revolution stellt die Entwicklung jedoch vor neue Herausforderungen. Dieser Beitrag zeigt die Herausforderungen für die gelebte Entwicklung und Lösungen, die diese Problemstellungen bewältigen.

2 Modellbasierte Entwicklung mechatronischer Systeme

Technische Systeme, respektive deren Entwicklung, werden durch die Virtualisierung effektiv unterstützt. Hierbei kommen definierte und etablierte Vorgehen zum Einsatz, welche die Reproduzierbarkeit gewährleisten. Eine dieser Vorgehensweisen ist durch [2] festgelegt und beschreibt eine Entwicklungsmethodik für mechatronische Systeme.

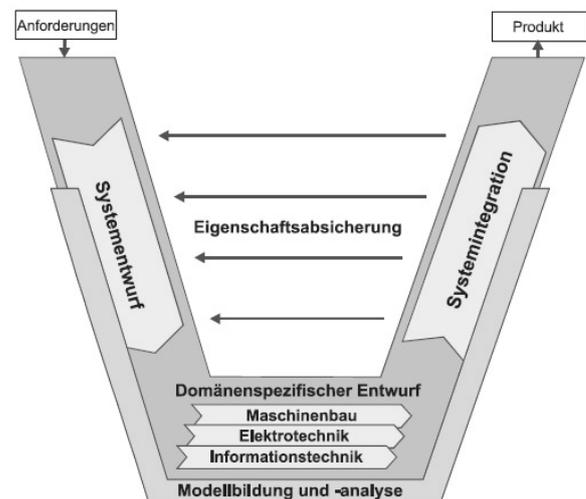


Abbildung 1 Makrozyklus des Entwurfs mechatronischer Systeme

Hier werden zunächst die Rahmenbedingungen definiert, beziehungsweise die Anforderungen, die das zu entwickelnde System erfüllen muss. Auf Basis dieser Anforderungen werden Funktionalitäten erarbeitet und definiert, welche die Anforderungen erfüllen. Im Anschluss an diese Funktionalitäten werden in der

fortgeschrittenen Systementwurfsphase, auch Spezifikationsphase genannt, Lösungselemente evaluiert, die wiederum die zuvor definierten Funktionen erfüllen. In dieser Phase des Prozesses werden bereits erste Simulationsmodelle erarbeitet, welche zur Bewertung verschiedener Prinziplösungen (Kombinationen aus Lösungselementen) herangezogen werden. Im weiteren Prozessverlauf wird dann eine finale Prinziplösung ausgewählt und in dem sogenannten domänenspezifischen Entwurf ausdetailliert. Hier werden die zuvor erstellten Modelle um zusätzliche Details erweitert. Es werden u.a. physikalische Vorgänge abgebildet, detailliertere Abläufe in Form von Steuerungsmodellen entwickelt und diese in einer Model-in-the-Loop-Simulation dazu genutzt, um Steuerungsalgorithmen und Komponenten zunächst virtuell zu entwickeln. In weiteren Phasen der domänenspezifischen Ausarbeitung werden z.B. erste virtuelle Steuerungsentwürfe genutzt, um sie mit realen Komponenten zu verknüpfen und weiter zu entwickeln. Die entwickelte, reale Steuerung wird dann in der Systemintegrationsphase als reale Komponente mit einem virtuellen Modell gekoppelt und kann so effizient diversen Testreihen unterzogen werden. Final wird dann in einem letzten Schritt noch das reale Gesamtsystem getestet, bevor die Entwicklung abgeschlossen ist.

Dieses Vorgehen wird hier am Beispiel eines Waschautomaten verdeutlicht. Hier werden zunächst die beschriebenen Model-in-the-Loop-Simulationen durchgeführt, um die Waschverfahren sowohl physikalisch als auch deren Steuerung auszulegen. Im Anschluss daran werden erste Prototypen aus der virtuellen Welt heraus angesteuert, um den genauen Prozessablauf zu definieren. Ist dieser anschließend real auf einem Steuergerät implementiert, kann dieses Steuergerät gegen ein virtuelles Abbild des Waschautomaten getestet werden. Hierbei können sämtliche Funktionalitäten abgeprüft werden, welche bspw. durch den Benutzer einstellbar sind. Zusätzlich allerdings und von entscheidender Bedeutung ist, dass auch Fehler gezielt in das System eingebracht werden können und somit auch sämtliche Fehlerfälle abprüfbar sind. Es können Sensorausfälle getestet werden, ohne die Hardware variieren oder gar manipulieren zu müssen. Diese Vorgehensweise inkl. der Modellierung der Benötigten Systemanteile ist unter [3][4][5] vorgestellt.

An dieser Stelle wird bereits deutlich, dass an unterschiedlichen Punkten entlang des Entwurfsprozesses

Modelle eingesetzt werden, um den Prozess deutlich effizienter zu gestalten. Diese Modelle weisen allerdings unterschiedliche Eigenschaften und Detaillierungsgrade auf, da sie kontinuierlich mit-/weiterentwickelt werden und unterschiedliche Eigenschaften beinhalten müssen. In frühen Phasen weisen die Modelle meist stärkere Abweichungen zu realen Systemen auf, wohingegen die Hardware-in-the-Loop-Tests mit Modellen durchgeführt werden, welche zum Beispiel die Wärmeentwicklung im Waschautomaten mit einer hohen Genauigkeit abbilden müssen, um die Funktionalität überhaupt aussagekräftig und reproduzierbar testen zu können.

3 Modellbasierte Entwicklung in gelebten Prozessen

Die beschriebenen Einsatzpunkte der modellbasierten Entwicklung erfordern unterschiedliche Ausprägungen von Modellen.

Die Schwierigkeit besteht in realen Entwicklungen oftmals darin, die genauen Detaillierungsgrade der Modelle festzulegen. Grundsätzlich gilt, dass der Aufwand zur Erstellung für diese Modelle mit zunehmender Detailausprägung deutlich steigen kann. Es müssen physikalische Grundlagen erarbeitet werden, welche in Form von bspw. Differenzialgleichungen in den Modellen abgebildet werden und Messreihen untersucht oder auch angefertigt werden, um die Modelle validieren zu können. Durch diesen Mehraufwand steigen dann auch die Kosten zur Erstellung dieser Modelle. Um den Nutzen der Simulationstechnik allerdings effizient gestalten und voll ausschöpfen zu können, sollten die Modelle auf den Einsatzzweck angepasst werden, ähnlich der Beziehung zwischen dem Produkt und den dazugehörigen Anforderungen. Die Modelle sollten die Anforderungen erfüllen, allerdings nach Möglichkeit nicht mehr abbilden als gefordert wird. Der Aufwand, welcher zusätzlich in diese Übererfüllung investiert wird, bringt letztlich keinen Mehrwert, da die Funktion ohnehin bereits erfüllt ist. Wird die Detaillierung des Modells zu niedrig gewählt, tritt das andere Extrem ein, eine Nicht-Erfüllung der Anforderungen an die Simulationsmodelle.

[6] schlägt hier eine Methodik vor, um den Anforderungen gerechte Modelle aufzusetzen und den Aufwand und Nutzen aufeinander abstimmen zu können. Diese Anforderungen werden in der Regel von den Modellierungsexperten definiert und iterativ angepasst/erweitert. In gelebten Entwicklungsprozessen

hat sich so ein Vorgehen etabliert, welches allerdings zukünftig vor neue Herausforderungen gestellt wird.

In Zeiten der Industrie 4.0 tritt vermehrt die Problematik auf, dass Systeme mit zusätzlichen, innovativen Funktionen ausgestattet werden müssen. Diese Funktionen müssen wiederum abgetestet werden, um qualitative Standards gewährleisten zu können, was den zeitlichen Aufwand für die Entwicklung ebenfalls erhöht.



Abbildung 2 Problemstellung im Engineering

Abbildung 2 verdeutlicht schematisch diese Problematik. Die Komplexität von Produkten nimmt im Vergleich zur Leistungsfähigkeit aktueller Werkzeuge stärker zu. Um am Markt erfolgreich agieren zu können, werden weitere Effizienzsteigerungen benötigt, welche die Prozesse der Entwicklung befähigen, die entstehende Lücke zwischen Komplexitätszunahme und Zunahme der Leistungsfähigkeit der Entwicklungsmethoden schließen.

4 Model Based Systems Engineering (MBSE)

Um dieser Problematik zu begegnen, müssen Methoden des modellbasierten Systems Engineering zum Einsatz kommen. Hierbei werden die zugrundeliegenden Prozesse ganzheitlich betrachtet. Bezogen auf die Entwicklung mechatronischer Systeme gilt es, das unter **Abbildung 1** gezeigte Vorgehen näher zu betrachten und Schnittstellen zwischen den Disziplinen näher zu untersuchen.

Beleuchtet werden hierbei die zuvor erläuterten Anforderungen an die Modellbildung der mechatronischen Systeme. Wichtiger Aspekt ist hier, wie und zu welchem Zeitpunkt diese Anforderungen definiert werden.

Getreu dem Ansatz des Systems Engineerings wird der Gedanke verfolgt, die bereits in frühen Phasen der Systemspezifikation definierten und dokumentierten Anforderungen an das Gesamtsystem zu nutzen und hieraus die Anforderungen an die Simulationsmodelle

abzuleiten. Dieses Vorgehen hat den Vorteil, dass bereits bestehendes, erarbeitetes Wissen genutzt wird, um in späteren Phasen der Entwicklung einen Mehraufwand zu vermeiden.

Zunächst ist hierbei zu definieren, welchen Zweck das zu erstellende Modell erfüllen muss. Um hierbei zurück auf das Beispiel zu verweisen, ist zu definieren, ob eine rein virtuelle Untersuchung in der Systementwurfsphase erfolgen soll, ein Einsatz eines Steuerungsmodells im Rahmen eines Rapid-Control-Prototyping-Prüfstandes oder der Steuergerätestest durch die Umsetzung eines Hardware-in-the-Loop-Prüfstandes. Die Anforderungen sind demnach um wichtige Informationen zu erweitern. Am Beispiel des Hardware-in-the-Loop Prüfstandes einer Waschautomatensteuerung gilt es demnach zunächst, ein Modell des Waschautomaten zu entwickeln. Die in der Spezifikationsphase ermittelten Anforderungen sind exemplarisch in **Abbildung 3** gezeigt. Wichtig im Bereich der weißen Ware ist primär das erfolgreiche Prozessergebnis, die saubere Wäsche.

Req.ID	Requirements	Priority
1	Wäsche Reinigen	A
3	Nutzerschnittstelle bereitstellen	A
4	Energy Label A+++	A
5	Wassersicherheitssystem	A
6	Befüllen ermöglichen	A
7	Futuristisches Design	C

Abbildung 3 Exemplarische Anforderungen

Zur Umsetzung dieser Anforderung ist in Waschautomaten u.a. eine Heizung vorgesehen, um reaktive Substanzen im Waschmittel zu aktivieren und darüberhinaus ggf. Bakterien thermisch zu neutralisieren. Diese Heizung muss, analog zum Zielsystem, ebenfalls im Simulationsmodell enthalten sein. Um zu spezifizieren, in welcher Detailgenauigkeit die Heizung bzw. das thermische Verhalten abgebildet werden muss, ist die Funktionshierarchie mit den definierten Funktionen zu betrachten. **Abbildung 4** zeigt eine exemplarische Auflistung einiger Funktionen und Subfunktionen, die zur Erfüllung der Anforderungen nötig sind. Hier ist zu überprüfen, welche der Funktionen Einfluss auf das thermische System des Waschautomaten haben. Neben vielen weiteren sensiert die Steuerung die Temperatur über eine implementierte Sensorik. Aus diesem Grund sind die thermischen Verhältnisse detaillierter abzubilden, wenn die Steuerung im Rahmen des HiL-Prüfstandes geprüft werden soll.

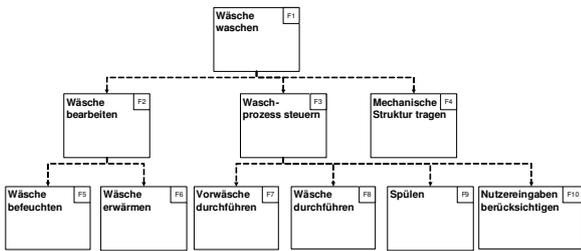


Abbildung 4 Funktionshierarchie Waschautomat

Das thermische Verhalten der beteiligten Komponenten beinhaltet Wärmeübertragungen unterschiedlicher Art, Massen und geometrische Daten der Komponenten. Diese müssen erfasst und umgesetzt werden, um die Entwicklung der Wärmeverteilung abbilden zu können. Es sind weiterhin die benötigten Größen (bspw. Temperatur) als Schnittstelle zu der zu prüfenden Steuerung vorzusehen, um die Logik der Steuerung auch valide prüfen zu können. Die Funktionen der Funktionshierarchie sind jeweils zu nummerieren, um eine Rückverfolgbarkeit und Durchgängigkeit zu erreichen. In gleicher Weise sind in den jeweiligen Modellen/Submodellen/Modellkomponenten die entsprechenden Funktionsnummern zu referenzieren, welche zur Erfüllung dieser Funktion beitragen.

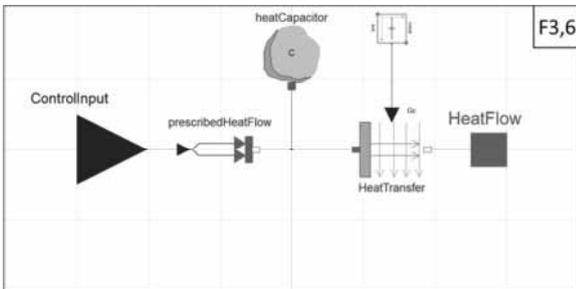


Abbildung 5 Submodell der Heizung

Abbildung 5 zeigt ein exemplarisches Submodell, aufgebaut unter Modelica-Dymola. Hier ist die Heizung mit den zugrundeliegenden Wärmeübertragungsmechanismen gezeigt. Erkennbar sind die Schnittstelle zur Ansteuerung der Heizung und die Referenzierung der durch die Heizung beeinflussten Funktionen der Funktionshierarchie. Hier ist der direkte Bezug zwischen dem Simulationsmodell und der Funktion/Anforderung hergestellt. Ein Ergebnis der Simulation zeigt **Abbildung 6**. Hier ist die simulierte Temperatur über die Zeit aufgetragen. Wichtig hierbei ist, dass gewisse Temperaturspitzen erreicht werden, um ein positives Waschergebn zu erhalten. In diesem Fall betrifft dies die maximale Temperatur

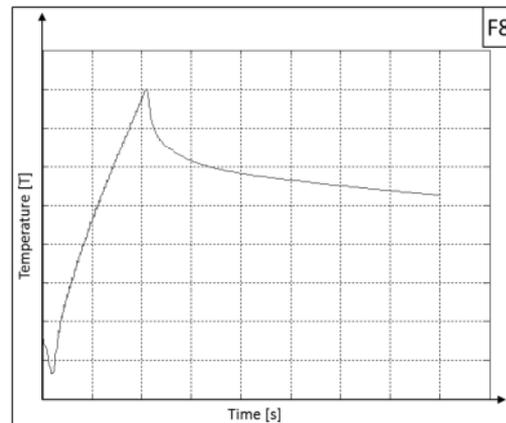


Abbildung 6 Simulationsergebnis

des dargestellten Verlaufes. So kann durch eine Verkopplung des Modells mit der Steuerung die Logik abgeprüft werden.

Neben der Abbildung der physikalischen Grundlagen und der Definition der nötigen Schnittstellen ist ein wichtiges Kriterium der Modellbildung des Waschautomaten die Echtzeitfähigkeit des Modells, da sichergestellt werden muss, dass die Steuerung zu festen Zeitschritten die entsprechenden Eingangssignale erhält. Andernfalls tritt das Steuergerät in einen Fehlermodus. Um diese Echtzeitfähigkeit herstellen zu können, sind wiederum die entsprechenden Anforderungen hinzuzufügen und zu verlinken. Im Rahmen einer rein virtuellen Nutzung von Modellen ist wiederum die Echtzeitfähigkeit nicht von Interesse, da dort der Fokus unter Umständen auf eine kurze Simulationszeit gelegt wird um schnell, viele Möglichkeiten und Umsetzungen untersuchen zu können.

Durch das zuvor beschriebene Vorgehen ist sichergestellt, an jeder Stelle der modellbasierten Entwicklung die zugrundeliegenden Anforderungen abprüfen

zu können. Darüberhinaus ist so die Basis geschaffen, die Systemanforderungen ebenfalls zur Modellbildung heranzuziehen, um Redundanzen zu vermeiden und somit iterative Prozesse zu minimieren.

5 Zusammenfassung

Dieser Beitrag thematisiert die Modellbildung im Kontext des Systems Engineering. Es wurde zunächst auf die modellbasierte Entwicklung von mechatronischen Systemen nach [2] eingegangen, die bereits eine Steigerung der Effizienz bewirkt. Vor dem Hintergrund der Industrie 4.0 und den resultierenden Konsequenzen für die intelligenten technischen Systeme entstehen zusätzliche Funktionalitäten, welche in die Systeme integriert werden. Dies steigert die Komplexität deutlich. Um weiterhin und im gleichen Maße die Effizienz der Entwicklungswerkzeuge und -methoden zu steigern wurde die Verknüpfung der definierten Anforderungen mit den im modellbasierten Kontext erstellten Simulationsmodellen gezeigt. Dies hat den Vorteil, dass bereits bestehende Arbeitsinhalte dazu genutzt werden, die Simulationsmodelle genauer zu spezifizieren, um einen Mehr- oder Minderaufwand zu vermeiden, da dieser in beiden Fällen zu erhöhten Entwicklungskosten/-zeiten führt. Die vorgestellte Methode ist somit ein wichtiger Beitrag zu dem Thema Modellbasiertes Systems Engineering und dient der effizienten Entwicklung im Kontext der Industrie 4.0.

6 References (Überschrift 1)

- [1] Auf dem Weg zu Industrie 4.0. Strategie des it's OWL Spitzenclusters, Herausgeber: it's OWL Clustermanagement GmbH, April 2014.
- [2] Verein Deutscher Ingenieure VDI: Entwicklungsmethodik für mechatronische Systeme, VDI 2206, Düsseldorf, 2004.
- [3] Kruse, Daniel; Trächtler, Ansgar: Modellbasierte Entwicklung eines neuartigen Heizverfahrens für Waschautomaten. Paderborner Workshop Entwurf Mechatronischer Systeme (EMS), Apr. 2013.
- [4] Kruse, Daniel; Fast, Viktor; Schweers, Christoph; Trächtler, Ansgar: Einheitliche Testumgebung für MiL und RCP mittels NI VeriStand am Beispiel eines Waschautomaten. Virtuelle Instrumente in der Praxis (VIP 2012), Okt. 2012.
- [5] Löffler, Alexander; Schweers, Christoph; Fast, Viktor; Kruse, Daniel; Trächtler, Ansgar: Multidomänen-Modell eines Waschvollautomaten für einen Hardware-in-the-Loop-Prüfstand. 8. Paderborner Workshop: "Entwurf mechatronischer Systeme", Mai 2011.
- [6] Schmuедderich, Tanja; Lochbichler, Matthias; Trächtler, Ansgar: Methodik zur anforderungsgerechten Wahl der Modellierungstiefe von Verhaltensmodellen fuer die virtuelle Inbetriebnahme. In: Tagungsband Mechatronik 2013 (ISBN 3-86130-958-0), S. 43-48, 2013.

Effiziente und durchgängige Entwicklung von Fahrfunktionen in der Lehre durch Simulation

Mirek Göbel, Hochschule Hamm-Lippstadt
mirek.goebel@hshl.de

Durch schnelle Entwicklungszyklen und eine ausgeprägte Arbeitsteilung bei der Entwicklung von Steuergeräten besteht ein hohes Anforderungsprofil an den Ingenieur. Dieser muss zugleich Informatiker und Funktionsspezialist sein, um zielgerichtet und effizient Funktionen entwickeln zu können. Dieser Vortrag zeigt auf, wie Ingenieure mit Hilfe einer modernen Werkzeugkette am Beispiel von automobilen Fahrfunktionen eines autonom fahrenden (Modell-) Fahrzeugs schnell und effizient Funktionen entwickeln können, ohne auf beiden genannten Gebieten gleichzeitig Spezialist zu sein. Dazu werden Funktionen bereits vor dem Test im Prototypen in der Simulation entworfen, analysiert, vorparametriert und getestet. Dadurch zeigt sich analog zum Vorgehen in der Industrie, wie mit wenigen Prototypen trotzdem viele Ingenieure simultan an Fahrfunktionen arbeiten können!

1 Einleitung

Dieser Artikel entstand im Rahmen der Lehrveranstaltung „Praktikum Systems Design Engineering“ an der Hochschule Hamm-Lippstadt. In diesem Praktikum haben Herr Prof. Dr.-Ing. Ulrich Schneider und der Autor dieses Artikels ein Modellfahrzeug entwickelt und realisiert, das autonom in einer Fahrspur fahren kann und außerdem in der Lage ist, weitere Zusatzaufgaben wie Überholen, Einparken, Anhalten an einer Stopplinie etc. zu absolvieren.

Anhand dieses Projektes lernen die Studierenden, wie Systeme entwickelt werden und erlangen praktische Kenntnisse vor allem in den Bereichen Informatik, Bildverarbeitung, Messtechnik und Regelungstechnik.

1.1 Grundkonzept Fahrzeug

Das Modellfahrzeug ist in Abbildung 1 dargestellt und besteht im Wesentlichen aus den folgenden Komponenten.

- Modellbau-Chassis mit Antrieb und Lenkung
- Kamera auf einer Halterung zur Erfassung von

vor dem Fahrzeug fahrenden Objekten sowie der Fahrspurlinien

- Ultraschallsensoren als Einparkhilfs-Sensoren
- PC in der Größe „ITX“ mit dSPACE-Rapid-Control-Prototyping-Karte des Typs „DS1104“ als Echtzeit-Rechner für die Fahrfunktionen
- Beschleunigungs- und Gierratensensor zur Erfassung der dynamischen Größen
- Laserscanner vorne zur Erfassung von Objekten

1.2 Werkzeugkette

Zum Einsatz kommt eine verbreitete Rapid-Control-Prototyping-Werkzeugkette. Diese ermöglicht es, dass Entwickler ohne Expertenkenntnisse in Informatik auf einfache Art und Weise Funktionen – hier Fahrfunktionen, die das Fahrgeschehen des Fahrzeugs vorgeben und beeinflussen – erstellen können. Die Folgende Aufzählung zeigt schrittweise, welche Werkzeugkette zum Einsatz kommt.

1. Fahrfunktionen erstellt in Matlab/Simulink (Mathworks)

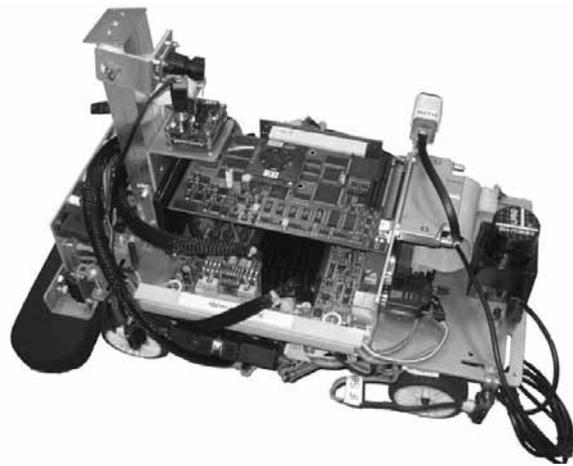


Abbildung 1: Foto des Modellfahrzeugs, zentral zu sehen ist der PC

2. Übersetzung des „online“-Modells (dieses wird unten genau erläutert) in Software für die DS1104-Karte von dSPACE
3. Übertragung der Software auf den PC des Fahrzeugs
4. Einrichten und Starten der Steuersoftware „ControlDesk (dSPACE)“ auf dem Fahrzeug-PC
5. Laden der erstellten Software auf die dSPACE-Karte DS1104
6. Fahrzeug erproben unter Zuhilfenahme der (durch „Drag and Drop“) selbst sehr einfach erstellbaren grafischen Benutzeroberfläche

1.3 Fahrfunktionen

Die Fahrfunktionen, die das Fahrzeug ansteuern und regeln, werden in Blöcke aufgeteilt, um diese in einer klaren Struktur abgegrenzt voneinander zu entwickeln. Einen Überblick über die Fahrfunktionen gibt Abb. 2.

Im Block „Sensordatenaufbereitung (SAB)“ werden die Sensordaten gefiltert sowie teilweise Offsetbereinigt und es werden zusätzliche Signale aus den Sensordaten berechnet. Zum Beispiel wird nach einer Offsetbereinigung aus der Gierrate der Gierwinkel durch eine offene Integration bestimmt.

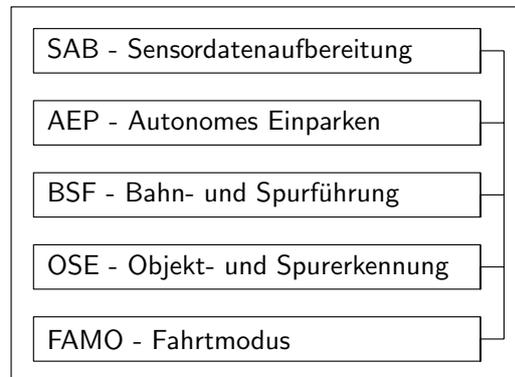


Abbildung 2: Fahrfunktionen

Der Block „Autonomes Einparken (AEP)“ enthält einen Zustandsautomaten, der das Einparken in mehreren Teilschritten übernimmt.

Die „Bahn- und Spurführung (BSF)“ ist für die Längs- und Querregelung des autonom fahrenden Fahrzeugs zuständig. D. h. es bestimmt den notwendigen Lenkwinkel zur Einhaltung der Sollbahn sowie das erforderliche Antriebsdrehmoment, um das Fahrzeug mit der Sollgeschwindigkeit fahren zu lassen, die ebenfalls in diesem Block bestimmt wird.

Die Bestimmung der Sollbahn sowie das Erfassen und Verfolgen der Spur und Objekten übernimmt der Block „Objekt- und Spurerkennung (OSE)“.

Im Block „Fahrtmodus (FAMO)“ werden die verschiedenen Fahrmodi umgeschaltet, die das Fahrzeug erfüllen soll. Diese sind z. B. autonomes Einparken, Fahren ohne Hindernisse, Fahren mit Hindernissen, etc..

1.4 Problemstellung

Die oben genannten Fahrfunktionen werden in einer Gruppe von Studierenden und Dozenten entwickelt. Da bei diesem Projekt nur ein Prototyp zur Verfügung steht, muss eine Lösung entwickelt werden, die ein gleichzeitiges Arbeiten an den Fahrfunktionen ermöglicht, auch wenn kein Zugriff auf den Prototypen möglich ist.

1.5 Lösungsansatz

Das genannte Problem wird mit Hilfe von zwei Ansätzen gelöst:

1. Strukturierung der Fahrfunktionen und Erzeugung von Bibliotheken. Damit ist jeder Entwickler in der Lage, seinen Block getrennt von den anderen Entwicklern zu bearbeiten.
2. Erprobung der Fahrfunktionen in der Simulation! Fahrfunktionen können somit ohne Zugriff auf den Prototypen getestet und vorparametriert werden.

2 Funktionsrahmen für den „online“-Einsatz und die MiL-Simulation

Dieser Abschnitt beschreibt die beiden genannten Lösungswege. Nach der Vorstellung der erzeugten Struktur zum simultanen Arbeiten an den Fahrfunktionen wird der Aufbau der Simulation erläutert, der im Folgenden mit „Funktionsrahmen“ bezeichnet wird. Damit wird es möglich, dass das Fahrfunktionsmodul in derselben Form „offline“ in einer „Model in the Loop (MiL)“-Simulation und auch „online“ auf dem dSPACE-Rapid-Prototyping-Rechner im Fahrzeug in Echtzeit läuft.

2.1 Struktur der Fahrfunktionen mit Bibliotheken

Um ein simultanes Arbeiten an den Blöcken der Fahrfunktionen zu ermöglichen, wird jeder Block getrennt von den anderen in einer eigenen Bibliothek entwickelt. Um mit allen Blöcken im Verbund zu simulieren oder zu fahren, ist darüber hinaus ein Funktionsrahmen notwendig, der alle Fahrfunktionen mit der notwendigen Umgebung für die Betriebsarten „online“ und „offline“ verknüpft, die im nächsten Abschnitt näher erläutert werden.

Der Funktionsrahmen ist in der Software Matlab/Simulink der Firma Mathworks erstellt. Wird mit dem Funktionsrahmen gearbeitet, so lädt dieser beim

Starten alle Blöcke aus den Bibliotheken und kann dadurch mit dem aktuellen Stand der Entwicklung jedes Blocks ausgeführt werden.

Ratsam ist dabei die Verwendung eines Versionskontrollsystems, um die Zusammenarbeit von vielen Entwicklern an unterschiedlichen Rechnern und ggf. zu unterschiedlichen Zeiten zusammenführen zu können. Im vorliegenden Projekt kam das Versionierungswerkzeug „Subversion (SVN)“ [1] zum Einsatz.

2.2 Verwendung desselben Fahrfunktionsmoduls in MiL-Simulation und in Echtzeit auf dSPACE-Rapid-Prototyping Rechner

Zur effizienten Entwicklung der Fahrfunktionen mit einer Anzahl an Entwicklern, die die Anzahl der zur Verfügung stehenden Prototypen bei weitem übersteigt, wird auf die Simulation zurückgegriffen. Bei der Erstellung/Erprobung/Parametrierung der Fahrfunktionen soll außerdem dasselbe Modell für die Simulation und für das Fahrzeug verwendet werden, d. h. es wird eine durchgängige Entwicklung der Fahrfunktionen gefordert.

Die Lösung besteht aus zwei verschiedenen Funktionsrahmen mit dem gleichen Fahrfunktionsmodul. Für die Entwicklung des Fahrfunktionsmoduls ist es somit unerheblich, ob dieses in der MiL-Simulation oder auf dem Fahrzeug in Echtzeit läuft. Zu Beginn muss der Anwender lediglich auswählen, mit welchem Funktionsrahmen gearbeitet werden soll.

- „offline“ zur Durchführung der MiL-Simulation, siehe Abb. 3
- „online“ zur Kompilierung und Transfer auf den dSPACE-Rapid-Prototyping-Rechner im Fahrzeug, siehe Abb. 4

Abbildung 3 zeigt den Funktionsrahmen als Skizze für die MiL-Simulation. Sensoren, Aktoren und Fahrzeug sind hier modelliert. Das beinhaltet viele unterschiedliche Modelle, die in Beispielen in der nachfolgenden Aufzählung kurz stichpunktartig erläutert werden.

- Kamera: Ermittlung einer Sollspur in Form einer Parabel aus einer vorgegebenen Bahn

- Fahrzeug: Einspurmodell nach [2]. Einfaches Fahrzeugmodell mit wenigen Parametern für die Darstellung der Fahrdynamik bis zu einer Querbeschleunigung von ca. 4 m/m^2
- Lenkwinkel: Umsetzung des Lenkwinkels aus einer Pulsbreite
- Ultraschallsensoren: Bestimmung eines Abstandes von Testobjekten (aus einer Objektliste) in Blickrichtung des Sensors

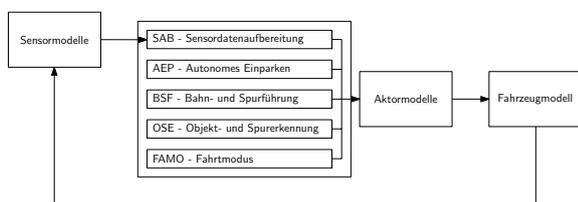


Abbildung 3: Funktionsrahmen „offline“

Um mit dem Funktionsrahmen arbeiten zu können, sind außerdem eine Manöversteuerung sowie eine Datennachbearbeitung umgesetzt, um die Ergebnisse der Simulation geeignet auswerten zu können.

Der Funktionsrahmen zum Einsatz auf dem Echtzeitrechner im Fahrzeug ist in Abbildung 4 dargestellt. Im Gegensatz zur Simulation werden in dieser Form keine Modelle, sondern die Schnittstellen zu den Sensoren und Aktoren im Fahrzeug benötigt und das Fahrzeugmodell entfällt bzw. liegt in realer Form vor. Aus diesem Grund ist das Fahrzeug und der Rückkopplungspfad in Abbildung 4 gestrichelt dargestellt.

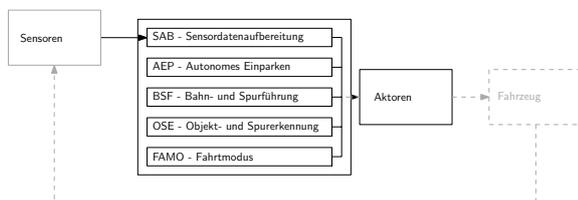


Abbildung 4: Funktionsrahmen „online“

Der Entwickler kann nach dem Laden dieses Funktionsrahmens direkt den Kompilervorgang einleiten und anschließend die so erstellte Software auf den dSPACE-Echtzeitrechner im Fahrzeug laden. Damit ist das Fahrzeug einsatzbereit und die Fahrfunktionen können erprobt werden!

3 Fazit

Durch die Zuhilfenahme der MiL-Simulation und die oben gezeigte Struktur mit zwei umschaltbaren Funktionsrahmen wird eine effiziente und durchgängige Entwicklung von Fahrfunktionen ermöglicht. In Kombination mit der gezeigten Struktur und der Verwaltung der Fahrfunktionen in Bibliotheken kann eine große Anzahl von Entwicklern an den Funktionen arbeiten, ohne sich gegenseitig zu behindern oder durch eine Belegung des Prototypen durch einen anderen Entwickler warten zu müssen!

Es hat sich in der Praxis gezeigt, dass ein Großteil der Arbeit bei der Entwicklung von Fahrfunktionen in der MiL-Simulation erfolgen kann. Dadurch wird die Gesamtzeit, in der mit dem Prototyp gearbeitet wird, deutlich reduziert. Übertragen auf das Arbeiten in der Industrie verdeutlicht auch das vorliegende Hochschulprojekt einmal mehr, dass durch den gezielten und maßgeschneiderten Einsatz von Simulation nicht nur die Entwicklungszeit verkürzt, sondern auch die Anzahl an notwendigen Prototypen reduziert werden kann!

Literatur

- [1] B. Collins-Sussman, B. W. Fitzpatrick, C. M. Pilato. *Versionskontrolle mit Subversion*. <http://svnbook.red-bean.com/nightly/de/svn-book.pdf> vom 04.02.2016.
- [2] P. Riekert, T. E. Schunck, *Zur Fahrmechanik des gummibereiften Fahrzeugs*, Ingenieur Archiv Bd. 11, Berlin/Heidelberg: Springer, S. 210-224, 1940,

Entwicklung eines Simulators für autonome Roboter zur Ausbildung von Studierenden in der Mechatronik und Informatik

Prof. Dr. Axel Thümmler
Hochschule Hamm-Lippstadt
Lippstadt Department 1
Marker Allee 76-78
D-59063 Hamm, Germany
Axel.Thuemmler@hshl.de

Durch die in den vergangenen Jahren zunehmende Leistungsfähigkeit elektronischer Systeme wird die Entwicklung effizient agierender autonomer Systeme stark begünstigt und dadurch neue Einsatzgebiete wie zum Beispiel autonome Fahrzeuge für die Zukunft erschlossen. Insbesondere die Entwicklung von Computeralgorithmen gewinnt dabei an Bedeutung, da heutzutage die meisten Funktionalitäten in Software realisiert werden. Es ist daher wichtig, dass sich Studierende in Fächern wie Mechatronik und Informatik mit den Aspekten autonomer Systeme beschäftigen.

In diesem Artikel wird ein Simulator für einen autonomen Roboter, welcher sich in einer unbekanntem Landschaft zurechtfinden muss, vorgestellt. Über eine C++-Programmierschnittstelle wird konfiguriert, mit welchen Sensoren der Roboter arbeiten soll. Während der Fahrt durch die unbekanntem Landschaft werden die Werte der Sensoren abgefragt und daraufhin Aktuatoren für die Vorwärts-/Rückwärtsfahrt und Drehbewegung angesteuert. Mit dem Simulator können unterschiedliche Szenarien, wie z.B. die vollständige Erkundung einer unbekanntem Landschaft, die Ansteuerung vorgegebener Positionen oder das Finden von Auswegen betrachtet werden.

Der Simulator wurde im Lehrbetrieb an der Hochschule Hamm-Lippstadt mit Erfolg eingesetzt. Die grafische Simulationsumgebung wirkt stark motivierend, sodass die von den Studierenden entwickelten Algorithmen die ursprünglichen Erwartungen zum Teil übertrafen.

1 Einleitung

Der Themenbereich autonom agierender Maschinen hat in den letzten Jahren stark an Bedeutung zugenommen. Dies liegt einerseits an den vielfältigen Anwendungsmöglichkeiten, wie z.B. autonome Fahrzeuge, die sich ohne Steuerung durch einen Fahrer auf Autobahnen oder auch im Stadtverkehr bewegen können [6], oder Roboter, die sich in schwer zugänglichen und unbekanntem Landschaften zurechtfinden müssen [4]. Andererseits hat die technische Entwicklung der vergangenen Jahre im Bereich elektronischer Systeme und Software dazu beigetragen, dass Lösungen für die beschriebenen Anwendungsszenarien immer mehr die gestellten Anforderungen und Erwartungen erfüllen. Insbesondere die Entwicklung von Computeralgorithmen gewinnt dabei an Bedeutung, da heutzutage die meisten Funktionalitäten in Software realisiert werden. Es ist daher wichtig, dass sich Studierende in Fächern wie Mechatronik und Informatik mit den Aspekten autonomer Systeme beschäftigen.

In diesem Artikel wird ein Simulator für einen autonomen mobilen Roboter mit Namen „Robi Tobi“ vorgestellt. Mit dem Simulator kann Robi Tobi bei einer Fahrt durch eine unbekanntem Landschaft beobachtet und das Verhalten des Roboters bewertet werden. Der Simulator stellt verschiedene Landschaftsformen mit unterschiedlichen Hindernissen bereit, mit denen Robi Tobi bei seiner Fahrt durch die Landschaft nicht kollidieren darf. Der Simulator wird über eine C++-Programmierschnittstelle bedient. Über die Schnittstelle wird sowohl die Landschaftsform als auch die Ausstattung von Robi Tobi mit Sensoren konfiguriert. Ebenfalls kann Robi Tobi über die Programmierschnittstelle vorwärts und rückwärts bewegt und gedreht werden. Mittels seiner Sensoren kann Robi Tobi den Abstand zu Hindernissen bestimmen. Während der Fahrt durch die unbekanntem Landschaft kann Robi Tobi die Werte seiner Sensoren abfragen und daraufhin seine Aktuatoren für die Vorwärts-/Rückwärtsfahrt und Drehbewegung ansteuern. Robi Tobi gewinnt dabei Informationen zum Aussehen der Landschaft, die er in einer Rasterkarte in

seinem Speicher ablegen kann. Aufgrund der gewonnenen Informationen kann Robi Tobi seine Fahrt dahingehend optimieren, dass er bereits untersuchte Landschaftsbereiche nicht erneut untersucht.

Der Simulator wurde im Lehrbetrieb der Hochschule Hamm-Lippstadt im Sommersemester 2015 und, in einer Vorversion, im Sommersemester 2014 eingesetzt. Mit dem Simulator können die Studierenden eigene Algorithmen bzgl. einer Aufgabenstellung für Robi Tobi entwerfen und bewerten. Typische Aufgabenstellungen sind z.B. die vollständige Erkundung einer unbekanntenen Landschaft, die Ansteuerung vorgegebener Positionen oder das Finden von Auswegen in einem Labyrinth.

Der Artikel ist wie folgt aufgebaut. In Kapitel 2 wird die Schnittstelle des Simulators aus Sicht des Nutzers des Simulators beschrieben. Kapitel 3 gibt einen Einblick in den internen Aufbau des Simulators und dessen Softwarearchitektur. In Kapitel 4 werden Anwendungsszenarien vorgestellt, für die mit dem Simulator unterschiedliche Lösungsalgorithmen ausprobiert und bewertet werden können. Kapitel 5 schließt mit einer Zusammenfassung und einem Ausblick auf anstehende Arbeiten.

2 Die Schnittstelle des Simulators

2.1 Auswahl der Landschaftsform

Der Simulator steht in Form einer C++-Library mit zugehöriger dokumentierter Header-Datei zur Verfügung. Die Library lässt sich einfach in ein eigenes C++-Projekt integrieren, um dann daraus den Simulator über seine Schnittstelle zu bedienen. Bevor eine Simulation gestartet wird, muss das zugehörige Simulationsszenario konfiguriert werden. Dazu gehört als erstes die Gestaltung einer Landschaft, in der Robi Tobi ausgesetzt wird. Hier stehen dem Nutzer unterschiedliche Landschaftsformen, die zum Teil auf eine zufällige Art und Weise generiert werden können, zur Verfügung. Abbildung 1 zeigt beispielsweise eine Landschaft die aus 60 rechteckigen Objekten zufälliger Größe und Position besteht. Abbildung 2 zeigt dagegen eine Landschaft mit einem Labyrinth. Nicht befahrbare Objekte sind als ausgefüllte, dunkelrote Flächen dargestellt. Robi Tobi ist jeweils als kleines, hellgraues Rechteck mit einer Größe von 20×20 Längeneinheiten (Pixel) in den beiden Landschaften zu erkennen.

Neben den dargestellten Landschaftsformen gibt es weitere Landschaften, die aus mehreren, zufällig

positionierten Ellipsen oder Sternen bestehen, sowie eine Landschaft, die aus einem großen HSHL-Logo besteht. Der Softwarearchitektur des Simulators erlaubt es aber, leicht neue Landschaftsformen hinzuzufügen.

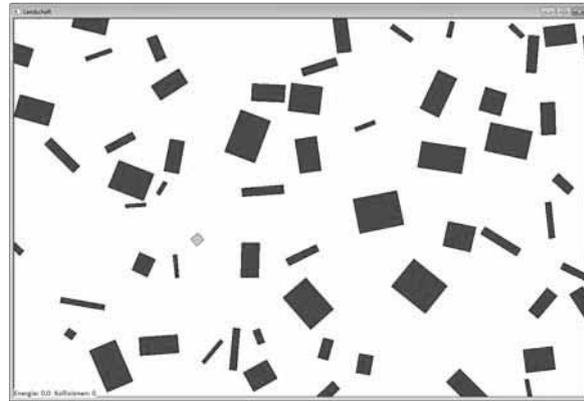


Abbildung 1. Landschaft mit 60 Rechtecken zufälliger Größe und Position.

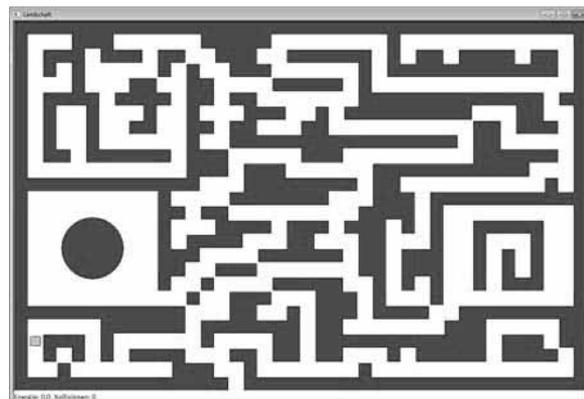


Abbildung 2. Landschaft mit einem Labyrinth.

2.2 Ausstattung von Robi Tobi mit Sensoren

Neben der Auswahl der zu verwendenden Landschaft muss vor dem Start einer Simulation festgelegt werden, mit welchen Sensoren Robi Tobi ausgestattet werden soll und wo diese Sensoren an seiner Oberfläche positioniert werden sollen. In der aktuellen Version des Simulators steht nur ein Sensortyp zur Verfügung und zwar ein abstrahierter Ultraschallsensor, der den Abstand eines Objektes in Blickrichtung bestimmt. Sowohl die Anzahl der Ultraschallsensoren, mit denen Robi Tobi ausgestattet werden soll, als auch die Ausrichtung der Sensoren sind frei konfigurierbar. Um die Pose (d.h. Position und Ausrichtung) eines Sensors festzulegen, sind zwei Punkte P_1 und P_2 zu definieren. Der Punkt P_1 ist ein Punkt auf dem Rand von Robi Tobi, an dem der Ultraschallsensor festgemacht werden soll. Der Punkt P_2 muss ein

Punkt im Inneren von Robi Tobi sein. Durch den Richtungsvektor $\vec{v} = P_1 - P_2$ wird die Ausrichtung des Ultraschallsensors bestimmt. Abbildung 3 zeigt Robi Tobi als ein Quadrat mit einer Kantenlänge von 20 Längeneinheiten sowie eine mögliche Positionierung und Ausrichtung eines Sensors.

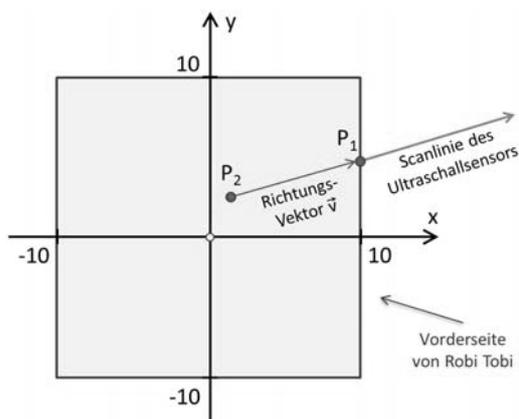


Abbildung 3. Positionierung eines Sensors durch Angabe der Punkte P_1 und P_2 .

2.3 Bedienung des Simulators über die C++-Schnittstelle

Die Konfiguration des Simulators erfolgt über eine C++-Schnittstelle. Die Parameter der Konfiguration werden dabei in dem Datentyp `SimConfigT` zusammengefasst (siehe Code 1). In Zeilen 3 und 4 wird die Landschaftsform definiert. Der Landschaftsparameter ist dabei abhängig von der Landschaftsform und bestimmt z.B. die Anzahl der Rechtecke in der Landschaftsform mit Rechtecken zufälliger Position und Ausrichtung. In Zeile 5 wird die Pose, die Robi Tobi zu Beginn der Simulation einnehmen soll, definiert. Dies kann eine zufällig gewählte Pose oder eine vom Nutzer vorgegebene Pose sein. Im Falle einer nutzerdefinierten Pose wird diese in Form von Position (x- und y-Koordinate) und Winkel (im Gegenuhrzeigersinn von der positiven x-Achse abgetragen) in Zeilen 6 und 7 angegeben. Die Posen von bis zu drei Ultraschallsensoren werden in einem Array in Zeile 8 wie in Kapitel 2.2 beschrieben definiert. Das Hauptprogramm zur Steuerung von Robi Tobi wird dem Simulator in Form eines Funktionszeigers auf eine parameterlose Funktion in Zeile 9 übergeben.

Um die Simulation visuell besser nachverfolgen zu können, ist es möglich, eine Verzögerung in Millisekunden für das Fahren und Drehen von Robi Tobi zu definieren. Die Parameter dafür werden in den Zeilen 10 und 11 festgelegt. Durch die Einstellung in Zeile 12 kann der Simulator angewiesen werden, eine

durchgeführte Messung mit einem Ultraschallsensor in Form einer hellblauen Linie entlang der Messrichtung auf dem Bildschirm auszugeben. Dadurch können die durchgeführten Messungen mit ihrer Richtung visualisiert werden. In Zeile 13 wird schlussendlich die Saat des verwendeten Zufallszahlengenerators festgelegt. Eine wiederholte Simulation mit der gleichen Saat liefert gleiche Zufallszahlen und somit reproduzierbare Ergebnisse. Um in einem Simulationslauf tatsächlich „neue“ Zufallszahlen zu verwenden, ist es eine gebräuchliche Praxis, die Saat des Zufallszahlengenerators mit der aktuellen Uhrzeit zu initialisieren.

```

1  typedef struct
2  {
3      LandschaftsformT landschaft;
4      int landschaftPar;
5      RobiStartPoseT robistartPose;
6      PunktT robistartPosition;
7      double robistartWinkel;
8      SensorPoseT sensorPose[3];
9      RobiMainFuncPtrT robimain;
10     unsigned int fahrenDelay;
11     unsigned int drehenDelay;
12     bool visualizeScanline;
13     unsigned int seed;
14 } SimConfigT;

```

Code 1. Datentyp für die Simulator Konfiguration.

Nach Abschluss der Konfiguration des Simulators kann die Simulation durch Aufruf der Funktion `DoSimulation(...)` gestartet werden. Als Parameter wird der Funktion die Simulationskonfiguration übergeben. Die Funktion startet die Simulation und erweckt Robi Tobi zum Leben, indem die in der Konfiguration definierte Funktion zur Steuerung von Robi Tobi aufgerufen wird (siehe Zeile 9, Code 1). Diese Steuerungsfunktion ist von dem Nutzer des Simulators gemäß der zu lösenden Aufgabenstellung selbst zu entwickeln. Aus der Steuerungsfunktion heraus kann auf die Schnittstelle zu Robi Tobi's Hardware zugegriffen werden. So können z.B. die Messungen der Ultraschallsensoren abgefragt oder Robi Tobi bewegt werden. Die Schnittstelle zu Robi Tobi's Hardware ist in Code 2 dargestellt.

Die Funktionen zur Bewegung und zum Drehen von Robi Tobi (siehe Zeilen 1 bis 7, Code 2) bewegen Robi Tobi jeweils eine Längeneinheit vor oder zurück bzw. drehen ihn um 1° nach links oder nach rechts. Eine Drehung kann Robi Tobi dabei im Stand ausführen, indem er sich um seinen eigenen Mittelpunkt dreht. Robi Tobi sollte bei einer Fahrt oder Drehung mit den Objekten in der Landschaft nicht kollidieren.

```
1 // Robi Tobi um eine Längeneinheiten nach vorne / zurück bewegen.
2 extern int RobiVorwaertsFahren();
3 extern int RobiRueckwaertsFahren();
4
5 // Robi Tobi um 1 Grad nach links / rechts drehen.
6 extern int RobiLinksDrehen();
7 extern int RobiRechtsDrehen();
8
9 // Abstand zu Objekt in Richtung des Ultraschallsensors n messen.
10 extern double GetUltraschallSensor(unsigned int n);
11
12 // Wert an Position (x, y) in Robi Tobi's Karte eintragen / auslesen.
13 extern void SchreibeRobiMap(int x, int y, unsigned char wert);
14 extern unsigned char LeseRobiMap(int x, int y);
15
16 // Robi Tobi's eigene Position in Karte eintragen bzw. auslesen.
17 extern void SetRobiPosInMap(int x, int y, int winkel);
18 extern void GetRobiPosInMap(int& x, int& y, int& winkel);
```

Code 2. Schnittstelle zur Steuerung von Robi Tobi.

