

# Tagungsband

Workshop 2017  
ASIM/GI-Fachgruppen

9. und 10. März Ulm

Walter Commerell, Thorsten Pawletta (Hrsg.)

Arbeitsgemeinschaft Simulation ASIM  
In der Gesellschaft für Informatik GI

Fachgruppe  
Simulation technischer Systeme STS

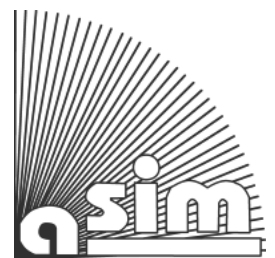
Fachgruppe  
Grundlagen und Methoden in  
Modellbildung und Simulation GMMS

Hochschule Ulm

**Hochschule Ulm**



University of  
Applied Sciences



ARGESIM Report AR 53

ASIM Mitteilung 162

ISBN print 978-3-901608-50-6

ISBN ebook 978-3-901608-90-2

DOI 10.11128/arep.53



# ASIM-Treffen STS/GMMS 2017

Workshop der ASIM/GI Fachgruppen STS und GMMS

**9./10. März 2017 in Ulm**

Tagungsband

Walter Commerell, Thorsten Pawletta  
(Hrsg.)

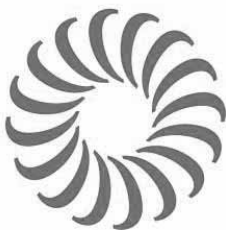


Arbeitsgemeinschaft Simulation ASIM  
in der Gesellschaft für Informatik GI

Fachgruppe Simulation technischer Systeme STS

Fachgruppe Grundlagen und Methoden in  
Modellbildung und Simulation GMMS

**Hochschule Ulm**



University of  
Applied Sciences

Hochschule Ulm

ARGESIM Report AR 53

ASIM Mitteilung 162

ISBN print 978-3-901608-50-6

ISBN ebook 978-3-901608-90-2

DOI 10.11128/arep.53

Hochschule Ulm  
Prittwitzstraße 10, 89075 Ulm  
Tel: +49 (0731) 50-208  
[www.hs-ulm.de](http://www.hs-ulm.de)

Redaktion Constantin Commerell

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 AR 53

ISBN print 978-3-901608-50-6

ISBN ebook 978-3-901608-90-2

DOI 10.11128/arep.53

ASIM Mitteilung 162

©ARGESIM Verlag Wien, Hochschule Ulm 2017

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](mailto:info@argesim.org); [www.argesim.org](http://www.argesim.org)

## **Vorwort**

Auf Einladung der ASIM/GI Fachgruppen STS und GMMS und der Hochschule Ulm findet am 9. und 10. März 2017 der Workshop "Simulation technischer Systeme inklusive der Grundlagen und Methoden in Modellbildung und Simulation" statt.

Die Workshops der beiden ASIM/GI Fachgruppen finden seit vielen Jahren 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 drei 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
- Mechatronische Systeme
- Methoden und Tools I und II
- Elektro- und Hybridfahrzeuge
- Simulation in Luft- & Raumfahrt I und II
- Echtzeitsimulation
- Modelbasierte Entwicklung
- Energietechnik
- Elektronikentwicklung
- Entwurfssprachen

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 Ulm 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, bei den Firmen für ihr Engagement und der Hochschule Ulm für die Austragung der Veranstaltung.

Walter Commerell, FG Simulation technischer Systeme (STS)

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

Ulm, im März 2017

**Tagungsleitung:**

Walter Commerell, Hochschule Ulm  
Thorsten Pawletta, ASIM/GI-GMMS

**Organisationsteam:**

Heinz-Theo Mammen  
Klaus Panreck  
Walter Commerell  
Joachim Haase  
Leo Gall  
Thorsten Pawletta  
Michael Striebel  
Robert Buchta  
Xiaobo Liu-Henke  
Umut Durak

**Tagungsorganisation:**

Hochschule Ulm  
Prittwitzstraße 10  
89075 Ulm  
Tel. +49 (0731) 50-208  
[www.hs-ulm.de](http://www.hs-ulm.de)

**Tagungsort:**

Hochschule Ulm  
Campus Prittwitzstraße  
Prittwitzstraße 10  
89075 Ulm

## **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](http://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/](http://www.asim-gi.org/gmms/)

## **Hochschule Ulm**

Die Hochschule Ulm ist eine Hochschule für Angewandte Wissenschaften mit überwiegend technisch ausgerichtetem Studienangebot. Die Studiengänge sind breit angelegt, um den Absolventen vielfältige berufliche Möglichkeiten zu eröffnen. Individuelle Qualifizierung ist möglich durch die Auswahl von Wahlpflichtfächern, Studien- und Abschlussarbeiten, Fremdsprachen, Studienaufenthalte an ausländischen Partnerhochschulen sowie durch den Erwerb von Zusatzqualifikationen.

Der Erfolg der Hochschule Ulm gründet auf der Zusammenarbeit mit der Wirtschaft. Sie schließt die Bereiche Forschung und Entwicklung über eigene Forschungsinstitute und mehrere Steinbeis-Transfer-Zentren der Steinbeis-Stiftung für Wirtschaftsförderung Stuttgart mit ein. Hiervon profitieren die Studierenden auch durch attraktive Praktikumsplätze und interessante Abschlussarbeiten. Ein ausgefeiltes Weiterbildungsprogramm für Berufstätige und andere runden das Angebot der Hochschule Ulm ab.

# Inhaltsverzeichnis

	Seite
<b>Plenarvortrag</b>	
Flüchtlings- und Migrationsströme in Europa - ein Simulationsversuch <i>Felix Breitenecker, Andreas Körner, Nikolas Popper, Tamara Vorbruda TU Wien</i>	6
<b>Plenarvortrag</b>	
Simulation und Prognose der Zuverlässigkeit im automobilen Entwicklungsprozess eDrive <i>Arnold Lamm, Matthias Grabert, Dr. Jörg Keller Daimler AG</i>	9
<b>Thermische Systeme</b>	
Entwicklung eines räumlich aufgelösten dynamischen Prozesssimulationsmodells eines Aluminiumschachtschmelzofens <i>Matthias Henninger, Wolfgang Schlüter Hochschule Ansbach</i>	12
Design of simulation model for novel solar thermal storage tank <i>Muthalagappan Narayanan, Gerhard Mengedocht, Walter Commerell Hochschule Ulm</i>	18
Simulation von Strömung und Wärmeübergang in der Energietechnik mit OpenFOAM <i>Peter Renze Hochschule Ulm</i>	24
<b>Elektro- und Hybridfahrzeuge</b>	
Systemdesign eines Batteriemanagementsystems für Elektrofahrzeuge <i>Sven Jacobitz, Sören Scherler, Xiaobo Liu-Henke Hochschule Ostfalia</i>	30
Prädiktives Energiemanagement für Elektrofahrzeuge <i>Matthias Fritsch, Sören Scherler, Xiaobo Liu-Henke Hochschule Ostfalia</i>	36
Development and Design Optimization of a High-Temperature Superconducting Quasi-Diamagnetic Motor Demonstration Unit <i>Marcell Baranyai, Dénes Kapi, István Vajda, Walter Commerell Óbuda University / Hochschule Ulm</i>	45
Digital modelling for EV optimization <i>Helge Tielbörger Siemens Industry Software GmbH, Digital Factory Division</i>	50

## **Simulation in Luft- & Raumfahrt I**

2Simulate - Werkzeug zur Online-Trimmung für die Simulation  
dynamischer Systeme 54  
*Jürgen Gotschlich*  
*DLR*

Open Innovation/Sagitta – Implementation and Validation of a Real-Time  
Flight Dynamics model for Simulation, Integration Testing and Pilot 61  
Training  
*Kuchar Richard*  
*DLR*

Modellierung komplexer Flughafen-Ereignisse für Human-In-The-Loop 69  
Simulationen  
*Schier Sebastian, Anne Papenfuß, Nils Carstengerdes*  
*DLR Braunschweig*

Ein Flug durch Raum und Zeit – Entwurf und Evaluation einer 75  
Simulationsdynamik für das Flughafenmanagement  
*Schier Sebastian, Yves Günther, Sandro Lorenz, Reiner Suikat, Florian*  
*Piekert*  
*DLR Braunschweig*

## **Simulation in Luft- & Raumfahrt II**

Design exploration process for aerospace industry 81  
*Peter Hoffmann*  
*Chias TEK*

Optimaler Flug eines Drachenfliegers durch eine Wolkenstraße 87  
*Britzelmeier Andreas, Anna-Lena Klingler, Kurt Chudej*  
*UNI Bayreuth*

Interaktion stochastischer Filter für unterschiedlich dimensionierte, 93  
heterogene Zustandsräume  
*Edmond Skeli, Dirk Weidemann*  
*FH-Bielefeld*

## **Methoden/Tools I**

Simulation als epistemologische Grundlage für intelligente Roboter 99  
*Andreas Tolk, Umut Durak*  
*DLR Institut für Flugsystemtechnik*

Entwurf, Simulation und Implementierung ereignisdiskreter Steuerungen 106  
mit PDEVS RCP Version 2.0  
*Birger Freymann, Artur Schmidt, Sven Pawletta, Sven Hartmann,*  
*Thorsten Pawletta*  
*Hochschule Wismar*

Conceptional problems of transaction-based modeling and its 112  
implementation in SimEvents 4.4  
*P. Junglas, Lars Austermann, Jan Schmidt, Christian Tiekmann*  
*PHWT Vecta/Diepholz/Oldenburg*

Vergleich von Modellbildungs- und Simulationszugängen zu strukturdynamischen Systemen - ARGESIM Benchmark C21 'State Events and structural-dynamic Systems' <i>Felix Breitenecker, Andreas Körner</i> <i>TU Wien</i>	118
--	-----

## **Methoden und Tools II**

Die Wissensgebiete der Simulationstechnik <i>Umut Durak, Torsten Gerlach</i> <i>DLR</i>	124
---	-----

Parameter der arteriellen Pulswellenanalyse zur modellbasierten Charakterisierung eines reduzierten Herzauswurfs <i>Andreas Bauer, Bernhard Hametner, Felix Breitenecker, Thomas Weber, Siegfried Wassertheurer</i> <i>TU Wien, AIT Wien</i>	130
--	-----

A Microscopic Model for Simulation of Time Efficiency Concerning the Supply in Steep Terrain <i>Johannes Tanzler, Martin Bicher, Felix Breitenecker</i> <i>TU Wien</i>	134
--	-----

## **Methoden und Tools III**

Fallstudien für die aggregierte Analyse von agentenbasierten Modellen mithilfe von Markovketten <i>Florian Kitzler, Martin Bicher, Niki Popper, Felix Breitenecker</i> <i>1TU Wien, dwh Simulation Services</i>	138
---	-----

Softwarebasierte Analyse von Simulationsdaten durch automatisierte Berechnung von Kennzahlen <i>Jakob Krieg, Wolfgang Schlüter</i> <i>Hochschule Ansbach</i>	144
--	-----

Automation-Data-Exchange (ADEX) - Durchgängiges Austauschsystem für Simulation und Visualisierung <i>Weigert David, Paul Aurich</i> <i>UNI Magdeburg</i>	150
--	-----

Simulationskonzepte in der Brandschutzausbildung <i>Dieter Wloka</i> <i>Universität Kassel</i>	157
--	-----

## **Mechatronische Systeme**

Projektierung mechatronischer Anlagen in der Hochschullehre am Beispiel von Labormodellen <i>Loose Tobias</i> <i>HS-Heilbronn</i>	162
---	-----

Simulationsgestützte Schwingungsstabilisierung hydraulischer Systeme <i>Mathias Niebergall, Rudolf Mönkle</i> <i>Hochschule Ulm</i>	168
---	-----

Modelling of friction in mechatronics systems <i>Sergey Petkun</i> <i>Brose Fahrzeugteile GmbH</i>	170
--	-----

## **Modellbasierte Entwicklung**

Bestimmung unbekannter Simulationsparameter durch eine mehrstufige Monte-Carlo-Simulation unter Einbeziehung gemessener Daten eines Referenzmanövers <i>Jan Michael Veith, Albert Albers, Matthias Behrendt, Achim Romer</i> <i>Porsche AG, KIT</i>	177
---	-----

Funktionale Sicherheit (ISO 26262) und Simulation <i>Walter Commerell</i> <i>Hochschule Ulm</i>	183
---	-----

Modellbasierte Entwicklung für sichere Software bei MAGIRUS <i>Kai Kriegel</i> <i>Magirus GmbH CNH Industrial</i>	187
---	-----

## **Echtzeitsimulation**

Echtzeitsimulation hochdynamischer Fahrzeugantriebe <i>Jakob Häckh, Günter Willmerding</i> <i>Steinbeis Transferzentrum Verkehrstechnik.Simulation.Software</i>	192
---	-----