Ein paar Kollisionen verkraftet er aber. Nach 25 Kollisionen ist Robi Tobi allerdings fahruntfähig. Die Funktionen in Zeilen 1 bis 7 liefern daher als Rückgabewert, ob Robi Tobi mit einem Objekt kollidiert ist, er fahruntfähig ist oder die Aktion wie gewünscht ausgeführt werden konnte. Mit der Funktion in Zeile 10 wird der Abstand zu einem Objekt in Richtung des Ultraschallsensors gemessen. Falls kein Objekt detektiert wird, wird die maximale Reichweite des Sensors zurückgegeben, welche fest auf 200 Längeneinheiten eingestellt ist.

2.4 Kartierung der Landschaft

Je nachdem, welche Aufgabe Robi Tobi zu lösen hat, kommt es vor, dass er sich eine Karte der durchfahrenen Landschaft in seinem internen Speicher anlegen muss. Eine Karte der Landschaft kann z.B. in Form einer Rasterung der Landschaft in vorgegebener Granularität erstellt werden. Durch die Rasterung wird die Landschaft in gleich große Rasterfelder zerlegt [4]. Ein Rasterfeld wird als *frei* gekennzeichnet, wenn der zugehörige Bereich in der Landschaft nicht von Objekten überdeckt wird. Befindet sich ein Objekt, sei es auch nur teilweise, in dem Bereich der Landschaft, wird das zugehörige Rasterfeld als *belegt* gekennzeichnet. Je nachdem, wie grob das Raster gewählt wird, desto gröber ist auch die Vorstellung, die Robi Tobi von der Landschaft bekommt.

In der aktuellen Version des Simulators bietet die Schnittstelle von Robi Tobi die Möglichkeit, eine Rasterkarte in Robi Tobi's Speicher zu beschreiben und den Zustand eines Rasterfeldes aus der Karte wieder auszulesen. Ein Rasterfeld bildet dabei einen Bereich in der Landschaft mit einer Größe von $30 \times$

30 Längeneinheiten ab. Da Robi Tobi 20×20 Längeneinheiten groß ist, kann er sich einmal im Kreis drehen, wenn er in der Mitte eines freien Rasterfeldes steht, ohne dabei mit seinen Kanten das Rasterfeld zu verlassen und eine Kollision mit benachbarten Objekten zu riskieren. Ausgehend von einem freien Rasterfeld hat Robi Tobi dann die Möglichkeit, die Rasterfelder in seiner Umgebung mit Hilfe seiner Sensoren zu erkunden. Die Felder, die als frei bzw. belegt detektiert werden, kann Robi Tobi in seiner Rasterkarte vermerken, so dass eine erneute Untersuchung der Felder entfallen kann. Auf diese Weise ist es z.B. möglich, Robi Tobi so zu steuern, dass er alle erreichbaren freien Rasterfelder auffindet und befährt. Die Funktionen in Zeilen 13 und 14 in Code 2 dienen dazu einen Wert zwischen 0 und 255 in Robi Tobi's interne Rasterkarte einzutragen bzw. wieder auszulesen.

Der Simulator erlaubt es einen Blick auf Robi Tobi's interne Rasterkarte zu werfen, indem die Karte auf dem Bildschirm ausgegeben werden kann. Initial ist jedes Rasterfeld mit dem Wert 0 gefüllt, d.h. das Rasterfeld wurde noch nicht erkundet (weißes Kästchen). Der Wert 1 in einem Rasterfeld bedeutet, dass das Feld frei befahrbar ist und erkundet wurde (hellgraues Kästchen). Der Wert 255 bedeutet, dass sich in dem Feld ein Objekt befindet und das Feld deswegen nicht befahrbar ist (dunkelrotes Kästchen). Die Bedeutung der Werte 2 bis 254 können in der zu entwickelnden Steuerung frei vergeben werden (hellblaues Kästchen). Mit den Funktionen in Zeilen 17 und 18 in Code 2 kann Robi Tobi auch seine eigene Pose in der Rasterkarte vermerken. Dies dient insbesondere der Visualisierung in der Simulation, in der der Inhalt der

Rasterkarte auf dem Bildschirm ausgegeben werden kann.

Abbildung 4 zeigt eine Landschaft mit 10 Rechtecken nachdem sie von Robi Tobi komplett abgefahren wurde. Die hellblauen Bereiche und Linien sind durch die Scanlinien der Ultraschallsensoren, die in der Simulation visualisiert wurden, entstanden. Der untere Teil der Abbildung stellt die zugehörige Rasterkarte dar, die Robi Tobi bei seiner Erkundungsfahrt angelegt hat.

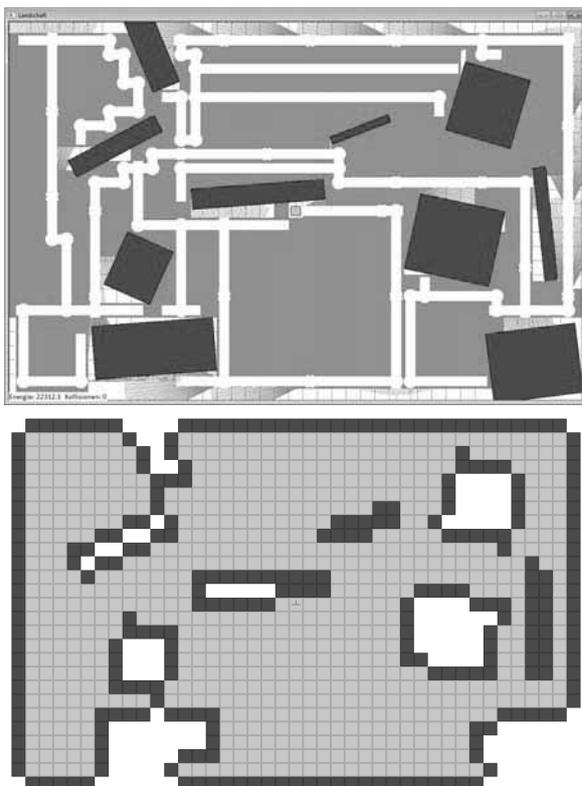


Abbildung 4. Landschaft mit 10 Rechtecken und zugehörige Rasterkarte. Hellblau dargestellt sind die Scanlinien der Ultraschallsensoren, die beim Abfahren der Landschaft entstanden sind.

2.5 Bewertung von Steueralgorithmien für Robi Tobi

Je nach Aufgabenstellung wird Robi Tobi durch seinen Steueralgorithmus auf unterschiedliche Art und Weise durch die Landschaft bewegt. Bei seiner Bewegung verbraucht Robi Tobi Energie. In einem realen Roboterszenario könnte der Energieverbrauch z. B. die Entladung der Antriebsbatterie oder auch verbrauchte (Lebens-)energie in Form von Verschleiß sein. Robi Tobi's Energievorrat wird in der Simulation in Energiepunkten (EP) gemessen. In der aktuellen

Version des Simulators stehen ihm maximal 200000 EP zur Verfügung; danach ist Robi Tobi fahruntfähig. Tabelle 1 zeigt Robi Tobi's Energieverbrauch aufgeschlüsselt nach den jeweiligen Verbrauchsquellen.

Eine Längeneinheit vorwärts- bzw. rückwärtsfahren	1 EP
Um 1 Grad nach links bzw. rechts drehen	0,2 EP
Eine Messung mit einem Ultraschallsensor durchführen	0,1 EP
Kollision mit einem Objekt	1000 EP

Tabelle 1. Robi Tobi's Energieverbrauch.

Die Aufgabe der Studierenden ist es, einen Steuerungsalgorithmus für Robi Tobi bzgl. einer ihm gestellten Aufgabe zu entwickeln. Robi Tobi's Aufgabe könnte es dabei z.B. sein, alle für ihn erreichbaren Rasterfelder einer unbekanntem Landschaft zu befahren und in seiner Rasterkarte zu vermerken. Zur Kontrolle des Ergebnisses der Erkundungsfahrt, stellt der Simulator eine Funktion bereit, mit der überprüft werden kann, ob die in der Rasterkarte vermerkten Felder den korrekten Status, d.h. frei oder belegt, haben. Neben der Erfüllung der Aufgabenstellung wird die benötigte Energie als Bewertungskriterium mit einbezogen. So ist es neben den rein funktionalen Anforderungen auch ein Ziel, den Energieverbrauch zu minimieren.

3 Architektur und Funktionsweise des Simulators

3.1 Softwarearchitektur

In Kapitel 2 wurde die Schnittstelle zur Konfiguration des Simulators und zur Steuerung von Robi Tobi dargestellt. Diese Schnittstelle ist rein prozedural realisiert, d.h. Robi Tobi wird durch Aufruf (globaler) C-Funktionen gesteuert. Dies entspricht der Sichtweise, die ein Entwickler eines Steueralgorithmus für einen echten Roboter hat. Typischerweise wird der Zugriff auf die Hardwareregister zum Abfragen der Sensoren und Ansteuern der Aktuatoren über einen sogenannten Hardware Abstraction Layer (HAL) in Form von C-Funktionen abstrahiert. Dank des HALs müssen keine direkten Zugriffe auf die Register der Hardwarebausteine des zugrundeliegenden Mikrocontrollers aus dem Steueralgorithmus heraus erfolgen. Die verwendete Mikrocontroller-Plattform wird dadurch leichter austauschbar, so dass eine Portierung der Steuerungssoftware auf eine andere zugrundeliegende Hardwarearchitektur vereinfacht wird.

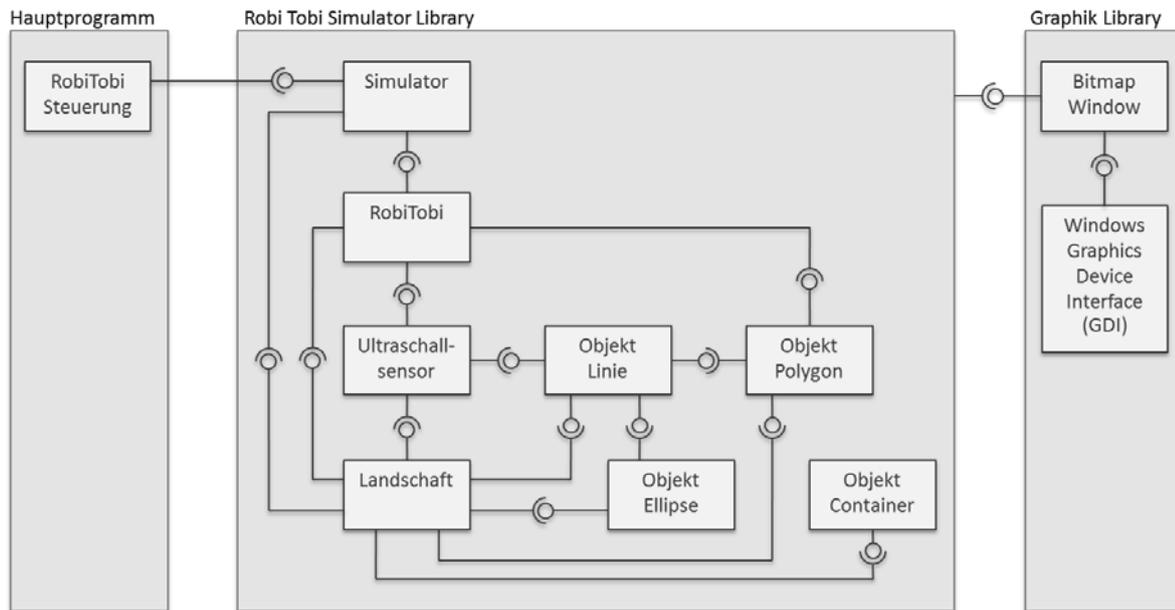


Abbildung 5. Softwarearchitektur des Simulators in Komponentendarstellung.

Im Gegensatz zu der Schnittstelle ist die Implementierung des Simulators selbst objektorientiert aufgebaut. So wird z.B. Simulator-intern die Funktionalität von Robi Tobi in einer C++ Klasse gekapselt. Dies schafft eine bessere Strukturierung der Software und mehr Flexibilität. In einer zukünftigen Version könnte der Simulator z.B. leichter für die Betrachtung von Multi-Roboter-Szenarien, in denen mehrere Instanzen von Robi Tobi kooperativ an einer Gesamtaufgabe arbeiten, erweitert werden [1].

Abbildung 5 zeigt die grobe Softwarearchitektur des Simulators in Form eines Komponentendiagramms. Der Simulator wird als C++-Library (mittlerer hellblauer Kasten) mit einem Hauptprogramm (linker hellblauer Kasten) zu einer ausführbaren Datei (*.exe) verlinkt. Das Hauptprogramm übernimmt die Konfiguration des Simulators und implementiert den Steueralgorithmus für Robi Tobi gemäß einer zu bearbeitenden Aufgabe. Die Simulator Library nutzt ihrerseits eine Graphik Library (rechter hellblauer Kasten), um ein Windows-Fenster für die darzustellende Landschaft zu öffnen und graphische Objekte wie Punkte, Linien, Ellipsen und Polygone in einer Bitmap-Graphik auszugeben. Die Graphik Library wurde in einem vorherigen Projekt entwickelt. Letztendlich baut sie auf dem Graphics Device Interface (GDI) des Windows-Kernels auf [7].

Die Implementierung der Simulator Library teilt sich in mehrere Komponenten auf, die im Wesentlichen in separaten Klassen realisiert sind (siehe Abbil-

dung 5). Der Einstiegspunkt ist die Komponente *Simulator*. Hier wird gemäß der Simulationskonfiguration sowohl die Landschaft als auch eine Instanz von Robi Tobi erzeugt und Robi Tobi in der Landschaft platziert. Um den Steueralgorithmus für Robi Tobi möglichst ungestört laufen zu lassen, wird dieser in einem separaten Windows-Thread gestartet. Im Haupt-Thread des Simulators werden die Ereignisse der Windows-Message-Queue abgearbeitet und dadurch die graphische Ausgabe des Landschaftsfensters in regelmäßigen Zeitabständen aktualisiert. Sowohl die Steuerfunktionen von Robi Tobi als auch der Simulator greifen aus nebenläufigen Threads auf die Bitmap-Graphik der Landschaft als gemeinsam genutzte Ressource zu (sog. kritischer Abschnitt). Um Dateninkonsistenzen, die bei einem Context-Switch der Threads im kritischen Abschnitt entstehen können, zu vermeiden, werden die Zugriffe auf die Bitmap-Graphik mit Hilfe einer Mutex-Variable synchronisiert. Auf diese Weise wird ein wechselseitiger Ausschluss beim Zugriff auf die gemeinsam genutzte Ressource garantiert.

Die Komponente *Landschaft* ist für die Erzeugung der Landschaft in Form einer Bitmap-Grafik zuständig. Eine Landschaft wird im Simulator als ein Array von Objekten gespeichert. Ein Objekt ist dabei eine abstrakte Klasse, von der vier unterschiedliche Typen abgeleitet sind: *Linie*, *Polygon*, *Ellipse*, *Container* (siehe Abbildung 6). Jeder Objekttyp muss die virtuellen Methoden der Basisklasse *Objekt* implementie-

ren. Objekte können verschoben, gedreht und skaliert werden, ein Objekt kann in die Bitmap-Grafik der Landschaft gezeichnet werden und es können die Schnittpunkte eines Objektes mit einem anderen Objekt bestimmt werden. Insbesondere das Bestimmen von Schnittpunkten wird an unterschiedlichen Stellen im Simulator verwendet. Zum einen wird durch das Auffinden von Schnittpunkten zwischen Objekten sichergestellt, dass diese sich bei einer zufälligen Positionierung in der Landschaft nicht überlagern. Des Weiteren ist Robi Tobi selbst auch nur ein Objekt, nämlich ein Quadrat mit einer Kantenlänge von 20 Längeneinheiten, das bei seiner Fahrt mit anderen Objekten nicht kollidieren darf. Zu guter Letzt ist die Scanlinie eines Ultraschallsensors ein Linien-Objekt, zu dem der Schnittpunkt mit dem nächstgelegenen Objekt bestimmt werden muss. Ein Container-Objekt ist ein spezieller Objekttyp der es erlaubt, neue Objekte aus den drei Grundobjekten Linie, Polygon und Ellipse zu erzeugen. Das HSHL-Logo ist z.B. ein Container-Objekt, welches aus zwei Polygonen besteht.

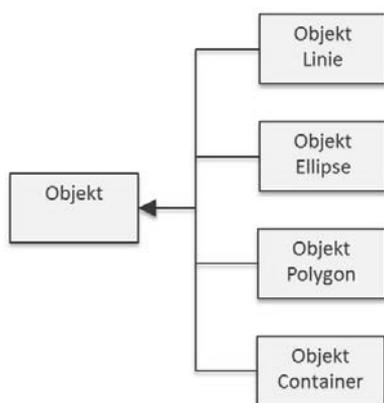


Abbildung 6. Abgeleitete Klassen der abstrakten Klasse Objekt.

Die Komponente *RobiTobi* ist in Form einer Klasse realisiert. Die Methoden der Klasse spiegeln im Wesentlichen die Funktionen zur Konfiguration und zum Steuern von Robi Tobi aus der Schnittstelle des Simulators wider. Ein Aufruf dieser Funktionen wird über die Schnittstelle des Simulators an eine Instanz von Robi Tobi weitergeleitet. Eine Instanz von RobiTobi kann mit einer beliebigen Anzahl an Ultraschallsensoren in beliebiger Ausrichtung ausgestattet werden. Die Komponente *Ultraschallsensor* ist ebenfalls in Form einer Klasse realisiert. Immer dann, wenn ein Sensorwert von Robi Tobi abgefragt wird, wird ein neues Liniensegment (Objekt Linie) mit der Richtung des Ultraschallsensors und einer Länge von

200 Längeneinheiten im Befestigungspunkt des Sensors angelegt und der nächstgelegene Schnittpunkt zu einem Objekt der Landschaft bestimmt.

3.2 Berechnung von Schnittpunkten zwischen Objekten

Wie in Kapitel 3.1 erwähnt, müssen an unterschiedlichen Stellen im Simulator die Schnittpunkte zwischen Objekten bestimmt werden. Beispielsweise wird die Scanlinie eines Ultraschallsensors durch ein Liniensegment dargestellt. Zur Bestimmung des nächstgelegenen Schnittpunkts des Liniensegments mit einem Objekt der Landschaft, müssen die Schnittpunkte des Liniensegments mit allen Objekten der Landschaft bestimmt werden. Um eine Vorauswahl der Objekte zu treffen, die überhaupt für einen Schnittpunkt in Frage kommen, werden dabei zunächst die Bounding-Boxen der Objekte auf Überlappung getestet. Eine Bounding-Box ist dabei ein achsenparalleles Rechteck. Ob sich zwei dieser Rechtecke schneiden kann sehr einfach festgestellt werden. Nur für die in Frage kommenden Schnittobjekte wird dann eine exakte Schnittpunktberechnung durchgeführt. Da Polygon-Objekte aus mehreren Linien-Objekten zusammengesetzt sind, reduziert sich die Schnittpunktberechnung auf zwei unterschiedliche Fälle:

- (i) Schnittpunkte zwischen zwei Liniensegmenten
- (ii) Schnittpunkte zwischen einem Liniensegment und einer Ellipse

Zwei Liniensegmente mit den Endpunkten P_1 und P_2 bzw. Q_1 und Q_2 können jeweils durch ihre Geradengleichungen $P_1 + \lambda(P_2 - P_1)$ und $Q_1 + \mu(Q_2 - Q_1)$ mit $\lambda, \mu \in [0, 1]$ beschrieben werden. Den Schnittpunkt der Geraden erhält man aus der Lösung (λ^*, μ^*) des linearen Gleichungssystems:

$$\lambda(P_2 - P_1) - \mu(Q_2 - Q_1) = Q_1 - P_1$$

Die Lösung des linearen Gleichungssystems kann z.B. mit der Cramerschen Regel berechnet werden. Falls die Koeffizientendeterminante des Gleichungssystems ungleich 0 ist, lässt sich das Gleichungssystem eindeutig lösen, d.h. die beiden Geraden schneiden sich in genau einem Punkt. Die zugehörigen Liniensegmente schneiden sich allerdings nur dann, wenn für die Lösung gilt: $\lambda^*, \mu^* \in [0, 1]$. Hat die Koeffizientendeterminante des Gleichungssystems den Wert 0, dann sind die beiden Liniensegmente kollinear, d.h. sie haben entweder keine Schnittpunkte

oder liegen übereinander. Diese beiden Fälle müssen gesondert unterschieden werden.

Die Schnittpunkte zwischen einem Liniensegment und einer Ellipse werden mit Hilfe der Geradengleichung $y = m \cdot x + c$ mit Steigung m und y -Achsenabschnitt c und der Ellipsengleichung $x^2/a^2 + y^2/b^2 = 1$ mit x -Radius a und y -Radius b bestimmt. Setzt man die Geradengleichung in die Ellipsengleichung ein, erhält man eine quadratische Gleichung, die je nach Diskriminante keine, eine oder zwei Lösungen hat. Die Lösungen entsprechen den Schnittpunkten zwischen der Geraden und der Ellipse. Ob sich die Schnittpunkte ebenfalls auf dem Liniensegment befinden, muss separat getestet werden. Des Weiteren muss ein Liniensegment, welches parallel zur y -Achse verläuft, als Sonderfall betrachtet werden, da dieses in der Form einer expliziten Funktion für die Gerade nicht dargestellt werden kann. Ellipsen deren Hauptachsen nicht den Koordinatenachsen entsprechen, werden derzeit im Simulator nicht unterstützt.

Eine Berechnung der Schnittpunkte zweier Ellipsen wird im Simulator derzeit ebenfalls nicht unterstützt. Die Berechnung solcher Schnittpunkte wäre nur bei der Generierung von Landschaften mit Ellipsen, die sich nicht überlappen dürfen von Bedeutung. Hier wird bisher nur, etwas grob-granularer, eine Überlappung der Bounding-Boxen der Ellipsen ausgeschlossen.

Eine weitere Möglichkeit der Überschneidung von Objekten muss noch betrachtet werden. Es kann vorkommen, dass ein Objekt in einem anderen Objekt enthalten ist, z.B. könnte sich Robi Tobi innerhalb eines Polygon-Objekts befinden, ohne dass sich die Außenkanten überschneiden. In so einem Fall sollte die Kollision der Objekte auch erkannt werden. Jedes Objekt ist dazu mit einem Referenzpunkt ausgestattet (i.d.R. der Mittelpunkt des Objektes). Befindet sich der Referenzpunkt eines Objektes in einem anderen Objekt, dann kollidieren die beiden Objekte. Für Polygon-Objekte ist hier ein einfacher Punkt-in-Polygon-Test implementiert, welcher die Anzahl der Schnittpunkte eines Strahls ausgehend vom Referenzpunkt parallel zur positiven x -Achse mit dem Polygon bestimmt. Bei einer ungeraden Anzahl an Schnittpunkten liegt der Referenzpunkt innerhalb des Polygons und bei einer geraden Anzahl außerhalb. Zusätzlich müssen die Randfälle betrachtet werden, in denen der Strahl durch einen Eckpunkt oder auf

einer Kante des Polygons verläuft oder der Referenzpunkt selbst auf einer Kante des Polygons liegt.

4 Anwendungsszenarien und Lösungsskizzen

4.1 Überblick zu ausgewählten Anwendungsszenarien

Mit Hilfe des Simulators können Algorithmen für unterschiedlichste Anwendungsszenarien entwickelt, ausprobiert und bzgl. ihrer Güte bewertet werden. Es kann zum einen überprüft werden, ob ein entwickelter Algorithmus eine vorgegebene Aufgabenstellung löst, und zum anderen können funktionsfähige Lösungsalgorithmen bzgl. Robi Tobi's Energieverbrauch verglichen werden. Eine Reihe von Anwendungsszenarien gehen davon aus, dass Robi Tobi in einer unbekanntem Landschaft an einer ihm unbekanntem Position und Ausrichtung ausgesetzt wird. Die Landschaft ist entweder nach außen hin begrenzt, so dass Robi Tobi die Landschaft nicht verlassen kann, oder nach außen hin offen bzw. an mindestens einer Stelle geöffnet. Im zweiten Fall könnte Robi Tobi die Landschaft verlassen. Aufgabenstellungen für diese Szenarien könnten z.B. sein:

- (i) Robi Tobi soll eine Rasterkarte der Landschaft erstellen, welche alle von seiner Startposition aus erreichbaren Rasterfelder verzeichnet. In einem solchen Szenario wird eine nach außen begrenzte Landschaft vorausgesetzt.
- (ii) Robi Tobi soll einen Ausweg aus der Landschaft finden. Die Landschaft könnte hier z.B. ein Labyrinth mit einem oder mehreren Ausgängen sein.
- (iii) Robi Tobi soll einen Zielpunkt innerhalb der Landschaft ansteuern, wobei er entweder (a) die Koordinaten des Zielpunktes relativ zu seiner Startposition und Ausrichtung kennt, oder (b) er nur den Abstand zum Zielpunkt messen kann.

Andere Anwendungsszenarien gehen davon aus, dass Robi Tobi bereits eine Karte der Landschaft in seinem Speicher hat. Hier könnte es das Ziel sein, einen Ausgang oder einen Zielpunkt in der Landschaft auf dem kürzesten Weg anzusteuern, sog. Pfadfindung. Eine Implementierung wäre z.B. auf Basis des Algorithmus von Dijkstra bzw. dem A*-Algorithmus möglich [5]. Eine weitere Aufgabe könnte es sein, die Landschaft auf möglichst kürzestem Wege vollständige abzufahren und am Ende

wieder am Ausgangspunkt herauszukommen. Ein derartiges Verhalten würde man z.B. von einem Rasenmäherroboter erwarten.

4.2 Erkunden von Rasterfeldern

Die Algorithmen, die auf der Erstellung einer Rasterkarte beruhen, haben gemeinsam, dass Robi Tobi mit seinen Sensoren zuverlässig erkennen muss, ob ein Rasterfeld befahrbar, also *frei*, oder aufgrund eines Hindernisses nicht befahrbar, also *belegt*, ist. Wir gehen zunächst davon aus, dass Robi Tobi mit nur einem Ultraschallsensor, der mittig an seiner Vorderseite angebracht ist, ausgestattet ist. Abbildung 7 zeigt ein mögliches Szenario mit einer Landschaft, in der sich zwei dreieckige Objekte (dunkelrot) befinden. Robi Tobi's Blickrichtung zeigt nach rechts. Sein Ultraschallsensor ist als kleine rote Linie mittig an seiner Vorderseite zu sehen. Legt man den Ursprung eines Koordinatensystems in Robi Tobi's Mittelpunkt, zeigt der Ultraschallsensor in Richtung der positiven x-Achse. Freie Rasterfelder sind in Abbildung 7 weiß ausgefüllt. Belegte Rasterfelder sind schwarz schraffiert dargestellt. Um das Rasterfeld unmittelbar vor Robi Tobi zu erkunden, muss er dieses mit seinem Ultraschallsensor abtasten. Er kann sich dafür z.B. 45° nach links und 45° nach rechts drehen und dabei jeweils in 1° -Schritten Abstandsmessungen durchführen. Aus einem gemessenen Abstand und dem Winkel des Sensors können die x- und y-Koordinaten eines Objektes in der Richtung des Sensors bestimmt werden. Falls unter dem Winkel $\alpha \in [-45^\circ, 45^\circ]$ der Abstand $d < 200$ gemessen wird, befindet sich der Schnittpunkt des Sensorstrahls mit einem Objekt an den Koordinaten:

$$x_\alpha = (d + 10) \cdot \cos \alpha \quad \text{und} \quad y_\alpha = (d + 10) \cdot \sin \alpha$$

Das Rasterfeld vor Robi Tobi ist frei, falls für alle Winkel α gilt:

$$(x_\alpha, y_\alpha) \notin [15, 45] \times [-15, 15]$$

Aufgrund der Sensormesswerte kann nicht nur das Feld unmittelbar vor Robi Tobi bewertet werden, sondern bis zu sechs vor ihm liegende Felder. Es wird dazu einfach der Reihe nach geprüft, ob die Schnittpunkte (x_α, y_α) in eines der Felder $[15 + 30k, 45 + 30k] \times [-15, 15]$ für $k = 0, 1, \dots, 5$ fallen. Je nachdem wie groß die kleinsten Objekte der Landschaft sind, die noch als Objekte erkannt werden sollen, muss die Abstandsmessung eventuell in Schritten kleiner als 1° durchgeführt werden. Bei einer Schrittweite von 1°

können Objekte mit einer Kantenlänge von weniger als ca. 3,7 Längeneinheiten unerkant bleiben.

Abbildung 8 zeigt den durch den Ultraschallsensor abgedeckten Sichtbereich von Robi Tobi bei einer Drehung um 45° nach links und 45° nach rechts. Zwei Rasterfelder in Robi Tobi's Blickrichtung würden als frei identifiziert.

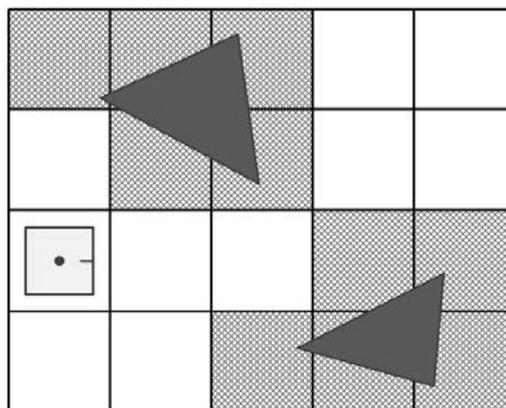


Abbildung 7. Robi Tobi mit einem Ultraschallsensor in einer Beispiellandschaft.

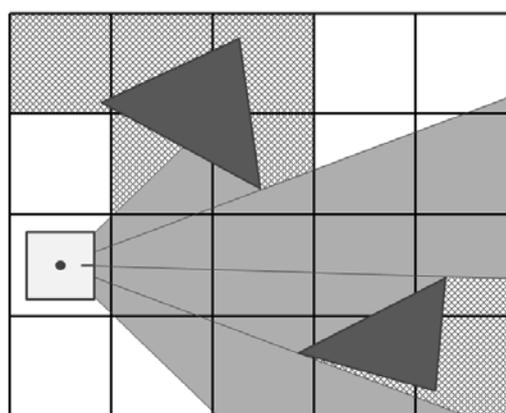


Abbildung 8. Robi Tobi mit seinem Sichtbereich zwischen -45° und 45° .

Eine Verfeinerung des Verfahrens erhält man unter der Annahme, dass Robi Tobi sich nur horizontal und vertikal und nicht diagonal zwischen den Rasterfeldern bewegt. In jeder Ecke eines Rasterfeldes gibt es dann ein Quadrat von 5×5 Längeneinheiten, welches Robi Tobi niemals durchfährt und auch bei einer Drehung nicht berührt. Berücksichtigt man diese Tatsache, dann reicht bereits eine Abtastung von 27° nach links und 27° nach rechts aus, um eine Aussage darüber zu erhalten, ob die Felder vor Robi Tobi frei oder belegt sind.

Eine andere Taktik, die Rasterfelder in Robi Tobi's Umgebung zu bewerten, könnte es sein, nicht nur die Felder unmittelbar vor Robi Tobi zu untersuchen, sondern die Abstände zu Objekten im 360° Winkel zu messen. Aus den gemessenen Abständen lassen sich wie oben dargestellt die Schnittpunkte (x_α, y_α) für $\alpha \in [0^\circ, 360^\circ]$ berechnen. Das Rasterfeld, in welches der Schnittpunkt fällt, gilt als belegt. Die Rasterfelder, die sich aus Sicht von Robi Tobi hinter dem belegten Feld befinden, liegen im Schattenschein und können nicht als frei bewertet werden. Alle Rasterfelder im Umkreis der Sensorreichweite um Robi Tobi's Mittelpunkt, die nicht als belegt oder im Schattenschein identifiziert werden, sind frei.

In den beschriebenen Verfahren wurde jeweils nur ein Ultraschallsensor verwendet, um Informationen über Robi Tobi's Umgebung zu erhalten. Es gibt zahlreiche Möglichkeiten, weitere Sensoren an Robi Tobi's Außenwand anzubringen, um insgesamt mit weniger Energieverbrauch ein gleiches Ergebnis zu erzielen, da eine Messung mit einem Ultraschallsensor weniger Energie kostet als eine Dreh- oder Fahrbewegung. Unterschiedliche Varianten sollen hier aber nicht diskutiert werden, da sie den Rahmen dieses Artikels sprengen würden.

4.3 Vollständiges Erkunden einer unbekanntem Landschaft

Die Möglichkeit zu bewerten, ob ein Rasterfeld frei oder belegt ist, stellt die Basis für viele Anwendungsszenarien dar. Beispielhaft soll hier das Anwendungsszenario, in dem Robi Tobi alle von seiner Startposition aus erreichbaren Rasterfelder erkunden soll, betrachtet werden. Eine solche Aufgabe kann z.B. mit einem Backtracking-Algorithmus in Form einer Tiefensuche gelöst werden: Ist das Rasterfeld vor Robi Tobi noch nicht erkundet, d.h. frei und bisher unbesucht, dann fährt er dort hin und startet seine Erkundungsfahrt mit diesem Feld als Startfeld erneut. Ist das Rasterfeld hingegen belegt oder bereits erkundet, dreht er sich, so dass er in Richtung eines Nachbarfeldes ausgerichtet ist und startet seine Erkundungsfahrt in die neue Richtung. Sind alle Nachbarfelder erkundet, dann fährt er wieder ein Feld zurück und beendet seine zuletzt gestartete Erkundungsfahrt. Der Backtracking-Algorithmus lässt sich recht einfach als rekursive Funktion implementieren. Es gibt zahlreiche Möglichkeiten, Robi Tobi's Energieverbrauch bei seiner Erkundungsfahrt zu minimieren. Dies fängt damit an, ihn immer auf kürzestem Wege in die nächste frei Richtung zu drehen, bis hin zu Verbesse-

rungsstrategien, die Robi Tobi's Rückfahrt zur letzten Kreuzung, an der noch weitere Richtungen untersucht werden müssen, optimiert. Die Rückfahrt könnte z.B. dahingehend optimiert werden, dass Robi Tobi die Zielkreuzung auf kürzestem Wege, unter Berücksichtigung seiner bisher erstellten Rasterkarte, anfährt.

Neben der Realisierung eines Backtracking-Algorithmus zur Lösung der Aufgabenstellung kommen auch Verfahren zum Durchlaufen von Labyrinth in Frage. Die Aufgabe, eine Landschaft vollständig zu erkunden, entspricht der Aufgabe, einen Ausweg aus einem Labyrinth zu finden, wobei der Ausweg auch im inneren des Labyrinths, z.B. in Form einer Bodenluke, liegen kann. Ein Algorithmus, der dies leistet, ist z.B. der Trémaux-Algorithmus [8]. Wie beim Backtracking wird das Wegesystem der Landschaft genau zweimal abgelaufen, allerdings in einer anderen Reihenfolge.

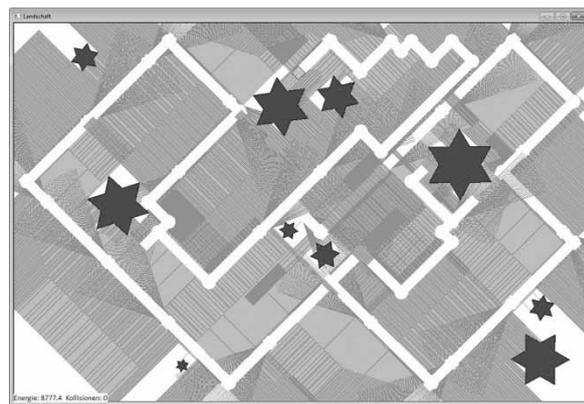


Abbildung 9. Robi Tobi bei einer Erkundungsfahrt in einer Landschaft mit Sternen.

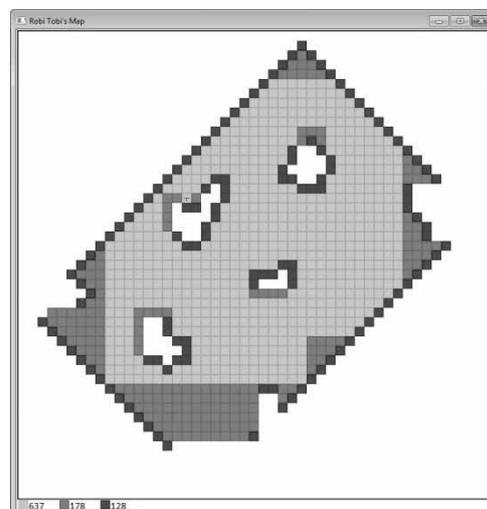


Abbildung 10. Robi Tobi's Rasterkarte zur Landschaft aus Abbildung 9.

Abbildung 9 zeigt wie Robi Tobi's Erkundungsfahrt im Simulator visualisiert wird. Die zu durchfahrende Landschaft besteht aus 10 Polygonen in Sternform unterschiedlicher Größe. Die hellblauen Linien stellen wieder die Scanlinien der Ultraschallsensoren dar. In dem Beispiel nutzt Robi Tobi drei Ultraschallsensoren, die vorne, rechts und links angebracht sind. Die Landschaft wird mittels eines Backtracking-Algorithmus erkundet. Abbildung 10 zeigt die zugehörige Sicht auf Robi Tobi's interne Rasterkarte kurz vor Ende der Erkundungsfahrt. Hellgraue Felder hat Robi Tobi als frei und befahren bewertet. Hellblaue Felder gelten als frei aber noch nicht befahren und dunkelrote Felder gelten als belegt. Die weißen Bereiche sind noch unbekannte Felder. Freie, unbefahrene Felder müssen nicht zwangsläufig von Robi Tobi befahren werden; sie stellen aber i.d.R. einen Weg zu unbekanntem Feldern dar. Im Vergleich zu der Landschaft in Abbildung 9 ist Robi Tobi's interne Rasterkarte aus Abbildung 10 um 45° nach links gedreht. Dies liegt daran, dass Robi Tobi seine Erkundungsfahrt in einem Winkel von -45° zur x-Achse gestartet hat, er in seiner Rasterkarte aber mit einem Startwinkel von 0° beginnt.

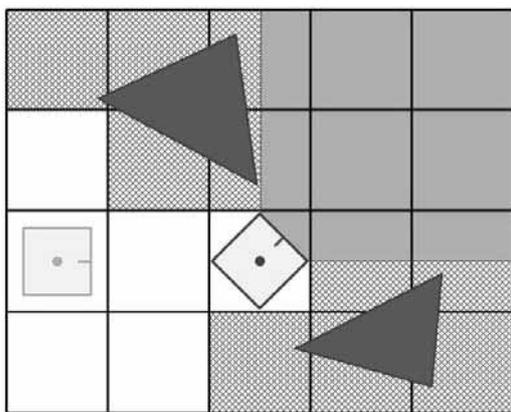


Abbildung 11. Robi Tobi mit Diagonalfahrt zwischen Rasterfeldern.

Als letztes sollen noch zwei Verbesserungsmöglichkeiten zum Erkunden von Landschaften diskutiert werden. Schaut man sich die Landschaft in Abbildung 7 an, dann sind die vier freien Rasterfelder in der rechten oberen Ecke der Landschaft für Robi Tobi nicht erreichbar, da der Zugang durch die beiden dreieckigen Objekte versperrt ist. Andererseits stellt man fest, dass der Platz zwischen den Objekten prinzipiell dafür reichen würde, Robi Tobi in den freien Bereich fahren zu lassen. Erlaubt man für Robi Tobi neben der horizontalen und vertikalen Fahrt zwischen

zwei Rasterfeldern auch eine diagonale Fahrt, dann wäre der Durchschlupf für ihn passierbar (siehe Abbildung 11). Eine andere Möglichkeit, um Robi Tobi den freien Bereich entdecken zu lassen, ist es, die Rasterkarte zu verfeinern. Abbildung 12 zeigt beispielsweise eine Rasterkarte mit Rasterfeldern halber Kantenlänge. Robi Tobi belegt dann insgesamt immer vier freie Rasterfelder. Wie an der Abbildung zu erkennen ist, kann Robi Tobi in diesem Szenario den Durchschlupf auch ohne Diagonalfahrt passieren.

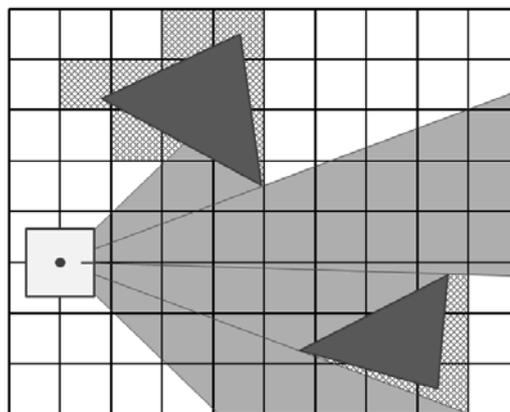


Abbildung 12. Rasterung der Landschaft mit Rasterfeldern halber Kantenlänge.

5 Zusammenfassung und Ausblick

Der Artikel stellt einen Simulator für einen autonomen mobilen Roboter vor. Neben der Beschreibung der Schnittstelle des Simulators werden auch seine Funktionsweise und seine objektorientierte Softwarearchitektur dargestellt. Es werden unterschiedliche Anwendungsszenarien beschrieben, wobei insbesondere das Bewerten von Rasterfeldern (frei oder belegt) als auch das vollständige Erkunden von Landschaften in einigen Details dargestellt wird. Der Simulator wurde im Lehrbetrieb an der Hochschule Hamm-Lippstadt im Sommersemester 2014 und 2015 mit Erfolg eingesetzt. Die von den Studierenden entwickelten Algorithmen zur Erkundung einer unbekanntem Landschaft übertrafen zumindest zum Teil die ursprünglichen Erwartungen.

Im Vergleich zur Realität ist das im Simulator abgebildete Szenario in vielen Aspekten idealisiert. So sind z.B. die Messwerte von realen Sensoren mit Fehlern behaftet und auch das Ansteuern von Aktuatoren führt zu einer fehlerbehafteten Ausführung einer Fahrbewegung, so dass Robi Tobi's Position und Ausrichtung mit Unsicherheit behaftet sind. In einer zukünftigen Version des Simulators sollen solche

Fehler in Form von normalverteilten Zufallsvariablen mit konfigurierbarer Varianz hinzugeschaltet werden können. Das Problem der Positionsbestimmung und Kartierung einer unbekanntem Landschaft ist als Simultaneous Localization and Mapping (SLAM) bekannt [2], [4]. Sowohl die Position und Ausrichtung des Roboters als auch die Rasterfelder der Landschaft werden dabei durch Zufallsvariablen modelliert.

Eine weitere Ausbaustufe des Simulators ist die Unterstützung von Multi-Roboter-Szenarien. Durch die objektorientierte Gestaltung des Simulators ist es möglich nicht nur eine Instanz von Robi Tobi sondern mehrere Roboter in der Landschaft zu positionieren. Eine Kommunikation der Roboter wird durch (drahtlosen) Zugriff auf einen gemeinsam genutzten Speicher in einer Basisstation realisiert. Dies ermöglicht es z.B. kooperative Algorithmen in einem Multi-Roboter-Szenario für das Erkunden von Landschaften zu implementieren [1], [3].

6 Literatur

- [1] S. Behnke, Kooperierende Mobile Roboter, *at - Automatisierungstechnik* **61**(4), 233-244, Oldenburg Verlag, 2013.
- [2] C. Brenner, SLAM lecture, unter: <http://www.clausbrenner.de/slam.html> (abgerufen: 22.1.2016)
- [3] W. Burgard, M. Moors, C. Stachniss, F.E. Schneider, Coordinated Multi-Robot Exploration, *IEEE Transactions on Robotics* **21**, 376-386, 2005.
- [4] J. Hertzberg, K. Lingemann, A. Nüchter, Mobile Roboter – Eine Einführung aus Sicht der Informatik, Springer Vieweg, 2012.
- [5] M. Herzog, W. F. Riedl, L. Velden, Der A* - Algorithmus, Technische Universität München, unter: <https://www-m9.ma.tum.de/graph-algorithms/spp-a-star> (abgerufen: 22.1.2016)
- [6] M. Maurer, J. Ch. Gerdes, B. Lenz, H. Winner (Hrsg.), Autonomes Fahren: Technische, rechtliche und gesellschaftliche Aspekte, Springer Vieweg, 2015.
- [7] Microsoft Windows graphics device interface (GDI), unter: <https://msdn.microsoft.com/en-us/library/windows/desktop/dd145203%28v=v.s.85%29.aspx> (abgerufen: 22.1.2016)
- [8] Trémaux' Methode, wikipedia, unter: https://de.wikipedia.org/wiki/Tr%C3%A9maux%E2%80%99_Methode (abgerufen: 22.1.2016)

Brandschutzausbildung mittels e-learning und graphischer Simulation

Prof. Dr-Ing. Dieter Wloka, M.Eng.
Fachgebiet Technische Informatik
FB 16 Elektrotechnik/Informatik
Universität Kassel
Wilhelmshöher Allee 71-73
34128 Kassel

dwloka@uni-kassel.de
www.katie-katastrophensimulation.de

1 Abstract

Das Projekt Katie (KI, Avatare, Training - in virtuellen Environments) [2] hat das Ziel,

- eine graphische Simulation von Katastrophenszenarien,
- in einer e-learning Umgebung,
- mit integrierten serious game Komponenten, zu schaffen.

Für Katie werden verschiedenste Technologien eingesetzt und verknüpft, wie

- E-learning
- Virtuelle Realität
- 3D-Modellierung
- Game-Engines
- Darstellung virtueller Menschen
- Immersion durch CAVE oder HMD

Das Projekt Katie unterstützt die Ausbildung, indem Handlungen und Lösungsstrategien aufgezeigt werden. Über KATIE wurde schon ausführlich berichtet, u.a. in [9], [10], [4], [8], [5], [6], [3] und [7].

Ein bedeutender Bereich ist die Prävention von Bränden. Ein Teilprojekt befasst sich mit der Entwicklung von Ausbildungsmedien zur Brandschutzunterweisung nach den Technischen Regeln für Arbeitsstätten. In diesem Beitrag wird ein Überblick über das Teilprojekt sowie Erfahrungen beim realen Einsatz mit Hilfe der Software gegeben.

2 Technischen Regeln für Arbeitsstätten

Die Technischen Regeln für Arbeitsstätten (ASR) ¹ konkretisieren die Anforderungen der Arbeitsstättenverordnung (ArbStättV). Die Regelungen der ASR A2.2 umfassen:

- Grundschutz: Deckt Gefahren, vergleichbar einer Büronutzung, ab. Für diesen Grundschutz werden weiterhin über Tabellen die Anzahl von Feuerlöschern sowie weiterhin die Anrechnung von Wandhydranten festgelegt.
- Erhöhte Brandgefahr: Darüber hinausgehende Gefährdungen werden als erhöhte Brandgefahr bezeichnet. Hier tritt die bereits in dem Arbeitsschutzgesetz geforderte Gefährdungsbeurteilung stärker in den Vordergrund.
- Berücksichtigung anderer Verordnungen: andere Verordnungen, wie z.B. der Gefahrstoffverordnung, sind zu berücksichtigen
- Verkürzte Reaktionszeit bei der Brandmeldung: Stärker berücksichtigt wird auch eine Verkürzung der Reaktionszeit durch schnelle Brandmeldung über Brandmelder oder Brandmeldesysteme. Zusätzlich soll sich die Reaktionszeit der Selbsthilfekräfte durch schnelles Auffinden der nächstgelegenen Löschgeräte möglichst verringern.
- Regelmäßige Unterweisung, Brandschutzhelfer: Die regelmäßige Unterweisung aller Mitarbeiter ² bzw. die Benennung und Ausbildung von Brandschutzhelfern und

¹www.baua.de/de/Themen-von-A-Z/Arbeitsstaetten/ASR/ASR-A2-2.

²im Text sind natürlich immer Frauen und Männer gemeint, auch wenn eine maskuline Form benutzt wird

die wiederkehrende Instandhaltung und Prüfung aller brandschutzrelevanten technischen Einrichtungen sichern die Maßnahmen bei Entstehungsbränden zusätzlich ab.

Wichtig ist regelmäßig unterweisen. Natürlich genügt eine ausreichende Anzahl an tragbaren Feuerlöschern nicht, um einen Brand abzuwehren, wenn es zu wenig unterwiesene Kräfte gibt oder diese nicht umfassend geschult sind. Die ASR A2.2 fordert deshalb, alle Mitarbeiter in regelmäßigen Abständen, mindestens einmal jährlich, in wesentlichen Fragen des Verhaltens im Gefahrenfall zu unterweisen.

3 Brandschutzhelfer

Zusätzlich muss der Arbeitgeber eine ausreichende Anzahl von Beschäftigten zu Brandschutzhelfern ernennen. Ausreichend heißt: ein Anteil von 5 % der Beschäftigten ist als Brandschutzhelfer auszubilden.

Auch die Brandschutzhelfer sind zu unterweisen:

- Grundzüge des vorbeugenden Brandschutzes
- Gefahren durch Brände
- Verhalten im Gefahrenfall
- Kenntnisse der betrieblichen Brandschutzorganisation
- Funktion und richtiger Einsatz der Löscheräte
- praktische Übung im Umgang mit Löscheräten

4 Aktuelle Ausbildungssituation

Viele Unternehmen versuchen ihrer Pflicht zur Unterweisung nachzukommen. Hierzu werden oft externe Trainer engagiert, die eine Schulung durchführen. Diese Präsenzs Schulung besteht typischerweise aus einem Theorieteil sowie praktischen Demonstrationen.

Hierbei werden sehr oft im Freien Brände, die mittels Gasfeuermodellen gezündet werden mit Feuerlöschern gelöscht. Manche Modelle verfügen über Aufsätze, die grob Einrichtungsgegenstände abbilden, wie z.B. Computer-Bildschirme.

Bei Unternehmen mit mehreren tausend Mitarbeitern entsteht hierbei ein hoher Aufwand. Auch der Einsatz einer großen Anzahl von Feuerlöschern die aktiviert werden, ist zu hinterfragen.