Systemkonzept eines modularen HiL-Systems für modellbasierte Funktions-entwicklung fahrzeugmechatronischer Systeme <i>Xiaobo Liu-Henke, Marian Göllner, Robert Buchta, Florian Quantmeyer</i> <i>Hochschule Ostfalia</i>	199
--	-----

Effiziente Absicherung von Systemen der Fahrzeugelektronik mit System-HiLs in der Elektronikintegration <i>Buchta Robert</i> <i>VW</i>	205
--	-----

## **Energietechnik**

Verlustanalyse im elektrischen Antrieb von Brennstoffzellenfahrzeugen bei variierender Zwischenkreisspannung <i>Anna-Lena Menn, Markus Henke</i> <i>TU Braunschweig</i>	210
---	-----

Schutzorgane und -konzepte in Niederspannungs-Gleichstromnetzen - Modellierung von Komponenten, Fehlersituationen und Gesamtsystem <i>Strobl, Christian, Maximilian Schäfer, Rudolf Rabenstein, Leopold Ott, Julian Kaiser</i> <i>ETA, Friedrich-Alexander Universität Erlangen-Nürnberg, FhG IISB</i>	216
--	-----

Gebäudesimulationen für ein Heimenergiemanagement im Kontext zu verschiedenen Gebäudetypologien mit Solarenergienutzung <i>Gerhard Mengedoht, Walter Commerell, Sebastian Schmidt, Mitat Tuzcu</i> <i>Hochschule Ulm</i>	222
--	-----

Sektorkopplung als Baustein in der Energiewende – Ergebnisse einer Bidirektionalen Simulation von Energienetzen 225  
*David Stakic, Falko Ebe*  
*Hochschule Ulm*

## **Elektronikentwicklung**

Leistungsverteilung in Elektrofahrzeugen mit Range Extender 231  
*Sören Scherler, Xiaobo Liu-Henke*  
*Hochschule Ostfalia*

Simulationsgestützter Entwurf eines 48 V/ 12,5 kVA Wechselrichters für Mild-Hybrid Fahrzeuge 239  
*Christoph Faraji-Tajrishi, Konstantin Siebke, Markus Henke*  
*TU Braunschweig*

Dynamische Fehlerinjektion für die Systemsimulation mittels SystemC – effizient, flexibel, automatisierbar 246  
*Thomas Markwirth, Stephan Gerth*  
*Fraunhofer IIS/EAS*

Vergleichende Untersuchungen an DC/DC-Wandler Modellen für die Bordnetzsimulation 252  
*Thomas Lang*  
*Robert Bosch GmbH*

## **Entwurfssprachen**

Konzeptplanung von Montagesystemen mit graphen-basierten Entwurfssprachen 256  
*Theresa Breckle, Jens Kiefer, Markus Kiesel, Martin Manns*  
*Hochschule Ulm, Hochschule Albstadt-Sigmaringen, Uni Siegen*

Generierung und Anreicherung von virtuellen Inbetriebnahmemodelle durch graphenbasierte Entwurfssprachen 262  
*Kiesel Markus, Nicolai Beisheim, Theresa Breckle*  
*HS Albstadt-Sigmaringen, Hochschule Ulm*

Ansatz zur rechnergestützten Synthese und Analyse von Entwurfsvarianten für Formula Student Getriebe mittels graphenbasierter Entwurfssprachen 268  
*Kevin Holder, Andreas Zech, Ralf Stetter, Markus Till*  
*ZF Friedrichshaven AG, Hochschule Ravensburg-Weingarten*

Modellierung und Simulation eines Formula Student Rahmens mittels graphenbasierter Entwurfssprachen 274  
*Manuel Ramsaier, Ralf Stetter, Markus Till*  
*Hochschule Ravensburg-Weingarten*

**Aussteller** 280

# Flüchtlings- und Migrationsströme in Europa: ein Simulationsversuch

## - Migration and Refugee Paths in Europe: a Simulation Attempt

Felix Breiteneker<sup>1,2</sup>, Andreas Körner<sup>1,2</sup>, Nikolas Popper<sup>1,3</sup>, Tamara Vorbruba<sup>2</sup>

<sup>1</sup>TU Wien, COCOS – Centre f. Computational Complex Systems, Wiedner Hauptstr. 8-10, 1040 Vienna, Austria

<sup>2</sup>TU Wien, ARGESIM/Mathematical Modelling and Simulation,

<sup>3</sup>dwh Simulation Services, Neustiftgasse 57-59, 1070, Vienna

*Felix.Breiteneker@tuwien.ac.at*

In September 2015 and October 2015 Europe was confronted with a dramatic migration movement - the so-called refugee crisis 2015. Can simulation help to analyse the pathway of migration in such a critical case, and can simulation forecast migration and refugee pathways? This contribution presents a simulation attempt by means of a discrete model and of UNHCR data.

### 1 Introduction

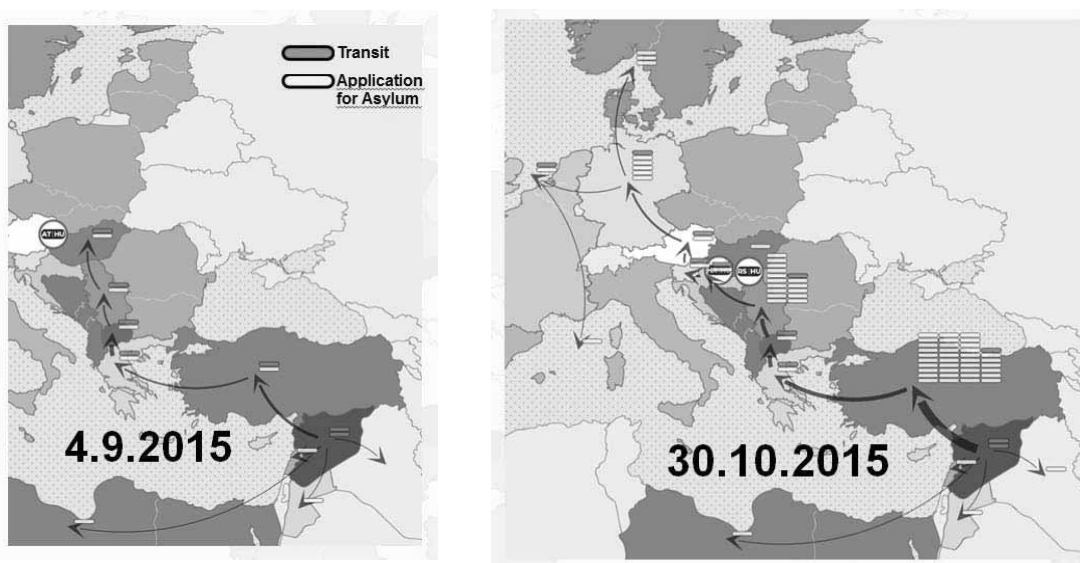
In September 2015 and October 2015 Europe was confronted with a dramatic migration movement - the so-called refugee crisis 2015 – visualized in Figure 1. UNHCR data show. Classic migration may follow ‘laws’ for population dynamics, but migration based / mixed with refugee pathways require a more sensitive simulation approach.

### 2 Modelling Migration Pathways

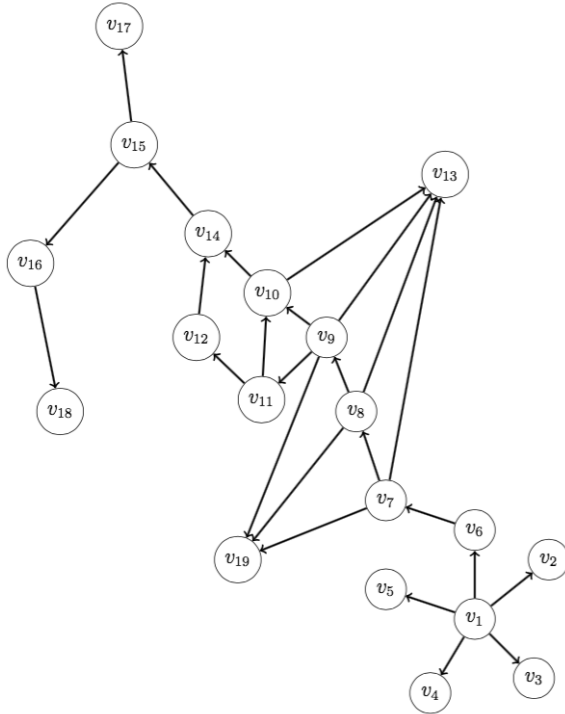
Migration behaviour seems to be driven by individual and personal circumstances of the refugees. From a macroscopic viewpoint, migration is a special kind of spatial population dynamics.

Parts of a population migrate from one region into another region, driven by repulsive forces in the start region, and by attractive forces of the destination region. Regions often try to control migration by border control - separation forces for the migration movement. These forces, together with the pathway of migration, allow a modelling approach for the migration movement as behavioural spatial interaction model.

This contribution presents a macroscopic time-discrete dynamic model for migration movement, based on the forces of repulsion, attraction, and separation between regions, countries, or states.



**Figure 1.** UNHCR data for transit and asylum applications at begin and at end of refugee crisis 2015



**Figure 2.** Directed graph for migration and refugee pathway from Syria (node 1) via Turkey (node 6) via Balkan route to Central Europe (Austria node 14, Germany node 15)

The topology - the pathway of migration - is described by a partly directed graph (Figure 2) with regions as nodes, and the migration force from one region into another is an appropriate function of the driving forces repulsion, attraction, and separation.

The model can be seen as a generalisation of classic gravitation- like model for population migration with John Steward's concept of demographic gravitation.

Attraction forces between regions are compiled from seven components:

- gross domestic product (GPD)
- fragile state index (FSI)
- migrants in the countries
- maximum of attractive forces of accessible countries
- exceeded capacity of the country
- asylum recognition rate
- asylum recognition quote in Europe.

Repulsion forces are compiled by five components:

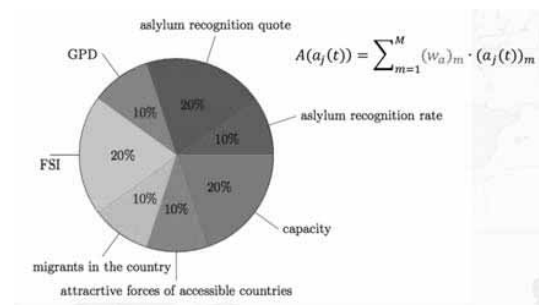
- gross domestic product (GPD)
- fragile state index (FSI)
- exceeded capacity of the country
- asylum recognition rate
- and asylum recognition quote in Europe.

Separation forces are controls by bilateral border actions. Some regions are transit regions, which allow a simpler compilation of migration flow.

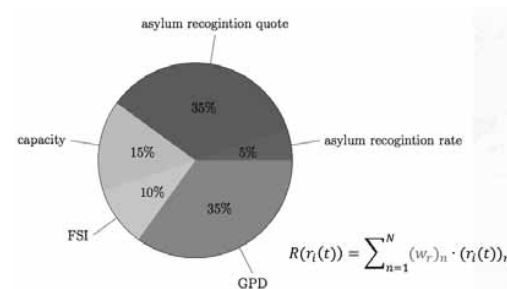
### 3 Model Identification for 2015 Crisis

Using data of the refugee crisis 2015 from official indeed allows to parameterize and to identify the model for the migration from Syria to Western Europe in September and October 2015 in a first step qualitatively. Parameters in the model are the weighting functions for the components of the driving forces (Figure 3, Figure 4), which are identified by heuristic knowledge and comparison with result data.

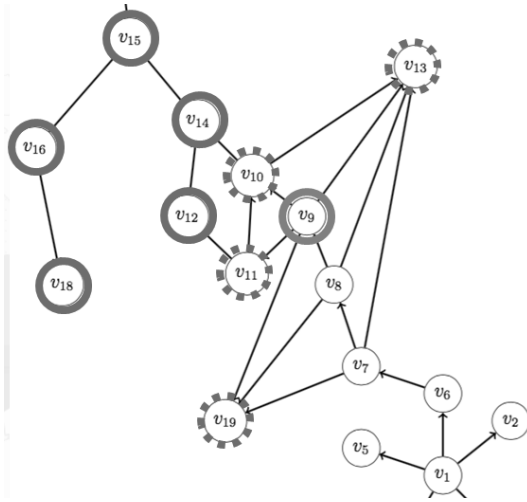
Time basis for the model is one day: update is given by the balance of inflow and outflow of migrants driven by the migration forces. This macroscopic approach models the migration movement on the basis of whole populations, so the simulation results have to be interpreted in a qualitative manner and not in absolute numbers.