5 KATIE-Teilprojekt Brandschutzunterweisung

Das KATIE-Teilprojekt Brandschutzunterweisung versucht die Ausbildung durch die Kombination zweier Elemente durchzuführen:

- E-Learning Phase für das Erlernen der Theorie
- Praxisphase durch Simulation

Geht man von einem größeren Unternehmen aus, ergeben sich zunächst die folgenden Fragen:

- Wie geschieht der Aufbau der Kursinhalte, inwieweit sind diese unternehmensspezifisch ?
- Klärung der Hosting- und Zugriffsfrage
- Klärung der Teilnehmerverwaltung

In der Praxis ist man gut beraten, sich zunächst mit dem "Teilnehmer" zu befassen und die Definition eines Ablauf- / und Teilnehmermodells vorzunehmen:

- Wie wird der Teilnehmer informiert ?
- Hat der Teilnehmer Zugriff auf einen Rechner und ein Intranet oder das Internet ?
- Wie sind externe Zugriffe geregelt ?
- Welche Rolle spielt die interne IT ?
- Kann ein Teilnehmer einen Rechner und einen e-learningkurs bedienen ?
- Falls nein, wie soll/ kann er dazu befähigt werden ?
- In welcher Form erfolgt die Dokumentation ?
- Was geschieht mit den Ergebnissen möglicher integrierter Tests?
- Wie homogen sind die Rechner-Arbeitsplätze ausgestattet ?

6 Aufbau von Simulationen

Für den Aufbau von Simulationen stellen sich die folgenden Fragen:

- Wie erfolgt die Definition der Inhalte: generisch, unternehmensspezifisch ?
- Wie ist die Kosten / Nutzen-Situation ?
- Wie erfolgt der Zugriff auf die Simulationen ?
- Will/ kann ein Teilnehmer überhaupt in Simulationen trainieren ?
- Welches Interface ist hierfür geeignet ?

Was wäre das Optimum: nach einer theoretischen Unterweisung Durchführung des praktischen Trainings in einer immersiven 3D-Welt ohne spürbares Interface? Was bedeutet das ?

- Aufbau eines 3D-Modells des gewohnten Arbeitsumfeldes
- Immersion mittels VR/ AR -Technologien
- Natürliche Interaktion in 3D-Welten

Modell des Patientenzimmers.

7 3D-Modell Arbeits-Umfeld

Für ein virtuelles Training ist ein 3D-Modell des Arbeitsumfeldes erforderlich. Dieses kann mittels Architektur-Plänen, Vor-Ort Photoaufnahmen und CAD-Verfahren erstellt werden. Ein Import des Modells in eine Game-Engine ermöglicht dann den Aufbau eines serious games, mit immersivem Charakter.

Ein spezieller Focus sollte hierbei auf die sinnvolle Nachbildung aller brandschutzbezogenen Elemente gerichtet werden: Rauchmelder, Feuerlöscher, Fluchtschilder, etc.



Abbildung 2: Patientenzimmer, 3D-Modell

8 3D Geometriemodell

Im Projekt KATIE werden verschiedene Konzepte getestet: 3D-Geometriemodell und 2D/3D-Photomodell. Exemplarisch wird hier der Einsatz in einem Krankenhausumfeld betrachtet. Bild 2 zeigt das Photo eines Zimmers einer Krankenstation.



Abbildung 1: Beispiel: reales Patientenzimmer in Klinik

Basierend auf Architekturplänen wurde ein 3D Modell der Krankenstation mit Patientenzimmern erstellt und dieses mittels einer Game-Engine navigierbar gemacht. Es zeigt sich, dass ein solches Modell mittels CAD-Verfahren nur aufwändig erstellbar ist. Bei größeren Unternehmen kann es nur für wenige Mitarbeiter das persönliche Arbeitsumfeld abbilden.

Durch die Nutzung aller verfügbaren Technologien einer Game-Engine ergeben sich aber wesentliche Vorteile, wie verfügbare Navigation, Partikelfeuer, etc. gegenüber einer eigenen Programmierung. Bild 3 zeigt exemplarisch das 3D-

9 2D-Photomodell

Weniger Aufwand ist der Einsatz von Photographien oder Video-Aufnahmen. Diese sind visuell realer und geben die tatsächliche Umgebung wieder. Photographien können verschieden genutzt werden:

- Durchführung von Rundgängen
- Basis für Überlagerungstechniken
- Aufbau eines 3D-Photomodells

Bei Überlagerungstechniken werden 3D-Elemente über das Bild gelegt, so dass eine Visualisierung, von z.B. Bränden möglich ist. Bild 4 zeigt als Beispiel ein Photo mit überlagertem Partikelfeuer als Konzeptbild. Für Löschübungen ist dieses Konzept weniger geeignet.



Abbildung 3: Test: Foto Patientenzimmer mit Feuerball als Overlay

10 3D-Photomodell

Um die Vorzüge der Immersion mit der schnellen Erstellung zu kombinieren wurde in KATIE ein 3D-Photomodell entwickelt. Basierend auf Photoaufnahmen wird ein navigierbares 3D-Modell erstellt. Das Konzept ist angelehnt an das Quicktime VR-Konzept. Zum Einsatz kommen Kugelpanoramen, welche navigierbar verknüpft werden.

Das entstandene Modell wird mit 3D-Objekten verbunden, so dass eine Interaktion mit Gegenständen, wie z.B. Feuerlöschern, möglich wird.

Bild 5 zeigt exemplarisch zwei Aufnahmen, aus mehreren dieser Aufnahmen wird das 3D Photomodell zusammengesetzt.



Abbildung 5: Löscher entsichern

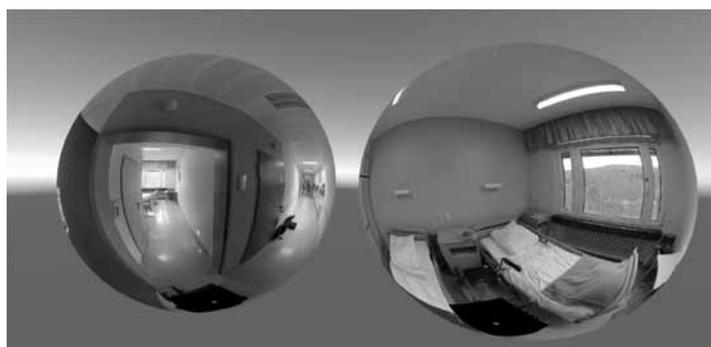


Abbildung 4: Aufnahmen 3D Photomodell

11 Löschen von Feuern

Für das praktische Training ist der Umgang mit Feuerlöschern und ggf. auch Wandhydranten wichtig. Hierbei können verschiedene Trainingsbereiche unterschieden werden:

- Elementares Bedienen des Löschers
- Erlernen einer Löschtaktik

Das elementare Bedienen umfasst die Nutzung der vor Ort befindlichen Löscher: Entsichern (siehe Bild 6) oder Löschen (siehe Bild 7).

Die Löschtaktik umfasst den richtigen Einsatz: u.a. Wahl des richtigen Löschertyps und das Vorgehen beim Löschen.

Hier ist die Integration von real zu bedienenden Feuerlöschern in die Simulation von Vorteil.

Im Rahmen des Projektes wurde ein Sensorlöscher entwickelt, mit dem die erforderlichen Trainingssequenzen absolviert werden können. Bild 8 zeigt einen Prototyp, welcher mit handelsüblichen Spielekonsolen-Sensoren aufgebaut wurde, Bild 9 die Weiterentwicklung mit eigenentwickelten Sensoren, welche sich im Innern des Löschers befinden.



Abbildung 6: Löscher einsetzen

Ein solcher Löscher kann nun in einer Simulation genutzt werden. Denkbar ist hier der Einsatz einer Großbildleinwand mit Rückprojektion. Bild 10 zeigt einen beispielhaften Einsatzfall der Simulationslösung der Firma Veis.

Das Training setzt eine sinnvolle algorithmische Umsetzung des Löschvorgangs voraus, u.a. die Bereitstellung der verschiedenen Löschmittel. Für viele Trainingssituationen ist der Einsatz von Partikelsystemen ausreichend. Der Löschvorgang wird bestmöglich visuell dargestellt. Bei anspruchsvolleren Ausbildungen, wie z.B. für berufliche Feuerwehrkräfte, ist eine physikalisch korrekte Simulation sinnvoll, diese kann u.a. durch die Einbindung von physikalischen Berechnungen auf Graphikkarten (CUDA) erreicht werden.

KATIE stellt typische Feuerlöschern zur Verfügung: Wasser-, Schaum- und Pulverlöscher.



Abbildung 7: Prototyp Sensorlöscher, COTS Sensoren, hier Wii

12 Training mittels e-learning

Der erste Schritt der Ausbildung ist die Vermittlung des erforderlichen Wissens. Hierfür nutzt KATIE professionelle Angebote, wie sie unter www.e-brandschutz.de zu finden sind, siehe auch [1]. Verfügbar sind Kurse für die Brandschutzunterweisung sowie zum Brandschutzhelfer.

13 Immersive Simulation

Einen großen Grad an Immersion bietet ein Head-Mounted Display. Diese sind schon seit den 1990iger Jahren bekannt. Durch die Nutzung leistungsstarker und kostengünstiger Smartphonekomponenten haben Unternehmen wie ehemals Oculus, nun Facebook, oder Sony und andere, einen neuen VR-Hype ausgelöst. Game-Engines erlauben zudem die leichte Integration der Anwendung in ein HMD.

Für den realen Einsatz bei einer größeren Zahl von Nutzern stellen sich aber andere Fragen:

- Wie erfolgt der Zugriff auf ein HMD?
- Gibt es hygienische Probleme ?
- Wie geht man mit den stereo-typischen Problemen bei



Abbildung 8: Prototyp Sensorlöscher, integrierte Sensoren, USB

größeren Gruppen um?

Ein HMD ist keine übliche Rechner-Komponente, wie Maus oder Tastatur und deshalb typischerweise nicht an jedem Rechner-Arbeitsplatz vorhanden. Eine erste Lösung bietet der Aufbau einer stand-alone Lösung, also die Bereitstellung eines HMD und eines Rechners. Nicht jeder Nutzer empfindet die stereoskopische Sicht als angenehm. Auch Kabelverbindungen stellen ein nicht zu unterschätzendes Problem dar, Stürze müssen unbedingt vermieden werden.

Erste Entwicklungen des neuen VR-Hypes starteten mit einem 3D-Druck eines tragbaren Gestells und der Nutzung handelsüblicher Smartphones (Projekt Opendive), wie Bild 11 zeigt.

Im Rahmen des Projektes KATIE wurde eine mobile Rucksackversion zum kabellosen Betrieb des Oculus-HMD entwickelt, wie Bild 12 zeigt.

Mittlerweile sind auch autarke HMD Lösungen verfügbar, die wieder mittels Smartphones arbeiten, wie z.B. Samsung Gear, wie Bild 13 zeigt. Der Umgang mit dieser Hardware ist für den Nutzer sehr einfach, es gibt keine Kabel.

Alternativ kann eine AR-Lösung genutzt werden, bei der ein Training in der realen Umgebung möglich wird. Bild 14 zeigt einen AR-Prototyp. Der abgebildete Mülleimer löst einen virtuellen Brand im Mülleimer aus.

Ein völlig anderes Training bietet der Einsatz einer Cave an, z.B. der CasCave. Hier können sich Trainer und Lerner gemeinsam in der Szene aufhalten. Einen Cave Einsatz zeigt Bild 15. Auch hier können reale Gegenstände sehr einfach in die Simulation integriert werden.



Abbildung 9: Einsatz Sensorlöcher, Quelle Fa. Veis



Abbildung 11: Kombination mobile Version Oculus und Sensor-Feuerlöscher



Abbildung 10: OpenDive

Denkbar sind auch die Kombination der verschiedenen Techniken, z.B. in einer Multiuser-Anwendung, bei der ein Teil der Teilnehmer HMDs tragen, PC und CAVE zu Einsatz kommen.

In Rahmen von KATIE wird als nächster Schritt für die Ausbildung zum Brandschutzhelfer der Praxistest mit den verschiedenen 3D- und VR-Konzepten in größeren Gruppen durchgeführt. Für die elementare Ausbildung sind Tests mit WebGL basierten Lösungen geplant.

13.1 Zusammenfassung

Die Kombination von e-learning Systemen mit graphischen Simulationen verspricht ein erfolgreiches Konzept im Rahmen der Brandschutzausbildung zu werden. Simulationen ermöglichen ein Training im bekannten Arbeitsumfeld. Wesentlich sind hier neben dem Erlernen des allgemeinen Vorgehens auch das Einüben des Löschvorganges, was in der Realität in einem Arbeitsumfeld nicht durchgeführt werden kann.

Der Einsatz von VR-Techniken ist aus Sicht des Trainingseffektes eine weitere Steigerung, der Einsatz mit vielen Teilnehmern ist jedoch mit erheblichen organisatorischen Herausforderungen verbunden, für die einfache Lösungen gefunden werden müssen.

14 Team Katie

Dieser Beitrag nutzt die Zuarbeit von vielen Mitgliedern des Katie-Teams, u.a. L. Li, F. Kreckler, S. Schiffner, N. Awadh. Mein Dank gilt diesen Personen.

Literatur

- [1] Dirk Ehrlich und Dieter Wloka: *Brandschutz - Sicherheit im Brandfall*; Crisis Prevention, Beta Verlag- und Marketinggesellschaft mbH, Bonn; (1), S. 37; 2013.
- [2] Dieter Wloka: *Graphische Simulation bei der Ausbildung in der Feuerwehr*; Crisis Prevention, Beta Verlag- und Marketinggesellschaft mbH, Bonn; (3), S. 13 – 16; 2012.
- [3] Dieter Wloka: *3D-Simulation für Feuerwehr*; TV-Beitrag, Ausstrahlung am 08.02.2013, RTL Hessen; Februar 2013.
- [4] Dieter Wloka: *Brände löschen in 3D*; Hessische Niedersächsische Zeitung, Ausgabe 14.01.2013; 2013.
- [5] Dieter Wloka: *Ersthelfer-Ausbildung durch graphische Simulation*; Crisis Prevention, Beta Verlag- und Marketinggesellschaft mbH, Bonn; (1), S. 34 – 36; 2013.
- [6] Dieter Wloka: *Feuerwehrleute üben am Computer-Simulator*; TV-Beitrag, Hessenschau HR; Januar 2013.
- [7] Dieter Wloka: *Virtuelles Training für die Feuerwehr*; TV-Beitrag, SAT 1; Februar 2013.



Abbildung 12: Kombination Samsung Gear und Feuerlöscher



Abbildung 13: AR Prototyp

- [8] Dieter Wloka: *Brandschutz-Ausbildung mit Serious Games und Virtueller Realität (Vortrag)*; 9. Virtual Fires Kongress, Virtual Dimension Center Technologiezentrum, St. Georgen,; Mai 2014.
- [9] Dieter Wloka: *Drohne als Löschhelfer*; *Hessische Niedersächsische Zeitung, Online-Ausgabe 03.02.2014*; 2014.
- [10] Dieter Wloka: *Was tun, wenn's brennt? E-Learning-Kurse zum Brandschutz entwickelt*; *Publik, Ausgabe 08.10.2014, Online-Ausgabe*; 2014.



Abbildung 14: Löschen in der CasCave

Blended Learning Methoden der Visualisierung, Demonstration, Selbstüberprüfung und Beurteilung in Mathematik, Modellbildung und Simulation

Andreas Körner¹, Stefanie Winkler¹, Martin Bicher¹, Felix Breitenecker¹

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

Visualisierung und Demonstration sind zwei Kernthemen welche in kaum einem anderen Gebiet so wichtig sind, wie in Modellbildung und Simulation. Blended Learning als gewinnbringende Kombination von klassischen Lehrkonzepten und eLearning Komponenten ist daher besonders geeignet für diese Fachdisziplin. In diesem Artikel werden elektronische Unterstützungen diskutiert und einige Anwendungsbeispiele gezeigt, wie in Mathematik und damit auch in mathematischer Modellbildung diese Hilfsmittel zum Einsatz gebracht werden. Des Weiteren werden Beispiele für Selbstüberprüfung und Prüfungen in Mathematik präsentiert und den Transfer in Fächer der Modellbildung und Simulation.

Ein Hauptaugenmerk wird hier auf Beispiele in MATLAB und Maple gelegt, welche besonders in akademischen Kreisen weit verbreitet sind aber auch in der Praxis zum Einsatz kommen. Ein wesentlicher Punkt, der in diesem Beitrag herausgearbeitet ist, behandelt den Unterschied zwischen Demonstration, Visualisierung und der Prüfungssituation. Diese Bereiche haben unterschiedliche Anforderungen, können voneinander aber nicht getrennt betrachtet werden.

1 Einleitung und Aufbau

In der Forschungsgruppe Mathematische Modellbildung und Simulation an der TU Wien werden seit vielen Jahren elektronische Tools zur Unterstützung der Lehre eingesetzt.

Angefangen hat es mit dem allseits bekannten Moodle. Dieses steht an der TU Wien allen Lehrenden zur Verfügung. Es wird meist verwendet. Um die Lehrinhalte zu organisieren und für die Studierenden sinnvoll aufzubereiten.

In der Forschungsgruppe wird es vor allem für die Durchführung der Mathematik Übung sowie auch für die Vorlesungen zu mathematische Modellbildung und Simulation verwendet. In den Mathematiklehreveranstaltungen eignet es sich vor allem für den Übungsablauf. Lehrende stellen die zu rechnenden Beispiele online zu Verfügung. Diese Studierende haben daraufhin eine bestimmte Zeit diese Aufgaben zu Lösen. Anschließend können die Studierende ihre ausgearbeiteten Beispiele in einem zusätzlichen Plug-In von Moodle ankreuzen um dem Lehrenden seine Hausübungsbeispiele mitzuteilen.

In den Simulationslehreveranstaltungen wird es weniger als aktives Tool sondern als reine Vorlesungsunterlagen Sammlung genutzt.

Diese einseitige Nutzung von Softwareunterstützung ist aber schon lange nicht mehr up to date. In den Simulationslehreveranstaltungen wird zusätzlich zu Moodle seit 2006 eine weitere Plattform namens MMT (Mathematics Modelling and Tools) eingesetzt. Diese beschäftigt sich vor allem mit der Realisierung von Simulationsmodellen und deren Resultate für unterschiedliche Parameter.

Dieses Onlinesystem wird in der Vorlesung eingesetzt, um die mathematischen Grundlagen der Modellbildung auch mit Hilfe von Vorimplementierten Beispielen zu verstehen.

In den Mathematiklehreveranstaltungen hat es sich allerdings herausgestellt, dass die Studierenden mehr als nur die schriftlichen Übungsbeispiele benötigen um sich die notwendigen Fertigkeiten anzulernen. Seit 2008 wird dort ein Tool eingesetzt, dass auf dem Computer Algebra System Maple basiert und wird daher Maple T.A. (Teaching and Assessment) ge-

nannt. Dieses Tool wird weniger für Visualisierung und Verständnis eingesetzt als für das Trainieren der Fertigkeiten.

2 Online System im Vergleich

Maple T.A.

Wie schon erwähnt wird dieses Tool bisher nur in den Mathematiklehreveranstaltungen eingesetzt. Es eignet sich besonders, den Studierenden individuell an ihren Schwächen und Lücken zu arbeiten.

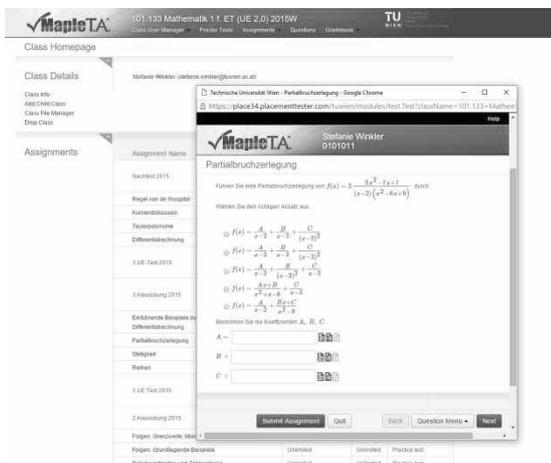


Abbildung 1. Zeigt die Struktur des online Systems Maple T.A.

Wie in Abbildung 1 gezeigt haben Studierende die Möglichkeit, Übungsbeispiele zu den einzelnen, in der Vorlesung besprochenen Themen, zu Üben. Jedes Thema verfügt über 4 bis 20 Beispiele. Diese sind nicht starr programmiert sondern werden bei jedem Aufruf neu generiert. Diese Vorteile werden dann auch für Klausuren und Prüfungen genutzt. Das neu generieren der Zahlen ermöglicht eine Szenario der Vorlesungsprüfung ohne vom Nachbarn abschauen zu Können. Das System hat sich mittlerweile sehr gut bewährt um Fertigkeiten zu trainieren und diese dann in verschiedenen Testsituationen abzufragen.

MMT.

Abbildung 2 zeigt die Plattform MMT. In der linken Spalte ist der Beispielbaum für die jeweilige Lehrveranstaltung abgebildet. In der Mitte befindet sich das Beispiele mit zugehöriger Beschreibung. In der rechten Spalte haben Studierende die Möglichkeit, den zugehörige Programmcode, welcher in MATLAB, OKTAVE, SIMULINK oder ähnlichem implementiert wurde, herunterzuladen und selbst zu testen.

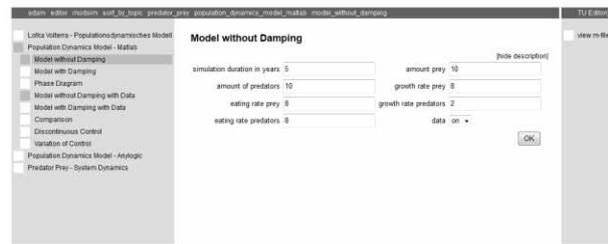


Abbildung 2. Zeigt ein Beispiel im MMT online Tool.

Dieses Tool dient vor allem der Visualisierung und Demonstration. Studierende haben die Möglichkeit für verschiedene Parameter der zugrundeliegenden Gleichung unterschiedliche Werte einzusetzen um so die Wirkung auf das Modell zu untersuchen.

Auf diese Art und Weise können aber nicht nur Differentialgleichungsmodelle sondern auch Modelle ganz anderer Struktur demonstriert werden.

Bisher wurden auch Tests durchgeführt bei denen Moodle und MMT zusammen verwendet wurde, um das Verständnis der Modelle abzufragen.

3 Ausblick

Da sich leider die Verbindung von Moodle und MMT nicht als zufriedenstellendes Testmodul herausgestellt hat, wird im Moment daran gearbeitet, die Vorteile aller Online Systeme bestmöglich zu verbinden. Durch die Weiterentwicklung des Maple T.A. ist es nun auch möglich Applikationen einzubinden, die ein Abbild des MMT Tools bilden könnten. Auf diese Art könnte man die Beurteilung der Modellbildungskennnisse noch etwas verbessern.

4 References

- [1] A. Zimmermann, V. Urbonaite, A. Körner, S. Winkler, S. Krause, M Kleinert. *Advanced Randomization and Grading in the E-Learning System Maple T.A..* Proceedings of the 7th Congress on Modelling and Simulation. ISBN: 978-80-01-04589-3; S. 1209 - 1214. 2010.
- [2] S. Winkler, A. Körner, V. Urbonaite: *Maple T.A. in Engineering Educations!* In: Preprints Mathmod 2012 Vienna - Full Paper Volume", F. Breitenecker, I. Troch (Hrg.); Argesim / Asim, 38 (2012), S. 258 - 259.E. Beispiel und S. Exempel. *Titel der V. Zeitschrift*, S. 111–123, 2012.

Simulation von Smart Fusing-Konzepten für die dezentrale Leitungsabsicherung in zukünftigen Kfz-Bordnetzen

Dipl.-Ing. Thomas Reinders
Delphi Deutschland GmbH
thomas.reinders@delphi.com

Kfz-Bordnetze sind in den vergangenen Jahren aufgrund der stetig steigenden Anzahl von Sicherheits-, Komfort- und Kommunikationsfunktionen zunehmend komplexer geworden. Bereits heute stößt der aktuelle Ansatz der hierarchischen Energieverteilung an seine Grenzen hinsichtlich Gewicht und Bauraumbedarf der elektromechanischen Stromverteiler und des Leitungssatzes, Verlustleistung, Spannungsstabilität sowie Flexibilität und Erweiterbarkeit. Die zukünftig weiterhin steigenden Anforderungen im Hinblick auf Robustheit und Verfügbarkeit des Bordnetzes für das autonome Fahren (Fail-Operational-Systeme) werden daher völlig neue Ansätze der E/E-Architektur erfordern.

Bei der Energieverteilung wird voraussichtlich eine Dezentralisierung zu verzeichnen sein, wobei intelligente Stromverteiler basierend auf Halbleiterschaltern mit reversibler Sicherungsfunktion eine wichtige Rolle spielen werden. Neben einer Diagnose des Bordnetz-Zustands wird hierdurch eine höhere Energieeffizienz durch geringere Verlustleistung und besseres Energiemanagement erwartet. Bezüglich des Leitungssatzes wird sich zudem eine Gewichtsverminderung durch kürzere Leitungslängen und durch Querschnittsreduzierung aufgrund einer besseren Anpassung der Auslösecharakteristik sowie geringerer Toleranzen der elektronischen Absicherung im Vergleich zu herkömmlichen Schmelzsicherungen ergeben. Zwar sind Halbleiterschalter mit Sicherungsfunktion (Smart FETs) bereits seit vielen Jahren im Kfz weit verbreitet. Für die Ausschöpfung des gesamten Einsparpotenzials zur CO₂-Verminderung sowie für eine Verbesserung des Leitungsschutzes sind allerdings neuartige Ansätze notwendig, die den Fokus nicht nur auf den Selbstschutz der Halbleiter richten. Hierbei können in Zukunft in den intelligenten Stromverteilern implementierte Softwarealgorithmen zum Tragen kommen, die kontinuierlich den Zustand der Leitung überwachen.

Im Rahmen dieses Vortrags werden Algorithmen diskutiert, mit denen die Leitungstemperatur beispielsweise auf Basis der Stromhistorie abgeschätzt und ggf. der Strompfad bei kritischen Situationen abgeschaltet werden kann. Zur Untersuchung der Tauglichkeit dieser Algorithmen zur elektronischen Leitungsabsicherung und Überprüfung des Einflusses verschiedener Betriebs- und Umgebungsbedingungen werden entsprechende Matlab/Simulink-Simulationsmodelle vorgestellt und die Ergebnisse mit einem durch Messungen validierten speziellen Berechnungsprogramm für das elektrothermische Verhalten von Leitungen verglichen.

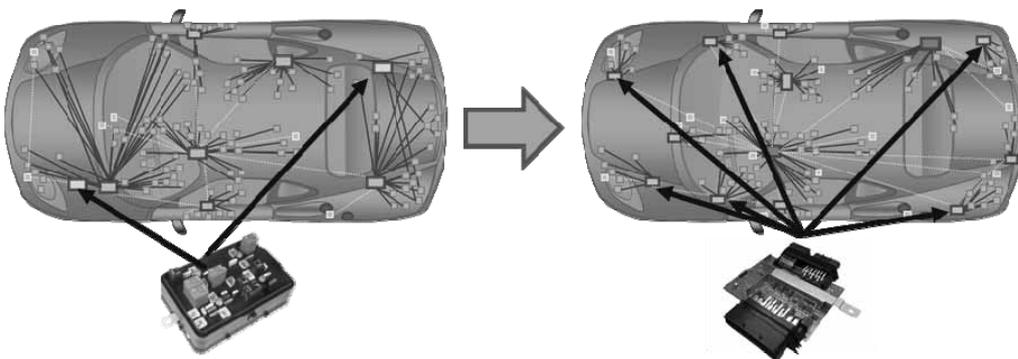


Abbildung 1. Heutige und zukünftige Bordnetz-Architektur und Leitungsabsicherung

Modellierung digitaler Schaltungsanteile in gemischten analog/digital Schaltungen für die analoge Simulation mittels Verilog-A

Michael Karagounis¹

¹Hochschule Hamm-Lippstadt, Hamm

Michael.Karagounis@hshl.de

In zunehmendem Maße werden klassische analoge Schaltungsblöcke in integrierten monolithischen Systemen durch gemischte analog/digital Architekturen ersetzt. Dadurch ist es möglich die höhere Integrationsdichte digitaler Schaltungen in modernen CMOS Prozessen zu nutzen und zusätzliche Konfigurationsmöglichkeiten für den Ausgleich von Technologie- und Prozessschwankungen durch gezielte Eichverfahren zu implementieren. Auf Grund der hohen Komplexität im digitalen Schaltungsteil erreicht die Simulationszeit gemischter analog/digital Schaltungen auf Transistorebene jedoch unzumutbar hohe Werte, was die Extraktion analoger Kenngrößen, die für den Schaltungsentwurf relevant sind, aus der analogen Simulation erschwert. An Hand praktischer Beispiele soll eine wenig beachtete Simulationsmethodik vorgestellt werden, die es ermöglicht die Simulationszeit zu verkürzen ohne auf die Genauigkeit der analogen Simulationsergebnisse verzichten zu müssen. Hierbei werden die digitalen Schaltungsanteile in der Hardwarebeschreibungssprache Verilog-A modelliert, die von vielen Analogsimulatoren für den Entwurf integrierter Schaltungen unterstützt wird. Es wird gezeigt, wie Verilog-A Sprachkonstrukte, die üblicherweise für die Erstellung vereinfachter Modelle von analogen Schaltungen verwendet werden, auch für die Modellierung digitaler Schaltungen genutzt werden können.

VDA FAT Arbeitskreis „Entwurfsmethodik und Simulation elektrischer und mechatronischer Systeme im Kfz

Thomas Lang¹

¹Robert Bosch GmbH

Thomas.Lang3@de.bosch.com

Im ersten Teil des Vortrags wird ein Überblick zu den Aktivitäten des Arbeitskreises anhand von Beispielen aus Bordnetzsimulation im Fahrzeug gegeben.

Im zweiten Teil werden Anregungen gegeben wie die Simulation aktuelle Fragestellungen in der Gestaltung der E/E-Architektur methodisch unterstützen kann.

Simulation eines Realfahrzyklus mit einer Lithium-Schwefel Batterie unter Berücksichtigung des "Shuttle-Effekts"

Christian Maurer¹, Walter Commerell¹, Claus Kröger¹, Ulrich Schrade¹

¹University of Applied Science Ulm

c.maurer@hs-ulm.de

Lithium-Schwefel Batterien sind aufgrund ihrer hohen spezifischen Energiedichte von ca. 2600 Wh/kg ein hochinteressanter Energiespeicher für Elektrofahrzeuge. In diesem Beitrag wird die Lithium-Schwefel Batterie vorgestellt. Bedingt durch den mehrstufigen Reaktionsprozess führt der "Shuttle-Effekt" zur Selbstentladung der Batterie. Für die Verwendung dieser Technologie in Elektrofahrzeugen, muss eine möglichst geringe Selbstentladung gewährleistet werden. Der weitere Fokus des Beitrags liegt auf dem Einfluss des "Shuttle-Effekts" auf einen Realfahrzyklus. Dieser Einfluss wurde simulativ untersucht.

1 Einleitung

Lithium Schwefel (Li-S) Batterien sind aufgrund ihrer theoretischen spezifischen Speicherkapazität (ca. 1675 Ah/kg), ihrer ca. fünf mal größeren theoretischen spezifischen Energiedichte (ca. 2600 Wh/kg) gegenüber Li-Ionen Batterien und der zu erwartenden geringeren Produktionskosten hochinteressant [1]. Zur Bewertung und Simulation von Li-S Batterien für Anwendungen im Automobil wird ein dynamisches Modell besonders für die Entladung benötigt. Entsprechende Modelle wurden in der Literatur untersucht [2] [3]. Bisher wurden Fahrzyklen mit dieser Batterietechnologie kaum simuliert. Dieser Beitrag berücksichtigt und modelliert die Selbstentladung durch den Shuttle-Effekt.

1.1 Li-S Batterietechnologie

Li-S Batterien besitzen eine Schwefel-Kathode und eine Lithium-Anode. Als leitfähiges Material wird zumeist ein flüssiges Elektrolyt verwendet. Die elektrischen Eigenschaften werden durch Additive beeinflusst. Die prinzipielle Redoxreaktion einer Li-S Zelle ist:

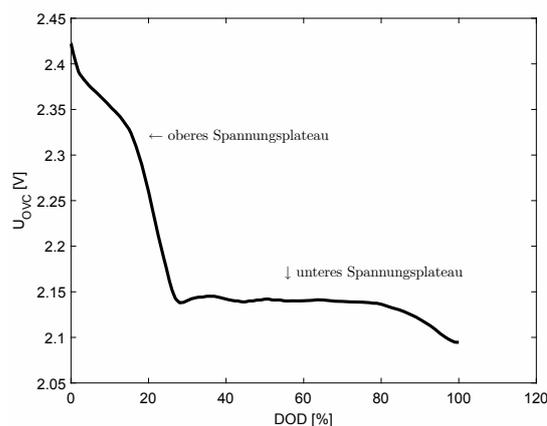


Abbildung 1: Leerlaufspannung einer Li-S Zelle mit den beiden Spannungsplateaus über dem Entladegrad (DOD).

Das Reaktionsprodukt wird durch mehrere Zwischenreaktionen erreicht. Es entstehen Polysulfidketten, die im Elektrolyt gelöst sind. $x = 1 \dots 8$ Schwefelatome sind pro Polysulfidkettenmolekül (Li_2S_x) verbunden [4]. Aufgrund der Zwischenreaktionen ist die Modellierung der Zelle schwieriger, als bei Li-Ionen Zellen. Ein entsprechendes mathematisches Modell wurde entwickelt [5].

Durch die Reduktion von langkettigen Polysulfide ($\text{Li}_2\text{S}_x, 4 \leq x \leq 8$) mit Lithium und die Oxidation

von kurzketigen Polysulfiden ($Li_2S_x, 1 \leq x \leq 3$) entlädt sich die Zelle selbst (Shuttle-Effekt). Dadurch beträgt die Zellentladung mehr als 10% der Zellkapazität pro Tag. Durch die Zugabe von $LiNO$ und $N - O$ Additiven kann die Selbstentladung durch den Shuttle-Effekt auf ca. 4-6% pro Monat reduziert werden [6]. In Folge der Zwischenreaktionen ergibt sich, durch die Überlagerung der Spannungspotentiale aus den Nernst-Gleichungen, eine Zellspannung mit zwei Spannungsplateaus (siehe Abbildung 1). Im zweiten (unteren) Spannungsplateau sind hauptsächlich Polysulfidketten mit weniger als 4 Schwefelatomen pro Molekül vorhanden [7]. Besonders die $Li_2S_{(2)}$ -Produkte sind im Elektrolyt der Zelle schwer löslich. Die Redoxreaktionen ergeben eine nominale Spannung von 2,1V. Bei gleicher Leistung und ohne Massenormierung der Zellen, wird eine Li-S Zelle mit dem 1,7 fachen Strom belastet, gegenüber einer Lithium-Ionen Zelle mit einer nominalen Spannung von 3,6V.

Die ersten Li-S Zellen sind seit wenigen Jahren kommerziell verfügbar [8]. Diese Forschungszellen haben für die Verwendung im Fahrzeug folgende Potentiale [1] [3] [6] [9]. Erstens, die spezifische Energiedichte ist mit ca. 300-400 Wh/kg weit von den theoretischen Werten entfernt. Zweitens, diese Zellen erreichen ca. 100-200 Zyklen bis sie nur noch 80% ihrer ursprüngliche Kapazität entladen können. Die Dichten von S_8 und Li_2S unterscheiden sich um den Faktor 1,79. Dies ist prinzipiell eine Ursache für den Kapazitätsverlust. Drittens, die Innenwiderstände sind im Vergleich mit Lithium-Ionen Zellen wesentlich höher [8]. Viertens, die entnehmbare Ladung der Zelle ist stark von der Temperatur und der Entladestromstärke abhängig.

2 Simulationsmodell

Für eine performante Berechnung von Simulationen und reale Algorithmen sind lineare Modelle notwendig.

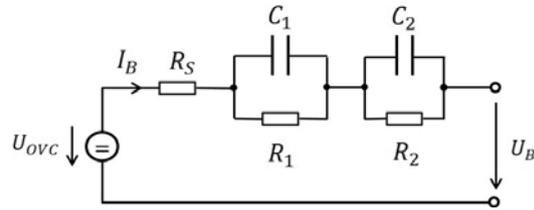


Abbildung 2: Elektrisches Ersatzschaltbild für eine Batteriemodell mit 2 RC-Netzen.

2.1 Li-S Batteriemodell

Batterien unterschiedlicher Technologien werden oftmals mithilfe eines elektrischen Ersatzschaltbildes, mit veränderlichen Parametern, modelliert. Ein allgemeines und parametrierbares Simulinkmodell mit zwei RC-Netzen ist in [10] beschrieben. Ein entsprechendes Ersatzschaltbild ist in Abbildung 2 dargestellt.

Nach [11] entspricht R_S dem Elektrolytwiderstand, R_1 dem Gesamtwiderstand der Oberflächenschicht der Schwefel- und der Lithiumelektrode, C_1 der verteilten Kapazität der Oberflächenschicht beider Elektroden, R_2 dem Ladungstransferwiderstand der Schwefelelektrode und C_2 der Doppelschichtkapazität der Poren der Schwefelelektrode. Die Parameter werden in "Lookuptables" abgelegt. Das in diesem Beitrag verwendete Simulationsmodell ist an [3] und [10] angelehnt und wurde mit [2] und [8] grundsätzlich validiert. Damit lassen sich aktuelle Fahrzustände und Fahrsituationen simulieren. Das Modell berücksichtigt keine Effekte der Zellalterung.

Der Shuttle-Effekt ist stark temperaturabhängig. Aus diesem Grund wurde ein Temperaturmodell 1. Ordnung in das Batteriemodell eingebunden:

$$\frac{dT}{dt} = \frac{1}{m_Z C_{th}} \cdot (P_V + \alpha(T - T_U)) \quad (2)$$

Dabei ist P_V die Zellverlustleistung, T die Zelltemperatur, m_Z die Zellmasse, C_{th} die Wärmekapazität, α der Wärmeleitkoeffizient und T_U die Umgebungstemperatur.

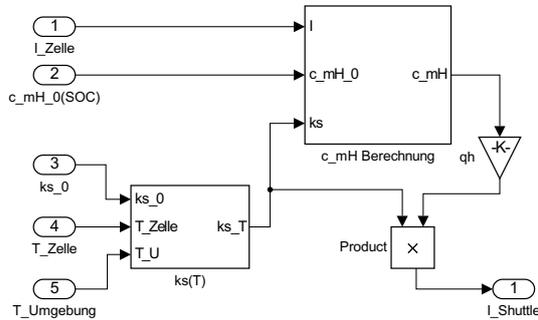


Abbildung 3: Simulink Modell des Shuttle-Effekts.

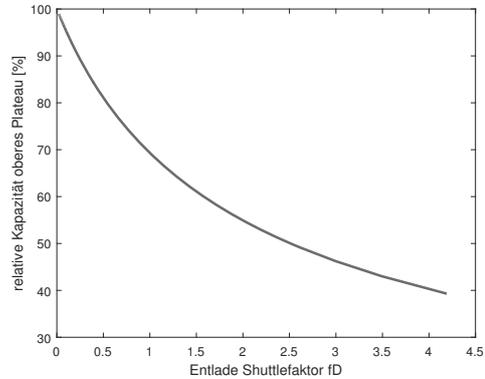


Abbildung 4: Relative Entladekapazität der Kapazität des oberen Spannungsplateaus abhängig vom Entladeshuttlefaktor.

2.2 Shuttle-Effekt

Da der Shuttle-Effekt aus der Reduktion langkettiger Polysulfide und der Oxidation kurzkettiger Polysulfide besteht, müssen langkettige Polysulfide dafür vorhanden sein. Im Verlauf der Zellentladung werden höhere Konzentrationen kurzkettiger Polysulfide gebildet. Langkettige Polysulfide sind nur im oberen Spannungsplateau vorhanden. Die Kapazität dieses Spannungsplateaus enthält vereinfacht ein Drittel der Zellkapazität. Das untere Spannungsplateau entsprechend zwei Drittel [7] [5]. Daraus lässt sich die spezifische Kapazität des oberen Spannungsplateaus vereinfacht ableiten.

Die folgende Modellbildung des Entladeprozesses wurde quantitativ in [7] auf Basis der folgenden Differentialgleichung für das obere Spannungsplateau durchgeführt:

$$\frac{dc_{mH}}{dt} = \frac{i}{q_H} - k_S \cdot c_{mH} \quad (3)$$

Dabei ist c_{mH} die massenormierte (bezüglich der Schwefelmasse in der Zelle) Konzentration langkettiger Polysulfide im oberen Spannungsplateau. i ist der massenormierte Lade- oder Entladestrom. q_H ist die massenormierte Kapazität des oberen Spannungsplateaus und k_S ist der chemische Reaktionsgeschwindigkeitskoeffizient. Aufgrund der Begrenzung des Shuttle-Effekts auf das obere Spannungsplateau darf dieser nur dafür modelliert werden. Ein Simulinkmodell des Shuttle-Effekts ist in Abbildung 3 dargestellt. Durch die Entladung der Li-S Zelle, sinkt die Konzentration langkettiger Polysulfide und der Shuttle-Effekt verringert sich. Die komplette Kapazität des oberen Plateaus kann nur bei ho-

hen Strömen entnommen werden. Zur Bewertung des Shuttle-Effekts wurde der sogenannte Entladeshuttlefaktor eingeführt [7].

$$f_D = \frac{k_S \cdot c_{mH}^0 \cdot q_H}{i_D} \quad (4)$$

f_D ist der Entladeshuttlefaktor. i_D ist der massenormierte Entladestrom. c_{mH}^0 ist die, zum Beginn eines Entladevorganges, massenormierte Anfangskonzentration langkettiger Polysulfide des oberen Spannungsplateaus. Der Entladeshuttlefaktor ist eine Kenngröße. Sie gibt an, wie stark das obere Spannungsplateau durch die Selbstentladung entladen wird. Der Faktor bezieht sich auf die Konstantstromentladung der gesamten zum betrachteten Zeitpunkt t vorhandene Kapazität des oberen Spannungsplateaus. Der Einfluss des Entladeshuttlefaktors auf die entnehmbare Kapazität ist in Abbildung 4 dargestellt. Beim Laden einer Zelle mit kleinen Strömen kann der Shuttlestrom und der Ladestrom im Gleichgewicht sein. Dieser Zustand führt zu einer hohen Ladungsaufnahme der Zelle, zu einer nie vollständig geladenen Zelle und zur Erwärmung der Zelle. Der massenormierte Shuttlestrom wird aus der aktuellen Konzentration langkettiger Polysulfide im oberen Spannungsplateau berechnet:

$$i_S = k_S \cdot q_H \cdot c_{mH} \quad (5)$$

Aus den Gleichungen (5), der Zellspannung und der

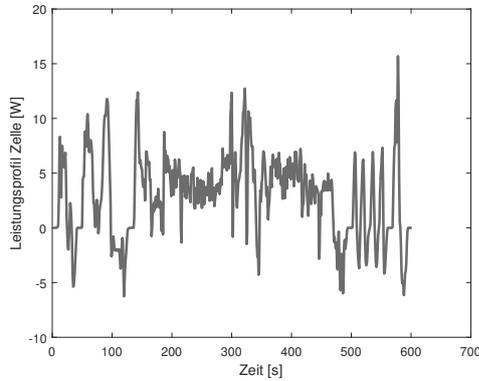


Abbildung 5: Leistungsprofil einer Li-S Zelle unter Verwendung eines 800kg Fahrzeug inklusive 80kg Batteriespeichers und dem US06 Fahrzyklus.

Gleichung (2) kann die Erwärmung der Zelle durch den Shuttlestrom berechnet werden.

k_S hängt stark vom Elektrolyt und von der Zelltemperatur ab. Die Temperaturabhängigkeit wird als zunehmende Diffusion der Polysulfide im Elektrolyt interpretiert. Dadurch beschleunigt oder reduziert sich der Shuttle-Effekt [7]. Eine Temperaturabhängigkeit des Shuttle-Effekts ist durch die Arrheniusgleichung (6) gegeben. Dieser Einfluss wurde ebenfalls modelliert.

$$k_S(T) = k_S(T_0) e^{\left(-\frac{A}{R} \cdot \left(\frac{1}{T} - \frac{1}{T_0}\right)\right)} \quad (6)$$

Dabei ist A die Shuttle-Aktivierungsenergie und R die universelle Gaskonstante. In der Literatur wurde zusätzlich eine Abhängigkeit des Shuttle-Effekts von der Stromstärke nachgewiesen [9].

2.3 Fahrzyklus und Fahrzeug

Für die Simulation ist ein Fahrzyklus notwendig. Unterschiedliche Fahrzeuge und Fahrzyklen wurden in [12] untersucht. Darauf basierend wurde in dieser Arbeit der Realfahrzyklus US06 verwendet. Aufgrund der hohen Beschleunigungs- und Rekuperationsleistungsanteile eignet sich dieser Fahrzyklus besonders gut zur Evaluierung von Batterietechnologien. Es wurde bei der Berechnung des Leistungsprofils (Gleichung (7)) lediglich die Längsdynamik berücksichtigt. Dazu wurde ein Kleinwagen mit den rel-

evanten Parametern und der Längsdynamik aus [12] verwendet. Die Fahrzeugleistung P_{FZ} berechnet sich aus dem Geschwindigkeitsprofil v_{FZ} und der Kraft der Längsdynamik F_{LD} . F_{LD} berechnet sich aus den kumulierten Kräften des Rollwiderstandes F_R , des Luftwiderstandes F_L , des Steigungswiderstandes F_{St} und des Beschleunigungswiderstandes F_B .

$$P_{FZ} = F_{LD} v_{FZ} = (F_R + F_L + F_{St} + F_B) v_{FZ} \quad (7)$$

$$F_R = \mu \cdot m_{FZ} \cdot g \cdot \cos(\alpha)$$

$$F_L = \frac{1}{2} \rho_L \cdot c_w \cdot A \cdot v_{FZ}^2 \quad (8)$$

$$F_{St} = m_{FZ} \cdot g \cdot \sin(\alpha)$$

$$F_B = e \cdot m_{FZ} \cdot a$$

In den Gleichungen 8 ist μ der Rollwiderstandskoeffizient, m_{FZ} die Fahrzeugmasse, g die Erdbeschleunigung, α der Steigungswinkel, ρ_L die Luftdichte, c_w der Strömungswiderstandskoeffizient, A die Fahrzeugfrontfläche, e der Fahrzeugmassenfaktor und a die Beschleunigung des Fahrzeugs.

Zur Berechnung des Zelleleistungsprofils, wird der Wirkungsgrad des Antriebsstrangs konstant berücksichtigt:

$$P_{BatBeschleunigung} = \frac{P_{FZ}}{\eta} \quad (9)$$

$$P_{BatRekuperation} = -P_{FZ} \cdot \eta \quad (10)$$

Die Leistung für die Zellentladung wurde positiv gewählt.

Daraus wurde die Leistung einer Batteriezelle nach Gleichung (11) berechnet.

$$P_{Zelle} = \frac{P_{Bat}}{N} \quad (11)$$

Eine Vergrößerung der Masse des Batteriespeichers hat eine geringere Zellbelastung und damit einen geringeren Zellstrom zur Folge. Der Einfluss von k_S auf die Selbstentladung ist aufgrund seiner Temperaturabhängigkeit wesentlich größer, als die Masseänderung des Batteriespeichers. Aus diesem Grund wurde die Simulation für eine konstante Batteriemasse, die in m_{FZ} enthalten ist, durchgeführt.

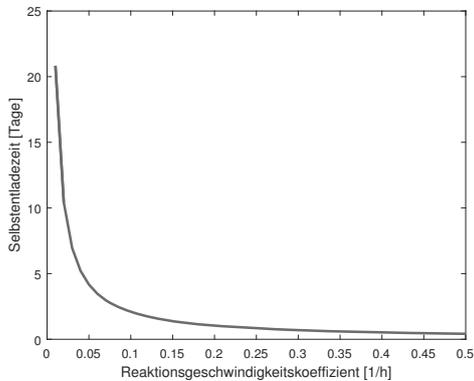


Abbildung 6: Vollständige Entladezeit der Kapazität des Oberen Spannungsplateaus abhängig von k_S .

Der Batteriestrom wurde mittels des parametrisierten Batteriemodells (Kapitel 2.1) berechnet. In Abbildung 5 ist das Leistungsprofil des US06 Fahrzyklus dargestellt.

3 Simulationsergebnisse

Zur Modellvalidierung wurde die Entladezeit von verschiedenen Reaktionsgeschwindigkeitskoeffizienten k_S simuliert und mit gemessenen Werten aus der Literatur verglichen. In Abbildung 6 ist die Entladezeit des oberen Spannungsplateaus in Abhängigkeit von k_S dargestellt. Eine gute Übereinstimmung des simulierten und beobachteten Shuttle-Effekts wurde für $k_S = [0,1 \quad 0,19 \quad 0,53]$ gezeigt [7]. Dadurch bestätigt sich, dass die Annahme der Aufteilung der Zellkapazität in ein oberes und ein unteres Spannungsplateau und die entsprechende Zuordnung der langkettigen und kurzketigen Polysulfide zu den Spannungsplateaus (Kapitel 1.1) für die Modellgenauigkeit des Shuttle-Effekts ausreichend ist.

Abbildung 6 zeigt zusätzlich, dass für $k_S > 0,05 \frac{1}{h}$ das obere Spannungsplateau spätestens nach 4 Tagen entladen ist. Dieser Koeffizient muss für Elektrofahrzeuge kleiner sein. Der Faktor k_S bestimmt die Geschwindigkeit der Entladung. Für beliebige Zellen lässt sich die Entladung durch den Shuttlefaktor (Gleichung (5)) normalisiert darstellen. Unabhängig von der verwendeten Zelle wird bei gleichem Entladeshuttlefaktor die gleiche relative Kapazität aus dem oberen

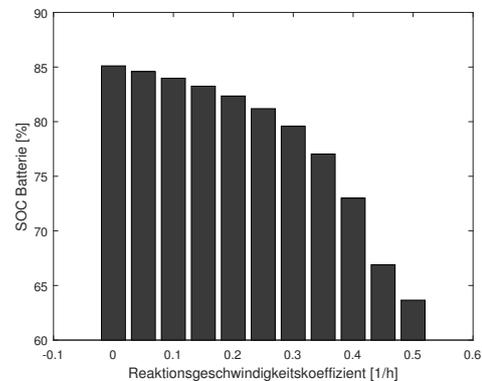


Abbildung 7: SOC einer Li-S Zelle nach der Simulation des doppelten US06 Fahrzyklus gegenüber k_S . Für die Simulation wurde Ein 80kg Batteriespeicher mit $SOC_{t_0} = 100\%$ und ein 800kg Fahrzeug (Gesamtmasse) verwendet.