**Figure 3.** Components of attraction forces with weights (parameters)



**Figure 4.** Components of repulsion forces with weights (parameters)



**Figure 5.** Components of attraction forces for node 9:  
dashed nodes – attraction by direct neighbours  
solid nodes – additional attraction by target countries

For a ‘better’ identification with ‘higher’ quantitative degree the approach for attractive forces had to be refined: note only the neighbouring countries are driving the migration, but also the attraction forces of some further countries, usually the target countries., as sketched in Figure 5.

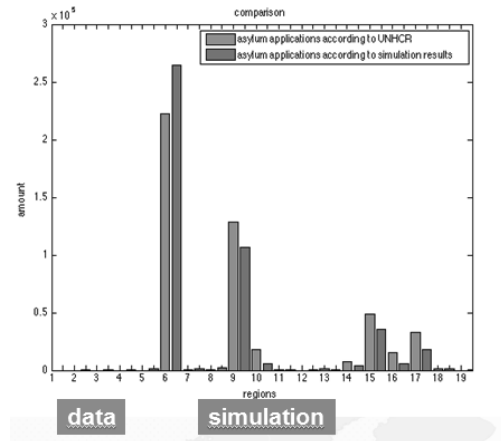
#### 4 Simulation of Migration and Refugee Pathway

The implemented model serves three tasks:

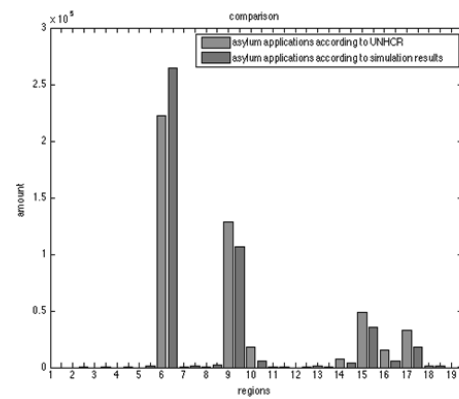
- (i) an a posteriori simulation of the refugee crisis 2015, (Figure 6)
- (ii) an a priori simulation (forecast) for the migration movement in summer 2016 with given prerequisites, and
- (iii) simulation with different scenarios of separation forces (border control)

Topological data from 2015 and ‘initial’ migration data for 2016 allow a forecast for the migration of Syria to Western Europe under closed Balkan route and applying so-called Turkey - Deal. Begin of 2017, model parameters will be adjusted wrt new data available.

With topological data from 2015 and ‘initial’ migration data for 2016 scenarios for migration movement in summer 2016 can be simulated under experimental change of repulsive forces, attraction forces, and especially of separation forces (closing at Brenner Pass, closing of German borders).



**Figure 6.** Comparison of data and simulation for refugee crisis 2015



**Figure 7.** Comparison of data and simulation for 2016 (Turkey deal, Balkan route closing)

#### 5 Summary

Models are in any case a simplification of reality, but they should help in better understanding of complex dynamics - as migration movement, and the intention of the model is to improve the situation of refugees under appropriate prerequisites.

#### 6 References

- [1] T. Hatton and J. Maloney. (2015). Applications for Asylum in the Developed World: Modelling Asylum Claims by Origin and Destination. ANU Working Papers in Economics and Econometrics, 625, Australian National Univ.
- [2] A. Sen and T. Smith T. (1995). Gravity Models of Spatial Interaction Behaviour. Springer.
- [3] T. Vobruba. Modellbildung, Analyse und Simulation eines räumlichen Interaktionsmodells. Master Thesis, TU Wien, 2016.

# Simulation und Prognose der Zuverlässigkeit im automobilen Entwicklungsprozess eDrive

Dr. Arnold Lamm, Dr. Matthias Grabert, Dr. Jörg Keller  
Daimler AG, Mercedes-Benz Pkw Entwicklung, D-89081 Ulm  
*joerg.keller@daimler.com*

Im industriellen Kontext ist es wichtig, fortlaufend Aussagen über die Ausfallraten von der Vorentwicklungsphase bis zum Produkteinsatz zu treffen. Hierzu werden prospektive Systemlebensdaueranalysen des thermisch- / elektrischen Verhaltens zum eDrive Antriebsstrang, sowie grundlegende statistische Untersuchungen zur Verschleißprognose von Hochvolt-Batteriesystemen durchgeführt. Diese erfolgen auf Basis von Feldfeedback aus Kundeneinsatz, Fahrzeugdauerlaufdaten, Vermessung und Expertenwissen der Materialhersteller. Hier setzt die Validierung der Lebensdauerprognosen und Simulationsmodelle an.

## 1 Einleitung

Die nachhaltige Strategie von Daimler basiert auf der Brennstoffzellentechnologie und den xEVs (Hybrid, Plug-in und Elektrofahrzeuge). Für ein qualitätsorientiertes Unternehmen wie Daimler ist es zwingend notwendig die Produktzuverlässigkeit dieser neuen Technologien zu gewährleisten und dadurch Ausfälle von Fahrzeugen zu vermeiden. Dies erfordert eine Fehlerursachenanalyse der zugrundeliegenden Versagensmechanismen und ein daraus folgender Rahmen statistischer Methoden.

## 2 Methoden und Analyseverfahren

### • Abschätzung der Alterung basierend auf “Design of Experiments” (DoE):

Durch umfangreiche und strukturierte Vermessungen der Batteriezellen wird im Laborbetrieb die Alterung der Zellen unter verschiedenen Betriebsbedingungen bestimmt.

In den durchgeführten Analysen erfolgt eine Schätzung alterungsbedingter Kapazitätsverluste durch kalendarische Alterung, gekoppelt mit einer nachgeschalteten Zyklusbelastung auf Basis Tagesnutungsprofilen. Es werden Alterungsprognosen durchgeführt, die je nach Betriebsstrategie zwischen 6 und 10 Jahren liegen.

### • Abschätzung der Alterung basierend auf Felddatenanalyse:

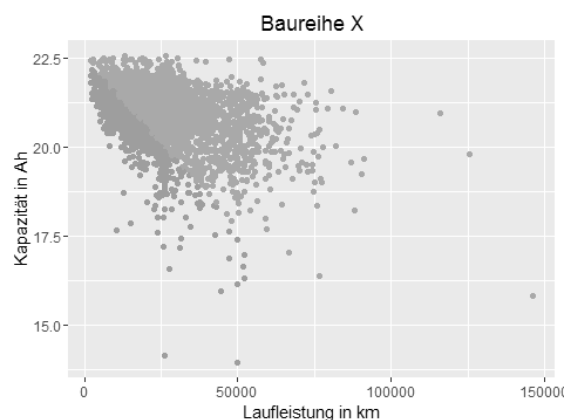
Die Alterungsanalyse erfolgt hier mit Hilfe von Felddaten anhand von realen Kundenprofilen und einer Extrapolation des Kundenverhaltens auf eine bestimmte Anzahl von Jahren.

Aufgabe ist die Identifikation von Auffälligkeiten in den Daten bzgl. des Verlaufes von Kapazität und Innenwiderstand.

### • Root cause analysis:

Vorgehen: Kontraste in Daten erkennen um die technischen Root Causes näher eingrenzen zu können. Hierbei ist das Clusterverfahren ein möglicher Ansatz.

Ziel hierbei ist die Ableitung von Schädigungsmechanismen auf Basis eines Clustering der Flottenfahrzeuge entsprechend des Kapazitätsverlustes sowie der Bestimmung von kausalen Lastkollektiven durch eine Diskriminanzanalyse (partial least squares discriminant analysis) und Regression. Methodisch erfolgt dies mit Hilfe von Clusterverfahren wie k-means mit einer Fahrzeug-Einteilung in „high/low fader“ (siehe Abb. 1) und Kopplung von Regressionsmodellen.

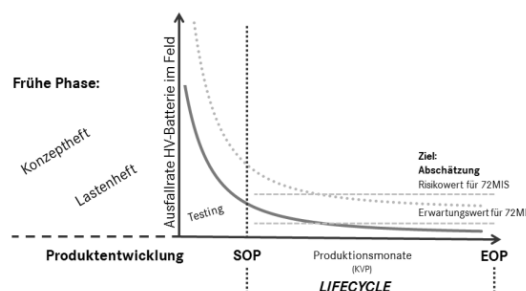


**Abbildung 1:** Kapazität über Laufleistung von Plug-In Zellen einer typischen Baureihe für „high fader“ in rot und „low fader“ in blau

Hier dient die Multiple Lineare Regression [1] als Ausgangspunkt. Komplexere Regressionsmodelle sind jedoch nötig, da die klassische multiple lineare Regression von einer homogenen Varianzstruktur der Residuen ausgeht, die in den Daten nicht bestätigt wird. Es zeigt sich eine größere Unsicherheit der Prognosen für länger/intensiver betriebene HV-Batterien. Zur Anpassung des Ordinary Least Squares (OLS) Schätzers wird als Lösungsansatz ein White-Schätzer eingesetzt, der robust gegenüber Abweichungen bezüglich der Annahme einer homogenen Varianzstruktur ist. Resamplingverfahren sowie Bootstrap Regression dienen als weitere Alternative zur Lösung.

- **Überlebensanalyse in der frühen Konzeptphase:**

Es handelt sich hierbei um die Prognose der Ausfallrate der Fehler in einer frühen Projektphase.

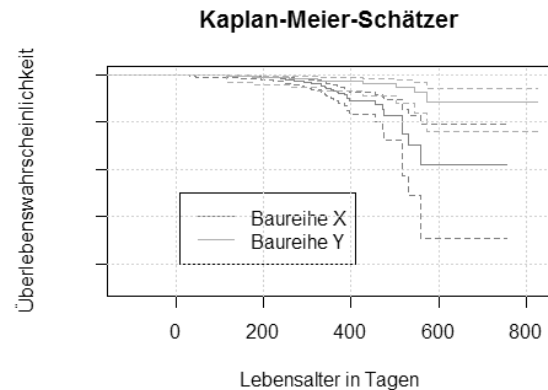


**Abbildung 2:** Ausfallraten Hochvoltbatterien über Lifecycle

Zum einen wird die Ausfallwahrscheinlichkeit bestimmt und eine quantitative Möglichkeit geschaffen, um Zuverlässigkeit und Robustheit beschreiben zu können [2]. Zum anderen wird der Erwartungswert und die 90%- Risikoabschätzung (Mittelwert über Lifecycle) für aktuelle Batterieprojekte ermittelt.

- **Überlebensanalyse in der Feldphase:**

Die Überlebensanalyse erfolgt mittels Hazard-Funktionen, Kaplan-Maier-Schätzer, sowie dem Cox-Aalen Modell, bezüglich kritisch geprüfter Daten. Letzteres ist eine Kombination des Cox proportional Hazards Modells und des additiven Aalen Modells [3]. Diese Erweiterung erlaubt insbesondere die Inklusion zeitabhängiger Effekte. Es handelt sich hierbei um beschleunigte Ausfallzeitmodelle für den Erwartungswert und Worst-Case-Prognosen.



**Abbildung 3:** Kaplan-Meier-Schätzer (mit 95% Konfidenzintervallen) für Überlebenswahrscheinlichkeit zweier Baureihen über Lebensalter in Tagen. End of Life als 20% Alterung bezüglich Kapazität angesetzt

- **Fehlerklassifikation:**

Das Auftreten von Fehlerbildern in Hochvoltbatterie Fahrzeugen ist Gegenstand laufender Forschung in den Ingenieurwissenschaften. Bis zum jetzigen Zeitpunkt sind Methoden zur Identifikation von Fehlern in xEVs kosten- und zeitintensiv. Die Implementierung eines Vorhersagesystems auf Basis maschinellen Lernens würde eine signifikante Verbesserung des bisherigen Prozesses bewirken und eine Skalierung erleichtern.

Die Idee der Fehlerklassifikation ist den Prozess der Erkennung von Fehlerbildern als eine Aufgabe des überwachten maschinellen Lernens zu formulieren [4]. Hierbei lernen statistische Klassifikationsalgorithmen anhand von DTCs aus dem Steuergerät eines Fahrzeuges fehlerspezifische Muster, welche sich zur Klassifikation neuer Fehler eignen. Neben der klassischen Mustererkennung macht die Fehlerklassifikation Gebrauch von Verfahren der Ausreißererkennung zur Identifikation neuer Fehler, die im Verlauf des Lebenszykluses einer Baureihe auftreten können [5].

Eine besondere Herausforderung stellt der Einsatz von datengetriebenen Methoden bei der Einführung neuer Baureihen dar. Die Knappheit an Daten in der frühen Feldphase erschwert das suffiziente Training eines lernenden Algorithmus'. Aus diesem Grund besteht die Notwendigkeit, Expertenwissen in die Klassifikation miteinfließen zu lassen, was über statistische Verfahren der Entscheidungsfindung ermöglicht wird [6].

- **Weitere Forschungsansätze:**

- **Prozessketten-Analyse [4]**

Unter dem Begriff Prozesskette (PK) versteht man eine Abfolge von sequentiellen Fertigungsprozessen im Bereich der Fertigungstechnik.