Spannungsplateau entnommen. Abbildung 4 zeigt die relative Kapazitätsentnahme aus dem oberen Spannungsplateaus bei Konstantstromentladung mit dem entsprechenden Entladeshuttlefaktor. Durch die Normalisierung lässt sich der Einfluss der Selbstentladung in einer Zahl darstellen.

Für die Bewertung des Shuttle-Effekts unter einem realen Fahrzyklus wurde k_S in einem Bereich von 0,01 bis $0,5 \frac{1}{h}$ variiert. In der Literatur gibt es Ergebnisse für Konstantstromentladungen für diesen Range von k_S [7].

In Abbildung 7 ist der "State of Charge" (SOC) einer Li-S Zelle nach der Simulation des doppelten Realfahrzyklus dargestellt. Der SOC entspricht dem Ladezustand der Zelle nach dem doppelten Fahrzyklus. Abhängig vom k_S ergibt sich ein Shuttlestrom der die Zelle über die Zeit des Fahrzyklus zusätzlich entlädt. Der Entladeshuttlefaktor steigt mit der Zunahme von k_S an. Die Höhe des äquivalenten Entladeshuttlefaktors kann aus Abbildung 4 in Kombination mit Abbildung 7 entnommen werden.

Mit der Simulation in Abbildung 7 wurde gezeigt, dass ein $k_S < 0,05$ eine Verringerung von weniger als einem Prozent des gesamten SOC, für die Simulation des doppelten Fahrzyklus, ausmacht. Das entspricht einer simulierten Strecke von ca. 26 km. Der Fahrzyklus selbst benötigt ca. 15% der Zellkapazität.

In Abbildung 8 ist die Zelltemperatur über den Verlauf des US06 Fahrzyklus exemplarisch für 3 k_S Werte

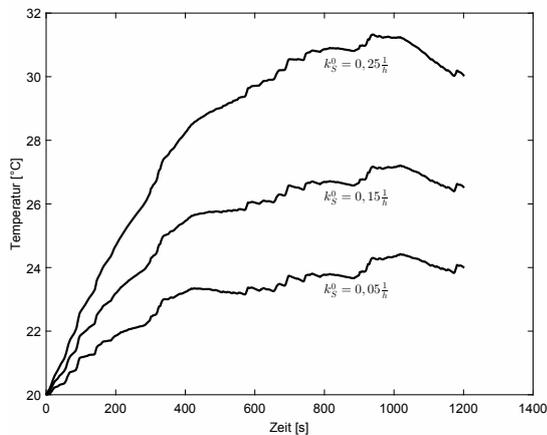


Abbildung 8: Zelltemperatur für k_S von 0.05, 0.15 und 0.25.

dargestellt. Die Selbstentladung für hohe k_S Werte hat eine signifikante Verlustleistung und damit eine Erhöhung der Zelltemperatur zur Folge. Aufgrund der Li-S Zellcharakteristik sinkt die Zellspannung besonders während des oberen Spannungsplateaus. Dadurch steigt bei gleicher Leistungsanforderung der Zellstrom. Durch die doppelte Simulation des US06 Fahrzyklus ist der entnommene Strom im zweiten Zyklus größer. Dadurch steigt auch die Zelltemperatur weiter an. Dieser Effekt ist besonders an $k_S^0 = 0,05$ ersichtlich. Dieser lässt nach dem ersten Fahrzyklus ($t = 600s$) einen stationären Zustand erkennen. In Abbildung 9 ist für die selben k_S Werte der relative Verlauf bezüglich ihrer Anfangswerte (k_S^0) dargestellt. Besonders die Temperaturabhängigkeit verstärkt den Shuttle-Effekt (siehe Abbildung 8 und 9). Diese Abhängigkeit ist ein weiterer Grund für einen möglichst niedrigen k_S Wert.

4 Zusammenfassung

In dem vorliegenden Beitrag wurde die Li-S Zelle Grundsätzlich vorgestellt. Dabei wurde besonders der Shuttle-Effekt betrachtet. Die Li-S Zelle und der Shuttle-Effekt wurden modelliert und die Auswirkung des Shuttle-Effekts unter Berücksichtigung des Fahrzyklus US06 simuliert. Für die Verwendung der Li-S Zelle ist es entscheidend, dass der Reaktionsgeschwindigkeitskoeffizient einen niedrigen Wert von $k_S^0 < 0,05$ hat. Dadurch ist die

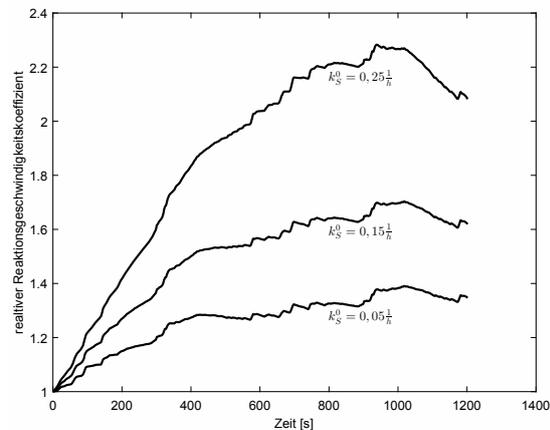


Abbildung 9: Relativer Reaktionsgeschwindigkeitskoeffizient k_S . Über dem Verlauf des doppelten US06 Fahrzyklus.

Verringerung des SOC $< 1\%$ bei doppeltem US06 Fahrzyklus. Die Selbstentladung bei diesem k_S^0 ist mit 4 Tagen immer noch groß. Der Shuttle-Effekt muss bei der Simulation eines realen Fahrzyklus berücksichtigt werden, wenn k_S in dieser Größenordnung liegt. Im Hinblick auf die Simulation langer Zeiträume kann das Modell ebenfalls Anwendung finden. Neben einem grundsätzlichen niedrigen Reaktionsgeschwindigkeitskoeffizienten ist die Temperaturabhängigkeit des Koeffizienten relevant. Dieser vergrößert die Selbstentladung. Weitere Verfeinerungen des Modells einer Li-S Zelle sollten die unterschiedlichen Charakteristika der Spannungsplateaus modellieren.

References

- [1] D. Bresser, S. Passerini und B. Scrosati. *Recent progress and remaining challenges in sulfur-based lithium secondary batteries - a review*. Chem. Commun., V. 49, S. 10545-10562, 2013
- [2] V. Knap, D. Stroe, R. Teodorescu, M. Swierczynski and T. Stanciu. *Electrical circuit models for performance modeling of Lithium-Sulfur batteries*. Konferenz Energy Conversion Congress and Exposition (ECCE), in Montreal, 2015
- [3] C. Parfitt. *Characterisation, Modelling and Management of Lithium-Sulphur Batteries for*

- Spacecraft Applications*. Dissertation, University of Warwick, 2012
- [4] C. Scherr. *Elektrochemisches Verhalten von Lithium-Schwefel-Zellen mit unterschiedlicher Kathodenstruktur*. Dissertation, Karlsruher Institut für Technologie, 2014
- [5] K. Kumaresan, Y. Mikhaylik und R. White. *A Mathematical Model for a Lithium-Sulfur Cell*. Journal of The Electrochemical Society, V. 155, S. A576-A582, 2008
- [6] Y. Mikhaylik, I.Kovalev, R. Schock, K. Kumaresan, J. Xu und J. Affinito. *High Energy Rechargeable Li-S Cells for EV Application: Status, Remaining Problems and Solutions*. Journal of The Electrochemical Society, V. 25, S. 23-34, 2010
- [7] Y. Mikhaylik und J. Akridge. *Polysulfide Shuttle Study in the Li/S Battery System*. Journal of The Electrochemical Society, V. 151, S. A1969-A1976, 2004
- [8] V. Knap, D. Stroe, R. Teodorescu, M. Swierczynski and T. Stanciu. *Comparison of parametrization techniques for an electrical circuit model of Lithium-Sulfur batteries*. Konferenz Industrial Informatics (INDIN), in Cambridge, 2015
- [9] M. Busche, P. Adelhelm, H.Sommer, H. Schneider, K. Leitner und J. Janek. *Systematical electrochemical study on the parasitic shuttle-effect in lithium-sulfur-cells at different temperatures and different rates*. Journal of Power Sources, V. 259, S. 289-299, 2014
- [10] L. Yao, J. Aziz, P. Kong und N. Idris. *Modeling of lithium-ion battery using MATLAB/Simulink*. Konferenz Industrial Electronics Society (IECON), in Wien, 2013
- [11] V. Kolosnitsyn und E. Kuzmina. *A study of the electrochemical processes in lithiumsulphur cells by impedance spectroscopy*. Journal of Power Sources, V. 196, S. 1478-1482, 2011
- [12] P. Keil und A. Jossen. *Aging of Lithium-Ion Batteries in Electric Vehicles: Impact of Regenerative Braking*. Konferenz EVS28, in Kintex Korea, 2015

Dr.-Ing. Carsten Matysczok
Markus Kühn
Moritz Schele
UNITY AG

End-to-End Kliniksimulation – Absicherung der Klinik-Planung durch Simulation vollständiger Prozessketten

Seit mehreren Jahren befindet sich das deutsche Gesundheitssystem in der öffentlichen Kritik. Die Fortschritte in der Medizin und -technik führen zu immer höheren Kosten. Parallel dazu stellen die Alterung der Gesellschaft und der Rückgang der Bevölkerungszahlen die Finanzierung der sozialen Sicherungssysteme vor große Herausforderungen. Die Kliniken sind nun gefordert, ihre immer aufwändigeren Leistungen bei gleicher Qualität wirtschaftlich zu erbringen. Der Druck zum Handeln ist hoch: die steigende Kosten und immer komplexere Organisationsabläufe stellen viele Kliniken vor große Herausforderungen.

Etablierte Methoden und der zielorientierte Einsatz der Ablaufsimulation, die seit vielen Jahren erfolgreich in der Fertigungsindustrie eingesetzt werden, können dazu beitragen, diese Herausforderungen zu meistern. Durch sie ist eine vorausschauende und durchgängige Ablaufplanung innerhalb der Klinik sowie in den vor- und nachgelagerten Prozessketten möglich. Ziel ist die Beantwortung unterschiedlicher Fragestellungen im Rahmen der digitalen Klinikplanung: Wie viele OP-Säle, Aufwachraumplätze und Betten auf Normal- und Intensivstationen werden in Zukunft benötigt? Ist die Notfallambulanz auch bei stärkerem Patientenandrang noch ausreichend dimensioniert? Wie kann die Versorgung mit sterilisierten Instrumenten bei steigender Fallzahl und sich ständig ändernden Anforderungen sichergestellt werden?

Die Simulationstechnik hat sich zu einem wirksamen Beratungswerkzeug für die digitale Klinikplanung bei der UNITY entwickelt. Zu Beginn wurde der OP-Bereich mit Hilfe der OP-Simulation geplant und optimiert. Wegen des großen Erfolgs wurden zunehmend auch weitere Klinikbereiche mit Hilfe der Simulationstechnik optimiert. Es folgten beispielsweise die Notaufnahme, die Elektivambulanzen und Sterilgut- und Materialversorgung. Mittels dieser Technik lässt sich ein Zukunftsmodell der Klinik erstellen, um die zuvor skizzierten Fragestellungen in unterschiedlichen Szenarien zu bewerten und frühzeitig die richtigen Entscheidungen zu treffen. Sie leistet dabei einen entscheidenden Beitrag, die Planungszeiten von Klinikneubauten oder Erweiterungen zu verkürzen und die bestmögliche Alternative für die Realisierung auszuwählen. Dieses belegen eine Vielzahl von erfolgreich durchgeführten Projekten: So konnte z.B. die Anzahl an durchgeführten OPs bei einem gleichbleibenden Personaleinsatz um 1.900 Fälle gesteigert werden, die Wechselzeiten um 66% reduziert und somit Leerstände mehr als halbiert werden. Resultat waren Mehrgewinne in sechsstelliger Höhe.

Dieser Beitrag beschreibt den konsequenten Einsatz der Simulation in der gesamten klinischen Prozesskette: sowohl innerhalb der Patientenbehandlung (die Abläufe im OP, der Notaufnahme und den Elektivbereichen – siehe Abbildung 1) als auch in den vor- und nachgelagerten Prozessketten (Sterilgutversorgung, Intralogistik und die Materialbelieferungen innerhalb von Klinikverbunden).

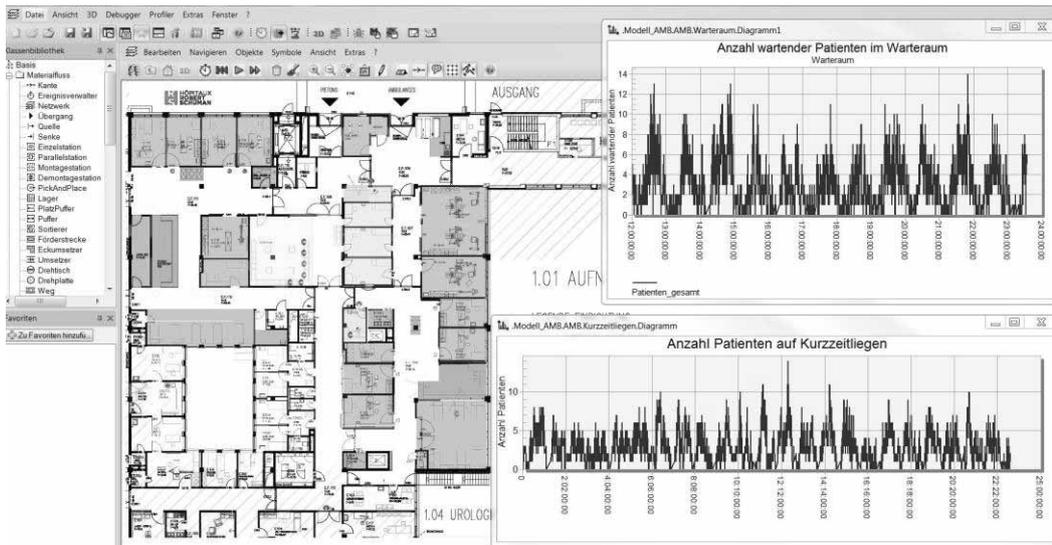


Abbildung 1: Simulation einer Elektivambulanz

Durch den Einsatz der Ablaufsimulation können die klinikinternen Prozesse effizienter gestaltet und besser aufeinander abgestimmt werden. Die Basis bildet ein Simulationsmodell, welches zur korrekten räumlichen Anordnung auf einem Grundriss (in Form eines 2D-CAD-Modells) aufgebaut wird. Die Standardfunktionalitäten der Simulationssoftware *Plant Simulation* von Siemens PLM wurden durch eine Reihe von speziell entwickelten Klassenbibliotheken erweitert, die durch Kombination und Parametrisierung individuell angepasst werden. Seit dem ersten Einsatz der Simulation zur Absicherung der Klinikplanung werden die Bibliotheken in der UNITY dabei stetig weiterentwickelt. Die zunächst aus Standardklassen abgeleiteten Bausteine wurden in jedem Projekt um weitere Funktionen und Schnittstellen erweitert und anschließend standardisiert in die Bibliothek aufgenommen. Die geschaffenen Klassen reichen dabei von kleineren Funktionsbausteinen wie Einleiträume oder der Bettenwarte, über Bausteine mit höherer Komplexität wie Operationssälen im OP-Bereich, bis hin zu ganze Abteilungen, z.B. der Sterilgutaufbereitung (ZSVA). Durch die konsequente Vererbung der Klasseigenschaften und der über Excel-Tabellen steuerbaren variablen Parametern werden individuelle Kliniken oder ganze Kliniknetzwerke abgebildet. Diese generisch aufgebauten Modelle lassen sich dadurch prinzipiell auf jede Klinik- oder Netzwerkgröße skalieren und um jede relevante Funktion im Kliniknetzwerk erweitern. Nach der erstmaligen Modellbildung können diese auch flexibel auf sich ändernde Planungsprämissen eingestellt werden.

In der OP-Simulation werden dazu beispielsweise die Abläufe mit direktem Bezug zu einer Operation im Simulationsmodell abgebildet. Diese reichen vom Abruf des Patienten von der Station bis zur Verlegung des Patienten aus dem OP mit dem Ziel „Bettenstation“. Dazwischen werden die medizinisch notwendigen Schritte von Umbetten und Anästhesieeinleitung über die Vorbereitung des Patienten und der Instrumente sowie die tatsächlichen Operation abgebildet. Grundlage dafür bilden die krankenhaus- und fachbereichsindividuellen Prozesse und Daten (Operationszeiten, Fallzahlen, Saalbelegung), die zur Simulation ausgewertet und aufbereitet werden.

Bei der Simulation von Notfall- und Elektivambulanzen sind die Prozesse deutlich heterogener. Auch hier beginnt die Simulation mit der Ankunft des Patienten im jeweiligen Bereich und endet nach erfolgter Untersuchung mit der Entlassung bzw. der Verlegung in angrenzende Bereiche. Es werden die Abläufe der Patientenannahme, der pflegerischen sowie ärztlichen Untersuchung und verschiedener Diagnostiken simuliert und die erforderlichen Kapazitäten dimensioniert. Ebenso werden die hier unumgänglichen Wartebereiche berücksichtigt und optimal ausgelegt.

Als dritter Bereich der Patientenversorgung sind die Bettenstationen zu berücksichtigen. Hier sind der Zeitpunkt der Patientenaufnahme, die Diagnose und die jeweilige Verweildauer ausschlagge-

bend, um die Größe und gegebenenfalls fachliche Zuordnung der Stationen zu bestimmen. Zusätzlich zu den Prozessen, die den Patienten betreffen gibt es eine Vielzahl von unterstützenden und versorgenden Prozessen. Im OP-Betrieb kommt es heute nicht selten vor, dass eine Operation auf Grund von fehlenden Instrumenten oder Verbrauchsgütern abgesagt werden muss. Die Planung und Optimierung dieser Abläufe ist daher der konsequente nächste Schritt.

Viele Kliniken und Klinikverbunde nutzen heute eine gemeinsame Sterilgutaufbereitung (ZSVA). Dabei stellt der OP-Bereich hohe Anforderungen an Qualität und rechtzeitige Wiederbereitstellung der Instrumente (Sterilgut) aus der ZSVA. Die oftmals kostenintensiven und daher knappen Instrumentensets müssen innerhalb von 24 Stunden aufbereitet und ausgeliefert werden. Interne Prozesse und Transporte zwischen ZSVA und der Kliniken sind daher vor große Herausforderungen gestellt. Gegenstand der Simulation ist daher die Auslegung und Optimierung von Stellflächen, Packplätzen, Gerätebedarfen innerhalb der ZSVA sowie Versorgungsrhythmen, Tourenpläne und Notfallkonzepte zwischen ZSVA und Kliniken (siehe Abbildung 2).

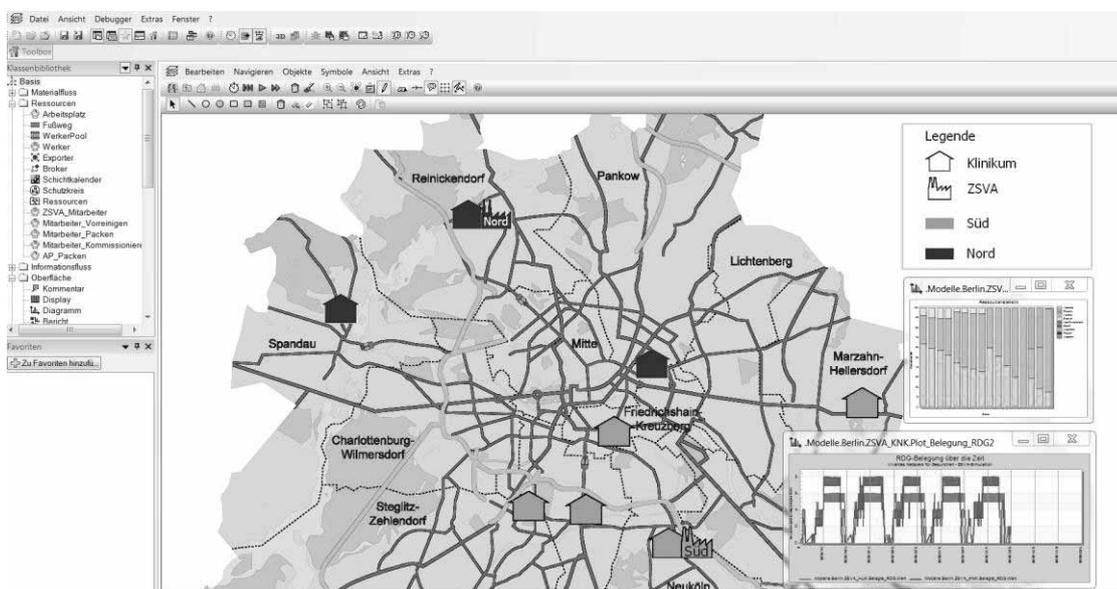


Abbildung 2: Dieser Klinikverbund wird durch zwei ZSVAs versorgt

Der nächste Schritt in der ganzheitlichen Optimierung des Klinik- und ZSVA-Betriebs ist die Verknüpfung der ZSVA- und der OP-Simulation. Durch die gemeinsame Planung kann eine optimale Ausnutzung der Ressourcen erzielt werden. In Einzelfällen kann dadurch die Wiederbeschaffungszeit von OP-Sieben nochmal drastisch verkürzt werden, sodass seltene Instrumentensätze an einem Tag mehrfach genutzt werden können. Ähnliche Synergieeffekte sind auch für weitere Versorgungs- und Reinigungsprozesse zu erwarten. Die Belieferung mit Verbrauchsmaterialien aus einem Zentrallager, Kleidung, Schuhe und Kittel aus der Wäscherei und Betten aus der Bettenzentrale sind nur einige Beispiele für diese klinikinternen Prozesse.

All diese Prozesse und ihre Zusammenhänge gilt es zukünftig in den Simulationsmodellen zu berücksichtigen und ganzheitlich abzubilden. Für eine vollständige End-to-End Betrachtung der Patientenversorgung und die optimale Planung und Dimensionierung der Bereiche ist eine Verknüpfung der Modelle erforderlich. UNITY hat die Bausteine zur Simulation von Ambulanzbereichen, OP-Bereich und Versorgungseinheiten wie die ZSVA so aufgebaut, dass sie sowohl einzeln, als auch verknüpft in einem End-to-End Simulationsmodell genutzt werden können. Je nach Aufgabenstellung kann es ausreichend sein, nur einen der Bereiche abzubilden und diesen zu optimieren. Doch birgt die Simulation von Einzelbereichen die Gefahr, zu lokalen Optima zu gelangen, statt zum gesamtheitlichen Optimum unter Berücksichtigung aller Interdependenzen.

Durchsatzoptimierung von Kommissionierrobotern durch Simulationsstudie

Prof. Dr.-Ing. Martin Lucas

Hochschule Hamm-Lippstadt, Lehrgebiet Technische Logistik

martin.lucas@hshl.de

Im Rahmen einer Simulationsstudie wurde die Kommissionierung für eine automatische Kommissionierzone mit Hochregallageranbindung abgebildet und optimiert. Den Kern der Kommissionierzone bildet ein Kommissionierroboter, welcher gemäß den Kundenaufträgen Packstücke von Quellpaletten entnimmt und auf Kundenpaletten positioniert. Das Ziel war die Durchsatzsteigerung durch Modifizierung der Steuerungsstrategien, der Roboterbelegung sowie der Optimierung der Auftragsprofile.

Auf Grund der hohen Komplexität und dynamischen Vernetzung der einzelnen Fördertechnikkomponenten innerhalb der Kommissionierzone sowie den Variablen in der Auftragsprofilegestaltung war dabei eine Untersuchung mit analytischen Methoden nur begrenzt möglich. Daher wurde eine ereignisorientierte Simulationsstudie für die Materialflüsse in der Kommissionierzone durchgeführt.

Auf Basis der aufgenommenen Materialflüsse und der Steuerungsstrategie wurde zunächst das Anlagenlayout abgebildet. Die Modellvalidierung erfolgt über Abgleich der Ist-Daten mit den Daten aus der Simulationsstudie. Hierzu wurden charakteristische Lastprofile herangezogen.

Zur Durchsatzoptimierung wurden mit variierenden Lastprofilen Simulationsläufe für Varianten in den Steuerungsstrategien sowie gestaffelte Anzahlen an Kundenaufträgen für die Batchbildung durchgeführt. Dabei beschreibt ein Lastprofil die Kundenaufträge in einem definierten Zeitraum mit den zugehörigen Daten wie Artikelnummer, Entnahmemengen, Lieferdatum. Mit Hilfe der Simulationsstudie konnten Erkenntnisse gewonnen und bewertet werden, die den gesamthaften Durchsatz der Kommissionierzone signifikant erhöhen.

1 Einleitung

Insbesondere durch die aktuell stark steigenden Umsätze und Auftragsvolumen im Onlinehandel nimmt die Kommissionierung zunehmend eine logistische Schlüsselfunktion ein. Dabei wird der Vorgang des Kommissionierens gemäß VDI 3590 definiert durch „Kommissionieren hat das Ziel, aus einer Gesamtmenge von Gütern (Sortiment) Teilmengen (Artikel) auf Grund von Anforderungen (Aufträge) zusammenzustellen“ [1]. Innerhalb der Intralogistik ist die Kommissionierung als besonders komplexer und personalintensiver Prozess zu kennzeichnen [2]. So entfallen in Abhängigkeit z. B. zur eingesetzten Technik oder dem Auftragsvolumen bis zu 70 % der gesamthaften Logistikkosten auf das Kommissioniersystem [3]. Vornehmlich aus diesen Gründen kann die Effizienz der Kommissionierung zu wesentlichen Wettbewerbsvorteilen führen.

Durch eine Automatisierung der Kommissionierprozesse lässt sich die Personalintensität reduzieren.

Dabei wird in Ware-zur-Person (WzP) Systemen die Bereitstellung der sortenreinen Quellpaletten sowie der Weiter- und Rücktransport der sortenunreinen Kundenpaletten und ggf. der Anbruchpaletten zumeist über automatisierte Fördertechnik realisiert. Der Kommissioniermitarbeiter entnimmt an zentralen Punkten die Packstücke von den Quellpaletten und positioniert diese auf der Kundenpalette. Durch diese Maßnahme lässt sich die Einsatzzeit des Kommissioniermitarbeiters auf den eigentlichen Entnahmeprozess fokussieren.

Bei entsprechender Homogenität des Sortimentes, also der zu kommissionierenden Packstücke, kann auch der Greifprozess automatisiert werden. Hier kommen z. B. Knickarmroboter oder Flächen- bzw. Linienportale zum Einsatz. Durch geeignete Lastaufnahmemittel, wie Vakuumsauger oder Klemmvorrichtungen, werden die Packstücke durch den Kommissionierroboter von der Quellpalette aufgenommen und auf der Kundenpalette positioniert.

In der im Folgenden beschriebenen Simulationsstudie wurde eine derartige Kommissionierzone mit Kommissionierroboter abgebildet, analysiert und optimiert.

2 Beschreibung Betrachtungsgegenstand

Abbildung 1 stellt das Layout der Kommissionierzone dar, mit den wesentlichen Komponenten Kommissionierroboter und Querverschiebewagen.

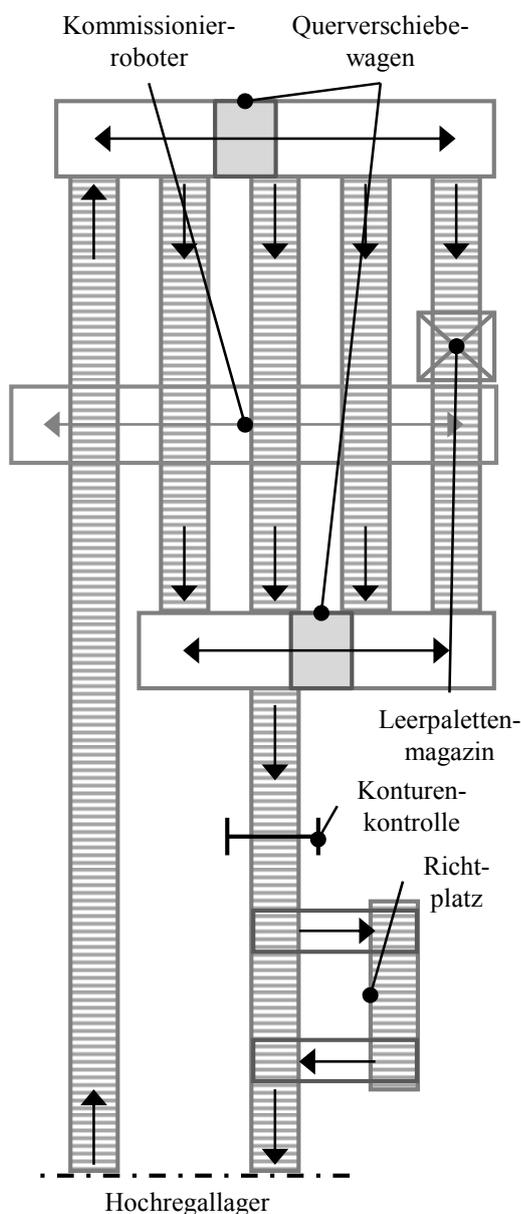


Abbildung 1. Layout der Kommissionierzone

Aus einem automatischen Hochregallager werden Quellpaletten in eine Kommissionierzone gefördert. Den Kern der Kommissionierzone bildet ein Kommissionierroboter. Als Kommissionierroboter wird ein Linienportal mit zwei Freiheitsgraden eingesetzt.

Der Kommissionierroboter entnimmt entsprechend den Kundenaufträgen Packstücke von der Quellpalette und positioniert diese auf der Kundenpalette gemäß den Packmustervorgaben. Dabei wird sowohl positiv als auch negativ kommissioniert, um die Anzahl der umzusetzenden Packstücke zu minimieren. Es werden parallel zwei Kundenpaletten kommissioniert.

Die Ladehilfsmittel der vollständig abkommissionierten Quellpaletten werden in Palettenmagazinen automatisch gesammelt oder bei Bedarf direkt als neue Kundenpalette bereitgestellt. Sowohl angebrochene Quellpaletten als auch vollständig kommissionierte Kundenpaletten gehen zurück in das Hochregallager. Kundenpaletten werden dann gemäß den Vorgaben aus dem Versand für die Lkw-Verladung ausgelagert und bereitgestellt.

3 Vorgehensweise Simulationsstudie

Die Durchführung der Simulationsstudie lässt sich in sechs charakteristischen Schritten beschreiben. Abbildung 2 verdeutlicht diese Vorgehensweise.

Zunächst wurde der Betrachtungsraum definiert und eingegrenzt sowie die Zielgrößen definiert. Als Betrachtungsraum wird die Kommissionierzone einschließlich der Prozesse zur Ladehilfsmittelver- und Entsorgung sowie der Konturenkontrolle beschrieben.

Das Hochregallager dient lediglich als Quelle bzw. Senke für Quell-, Kunden- und Anbruchpaletten sowie Leerraum für die Kontrolle der Konturen. Zur Bewertung der Prozesse innerhalb der Kommissionierzone ist eine Abbildung der Hochregallagerprozesse nicht relevant, da die Leistungs- sowie Kapazitätskennzahlen für das betrachtete Gesamtsystem keine begrenzenden Faktoren darstellen.

Im Rahmen der Datenanalyse und -erhebung wurden sämtliche zur Beschreibung des Systems und der ablaufenden Prozesse erforderlichen Daten und Informationen zusammengestellt. Es wurden Lastprofile und Strategien definiert.

Auf dieser Basis wurden zunächst das physische Layout sowie die Strategien abgebildet. Dabei basiert der Modellierungsansatz auf einer ereignisorientierten Ablaufsimulation. Dieser Modellierungsansatz stellt

für derartige Fragestellungen in Punkto Abbildungsgenauigkeit sowie Modellierungs- und Rechenaufwand den aktuellen Stand der Technik dar.

Die Modellvalidierung erfolgt über Abgleich der Ist-Daten mit den Daten aus der Simulationsstudie. Hierzu wurden charakteristische Lastprofile herangezogen. Die Validierung erfolgt sowohl an Einzelkomponenten der Fördertechnik sowie dem Gesamtsystem.

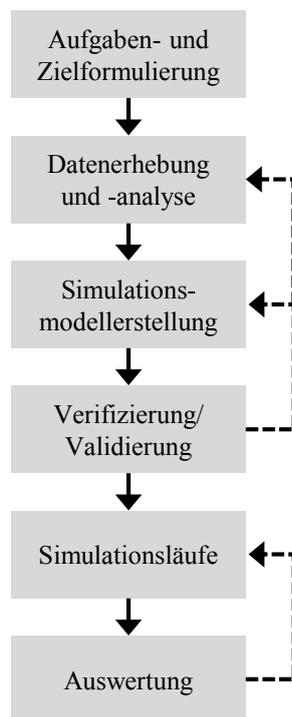


Abbildung 2. Ablauf Simulationsstudie [in Anlehnung an 4]

Anhand des definierten Versuchsprogrammes wurden die erforderlichen Simulationsläufe durchgeführt unter Dokumentation und Analyse der relevanten Kenngrößen.

4 Lastprofile

Zur Durchsatzoptimierung wurden mit variierenden Lastprofilen Simulationsläufe für Varianten in den Steuerungsstrategien sowie gestaffelte Anzahlen an Kundenaufträgen für die Batchbildung durchgeführt. Dabei beschreibt ein Lastprofil die Kundenaufträge in einem definierten Zeitraum mit den zugehörigen Daten wie Anzahl Artikelnummer, Entnahmemengen,

Auslieferungsdatum. Tabelle 1 gibt eine Übersicht zu beispielhaften Lastprofilen.

Lastprofil	Anzahl Kundenaufträge	Durchschnittliche Entnahmemenge	Anzahl Artikelnummern
I	500	14	49
II	400	13	34
III	300	13	27

Tabelle 1. Beispielhafte Lastprofile

Das Auslieferungsdatum hat in der Simulationsstudie eine untergeordnete Priorität. Der Großteil der Kundenaufträge geht mit ausreichendem zeitlichen Vorlauf ein, so dass die Kommissionierung mit einem zeitlichen Vorlauf von einigen Werktagen erfolgt.

Dies bedeutet, dass kommissionierte Kundenpaletten vor der Bereitstellung im Versand im Hochregallager eingelagert bzw. gepuffert werden und von dort entsprechend abgerufen werden.

Die Variation des Auftragspools, also der Anzahl an Kundenaufträgen über die eine Batchbildung erfolgt, ist eine der wesentlichen Stellgrößen für die Optimierung der Steuerungsstrategie. Die Vergrößerung des Auftragspools hat damit zur Folge, dass eine größere Anzahl an Kundenpaletten mit den gleichen Artikeln von einer Quellpalette bestückt werden können. Für das Gesamtsystem bedeutet dies eine Reduzierung der zu fördernden Quell- und Anbruchpaletten.

Kurzfristige Kundenbestellungen werden mit Priorität als sogenannte Eilaufträge bearbeitet. Diese Aufträge stellen für den konstanten Betrieb des Kommissioniersystems eine signifikante Störgröße dar. Aus diesem Grund sind für diese Auftragsprozesse definierte Zeitfenster zu Schichtbeginn reserviert, so dass ein konstanter Betrieb für die Standardprozesse gewährleistet werden kann.

5 Auswertung und Ergebnisse

Zur Durchsatzoptimierung wurden mit variierenden Lastprofilen Simulationsläufe für

- Varianten in den Steuerungsstrategien sowie
- gestaffelte Anzahlen an Kundenaufträgen für die Batchbildung

durchgeführt.

Als Kenngröße zur Bewertung des Durchsatzes des Kommissioniersystems werden die kommissionierten Kundenpaletten pro Stunde ausgewertet.

In einem ersten Schritt wurden die Arbeitszustände des Kommissionierroboters untersucht. Dabei sind die Arbeitszustände

- wartet,
- kommissioniert sowie
- gestört

abzugrenzen.

Im Arbeitszustand „wartet“ steht aktuell keine Quell- und/oder Kundenpalette bereit. Es wird nicht kommissioniert, da über die Fördertechnik zunächst eine Quell- und/oder Kundenpalette bereitgestellt werden muss.

Mit „kommissioniert“ wird der Produktivbetrieb des Kommissionierroboters beschrieben. In diesem Arbeitszustand wird gemäß den Kundenaufträgen kommissioniert. Zur Durchsatzoptimierung wird angestrebt, diesen Arbeitszustand gesamthaft zu erhöhen.

Störungen im Kommissionierbetrieb oder am Kommissionierroboter werden mit dem Arbeitszustand „gestört“ zusammengefasst. Dies können z. B. auftretende Fehler beim Greifprozess oder in der Ladehilfsmittelbereitstellung sein. In der Simulationsstudie wurde dieser Arbeitszustand über einen fixen Prozentsatz abgebildet. Ansätze zur Reduzierung von Störungen waren nicht Gegenstand der Simulationsstudie, da zumeist mechanische oder sensorische Ursachen der Auslöser sind.

Abbildung 3 zeigt am Beispiel von Lastprofil I zunächst die Arbeitszustände in der Ausgangssituation, also mit der aktuell implementierten Steuerungsstrategie.

In Übereinstimmung mit den Auswertungen zu den Event-Log-Dateien der Anlagensteuerung kommissioniert der Kommissionierroboter zu 55 % der Ge-

samtlaufzeit. 40 % entfällt auf den Arbeitszustand „wartet“. Dies ist im Wesentlichen auf eine nicht vollständige Sequenzierung der Quellpaletten auf der zuführenden Fördertechnik zurückzuführen.

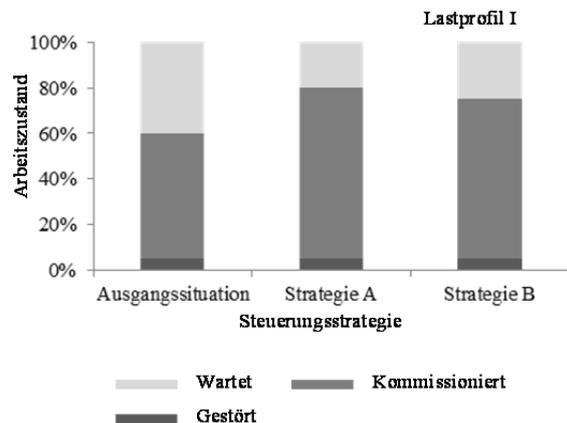


Abbildung 3. Arbeitszustand des Kommissionierroboters in Abhängigkeit zur Steuerungsstrategie für Lastprofil I

In den Simulationsläufen wurden ausgehend von der Steuerungsstrategie in der Ausgangssituation Anpassungen vorgenommen. Hierzu wurde in den Strategien A bzw. B eine konsequente Sequenzierung der auszulagernden Quellpaletten vorgenommen. Zudem wurde bei Strategie A der Meldezeitpunkt zur Anforderung von neuen Quellpaletten auf den Startpunkt des Übergabeförderers aus dem Hochregallager gelegt.

Für Strategie B erfolgt dies weiterhin in Analogie zur Ausgangssituation auf dem zuführenden Querverschiebewagen. Diese Maßnahmen führen für Strategie B zu einer prozentualen Erhöhung des Arbeitszustandes „kommissioniert“ von + 15 %. Mit der Verlagerung des Meldezeitpunktes lässt sich in der Strategie A eine weitere Steigerung um + 5 % erreichen auf einen gesamthaften Anteil von 75 %.

Abbildung 4 zeigt auszugsweise Ergebnisse der Simulationsläufe mit unterschiedlichen Lastprofilen. Es wurden für drei vergleichbare Lastprofile die Anzahl an Kundenaufträgen für die Batchbildung zwischen 5 und 30 Aufträgen variiert.

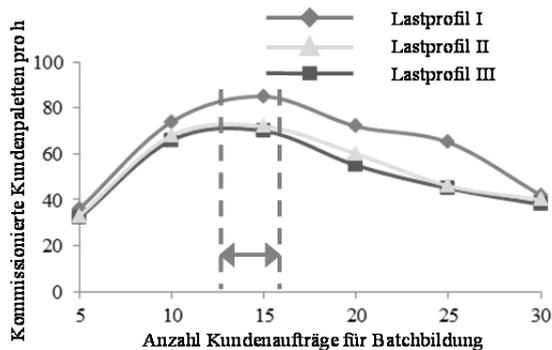


Abbildung 4. Kundenpaletten pro Stunde in Abhängigkeit zur Anzahl Kundenaufträge für Batchbildung

Als Bewertungsgröße wurden die kommissionierten Kundenpaletten pro Stunde aus den Simulationsläufen ausgewertet. Für dieses Beispiel lässt sich der Durchsatz an kommissionierten Kundenpaletten pro Stunde für eine Batchbildung über 13 bis 16 Kundenaufträgen optimieren. Das heißt, in diesem Intervall werden je nach gewähltem Lastprofil 70 - 82 Kundenpaletten pro Stunde kommissioniert. Eine deutliche Vergrößerung der Anzahl an Kundenaufträgen hat negative Auswirkungen auf den Durchsatz. So reduziert sich der Durchsatz bei 25 Kundenaufträgen um etwa 30 %. Eine vergleichbare Situation lässt sich für eine Anzahl an Kundenaufträgen von kleiner 13 beschreiben. Für diesen Bereich reduziert sich die Anzahl an kommissionierten Kundenpaletten.

6 Zusammenfassung

Mit Hilfe der Simulationsstudie konnten Erkenntnisse gewonnen und bewertet werden, die den gesamthafte Durchsatz der Kommissionierzone signifikant erhöhen. Dabei wurden zum einen die Steuerungsstrategien zur Versorgung der Kommissionierzone variiert und zum anderen unterschiedliche Batchbildungsvarianten getestet und bewertet. Auf dieser Basis ist es möglich, den Durchsatz der Kommissionierzone um ca. 30 % gegenüber der Ausgangssituation zu steigern.

Die dargestellte Simulationsstudie verdeutlicht, dass für die Analyse und Optimierung derartiger Systeme eine ereignisorientierte Ablaufsimulation eine geeignete Methode darstellt. Dabei lassen sich die vernetzten Wirkzusammenhänge effizient abbilden, analysieren und bewerten.

7 References

- [1] Verein Deutscher Ingenieure, *VDI 3590 Blatt 1 - Kommissioniersysteme Grundlagen*. Beuth Verlag, Deutschland, 1994.
- [2] M. ten Hompel, V. Sadowsky und M. Beck. *Kommissionierung*. Springer-Verlag, Deutschland, 2011.
- [3] M. Pulverich und J. Schietinger. *Handbuch Kommissionierung*. Verlag Heinrich Vogel, Deutschland 2009.
- [4] Verein Deutscher Ingenieure, *VDI 3633 Blatt 1 - Simulation von Logistik-, Materialfluß- und Produktionssystemen Grundlagen*. Beuth Verlag, Deutschland, 1993.

A Simple Model of Cruising for Garage Parking

Oliver Ullrich¹, Benjamin Brandt¹, Daniel Lückerath², Naphtali Rische¹

¹School of Computing and Information Sciences, Florida International University,
ECS 243C, 11200 SW 8th St, Miami FL-33199

²Institut für Informatik, Universität zu Köln, Weyertal 121, 50931 Köln, Germany

oullrich@cs.fiu.edu

Parking guidance systems have to be calibrated and evaluated before their application in the field. One step to accomplish this is to evaluate them in a simulated environment.

This paper presents a simple model of cruising for parking in a parking garage with focus on individual drivers' behavior. While the described model itself is too basic to completely depict a driver's decision making process, it is a step towards a more comprehensive representation. In the process of developing the model it became clear which pieces are still missing from the puzzle, and what data has to be collected to successfully represent a parking garage's operational day.

After a short introduction this paper shares some background on garage parking and related work. A simple garage parking model is presented, followed by some thoughts on necessary steps to expand and refine the model. The paper concludes with a summary of the lessons learned and an outlook on further research.

1 Introduction

With parking garages containing 2,000 or more individual spaces (see figure 1), computer based systems which provide guidance and recommendations to find available parking in these major structures are significantly beneficial to users, and also improve resource utilization for parking providers. Such parking guidance algorithms have to be calibrated and evaluated before their application in the field. One step to accomplish this is to evaluate them in a simulated environment.

This paper presents a simple simulation model of cruising for garage parking, which will, once extended and rigorously validated, serve as a virtual testbed for calibrating and evaluating garage parking guidance algorithms. The model applies a combination of two simulation paradigms: while the model's basic mechanics, e.g. the arrival of cars, is modeled in an event-based (see [4] and [6]) fashion, the agent-

based paradigm (see [14]) is utilized for modeling the drivers' decision making.

After this introduction, the paper continues with sharing some background on garage parking and related research (see section 2); then a simple model of cruising for garage parking is presented (see section 3). Following on to this, the paper describes the future research steps necessary (see section 4) and closes with a short summary (see section 5).

2 Background

2.1 Garage parking

The term garage parking refers to the process of entering a building at least partially designated for car parking, finding and navigating to an available parking spot, leaving the car at that spot for a while, and

then de-park by finding the shortest or most convenient path from the parking spot to a vehicular exit. As the intended application for the developed model is to test recommendation algorithms which are concerned with reducing the time spent cruising for available parking, the last part of the process, de-parking, is beyond the scope of this paper and will not be discussed further. The described buildings are often referred to as parking garages, but also as multistorey car parks, parkades, or parking structures.

Garage parking, together with parking lot parking, is often described by the more general term off-street parking. This contrasts with on-street parking with its diverse modes: parallel parking, angular parking, perpendicular parking.

The parking garage usually consists of a number of connected levels, which are themselves composed of a number of areas. Each area contains of a number of parking spots fit for individual cars. The readers will know this decomposition from their own experience: "I parked my car on a spot on level 3, in area C."

Vehicular access to the parking garage is granted, often at the ground floor, by entry and exit lanes, which are usually unidirectional. Pedestrians access the garage via elevators or stairways, or on the ground floor by doorways. Pedestrian access ways are usually bi-directional.

2.2 Related research

Corresponding to its importance in planning and design of public spaces, on-street parking has seen a lot of research attention, both in general modeling (see e.g. [2], [3], [13], [18], [19], [22], [23], [24]) and in simulation modeling (see [5], [7], [9], [8], [15], [16]). Most of the more recent simulation models being at least partially agent-based (see [5], [7], [9], [15]). Building upon this, Dieussaert et al. (see [7]) and Horni et al. (see [9]) combine agent-based modeling with the cellular automata paradigm, while Gallo et al. (see [8]) construct a multi-layer network supply model. Some authors (see [7], [15]) utilize the described models to evaluate pricing and other policy considerations, while others (see [5], [9], [8], [16]) apply them to analyze technical methods to reduce cruising time and thereby traffic in general.



Figure 1: Parking garage with approx. 2,000 parking spaces

Only a few models (see [1], [5], [7], [22]) consider off-street parking: Asakura and Kashiwadani (see [1]) apply a model to examine the effect of different types of on-street and off-street parking availability information on overall system performance, but do not examine the drivers' behavior inside of individual parking lots. Benenson et al. (see [5] and also [13]) develop a spatially explicit model of parking search and choice, with simulated drivers cruising through an artificial or real-life city center model, giving them both on-street and off-street parking options. Dieussaert et al. (see [7]) also are interested in the traffic patterns generated by cruising for parking. They model on-street parking as well as parking lots and garages, but consider parking lots and garages as simple sinks, not modeling their interior. Van der Waerden et al. (see [22]) develop a simple cellular automata based sub-model for choosing parking spaces inside a parking lot, but clearly set their focus on modeling traffic patterns resulting from the whole, city-wide process of traveling and parking.

Only a few of the described models consider off-street parking in any form. None of these is detailed enough for the evaluation of garage parking recommendation systems.

3 Modeling cruising behavior

An agent-based model usually includes two components (see [14]): the agents themselves, and the environment they interact with.

The agents are usually self-contained and autonomous; they have attributes whose values change

over the course of a simulation run. Their behavior is determined by a set of rules, and they interact dynamically with other agents and the environment they exist in. In more complex models, agents are often goal-directed and adaptive, and may even be heterogeneous. Individual agents usually only interact with a local sub-set of the environment and other agents, i.e. in addition to their own memory, only local information is available to them.

In addition to their communication with their set of neighbors, agents interact with their environment. This information might provide only basic information, e.g. the agent's position in the environmental model. It may also provide more detailed information, e.g. the capacity and real-time rate of occupancy of parking garage areas. While in many cases the environment might be modeled as an attributed graph structure (see [20]), it sometimes is built as a complex simulation itself, e.g. based on cellular automata (see [12]).

In the described model, drivers and their cars are modeled as agents adhering to a set of rules and acting on local information, while the parking garage is modeled as an attributed neighborhood graph, and constitutes the agents' environment.

3.1 Modeling parking garages

The parking garage is modeled as an attributed graph $G(A, E)$ representing the garage's layout and the areas' neighborhood relations. A node $a \in A$ represents an area of the parking garage, an edge $e(a_i, a_j) \in E$ with $a_i, a_j \in A$ represents a direct connection of two areas a_i and a_j which is traversable by car. If all access lanes in the parking garage are two-way, the garage can be modeled as an undirected graph. If some or all segments only allow for one-way traffic, a directed graph can be established. As it is the garage planner's basic objective to ensure reachability of each parking area, the graph consists generally only of one connected component.

Each node $a \in A$ is attributed by its total number of parking spots z_a , the number of currently occupied spots $o_a(t)$ at time t , by extension also the number of free spots $f_a(t) = z_a - o_a(t)$ at time t , and the average time r_a a car needs to traverse the area. The recommendation method to be tested (see [21]) explicitly

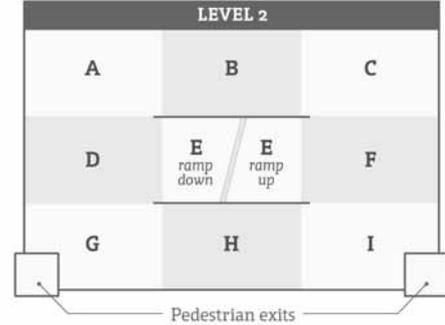


Figure 2: Simplified parking garage level with two exits, two bi-directional ramps, and nine areas

considers only these areas, and does not depict individual parking spots. Therefore, a spatially explicit modeling of these individual spots is not necessary in this context.

Each edge $e(a_i, a_j) \in E$ is attributed by a time r_e a car needs to traverse the edge from leaving a_i , and to reach the indicated area a_j . In most cases, if areas are directly adjoining, $r_e = 0$ can be assumed.

In this simplified model we assume an infinite traversal capacity for nodes and edges, therefore ignoring congestion resulting from multiple cars cruising the same area.