Es handelt sich um eine aufeinander aufbauende Folge von Aktivitäten, die als Gesamtsystem betrachtet wird. Dabei wird angenommen, dass sich die einzelnen Teilprozesse gegenseitig beeinflussen. Entscheidend ist die Übertragung von Bauteileigenschaften von einem Kettenglied zum nächsten. Inputs werden dabei sequentiell von einem Anfangs- in einen Endzustand überführt. Moderne Produktionsprozesse bestehen aus mehreren Teilprozessen, die jeweils individuell modellierbar sind. Unser Interesse gilt vorallem der Entwicklung von Unsicherheit innerhalb dieser Prozessketten.

- **QUARTZ-Analyse**

Quantitative Analysemethoden zur Bestimmung der Zuverlässigkeit sind im Fahrzeugbau etabliert [8]. DAIMLER setzt die QUARTZ Methodik zur Abschätzung der Hochvoltbatterie-Ausfallrate in der Lastenheftphase zur Konzeptbewertung und GuK-Abschätzung ein. Die QUARTZ Methodik liefert den Erwartungswert für den Ausfall des Batteriesystems. Basis sind die Expertenschätzung oder das Feldfeedback der Parts-Per-Million (ppm)-Raten für Zelle, Batterie-Management System (BMS), Kühlsystem und weiteren Komponenten. Die Prognose erfolgt für Fahrzeugflotten für einen Einsatz von bis zu 96 Monate.

### 3 Zusammenfassung und Ausblick

Fehlerratenprognosen und Risikobewertungen werden als Teil des DAIMLER R&D von Hochvoltbatterien in Hinblick auf die Zuverlässigkeitsprognose eingesetzt. Dabei zeigen Feldanalysen, dass Ausfälle der Batriekomponenten nicht einfach durch eine einparametrische Weibull-Verteilung dargestellt werden können. Das Einbeziehen physikalischer und phänomenologischer Modelle, sowie das Erweitern der wesentlichen Alterungssimulation auf nichtlineare Modelle werden die zukünftige Entwicklungsarbeit stark beeinflussen.

### 4 Literatur

[1] R. Johnson und D. Wichern. Multivariate Linear Regression Models. In: Applied Multivariate Statistical Analysis, Herausgeber: Pearson Education Limited, 2014.

[2] T. Lehmann, O. Özgen und J. Keller. Methodological framework for estimation of failure rates for various high-voltage battery systems. Daimler AG, Ulm, Germany. ESREL2016, Glasgow, 2016.

[3] T. Martinussen und T. Scheike. Dynamic Regression Models for Survival Data, Herausgeber: Springer, 2006.

[4] T. Hastie, R. Tibshirani und J. Friedman. The Elements of Statistical Learning. Data Mining, Inference and Prediction. Berlin: Springer, 2009.

[5] M. Pimentel, D. Clifton, L. Clifton und L. Tarassenko. A review of novelty detection. In: Signal Processing. Vol. 99. Amsterdam: Elsevir, 2014.

[6] T. Dietterich. Ensemble Methods in Machine Learning. In: International Workshop on Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science. Vol. 1857. Berlin: Springer, 2000.

[7] O. Meyer, J. Keller und C. Weihs. Uncertainty in Process Chains, Daimler AG, Ulm, Germany. ESREL2016, Glasgow, 2016.

[8] B. Bertsche, G. Lechner. Zuverlässigkeit im Fahrzeug- und Maschinenbau, Springer Verlag, 2004.

# Entwicklung eines räumlich aufgelösten dynamischen Prozesssimulationsmodells eines Aluminiumschachtschmelzofens

Matthias Henninger<sup>1</sup>, Wolfgang Schlüter<sup>1</sup>

<sup>1</sup>Hochschule Ansbach

*matthias.henninger@hs-ansbach.de*

In der vorliegenden Arbeit wird der „multi node“ Ansatz, welcher aus der Modellierung von sensiblen Wärmespeichern bekannt ist, auf den Schmelzschacht eines Aluminiumschachtschmelzofens angewandt. Der Ansatz beruht auf der Zerlegung des betrachteten Problems in eine beliebige Anzahl von Segmenten (nodes). In jedem dieser Segmente werden die dort lokalisierten Vorgänge mit einem Satz gekoppelter gewöhnlicher Differentialgleichungen erster Ordnung, welcher die Energie- und Massenerhaltungsgleichungen enthält, erfasst und berechnet. Zusätzlich können sehr einfach für jedes Segment unterschiedliche Materialwerte, Gleichungen und Rahmenbedingungen vorgegeben werden. Dies ermöglicht eine relativ detaillierte Betrachtung des Problems.

Im Rahmen der Modellbildung werden die Konzepte für die Zerlegung des Schachtschmelzofens in die Teilsegmente anhand der Aluminiummasse, sowie für die Grundfunktionen Materialeinbringung (Chargieren), Erwärmen und Schmelzen beschrieben. Im anschließenden Kapitel der Modellierung wird die Umsetzung des entwickelten Modellkonzeptes mittels Algorithmen, sowie die mathematische Modellierung der Vorgänge innerhalb eines Segments erklärt. Die Vorgänge umfassen im Wesentlichen die Aluminiumkomponente, die Rauchgaskomponente und die Wärmeübertragung zwischen diesen beiden Komponenten.

## 1 Einleitung

Fortschrittliche Automatisierungstechnik und Energieeffizienz sind eine wichtige Grundlage für den Erfolg von energieintensiven Betrieben mit Produktionsstandorten in Deutschland. Für die Nicht-Eisen-Gussindustrie, welche im Fokus der vorliegenden Arbeit steht, trifft dies aufgrund des sehr hohen Energieverbrauchs besonders zu. Für eine Tonne guten Guss beträgt der Energieeinsatz im Durchschnitt zwischen 3000 und 5000 kWh [1, 2]. Mit 50 bis 60 % des Energieverbrauchs sind die Schmelzöfen, welche für das betriebsinterne Aufschmelzen des Aluminiums verwendet werden, die Hauptverbraucher [1, 3]. Durch Nutzung modernerer Ofentechnik, eine kontinuierliche Betriebsweise und hohe Auslastung der Schmelzöfen könnte der Verbrauch deutlich gesenkt werden [4–6].

Das Kompetenzzentrum für industrielle Energieeffizienz der Hochschule Ansbach forscht im Rahmen des „Smart-Melting“ Projektes an Möglichkeiten die Betriebsweise und Auslastung von Schmelzöfen durch Einsatz eines Prozessmanagementsystems (PMS) zu verbessern. Zur Identifikation beziehungsweise Untersuchung von Verbesserungsmaßnahmen ohne Eingriff in den laufenden Fertigungsprozess dient ein Simulationsmodell des Gesamtbetriebes, welches wiederum aus verschiedenen Teilmodellen besteht. Eines dieser Teilmodelle ist das sogenannte Energiemodell (EM), welches die Modelle der Schmelzöfen enthält. Im ersten Schritt wurde von Schmidt ein Modell auf Grundlage eines Wärmetauscheransatzes entwickelt [7, 8]. Dieses Modell wurde

anschließend von Henninger mithilfe eines iterativen Verfahrens verbessert, weiterentwickelt und validiert [9]. Durch die Validierung konnte eine gute Übereinstimmung zwischen realen Prozess und Simulationsmodell festgestellt werden. Mit dem Modellansatz ist es jedoch nicht möglich den Schmelzofen lokal aufzulösen, was dazu führt, dass Materialschichtungen mit unterschiedlichen Eigenschaften nicht detailliert abgebildet und untersucht werden können.

Um eine räumliche Auflösung zu ermöglichen wird in der vorliegenden Arbeit der aus der Modellierung von sensiblen Wärmespeichern bekannte „multi-node“ Ansatz auf den Schmelzschacht angewendet [10–12].

## 2 Grundlagen

In deutschen Gießereibetrieben mit eigener Schmelzerei werden häufig erdgasbefeuerte Schachtschmelzöfen zum Aufschmelzen von extern angelieferten und werksintern anfallenden Sekundäraluminium eingesetzt. Das Beschickungskonzept und der Aufbau des Ofens sind so konzipiert, dass neu eingebrachtes Material immer von oben eingebracht und bis zum Schmelzen nahezu die komplette Schachthöhe durchlaufen muss und somit eine maximale Verweildauer und Vorwärmung des Materials erreicht wird (Abbildung 1). In der Abbildung sind die durch unterschiedliche Beschickungsvorgänge eingebrachten Materialien, welche stark unterschiedliche Eigenschaften aufweisen können, durch Grenzen zueinander gekennzeichnet.

Die Energie für den Aufwärm- und Schmelzvorgang des Aluminiums wird durch die Verbrennung von Erdgas zu Verfügung gestellt. Durch die Erdgasbrenner wird ein heißes Rauchgas erzeugt, welches am unteren Ende des Schmelzschachtes mit maximaler Energie ein- und am oberen Ende mit minimaler Energie austritt. Aufgrund der hohen Energie des Rauchgases am unteren Ende und des Beschickungskonzeptes mit der damit verbundenen allmählichen Erwärmung des Aluminiums findet der Schmelzvorgang vorwiegend im unteren Teil des Schmelzschachtes statt. Das geschmolzene Aluminium verlässt den Schmelzschacht und das feste Aluminium rutscht immer von oben nach unten nach. Die Aluminiummasse im Schmelzschacht kann dabei sehr stark schwanken.

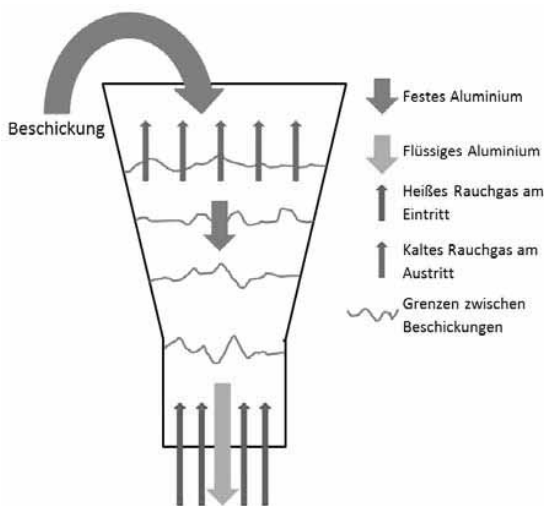


Abbildung 1: Schematische Darstellung - Aufbau und Funktionsprinzip des Schmelzofens

### 3 Grundlagen Modellierung

Ein weiteres Ziel neben der möglichst genauen Abbildung der physikalischen Vorgänge im Schmelzschacht ist das Erreichen einer möglichst geringen Rechenzeit. Diese wird benötigt um das Modell gegebenenfalls für eine modellprädiktive Betriebssteuerung verwenden zu können. In der Arbeit von Henninger wurde gezeigt, dass mit einem Prozesssimulationsmodell mit hohem Abstraktionsgrad sowohl eine geringe Rechenzeit wie auch eine ausreichende Ergebnisqualität für definierte Rahmenbedingungen erreicht werden kann [9]. Um den Schmelzofen und dessen Beschickungs- und Betriebsstrategien genauer untersuchen zu können wird der aus der Modellierung von sensiblen und latenten Wärmespeichern bekannte

„multi node“ Ansatz auf den Schmelzofen angewendet [10–12]. Dieser beruht auf der Zerlegung des betrachteten Speichers in eine beliebige Anzahl von Teilsegmenten mit idealer Durchmischung. Dadurch weisen alle Zustandsgrößen an jedem Punkt innerhalb eines Segmentes einen identischen Wert auf. Für jedes dieser Segmente wird dann auf Basis der Energie- und Massenbilanz ein Satz von Differentialgleichungen für die mathematische Beschreibung der ablaufenden Vorgänge erarbeitet.

#### 3.1 Annahmen und Vereinfachungen

Für die Anwendung des Ansatzes und die anschließende mathematische Modellierung werden die folgenden Annahmen getroffen.

- Vernachlässigung der realen Schachtschmelzofengeometrie
- Abbildung des zu schmelzenden Materials anhand von Materialgrößen, aber keine reale Darstellung des Materials
- Idealisierte Verteilung des eingebrachten Materials
- Vernachlässigung der Wärmeverluste an die Umgebung
- Idealisierte Berechnung der Wärmeübertragung mithilfe eines kombinierten Wärmeübergangskoeffizienten
- Idealisierte Berechnung der Schmelzleistung in Abhängigkeit der Energiebilanz

#### 3.2 Idealisierung und Zerlegung Schmelzschacht

Im Rahmen der Modellbildung wird der Schmelzschacht anhand der darin enthaltenen Aluminiummasse in eine variable Anzahl Schichten zerlegt (Abbildung 2). Dabei dient die Schichtnummer zur Beschreibung der Lage der Schicht und deren Abstand zu den Schmelzbrennern. Eine steigende Nummer ist immer mit einer höheren Entfernung zu den Schmelzbrennern und einer daraus resultierenden verringerten Rauchgastemperatur und Strömungsgeschwindigkeit verbunden. Die maximale Größe einer Schicht wird anhand der Masse begrenzt. Für jede Schicht werden die aufgelisteten Zustandsgrößen und deren Änderungen anhand eines mathematischen Modells berechnet. Generell kann dabei in jeder Schicht Aluminium geschmolzen werden.

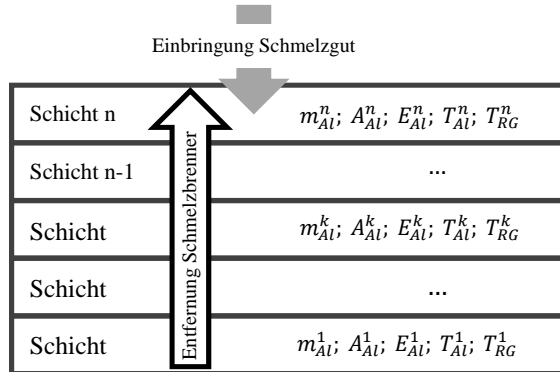


Abbildung 2: Schichteneinteilung eines idealisierten Schmelzschachtes mit den Zustandsgrößen Aluminiummasse  $m_{Al}$ ; Aluminiumoberfläche  $A_{Al}$ ; Energieinhalt Aluminium  $E_{Al}$ ; Aluminiumtemperatur  $T_{Al}$ ; Rauchgas-temperatur  $T_{RG}$

### 3.3 Aluminiumverteilung

Um die Vorteile des Modellierungsansatzes nutzen zu können muss darauf geachtet werden, dass keine Materialien mit unterschiedlichen Eigenschaften in einer Schicht vermischt werden. Die Konsequenzen aus dieser Forderung werden nun für die Grundvorgänge innerhalb des Schmelzschachtes beschrieben.

Wird Aluminium durch einen Chargiervorgang in den Schmelzschacht eingebracht, muss dieses auf die Schichten verteilt werden. Das entsprechende Verteilungskonzept ist in Abbildung 3 dargestellt. Das eingebrachte Material wird dabei immer in die erste komplett leere Schicht und je nach Größe der Beschickung auf die darüber liegenden Schichten verteilt.



Abbildung 3: Konzept für die Verteilung des eingebrachten Materials am Beispiel von 4 zeitlich nacheinander erfolgten Beschickungsvorgängen

### 3.4 Aufwärm- und Schmelzvorgang

Durch die Aktivierung der Schmelzbrenner wird das Aluminium erwärmt. Dies führt dazu, dass sich auch zwischen Schichten mit vorher identischen Eigenschaften, beispielsweise Schicht 3 und 4 in Abbildung 3, Unterschiede einstellen (Abbildung 4 links). Wird dem Aluminium ausreichend Energie zugeführt setzt der Schmelzvorgang ein. In diesem Fall schmilzt das

Aluminium der jeweiligen Schicht entsprechend der Energiebilanz ab. Damit ist die Abnahme der Aluminiummasse innerhalb der Schicht verbunden (Abb. 4 rechts). Aus den darüber liegenden Schichten kann kein Aluminium nachfließen, da diese andere Eigenschaften aufweisen. Die Verringerung der Aluminiummasse wird im Rahmen der Modellierung der Wärmeübertragung wieder aufgegriffen.

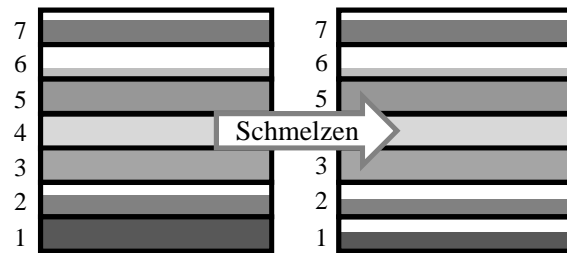


Abbildung 4: Auswirkungen der Materialaufwärmung und des Schmelzvorgangs auf die einzelnen Schichten ausgehend vom Startzustand Abbildung 3

Ist das Aluminium in einer Schicht komplett abgeschmolzen (Abbildung 5 links) rutscht das Aluminium jeder Schicht oberhalb der leeren Schicht um eine Schicht nach unten.

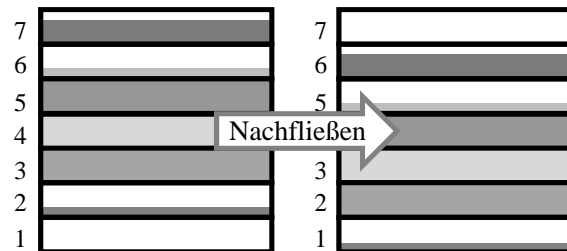


Abbildung 5: Interne Schichtverwaltung nach Abschmelzen einer Schicht (Nachfließvorgang); links: Aluminium der Schicht 1 komplett abgeschmolzen; rechts: Nachfließvorgang abgeschlossen

## 4 Modellierung

Es wird beschrieben wie das entwickelte Grundkonzept umgesetzt und modelliert werden kann. Dabei wird zunächst auf die Gesamtstruktur der Schmelzofenberechnung und anschließend auf die Berechnung der einzelnen Teilbereiche eingegangen.

### 4.1 Berechnungsablauf

Der Gesamtablauf der Berechnung erfolgt nach dem in Abbildung 6 visualisierten Ablauf, welcher für jeden Zeitschritt durchlaufen wird. Es werden zunächst die schichtexternen, und anschließend für jede Schicht die schichtinternen Vorgänge berechnet.

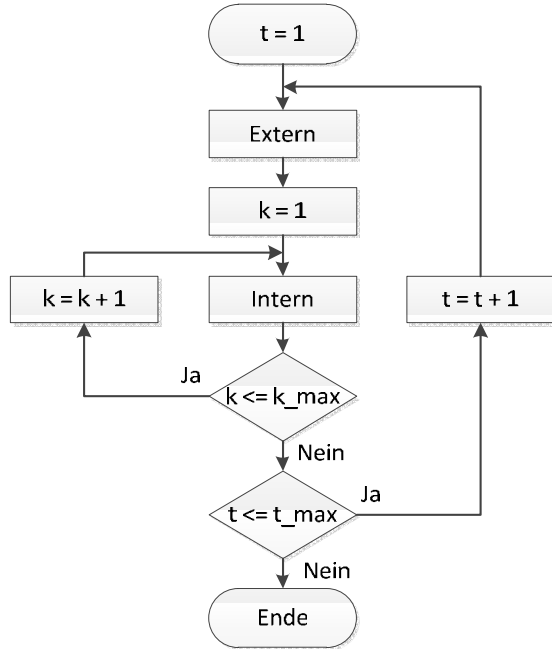


Abbildung 6: Aufbau und Ablauf der Schmelzschachtsimulation

## 4.2 Schichtexterne Vorgänge

Die schichtexternen Vorgänge umfassen den Nachfließ- und Beschickungsvorgang.

### 4.2.1 Nachfließen

Das Nachfließen erfolgt immer, wenn innerhalb einer Schicht das Aluminium komplett abgeschmolzen wird (vgl. Abb. 5). In diesem Fall werden die Zustands- und Prozessgrößen aller Schichten, welche eine höhere Schichtnummer aufweisen als die abgeschmolzene, an die Schicht mit einer um eins geringeren Schichtnummer übergeben.

### 4.2.2 Materialeinbringung

Die Materialverteilung des durch einen Beschickungsvorgang eingebrachten Aluminiums startet wie in Abbildung 3 gezeigt immer mit der komplett leeren Schicht, welche die kleinste Schichtnummer aufweist. Von dieser ausgehend wird das Aluminium in Abhängigkeit der Beschickungsmasse  $m_{Al,ein}$  und der maximalen Schichtkapazität  $m_{Al,max}$  auf die folgenden Schichten mit steigender Schichtnummer verteilt. Die maximale Anzahl der beschickten Schichten  $n_{max}$  wird nach (1) berechnet.

$$n_{max} = \left\lceil \frac{m_{Al,ein}}{m_{Al,max}} \right\rceil \quad (1)$$

Die Masse der Schichten, welche komplett mit Aluminium gefüllt werden können ergibt sich nach (2).

$$m_{Al}^k = m_{Al,max} \text{ für } k = b \dots e \quad (2)$$

mit:

$b = \text{Nr. der ersten Schicht}$

$e = b + n_{max} - 1 = \text{Nr. der letzten Schicht}$

Die Masse der letzten Schicht, welche durch die Beschickung Material erhält, ergibt sich nach (3).

$$m_{Al}^e = m_{Al,ein} - m_{Al,max} \cdot (n_{max} - 1) \quad (3)$$

Die beschriebene Verteilung der eingebrachten Aluminiummasse soll nun anhand eines kurzen Beispiels verdeutlicht werden. In diesem werden 523 kg eingebracht und die maximale Schichtkapazität  $m_{Al,max}$  beträgt 50 kg. Nach Gleichung (1) werden somit 11 Schichten beschickt. Nach Gleichung (2) werden die ersten 10 Schichten komplett (50 kg) und die elfte Schicht mit 23 kg beschickt. Neben der Masse werden mit der Beschickung die Temperatur und die spezifische Oberfläche des eingebrachten Materials vorgegeben.

## 4.3 Schichtinterne Vorgänge

Die schichtinternen Vorgänge können in die folgenden drei Teilbereiche, welche wiederum verschiedene Größen umfassen, eingeteilt werden.

### A) Wärmeübertragung

- Temperaturdifferenz  $\Delta T_{RG-Al}^k$
- Wärmeübergangskoeffizient  $\alpha^k$
- Wärmestrom  $\dot{Q}_S^k$

### B) Aluminiumkomponente

- Masse  $m_{Al}^k$
- Oberfläche  $A_{Al,RG}^k$
- Energie  $E_{Al}^k$
- Temperatur  $T_{Al}^k$

### C) Rauchgaskomponente

- Temperatur  $T_{RG,aus}^k$

### 4.3.1 Wärmeübertragung

In der Modellierung werden die Verluste an die Wand des Schmelzschachtes und somit an die Umgebung vernachlässigt. Somit muss nur der Wärmeübergang zwischen Aluminium (Al) und Rauchgas (RG) in jeder Schicht abgebildet werden. Die Berechnung erfolgt anhand von Formel (4), welche den kombinierten Wärmeübergangskoeffizienten  $\alpha^k$ , die wärmetauschenden Oberfläche zwischen RG und Al  $A_{Al,RG}^k$  und die Temperaturdifferenz zwischen RG und Al enthält.

$$\dot{Q}_S^k = \alpha^k \cdot A_{AL, RG}^k \cdot \Delta T_{RG-Al}^k \quad (4)$$

Der Koeffizient  $\alpha^k$  ist wiederum abhängig von der Position im Schmelzofen, welche durch die vom Rauchgas überströmte Oberfläche  $A_{\bar{u}}^k$  bis zum Erreichen der betrachteten Schicht  $k$  (6) erfasst wird, den Rauchgasmassenstrom am Schachteintritt  $\dot{m}_{RG, ein}$  und die Rauchgastemperatur am Schachteintritt  $T_{RG, ein}$  (5).

$$\alpha^k = f(A_{\bar{u}}^k, \dot{m}_{RG, ein}, T_{RG, ein}) \quad (5)$$

$$A_{\bar{u}}^k = \sum_{i=1}^{i=k-1} A_{Al}^i + \frac{A_{Al}^k}{2} \quad (6)$$

$A_{AL, RG}^k$  kann dabei in Abhängigkeit der Aluminiummasse  $m_{Al}^k$  und der spezifischen Oberfläche des Aluminiums  $A_{spez}^k$  berechnet werden (7)

$$A_{AL, RG}^k = m_{Al}^k \cdot A_{spez}^k \quad (7)$$

Der Wert für  $\Delta T_{RG-Al}^k$  ergibt sich aus der Berechnung der einfachen Temperaturdifferenz der Schichttemperaturen nach (8)

$$\Delta T_{RG-Al}^k = T_{RG, ein}^k - T_{Al}^k \quad (8)$$

#### 4.3.2 Aluminiumkomponente

Die Masse des Aluminiums in einer Schicht und dem gesamten Ofen, insbesondere das Abschmelzverhalten und die Schmelzleistung sind zentralen Eigenschaften des Schmelzofens. Die schichtinterne Änderung der Aluminiummasse entspricht dabei der Schmelzleistung der Schicht  $\dot{m}_{Al, aus}^k$  (9).

$$\frac{dm_{Al}^k}{dt} = -\dot{m}_{Al, aus}^k \quad (9)$$

Die Modellierung der Schmelzleistung basiert auf der Energiebilanz des Aufwärm- und Schmelzvorgangs. Eine wesentliche Größe dabei ist der theoretische Energiebedarf für diesen Vorgang. Dieser setzt sich aus einem sensiblen Anteil, welcher für die Aufwärmung des Aluminiums auf die Schmelztemperatur benötigt wird, und latenten Anteil, welcher für den Phasenübergang benötigt wird, zusammen. Die für den kombinierten Aufwärm- und Schmelzvorgang benötigte Energie  $E_{A+S}$  ergibt sich aus der Differenz zwischen Schmelztemperatur  $T_S$  und Referenztemperatur  $T_{Ref.}$  und der benötigten Schmelzenthalpie  $\Delta h_S$  (10).

$$E_{A+S}^k = m_{Al}^k \cdot (c_{pAl} \cdot (T_S - T_{Ref.}) + \Delta h_S) \quad (10)$$

Ziel der Modellierung der Schmelzleistung ist es, im Modell diese Energie während des Aufwärm- und Schmelzvorgangs an das Aluminium zu übertragen. Um dies zu erreichen wird die Berechnung an die durch die Wärmeübertragung an das Aluminium der Schicht übertragene Energie  $E_{tr, Al}^k$  (11) gekoppelt.

$$\frac{dE_{tr, Al}^k}{dt} = \dot{Q}_S^k \quad (11)$$

mit  $E_{tr, Al}^k(0)$  = anfänglicher Energieinhalt  
Übersteigt  $E_{tr, Al}^k$  einen bestimmten Anteil des theoretischen Energiebedarfs  $E_{A+S}^k$  wird die Höhe der Schmelzleistung nach (12) berechnet. Bei einer Änderung der Randbedingungen der Wärmeübertragung wird die Berechnung der Schmelzleistung erneut ausgeführt. Dies ist nötig um den Fehler aufgrund der Wärmeübertragungsprognose innerhalb eines definierten Bereichs zu halten.