The garage's entry lanes are modeled as special nodes $a_e \in A_e \subset A$ with $z_{a_e} = 0$, which serve as sources for the transient car agents. In discrete modeling, interarrival times are usually approximated with an exponential distribution with an arrival rate of λ and average interarrival times of $\mu = 1/\lambda$ (see [4], pg. 248). In the described model the distribution parameter $\lambda_{a_e}(t)$ is established for each entry lane a_e by input data analysis (see section 3.3) and dependent on the time t of day. Technically, the agents are generated by the event-based framework at each entry node at appropriately distributed simulation times.

A parking garage's exit lanes are modeled as special nodes $a_x \in A_x \subset A$, again with $z_{a_x} = 0$, doubling as sinks for transient entities, collecting statistic data and removing the agents from the model.

Figure 2 shows a simplified layout of a parking garage level, while figure 3 shows the corresponding partial model graph.

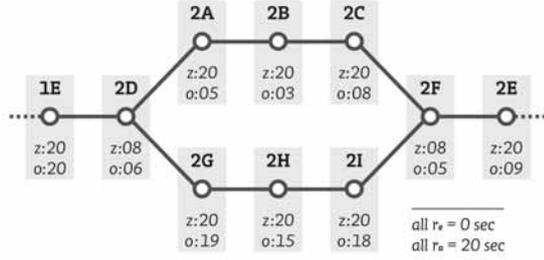


Figure 3: Partial model graph of a parking garage level

3.2 Towards modeling driver behavior

Agents enter the model from one of the entry lane nodes in A_e , and iteratively move from node a_i to node a_j along edge $e(a_i, a_j) \in E$. On any given node $a_i \in A$ the agent, after spending a time of r_{a_i} searching the area for available parking, has to take two decisions: It has to decide whether to park in the current area (parking decision), and, if not, where to go next (routing decision).

To enable the agents to take these decisions, the model considers a number of points:

Basic maneuvering: To avoid infinite loops, an agent administers a counter $v(a_i)$, which represents the number of times an area a_i has been visited by that agent. If an agent always chooses the routing option a_j with the lowest $v(a_j)$, every circle will eventually be broken. In addition to that, as cars are rarely seen to turn on the spot in a parking garage, agents can never move onto the area they just left.

Attractiveness: The model assumes that a driver prefers to park at a spot which is as attractive as possible. This attractiveness might correspond to the spot's distance to a pedestrian exit or vehicular entry. The model therefore assumes an order of attractiveness on a parking garage's areas: $1.0 \geq c(a_{i_1}) \geq \dots \geq c(a_{i_n}) \geq 0.0$ (see figure 4). Agents prefer areas with greater values of $c(a_i)$ to areas with lower attractiveness.

Real-time availability: Drivers also consider real-time availability: if they observe that no spaces are available in an area ahead, they are not attracted to it. Obviously, without technical measures (which are not assumed at this point) the drivers cannot have total knowledge of the current state of the garage, but can look ahead only locally. How far drivers can look

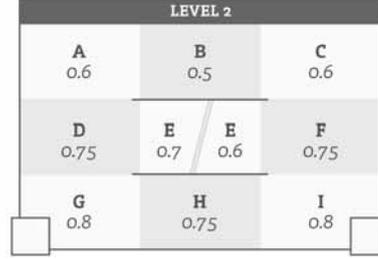


Figure 4: Parking garage level with attractiveness values

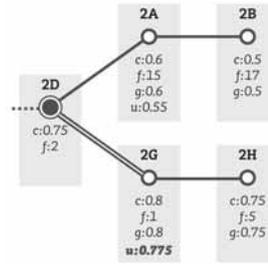


Figure 5: A simplified option tree

ahead is dependent on their individual position and architectural attributes of the garage. This simple model assumes a local look-ahead of only one area: an agent therefore knows the number $f_{a_i}(t)$ of available slots in the current area a_i , and in directly neighboring areas. Therefore we assign a look-ahead set $L_{a_i} \subseteq A$ for any current area a_i .

Based on these considerations, and starting out from the current position a_i as root, an option tree is constructed. This is accomplished by considering iteratively all neighboring areas a_j reachable from a_i via an edge $e(a_i, a_j) \in E$, and from there on succeeding neighbors with a maximum depth of m . In a simple implementation this value might be set to $m = 2$ (see figure 5). The branch starting with the area last visited is removed from the tree, adhering to the no-turn-around rule.

For each element a_j in that tree, a value $g(a_j)$ is calculated: if $a_j \in L_{a_i}$ and $f_{a_j}(t) = 0$ then $g(a_j) = 0$, else $g(a_j) = c(a_j)$. Thus, if the agent observes an area with currently zero available slots, it is not at all attracted to that area. Then for each immediate option a_j a value $u(a_j)$ is calculated by averaging the area's $g(a_j)$ value and its n successors' $a_{j_1} \dots a_{j_n}$ individual $g(a_{j_i})$ values (again see figure 5):

$$u(a_j) = \frac{g(a_j) + g(a_{j1}) + \dots + g(a_{jn})}{n+1} \quad (1)$$

At each simulation step, the agent makes a parking decision, followed by a routing decision if necessary. To take a *parking decision*, it selects the a_g with the maximum $u(a_g)$ out of the current area a_i 's immediate neighbors. If $f_{a_i}(t) > 0$ and $c(a_i) \geq u(a_g)$, the agent decides to park at the current area a_i , else it moves on with the routing decision.

To take the *routing decision*, the agent only considers the options with the least $v(a)$. From these, the agent selects the option with the greatest $u(a_j)$. It moves to that area via the edge $e(a_i, a_j)$, a move that will be completed after a time of $r_{e(a_i, a_j)}$.

If all areas have been visited, i.e. all $v(a) > 0$, and no available parking spot has been found, the agent concludes that the parking garage is full, and stops searching.

3.3 Input and output data

To successfully model a given parking garage's operations on a specific operational day a variety of *input data* has to be available. While part of that data is fundamental for a given parking garage, other elements are dependent on days of the week, special events, etc, which are to be modeled.

Fundamental to a specific parking garage are its layout and configuration, represented in the attributed graph $G(A, E)$. This data can be gathered by studying floor plans, or by endeavoring on a fact finding expedition.

To model a specific operational day the distribution of the number of entering cars over time t has to be available, especially the arrival rate $\lambda_{a_e}(t)$ on each entry lane $a_e \in A_e$. As many parking garages control access with technical means, e.g. license plate reader and inventory systems, this data is often available. It can be complemented, or even replaced, by own measurements.

Additional information is necessary to model local drivers' preferences. The attractiveness value $c(a)$ for each parking garage area $a \in A$ can be established by computational means, e.g. by combining the distances of a spot to the vehicular entry and exit lanes, as well

as to the pedestrian exits, or by interviewing local experts.

There might be more than one of these orders, based on whether the parking garage is used by customers with distinct destinations, e.g. 40% might come to park as close to the supermarket as possible, while 60% might be staff, patients, or visitors of a hospital. These distribution could change over the time of day, e.g. when the hospital might be closed for the evening, but a cinema starts to attract parking visitors. As these classes of preferences are shared by many drivers, only a few different orders of attractiveness might be enough to represent all drivers' intentions for any given garage.

The distribution of parking duration again can be measured with the help of cameras and license plate reader and inventory systems, or by interviewing local experts. The average cruising speed can simply be measured.

The model's *output data* is utilized both as the primary result of the simulation, as well as for validation purposes. The main result necessary to evaluate the effect of parking recommendation methods is of course the distribution of the time spent to search for parking. For validation purposes a variety of other data is interesting, e.g. the number of cars in individual areas over the time of the operational day. These results can easily be compared to real-world observations.

4 A look ahead

Several important points are obviously missing from the described model: A comprehensive model should consider long-term experience and expectations, especially on the state of parking garages at any given time of day. It should also consider availability information and parking recommendation systems (ranging from simple red/green light arrangements at the entry lanes, via computer screens showing the number of free spots on each garage level, to more comprehensive, smart-phone based systems (see [21])). Furthermore, many parking providers offer different classes of parking decals, with some classes having more options than others: at a university campus, administrative and faculty/staff might be allowed to park at any given area, while students might only park at la-

beled student parking. Disabled drivers might have reserved spots available, likewise electric cars. A business park's parking provider might distinguish executive, employee, and visitor parking.

While in this simple model parking and routing decisions are seen as distinct steps, a more comprehensive representation would probably model them as one encompassing decision.

5 Summary and further research

This paper presented a simple agent-based model of cruising for parking in a parking garage. Beyond the parking structure's layout and attributes, the model considers basic navigation, an order of attractiveness on the garage's areas, and local information on current availability. In addition to this, the paper also presented some thoughts on necessary steps to expand and refine the model to make it applicable as a tool for recommendation method evaluation.

Though the model in its current state is still very basic, it has thus been confirmed that discrete modeling techniques, especially the agent-based paradigm, are suitable to model garage parking behavior.

As described, a lot of work still remains to be done: The parking behavior model has to be refined, both sub-models have to be implemented based on an in-house simulation software framework, then calibrated and validated utilizing available data on specific parking garages. After the completion of these steps, the model will be applied to the evaluation of parking recommendation methods.

6 Acknowledgements

This material is partially based upon work supported by the National Science Foundation under grants I/UCRC IIP-1338922, AIR IIP-1237818, SBIR IIP-1330943, III-Large IIS-1213026, MRI CNS-0821345, MRI CNS-1126619, CREST HRD-0833093, I/UCRC IIP-0829576, MRI CNS-0959985, and FRP IIP-1230661, as well as U.S. Department of Transportation under a 2013 TIGER grant.

References

- [1] Asakura, Y., Kashiwadani, M.: Effects of Parking Information on System Performance - A Simulation Model Approach. In: Proceedings of the 1994 Vehicle Navigation and Information Systems Conference, 1994, pp. 251-254.
- [2] Ayala, D., Wolfson, O., Xu, B., DasGupta, B., Lin, J.: Pricing of Parking for Congestion Reduction. Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), Redondo Beach, CA, November 6-9, 2012, pp. 43-51.
- [3] Ayala, D., Wolfson, O., Xu, B., DasGupta, B., Lin, J.: Parking Slot Assignment Games. Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), Chicago, IL, November 1-4, 2011.
- [4] J. Banks, J. S. Carson, B.L. Nelson, D. M. Nicol. Discrete-Event System Simulation, 5th edition. Pearson, 2010.
- [5] Benenson, I. Martens, K., Birfir, S.: PARKAGENT: an agent-based model for parking in the city. In: Computers, Environment and Urban Systems, Vol. 32, No. 6, 2008, pp. 431-439.
- [6] Choi, B.K., Kang, D.H.: Modeling and simulation of discrete-event systems. Wiley, 2013.
- [7] Dieussaert, K., Aerts, K., Steenberghen, T., Maerivoet, S., Spitaels, K.: SUSTAPARK: an agent-based model for simulating parking search. In: Proceedings of AGILE International Conference on Geographic Information Science, 2009.
- [8] Gallo, M., D'Acierno, L., Montella, B.: A multilayer model to simulate cruising for parking in urban areas. In: Transport Policy, Vol. 18, 2011, pp. 735-744.
- [9] Horni, A., Montini, L., Waraich, R. A., Axhausen, K. W.: An Agent-Based Cellular Automaton Cruising-For-Parking Simulation. In: Proceedings of the 13th International Conference on Travel Behaviour Research, Toronto, Canada, July 15-20, 2012.

- [10] Kaplan, S.; Bekhor, S.: Exploring en-route parking type and parking-search route choice: Decision making framework and survey design. In: Proceedings of 2nd International Choice Modelling Conference, 2011.
- [11] Kanev, K., Mirenkov, N., Urata, A.: Parking Simulation and Guidance in a Model Environment. Proceedings of the 2006 Japan-China Joint Workshop on Frontier of Computer Science and Technology (FCST), 2006.
- [12] Kuckertz, P.; Ullrich, O.; Linstädter, A.; Speckenmeyer, E.: Agent based modeling and simulation of a pastoral-nomadic land use system. In: Simulation Notes Europe (SNE), Volume 21, Number 3-4, Dezember 2011, pp. 147-152.
- [13] Levy, N.; Martens, K.; Benenson, I.: Exploring cruising using agent-based and analytical models of parking. In: *Transportmetrica A: Transportation Science*, Vol. 9, No. 9, 2012, pp. 773-797.
- [14] Macal, C. M.; North, M. J.: Tutorial on agent-based modeling and simulation. In: *Journal of Simulation* 4 (2010), pp. 151-162.
- [15] Martens, K.; Benenson, I.: Evaluating Urban Parking Policies with Agent-Based Model of Driver Parking Behavior. In: *Transportation Research Record*, No. 2046, 2008, pp. 37-44.
- [16] Saltzman, R. M.: An Animated Simulation Model for Analyzing On-Street Parking Issues. In: *Simulation*, August 1997, pp. 79-90.
- [17] Sargent, R.G.: Verification and Validation of Simulation Models. In: Proceedings of the 2010 Winter Simulation Conference, 2010, pp. 166-183.
- [18] Shoup, D. C.: *The high cost of free parking*. Planners Press, Vol. 206, 2005.
- [19] Shoup, D. C.: Cruising for parking. In: *Transport Policy* 13 (2006), pp. 479-486.
- [20] Ullrich, O.; Proff, I.; Lückerrath, D.; Kuckertz, P.; Speckenmeyer, E.: Agent-based modeling and simulation of individual traffic as an environment for bus schedule simulation. In: *ITS for Connected Mobility - Selected Papers from the mobil.TUM 2013 International Conference on Mobility and Transport*, Hrsg: Busch, F., Spangler, M., 2014, pp. 89-98.
- [21] Ullrich, O.; Rische, N.: Emissions Reduction in Vehicle Parking. U.S. Patent pending, Application No. 14/660,283, filed March 17, 2015.
- [22] van der Waerden, P., Timmermans, H., Borgers, A.: PAMELA - Parking Analysis Model for Predicting Effects in Local Areas. In: *Transportation Research Record* 1781, Paper No. 02-2078, 2002, pp. 10-18.
- [23] Xu, B., Wolfson, O., Yang, J., Stenneth, L., Yu, P. S., Nelson, P. C.: Real-time Street Parking Availability Estimation. Proceedings of the 2013 IEEE 14th International Conference on Mobile Data Management (MDM). Milan, Italy, June 3-6, 2013, pp. 16-25.
- [24] Young, W., Weng, T. Y.: Data and Parking Simulation Models. In: Kitamura, R., Kuwahara, M. (Ed.): *Simulation Approaches in Transportation Analysis*, Springer Operations Research/Computer Science Interface Series, Vol. 31, 2005. pp. 237-267.

Generische Visualisierung von FMU-basierten Modellen für die interaktive Simulation

Volker Waurich¹, Martin Großer¹, Sebastian Voigt¹

¹Professur für Baumaschinen- und Fördertechnik, TU Dresden
volker.waurich@tu-dresden.de

Standardisierte, offene Modellierungssprachen wie Modelica und Schnittstellenstandards zur Kopplung von Modellen wie FMI ermöglichen die Entwicklung umfassender, virtueller Systeme. Für die Entwicklung von interaktiven Simulatoren werden die physikalischen Modelle mit realen Bedienelementen (bspw. Joystick), visueller Datenaufbereitung (bspw. 3D Animation) und eventuell noch mit einer kinetischen Rückmeldung (Bewegungssystem) gekoppelt. Dies geschieht händisch für jedes System von neuem. Die FMU-basierte Simulation sieht nicht vor, die Modellstruktur oder die Visualisierungsinformationen aus dem Modell bereitzustellen um generisch ein Simulatorszenario bestehend aus physikalischem Modell, dreidimensionaler Animation, Eingabegeräten und weiterführender Datenaufbereitung zu erzeugen. Die vorgestellte Visualisierungssoftware erzeugt die dreidimensionale Animation für eine interaktive FMU. Dazu muss das zugehörige Modelica-Modell vom OpenModelica Compiler übersetzt werden um die Visualisierungsinformationen in Form einer XML-Datei bereitzustellen. Der Artikel beschreibt den aktuellen Stand der Software, diskutiert mögliche Alternativen im Hinblick auf die generische Erzeugung von FMU-basierten Animationen und beschreibt die Nutzung für die Entwicklung von Simulatoren von mobilen Arbeitsmaschinen im Hinblick auf die Optimierung zur Echtzeitfähigkeit.

1 Einleitung

Die Simulation physikalischer Systeme erlaubt es mit modellhaften Abbildungen technischer Systeme, virtuell zu experimentieren. Bei einer interaktiven Simulation nimmt ein Bediener zur Laufzeit Einfluss auf bestimmte Systemgrößen und erhält in entsprechender Aufbereitung Rückmeldung über das Systemverhalten. Bei der Entwicklung von Trainingssimulatoren für mobile Arbeitsmaschinen, wie sie an der TU Dresden erfolgt, bedeutet dies meist die Interaktion mittels Bedienelementen wie Joystick, Pedal und Lenkrad sowie der optischen Rückmeldung in Form einer dreidimensionalen Animation der Maschinen im Einsatz. Dabei muss die Simulationsrechnung in Echtzeit ablaufen. Zur Abbildung des technischen Systems wird ein Modelica-Modell in Form einer FMU (*functional mock-up unit*) eingesetzt. Die FMU errechnet Daten wie Lage und Orientierung der Körper aus dem Mehrkörpersystem. In einer Visualisierungsgebung werden die Animationskörper

durch die errechneten Größen in der Szene bewegt. Die Zuordnung zwischen Transformationsgröße und Modellvariable wird dabei händisch konfiguriert und kann nur unter Kenntnis des Modellaufbaus erfolgen. Die Animation eines mechanischen Modells in Form einer FMU ohne Kenntnis der Modellstruktur ist nur schwer realisierbar. Die automatische Erzeugung einer Animation für eine FMU wäre daher förderlich. Die dafür entwickelte Prozesskette wird im Folgenden vorgestellt.

2 Modelica und FMI

2.1 Modelica als Beschreibungssprache technischer Systeme

Modelica ist ein offener Sprachstandard zur Beschreibung physikalischer Systeme. Über Variablen und Gleichungen wird das Modellverhalten akausal

beschrieben. Akausal bedeutet in diesem Fall, dass die Wirkrichtung im Modellelement noch nicht vorgegeben ist. Die Gleichung kann also noch nach einer anderen Variablen umgestellt werden. Die Überführung der Modellbeschreibung in ein mathematisch lösbares Modell erfolgt über einen ModelicaCompiler. Dieser generiert aus der mathematischen Beschreibung und der Verknüpfung der Modellelemente ein differential-algebraisches Gleichungssystem (DAE-System). Modelica stellt lediglich die textuelle Beschreibung des Systems dar, für die Simulation ist entsprechende Software notwendig. Aufgrund der zunehmenden Etablierung von Modelica für die Modellbeschreibung können mittlerweile eine Vielzahl von domänenspezifischen Bibliotheken herangezogen werden. Auch Mehrkörpersysteme mit Visualisierungsinformation werden in der Modelica Standard Library unterstützt. Die Animation von Mehrkörpersystemen erfolgt jedoch bisher nur in kommerziellen Modelica-Werkzeugen.

2.2 Der FMI Standard zur gekoppelten Simulation

Um ausführbare Simulationsmodelle wie Bausteine auszutauschen und softwareübergreifend zu kombinieren, wurde das Functional-Mockup-Interface FMI entwickelt. Ein übersetztes Modell kann als Functional Mockup Unit (FMU) exportiert werden. Aufgrund der standardisierten Schnittstelle kann die FMU von einer konformen Software simuliert werden, ohne dass das ursprüngliche Modell vorhanden ist. Das hat zum einen den Vorteil, dass Modelica-basierte Systemmodelle mit Modellen aus anderen Simulationswerkzeugen gekoppelt werden können und dass die FMU die Möglichkeit bietet, das Modellverhalten darzustellen ohne den Modellaufbau preiszugeben. Die Systemvariablen werden in der sogenannten `modelDescription.xml` aufgelistet. Somit kann auf einzelne Größen des Modells zugegriffen werden. Prinzipiell existieren zwei verschiedene Arten von FMUs. Zum einen gibt es sogenannte *model exchange* FMUs welche, kurz gefasst, lediglich die Berechnung der Zustandsableitungen für einen Zeitpunkt ermöglichen. Für eine Zeitintegration ist ein externer Löser in die Laufzeitumgebung zu implementieren. Im Gegensatz dazu existieren *co-simulation* FMUs, welche eine komplette Zeitsimulation durch-

führen können, da sie einen eigenen Löser für die Integration bereitstellen. Mit *co-simulation* FMUs können mehrere Teilmodelle gekoppelt und jeweils mit einem geeigneten Löser berechnet werden.

3 Generische Visualisierung

3.1 Stand der Technik zur Visualisierung im Modelica-Umfeld

Die Visualisierung von Modelica-basierten Simulationen ist seit längerem in kommerziellen Modelica-Simulationsumgebungen wie Dymola oder SimulationX verfügbar. Dabei wird schon während der Berechnung die Szene aktualisiert. Nach Beendigung der Simulation ist es möglich die Ergebnisse für eine echtzeitsynchrone Visualisierung zu benutzen. Neben der Visualisierung von Mehrkörpersystemen sind zudem weitere Animationen verfügbar wie beispielsweise die Visualisierung von Rohrleitungssystemen mit farblich kodierter Temperatur oder Druckverteilung und Geschwindigkeitsdarstellungen. In [1] wird vorgestellt, wie solch eine Visualisierung aufbauend auf Dymola und Catia realisiert wurde. SimulationX bietet beispielsweise eine zweidimensionale, dynamische Darstellung von thermischen Zustandsdiagrammen und Sankey-Diagrammen an. Im Bereich der Tool-unabhängigen Visualisierungsmöglichkeiten ist zuerst Modelica 3d von C. Höger zu nennen.[2]. Diese Visualisierung erlaubt über eine Netzwerkverkopplung die Modellvariablen während einer Simulation für die Darstellung mittels OpenSceneGraph zu benutzen. Das verwendete Modelica-Werkzeug spielt dabei keine Rolle da nur standardkonformes Modelica für die Kommunikation mittels externen C-Funktionen benutzt wird. Leider wird Modelica 3d nicht mehr gepflegt und wird auch nicht von freien Modelica-Compilern wie OpenModelica unterstützt. Als besonders umfangreiches Visualisierungswerkzeug ist die DLR Visualization Library zu nennen [3] [4]. Diese umfasst eine eigene Modelica-Bibliothek mit Elementen für dynamische Texturen, Lichtquellen, Kameras, Partikelmengen, Wettereffekte und vieles mehr. Der zugehörige Viewer ist ebenfalls mit der open-source Bibliothek OpenSceneGraph realisiert. Die Möglichkeit einer generischen FMU-Visualisierung existiert nach

Wissen der Autoren bisher nicht.

3.2 Die Visualisierungsspezifikation

Für die Animation eines Modells wird unter anderem die Information benötigt, welche geometrischen Körper auf welche Weise im Raum angeordnet sind. Die Modelica Standard Library bietet für die Animation von Mehrkörpersystemen das Modellelement *Modelica.Mechanics.MultiBody.Visualizers.Advanced.Shape* an. Damit können Geometrieprimitive wie Quader oder Kugeln sowie auch CAD-Körper durch Modellvariablen beschrieben werden. Um die Visualisierungsinformationen, die sich hinter den Objekten verbirgt, der FMU-basierten Simulation zugänglich zu machen, wurde eine zusätzliche Datei erzeugt, die die Visualisierungsdaten für die Modellelemente zusammenfasst. Dazu wird das Modelica-Modell vom opensource Modelica-Compiler OpenModelica eingelesen und es wird eine XML-Datei generiert, die die Informationen über die genannten Objekte bereitstellt. In Folgenden ist ein Auszug aus der Visualisierungs-XML dargestellt.

```

1 <shape>
2   <ident>prismatic.box</ident>
3   <type>box</type>
4   <r>
5     <cref>prismatic.box.r[1]</cref>
6     <cref>prismatic.box.r[2]</cref>
7     <cref>prismatic.box.r[3]</cref>
8   </r>
9   <color>
10    <exp>255.0</exp>
11    <exp>0.0</exp>
12    <exp>0.0</exp>
13  </color>
14 </shape>

```

Elemente wie Positionsvektor r oder Farbe $color$ beinhalten entweder konkrete Werte (`<exp>`) oder Referenzen auf Variablen (`<cref>`), welche aus der FMU ausgelesen werden müssen.

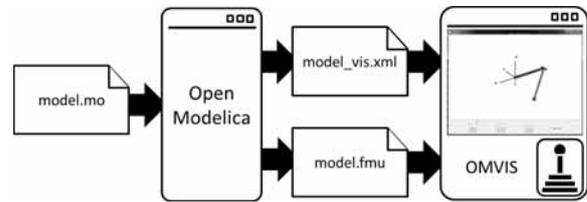


Abbildung 1: Ablaufplan

3.3 Die Visualisierungssoftware OMVis

Wie bereits erwähnt, wurde die Erzeugung der Visualisierungsinformation in OpenModelica implementiert. OpenModelica ist ein quelloffener Modelica Compiler und kann daher einfach um weitere Funktionalitäten erweitert werden. Die FMU kann ebenso durch OpenModelica generiert werden. Um das Visualisierungs-XML zu erzeugen, muss beim Kompilieren das Flag `+d=visxml` angegeben werden. In Abbildung 1 ist der Ablaufplan für eine generische Visualisierung mittels FMU aufgezeigt.

Die Visualisierungssoftware OMVis lädt zunächst die in der XML-Datei angegebenen Geometrien ein. Die Parametrierung der Geometriekörper erfolgt über die zugehörigen Modellvariablen. Momentan wird der FMU 1.0 *model exchange* Standard unterstützt, wobei im Weiteren auch die Erweiterung auf FMU 2.0 und *co-simulation* angestrebt wird. Die FMU wird initialisiert und die Berechnung kann gestartet werden. Da für die Modellsimulation Echtzeitkriterien gelten, wird ein expliziter Zeitintegrator mit fester Schrittweite eingesetzt, wie beispielsweise die explizite Euler-Methode. Die verfügbaren Input-Signale der FMU werden erkannt und können mit Eingabegeräte wie beispielsweise Joysticks angesteuert werden. Die dargestellte Szene ist in OpenSceneGraph implementiert, einer freien Bibliothek zur 3d-Animation. Eingebettet ist der Viewer in Qt, welche die grafische Benutzeroberfläche liefert. Das Erfassen der Eingaben erfolgt mit Hilfe der SDL-Bibliothek. Abbildung 2 zeigt einen Screenshot des Programms OMVis während der interaktiven Simulation eines Baggers.



Abbildung 2: OMVis -Animation des Baggermodells

3.4 Diskussion über FMU-basierte Visualisierungsmöglichkeiten

Grundumfang einer FMU-basierten Visualisierung sollte zunächst die Darstellung von dreidimensionalen Mehrkörpersystemen sein. Diese Animation wird von den meisten Visualisierungswerkzeugen im Modelica-Umfeld angeboten und deckt die Visualisierungsklassen in der Modelica-Standardbibliothek ab. Des Weiteren gibt es mehrere Anwendungen, in denen Rohrleitungen oder Netzwerke zwei- oder dreidimensional dargestellt werden und neben der geometrischen Netzwerkstruktur auch noch Feldinformationen wie Temperatur- Druck oder Geschwindigkeitsverteilungen auf verschiedene Weise kodieren. Die Darstellung solcher Visualisierungsfunktionen ist zwar über die Standardgeometrieprimitiven möglich, jedoch ist eine einfache Parametrierung über neue Bibliothekselemente wünschenswert. Neben den dreidimensionalen Darstellungen sind zweidimensionale Darstellungen ebenso von großer Bedeutung. Die Darstellungen von thermodynamischen oder chemischen Zustandsgrößen in Phasendiagrammen oder der Energiefluss in Sankeydiagrammen sind etablierte Werkzeuge um Systemgrößen kompakt zu präsentieren. Dies sind grundlegende Funktionen, die in einer Visualisierungsspezifikation von FMU-basierten Simulationen auftreten können. Es ist außerdem denkbar weitere Funktionen wie akustische oder kinetische Aktoren zu definieren.

3.5 Möglichkeiten der Spezifikation

Im vorgestellten Visualisierungsprogramm wurde eine Spezifikation in Form einer XML – Datei mit

eigenem Schema gewählt. Im Bereich der Computergrafik gibt es jedoch bereits Standards zur Spezifikation von Animationsmodellen. Extensible 3D oder kurz X3D ist ein offener Beschreibungssprachstandard für 3D Szenen. Er ist im XML Format gehalten und wird hauptsächlich für Web-Anwendungen benutzt. Die Beschreibung ähnelt prinzipiell der bereits umgesetzten XML-Spezifikation jedoch ist dieser Standard auf die Darstellung in einer dreidimensionalen Szene beschränkt. X3D umfasst zusätzlich die Möglichkeit zur Beschreibung von Animationen über Event-Elemente und gleichzeitig auch über zahlreiche Laufzeitumgebungen die die Szenenbeschreibung direkt umsetzen. COLLADA ist ebenso eine XML basierte Beschreibungssprache von 3d-Szenen wobei dieses Format als reiner Datencontainer zu sehen ist. Interaktion und Datenaufbereitung werden individuell gehandhabt. Die Möglichkeit zur Spezifikation von zweidimensionalen Diagrammen oder Anzeigen ist in diesen Standards nicht enthalten. Da für die Animation einer so spezifizierten 3d Szene die Ergebnisse der FMU herangezogen werden müssen, können die bereits vorhandenen Laufzeitumgebungen für die 3d-Formate nicht genutzt werden. Daher ist eine neue Visualisierungsspezifikation im FMU-Standard ratsam. Bisher gibt es jedoch noch keine Konzepte dahingehend. Die vorliegende Arbeit stellt eine Referenzimplementierung für eine XML-Spezifikation dar und soll einen Anstoß für weitere Standardisierung auf diesem Gebiet geben.

4 Simulatorentwicklung von Mobilien Arbeitsmaschinen

4.1 Simulationsmodelle mobiler Arbeitsmaschinen

Mithilfe eines Maschinenmodells lässt sich das Maschinenverhalten kostengünstig, modular und risikoarm abbilden. Gerade im Bereich der mobilen Arbeitsmaschinen ist die Möglichkeit einer virtuellen Simulationsumgebung sinnvoll und lohnend. Ein wichtiger Einsatzbereich ist die Ausbildung von Baugeräteführern an interaktiven Simulatoren [12]. Durch Trainingssimulatoren sind die Auszubildenden in der Lage ohne hohe Miet- und Betriebskosten das Bedienen der Maschinen, auch in Gefahren und

Grenzsituation risikofrei zu üben. Darüber hinaus sind Untersuchungen von Maschinen, Bedienern und Einsatzfällen am Simulator für die Neuentwicklung von Baumaschinen sehr nützlich. Ein Anwendungsfall ist die simulatorgestützte Analyse zur Sichtfelderfassung [13].

Bei diesen Einsatzzwecken ist sowohl ein realistisches Fahrverhalten als auch eine realitätsgetreue Berechnung der physikalischen Zustandsgrößen des mechanischen und hydraulischen Teilmodells notwendig. Die Simulationsmodelle umfassen in der Regel die Modellierung von Fahrtrieb, Hydraulikkreis, Mehrkörpersystem, Reifen, Arbeitsausrüstung und Bodeninteraktion. Aufgrund der domänenübergreifenden Teilsysteme und der Modularität, wird das Modell in *Modelica* implementiert. Das kompilierte Modell wird als FMU exportiert und mit der Visualisierungsumgebung auf der Zielhardware in Echtzeit ausgeführt.

4.2 Echtzeitsimulation

Echtzeitsimulation bedeutet, dass die Simulationsdauer für einen bestimmten Zeitabschnitt Δt schneller erfolgen muss als Δt . Damit geht die Maßgabe einher, dass die Dauer für die Berechnung stets deterministisch sein muss. Das heißt die Ausführungszeit für die Systemgleichungen und die Zeitintegration müssen gleichbleibenden Rechenaufwand haben. Daher sind verschiedene Restriktionen zu beachten:

- Keine zeitdiskreten Zustandsänderungen (Events) mit Eventiteration
- Keine nichtlinearen Systeme mit unbegrenzter Iterationsanzahl
- Keine impliziten Zeitintegrationsverfahren mit unbegrenzter Iterationszahl
- Keine variierenden Schrittweiten oder Verfahrensordnungen bei der Zeitintegration

In der Praxis werden diese Maßgaben jedoch zuweilen aufgeweicht um entweder das Systemverhalten genauer abzubilden oder um die Stabilität der Zeitintegration zu gewährleisten. Dazu werden beispielsweise die möglichen Iterationen für die Berechnung

nichtlinearer Systeme oder die impliziten Löser festgelegt. Maßgeblich ist stets das Echtzeitkriterium.

Abbildung 3 gibt eine Übersicht, welche Möglichkeiten es gibt, die Echtzeitfähigkeit einer Modelica-Simulation herzustellen.

Randbedingungen	Modell	Verfahren
Compiler-optimierung	Vereinfachung	Schrittweiten
Rechentechnik	Steifigkeit reduzieren	Anzahl an Iterationen
Parallelisierung	Nicht-linearitäten	Zeitintegrationsverfahren

Abbildung 3: Möglichkeiten zum Erreichen der Echtzeitfähigkeit

- Die Randbedingungen umfassen die Maßnahmen, die keine (oder nur geringfügige) Auswirkungen auf die Simulationsergebnisse haben. Dazu gehört die Anwendung verschiedener Compileroptimierungen oder sogar der Wechsel eines Compilers. Die Auswertungen der Systemgleichungen kann mittels symbolischer Optimierungsverfahren seitens des Modelica Compilers beschleunigt werden. Zu nennen sind dabei die Behandlung algebraischer Schleifen (*tearing* [5], *reshuffling* [6]), die Vereinfachung von Funktionsaufrufen (*partielle Funktionsauswertung*, *common-sub-expression-elimination*), die Systemverkleinerung durch Auswerten von Parametern oder Output-Only-Berechnung [7]. Bessere Rechentechnik erhöht die Simulationsgeschwindigkeit und ist für Einzelanwendungen oftmals ein adequates Mittel. Für manche Simulationsmodelle lohnt sich auch die Berechnung auf mehreren Prozessoren [8], auf der Grafikkarte oder anderer spezifischer Hardware.
- Die Vereinfachung des Modells geschieht zumeist auf Kosten der physikalischen Abbildungsgüte, kann aber wesentliche Geschwindigkeitsvorteile bringen sofern Nichtlinearitäten reduziert werden. Wird das Modell weniger steif gestaltet, so sind bei der Zeitintegration größere Schrittweiten oder Verfahren geringerer Ordnung möglich. Eine stückweise Linearisierung des Modellverhaltens oder eine Abbildung als *look-up-table* ist ebenfalls denkbar.

- Für die Zeitintegration ist eine Aufteilung in verschiedene Subsysteme für unterschiedliche Schrittweiten oder Lösungsverfahren möglich (*multirate, multimode integration* [9]) sowie das Einsetzen der Zeitintegration in die Systemgleichungen bei gleichzeitiger, symbolischer Optimierung (*inline integration* [10]).

4.3 Bewertung von Echtzeitfähigkeit

Die Erstellung eines komplexen Maschinenmodells für eine Echtzeitanwendung ist stets ein Kompromiss zwischen Detaillierungsgrad und Simulationsgeschwindigkeit. Kann trotz compilerseitiger Optimierung das Echtzeitkriterium nicht eingehalten werden, so muss entweder das Modell vereinfacht oder die Zeitintegration angepasst werden. Modellvereinfachungen haben meist negativen Einfluss auf ein wirklichkeitsnahes Maschinenverhalten. Bei der Wahl eines weniger rechnintensiven Zeitintegrators oder einer größeren Schrittweite besteht die Gefahr, dass das Modell zu ungenau rechnet oder instabil wird [14]. Um die Echtzeitfähigkeit zu erreichen werden die verschiedenen Optimierungsszenarien durchgespielt und evaluiert. Dabei muss stets die Aussagekraft des Modells sichergestellt werden, beispielsweise durch einen Abgleich der Ergebnisse für definierte Referenzszenarien.

Um diesen Iterationsprozess zu beschleunigen, wurde das vorgestellte FMU-Visualisierungswerkzeug entwickelt. Damit kann für verschiedene Modelle automatisch eine Simulatorszenerie generiert werden, ohne bei Modelländerungen Anpassungen an der Visualisierungsumgebung vorzunehmen.

5 Zusammenfassung

Modelica und FMI sind ideale Werkzeuge um Systemverhalten effizient zu modellieren, zu simulieren und mit weiteren Anwendungen zu koppeln. Die interaktive Simulation mit echtzeitsynchroner Visualisierung ist dabei eine häufige Anwendung. Die Umsetzung eines solchen Szenarios auf der Basis von FMUs ist jedoch aufgrund der fehlenden Visualisierungsinformationen innerhalb der FMU mühsam. Die vorliegende Arbeit präsentiert einen Vorschlag für die

Realisierung einer automatischen Visualisierung einer FMU-basierten Simulation. Ein Visualisierungsdatenformat wird mit Hilfe von OpenModelica erzeugt und kann dann im Visualisierungswerkzeug OMVis für eine automatische Animation benutzt werden. Weitere nützliche Darstellungsmöglichkeiten neben der dreidimensionalen Mehrkörpersimulation wurden genannt. Bestehende XML-basierte Spezifikationen aus dem Bereich der Computergrafik sind jedoch für die ingenieurtechnischen Erfordernisse der Darstellung nicht ausreichend. Daher wird die Entwicklung einer Visualisierungsspezifikation im FMU-Standard befürwortet. Mit Hilfe einer generischen FMU-Visualisierungsumgebung wird die Erarbeitung eines echtzeitfähigen Simulationsmodells beschleunigt. Die Anpassungen eines Simulationsmodells für eine Echtzeitanwendung werden schrittweise vorgenommen, bis das Echtzeitkriterium erfüllt ist. Die FMU-basierte Visualisierung ist dabei eine grünstige Möglichkeit.

References

- [1] X. Rémond, T. Gengler und C. Chapius. *Simulation of Piping 3D Designs Powered by Modelica*. Proceedings of the 11th International Modelica Conference, Versailles, Frankreich, 2015
- [2] C. Höger, A. Mehlhase et al. *Modelica3D - Platform Independent Simulation Visualization*. Proceedings of the 9th International Modelica Conference, München, Deutschland, 2012
- [3] T. Bellmann. *Interactive Simulations and advanced Visualization with Modelica*. Proceedings of the 7th Modelica Conference, Como, Italien, 2009.
- [4] M. Hellerer, T. Bellmann und F. Schlegel. *The DLR Visualization Library Recent development and applications*. Proceedings of the 10th International Modelica Conference 2014, Lund, Schweden, 2014
- [5] V. Waurich. *Untersuchung und Bewertung von Tearing-Algorithmen* Diplomarbeit, TU Dresden, 2013
- [6] V. Waurich. *Reshuffling: A Symbolic Pre-processing Algorithm for Improved Robustness*,

- Performance and Parallelization for the Simulation of Differential Algebraic Equations* Proceedings of the 6th International Workshop on Equation-Based Object-Oriented Languages and Tools (EOOLT), Berlin, 2014
- [7] V. Manzoni und F. Casella. *Minimal Equation Sets for Output Computation in Object-Oriented Models*, Proceedings of the 8th International Modelica Conference 2011, Dresden, 2011
- [8] M. Walther, V. Waurich, C. Schubert und I. Gubsch. *Equation based parallelization of Modelica models*, Proceedings of the 10th International Modelica Conference 2014, Lund, Schweden, 2014
- [9] A. Schiela, H. Olsson *Mixed-mode Integration for Real-time Simulation*, Modelica Workshop 2000 Proceedings, Lund, Schweden, 2000
- [10] H. Elmqvist, M. Otter und F.E. Cellier *A New Mixed Symbolic/Numeric Approach for Solving Differential-Algebraic Equation Systems*. European Simulation Multiconference, 1995
- [11] P. Hoske, G. Kunze, K. Bürkle, M. Schmauder, M. Brütting, Ch. Böser *Interaktiver Simulator für mobile Arbeitsmaschinen - Virtuelle Prototypen im Einsatzkontext erleben* Tagungsband zur Konferenz „Entwerfen - Entwickeln - Erleben“, Methoden und Werkzeuge in Produktentwicklung und Design, Dresden, 2012."
- [12] Forschungsvorhaben AWIMAS Schlussbericht Konsortium *Aus- und Weiterbildung von Maschinenführern für die Bauindustrie durch interaktive Maschinensimulationen in virtuellen Umgebungen (AWIMAS)*, 2015
- [13] Abschlussbericht zum DFG-Projekt "DYNAMISCHT" *Methode zur dynamischen Sichtfelderfassung und -bewertung für mobile Arbeitsmaschinen unter Einbeziehung der Prozess- und Nutzercharakteristik*, 2012
- [14] S. Miller und J. Wendlandt *Real-Time Simulation of Physical Systems Using Simscape*, The MathWorks, 2011

Verschaltung von Functional-Mockup-Units (FMUs) mit physikalischen Konnektoren am Beispiel der thermischen Simulation eines Supermarktes

W. Tegethoff¹, J.-N. Jäschke¹, C. Schulze¹, N. Fidorra²

¹TLK-Thermo GmbH, Hans-Sommer-Straße 5, 38106 Braunschweig

²TU Braunschweig, Institut für Thermodynamik, Hans-Sommer-Straße 5, 38106 Braunschweig

w.tegethoff@tlk-thermo.de

Um die Energieeffizienz eines Supermarktes zu steigern, ist eine Simulation des gesamten thermischen Systems sinnvoll. Da dieses aus einer Vielzahl von miteinander interagierenden Teilmodellen aus unterschiedlichen physikalischen Domänen (Gebäude, Kühlmöbel, Kälteanlagen etc.) besteht, werden unterschiedliche Werkzeuge für deren Modellierung und Simulation verwendet. Um alle Modelle des gesamten Systems gemeinsam und unabhängig von der Modellierungsumgebung simulieren zu können, bietet sich die Verwendung des Functional-Mockup-Interface (FMI) an. Hierbei ergibt sich allerdings folgendes Problem: Der FMI-Standard sieht eine Kopplung von Modellen lediglich über Einzelverbindungen vor, was bei steigender Komplexität des Gesamtsystems zu Unübersichtlichkeit führt. Insbesondere physikalische Transportprozesse werden über mehrere Größen beschrieben, wobei ein bidirektionaler Signalfluss möglich sein kann.

Aus diesem Grund werden im Rahmen dieser Arbeit benutzerdefinierte Mehrfachverbindungen, die eine bidirektionale Gruppierung von Einzelverbindungen ermöglichen und insofern eine sinnvolle Erweiterung des FMI-Standards sein können, vorgestellt.

Da diese Neuentwicklung die Schaffung weiterer Modellbibliotheken und hierarchischer Systeme begünstigt, wird im vorliegenden Artikel auch ein Beschreibungsformat für Modelle, beispielsweise FMUs, vorgestellt. Es basiert auf den objektorientierten Mechanismen von Vererbung und Polymorphie, was die Definition von Schnittstellen und somit Typenkompatibilität ermöglicht. Ebenfalls können Anforderungen an einzelne Modelle (z.B. Betriebsräume) gestellt werden.

1 Einleitung

Der FMI-Standard („Functional Mockup Interface“) wurde zum Zwecke der toolunabhängigen Simulation von Modellen aus verschiedenen, zueinander inkompatiblen Quellen entwickelt [FMI]. Der Export eines Modells als FMU („Functional Mockup Unit“) ermöglicht ein Laden und Weiterverwenden in unterschiedlichen Simulationsumgebungen (z.B. Simulink, Dymola) oder anderen Sprachen (z.B. Python). Eine Liste kompatibler Simulationsumgebungen ist unter [FMI] zu finden. Der FMI-Standard definiert die Schnittstellen, anhand derer die Kommunikation von Modellen stattfinden kann, wobei diese auf den Transfer von Daten elementarer Datentypen beschränkt ist und somit lediglich eine signalorientierte Sichtweise erlaubt. Eine Verbindung mit Hilfe physikalischer Mehrfachverbindungen, wie sie bei gleichungsorientierten Sprachen wie beispielsweise Modelica möglich ist, wird im

FMI-Standard bisher nicht unterstützt. In den folgenden Abschnitten werden Konzepte vorgestellt, die die Nutzung physikalischer Mehrfachverbindungen für FMUs ermöglichen sollen.

Die Verwendung physikalischer Mehrfachverbindungen stellt eine wichtige Basis für ein intuitives Schnittstellenkonzept zur Simulation von Systemen dar. Hierauf können auch neue FMU-Modellbibliotheken aufbauen. Die Einführung objektorientierter Konzepte unterstützt die Nachhaltigkeit dieser Bibliotheken. In objektorientierten Sprachen können die in Basisklassen angelegten Schnittstellen weitervererbt und somit Modelle mit identischen Schnittstellen aber unterschiedlichem mathematischen Inhalt erzeugt werden (Polymorphie). Im Folgenden wird auch hierzu ein Vorschlag unterbreitet. Ziel dieses Artikels ist es, einen Diskussionsbeitrag zur Benutzung von FMUs sowie einer möglichen Erweiterung des FMI-Standards zu

leisten. Die hier vorgestellten Überlegungen sind unabhängig von der verwendeten Software zur Verschaltung und Simulation von FMUs zu sehen und können prinzipiell mit anderen Umgebungen, die den Import von FMU Modellen erlauben, umsetzen.

Dennoch sollen die geschaffenen Möglichkeiten der Verwendung von physikalischen Verbindungen sowie der objektorientierten Beschreibung von Modellen anhand eines konkreten Beispiels vorgestellt werden. Hierzu wird ein Supermarkts als thermisches System mittels der vorgestellten Ideen/Konzepte beschrieben und mit Hilfe einer Simulationsumgebung der TLK-Thermo GmbH im Sinne eines „Proof of Concept“ umgesetzt.

2 Objektorientierte Modellbeschreibung

Die Verschaltung von Modellinstanzen zu einem Simulationssystem erfolgt für das in diesem Artikel genutzte Beispiel eines Supermarkts unter Verwendung einer von der TLK-Thermo GmbH entwickelten Simulationsumgebung. Diese bietet die Möglichkeit, Modelle aus unterschiedlichen Quellen wie Simulink, FMI (Co-Simulation und Modelexchange), Python oder C/C++ miteinander zu koppeln. Ein Modell wird als hybrides DAE (Differential Algebraic Equation), also als Algebrodifferentialgleichungssystem mit Eventunterstützung interpretiert. Für jeden unterstützten Modelltypen (FMU, Simulink, Python, ...) sind spezialisierte DAEs vorhanden. Auf diesen gekoppelten DAEs können anschließend verschiedene Operationen wie beispielsweise Simulation oder Optimierung durchgeführt werden. Ebenfalls kann ein System wiederum als FMU exportiert und somit in anderen Umgebungen verwendet werden.

Bilanzkorrektur

Die Simulationsumgebung der TLK-Thermo GmbH unterstützt die zeitkontinuierliche (z.B. FMU für Modell Exchange) und die zeitdiskrete Simulation (z.B. FMU für Co-Simulation). Bei der zeitkontinuierlichen Simulation sind aufgrund der exakten Formulierung und Lösung der gekoppelten DAE Systeme genauere Ergebnisse zu erwarten als bei der zeitdiskreten Simulation. Bei der zeitdiskreten Simulation von Energiesystemen kann es aufgrund des zeitdiskreten Informationsaustauschs zwischen den Modellen zu Verletzungen von Erhaltungssätzen, etwa für Energie, Masse oder Impuls kommen. Eine Extrapolation der Signale

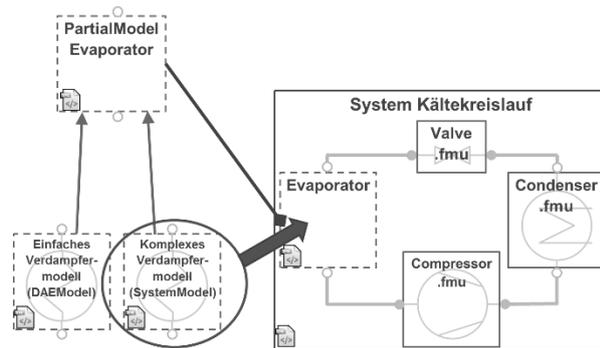


Abb. 1: Verschaltung eines Kältekreislaufs mit abstrakten Modellbeschreibungen und physikalischen Verbindungen. Durch Verwendung eines abstrakten Modells wird Polymorphie ermöglicht.

reicht i.A. nicht aus um dies zu verhindern. Um die integrale Erfüllung der Erhaltungssätze zu verbessern, können beispielsweise auf Signalempfängerseite PID-Glieder zum Einsatz kommen, die das Signal für Energie-, Massen- oder Impulsstrom aufbereiten und Fehler in den Bilanzen herausfiltern. Bei Extrapolation nullter Ordnung ist eine Gewichtung des Proportionalanteils von 1, des Integralanteils von 0 und des Differentialanteils von der Hälfte des Quadrats der Koppungsschrittweite sinnvoll.

Modellbeschreibungen

Zur Steigerung der Effizienz bei der Verwendung von Modellen und deren Verschaltung zu Systemen wird ein objektorientiertes Beschreibungsformat eingeführt, welches aus vier verschiedene Beschreibungstypen besteht. Drei dieser Typen entsprechen dem aus der Objektorientierung bekannten Begriff der Klasse.

Das *PartialModel* stellt eine Klasse dar und beschreibt ein abstraktes Modell. Innerhalb dieses Beschreibungstyps kann die externe Schnittstelle, also Modellein- und -ausgänge sowie Konnektoren (s. Kap. 3) definiert werden. Weiterhin können interne Modellvariablen inklusive Betriebsbereiche angegeben werden. Es kann durch Angabe weiterer Informationen eine Vererbungshierarchie ermöglicht werden. Es wird jedoch nicht festgelegt, um was für einen Modelltyp es sich handelt, weshalb ein Modell dieses Typs nicht simulierfähig ist.

Das *DAEModel* stellt eine Erweiterung des *PartialModels* dar und entspricht somit ebenfalls einer Klasse. Durch die zusätzliche Information des Modelltyps wird die Simulierfähigkeit sichergestellt. Der dritte Beschreibungstyp *SystemModel* erweitert ebenfalls das *PartialModel* und wird verwendet, um ein System

aus mehreren instanziierten Modellen zu erstellen. Neben den über den Beschreibungstyp *ModellInstance* (s.u.) beschriebenen Modellinstanzen enthält ein *SystemModel* eine Liste aus Verbindungen, durch die eine Interaktion zwischen Modellinstanzen und an die Systemgrenze ermöglicht wird. Ein *SystemModel* ist simulierfähig, wenn alle Modellinstanzen simulierfähig sind. Eine *ModellInstance* stellt die Instanziierung eines der vorher genannten Beschreibungstypen dar und beinhaltet neben einem eindeutigen Bezeichner einen Verweis auf die zu instanziiierende Modellbeschreibung. Optional wird durch Angabe eines *constraining type* Polymorphie ermöglicht. Jede Modellbeschreibung, die in ihrer Vererbungshierarchie den *constraining type* enthält, kann in der Instanz eingesetzt werden. Weiterhin kann in einer Modellinstanz der Inhalt der Modellbeschreibung über Modifizierer geändert werden. Die Simulierfähigkeit einer *ModellInstance* ist gegeben, wenn das instanziierte Modell vom Typ *DAEModel* oder *SystemModel* ist.

Durch Verwendung dieser Modellbeschreibungen lassen sich Modelle aus unterschiedlichen Quellen effizient miteinander kombinieren. Über Polymorphie können kompatible Modelle ausgetauscht werden, ohne die Systemstruktur zu verändern. Auf diese Weise lässt sich beispielsweise der Einfluss eines einzelnen Modells auf das Gesamtsystem effizient und schnell untersuchen. Ebenfalls können durch unterschiedliche Parametrisierung eines einzelnen Modells über mehrere Modellbeschreibungen mehrere physikalische Modelle dargestellt werden. Dies wird in Kapitel 5 näher erläutert.

3 Einführung von physikalischen Verbindungen

Die beschriebene Kombination von Modellen zu physikalischen Systemen ist mit dem aktuellen FMI-Standard nur eingeschränkt möglich und wurde bereits u.a. bei [Otter2014] thematisiert. Der FMI-Standard sieht zur Kommunikation zwischen Modellen während der Simulation lediglich Modellein- und -ausgänge der elementaren Datentypen *Real*, *Integer*, *Boolean* und *String* vor. Dies entspricht einer signalorientierten Sichtweise, wie sie beispielsweise in der Regelungstechnik oder auch in Simulink üblich ist. Die Modellierung von physikalischen Systemen beinhaltet jedoch oftmals Transportprozesse, für deren Beschreibung mehrere Größen benötigt werden, deren Signalflussrichtungen sich unterscheiden können (vgl. Tabelle 1).

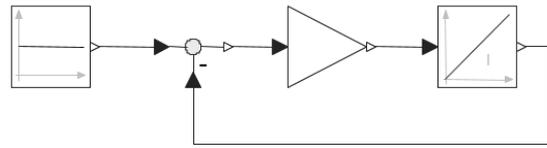


Abb. 2: Beispielsystem mit signalorientierter Verschaltung (Simulink). Physikalische Zusammenhänge sind nicht sofort ersichtlich.

Gleichungsorientierte Modellierungssprachen wie beispielsweise Modelica bieten die Möglichkeit der physikalischen Verbindung, die alle Größen eines Transportprozesses zusammenfasst. Diese Verbindung stellt dann nicht nur den Signalfluss dar, sondern repräsentiert

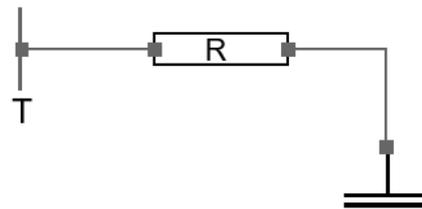


Abb. 3: Beispielsystem mit physikalischen Verbindungen (Modelica). Die physikalische Semantik des Systems bleibt erkennbar.

tiert gleichzeitig auch reale physikalische Verbindungen. Die Übersichtlichkeit des Systems wird gesteigert und die physikalischen Transportprozesse bleiben sichtbar. Abb. 2 und Abb. 3 verdeutlichen den Unter-

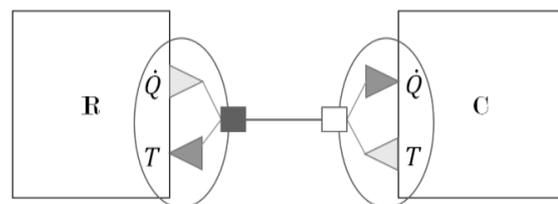


Abb. 4: Darstellung der Signalflussrichtungen für die Übertragung eines Wärmestroms.

schied zwischen signalorientierter und physikalisch motivierter Sichtweise anhand eines einfachen thermischen Netzwerks. Um solche physikalisch motivierten Mehrfachverbindungen auch für Modelle innerhalb der Simulationsumgebung zu ermöglichen, wird ein Datenformat eingeführt, das Mehrfachverbindungen definiert. Diese Definition erfolgt durch die Beschreibung sämtlicher Einzelverbindungen anhand von Signalflussrichtung, Datentyp und eindeutigem Namen. Daraus lassen sich zwei komplementäre Konnektortypen ableiten, die eine charakteristische Kombination aus Ein- und Ausgängen besitzen, die sich aus der Signalflussrichtung der jeweiligen Einzelverbindung ergeben.

Tabelle 1: Darstellung von Zustandsgrößen und strömenden Erhaltungsgrößen für verschiedene physikalische Transportprozesse.

	Strömende Erhaltungsgröße	Zustandsgröße	Streamgröße
Elektrotechnik	Stromstärke I	Spannung U	
Thermodynamik	Wärmestrom \dot{Q}	Temperatur T	
Thermofluid- dynamik	Massenstrom \dot{m}	Druck p	Spez. Enthalpie h , Gemischzusammensetzung ξ
Mechanik	Kraft F	Ort x , Geschwindigkeit v	

Ein Konnektor stellt einen Endpunkt einer Mehrfachverbindung dar und benötigt zu seiner eindeutigen Beschreibung dementsprechend Informationen zu der Anzahl, den Namen und den Datentypen aller Ein- bzw. Ausgänge.

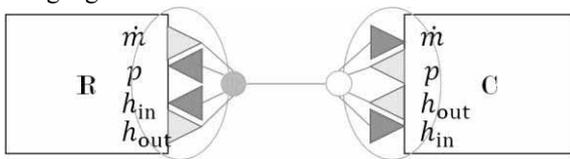


Abb. 5: Darstellung der Signalflussrichtungen für die Übertragung eines Kältemittels. Die spez. Enthalpie als konvektiv übertragene Größe wird in beide Richtungen übertragen.

Die beiden komplementären Konnektortypen, die gemeinsam Anfangs- und Endpunkt einer Mehrfachverbindung darstellen, stimmen in Anzahl, Name und Datentyp der Ein- bzw. Ausgänge überein. Für die Erhaltungs- und Zustandsgrößen gilt jedoch, dass sie für beide Typen unterschiedliche Signalflussrichtungen haben müssen. Dies bedeutet, dass die Erhaltungsgröße in einem Konnektortyp ein Ausgang ist, während sie im komplementären Konnektortyp einen Eingang darstellt. Für die Zustandsgröße gilt es entsprechend (vgl. die Ein- und Ausgänge in den Abb. 4 – 6).

Im Rahmen dieser Arbeit werden physikalische Verbindungstypen für die Domänen Thermodynamik und Thermofluidodynamik entsprechend Tabelle 1 eingeführt. Bei dem gewählten Ansatz eines sogenannten Widerstand-Kapazitäts-Modells verlaufen die Signalflussrichtungen von strömender Erhaltungsgröße und Zustandsgröße grundsätzlich entgegengesetzt (vgl. u.a. [Jäschke2015], [Scharff2012]). Im Speichertherm „Kapazität“ wird die Zustandsgröße (Temperatur bzw. Druck) berechnet, während im Transportmodell der „Widerstand“ die strömende Erhaltungsgröße (Wärme- bzw. Massenstrom) bestimmt. Die Konnektoren, die Anfangs- und Endpunkt einer

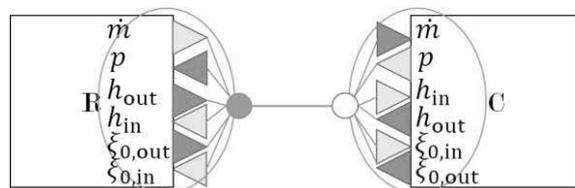


Abb. 6: Darstellung der Signalflussrichtungen für die Übertragung von feuchter Luft. Diese wird als Gemisch mit zwei Komponenten betrachtet, weswegen diese Verbindung zusätzlich zur spez. Enthalpie ebenfalls die Gemischzusammensetzung als konvektive Größe überträgt.

Mehrfachverbindung zwischen einem Widerstands- und einem Kapazitätsmodell darstellen, müssen also komplementär zueinander sein.

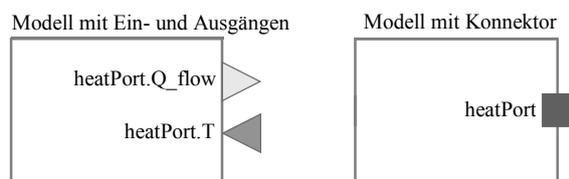


Abb. 7: Schematische Darstellung eines Modells mit physikalischem Konnektor. Entsprechen Ein- und Ausgänge des Modells der geforderten Syntax, können sie mit Hilfe der Modellbeschreibungen durch einen Konnektor ersetzt werden.

In Abb. 4 ist für die Domäne Thermodynamik der Verbindungstyp zur Beschreibung eines Wärmestroms dargestellt. Der Verbindungstyp zur Übertragung eines Kältemittelstroms wird in Abb. 5 dargestellt. Die Abbildung des Transportvorgangs der konvektiv mitgeführten spezifischen Enthalpie h wird analog zu dem in Modelica verwendeten Streamkonzept durch Verbindungen in beide Richtungen realisiert (vgl. [Franke2009]). Zur Abbildung des Transportvorgangs von feuchter Luft wird dieser Verbindungstyp um die Übertragung der Gemischzusammensetzung für ein binäres Gemisch ξ_0 erweitert (Abb. 6).

Die Verwendung von physikalischen Konnektoren setzt voraus, dass die Ein- und Ausgänge eines Modells einer passenden Syntax entsprechen. Diese beinhaltet den Namen der Konnektorinstanz, gefolgt von einem Punkt und dem Namen der Einzelverbindung (vgl. Abb. 7). Konnektoren können einem Modell über die Modellbeschreibungen (vgl. Kap. 2) hinzugefügt und anschließend innerhalb der Simulationsumgebung der TLK-Thermo GmbH verwendet werden.

4 Supermarkt als thermisches System

Das beschriebene Konzept der physikalischen Konnektoren wird für die Modellierung eines Kühlmöbels angewendet. Im Folgenden wird aufgezeigt, welche physikalischen Interaktionen dabei auftreten und wie sich diese durch signalorientierte Verbindungen und physikalische Konnektoren darstellen lassen.

Kühlmöbel dienen im Supermarkt der Lagerung und Präsentation von Kühlgut im Verkaufsraum. Aufgrund der tendenziell höheren Temperatur im Supermarktinnenraum findet ein Wärmeeintrag in das Kühlmöbel statt, sodass ihm gleichzeitig Wärme entzogen werden muss, um das gewünschte, niedrige Temperaturniveau halten zu können. Die dafür notwendige Kälteleistung kann mit einer zentralen Kälteanlage oder einer integrierten Kälteanlage bereitgestellt werden. Letztere werden hier nicht weiter behandelt. Stattdessen soll ein Kühlmöbel mit Anschluss an eine zentrale Verbundkälteanlage betrachtet werden. Diese sind mit einem Wärmeübertrager ausgestattet, in dem ein Kältemittel

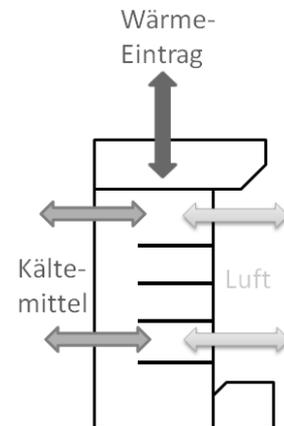


Abb. 8: Wärme- und Stoffströme in der thermischen Simulation von Supermärkten am Beispiel eines Kühlmöbels

auf niedrigem Druckniveau verdampft. Das Kältemittel wird durch Leitungen zur Kälteanlage zurückgeführt, wo es mit Kompressoren verdichtet wird und auf hohem Druckniveau seine Abwärme an die Umgebung abgibt. Anschließend wird es auf niedrigem Druckniveau wieder zum Kühlmöbel zurückgeführt (vgl. Abb. 10).

Obwohl es eine Vielzahl von Quellen bzw. Mechanismen gibt, durch die dem Kühlmöbelinnenraum Wärme zugeführt wird, dominiert der Austausch feuchter Luft sowie Wärmeleitung über die Kühlmöbeloberfläche. Der Austausch von Wärme und Stoffströmen über die Bilanzgrenzen des Kühlmöbels wird in Abb. 8 dargestellt: Es werden ein Wärmestrom, zwei Luft- und zwei Kältemittelströme ausgetauscht.

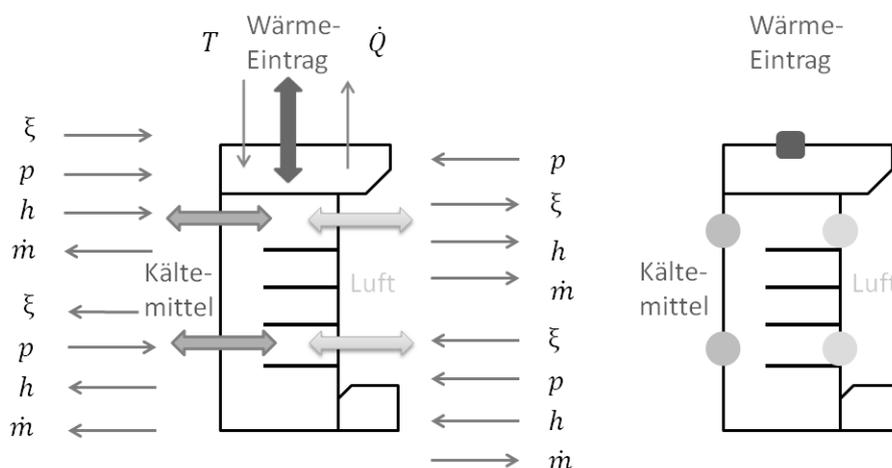


Abb. 9: Vergleichende Darstellung eines Kühlmöbelmodells in der signalorientierten (links) und der physikalisch motivierten Sichtweise (rechts), wobei deutlich wird, dass die Anzahl der benötigten Verbindungen zur Kommunikation dieses Modells mit anderen Modellinstanzen drastisch gesenkt wird und physikalische Transportprozesse weiterhin sichtbar bleiben.

Informationsaustausch

Um den Austausch eines Luftstroms über eine Bilanzgrenze zwischen zwei Modellen beschreiben zu können, sind, wie im vorherigen Abschnitt (Kap. 3) ausgeführt, Informationen zum Druck, zum Massenstrom, zur spezifischen Enthalpie und zur Zusammensetzung notwendig. Der Luftvolumenstrom, den ein Kühlmöbel mit dem Supermarkttinnenraum austauscht, hängt im Wesentlichen von seiner Geometrie und Bauart (mit/ohne Tür, mit/ohne Luftschleier etc.) ab. Da diese Informationen Teil des Kühlmöbelmodells sind, bietet es sich an, den Luftmassenstrom von diesem Modell berechnen zu lassen. Aus diesem Grund wird folgende Konvention für die Orientierung des bidirektionalen Informationsaustausches für den Luftstrom gewählt:

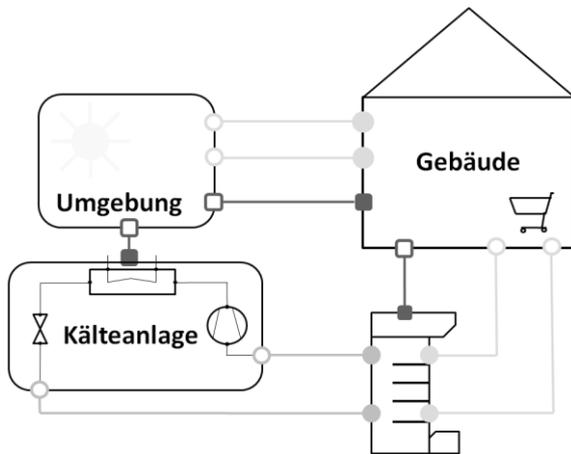


Abb. 10: Schematische Darstellung eines Supermarkts als thermisches Gesamtsystem bestehend aus Gebäude-, Kühlmöbel-, Kälteanlagen- und Umgebungsmodell

Das Kühlmöbelmodell erhält die Zustandsgröße (Druck) von einem Kapazitätsmodell, was in diesem Fall durch das Innenraummodell des Supermarktes dargestellt ist und berechnet die strömende Erhaltungsgröße (in diesem Fall den Massenstrom). Diese wird über den Konnektor aus der Bilanzgrenze hinausgeführt. Diese Konvention ist in Abb. 9, rechts analog zu Abb. 6 durch die ausgefüllten runde Konnektorsymbole gezeigt. Neben dem Luftaustausch ist auch der Wärmestrom, der über das Kühlmöbelgehäuse tritt, ein relevanter Teil des Wärmeeintrages, der dem Kühlmöbel von der Kälteanlage wieder entzogen werden muss. Dieser in Abschnitt 3 beschriebene Verbindungstyp wird in Abb. 9 durch rechteckige Konnektorsymbole dargestellt. Ähnlich wie der Luftstrom hängt der Wärmestrom im Wesentlichen von Geometrie und Material des Kühlmöbels ab. Die Konvention für die

Orientierung des bidirektionalen Informationsaustausches für den Wärmestrom ist so gewählt, dass das Kühlmöbelmodell die Potentialinformation (Temperatur) vom Innenraummodell des Supermarktes erhält und die Information über die strömende Erhaltungsgröße (Wärmestrom) zurückgibt. Dies ist in Abb. 10 analog zu Abb. 4 durch ein ausgefülltes Konnektorsymbol am Kühlmöbel dargestellt.

Für den Austausch eines Wärmestroms über eine Bilanzgrenze zwischen zwei Modellen sind Informationen zur Zustandsgröße (Temperatur) und zur strömenden Erhaltungsgröße (Wärmestrom) notwendig.

Die beschriebenen, signalbasierten Verbindungen für ein Kältemittel, feuchte Luft bzw. einen Wärmestrom können demnach durch physikalisch motivierte Mehrfachverbindungen abgebildet werden. Damit ergibt sich für jeden ausgetauschten Wärme- oder Stoffstrom ein physikalischer Konnektor, wie in Abb. 9 auf der rechten Seite dargestellt.

Supermarkt als Gesamtsystem / Proof of Concept

Basierend auf dem beschriebenen Konzept des Informationsaustauschs zwischen Modellen können diese zur Abbildung von Systemen kombiniert werden. Im Folgenden wird ein einfaches Supermarktsystem vorgestellt, welches als „Proof-of-concept“ mittels FMUs simuliert wird. Das Supermarktsystem besteht aus einem Gebäude, einem Kühlmöbel, einer Kälteanlage sowie der Umgebung. Die Verschaltung wird durch eine Modelbeschreibung vom Typ *SystemModel* realisiert, in der Modelle für Umgebung, Gebäude, Kühlmöbel und Kälteanlage instanziiert sind, wobei jeweils ein *PartialModel* als *constraining type* vorgegeben ist. Diese enthalten die äußere Schnittstelle und ermöglichen die Festlegung der Interaktion der Modellinstanzen unabhängig von der jeweils gewählten Modelldescription. Dadurch wird die Typenkompatibilität sichergestellt.

Für dieses Beispiel wurden Modelle in Form von FMUs verwendet, die unter Dymola [Dymola] in Modelica erstellt wurden und auf der TIL-Bibliothek [TIL] der TLK-Thermo GmbH basieren. Das Gebäude besteht aus einem dynamischen modellierten Luftvolumen und einer Gebäudehülle, die als ein thermisches Netzwerk modelliert ist. Der Kältekreislauf verwendet eine modulare Struktur, ähnlich der TIL-Bibliothek, basiert jedoch auf stationären Modellen der sich im Aufbau befindlichen „SuperSmart“-Bibliothek [Fidorra2014a].

Insgesamt besteht der gewählte Supermarkt aus vier FMUs, die über insgesamt neun physikalische Verbindungen verbunden sind, von denen vier die Übertragung feuchter Luft abbilden, drei die Übertragung eines Wärmestroms und zwei den Transport eines Kältemittels.

Bei einer signalorientierten Modellierung wären insgesamt 38 Einzelverbindungen nötig. Daraus ergibt sich eine Reduzierung der Anzahl der sichtbaren Verbindungen auf ein Viertel, was die Übersichtlichkeit deutlich erhöht. Des Weiteren bleiben physikalische Zusammenhänge ersichtlich (vgl. Abb. 10). Das Gesamtsystem besitzt 12 differenzielle Zustände und keine algebraischen, iterativen Abhängigkeiten. Die Simulation eines deutlich größeren Systems, welches mehr Modelle und mehr differentielle und algebraische Zustände beinhaltet, wäre ebenfalls möglich und ist bereits erfolgreich durchgeführt worden.

Simulation

Mit dem gewählten Supermarktsystem wurde exemplarisch ein Tag simuliert. Hierbei wurde nur die Umgebungstemperatur stundenweise variiert.

Das Gebäudemodell enthält eine Heizung, die den Heizwärmestrom berechnet, der notwendig ist, um eine gewünschte Innentemperatur zu halten.

Abb. 11 zeigt den Verlauf der Umgebungstemperatur, der Innenraumtemperatur des Gebäudes sowie den benötigten Heizwärmestrom. Es ist zu sehen, dass bei höheren Umgebungstemperaturen geringere Wärmeverluste auftreten und somit ein geringerer Heizwärmestrom benötigt wird. Abb. 12 zeigt den Verlauf der elektrischen Verdichterleistung der Kälteanlage. Bei höherer Umgebungstemperatur steigt die elektrische Verdichterleistung. Um Wärme abgeben zu können,

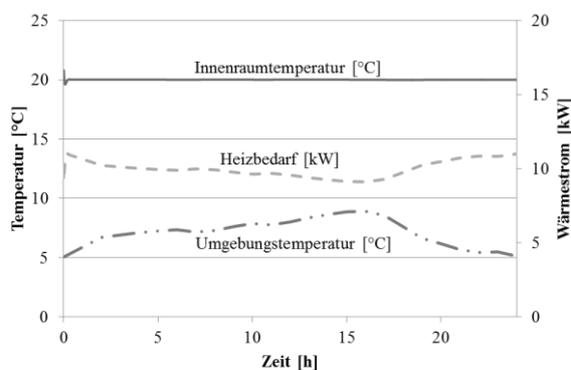


Abb. 11: Zeitlicher Verlauf der Umgebungs- und Gebäudeinnenraumtemperatur, sowie des Heizbedarfs

muss die Kältemitteltemperatur über der Umgebungstemperatur liegen. Bei höheren Kältemitteltemperaturen sind daher höhere Drücke notwendig, was einen höheren Leistungsbedarf für die Verdichtung mit sich führt.

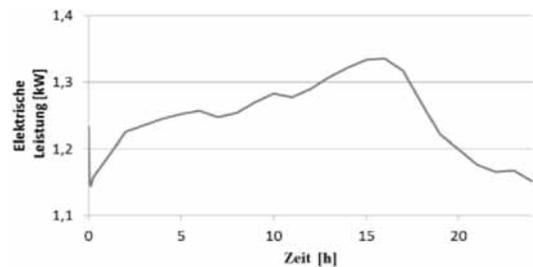


Abb. 12: Zeitlicher Verlauf der elektrischen Verdichterleistung der Kälteanlage

Die vorgestellten Berechnungen wurden sowohl in der Simulationsumgebung der TLK-Thermo GmbH als auch mit den Modelica-basierten Modellen in Dymola durchgeführt. Die Ergebnisse sind identisch, jedoch gibt es Unterschiede in Hinblick auf die Simulationsdauer. Die Simulationsdauer unter Dymola ist für dieses nicht-optimierte System in etwa halb so lang wie in der im Aufbau befindlichen Simulationsumgebung der TLK-Thermo GmbH. Weiterhin gibt es Unterschiede hinsichtlich der Anzahl der Integrationschritte und Funktionsauswertungen. Dies kann mit den unterschiedlichen Lösern der beiden Simulationsumgebungen erklärt werden. In Dymola wurde der Standardsolver DASSL (Differential-Algebraic System Solver) [Petzold1982] genutzt, während in diesem Beispiel in der Simulationsumgebung der TLK-Thermo GmbH der Sundials CVode [Sundials] als Löser verwendet wurde

5 Zusammenfassung und Ausblick

Der vorliegende Artikel behandelt die Nutzung von physikalischen Mehrfachverbindungen sowie die Erstellung eines Beschreibungsformats zur Ermöglichung von objektorientierter Strukturen (Vererbung, Polymorphie) zum Zwecke der Erstellung nachhaltiger Modellbibliotheken beispielsweise auf Basis von FMU-Modellen.

Die Vorteile der Verwendung von physikalischen Konnektoren gegenüber signalorientierten Verbindungen werden aufgezeigt und exemplarisch für den Austausch von Wärme- und Stoffströmen dargestellt.

Ein neu entwickeltes, objektorientiertes Beschreibungsformat ermöglicht die Definition von Schnittstellen für einzelne Modelle. Diese können Ein- und

Ausgänge sowie Modellvariablen inklusive Betriebsbereiche enthalten. Außerdem können Konnektoren von physikalischen Verbindungen hinzugefügt werden. Das abstrakte Beschreibungsformat ermöglicht die Anwendung polymorpher Prinzipien nicht nur auf einzelne Modelle wie beispielsweise FMUs, sondern auch auf das gesamte System. Dadurch wird es möglich, innerhalb des Systems ein einzelnes Modell auszutauschen, ohne dass sich dadurch die Systemstruktur ändert.

Ein denkbare Anwendungsfeld ist die Simulation von Supermärkten, z.B. zur Steigerung der Energieeffizienz oder zur Evaluierung neuer Konzepte der Kältebereitstellung. Im vorliegenden Artikel wird exemplarisch aufgezeigt, wie die realen, physikalischen Interaktionen eines Kühlmöbels innerhalb eines Supermarkts durch signalorientierte bzw. durch physikalische Verbindungen dargestellt werden können. Basierend auf vereinfachten Modellen wird beispielhaft ein gesamter Supermarkt in der Simulationsumgebung der TLK-Thermo GmbH simuliert und somit ein „Proof-of-concept“ durchgeführt.

Die hier aufgezeigte Anwendung objektorientierter Methoden sowie physikalischer Konnektoren auf die Verschaltung und Simulation von FMUs soll in Zukunft für die Entwicklung und Verwendung einer Komponentendatenbank genutzt werden.

Danksagung:

Teile dieser Arbeit wurden durch das Programm KMU-Innovativ des BMBF finanziert (Förderkennzeichen: 01LY1502A)

6 Referenzen

[FMI] Modelica Association: FMI-Standard. <http://www.fmi-standard.org>, Abruf: 29. Januar 2016

[Otter2014] Otter, M.: FMI Tutorial – Connecting tightly coupled FMUs, *Modelica 2014 Conference, Lund, Mar. 10-12, 2014*

[Franke2009] Franke et al.: Stream Connectors – An Extension of Modelica for Device-Oriented Modeling of Convective Transport Phenomena, *Proceedings 7th Modelica Conference, Como, Italy, Sep. 20-22, 2009*

[Scharff2012] Scharff, D., Kaiser, C., Tegethoff, W., Huhn, M.: Ein einfaches Verfahren zur Bilanzkorrektur in Kosimulationsumgebungen, *Berechnung und Simulation im Fahrzeugbau, November 2012*

[Jäschke2015] Jäschke, J.: Strukturelle Analyse und abstrakte Modellierung eines Supermarktes als thermisches System, *Masterarbeit, Institut für Thermodynamik, TU Braunschweig, 2015*

[Fidorra2013] Fidorra, N., Hafner, A., Lemke, N., Köhler, J.: SuperSmart Energiebenchmarktool für Supermärkte, *DKV-Tagung Hannover, 2013*

[Fidorra2014] Fidorra, N., Hafner, A., Minetto, S., Köhler, J.: SIMULATION MODELS IN THE SUPERSMART SUPERMARKET ENERGY-BENCHMARK TOOL, *3rd IIR International Conference on Sustainability and the Cold Chain, London, UK, 2014*

[Fidorra2014a] Fidorra, N., Hafner, A., Minetto, S., Köhler, J.: APPLICATIONS OF THE SUPERSMART ENERGY-BENCHMARK TOOL FOR SUPERMARKETS, *11th IIR Gustav Lorentzen Conference on Natural Refrigerants, Hangzhou, China, 2014*

[IEA2011] IEA Heat Pump Program Annex 31, 2011: ADVANCED MODELING AND TOOLS FOR ANALYSIS OF ENERGY USE IN SUPERMARKET SYSTEMS, Final Report, Sweden, 2011

[Gemini2014] Gemini 2014, <http://gemini.no/en/2014/06/drastic-cut-in-electricity-bill-for-supermarket/>, letzter Zugriff 27.1.2016

[Titze2012] Titze, M.; Lemke, N.; Neksa, P.; Hafner, A.; Köhler, J.: Dynamic Modeling of a combined supermarket refrigeration and HVAC system, *Gustav Lorentzen Conference 2012, Delft*

[Arias2005] Arias, J.: Energy Usage in Supermarkets - Modelling and Field Measurements, *Dissertation, Stockholm, 2005*

[TIL] TIL Suite – Softwarepaket zur Simulation thermischer Systeme, TLK-Thermo GmbH, <http://www.tlk-thermo.com/>, letzter Zugriff: 04.02.2016

[Sundials] Sundials solver Suite, https://computation.llnl.gov/casc/sundials/description/description.html#descr_cvode, letzter Zugriff: 04.02.2016

[Dymola] Dynamic Modeling Laboratory, <http://www.3ds.com/products-services/catia/products/dymola>, letzter Zugriff: 04.02.2016

[Petzold1982] Petzold, L. R.: A Description of DASSL: A Differential/Algebraic System Solver, *IMACS World Congress, Montreal, 1982*

Algorithmen zur Simulatorkopplung

Christoph Clauß

Fraunhofer IIS EAS Dresden

christoph.clauss@eas.iis.fraunhofer.de

Kurzfassung

Seitdem das Functional-Mockup-Interface (FMI) für Modellaustausch und insbesondere für Co-Simulation definiert wurde und seitdem es mehrere Simulations-Tools gibt, die Functional-Mockup-Units (FMUs) exportieren können, ist zu erwarten, dass gekoppelte Simulationen in weitaus größerem Umfang durchgeführt werden als vordem. Daher ist die Notwendigkeit gestiegen, Algorithmen zur Simulatorkopplung (Masteralgorithmen) zu entwickeln, die das Zusammenwirken von zwei oder mehreren Simulatoren steuern derart, dass die gekoppelte Simulation performant ist und im Vergleich zur ungekoppelten Gesamtsimulation eingeschleppte Fehler begrenzt bleiben. Dieser Problematik widmet sich der Beitrag für Simulationen, die die Lösung von DAEs (Analogsimulationen) umfassen.

Im **ersten** Abschnitt wird die Aufgabe der gekoppelten Simulation als im Allgemeinen nichtlineare Gleichung im Funktionenraum beschrieben. Einige für die Lösung nichtlinearer Gleichungen bekannte Verfahren werden auf diese Aufgabe angewendet (Gauß-Seidel- sowie Gauß-Jacobi-Iterationen und Newton-Verfahren). Dabei entstehen schwer implementierbare Algorithmen, die vereinfacht und praktikabel gemacht werden können durch:

- Die Beachtung der Richtung des Signalflusses zwischen den Simulatoren
Dabei entsteht (global oder lokal) eine Abarbeitungsreihenfolge mit eingebetteten zyklischen Teilen. Im Allgemeinen wird das Gesamtproblem reduziert.
- Die Diskretisierung der Zeitachse
Dies vereinfacht die Beschreibung der Lösungstrajektorien durch die Angabe von Zeit-Wert-Paaren. Verschiedene Schemata der Behandlung einzelner Zeitintervalle werden möglich.

Im Ergebnis entsteht eine Vielzahl von einzelnen Algorithmen.

Im **zweiten** Abschnitt wird der aktuelle Stand der Implementierung einiger weniger dieser Algorithmen vorgestellt, auch anhand von Beispielen.

Im **dritten** Abschnitt wird die Fragestellung von Algorithmen für die Kopplung von „stark unterschiedlichen“ Simulatoren diskutiert. So können zeitaufwendige Simulationen mit hoch performanten gekoppelt werden, was spezielle Algorithmen erfordert.

Der Beitrag berichtet von aktuell laufenden Forschungsarbeiten insbesondere für die Fragestellung des dritten Abschnittes. Bis zum Workshop ist mit neuen Erkenntnissen zu rechnen. Daher verzichte ich auf eine Aufnahme im Tagungsband und möchte den Beitrag als „zu diskutieren“ verstanden wissen.

Auslegung einer erweiterten Betriebsstrategie für ein energieoptimales Elektrofahrzeug

Sören Scherler, Xiaobo Liu-Henke

Ostfalia Hochschule für angewandte Wissenschaften, Fakultät Maschinenbau,
Institut für Mechatronik, Salzdhumer Str. 46/48, 38302 Wolfenbüttel

so.scherler@ostfalia.de

Im vorliegenden Beitrag wird die Auslegung einer erweiterten Betriebsstrategie für ein energieoptimales Elektrofahrzeug dargestellt. Zur Reichweitenerhöhung wird das Elektrofahrzeug um eine PEM-Brennstoffzelle als Range Extender (RE) ergänzt. Ein RE stellt dem Bordnetz zusätzlich zur Batterie elektrische Leistung, welche in diesem Beitrag aus dem Energieträger Wasserstoff gewonnen wird, zur Verfügung. Hauptziel der Betriebsstrategie ist die Maximierung der erzielbaren Gesamtreichweite mit dem Nebenziel der Maximierung der rein mit Leistung aus der Batterie erzielbaren Reichweite, da die Verwendung elektrischer Leistung der Batterie kostengünstiger und energetisch sinnvoller als die Verwendung der vom RE bereitgestellten Leistung ist. Zusätzlich wird die Wirkungsgradcharakteristik der Brennstoffzelle berücksichtigt, um eine hohe Energieausnutzung des Wasserstoffs zu erzielen. Mittels Model-in-the-Loop-Simulation (MiL) wird die ausgelegte Betriebsstrategie anhand verschiedener Testszenarien validiert und die erzielte Reichweitenerhöhung bewiesen.

1 Einleitung

Ein wesentliches Hindernis der breiten Marktdurchdringung rein elektrisch angetriebener Fahrzeuge (Battery Electric Vehicles - BEV) liegt in der kaum vorhandenen Kundenakzeptanz, die aus der geringen Reichweite verglichen mit verbrennungsmotorisch betriebenen Fahrzeugen resultiert. Paradoxerweise reicht die durchschnittliche Reichweite eines BEV mit 80 km für die täglichen Fahrten von ca. 80 % der Fahrer aus [1].

Zur Erhöhung der Reichweite und somit der Kundenakzeptanz werden Range Extended Vehicles (REV) diskutiert, bei welchen es sich um BEV mit einer zusätzlichen elektrischen Energiequelle bzw. serielle Hybridfahrzeuge handelt. Prinzipiell können Wärmekraftmaschinen (WKM) wie Otto-, Diesel-, Wankel-, Freikolben-, Stirling- oder Kugelmotoren sowie Mikrogasturbinen oder elektrochemische Energiewandler wie Brennstoffzellen verwendet werden. Die Verwendung von WKM hat drei wesentliche Nachteile gegenüber Brennstoffzellen, denn erstens ist der Gesamtwirkungsgrad aufgrund des thermodynamischen Carnot-Prozesses begrenzt, zweitens wird zusätzlich ein wirkungsgradbehafteter Generator zur Wandlung mechanischer in elektrische Energie benötigt und drittens wird durch die Verwendung fossiler Energieträger CO₂ ausgestoßen, wodurch das Ziel einer nachhaltigen, emissionsfreien Mobilität verfehlt wird. Aufgrund

dieser Nachteile liegt der Fokus dieses Beitrags auf einer Brennstoffzelle als RE, die je nach verwendeter Energiequelle zur Wasserstoffherzeugung zumindest lokal emissionsfrei ist.

Beim betrachteten Fahrzeug handelt es sich um den, an der Ostfalia Hochschule im Aufbau befindlichen, FREDY – den **F**unktionsträger für **r**egenerative **E**lektromobilität und **F**ahrdynamikregelsysteme [2]. Der FREDY ist ein Elektrofahrzeug mit zwei radnahen Elektromotoren, welche aus einem LiFePO₄-Batteriepaket mit elektrischer Leistung versorgt werden.

Erstes Ziel dieses Beitrags ist die Entwicklung einer neuen Fahrzeugkonfiguration mit einer Brennstoffzelle, welche elektrische Leistung entweder für das Laden der Batterie oder für die direkte Versorgung der elektrischen Antriebsmotoren und Bordnetzverbraucher bereitstellen kann.

Das zweite Ziel dieses Beitrags ist ein Vergleich verschiedener Betriebsstrategien hinsichtlich der erzielbaren Reichweitenerhöhung durch die Brennstoffzelle bei gleicher Wasserstoffmenge.

2 Methodik

Um die Komplexität bei der Entwicklung hochgradig vernetzter Systeme zu beherrschen, wird das Gesamtsystem gemäß mechatronischer Methodik [3] modular

und hierarchisch strukturiert, sodass die Entwicklung einzelner Funktionen möglich ist [4].

Die einzelnen Funktionen werden modellbasiert ausgelegt und in einem durchgängigen Prozess bestehend aus Model-in-the-Loop-(MiL), Software-in-the-Loop-(SiL) und Hardware-in-the-Loop-(HiL)-Simulationen validiert und optimiert. Durch diesen Prozess finden bereits in frühen Entwicklungsphasen Erprobungen an Prototypen (u.a. Modellen) statt, sodass die Funktion frühzeitig abgesichert werden kann.

In diesem Beitrag wird die MiL-Simulation im Rahmen der modellbasierten Auslegung der Betriebsstrategien für REV mit Brennstoffzellen dargestellt.

3 Fahrzeugtopologie

Die Fahrzeugtopologie des FREDY ist in Abbildung 1 dargestellt. Zur Traktion werden zwei Synchronmotoren mit radnahen Getrieben eingesetzt, welche ihre Energie aus dem Hochvolt-Bordnetz (HV-Bordnetz) beziehen und im Bremsfall elektrische Energie in dieses einspeisen. Das HV-Bordnetz versorgt zudem mittels unidirektionalem DC/DC-Wandler das Niedervolt-Bordnetz (NV-Bordnetz), welches elektrische Energie für Verbraucher wie Fahrzeugbeleuchtung bereitstellt, mit Energie. Zur Reichweitenerhöhung wird eine Brennstoffzelle mitsamt Wasserstofftank und einem weiteren unidirektionalen DC/DC-Wandler als RE im FREDY verbaut, wodurch dem HV-Bordnetz zusätzliche elektrische Energie bereitgestellt und die erzielbare Reichweite erhöht wird.

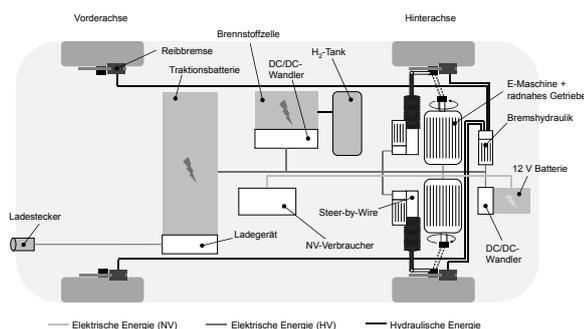


Abbildung 1. Fahrzeugtopologie des FREDY mit RE

4 Modellbildung

In diesem Abschnitt werden das verwendete Gesamtfahrzeugmodell und notwendige Erweiterungen zur Integration einer PEM-Brennstoffzelle dargestellt.

4.1 Das Gesamtfahrzeug

Für die Simulation wird ein vorhandenes echtzeitfähiges Fahrzeugmodell [5] verwendet, welches in der Softwareumgebung von Matlab/Simulink ausgeführt wird. Basierend auf einem Mehrkörpersystem mit 24 Freiheitsgraden werden die Eigenschaften des Fahrzeugs abgebildet. Der bestehende Antriebsstrang wird um ein Zweispannungsbordnetz mit PEM-Brennstoffzelle als RE erweitert.

4.2 Lithium-Ionen-Batterie des HV-Bordnetzes

Energiespeicher im HV-Bordnetz ist ein Batteriepaket bestehend aus zwei parallelen Strängen mit je 15 seriell angeordneten Lithium-Ionen-Batteriezellen (LiFeYPO₄) mit einer Nennspannung von 48V und einer Ladeschlussspannung von 60V bei einer Nennkapazität von 120 Ah.

Die verwendete Batteriezelle WB-LYP60AHA der Firma Winston Batteries wurde mithilfe der Elektrochemischen Impedanzspektroskopie (EIS) im Frequenzbereich identifiziert [6].

Eine für die Batterie wesentliche Größe ist ihr Ladezustand (State of Charge, SoC), welcher nach (1) aus einem Startwert SoC₀, dem effektiven Strom $i(t)$, dem Coulombschen Wirkungsgrad η und der Zellkapazität C_n bestimmt werden kann:

$$SoC(t) = SoC_0 + \int \frac{\eta \cdot i(t)}{C_n} dt \quad (1)$$

Für die Batterie wurde ein elektrisches Ersatzschaltbild (vgl. Abbildung 2) bestehend aus einer Spannungsquelle (Wiedergabe der nichtlinearen Open-Circuit-Voltage, OCV, Ruhespannung), einem Widerstand R_S sowie vier RC-Gliedern zur Abbildung der Dynamik aufgestellt.

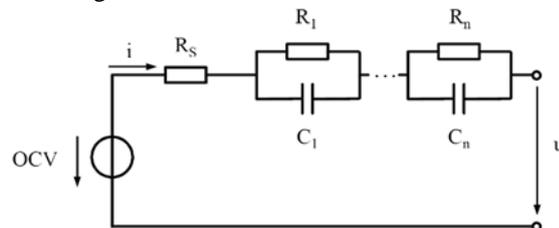


Abbildung 2. Elektrisches Ersatzschaltbild der Batterie

Mathematisch lässt sich der Frequenzgang laut [6] nach (2) beschreiben.

$$\frac{u(j\omega)}{i(j\omega)} = R_S + \frac{R_1}{R_1 \cdot C_1 \cdot j\omega + 1} + \dots + \frac{R_n}{R_n \cdot C_n \cdot j\omega + 1} \quad (2)$$

Der Frequenzgang ist vom Ladezustand der Zelle abhängig. Dementsprechend wurde der Frequenzgang für verschiedene SoC gemessen (vgl. Abbildung 3).

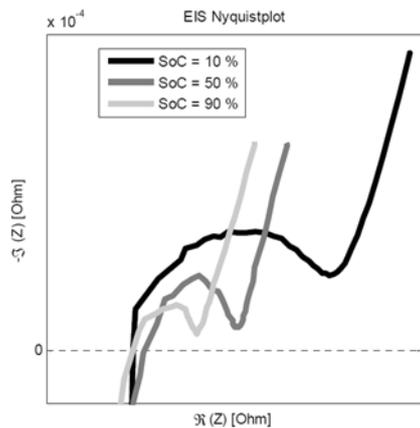


Abbildung 3. Messergebnis der EIS [6]

Zur Berücksichtigung des nichtlinearen Verhaltens der Ruhespannung abhängig vom SoC wurden Messungen im Zeitbereich durchgeführt. Das Ergebnis der Messungen zeigt Abbildung 4.

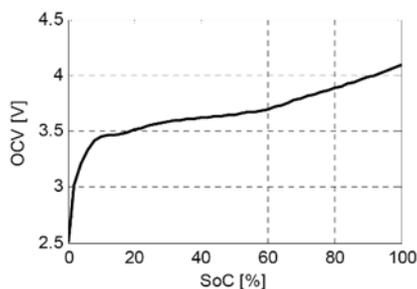


Abbildung 4. OCV-Kennlinie [6]

Das Modell der Batterie liefert abhängig vom Eingangsstrom eine Spannung und ändert seinen SoC , wodurch das Batterieverhalten hinreichend gut wiedergegeben wird.

4.3 Bleibatterie des NV-Bordnetzes

Für die Bleibatterie des NV-Bordnetzes wurde das im Gesamtfahrzeug (vgl. Kap. 4.1) bereits integrierte Modell verwendet. Die Parameter für die Zelldynamik wurden beibehalten und lediglich Kapazität, Masse und Abmessungen wurden an die im FREDY verwendete Varta E44 mit einer Kapazität von 77 Ah bei einer Nennspannung von 14 V angepasst.

Berücksichtigt werden im dSpace-Modell Abhängigkeiten von SoC , I_{Bat} , T_{Bat} und SoH auf den Batteriewiderstand. Weiterhin ist die OCV eine Funktion des

SoC . Die Dynamik der Batterie wird durch eine Induktivität sowie je einem RC-Glied für Doppelschicht und Diffusion abgebildet.

4.4 DC/DC-Wandler für HV- und NV-Bordnetz

Das NV-Bordnetz wird aus dem HV-Bordnetz mithilfe des DC/DC-Wandlers Mean Well SD-1000L-12 mit elektrischer Leistung versorgt.

Der Wirkungsgrad der Leistungsübertragung ist abhängig von den Ein- und Ausgangsspannungen sowie dem Ausgangsstrom. Ein gemessenes Wirkungsgradkennfeld für eine konstante Ausgangsspannung von 14 V ist in Abbildung 5 dargestellt.

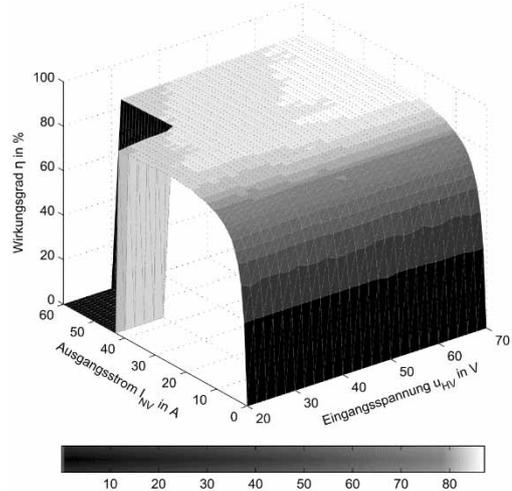


Abbildung 5. Wirkungsgradkennfeld des Mean Well SD-1000L-12 bei $u_{NV} = 14$ V

Für die Leistungsübertragung unter Berücksichtigung des Wirkungsgrades gilt (3).

$$i_{NV} \cdot u_{NV} = \eta(i_{NV}, u_{HV}) \cdot i_{HV} \cdot u_{HV} \quad (3)$$

4.5 Elektrische Verbraucher

Die elektrischen Verbraucher des FREDY im NV-Bordnetz sind in Tabelle 1 aufgeführt.

Tabelle 1: Leistungsdaten der elektrischen Verbraucher im NV-Bordnetz.

Verbraucher	Leistungsaufnahme in W
Fahrzeugrechner	150
Abblendlicht	110 (2x55)
Fernlicht	96 (2x48)
Bremsleuchten	42 (2x21)
Rückleuchten	88 (2x44)
Blinklichter	84 (2x42)

Bei den Verbrauchern im HV-Bordnetz handelt es sich um die Antriebsmaschinen (vgl. Abschnitt 4.8), das Bremssystem, für den Fall, dass das rekuperative Bremsen mit den Antriebsmaschinen nicht ausreicht, und das NV-Bordnetz.

4.6 Brennstoffzelle

Betrachtet wird die PEM-Brennstoffzelle Nuvera PowerFlow mit einer elektrischen Ausgangsleistung von maximal 5 kW. Das Modell nach Boscaino [7] ist ein parametrisiertes, empirisches Modell des statischen Verhaltens mit Approximation der Dynamik durch ein elektrisches Ersatzschaltbild. Die statische Ausgangsspannung

$$u_{BZ} = A - B \cdot \ln\left(1 + \frac{i_{BZ}}{C}\right) - D \cdot \exp\left(\frac{i_{BZ}}{E}\right) \quad (4)$$

ergibt sich aus den empirischen Parametern A, B, C, D und E sowie dem durch die Brennstoffzelle fließenden Strom

$$i_{BZ} = i_{Last} + i_{dyn}. \quad (5)$$

Mithilfe des dynamischen Stroms

$$i_{dyn} = \frac{u_{BZ}}{Z} \quad (6)$$

wird das dynamische Verhalten durch Verwendung der Impedanz

$$Z = \frac{1}{\frac{1}{R_1 \cdot C_1} + \frac{1}{C_2}} + \frac{1}{\frac{1}{R_3} + \frac{1}{C_3}} \quad (7)$$

approximiert. Der Wirkungsgrad der Brennstoffzelle

$$\eta_{BZ} = \frac{E_Z}{E_H^0} \quad (8)$$

wird nach Eichseder [8] als Verhältnis der Zellspannung

$$E_Z = \frac{u_{BZ}}{n_{Zellen}} = \frac{u_{BZ}}{40} \quad (9)$$

zur thermoneutralen Spannung

$$E_H^0 = -\frac{\Delta RH^0}{z \cdot F} = -\frac{-\frac{285 \text{ kJ}}{\text{mol}}}{2 \cdot 96485 \frac{\text{As}}{\text{mol}}} \cong 1,47 \text{ V} \quad (10)$$

berechnet. Der Verbrauch der Arbeitsgase Wasserstoff

$$\dot{m}_{H_2} = \frac{n_{Zellen} \cdot M}{n \cdot F} i \quad (11)$$

und Sauerstoff

$$\dot{m}_{O_2} = \frac{1}{2} \dot{m}_{H_2} \quad (12)$$

wird nach Amphlett [9] als Massestrom abhängig von der Anzahl der Zellen n_{Zellen} , der molaren Masse M

von Wasserstoff, der Ladungszahl n , der Faraday-Konstante F sowie der abgegebenen Stromstärke i bestimmt.

Als Wasserstoffspeicher wird der Metallhydridspeicher MyH2 3000 mit einem Füllvolumen von 3000 NI gewählt. Bei einer Wasserstoffdichte unter Normbedingungen von $\rho_{H_2} = 0,0899 \text{ kg/m}^3$ entspricht dies 270 g Wasserstoff.