Die Höhe der Schmelzleistung berechnet sich in Abhängigkeit der aktuellen Wärmeübertragungssituation und der bis zum Erreichen von  $E_{A+S}^k$  fehlenden Energiemenge (12). Aufgrund der konstanten Schmelzleistung und Rahmenbedingungen kann mit einem gemittelten Wert für die wärmetauschende Oberfläche  $\bar{A}_{AL, RG}^k$  gerechnet werden.

$$\dot{m}_{Al, aus}^k = \frac{m_{Al}^k \cdot \alpha^k \cdot \bar{A}_{AL, RG}^k \cdot \Delta T^k}{E_{Al, max}^k - E_{tr, Al}^k} \quad (12)$$

$$\text{mit } \bar{A}_{AL, RG}^k = \frac{A_{AL, spez}^k \cdot m_{Al, Start SV}^k}{2} \quad (13)$$

#### Energie und Temperatur

Die Werte für Energie und Temperatur werden für die Berechnung der Wärmeübertragung und Schmelzleistung benötigt. Die Änderung des Energieinhaltes im Aluminium  $\frac{dE_{Al}^k}{dt}$  ergibt sich durch den Wärmeaustausch zwischen Rauchgas und Aluminium und der Änderung durch das Abfließen des geschmolzenen Aluminiums (14).

$$\frac{dE_{Al}^k}{dt} = \dot{Q}_S^k - (\dot{m}_{Al, aus}^k \cdot c_{pAl} \cdot (T_{Al}^k - T_{Umg.}) + \dot{m}_{Al, aus}^k \cdot \Delta h_S) \quad (14)$$

Unter Anwendung des ersten Hauptsatzes der Thermodynamik kann die Temperatur des Aluminiums im sensiblen Bereich bestimmt werden. Für den Phasenübergang wird die Temperatur fest vorgegeben und entspricht der Schmelztemperatur (15).

$$T_{Al}^k = \begin{cases} \frac{E_{Al}^k}{m_{Al}^k \cdot c_{pAl}} - T_{Ref.} & \text{für } T_{Al}^k < T_{Al, S} \\ T_{Al, S} & \text{für } T_{Al}^k \geq T_{Al, S} \end{cases} \quad (15)$$

### 4.3.3 Rauchgaskomponente

In Bezug auf die Rauchgaskomponente muss lediglich die Temperatur am Austritt ausgehend von der Temperatur am Eintritt jeder Schicht berechnet werden. Die Änderung der inneren Energie einer Schicht wird dabei vernachlässigt (16). Für Schicht 1 ergibt sich die eingebrachte Leistung  $P_{ein}^1$  aus der Verbrennungsrechnung der verwendeten Schmelzbrenner. Für alle folgenden Schichten entspricht  $P_{ein}^k$  der Leistung am Austritt der vorhergehenden Schicht. Der Energieinhalt am Schichtaustritt ist abhängig von der eingebrachten Leistung und der in der Schicht übertragenen Energie (16).

$$P_{aus}^k = P_{ein}^k - \dot{Q}_S^k \quad (16)$$

In Kombination mit dem ersten Hauptsatz der Thermodynamik erhält man die Gleichung für die Berechnung der Rauchgastemperatur am Schichtaustritt (17).

$$T_{RG,aus}^k = \frac{\dot{m}_{RG,ein} \cdot T_{RG,ein}^k \cdot c_{pRG} - (\alpha^k \cdot A_{Al}^k \cdot \Delta T^k)}{\dot{m}_{RG,ein} \cdot c_{pRG}} \quad (17)$$

## 5 Validierung

Um die Qualität des Modells zu überprüfen wird mit diesem eine Validierung durchgeführt. Diese erfolgt anhand von Messdaten zu den wichtigsten Leistungsdaten eines Schmelzofens. Dazu gehören die Schmelzleistung, der Gasverbrauch, die Rauchgastemperatur am Schachtende und die Aluminiummasse im Schacht. Der Vergleich ergibt dabei für alle betrachteten Größen mittlere relative Abweichungen kleiner 10 Prozent.

## 6 Ausblick

Mit dem bereits validierten Modell sollen die Auswirkungen unterschiedlicher Beschickungsstrategien sowie der externen Materialvorwärmung auf die Leistungsdaten, wie die Schmelzleistung und den spezifischen Energieverbrauch des Ofens untersucht werden. Um die Qualität und Adaptierbarkeit des entwickelten Mehrschichtenmodells auch für unterschiedliche Schmelzofentypen und Leistungsklassen bewerten und gegebenenfalls verbessern zu können, müssen weitere Messdaten von verschiedenen Schmelzöfen erhoben werden. Im Rahmen der Messkampagne soll auch das Teillastverhalten der Öfen näher analysiert werden. Aus den gewonnenen Informationen soll das Modell im Bereich der Teillast verbessert werden.

## 7 References

- [1] M. Bosse, E. Frost, M. Hazrat, J.-M. Rhiemeier, and H. Wolff, "Ermittlung von branchenspezifischen Potentialen zum Einsatz von erneuerbaren Energie in besonders energieintensiven Industriesektoren am Beispiel der Gießerei-Industrie," Umweltbundesamt, Dez/2013.
- [2] C. Herrmann, H. Pries, and G. Hartmann, *Energie- und ressourceneffiziente Produktion von Aluminiumdruckguss*: Springer Berlin Heidelberg, 2014.
- [3] J. Hagenauer and A. Albertini, "Vergleichende Energieeinsatzbetrachtung von Druckguss und Kokillenguss zur Produktion von komplexen Gussteilgeometrien," *Giesserei-Rundschau*, no. 5/6, pp. 118–122, 2013.
- [4] P. Felder, U. Flückiger, T. Weisskopf, and T. Bürki, "Giessereibetrieb: Pinch-Analyse Prozessanlagen und Heizungsanlagen," Sektion Industrie und Dienstleistungen, Aug/2014.
- [5] C. Belt, "Current State of Aluminum Melting and Holding Furnaces in Industry," *JOM*, vol. 67, no. 11, pp. 2690–2695, 2015.
- [6] R. Riedel and R. Hille, "Energiesparwunder Neue Effizienzklasse bei Aluminium-Schmelzöfen: Best in Class - Highly - efficient Aluminum Melting Furnaces," *Giesserei-Rundschau*, vol. 59, no. 1/2, pp. 17–18,
- [7] J. Schmidt and W. Schlüter, "Ein dynamisches Prozesssimulationsmodell für die energetische Betrachtung von Aluminium-Schmelzöfen in einer betriebsumfassenden Materialflusssimulation," in *ASIM-Treffen STS/GMMS 2016: Workshop der ASIM/GI Fachgruppen STS und GMMS*, Wien: ARGESIM, 2016, pp. 29–36.
- [8] H. Scherf, *Modellbildung und Simulation dynamischer Systeme: Eine Sammlung von Simulink-Beispielen*. s.l.: Oldenbourg Wissenschaftsverlag, 2010.
- [9] M. Henninger and W. Schlüter, "Iterative Entwicklung und Verbesserung eines dynamischen Prozesssimulationsmodells eines Aluminiumschachtschmelzofens anhand globaler Prozessgrößen," in *Tagungsband: ASIM2016 - Symposium Simulationstechnik*, Wien: ARGESIM, 2016, pp. 33–40.
- [10] S. A. Klein, W. A. Beckman, and J. A. Duffie, "A design procedure for solar heating systems," *Solar Energy*, vol. 18, no. 2, pp. 113–127, 1976.
- [11] A. Athienitis and W. O'Brien, *Modeling, Design, and Optimization of Net-Zero Energy Buildings*: Wiley, 2015.
- [12] J. A. Duffie and W. A. Beckman, *Solar engineering of thermal processes*, 4th ed. Hoboken NJ: Wiley, 2013.

# Design of simulation model for novel solar thermal storage tank

Muthalagappan Narayanan, Prof. Dr.-Ing. Gerhard Menedoht, Prof. Dr. Walter Commerell

Hochschule Ulm, Ulm, Germany

*narayanan@hs-ulm.de*

Due to the fluctuated availability of solar energy, the solar thermal energy systems are designed by simulation as to have a system which will satisfy the loads as expected. The thermal storage tanks are existing since a long time but however there are still research going on and new novel tanks are introduced. One such tank is the Hybrid Quattro thermal storage tank from Sailer GmbH. This storage tank has a new patented lances which helps in stratification of the tank and thus higher efficiency of the energy storage without mixing of the layers in the tank. For designing of an energy system with this storage tank, a simulation model of this tank is required. Thus a test of this storage tank is done and the parameters were identified which later was used to build up a component model for storage in TRNSYS software using the type 340 storage tank model. This paper explains the process of identifying parameters from a real storage tank to a simulation model which can be used further for simulating a storage tank which would also make possible for the storage tank to be optimized for a development in the efficiency for the system performance.

## 1 Introduction

Currently Hochschule Ulm along with Sailer GmbH and other partners is working in the project - Intelligent Home Energy Management (iHEM) funded by BMWi. The main objective of this project is to integrate thermal and electrical components with more energy sources at high efficient operation in a household for high self utilization. For an intelligent control of such system, the load and production energy are forecasted and an optimum predictive operation strategy is found out so that the system is not only efficiently optimized and self utilization prioritized with maximum possible renewable energy ratio but also economically and strategically viable. The subjective goal of the project is to have an integrated home energy system communicating to all the components through a superior management component.

Hochschule Ulm is involved in simulating different buildings with TRNSYS and finding out the design and sizing of different thermal and electrical systems like fuel cell CHP, gas boiler, solar thermal collectors, PV, thermal storage tank, battery such that the demand of the buildings are met. Also one final step of the project is to erect a demonstrator with all these system components and showcase the intelligent control and integrated working of these. Also Hochschule Ulm is involved in design and monitoring of the demonstrator. More information about the project can be found at [1]

In project iHEM (intelligent home energy management), for system design the novel solar thermal storage tank from Sailer GmbH needs to be sized.

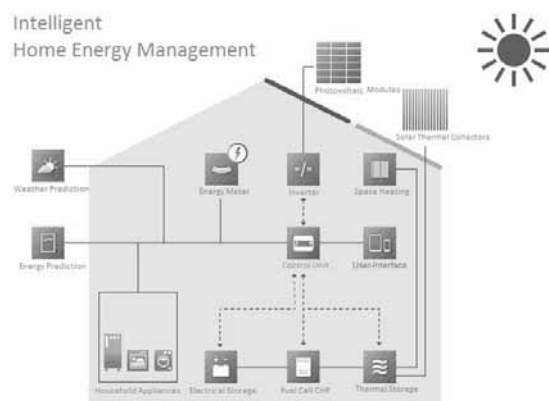


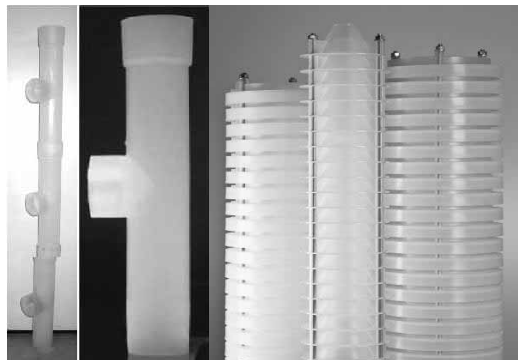
Figure 1. Overview of iHEM [1]

The storage tanks includes patented lances which is efficient enough to form layers in the tank such that the temperature gradients inside the tank is maintained and hence avoiding mixing which results in efficient storage of heat. The first sub-motive of the project is to size the storage tank for design of the entire energy system. Due to the fact that this is a complicated system, simulation of the system is required for designing the size of each component. For such simulations with high dynamic possibilities is the software TRNSYS (Transient System Simulation Tool) [2] famous. For the degree of freedom and its module plug and play concept, they are so flexible and customly designable. Each of the component in the system is a separate block in which the parameters can be defined. Also each of the output and input of the components can be interconnected so that they can communicate with each other. In TRNSYS there are lots of pre-developed

models available which can be connected together and integrated as a system, just like in Matlab Simulink. Further more, new component models can be coded and created. To design the system, the simulations have to be carried out with a storage tank model which would be able to simulate the Hybrid Quattro storage tank. Type 340 component model, pre-developed complicated tank component would be used here to model the tank. Below in section 1.1 and 1.2 the unique storage tank and the TRNSYS Type 340 model features are explained.

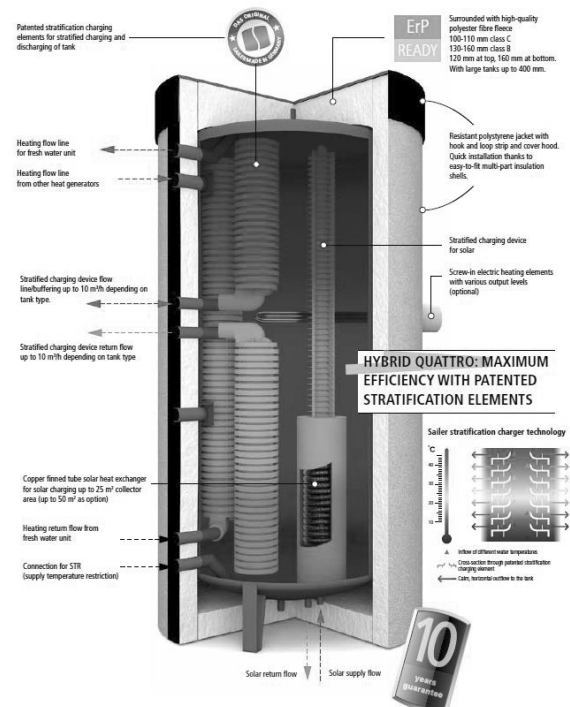
### 1.1 Hybrid Quattro storage tank

Usually a hot water storage tank is charged and discharged using fixed direct ports. So irregardless of what the temperature of the hot water produced by solar thermal collectors or the temperature in the layer where the port is, the hot water is injected in this layer and this in return mixes the tank layers and the temperature gradient inbetween the top and bottom is less. This causes the temperature of the discharge low. One more aspect is that the discharge return temperature depends on the application it is being used for and if return temperature is hotter and still useful this is also wasted by mixing the layers. For example when a solar collector produces 70°C hot water which is being fed into the top of the tank and after some time when the sun radiations are reduced the temperature reduces to say 50°C then this also is fed into the same layer in tank which mixes the layer and reduces the efficiency. For this the heat exchangers were built inside the tank. The same applies for the discharging, when warm water is being used for space heating in buildings, the return temperature really depends on the demand of heat enery in buildings and the flow rate of the radiators. Also one more factor is that the discharging outlet temperature also depends on the temperature taken out from the tank for discharge to deliver energy. Even though heat exchangers were better than direct charging and discharging, they were still not perfect and energy efficient as the heat exchangers cover only some layers and in these layers there may be a temperature gradient which might be destroyed. Thus came into existence, the stratification lances. These lances are mere plastic pipes which have opening at several layers inside the tank. The first kind of lances were developed by Solvis GmbH. These lances are normal pipes which has openings at perpendicular angles in a right angle direction. The Solvis lances are shown in **Figure 2** to the left. Also in the same figure you can see the Sailer lances in the right.



**Figure 2.** Stratification lances from Solvis GmbH (on the left) [3] and Sailer GmbH (on the right where the middle lance is used for solar energy i.e. natural convection input) [4]

Defenitely solvis lances were a break through but since it was right angled and directed to one point in a cylindrical layer, they did have some mixing. Hence were the newly developed Sailer lances, these had a 360 degree fool proof lances which had to take a convection fool proof design which made them even efficient. In a whole system point of view these tanks improve the efficient use of the energy system and higher useful energy ratio (useful energy/energy input).



**Figure 3.** Sectional view of the Hybrid Quattro tank with the stratification lances [5]

This combistore is claimed to be 60% more efficient than direct charging/discharging combistores by Sailer GmbH [5]. A comparison of the two stratification lances were made by Andersen, E et al through CFD simulations, forced flow and the thermosiphon flow tests [6].

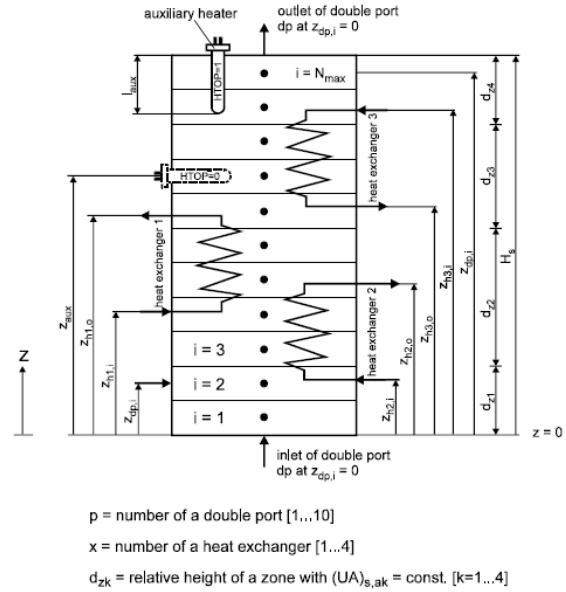
More efficient than the normal solar energy heat exchangers is the heat exchangers with stratification lances concept used here by the hybrid quattro tank. Since the heat exchangers if installed alone, the stratification is disturbed and when only lances are used glycol solution cannot be used in the tank. Until now tanks used an external heat exchanger which transfers heat from glycol loop to the water loop and the hot water is fed into the tank with lances. But here they use the natural convection of the water and the density in the heat exchange-lance combo. The heat exchanger inside the tank brings heat and transfers it to the near by water and due the fact that this water in the enclosure is high of density it moves via the lance to the respective layer and the cold water flows in the enclosure of the heat exchanger from the bottom to absorb the heat from the glycol-water solution from the solar collectors.

## 1.2 TRNSYS Type 340

The TRNSYS Type 340 is a storage model for a multiport tank component. This was developed by ITW, Uni-Stuttgart and is a no standard model available for purchase and can be easily integrated into the TRNSYS Library. The storage tank has a provision for up to 10 lances and 4 heat exchangers with up to 200 nodes (same temperature layers) and an electrical auxiliary heater. In **Figure 4**, it can be seen that the model is very flexible and is mathematically coded. The number of nodes can be input and they are equally divided from the input height and volume. The tank heat loss can be either given as a value same in all directions or can be varied between different parts of the store. The heat exchanger and the lances can be combined to have a heat exchanger which will send the hot fluid via lances to the stratified layer. Also all the lance's input height, output height, flow direction and also if it there is a lance (stratified or direct ports) can be given. Also the heat transfer capacity rate have to be input. There is also the facility to have one internal electrical auxiliary heater.

There are also some limitations to the model. It cannot have a lance with only one inlet/outlet port. It is assumed in the model that the mass flow in one lance,

i.e. in the outlet and inlet is meant to be a constant value.



**Figure 4.** Schematic of the multiport store model as it is in TRNSYS type 340 component model [7]

So always the lances should be connected to a closed loop in which the mass flow of the inlet and outlet of the lances should be same. Each of the heat exchanger is substituted for a lance and if I use the maximum available 4 heat exchangers then there is only 6 lances available for the connection. This is because whenever a heat exchanger is connected, there arises sometimes a need to have a lance connected to this heat exchanger. Thus mathematically either of the heat exchanger or the connected lance could only be used.

## 2 Methodology

As there are different types of hot water storage tanks and there are differences in how they perform a common test method in which parameters can be identified in order to compare them is required. Also for simulation of these tanks and designing a whole energy system these parameters are required. As a development of IEA Task 16 a European Standard EN 12977-4 was formed for Performance test analysis of solar combistores.

According to EN 12977-4, as specified in [8], there are different types of solar thermal storage tanks and briefly combistores can be classified into 5 types with direct/indirect charge and discharge modes. In this case for hybrid Quattro tank, the charge ports are both direct and indirect while the discharge ports are only direct

discharge. There are fresh water station heat exchangers externally which also have to be taken into account. Depending on the type of combistore different test methodology is specified and the tests are done in two types-entire store and a certain volume (auxiliary part) to determine heat loss capacity rate and thermal stratification loss (effective vertical thermal conductivity) respectively. Other than this the whole procedure gives many additional useful parameters like effective volume, heat transfer capacity rate, stratified charging capacity, etc. And the end of post processing of the test results with parameter fitting, the parameters for the respective tank is identified. The other physical parameters like inlet and outlet port heights, lances length, tank volume, height and sensors, heat exchanger and lances performances etc. can be obtained from the manufacturer. This standard testing can be done in any authorized test centers who has this combistore test bed.

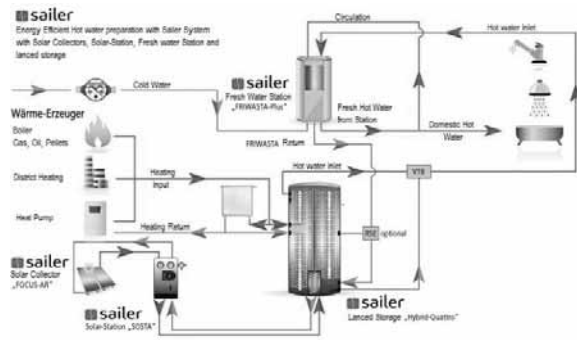
For this case, a 2 m<sup>3</sup> Hybrid Quattro Sailer GmbH's Combistore tank was tested by ITW, Uni-Stuttgart and the performance parameters were identified. The challenge then is to design a model to simulate such a complex storage tank. Not only in this project is it looked to simulate the storage tank but also to find an optimum tank parameters for different energy systems and buildings. The optimization of tank includes varying the height of the inlet and outlet ports, volume of the tank, electrical heater placement, electrical heater power, insulation, etc. Hence also the store size have to be varied and hence the parameters such as heat loss rate would vary and influence the simulation results. As known, the heat loss rate varies with respect to the insulation thickness, insulation material and surface area of the storage tank. Therefore basically the insulation can assumed to be in varied such a way that the surface area alone varies the heat loss rate, so that the magnification of the combistore for simulations is viable. The effective vertical thermal conductivity increases as the tank size decreases but as they are small in context they are neglected. Overall when there is a need to increase the size of the combistore, the height and the volume is modified with the heat loss modified and all the other parameters are withheld the same.

### 3 Simulation Realization

Now with the test results done and the parameters identified, the task following is to bring in these to the simulation model. As already explained TRNSYS software is used and type 340, a multiport storage model

is used. Even though this is one of the best component model for storage tanks, it has some limitations.

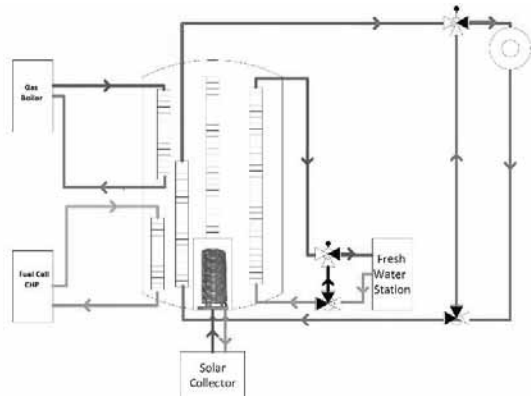
In **Figure 3**, it can be seen that there are two ports in the middle of the tank which is connected to the stratification lances. The main idea of these lances are that these two lances can be used in for two different operations together as shown in **Figure 5**.