4.7 DC/DC-Wandler für Brennstoffzelle und HV-Bordnetz

Da kein Wirkungsgradkennfeld eines entsprechenden DC/DC-Wandlers vorliegt, wird vereinfacht ein konstanter Wirkungsgrad von 85 % für die Leistungsübertragung von der Brennstoffzelle in das HV-Bordnetz angenommen. Hieraus folgt die Leistungsübertragung der Brennstoffzelle nach (13).

$$i_{HV} \cdot u_{HV} = \eta_{DCDC} \cdot i_{BZ} \cdot u_{BZ} \quad (13)$$

4.8 Antriebsmaschinen

Der FREDY wird von zwei permanenterregten Synchronmaschinen ME0907 der Fa. Montenergy mit radnahen Getrieben an der Hinterachse angetrieben. Das Maximalmoment bei einer Nennspannung von 48 V beträgt 26 Nm bei einer Stromaufnahme von 200 A.

Betrachtet werden die Synchronmaschinen mit feldorientierten Raumzeigern in d- und q-Richtung eines rotorfesten Koordinatensystemen. Daraus ergeben sich die Spannungen in d-Richtung

$$u_d = R_S \cdot i_d + L_d \cdot \frac{di_d}{dt} - \omega \cdot L_q \cdot i_q \quad (14)$$

sowie in q-Richtung

$$u_q = R_S \cdot i_q + L_q \cdot \frac{di_q}{dt} + \omega \cdot (L_d \cdot i_d + \Psi_{PM}). \quad (15)$$

Unter Vernachlässigung von Streuinduktivitäten gilt für das Motormoment (16).

$$M_{Mot} = \frac{3}{2} p [\Psi_{PM} \cdot i_d + (L_d - L_q) i_d \cdot i_q] \quad (16)$$

Die Parameteridentifikation erfolgte zunächst durch Frequenzgangmessungen mittels Präzisions-FFT-Analysator SigLab 20-22 der Fa. Spectral Dynamics und anschließend durch Optimierung im Zeitbereich zur Berücksichtigung nichtlinearer Effekte.

Die Auslegung und Validierung der feldorientierten Regelungen mittels MiL-, SiL- und HiL-Simulation wurde in der vorhergehenden Arbeit [10] durchgeführt und wird hier nicht weiter ausgeführt.

4.9 Radnahe Getriebe

Die Drehmomente der Synchronmaschinen werden mithilfe der radnahen Getriebe PLN 115-7 der Fa. Neugart mit einer Übersetzung $i = 7$ erhöht. Es ergeben sich die Zusammenhänge (17) und (18) für Ausgangsdrehzahl und -drehmoment unter Berücksichtigung eines Getriebewirkungsgrades η .

$$\omega_{aus} = i \cdot \omega_{ein} \quad (17)$$

$$M_{aus} = \eta \cdot i \cdot M_{ein} \quad (18)$$

5 Betriebsstrategie

Ein wesentliches Ziel der Betriebsstrategie ist die Lösung des Zielkonflikts zwischen dem Wunsch nach möglichst maximaler Reichweite bei gleichzeitig hoher Fahrdynamik.

Als Nebenbedingung sind die Energiekosten zu berücksichtigen, denn der in der HV-Batterie gespeicherte Strom aus der Ladesäule ist deutlich kostengünstiger als der aus dem Wasserstoff erzeugte. Ziel ist es, die ausschließlich mithilfe der Batterie erzielbare Reichweite zu erhöhen. Somit werden Kurzstrecken, welche einen hohen Anteil der täglichen Fahrten ausmachen, möglichst ohne Einsatz des RE absolviert.

Diese Strategie wird auch von Rogge et al. [11] angewandt (vgl. Abbildung 6) und ist charakterisiert durch die Phasen der Ladungsentnahme (Charge Depleting) und des Ladungserhalts (Charge Sustaining).

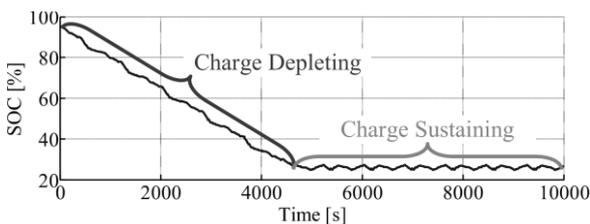


Abbildung 6. Betriebsführung eines REV mit Charge Depleting und Sustaining nach Rogge et al. [11]

Während des Charge Depletions wird der RE nicht verwendet und dementsprechend reduziert sich der *SoC* der HV-Batterie aufgrund der Energieentnahme. Bei Erreichen eines kritischen unteren Ladezustands wird der RE zugeschaltet, um den *SoC* konstant zu halten. Ein weiteres Laden der HV-Batterie macht keinen Sinn, da unnötiger Zusatztreibstoff verbraucht werden würde.

Neben dieser Vorgehensweise ist der Brennstoffzellenbetrieb mit einem hohen Wirkungsgrad entscheidend

für eine hohe Reichweite. Aus Abbildung 7 wird erkenntlich, dass dieser Wirkungsgrad mit steigender Stromstärke geringer wird. Im Umkehrschluss dürfen für einen hohen Wirkungsgrad der Brennstoffzelle nur geringe Stromstärken entnommen werden.

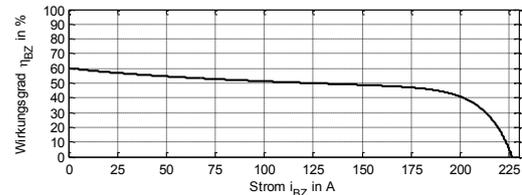


Abbildung 7. Wirkungsgrad der Nuvera PowerFlow

Es muss demnach ein Kompromiss zwischen der Stromstärke und dem Wirkungsgrad gefunden werden, denn bei zu kleiner Stromstärke reicht die aufgebrauchte Leistung nicht für den Ladungserhalt der HV-Batterie und bei zu hoher Stromstärke sinkt die Reichweite, da die Energieverluste größer werden.

Der Ladungserhalt bei geringem *SoC* kann durch eine Verknüpfung mit dem elektrischen Energiemanagement (EEM, vgl. Vorarbeit [12]) sichergestellt werden: Das EEM sorgt für eine bedarfsgerechte und prioritätsbasierte Verteilung der im Elektrofahrzeug begrenzt verfügbaren elektrischen Leistung auf die elektrischen Verbraucher bei gleichzeitiger Gewährleistung von Fahrsicherheit und möglichst nicht wahrnehmbaren Einbußen im Fahrkomfort. Bei geringerem *SoC* wird mittels EEM die Leistung der Antriebsmaschinen begrenzt, womit zwar das Beschleunigungsvermögen und die Maximalgeschwindigkeit sinken, aber die Reichweite jedoch erhöht wird. Die Reichweitenerhöhung lässt sich zum einen auf die geringeren Verluste der Antriebsmaschinen (höherer Wirkungsgrad) und zum anderen auf den besseren Betriebspunkt der Brennstoffzelle zurückführen. Weiterhin werden die Fahrwiderstände durch die geringere Endgeschwindigkeit verringert.

6 MiL-Simulation

Zur Validierung der ausgelegten Betriebsstrategie werden zahlreiche MiL-Simulationen mit verschiedenen Testszenarien herangezogen. Durch die Gegenüberstellung verschiedener Simulationsergebnisse wird das Potential der Reichweitenerhöhung der ausgelegten Betriebsstrategie aufgezeigt.

6.1 Fahrmanöver und Testszenarien

Der FREDY erreicht eine Maximalgeschwindigkeit von ca. 62 km/h, weshalb es als Elektrofahrzeug für Fahrten in der Stadt betrachtet wird. Untersuchungen mit gängigen Fahrzyklen wie dem New York City Cycle, FTP75 oder CADC Urban können allerdings nicht durchgeführt werden, da das Beschleunigungsvermögen des FREDY nicht ausreicht, um diesen zu folgen. Als Fahrmanöver wird deshalb der Handling-Track verwendet (vgl. Abbildung 8).

Es handelt sich um einen Rundkurs der Länge 3040 m mit vielen Kurven, weshalb der FREDY wie im Stadtverkehr häufig positiv und negativ beschleunigt wird.

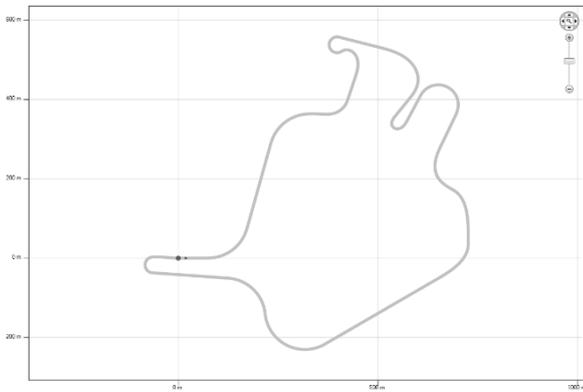


Abbildung 8. HandlingTrack (Fa. dSpace)

Es werden die Szenarien

- Fahrzeug ohne RE als Benchmark,
- Ladungserhalt bei einem SoC von 20 % und
- Ladungserhalt bei einem SoC von 20 % mit Begrenzung der Antriebsleistung

betrachtet, um die zuvor getroffenen Aussagen bzgl. der Betriebsstrategie zu bestätigen.

6.2 Verbesserungspotential

In diesem Abschnitt werden die Ergebnisse der einzelnen Simulationsszenarien diskutiert und abschließend das Potential zur Reichweitenerhöhung dargestellt.

1) Ohne Brennstoffzelle als RE

In Abbildung 9 sind wesentliche Kenngrößen über der simulierten Dauer dargestellt.

Ohne Brennstoffzelle beträgt die erzielte Reichweite 82 km. Diese wird nachstehend als Benchmark für die entwickelten Betriebsstrategien verwendet.

Nach ca. 1,7 h wird ein kritischer SoC (mittig) von 5 % erreicht. Zum Schutz der HV-Batterie vor Tiefenentladung wird diese vom Bordnetz getrennt. Infolgedessen

rollt das Fahrzeug aus und die gefahrene Strecke (oben) steigt nicht weiter an. Um die Betriebsfähigkeit wieder zu erlangen, muss das Fahrzeug geladen werden.

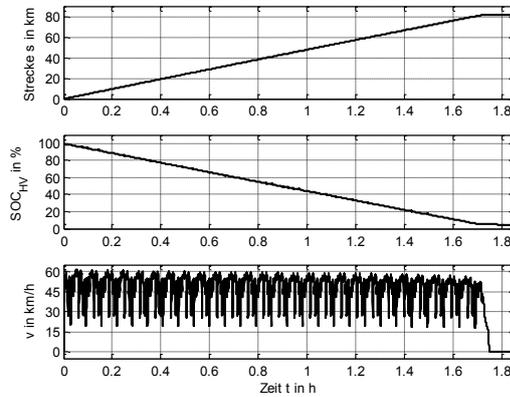


Abbildung 9. Erzielte Reichweite (oben), SoC der HV-Batterie (mittig) und Geschwindigkeit (unten) ohne RE

2) Ladungserhalt

In diesem Fall wird der RE erst bei Unterschreiten eines SoC von 20 % eingesetzt, um für einen Ladungserhalt zu sorgen. Die Simulationsergebnisse sind in Abbildung 10 dargestellt. Es wird mit 142,5 km eine um 60,5 km (+73,8 %) höhere Reichweite bezogen auf die Fahrt ohne RE erzielt.

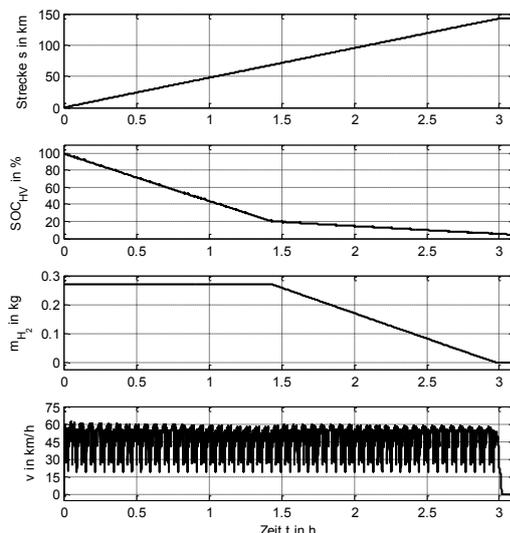


Abbildung 10. Charakteristische Fahrzeuggrößen bei Ladungserhalt ab einem SoC von 20 %

Bei Erreichen eines SoC von 20 % beginnt die Umwandlung der im Wasserstoff gebundenen Energie in

elektrische Energie. Dies ist an der abnehmenden Wasserstoffmenge in der dritten Zeile von Abbildung 10 zu erkennen. Der Ladezustand der Batterie sinkt jedoch weiter, was auf eine unzureichende Stromabgabe der Brennstoffzelle zurückzuführen ist. Die gemittelte Stromabgabe der Brennstoffzelle beträgt 99,15 A bzw. transformiert 59,91 A im HV-Bordnetz bei einem mittleren Wirkungsgrad von 51,02 %. Demgegenüber steht eine gemittelte Stromaufnahme der Antriebsmaschinen und des NV-Bordnetzes von 72,21 A, sodass die Leistung der Brennstoffzelle nicht ausreicht um die Leistung der HV-Verbraucher zu kompensieren.

3) Ladungserhalt und Leistungsbegrenzung

Um einen Ladungserhalt bei einem SoC von 20 % durchführen zu können und gleichzeitig den Wirkungsgrad der Brennstoffzelle zu erhöhen, wird die Stromabgabe der Brennstoffzelle in das HV-Bordnetz auf 50 A begrenzt. Gleichzeitig wird die maximale Antriebsleistung bei einem SoC von 20 % und bei einem SoC von 10 % schrittweise reduziert, bevor die Batterie bei 5 % vom Netz getrennt wird. Abbildung 11 zeigt die Ergebnisse dieses Szenarios.

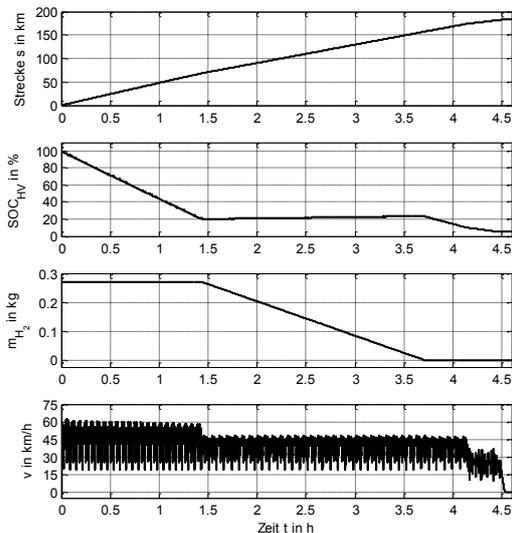


Abbildung 11. Charakteristische Fahrzeuggrößen bei Ladungserhalt ab einem SoC von 20 % mit auf 50 A begrenzter Stromabgabe ins HV-Bordnetz und schrittweiser Begrenzung der Antriebsleistung

Die erzielte Reichweite (oben) konnte um weitere 40,2 km auf 182,7 km gesteigert werden. Dies ist zum einen auf den höheren Wirkungsgrad der Brennstoffzelle von 52,72 % und zum anderen auf die deutlich geringere Leistungsaufnahme der Antriebsmaschinen

und die damit einhergehende Reduktion der Verluste an den Kupferwicklung zurückzuführen.

Allerdings sinkt durch die begrenzte Antriebsleistung die Maximalgeschwindigkeit (unten) von 62 km/h auf zunächst 50 km/h und schließlich auf 35 km/h. Die erste Begrenzung der Maximalgeschwindigkeit ist insbesondere in Hinblick auf den Einsatz im Stadtverkehr unbedenklich. Durch die zweite Begrenzung wird das Notlaufverhalten berücksichtigt, welches dem Fahrer bei Fahrtende eine möglichst hohe Restreichweite zum Aufsuchen einer Ladesäule ermöglicht.

Abbildung 12 zeigt die Ergebnisse des gleichen Szenarios mit einer auf 30 A begrenzten Stromabgabe der Brennstoffzelle in das HV-Bordnetz.

Die erzielte Reichweite (oben) konnte abermals um 30,4 km auf 213,1 km und der mittlere Wirkungsgrad der Brennstoffzelle auf 55,88 % erhöht werden. Allerdings reicht der Strom der Brennstoffzelle nicht mehr für den Ladungserhalt bei einem SoC von 20 %, sodass er verhältnismäßig schnell auf 10 % sinkt (2. von oben) und die Antriebsleistung weiter begrenzt wird. Der Fahrer könnte weitere 4 Stunden mit einer begrenzten Maximalgeschwindigkeit von 35 km/h weiterfahren. Dieses Beispiel verdeutlicht den Konflikt zwischen einer hohen Reichweite bei geringen Stromstärken und einer angemessenen Fahrgeschwindigkeit, die hohe Stromstärken erfordert.

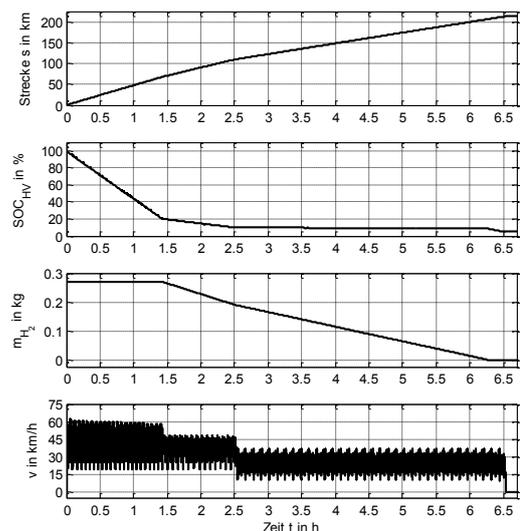


Abbildung 12. Charakteristische Fahrzeuggrößen bei Ladungserhalt ab einem SoC von 20 % mit auf 30 A begrenzter Stromabgabe ins HV-Bordnetz und schrittweiser Begrenzung der Antriebsleistung

7 Resümee

In diesem Beitrag wurde eine Betriebsstrategie für REV mit dem Ziel der Reichweitenmaximierung modellbasiert ausgelegt.

Hierzu wurde zunächst das bestehende Fahrzeugmodell des FREDY um eine Brennstoffzelle erweitert, um darauf aufbauend eine Betriebsstrategie auszulegen. Durch Berücksichtigung der Wirkungsgradcharakteristik der Brennstoffzelle und der Begrenzung des Beschleunigungsvermögens ab einem unteren Grenzwert konnte die Reichweite im betrachteten Fahrzyklus von 82 km ohne Brennstoffzelle um bis zu 159,9 % auf 213,1 km erhöht werden.

Deutlich wurde der Zielkonflikt zwischen maximaler Reichweite und hoher Fahrtgeschwindigkeit, welcher im Rahmen zukünftiger Untersuchungen genauer untersucht werden wird.

Aufbauend auf diese Arbeit gilt es Fahrzeuge mit höheren Antriebsleistungen zu betrachten, da die Maximalgeschwindigkeit von 62 km/h für die Fahrt langer Strecken, wie sie mittels RE ermöglicht werden sollen, deutlich zu gering ist.

Ferner gilt es einen HiL-Prüfstand zu entwickeln um die in diesem Beitrag entworfenen Algorithmen in der Praxis zu testen und zu optimieren, bevor der RE in den FREDY eingebaut wird.

8 Danksagung

Dieser Beitrag wurde im Rahmen des Forschungsprojekts *Energy Consumption Optimization for electrical Cars* (kurz ECOCar) durch das Bundesministerium für Bildung und Forschung unter dem Förderkennzeichen 03FH035I3 gefördert. Die Verantwortung für den Inhalt liegt bei den Autoren, welche sich für die Förderung herzlichst bedanken.



9 Literaturverzeichnis

[1] Bertram, M. und Bongard, S.: Elektromobilität im motorisierten Individualverkehr: Grundlagen,

Einflussfaktoren und Wirtschaftlichkeitsvergleich. Wiesbaden: Springer Fachmedien Wiesbaden 2014.

- [2] Liu-Henke, X.; Göllner, M.; Fritsch, M.; Feind, R.; Quantmeyer, F. und Buchta, R.: FreDy – an electric vehicle with intelligent chassis control systems. 10th International Conference on Ecological Vehicles and Renewable Energies (EVER'15), Monte Carlo, Monaco, March 31 - April 2 2015.
- [3] Liu-Henke, X.: Mechatronische Entwicklung der aktiven Feder-, Neigetechnik für das Schienenfahrzeug RailCab. Düsseldorf: VDI-Verl. 2005.
- [4] Quantmeyer, F.; Liu-Henke, X. und Scherler, S.: Design and Test of the Vehicle Management of Electric Vehicles with Decentralized Drives. 12. Symposium: Hybrid- und Elektrofahrzeuge (HEV 2015), Braunschweig, Germany, February 24 - 25 2015.
- [5] Buchta, R.: Mechatronische Entwicklung eines Forschungselektrofahrzeugs zur Erprobung von Fahrdynamikregelungen und Fahrerassistenzsystemen: Dissertation, Bericht aus der Fahrzeugtechnik, Shaker Verlag, Aachen, 2015 2015.
- [6] Quantmeyer, F. und Liu-Henke, X.: Modeling the Electrical Behavior of Lithium-Ion Batteries for Electric Vehicles, Solid State Phenomena (2014) H. 214. S. 40–47.
- [7] Boscaino, V.; Capponi, G.; Miceli, R.; Rizzo Galluzzo, G. und Rizzo, R.: Comparison of models of fuel cells based on experimental data for the design of power electronic systems, IET Renewable Power Generation 9 (2015) H. 9. S. 660–668.
- [8] Eichlseder, H.: Wasserstoff in der Fahrzeugtechnik: Erzeugung, Speicherung, Anwendung. 3. Auflage. Wiesbaden: Vieweg+Teubner Verlag 2012.
- [9] Amphlett, J. C.; Mann, R. F.; Peppley, B. A.; Roberge, P. R. und Rodrigues, A.: A practical PEM fuel cell model for simulating vehicle power sources. 10th Annual Battery Conference on Applications and Advances (BCAA 1995), Long Beach, CA, USA, Jan 10 - 13 1995.
- [10] Liu-Henke, X.; Quantmeyer, F. und Scherler, S.: Hierarchical Vehicle Management for Battery-Electric Vehicles with Decentralized Drives.

10th International Conference on Ecological Vehicles and Renewable Energies (EVER'15), Monte Carlo, Monaco, March 31 - April 2 2015.

- [11] Rogge, M.; Rothgang, S. und Sauer, D. U.: Operating Strategies for a Range Extender Used in Battery Electric Vehicles. IEEE Vehicle Power and Propulsion Conference (VPPC 2013), Beijing, China, October 15 - 18 2013.
- [12] Scherler, S.; Quantmeyer, F. und Liu-Henke, X.: Modellbasierte Entwicklung des elektrischen Energiemanagements für Elektrofahrzeuge mit einem Zweispannungsbordnetz. Workshop der ASIM/GI-Fachgruppen, Stralsund, Germany, June 18 - 19 2015.

Cross-boundary Simulations of Fuel Cells

Szilárd Dombi^{1,2}, Prof. Dr. Walter Commerell¹, Prof. Dr. rer.nat. Hubert Mantz¹, Dr. Ferenc Lezsovits²

¹Hochschule Ulm

²Budapest University of Technology and Economics

dombi@mail.hs-ulm.de

Simulation software systems play an important role in Automotive Fuel Cell System simulations. However, a considerable number of simulation system types are available on the market, which poses a dilemma for researchers. The question this review paper aims to answer is the following: What is the most effective way to choose the appropriate simulation system? The paper (1) presents an inventory of typical issues that researchers seek to address and (2) discusses software that can help them find the solutions. When an issue can be addressed using several simulation systems, a comparison is made between them. When an issue is so complex that it cannot be addressed independently with any one of the available software, viable software combinations are proposed and examined. In such cases, the output of a software is used as the input for another, and the cross-boundary connections between them are addressed. Finally, the paper presents a brief overview of modelling possibilities and a tabulated summary of the findings.

Keywords: simulation software system, fuel cell, benchmark

1 Introduction

Fuel Cells constitute a promising technology for the future, for example, because of their favorable environmental impact, efficiency, power density, and low noise level.

There are different types of Fuel Cells, but in the automotive sector the Polymer Electrolyte Fuel Cell (PEFC) is the most common. The PEFC generates electric current from electrochemical reactions between hydrogen and oxygen with the byproduct of heat and water.

1.1 Fuel Cell system

A Fuel Cell System (FCS), as shown on Figure 1., has several components, which are defined in IEC 62282-1:2005 [1].

These components work together to produce electric and thermal power from the inputs of hydrogen, air, water, and the system also requires electric and thermal power.

The main components of a FCS are the Fuel cell stack; the air processing system, which contains a humidifier and a compressor; the fuel gas supply; and the thermal- and water management systems. Furthermore, there are several peripheral components such as pumps, valves, and electrical devices. The

fuel and the air are not part of the FCS; they are fed into it.

The FCS and its parts raise many questions that researchers try to answer with the help of different types of simulation software systems. This paper aims to investigate which simulation software system are the most suitable for addressing the researchers' issues.

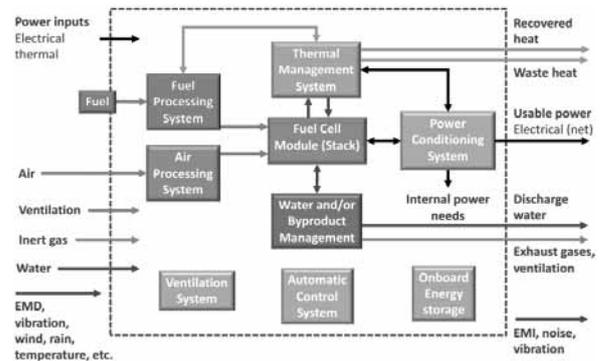


Figure 1. Fuel Cell System according to IEC 62282-1 [2]

2 Fuel Cell simulation software systems

There are several types of Fuel Cell simulation software systems available on the market, but only a few of them are used for scientific research. This does not mean that the others do not work properly. Neverthe-

less, this paper only discusses those types of Fuel Cell simulation software systems that are preferred by the scientific community.

These simulation software systems are:

- MATLAB® & Simulink®
- ANSYS Fluent
- AspenTech
- COMSOL Multiphysics

Each simulation software system has its own Fuel Cell specialised packages, which help making quick calculations. Whereas these calculations may be sufficient sometimes, all of the packages have limitations that can only be solved by declaring equations and by setting up subsystems that are problem-specific.

Cross-boundary Simulations are available between:

- MATLAB® & Simulink® and ANSYS Fluent by means of writing a journal file
- MATLAB® & Simulink® and COMSOL Multiphysics by means of MATLAB S-function and COMSOL Livelink
- MATLAB® & Simulink® and AspenTech Dynamics modul by means of Control System Toolbox
- ANSYS Fluent and AspenTech by means of APECS [3]

ANSYS Fluent and COMSOL Multiphysics have been built for the same purpose; therefore, in their case there is no need for cross-boundary connection.

MATLAB® & Simulink® is mostly used for dynamic system simulations, ANSYS Fluent and COMSOL Multiphysics are suitable for Computational Fluid Dynamics, and AspenTech has been developed for chemical reactions and system optimizing.

3 Typical questions

Some issues relate to an FCS as a whole whereas some others pertain to only a specific part of the system.

Issues

Stack operating conditions in different stack load points

Lower stack cathode pressure effect of FCS efficiency

Hydrogen consumption at different driving cycles

The effect of the compressor efficiency on the overall system efficiency

Stack idle operating conditions

Impact of the membrane resistance to the FCS peak efficiency

The impact of the compressor dynamic on the system dynamics

Table 1. Issues that can be addressed using MATLAB® & Simulink® [4]

Issues

Electochemistry modelling

Current and mass conservation

Heat source problems

Liquid water formation and transport and their effects

Transient simulations

Leakage current

Table 2. Issues that can be addressed using ANSYS Fluent [5]

Issues

Fuel processing, hydrogen purification

Heat recovery, and water recovery

Steady-state material balances, heat integration

Dynamic analysis of heat-up, cool-down, power ramping

Safety analysis

Modeling of adsorption processes used to purify the fuel gas in the fuel conditioning section

Table 3. Issues that can be addressed using AspenTech [6]

Issues

Transport of charged and neutral species

Current conduction

Fluid flow, heat transfer, and the nature and driving forces of electrochemical reactions at planar and in porous electrodes

Design and optimization of the geometries and material choices of the system's electrodes, separators, membranes, electrolyte, current collectors and feeders with respect to performance, thermal management, and safety

Table 4. Issues that can be addressed using COMSOL Multiphysics [7]

4 Conclusion

In this paper, different kinds of simulation software systems have been reviewed. Tables 1-4 summarize the issues according to the simulation software systems with which they can be addressed. None of the software can by itself provide the answers to all the questions, so for a complex analysis we need to use multiple software systems. Section 2 above presents the different feasible simulation software system combinations.

Acknowledgement

This work was supported by the DAAD (German Academic Exchange Service).

5 References

- [1] Fuel Cell Technologies BSI Technical Committee GEL/105. *DD IEC TS 62282-1:2005 Fuel cell technologies - Part 1: Terminology*. BSI British Standards, 18 April 2005.
- [2] Dr. Alexander Kabza. *Fuel Cell Formulary*. Formulary, Ulm, S. 19, 16. December 2015.
- [3] <http://www.colan.org/Catalog/ANSYS.html>
- [4] Dr. Alexander Kabza. *Automotive Fuel Cell Systems*. Presentation, ZSW Ulm, S. 60, 10. December 2015.
- [5] ANSYS, Inc. *ANSYS Fluent Fuel Cell Modules Manual*. Manual, Canonsburg, November 2013
- [6] AspenTech. *aspen Fuel Cell Engineering Solution*. Brochure, Cambridge.
- [7] <https://www.comsol.com/batteries-and-fuel-cells-module>

Eine intelligente Fahrerassistenzfunktion zum energieoptimalen Fahrbetrieb für Elektrofahrzeuge

Matthias Fritsch, Sören Scherler, Xiaobo Liu-Henke

Ostfalia Hochschule Wolfenbüttel

matthias.fritsch@ostfalia.de

Der vorliegende Beitrag fokussiert die Entwicklung einer intelligenten Fahrerassistenzfunktion, eines energieoptimalen Geschwindigkeitsprofils, für Elektrofahrzeuge. Im ersten Schritt werden Streckendaten verarbeitet und die zur Beeinflussung der Längsdynamik verfügbaren Betriebsstrategien vorgestellt. Darauf aufbauend wird eine Kostenfunktion zur Optimierung aufgestellt, in der sowohl energetische und fahrdynamische Aspekte als auch die Gesamtfahrdauer zum Erreichen des Ziels berücksichtigt werden. Mit dieser lässt sich der optimale Geschwindigkeitsverlauf durch Kombination der realisierbaren Betriebsarten und Minimierung der anfallenden Kosten bestimmen. Der Algorithmus konnte an ausgewählten Fahrscenarien validiert werden. Simulationsergebnisse zeigen, dass der Energieverbrauch des Fahrzeugs durch den Einsatz des vorliegenden energieoptimalen Geschwindigkeitsprofils wesentlich reduziert werden kann.

1 Einleitung

Durch das steigende Umweltbewusstsein innerhalb der Gesellschaft gewinnt der Kraftstoffverbrauch von Fahrzeugen als Kaufkriterium zunehmend an Bedeutung. Wobei nicht mehr nur rationale Aspekte wie Kostenreduzierung, sondern zunehmend emotionale Aspekte (z.B. soziale Akzeptanz) eine entscheidende Rolle spielen [1]. Durch das alleinige Optimieren einzelner Komponenten lässt sich der Kraftstoffverbrauch jedoch nur innerhalb gewisser Grenzen reduzieren. Aus diesem Grund werden in neueren Forschungen ganzheitliche Ansätze verfolgt, um den Verbrauch mit Hilfe von intelligenten Betriebsstrategien zu verringern.

Insbesondere bei reinen Elektrofahrzeugen ist die Reichweitenerhöhung durch den Einsatz intelligenter Betriebsstrategien interessant, da diese bedingt durch die noch geringe Energiedichte der Akkumulatoren nur über eine geringe Reichweite verfügen. Deshalb liegt ein Forschungsschwerpunkt der Verfasser auf einem intelligenten, elektronischen Fahrzeugmanagement. Im vorliegenden Beitrag wird eine Teilfunktion dieses Fahrzeugmanagements, der Algorithmus eines energieoptimalen Geschwindigkeitsprofils für ein reines Elektrofahrzeug mit zwei radnahen Traktionsantrieben, vorgestellt.

Zunächst werden mögliche Betriebsarten der betrachteten Fahrzeugkonfiguration sowie deren Bedeutung für den Energieverbrauch aufgezeigt. Anschließend wird eine mögliche Optimierungsstrategie basierend auf der dynamischen Optimierung nach Bellman [2]

entwickelt. Anhand von beispielhaften Szenarien werden die Funktion des Algorithmus sowie das daraus resultierende Einsparpotential aufgezeigt.

2 Stand des Wissens

Die Entwicklung einer energieoptimalen Fahrzeuglängsführung baut auf vorhandenen Systemen zur automatisierten Fahrzeuglängsführung auf. Unter Zuhilfenahme eines elektronischen Horizonts und unter Berücksichtigung darstellbarer Betriebsstrategien werden derzeit von vielen OEM Assistenzfunktionen für eine energieoptimale Längsführung entwickelt.

In 2011 stellte die Porsche AG das Forschungsprojekt „InnoDrive“ vor [1], welches in Zusammenarbeit mit dem Karlsruher Institut für Technologie entwickelt wurde [3]. Dieses optimiert die Betriebsstrategie unter Verwendung von Fahrzeug- und Umfeldinformationen wie Fahrzeugposition, vorausliegendem Streckenverlauf, Erkennen vorausfahrender Fahrzeuge und Verkehrszeichen. Die vorliegenden Daten fließen gewichtet in eine Zielfunktion ein. Anschließend wird das Optimierungsproblem mit Hilfe der dynamischen Programmierung nach Bellman gelöst. Um die Echtzeitfähigkeit sicherzustellen, wird der betrachtete Geschwindigkeitsbereich beispielsweise anhand der maximal zulässigen Geschwindigkeit auf einen sinnvollen Bereich begrenzt. Bei der Optimierung werden u.a. Beschleunigung, Konstantfahrt, Schubabschaltung, Segeln, Rekuperation und elektrische Fahrt als Betriebsstrategien vorgesehen.

Einen ähnlichen Ansatz verfolgt die Volkswagen AG mit der Fahrstrategie „Green Driving“ [4]. Diese erweitert die Funktionalität des herkömmlichen ACC, wobei ausschließlich auf Daten der Eigenortung des Fahrzeugs sowie Informationen aus den Navigationskarten zurückgegriffen wird. Der aus diesen Daten generierte elektronische Horizont ermöglicht die Durchführung energieeffizienter Fahrmanöver.

Neuere Forschungsansätze verfolgen darüber hinaus die Integration dynamischer Daten aus Car-to-X-Kommunikation. Hieraus ergeben sich zusätzliche Informationen wie aktueller Status und Umschaltzeiten von Lichtsignalanlagen, Position und Geschwindigkeiten anderer Verkehrsteilnehmer sowie Verkehrsstaus. An der RWTH Aachen erfolgt die Anwendung dieser Informationen in einem zweistufigen Ansatz. Zunächst wird das durchschnittliche Fahrverhalten der umliegenden Fahrzeuge unter Nutzung der Car-to-Car-Kommunikation prädiziert. Anschließend wird die Geschwindigkeitstrajektorie für eine höhere Energieeffizienz optimiert [5].

Die Auslegung einer solchen Strategie erfolgt jedoch größtenteils an Kraftfahrzeugen mit Verbrennungsmotor [3; 6], weshalb die Übertragbarkeit und der mögliche Nutzen auf rein elektrisch angetriebene Fahrzeuge in diesem Beitrag geprüft wird.

3 Forschungsträger FREDY

Betrachtet wird der momentan an der Ostfalia Hochschule im Aufbau befindliche FREDY – der Funktionsträger für regenerative Elektromobilität und Fahrdynamikregelsysteme [7]. Der FREDY (vgl. Abbildung 1) ist ein an der Hinterachse angetriebenes Elektrofahrzeug mit einem Zweispannungsbordnetz. Der Antrieb wird durch zwei permanenterrregte Synchronmaschinen mit radnahen Getrieben realisiert.

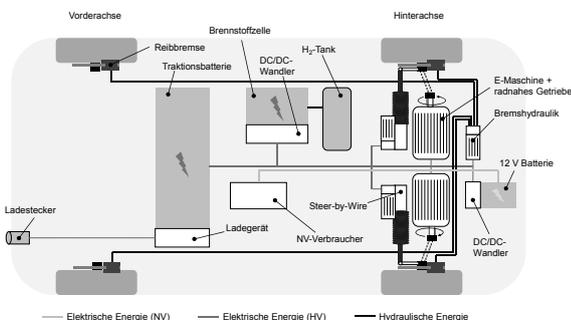


Abbildung 1: Fahrzeugtopologie des FREDY

3.1 Fahrzeugmanagement

Beim globalen Fahrzeugregelsystem handelt es sich um ein neuartiges elektronisches Fahrzeugmanagement [8]. Es handelt sich um die Integration einer Fahrdynamikregelung und eines elektrischen Energiemanagements. Ziel ist die Erzielung eines sicheren und dynamischen Fahrbetriebs bei gleichzeitig energieoptimaler Nutzung aller Aggregate in Hinblick auf eine Reichweitenmaximierung.

Als ein zentraler Baustein wird ein elektrisches Energiemanagementsystem [9] angesehen, welches für eine bedarfsgerechte und prioritätsbasierte Verteilung der im Elektrofahrzeug begrenzt verfügbaren elektrischen Leistung auf die elektrischen Verbraucher bei gleichzeitiger Gewährleistung von Fahrsicherheit und möglichst nicht wahrnehmbaren Einbußen im Fahrkomfort sorgt. Durch den Einsatz eines energieoptimalen Geschwindigkeitsprofils kann der Energiebedarf der Fahrzeuglängsbewegung für die betrachtete Strecke prädiziert werden, sodass das elektrische Energiemanagement die Verbraucher im Bordnetz energieoptimal betreiben kann. In Abbildung 2 wird eine mögliche Integration des energieoptimalen Geschwindigkeitsprofils in das elektrische Energiemanagement aufgezeigt.

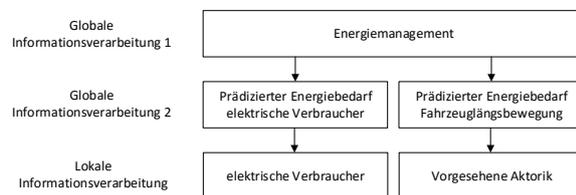


Abbildung 2: Einordnung in die globale Fahrzeugregelung

Zusammen mit dem prädizierten Energiebedarf elektrischer Verbraucher lässt sich der Energiebedarf des gesamten Fahrzeugs und mit Kenntnis des Batterieladestandes die Restreichweite prädizieren.

3.2 Betriebsarten

Aus der Topologie des elektrischen Antriebsstrangs ergeben sich nachfolgende Betriebsstrategien zur Beeinflussung der Längsdynamik.

- Geschwindigkeit halten

Die beiden Antriebsmotoren gleichen die Fahrwiderstände aus, um die Fahrzeuggeschwindigkeit konstant zu halten.

- Beschleunigen

Über das Maximalmoment der beiden Motoren von je $M_{max} = 38 \text{ Nm}$, die verwendete Getriebeübersetzung $i = 7$, den Reifenhalbmesser $r = 0,293 \text{ m}$ sowie die Fahrzeugmasse $m = 880 \text{ kg}$ kann auf die maximale Beschleunigung geschlossen werden. Diese beläuft sich auf $a_{max} = 2 \frac{\text{m}}{\text{s}^2}$. Im Bereich höherer Fahrzeuggeschwindigkeiten werden die Synchronmotoren im Bereich der Feldschwächung betrieben, weshalb das Beschleunigungsvermögen sinkt.

- Rekuperation

Wenn die elektrischen Antriebe als Generatoren betrieben werden, lassen sich das Fahrzeug verzögern und Energie zurückgewinnen. Diese Möglichkeit ist insbesondere bei höheren Geschwindigkeiten und somit hohen induzierten Spannungen interessant.

- Segeln

In diesem Fall wirken lediglich die Fahrwiderstände verzögernd auf das Fahrzeug.

- Bremsen

Um die hohen Energien bei Bremsingriffen sicher abzubauen zu können und um die Sicherheitsbestimmungen zum Betrieb von Kraftfahrzeugen [10] zu erfüllen, ist die Integration und Berücksichtigung einer zusätzlichen hydraulischen Bremsanlage erforderlich. Die integrierten Elektromotoren sind für das alleinige Abbremsen unterdimensioniert und eignen sich aus energetischer Sicht nicht für diesen Zweck.

4 Konzeption eines energieoptimalen Geschwindigkeitsprofils

In diesem Abschnitt wird die Konzeption einer intelligenten Fahrerassistenzfunktion, eines energieoptimalen Geschwindigkeitsprofils, beschrieben. Ziel dieser Funktion ist u.a. die Ermittlung einer optimalen Betriebsführung des FREDY für eine gegebene Strecke in Abhängigkeit einer vorgegebenen Gewichtung verschiedener Parameter. Das Hauptaugenmerk liegt hierbei auf dem Auffinden der optimalen Lösung.

4.1 Relevante Fahrphysik

Um einen Optimierungsalgorithmus entwickeln zu können, muss eine geeignete mathematische Beschreibungsform des Systems vorliegen.

In diesem Beitrag wird die Längsdynamik des Fahrzeugs betrachtet, welche dem Newton-Euler-Ansatz

entsprechend über die folgende Gleichung beschrieben werden kann:

$$\dot{v} = \frac{1}{m} \cdot F_M + \frac{1}{m} \cdot F_w(v) \quad (1)$$

Die Änderung der Geschwindigkeit hängt demnach von der durch die Antriebsmotoren aufgebrachte Längskraft F_M sowie den Fahrwiderständen F_w ab.

Als Fahrwiderstände werden Luft- und Rollwiderstand betrachtet. Diese wirken zu jeder Zeit auf das Fahrzeug und müssen daher bei jedem Betriebsfall berücksichtigt werden.

Der Rollwiderstand wird durch eine nichtlineare Abhängigkeit der Geschwindigkeit abgebildet, wodurch dessen überproportionales Ansteigen bei höheren Geschwindigkeiten berücksichtigt wird.

$$F_{roll} = 4 \cdot (f_{r0} + f_{r1} \cdot v + (f_{r4} \cdot v)^4) \quad (2)$$

Der Luftwiderstand ergibt sich aus dem Luftwiderstandsbeiwert des Fahrzeugs c_w , der Fahrzeugstirnfläche A , der Luftdichte ρ_{luft} sowie der Fahrzeuggeschwindigkeit v .

$$F_{luft} = \frac{1}{2} \cdot c_w \cdot A \cdot \rho_{luft} \cdot v^2 \quad (3)$$

Somit lassen sich die Fahrwiderstände, zumindest unter Annahme konstanter Straßenbedingungen und vernachlässigbarem Wind, über die bekannte Fahrzeuggeschwindigkeit berechnen.

$$F_w = F_{roll} + F_{luft} \quad (4)$$

4.2 Prinzipieller Lösungsansatz

Optimierungsprobleme können allgemein durch eine Vielzahl verschiedener Algorithmen gelöst werden, von denen viele jedoch nicht zwingend das globale Optimum finden. Ein Ansatz, der dies stets gewährleistet, stammt aus dem Fachgebiet der Kombinatorik.

Unter der Voraussetzung, dass es sich bei dem betrachteten System um eine Markov-Kette erster Ordnung handelt, spricht die Zukunft des Systems nur von der Gegenwart und nicht von der Vergangenheit abhängt, können Zustandsänderungen unabhängig von der Vergangenheit mit jeder Betriebsstrategie erfolgen. Dementsprechend können alle Kombinationsmöglichkeiten der verfügbaren Betriebsarten über den betrachteten Horizont miteinander verglichen werden. Somit wird sichergestellt, die optimale Betriebsstrategie zu finden.

Abbildung 3 illustriert beispielhaft die möglichen Zustände innerhalb einer vorgesehenen Wegstrecke.

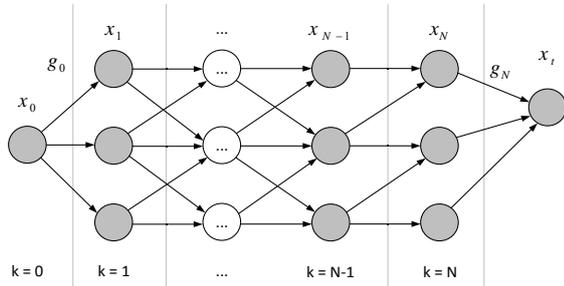


Abbildung 3: Zustandsgraph eines deterministischen endlichen Systems

Der Verbindung zwischen den einzelnen Zuständen werden Kosten g_i zugewiesen. Für jede Kombination können die Gesamtkosten berechnet werden, sodass die Fahrstrategie mit den geringsten Kosten ermittelt und als energieoptimales Profil gewählt werden kann.

Nachteilig an diesem Ansatz ist der so genannte *Fluch der Dimension*. An jeder Wegstützstelle werden x mögliche Zustandsänderungen untersucht. Bei N Wegstützstellen ergeben sich demnach x^N Kombinationsmöglichkeiten. Es ist ersichtlich, dass die Echtzeitanforderungen an ein solches System insbesondere in Abhängigkeit der betrachteten Wegstützstellen nicht mehr gewährleistet sind. Da es jedoch bei der Berechnung jeder möglichen Kombination stets Varianten gibt, die mehrfach berechnet werden, kann die Anzahl der Rekursionen durch den Einsatz der dynamischen Programmierung verringert werden.

Dem Bellmanschen Optimalitätsprinzip entsprechend setzt sich die optimale Lösung des Gesamtproblems aus den optimalen Teillösungen kleinerer Teilprobleme zusammen. Demgemäß werden bereits berechnete, optimale Lösungen bei der Berechnung des energieoptimalen Geschwindigkeitsprofils in einem Array gespeichert, sodass diese bei Erreichen einer identischen Ausgangsposition als Ergebnis verwendet werden können. Durch diese Erweiterung steigt die Zahl an Rekursionen pro zusätzlicher Wegstützstelle lediglich linear statt exponentiell an. Maximal werden lediglich $N \cdot x^2$ Aufrufe durchgeführt (vgl. [3]).

Auf den bereits in Abbildung 3 vorgestellten Zustandsgraph bezogen, kann das Prinzip wie folgt beschrieben werden: Die Zustände x_N können von allen vorherigen Zuständen x_{N-1} erreicht werden. Demzufolge würde jeder der möglichen Zustände von x_N für

jede bis dahin mögliche Kombination erneut aufgerufen. Die von x_N durchführbaren Zustandsänderungen zum Zielzustand liegen jedoch bereits nach dem ersten Aufruf als Optimum vor, weshalb diese nicht erneut berechnet werden müssen.

4.3 Auslegung eines Geschwindigkeitsprofils

Um die beschriebene Optimierungsstrategie auf ein Geschwindigkeitsprofil anwenden zu können, muss es als diskretes System vorliegen.

Anhand des zu fahrenden Streckenverlaufs wird ein diskretes Geschwindigkeit-Weg-Profil gebildet (Abbildung 4). Die Strecke wird in N diskrete Wegstützstellen von S_0 bis S_N eingeteilt. Weiterhin wird die Fahrzeuggeschwindigkeit an jeder Wegstützstelle diskretisiert (in diesem Beispiel mit einer Schrittweite von 5 km/h). Somit ergibt sich auf jeder Wegstützstelle eine Vielzahl möglicher Geschwindigkeitsknoten.

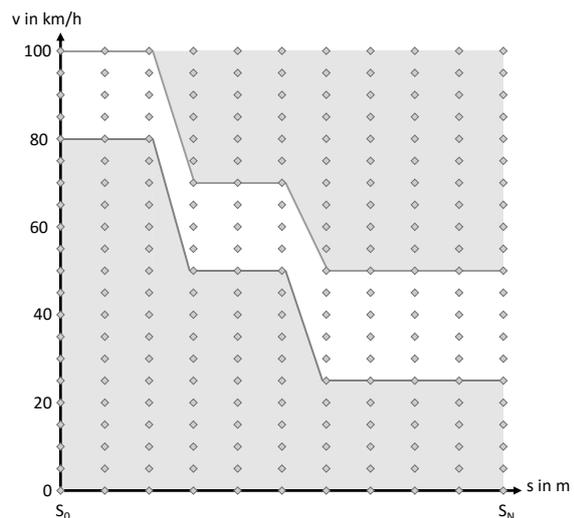


Abbildung 4: Exemplarisches Geschwindigkeit-Weg-Profil für die Fahrt von der Landstraße in die Stadt

Die Geschwindigkeit an jeder Wegstützstelle wird durch eine obere und untere Geschwindigkeit eingegrenzt, um sowohl nicht erreichbare als auch nicht sinnvolle Geschwindigkeiten als mögliche Lösungen auszuschließen.

Die Begrenzung der maximalen Geschwindigkeit erfolgt in diesem Beitrag durch die an der jeweiligen Wegstützstelle zulässige Höchstgeschwindigkeit. Die minimale Geschwindigkeit orientiert sich an der maximalen Geschwindigkeit und soll verhindern, dass ein energetisch vorteilhafter aber extrem langsamer Fahrbetrieb eingestellt wird, wodurch der Gesamtverkehr

deutlich gestört werden würde. Durch diese Einschränkungen ergibt sich ein wie in Abbildung 4 dargestellter zulässiger Lösungsraum innerhalb der roten und grünen Begrenzungen. Dieser Bereich wird durch den Optimierungsalgorithmus verarbeitet.

4.4 Betriebsstrategien

Für den Optimierungsalgorithmus müssen weitere Informationen über die maximal mögliche Geschwindigkeitsänderung Δv vorliegen um bestimmen zu können, ob das Fahrzeug seine Geschwindigkeit v_0 an der Position s_0 innerhalb der Wegdifferenz Δx zu einer Geschwindigkeit v_1 an Position s_1 ändern kann.

Bekannte Größen sind die durch das Fahrzeug in den verschiedenen Betriebsarten abhängig von der Fahrzeuggeschwindigkeit v erzielbare Beschleunigung bzw. Verzögerung a , die Geschwindigkeit v_0 an der Start-Wegstützstelle und die Wegdifferenz Δx zur nächsten Wegstützstelle.

Aus zweimaliger Integration der Differentialgleichung für die Fahrzeuglängsbewegung (Gleichung 5) folgt der vom Fahrzeug zurückgelegte Weg

$$s(t) = s_0 + v_0 \cdot t + \frac{1}{2} a \cdot t^2 \quad (5)$$

unter Berücksichtigung einer Startposition s_0 und einer Startgeschwindigkeit v_0 .

Zur Berechnung der gefahrenen Zeit t_Δ zwischen den Wegstützstellen s_0 und s_1 wird $s(t = t_\Delta) = s_1$ gesetzt und Gleichung 4 nach t_Δ aufgelöst. Unter Berücksichtigung des Zusammenhangs $s_1 - s_0 = \Delta x$ und der bekannten Beschleunigung a gilt Gleichung 6:

$$t_\Delta = -\frac{v_0}{a} \pm \sqrt{\left(\frac{v_0}{a}\right)^2 + \frac{2 \cdot \Delta x}{a}} \quad (6)$$

Die maximale Geschwindigkeitsänderung Δv in Δx kann mithilfe der Zeit t_Δ und der geschwindigkeitsabhängigen Beschleunigung a nach Gleichung 7 berechnet werden:

$$\Delta v = a \cdot t_\Delta \quad (7)$$

Mithilfe dieses Zusammenhangs kann ein Kennfeld berechnet werden, welches die maximale Geschwindigkeitsänderung abhängig von der Fahrzeuggeschwindigkeit v und der Wegdifferenz Δx wiedergibt.

Abbildung 5 zeigt die durch die einzelnen Betriebsarten des FREDY maximal realisierbaren Geschwindigkeitsänderungen bei $\Delta x = 50$ m.

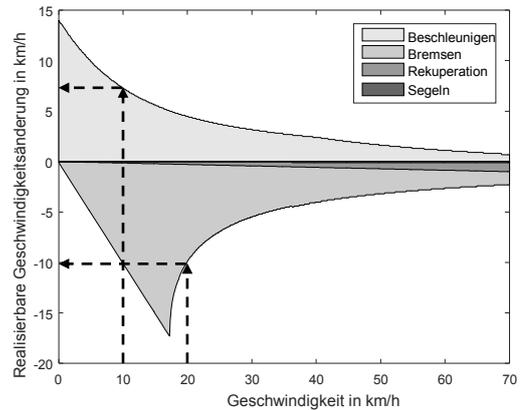


Abbildung 5: Realisierbare Geschwindigkeitsänderung über Fahrzeuggeschwindigkeit abhängig von gewählter Betriebsart bei einer Beschleunigungsstrecke von 50 m

Die folgenden beiden Beispiele verdeutlichen die Aussage der Abbildung:

1. Bei einer Geschwindigkeit von 10 km/h soll das Fahrzeug beschleunigt werden. Innerhalb einer Strecke von 50 m kann eine Geschwindigkeitserhöhung um maximal 7,5 km/h erzielt werden.
2. Bei einer Geschwindigkeit von 20 km/h soll das Fahrzeug gebremst werden. Die maximale Geschwindigkeitsreduktion beträgt 10 km/h durch Einsatz des rekuperativen Bremsens.

Aus der Abbildung wird deutlich, dass die darstellbare Geschwindigkeitsänderung mit steigender Fahrzeuggeschwindigkeit abnimmt. Dies ist zum einen auf die geringer werdende Zeitdauer zwischen zwei Stützstellen und zum anderen auf das geringere Beschleunigungsvermögen der Elektromotoren (Feldschwächetrieb) zurückzuführen. Weiterhin wurde die maximale Geschwindigkeitsänderung beim Bremsen auf die Fahrzeuggeschwindigkeit begrenzt.

Für den Optimierungsalgorithmus wird die Betriebsart B in einer Matrix (Gleichung 8) bereitgestellt. Der Zeilenindex Δv_i stellt die mögliche Geschwindigkeitsänderung von Δv_{max} bis Δv_{min} , der Spaltenindex v_j die Fahrzeuggeschwindigkeit von v_{min} bis v_{max} dar.

$$\begin{bmatrix} B_{\Delta v_0, v_0} & B_{\Delta v_0, v_1} & B_{\Delta v_0, v_2} & \cdots & B_{\Delta v_0, v_n} \\ B_{\Delta v_1, v_0} & B_{\Delta v_1, v_1} & B_{\Delta v_1, v_2} & \cdots & B_{\Delta v_1, v_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ B_{\Delta v_m, v_0} & B_{\Delta v_m, v_1} & B_{\Delta v_m, v_2} & \cdots & B_{\Delta v_m, v_n} \end{bmatrix} \quad (8)$$

4.5 Definition einer Kostenfunktion

Allgemein adressiert die Assistenzfunktion die Fahrsicherheit, den Fahrkomfort, die Energieeffizienz und die Fahrdynamik. Bei der Entwicklung einer Fahrerassistenzfunktion muss somit sichergestellt werden, dass die oben genannten Aspekte ausreichend berücksichtigt werden. Die Fahrsicherheit wird bereits bei der Erstellung des zulässigen Geschwindigkeitsschlauchs berücksichtigt. Ebenso wird der Fahrkomfort dort durch Begrenzungen der maximal zulässigen Beschleunigungen vorgesehen. Somit bleiben zum Aufstellen der Gütefunktion zwei miteinander konkurrierende Zielgrößen – die Fahrdynamik und die Energieeffizienz. Die Fahrdynamik wird durch die Fahrdauer, die Energieeffizienz durch den Kraftstoffverbrauch repräsentiert.

Die Kosten der Fahrdauer werden über die Ausgangs- und Folgegeschwindigkeiten v_1 und v_2 , die Wegdifferenz Δx zwischen den beiden Wegstützstellen sowie den Gewichtungsfaktor g_t , wie in Gleichung (9) gezeigt, berechnet.

$$K_t = g_t \cdot \frac{2 \cdot \Delta x}{v_1 + v_2} \quad (9)$$

Die Werte dieser Funktion sinken bei höheren Geschwindigkeiten, wodurch sich hohe Geschwindigkeiten positiv auf die Kosten auswirken. Ein hoher Gewichtungsfaktor bewirkt hier, dass eine schnellere Durchschnittsgeschwindigkeit gewählt wird und energetische Aspekte weniger Berücksichtigung finden.

Der Energieverbrauch wird anhand der vorher aufgestellten Bewegungsgleichungen des Fahrzeugs ermittelt. Aus der Definition, verrichtete Arbeit ist das Integral der Kraft in Richtung des Weges \vec{s} , kann die zum Überwinden der Fahrwiderstände \vec{F}_W benötigte Energie

$$W_w = \int_0^{\vec{x}} \vec{F}_W(\vec{s}) \cdot d\vec{s} \quad (10)$$

und die zum Beschleunigen des Fahrzeugs erforderliche Energie

$$W_B = \frac{1}{\eta_{Antrieb}} \cdot \int_0^{\vec{x}} \vec{F}_B(\vec{s}) \cdot d\vec{s} \quad (11)$$

unter Berücksichtigung des Antriebsstrangwirkungsgrades $\eta_{Antrieb}$ und der Beschleunigungskraft \vec{F}_B berechnet werden. Die Fahrwiderstände wirken jederzeit auf das Fahrzeug und müssen somit bei jedem Be-

triebsfall berücksichtigt werden. Insbesondere bei hohen Geschwindigkeiten resultiert aus ihnen ein hoher Energieverbrauch.

Die Kosten für den Energiebedarf des Fahrzeugs werden aus der Summe der Arbeiten W_w und W_B multipliziert mit dem Gewichtungsfaktor g_v ermittelt:

$$K_v = g_v \cdot (W_w + W_B) \quad (12)$$

Um verschiedene Betriebsstrategien energetisch vergleichen zu können und ein Optimum zu finden, wird eine Kostenfunktion J_β eingeführt.

$$J_\beta(s) = \int_0^{s_N} (K_t(s) + K_v(s)) \cdot ds \quad (13)$$

Die Kosten werden in Abhängigkeit der gewählten Betriebsstrategie β über der betrachteten Gesamtstrecke bis s_N aufsummiert. Diese Kosten dienen die Grundlage für die Optimierungsstrategie, welche die Kosten minimieren soll.

5 Realisierung

Die Realisierung des Geschwindigkeitsprofils erfolgt im CAE-Werkzeug MATLAB[®] und wird folgend vorgestellt. Zunächst erfolgt die Einordnung der Assistenzfunktion in die globale Fahrzeugregelstrategie.

5.1 Programmablauf

Abbildung 6 zeigt einen Programmablaufplan (PAP) des Hauptprogramms.

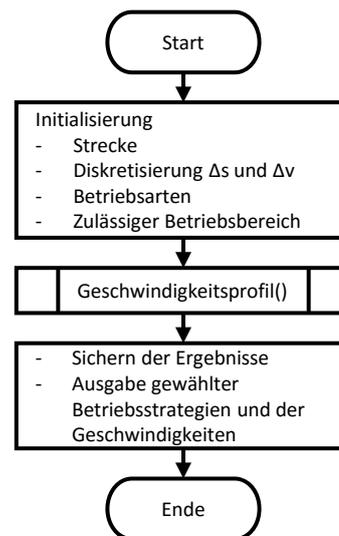


Abbildung 6: PAP des Hauptprogramms

Im ersten Schritt erfolgt die Initialisierung des Programms. Hierbei werden Parameter der vorliegenden

Strecke, der Betriebsarten sowie des zulässigen Betriebsbereichs berechnet. Anschließend werden durch Aufruf der Funktion *Geschwindigkeitsprofil()* zwei Schleifen zur Ermittlung des Geschwindigkeitsprofils ausgeführt. Am Ende ruft sich die Funktion rekursiv selbst erneut auf.

In jedem Funktionsaufruf wird genau eine Betriebsstrategie bzw. eine Kombination vorhandener Strategien gewählt. Anschließend wird die Geschwindigkeitsfunktion (Rekursion) für die nächste Wegstelle aufgerufen, sodass überprüft werden kann, welcher Weg der kostengünstigste ist.

Sobald für eine Wegstelle die Kosten für alle Differenzgeschwindigkeiten vorliegen, wird überprüft, welche die mit den niedrigsten Kosten ist. Diese wird als das optimale Ergebnis an der betrachteten Stelle angesehen. Wird diese Stelle wieder erreicht, wird der dort gespeicherte Wert verwendet und somit zusätzliche Rekursionen vermieden.

Da es sich um eine übergeordnete, globale Betriebsstrategie handelt, ist die Berechnung des Geschwindigkeitsprofils nicht in jeder Millisekunde notwendig. Die Häufigkeit der Berechnungen ist dann ausreichend, solange Änderungen der Streckenbegebenheiten zuverlässig berücksichtigt werden.

5.2 Rechenzeit

Bei einer Diskretisierung der Geschwindigkeit von 0,1 km/h dauert die Berechnung des Profils derzeit zwischen 3 und 6 Sekunden. Dies ist abhängig von der Wahl der Anfangswerte und Randbedingungen. Wenn die betrachteten Geschwindigkeitsunterschiede zu 1 km/h angenommen werden, verringert sich die Berechnungszeit auf unter 0,1 Sekunde. Gleichzeitig sinkt jedoch die Güte der berechneten Trajektorien.

In weiteren Arbeiten ist die maximal zulässige Rechenzeit zu spezifizieren und anhand derer der Algorithmus zu optimieren.

5.3 Validierung durch Simulation

Abbildung 7 stellt verschiedene Geschwindigkeitstrajektorien bei Variation des zeitlichen Gewichtungsfaktors gegenüber. Die Geschwindigkeit wurde in diesem Fall in 0,1 km/h Schritten diskretisiert. Bei den einzelnen Wegstützstellen wird die verwendete Betriebsart angezeigt.

- 1 Geschwindigkeit halten
- 2 Segeln

- 3 Bremsen
- 4 Beschleunigen
- 5 Rekuperation

Somit lassen sich der Geschwindigkeitstrajektorie passende Betriebsarten zuordnen.

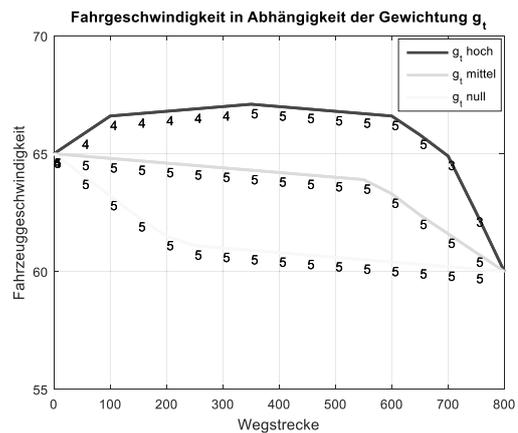


Abbildung 7: Geschwindigkeitstrajektorie bei Variation der Gewichtung

Bei der Simulation werden die gleichen Start- und Zielwerte verwendet. Die Abbildung spiegelt den Einfluss des zeitlichen Gewichtungsfaktors wie erwartet wider. Wenn dieser hoch gewichtet wird, beschleunigt das Fahrzeug zunächst und bremst am Ende innerhalb kurzer Zeit stark ab. Wird der Zeitfaktor gleich 0 gesetzt, werden lediglich energetische Aspekte berücksichtigt. In diesem Fall erzielt das Fahrzeug das energetisch beste Ergebnis (vgl. Tabelle 1).

Tabelle 1: Energiebedarf in Abhängigkeit von g_t

Gewichtung g_t	Energiebedarf E in kJ
niedrig	-26,67
mittel	-26,66
hoch	120,60

In Abbildung 8 sind die Ergebnisse einer Simulation mit den gleichen Randbedingungen wie zuvor, jedoch mit einer veränderten Geschwindigkeitsdiskretisierung von $\Delta v = 1 \text{ km/h}$, dargestellt.

Abbildung 8 zeigt einen ähnlichen Verlauf wie Abbildung 7, jedoch ist die Wahl der Betriebsarten sehr eingeschränkt. Dementsprechend wird bei dem betrachteten Beispiel auch ein deutlich schlechterer Verbrauch erreicht (vgl. Tabelle 2).

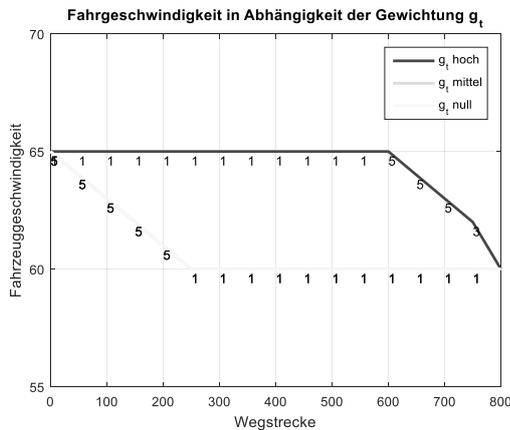


Abbildung 8: Geschwindigkeitstrajektorie bei größerer Geschwindigkeitsdiskretisierung ($\Delta v = 1 \text{ km/h}$)

Tabelle 2: Energiebedarf in Abhängigkeit von g_t

Gewichtung g_t	Energiebedarf E in kJ
niedrig	24,00
mittel	24,00
hoch	46,70

6 Resümee

In diesem Beitrag wurde die Auslegung des Algorithmus eines optimalen Geschwindigkeitsprofils als intelligente Fahrerassistenzfunktion für den energieoptimalen Fahrbetrieb vorgestellt. Die Funktionalität dieses energieoptimalen Geschwindigkeitsprofils wurde anhand mehrerer Testszenarien erfolgreich validiert. Damit einhergehend wurde der Energieverbrauch für die betrachtete Strecke durch geeignete Auswahl der zur Verfügung stehenden Betriebsarten wesentlich reduziert. Somit konnte ein Schritt in Richtung der energieoptimalen Fahrt gemacht werden.

In nächsten Arbeiten wird das Erstellen der oberen und unteren Grenzen im diskreten Geschwindigkeitsprofil fokussiert, sodass dieses die Streckengegebenheiten realitätsnäher nachbildet. Berücksichtigt werden sollen bspw. Umfelddaten wie Kreuzungen oder Lichtsignalstellen aber auch fahrdynamische Aspekte wie die Begrenzung der Maximalgeschwindigkeit in Kurven zur Einhaltung komfortabler Querbeschleunigungen.

Ein weiterer Schwerpunkt liegt in der flexiblen Wahl der Wegdifferenz Δx zwischen zwei Wegstützstellen, sodass die Rechenleistung optimiert werden kann. Auf

einer geraden Strecke kann z.B. eine höhere Wegdifferenz als in einer Kurve oder an einer Kreuzung verwendet werden.

Nach weiteren Optimierungsarbeiten soll das energieoptimale Geschwindigkeitsprofil in das bestehende Energiemanagementsystem eingebettet werden, um die sich ergebenden Synergien und Potentiale bewerten zu können.

7 Danksagung

Dieser Beitrag wurde im Rahmen des Forschungsprojekts *Energy Consumption Optimization for electrical Cars* (kurz *ECOCar*) durch das Bundesministerium für Bildung und Forschung unter dem Förderkennzeichen 03FH03513 gefördert. Die Verantwortung für den Inhalt liegt bei den Autoren. Für die Förderung bedanken sich diese herzlichst.



GEFÖRDERT VOM



Im aktuellen Dokument sind keine Quellen vorhanden.

8 Literaturverzeichnis

- [1] Markschläger, P.; Wahl, H.-G.; Weberbauer, F. und Lederer, M.: Assistenzsystem für mehr Kraftstoffeffizienz, *ATZ* 114 (2012) H. 11. S. 850–855.
- [2] Bellman, R.: *The theory of dynamic programming*. Santa Monica 1954.
- [3] Radke, T.: *Energieoptimale Längsführung von Kraftfahrzeugen durch Einsatz vorausschauender Fahrstrategien*. Dissertation, Karlsruher Institut für Technologie 2013.
- [4] Dornieden, B.; Junge, L. und Pascheka, P.: *Vorausschauende Energieeffiziente Fahrzeuglängsregelung*, *ATZ* 114 (2012) H. 3. S. 230–235.
- [5] Themann, P.; Zlocki, A. und Eckstein, L.: *Energieeffiziente Fahrzeuglängsführung durch V2X-Kommunikation*. In: *Fahrerassistenzsysteme und Effiziente Antriebe*. S. 63–67.
- [6] Gausemeier, S. F.: *Ein Fahrerassistenzsystem zur prädiktiven Planung energie- und zeitoptimaler Geschwindigkeitsprofile mittels Mehrzieloptimierung*. Dissertation, Universität Paderborn 2013.

- [7] Liu-Henke, X.; Göllner, M.; Fritsch, M.; Feind, R.; Quantmeyer, F. und Buchta, R.: FreDy – an electric vehicle with intelligent chassis control systems. 10th International Conference on Ecological Vehicles and Renewable Energies (EVER'15), Monte Carlo, Monaco, March 31 - April 2 2015.
- [8] Liu-Henke, X.; Quantmeyer, F. und Scherler, S.: Hierarchical Vehicle Management for Battery-Electric Vehicles with Decentralized Drives. 10th International Conference on Ecological Vehicles and Renewable Energies (EVER'15), Monte Carlo, Monaco, March 31 - April 2 2015.
- [9] Scherler, S.; Quantmeyer, F. und Liu-Henke, X.: Modellbasierte Entwicklung des elektrischen Energiemanagements für Elektrofahrzeuge mit einem Zweispannungsbordnetz. Workshop der ASIM/GI-Fachgruppen, Stralsund, Germany, June 18 - 19 2015.
- [10] Deutsche Gesetzliche Unfallversicherung: DGUV Vorschrift 71 - Unfallverhütungsvorschrift Fahrzeuge 1997.

PC-basierte Simulation heterogener Modelle in der automobilen Software-Entwicklung

Gernot Deutenberg¹, Martin Rühl¹

¹dSPACE GmbH

GDeutenberg@dSPACE.de

Die Entwicklung von automobiler Steuergeräte-Software verlangt nach Tests in unterschiedlichen Domänen, vom Antriebsstrang bis zu Komfortfunktionen. Die Absicherung solcher Software erfordert die Simulation heterogener Modelle wie auch der Steuergeräte-Software. Zudem steigt der Grad der Komplexität der Software ständig an. Mit den bisher verwendeten Methoden wie Hardware-in-the-Loop (HIL)-Simulation lassen sich aber Szenarien, wie sie zum Beispiel bei der Entwicklung von Fahrerassistenzsystemen auftreten, nicht vollständig absichern. Wiederverwendbarkeit bestehender Modelle spielt hier genauso eine Rolle wie die Reproduzierbarkeit von Simulationsergebnissen. Für reine Software-in-the-Loop (SIL)-Tests bestehen oft keine Echtzeitanforderungen. Eine hierdurch mögliche erhöhte Ausführungsgeschwindigkeit der Simulation, das heißt schneller als Echtzeit, ermöglicht erhöhte Testdurchsätze.

Um diesen Anforderungen gerecht zu werden, bedarf es nicht nur einer geeigneten Simulationssoftware, sondern auch Schnittstellen für den Austausch der Modelle und der Simulationsdaten. Eine wichtige Rolle spielt hier die Standardisierung, ob durch AUTOSAR für die Steuergeräte-Software oder das Functional Mock-Up Interface (FMI) für Simulationsmodelle. Dieser Beitrag stellt die Anforderungen im Kontext dar und zeigt, wie die sogenannte Virtuelle Absicherung durch PC-basierte Simulation derartige Herausforderungen angeht.

Mit einem Praxisbeispiel wird gezeigt, wie diese Anforderungen umgesetzt werden und wie hierdurch die Serienreife der Software früher als bisher erreicht werden kann.

1 Absicherung von automobiler Software heute

Mit dem ständig wachsenden Anteil an Software in modernen Fahrzeugen steigen auch die Anforderungen an die Absicherung der implementierten Funktionalitäten. Das trifft insbesondere bei komplexen und sicherheitsrelevanten Funktionen zu, wie sie bei modernen Fahrerassistenzsystemen vorkommen. Diese Funktionen sind auch vermehrt nicht mehr nur auf einem Steuergerät (Electronic Control Unit, ECU) umgesetzt, sondern verteilen sich über mehrere ECUs. Mit herkömmlichen Testmethoden auf Komponenten-, Modell- und System-Niveau wird hier keine ausreichende Qualitätssicherung mehr erreicht bzw. erfolgt die Absicherung sehr spät im Entwicklungszyklus. Je später jedoch Fehler oder Qualitätsmängel gefunden werden, desto schwerwiegender sind die Auswirkungen und auch die Kosten zur Behebung steigen sehr stark an. Als Faustregel für den Anstieg der Kosten kann hier die sogenannte Zehnerregel aus der Qualitätssicherung genutzt werden.

Diese Erfahrungsregel besagt, dass die Kosten für die Behebung eines Fehlers mit jeder Stufe des Entwicklungszyklus um den Faktor 10 ansteigen [1]. Es ist naheliegend, diesen Kosten durch frühe Qualitätssicherungsmaßnahmen entgegenzuwirken. Die Randbedingungen der bisher verwandten Testmethoden kollidieren allerdings teilweise mit solchen zeitlichen Ansätzen. Hardware-Prototypen stehen erst recht spät im Entwicklungszyklus zur Verfügung und eine Integration auf Systemniveau ist meist erst mit diesen Prototypen möglich. Sicherheitsrelevante oder besonders komplexe Funktionen bringen zusätzliche – auch zeitliche – Auflagen mit sich.

Deshalb hat sich die Virtuelle Absicherung in den letzten Jahren als Methode etabliert und kommt vermehrt zum Einsatz. Dabei werden die herkömmlichen Testmethoden entlastet und durch neue ergänzt, um den oben genannten Anforderungen zu begegnen.

2 Stand der Technik in der Software-Absicherung

Entsprechend den Entwicklungsschritten im V-Modell basiert die bisherige Absicherung der Software im Entwicklungsprozess oft auf X-in-the-Loop (XIL)-Testmethoden (siehe Abbildung 1). Dabei wird das zu testende System X in einer Schleife (Loop) zusammen mit einem entsprechenden Strecken- oder Umgebungsmodell zu Testzwecken simuliert. Während der Modellierung werden einzelne Software-Komponenten mittels Model-in-the-Loop (MIL) getestet. Sobald der Seriercode dieser Modelle zur Verfügung steht, kommt Software-in-the-Loop (SIL) zum Einsatz. Ein Prototyp oder Muster des realen Steuergerätes wird schließlich per Hardware-in-the-Loop (HIL) getestet.

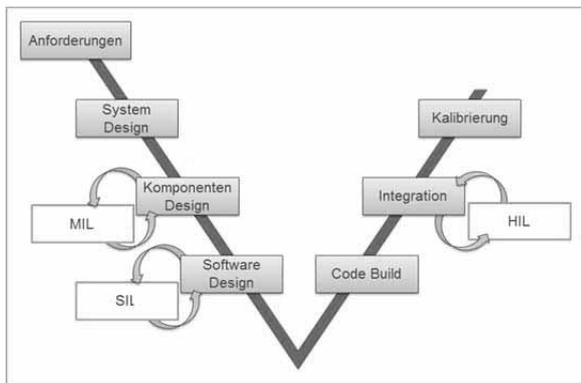


Abbildung 1: XIL-Methoden im V-Modell

Auf diese Weise kann die zu entwickelnde Software schon zu einem hohen Reifegrad gebracht werden, bevor sie in einem Versuchsfahrzeug verwendet wird. Um effizient MIL- oder SIL-Tests durchzuführen, musste bisher die Simulation der Software-Komponenten aber in derselben Modellierungsumgebung durchgeführt werden, in der auch andere Modelle, wie zum Beispiel das Umgebungsmodell, entworfen wurden. Dadurch wird die Software-Entwicklungsumgebung auch zur Simulations- und Testumgebung. Für Umgebungsmodelle, die mehrere Domänen abdecken müssen, kann das bereits Einschränkungen in Hinblick auf den Realitätsgrad bedeuten. Andersherum wird die Spezialisierung eines Werkzeugs auf eine Domäne erschwert, falls auch konträre Anforderungen aus anderen Domänen umgesetzt werden müssen. Die herkömmlichen Methoden stellen also zumeist einen Kompromiss dar, der zwar ein Testen erst ermöglicht, jedoch nicht die Abde-

ckung und Testtiefe liefern kann, die mittlerweile erforderlich ist.

Sobald die Hardware verfügbar ist, kann durch HIL-Test die Absicherung erfolgen. Diese Methode der Vorverlagerung von Tests aus dem Versuchsfahrzeug in das Labor hat sich als zuverlässig und wichtig erwiesen. Aufgrund der stark ansteigenden Komplexität und Vernetzung der ECUs steigt aber auch der Testumfang an und ein spätes Aufdecken eines Software-Fehlers führt zu deutlich höheren Kosten, als wenn dieser Fehler bereits früher gefunden worden wäre. Um den Fehler jedoch durch Tests zu finden, ist eine Integration der Software auf die Hardware notwendig, was durch die relativ späte Verfügbarkeit der Hardware mit den zeitlichen Anforderungen kollidiert. Eine späte Integration der Software birgt zudem das Risiko, durch den großen Umfang der Integration Fehler nicht mehr direkt auf die verursachende Komponente zurückführen zu können. Einige Szenarien erfordern zudem hohe Reproduzierbarkeit, das ist zum Beispiel der Fall, wenn unterschiedliche Varianten eines Modells getestet werden müssen. Diese Reproduzierbarkeit ist in einem Versuchsfahrzeug so nicht herzustellen. Ein Echtzeitsystem hat zudem eine systemimmanente Begrenzung der Skalierbarkeit. Sobald alle benötigten Modelle unter Berücksichtigung der Echtzeitbedingungen ausgeführt werden können, kann durch eine Aufrüstung des Systems keine Verringerung der benötigten Zeit zur Simulation mehr herbeigeführt werden. Das HIL-System wird aufgrund von Echtzeitanforderungen schnell an seine Auslastung gebracht. Um den Testdurchsatz zu erhöhen, wird hier also eine weitere Vorverlagerung notwendig, diesmal auf den PC und ohne Echtzeitanforderungen. Hier muss nach der Berechnung eines Zeitschrittes nicht auf den nächsten Abtastschritt gewartet werden, so dass sich die PC-Hardware optimal nutzen lässt. Im Endeffekt kann dadurch unter Verwendung der gleichen Modelle eine bessere Auslastung der Hardware erreicht und somit der Testdurchsatz erhöht werden. Das macht bisherige Absicherungsverfahren aber keineswegs überflüssig, das heißt, es werden weiterhin HIL-Tests als auch Versuchsfahrzeuge benötigt. Ein Teil der steigenden Umfänge bei der Absicherung kann aber schon frühzeitig abgedeckt werden. So lassen sich Folgefehler vermeiden oder deren Anzahl reduzieren und die Auslastung späterer Absicherungsverfahren verringern. Insgesamt steigt der Reifegrad der Software.

3 Schnittstellen und Standardisierung

Mit AUTOSAR lassen sich Komponenten entwickeln und über vordefinierbare Schnittstellen integrieren. Dadurch kann nicht nur der Entwicklungsprozess deutlich strukturierter ausgelegt werden, es können auch wesentlich einfacher Komponenten, die nicht selbst entwickelt wurden, eingebunden werden. Weiter ist es möglich, von der Mikrocontroller-Architektur zu abstrahieren und so große Teile der Software einer oder mehrerer ECUs bereits frühzeitig auf dem PC zu simulieren.

kann aber auch auf Signalebene abgebildet werden. Diese Konkretisierung ist ein wichtiger Schritt zu einer realistischeren Repräsentation des echten Steuergerätes.

Solche Modelle werden auch virtuelle Steuergeräte oder Virtual ECUs (V-ECUs) genannt.

Abbildung 3 zeigt eine detailliertere Sicht auf das AUTOSAR-Schichtenmodell. Darin ist die Applikationsschicht mit den einzelnen Software-Komponenten durch eine Laufzeitumgebung von der Basissoftware getrennt. Während zumeist Zulieferer die Basissoftware für reale ECUs erstellen, liegt die Verantwor-

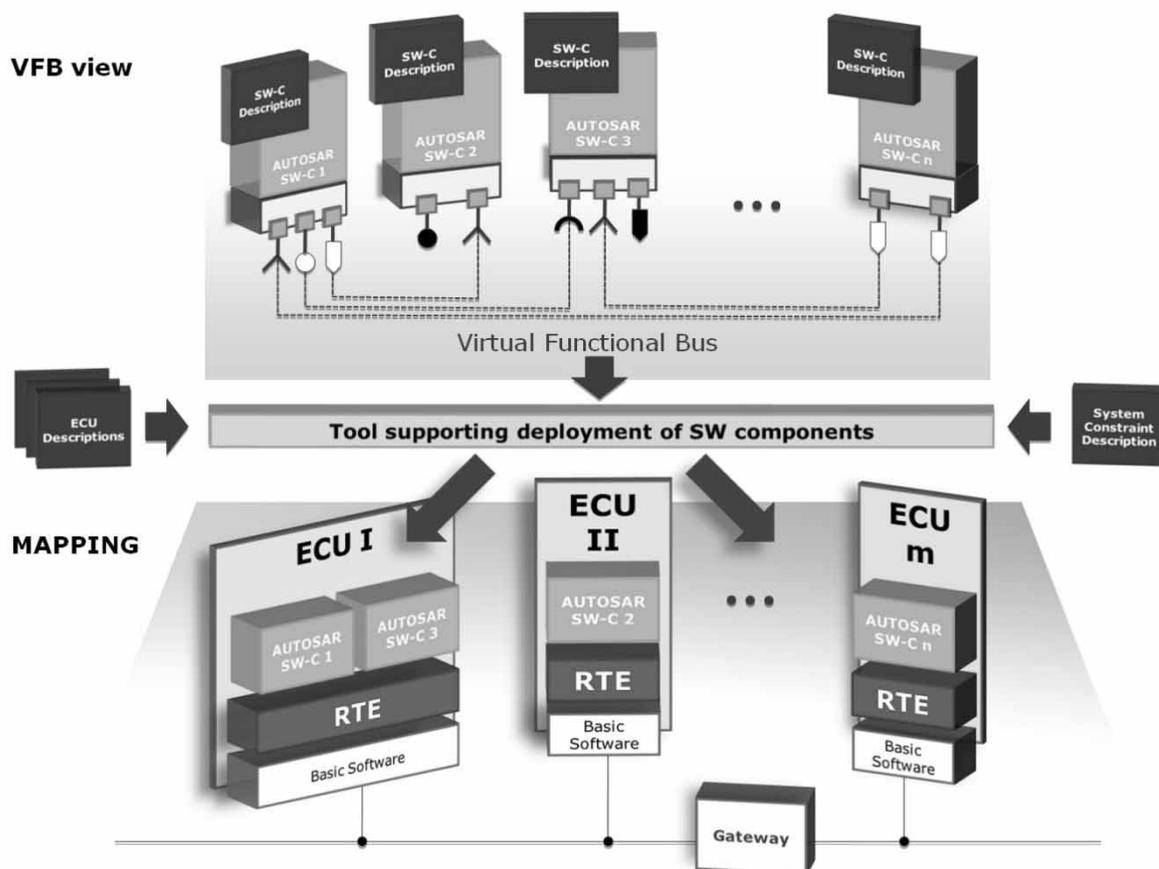


Abbildung 2: ECU Mapping nach AUTOSAR [2]

Wie aus einer Software-Architektur ein System aus vernetzten ECUs nach AUTOSAR wird, ist in Abbildung 2 zu sehen. Während auf Architektur-Niveau (Virtual Functional Bus view) weder das Mapping der Komponenten auf die ECUs noch die Art, wie diese kommunizieren, von Bedeutung ist, kommt nach der Verteilung auf konkrete ECUs die Kommunikation der ECUs untereinander hinzu. Diese erfolgt dann zum Beispiel über Bussysteme wie den CAN-Bus,

tung für einzelne Komponenten oder ganze Software-

Architekturen oft bei den Herstellern (OEMs) selbst. Die Software-Integration der einzelnen Komponenten oder auch die Integration der Applikationsschicht mit einem Basissoftware-Stack obliegt ebenso oft dem Hersteller selbst.

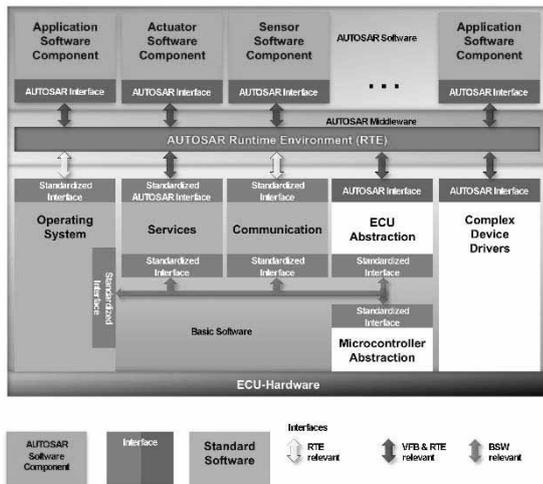


Abbildung 3: Das AUTOSAR-Schichtenmodell [2]

Der FMI (Functional Mock-up Interface)-Standard [3] definiert über FMI einen in sich konsistenten Container, um ganze Modelle plattformunabhängig zu verpacken und so auf unterschiedlichen Plattformen simulieren zu können. So kann man Strecken- und Umgebungsmodelle aus unterschiedlichen Domänen und den dazugehörigen Modellierungsumgebungen exportieren, um sie später auf einer Plattform zusammen zu simulieren. Im Gegensatz zum Stand der Technik können nun Modellierungsumgebung und Simulations- beziehungsweise Testumgebung unabhängig voneinander gewählt werden.

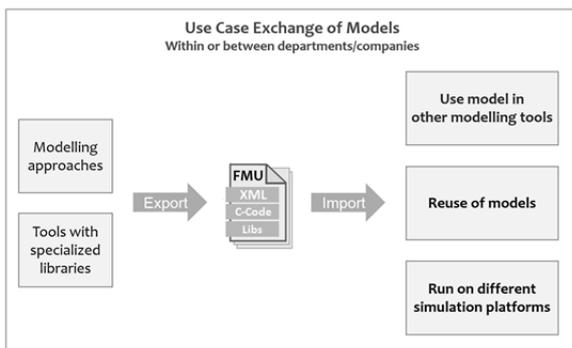


Abbildung 4: Modellaustausch mit FMI

Plattformunabhängige FMUs tragen dazu bei, die durchgängige Verwendung von Testbibliotheken inklusive der benötigten Modelle über mehrere Phasen des Entwicklungsprozesses zu etablieren. In Verbindung mit virtuellen Steuergeräten entstehen so realitätsnahe Simulationen. Ausgeführt auf einem PC, stehen sie bereits dem Funktionsentwickler zur Ver-

fügung, wodurch der Reifegrad der damit getesteten Software frühzeitig deutlich erhöht werden kann.

Diese Tests werden entweder manuell bzw. teilautomatisiert oder vollautomatisiert ausgeführt. Sowohl die Parametrierung als auch das Messen von Größen im Modell gehören aus Testsicht zu den Standardanforderungen. Werden die Modelle auf anderen Plattformen wie dem HIL-System weiterverwendet, ist die Verwendung standardisierter Schnittstellen von erheblichem Vorteil.

Für die Messung und Kalibrierung von realen Steuergeräten wird meist das vom ASAM e.V. standardisierte XCP-Protokoll verwendet. Als Transportschicht des Protokolls umfasst der Standard automobiler Bussysteme wie CAN oder FlexRay, aber auch Netzwerkprotokolle wie Ethernet. Als Zweischichtenprotokoll ausgelegt, kann man bei gleichbleibender Protokollschicht je nach Anwendungsfall die passende Transportschicht wählen. Auf einem PC bietet sich also XCP on Ethernet an, um auf virtuelle Steuergeräte und ggf. auch Funktionsmodelle zuzugreifen. Der Zugriff auf Variablen zur Kalibrierung oder Messung kann also bereits bei der PC-basierten Simulation mit den gleichen Werkzeugen und Methoden, wie sie in späteren Test- und Entwicklungsphasen für analoge Arbeitsschritte mit der realen ECU zum Einsatz kommen, geschehen.

Messung und Kalibrierung von Umgebungs- und Streckenmodellen stellen andere Anforderungen an die zu verwendende Schnittstelle. Während beim Messzugriff auf Steuergeräte die beschränkten Ressourcen des Steuergerätes eine effiziente Kommunikation und eine möglichst kompakte Variablenbeschreibung erfordern, steht bei Streckenmodellen möglichst umfassender und komfortabler Zugriff auf die Parameter des Modells im Vordergrund. Hier kann ASAM XIL API als Standard dienen, um Testfälle von den Testsystemen zu entkoppeln. Somit können auch hier einheitliche Werkzeugketten zum Einsatz kommen.

4 Virtualisierung durch Simulation

Die größte Herausforderung bei der Absicherung von Steuergeräte-Software mit PC-basierter Simulation besteht klar in einer möglichst realitätsnahen Simulation der Umgebung sowie der Software selbst. Es müssen also für unterschiedliche Domänen ausreichend detaillierte Modelle zur Verfügung stehen. Das kann nur erreicht werden, wenn diese Modelle auch

domänenspezifisch entwickelt werden können, was wiederum Entwicklungsumgebungen erfordert, die mit spezialisierten Bibliotheken die Modellentwicklung unterstützen. Diese Aufgabe kann daher nicht von einer einzigen Modellierungsumgebung durchgeführt werden. Stattdessen müssen Modelle aus verschiedenen Modellierungswerkzeugen zu einer Gesamtsimulation kombiniert werden.

Aber auch der Realitäts- und Reifegrad der Steuergeräte-Software sollte so nah wie möglich an dem des später real umgesetzten Steuergerätes sein. Ein rein funktionaler Test auf algorithmischem Level reicht hier nicht mehr aus.

Durch Varianten der jeweiligen Modelle hinsichtlich ihrer Ausführbarkeit lassen sich Systeme von Umgebungsmodellen erstellen, die beliebige Testszenarien ohne Abstriche in Bezug auf den Realitätsgrad oder die Simulationsperformanz bedienen. Ist die Software eines Steuergerätes noch nicht auf Serienebene verfügbar, können hier Verhaltensmodelle zum Einsatz kommen, sogenannte Soft-ECUs. Diese sollten zu einem späteren Zeitpunkt möglichst leicht gegen realitätsnähere Modelle einer ECU ausgetauscht werden können. Eine PC-basierte Absicherungsmethode sieht sich also mit heterogenen Anforderungen bezüglich der zu simulierenden Modelle konfrontiert.

Neben der modellbasierten Entwicklung stehen oft aber auch Funktionalitäten in Form von Quelltext, zum Beispiel als C-Code, zur Verfügung. Eine weitere Anforderung besteht also darin, diesen Quelltext in die Simulation einzubinden. Nutzt man Standards, gelingt die Einbindung umso einfacher, wenn diese Standards von der genutzten Simulationsplattform unterstützt werden. Dazu kann über Hilfsprogramme der vorliegende Quelltext so angepasst werden, dass er der Schnittstelle eines Standards entspricht und kann dann zum Beispiel wie eine normale AUTOSAR-Software-Komponente in eine Software-Architektur integriert werden. Ein solches Vorgehen erleichtert auch die Migration, sobald auf modellbasierte Entwicklung umgestellt wird.

Ein weiterer Vorteil der PC-basierten Simulation ist die Möglichkeit, mit gewöhnlichen Debug-Werkzeugen zu arbeiten. Im Gegensatz zu Echtzeitsystemen kann hier zum Beispiel die gesamte Simulation beim Erreichen eines Breakpoints angehalten und danach fortgesetzt, schrittweise ausgeführt oder pausiert werden.

5 Beispiel aus der Praxis

Wie eine vorverlagerte Absicherung von Steuergeräte-Software aussehen kann, zeigt Abbildung 5. Bevor die komplette Software auf einem Hardware-Prototyp geflasht und am HIL getestet wird, wird eine Software-Software-Integration mittels PC-basierter Simulation durchgeführt. Als PC-basierte Simulationsplattform kommt hier dSPACE VEOS® [4] zum Einsatz.

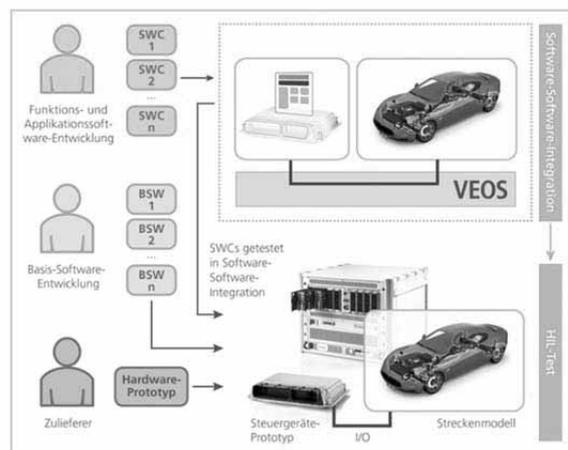


Abbildung 5: Software-Hardware-Integration mit vorgelagerter Software-Software-Integration

Ein Software-Integrator generiert aus den AUTOSAR-Software-Komponenten sowie der Laufzeitumgebung und den für die Simulation benötigten Basissoftwaremodulen entsprechende V-ECUs und verschaltet diese mit einem Streckenmodell auf Signalebene. Hier wurden drei dieser V-ECUs erstellt, die Fahrerassistenzfunktionen beinhalten. Ist eine solche Integration des aktuellen Software-Standes erstellt, kann sie für Tests genutzt werden. Dazu wurden spezielle Arbeitsplätze aufgebaut, die es einem Funktionsentwickler erlauben, eine neue Version seiner Komponente in das System aus V-ECUs und Streckenmodell zu integrieren und vordefinierten, automatischen Tests zu unterziehen. So lassen sich Fehler in einer Komponente eben auf diese zurückführen und direkt beheben. Bei der späteren Software-Hardware-Integration wäre die Beseitigung des gleichen Fehlers nicht nur risikoreicher und deutlich kostenintensiver. Sondern durch die Integration sowohl der Applikationssoftware mit der Basissoftware als auch das Integrieren auf die Hardware fällt es auch deutlich schwerer, bei einem auftretenden Fehler überhaupt erst die Ursache zu lokalisieren. Durch den beschriebenen Ansatz der Vorverlagerung kann dieses Risiko der Software-Hardware-Integration deutlich

gesenkt und die HIL-Systeme können effizienter genutzt werden. Da beim Testen durch die Verwendung von Standards wie XCP und XIL API (vgl. Absatz 3) die gleichen Werkzeuge wie am HIL zum Einsatz kommen, lässt sich der Aufwand für die Testentwicklung deutlich reduzieren und es entstehen Synergien, die auch den späteren Teststufen zugutekommen. Beispielsweise konnten die automatisch erzeugten Experimente bei Einführung der Virtuellen Absicherung in diesem Beispiel aus der HIL-Abteilung übernommen werden. Durch die konsequente Weiterentwicklung fließen die Erfahrungen aus der PC-basierten Simulation auch in die Erstellung der Experimente zurück und wirken sich somit positiv auf die HIL-Tests aus. Der AUTOSAR-Standard erlaubt nicht nur die systematische Integration der Software-Komponenten in einem Integrations-schritt, sondern auch das Integrieren neuer Stände einzelner Komponenten durch die komponentenverantwortlichen Funktionsentwickler. In dem vorliegenden Beispiel konnte erfolgreich eine Integration von Komponenten aus verschiedenen Werkzeugen vorgenommen werden. Verglichen mit Modellen bei herkömmlichen MIL- oder SIL-Tests in modellbasierten Umgebungen, steht dem Entwickler hier also eine deutlich detailliertere, seriennahe Umgebung zum Testen zur Verfügung.

6 Zusammenfassung

Zusammenfassend bietet die PC-basierte Absicherung von Software mittels Virtualisierung der Zielhardware beziehungsweise einer durch Standards unterstützten Abstraktion von der Hardware eindeutige Vorteile gegenüber herkömmlichen Ansätzen. Da es in diesem Szenario keine Echtzeitbedingungen gibt, wird die Ausführungsgeschwindigkeit der Simulation nur durch die Komplexität des Modells und der Leistung des eingesetzten PCs beschränkt. Diese Skalierbarkeit, verbunden mit kurzen Iterationen, erhöht den Testdurchsatz und damit den Reifegrad der Software signifikant. Weiterhin kann die Modellierung besser vom Testen getrennt und so entlastet werden.

7 References (Überschrift 1)

- [1] N. Grünwald und C. Pagenkemper. *Qualitätsmanagement mit neuen Arbeitsformen: Eine Arbeitshilfe zur Planung, Einführung, Qualifizierung und Auditierung von Qualitätssystemsmanagementssystemen*. Expert, Deutschland, 2004
- [2] AUTOSAR, www.autosar.org/, Zugriffsdatum 29.01.2016
- [3] FMI, <https://fmi-standard.org/>, Zugriffsdatum 29.01.2016
- [4] VEOS, <http://www.dspace.com/go/veos>, Zugriffsdatum 29.01.2016

Autorenverzeichnis

	Seite
A	
Alkov, Ilja	82
B	
Bicher, Martin	202
Brandt, Benjamin	223
Breitenbach, Thomas	113
Breitenecker, Felix	127, 165, 202
Brunnemann, Johannes	131
C	
Chudej, Kurt	101
Clauß, Christoph	245
Commerell, Walter	207, 255
D	
Deutenberg, Gernot	267
Dombi, Szilárd	255
Durak, Umut	6, 142, 148
F	
Fidorra, Nicolas	237
Freymann, Birger	100
Fritsch, Matthias	258
G	
Glock, Barbara	165
Gotschlich, Jürgen	142
Großer, Martin	230
Guerrero, Domingo	109
Gulbins, Matthias	37
Göbel, Mirek	179
Göllner, Marian	71
H	
Haase, Joachim	37
Hafner, Irene	127
Hartmann, Sven	148
Heinzl, Bernhard	83
Henke, Christian	174
Henninger, Matthias	45
Holtkötter, Jens	174
J	
Jacobitz, Sven	54
Junglas, Peter	90, 155
Jäschke, Jan-Niklas	237
K	
Karagounis, Michael	205
Kastner, Wolfgang	83
Kertscher, Ulrich	7
Krome, Jürgen	109, 120
Kröger, Claus	207

Körner, Andreas	98, 202
Kühn, Markus	214
L	
Lang, Thomas	206
Lei, Boxia	8, 37
Lerche, Jens	173
Lezsovits, Ferenc	255
Liu-Henke, Xiaobo	54, 71, 246, 258
Lucas, Martin	217
Lückerath, Daniel	223
M	
Mantz, Hubert	255
Maurer, Christian	207
Markwirth, Thomas	37
Matyszok, Carsten	214
Melcher, Andreas	8, 37
Michael, Jan	174
Mukbil, Awad	148
P	
Pawletta, Sven	100
Pawletta, Thorsten	100, 155
Petkun, Sergey	63
Popper, Niki	127, 165
Preyser, Franz	83
R	
Raich, Philipp	83
Reinders, Thomas	204
Rettberg, Achim	-
Rishe, Naphtali	223
Rohde, Magnus	8
Rund, Armin	101
Rühl, Martin	267
S	
Schele, Moritz	214
Scherler, Sören	246, 258
Schlüter, Wolfgang	29, 45
Schmidt, Artur	100, 155
Schmidt, Jörg	29
Schrade, Ulrich	207
Schulze1, Christian	237
Seibt, Michael	-
Seifert, Hans Jürgen	8
Sieben, Stephan	173
Stroganov, Peter	148
T	
Tao, Haoqi	71
Tegethoff, Wilhelm	237
Thümmeler, Axel	183
Trächtler, Ansgar	174
Tumforde, Timo	131

U	
Ullrich, Oliver	223
V	
Voigt, Sebastian	230
W	
Waurich, Volker	230
Weidemann, Dirk	82
Winkler, Stefanie	202
Wischhusen, Stefan	131
Wloka, Dieter	195
Wurm, Daniela	109
Wächter, Markus	101
Wüstenberg, Henrik	142
Z	
Ziebert, Carlos	8, 37
Zhao, Wen Jiao	8
Zorn, Sophie	113

Sponsor- und Ausstellerverzeichnis

Sponsor

HELLA KGaA Hueck & Co.

Rixbecker Straße 75, 59552 Lippstadt

www.hella.com

Aussteller

HELLA KGaA Hueck & Co.

Rixbecker Straße 75, 59552 Lippstadt

www.hella.com

TLK-Thermo GmbH

Hans-Sommer-Straße 5, 38106 Braunschweig

www.tlk-thermo.com

LTX Simulation GmbH

Wohlfahrtstraße 21b, 80939 München

www.ltx.de

CENIT AG

Kaiserwerther Straße 115, 40880 Ratingen

www.cenit.com

Dassault Systemes Deutschland GmbH

Joseph-Wild-Straße 20, 81829 München

www.3ds.com