**Figure 5.** Schematic of the Sailer hybrid quattro tank as it is functioning in the real system. [9]

In most of combistores there is an auxiliary boiler and is also connected to radiators for space heating. The principle of the radiators operation is that it takes out warm water from the middle of the tank i.e. from the top of a lance and returns back to the bottom of the same lance which inserts the water in the relevant temperature. Vice versa, for the auxiliary boiler, in this project a gas boiler, the warm water is taken from the middle of the tank i.e. from the bottom of another lance and is heated. The hot water from the gas boiler is fed back to the tank via the top of the same lance so that the hot water reaches the layer with relevant temperature. One more advantage of this configuration, is that during the winter days, without using the tank for buffer, the gas boiler can directly power the space heating radiators. One necessary equipment for this setup is a two way pump which is missing in TRNSYS library. Most importantly, type 340 is modelled or coded in such a way that always the input and output of each lance is two different ports and only with the direction of the lance, the component model defines if it is charging in energy or taking out. Also it is designed that the mass flow in between these input and output port is constant and hence if one of the port is not connected, it gives an error. Therefore it is practically possible to use a stratification lance with just one port but mathematically, not just in TRNSYS for type 340 but also in any other platform, this is not viable. Therefore this configuration setup has to be varied in simulation which differs from reality but the principle of the

lances is still feasible. The difference in the simulation model than reality can be seen in **Figure 6**. One more similar change is the lances connecting the fresh water station.



**Figure 6.** Overview of the simulation setup for the tank model.

The real operation is that the fresh water station takes hot water from the top of the tank via a direct port and exchanges the heat to the fresh cold water and the return warm water is fed to the lances which in turn delivers to the respective layer in the tank. Instead of this setup in the tank the inlet and outlet to the fresh water station is connected to the single lance.

## 4 Results and Conclusion

The primary aim of the paper is to build a component model for the hybrid quattro storage tank from Sailer GmbH. The process for building a simulation model for a storage tank is showcased in this paper. Since the storage tank is a novel tank with the stratification lances, the parameter identification and simulation turns out to be challenging. With certain conceptual changes between the real system and the simulation model, the simulation model is executed. The advantage of the simulation model is that the size of the tank can be varied and can be looked into for various cases. The simulation model also is helpful to vary the port inlet and outlet heights according to the system so that the total system efficiency is high.

In a following task as part of the iHEM project, the developed simulation model is used to build an energy system model consisting of solar thermal collectors, gas boiler, fuel cell CHP, building heat load and domestic hot water load. Also PV, battery and inverter is modelled so that the electrical auxiliary heater in the hot water storage tank is supplied with the excess electrical energy. Finally the motive of the project is to

compare the different buildings in Germany and the strategy for such a home energy system and the sizing. For example, for a sun house standard, the solar fraction needs to be minimum 50% while for a KfW55 house the primary energy used in the house should be less than 55 kWh/m<sup>2</sup>. Thus the collector area, PV size, storage size, etc are different for different buildings and the approach for each building types's system optimization is different. Not only that but also the tank also needs to be optimized for each of the building type according to their energy requirements and system size. The fact that Sailer GmbH have the option to build custom made different tanks according to the requirements of each customer is an advantage. So the end result might be to find a tank design and a sizing factor for each of the building types. Also is a real running system of all these components and their management strategies and control would be realized in a demonstrator system at the end of the project.

## 5 Acknowledgement

The work described in this paper is a part of the project *Intelligentes Heimenenergiemanagement (iHEM)* and the Research Initiative *Energiespeicher* funded by the *Bundesministerium für Wirtschaft und Energie* under the Project Number -03ET1205A.



## 6 References

- [1] "Photovoltaik Augsburg: Solarthermie, Photovoltaikanlage - iHEM." [Online]. Available: <http://www.ihem.eu/>. [Accessed: 13-Jan-2017].
- [2] *Transient System Simulation Tool - TRNSYS*. Madison USA.: Thermal Energy System Specialists, LLC., 2014.
- [3] L. J. Shah, E. Andersen, and S. Furbo, "Theoretical and experimental investigations of inlet stratifiers for solar storage tanks," *Appl. Therm. Eng.*, vol. 25, no. 14–15, pp. 2086–2099, Oct. 2005.
- [4] "Sailer GmbH Ehingen - Patented stratification charger technology." [Online]. Available: <http://www.sailergmbh.de/en/products/sailer-stratified-storage-tanks/patented-stratification-charger-technology.html>. [Accessed: 13-Jan-2017].
- [5] "Sailer GmbH Ehingen - Schichtenspeicher." [Online]. Available: <http://www.sailergmbh.de/en/products/schichtenspeicher/schichtenspeicher.html>. [Accessed: 13-Jan-2017].

- [6] E. Andersen, S. Furbo, M. Hampel, W. Heide-  
mann, and H. Müller-Steinhagen, "Investigations  
on stratification devices for hot water heat stores,"  
*Int. J. Energy Res.*, vol. 32, no. 3, pp. 255–263,  
Mar. 2008.
- [7] H. Drück, "Mutliport Store Model for TRNSYS -  
Type 340." Institut für Thermodynamik und Wär-  
metechnik (ITW), Universität Stuttgart, Mar-  
2006.
- [8] *FprEN 12977-3:2016, Thermal solar systems and  
components - Custom built systems - Part 4: Per-  
formance test methods for solar combistores.*
- [9] "Sailer GmbH Ehingen - Diagram: Hybrid Quat-  
tro in heating circuit." [Online]. Available:  
[http://www.sailergmbh.de/en/products/sailer-  
stratified-storage-tanks/diagram-hybrid-quattro-  
in-heating-circuit.html](http://www.sailergmbh.de/en/products/sailer-stratified-storage-tanks/diagram-hybrid-quattro-in-heating-circuit.html). [Accessed: 13-Jan-2017].

# Simulation von Strömung und Wärmeübergang in der Energietechnik mit OpenFOAM®

Peter Renze<sup>1</sup>

<sup>1</sup>Institut für Energie- und Antriebstechnik, Hochschule Ulm, Eberhard-Finckh-Str. 11, 89075 Ulm  
[renze@hs-ulm.de](mailto:renze@hs-ulm.de)

Strömungsmechanik und Wärmeübertragung spielen eine wesentliche Rolle in jedem thermischen Energieprozess. Eine Verbesserung des Wärmeübergangs in Apparaten kann deren Wirtschaftlichkeit erhöhen und den Ressourcenverbrauch senken. Zur Entwicklung innovativer Produkte in diesem Bereich ist ein genaues Verständnis der komplexen Wärmetransportphänomene in technischen Apparaten notwendig. Dies kann heutzutage durch numerische Simulationen (CFD) erlangt werden. Allerdings erfordern reale technische Problemstellungen häufig die Berücksichtigung dynamischer Effekte und komplexer 3D-Wandstrukturen. Dies führt zu einem enormen Bedarf an Rechenressourcen, insbesondere da sich die zeitlichen und räumlichen Skalen beim Wärmetransport über viele Größenordnungen erstrecken. Hierfür bietet sich der Einsatz der quelloffenen Softwarebibliothek OpenFOAM® als Basis an, die massiv paralleles Rechnen unterstützt. Dies wird am Beispiel der numerischen Berechnung der Intensivierung des Wärmeübergangs in thermischen Anlagen mit Hilfe von Large-Eddy-Simulation (LES) demonstriert.

## 1 Einleitung

Moderne Hochleistungs-Wärmeübertrager sind von großer Bedeutung in vielen Industriezweigen, darunter sind Chemie, Energie (Öl, Gas, Strom), Kälte/Klima und Automobil. In den dortigen Produktionsprozessen besteht zunehmend die Notwendigkeit, bei steigender Produktion die Menge der eingesetzten Ressourcen zu senken, die Ressourcenproduktivität zu steigern und somit den damit verbundenen  $CO_2$ -Ausstoß zu senken. Effizientere Wärmeübertragungstechnologien können hier ein Schlüssel sein, um diese hohen Anforderungen zu erfüllen. Allerdings müssen dafür die bestehenden Wärmeübertragungstechnologien optimiert werden.

Hierfür bieten sich moderne Simulationsverfahren wie Computational Fluid Dynamics (CFD) an. Für stationäre Wärmeübertragungsprobleme auf komplexen Geometrien entsteht allerdings ein massiver Bedarf an Rechenressourcen. Die quelloffene Softwarebibliothek OpenFOAM® kann hier eine Lösung sein, was in dieser Arbeit am Beispiel der numerischen Berechnung der Intensivierung des Wärmeübergangs an Rohren demonstriert wird.

Unter dem Begriff *Verbesserung des Wärmeübergangs* (englisch: heat transfer enhancement oder he-

at transfer augmentation) versteht man klassisch die Erhöhung des Wärmedurchgangskoeffizienten  $k$ , so dass entweder bei gleichem Wärmestrom die Fläche oder die Temperaturdifferenz kleiner werden oder bei gleicher Fläche und Temperaturdifferenz der Wärmestrom steigt. Beides führt zu einer höheren Effektivität des Apparats. Dieses Thema ist aufgrund seiner hohen Bedeutung international stark beforcht, siehe [1, 2].

Sogenannte passive Maßnahmen zur Verbesserung des Wärmeübergangs sind z. B. strömungswirksame Einbauten oder die Modifikation der Wandstrukturen des Wärmeübertragers. In dieser Publikation steht letzteres im Vordergrund. Durch eine Modifikation der Wandstrukturen werden Störungen in den wandnahen Grenzschichten erzeugt, die diese entweder verkleinern und destabilisieren oder durch kohärente Wirbelstrukturen und/oder Turbulenzanregung deren Wärmewiderstand reduzieren.

## 2 Methoden

### 2.1 OpenFOAM®

OpenFOAM® (Open Source Field Operation and Manipulation) ist eine objektorientierter Softwarebi-

blibliothek programmiert in C++, die für die numerische Lösung kontinuumsmechanischer Probleme eingesetzt werden kann, siehe Jasak und Weller [3]. Das Hauptaugenmerk liegt dabei auf dem Lösen von Strömungsproblemen. OpenFOAM® wurde 2004 erstmals als OpenSource deklariert und frei zugänglich gemacht. Inzwischen ist es unter GPL lizenziert und wird als Grundlage für Berechnungssoftware im akademischen und industriellen Umfeld intensiv genutzt.

OpenFOAM® wird mit einer Reihe vorimplementierter Löser verteilt. Üblicherweise wird eine Finite-Volumen-Methode (FVM) verwendet, es ist aber auch die Finite-Elemente-Methode (FEM) prinzipiell verfügbar. Neue Lösungsverfahren können durch Adaption und Kombination der vorhandenen physikalischen Modelle in Form von Klassen leicht implementiert werden, z.B. stehen innerhalb der Turbulenz-Klasse verschiedene Modelle für RANS, LES, DNS oder hybride Verfahren zur Verfügung.

Durch eine Top-Level-Struktur wird eine vergleichsweise einfache Programmierung von Lösungsverfahren für spezielle Anwendungen möglich, denn OpenFOAM® ist hochgradig objektorientiert, d.h. der Fokus liegt auf Objekten und nicht Funktionen. Das Preprocessing erfolgt klassisch auf der Kommandozeile ohne GUI, allerdings stehen heutzutage eine Reihe (teils kommerzieller) externer Programme zur Verfügung. Unstrukturierte Gitter hexagonaler oder hexagonaler Zellen können über zwei verschiedene Verfahren erstellt werden. Die graphische Aufbereitung der Simulationsergebnisse (Postprocessing) geschieht standardmäßig mit Hilfe von ParaView [4].

## 2.2 Physikalische Modellierung

Die physikalische Modellierung von Strömungen mit Wärmetransport erfolgt über die Erhaltungsgleichungen für Masse, Impuls und Energie. Die Erhaltung der Masse wird bei instationärer Strömung eines kompressiblen Mediums beschrieben durch

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (1)$$

mit dem Nabla-Operator  $\nabla = \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$ , der Dichte  $\rho$  und dem Vektor der Geschwindigkeit  $\mathbf{u}$ . Die Impul-

serhaltung ist formuliert als

$$\begin{aligned} \frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) &= -\nabla p + \rho \mathbf{g} \\ &+ \nabla \cdot (2\mu_{eff} \mathbf{S}(\mathbf{u})) - \nabla \cdot \left( \frac{2}{3} \mu_{eff} (\nabla \cdot \mathbf{u}) \right), \end{aligned} \quad (2)$$

wobei  $p$  der statische Druck ist,  $\mathbf{g}$  der Gravitationsvektor, und  $\mu_{eff}$  ist die Summe aus molekularer und turbulenter Viskosität. Es ist  $\mathbf{S}(\mathbf{u})$  der Deformationstensor definiert als  $\mathbf{S}(\mathbf{u}) = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \cdot \mathbf{u})^T)$ .

Die Energieerhaltung wird über die innere Energie  $e$  ausgedrückt

$$\begin{aligned} \frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho \mathbf{u} e) &+ \frac{\partial (\rho k)}{\partial t} + \\ \nabla \cdot (\rho \mathbf{u} k) + \nabla \cdot (p \mathbf{u}) &= \\ \nabla \cdot (\alpha_{eff} \nabla e) + \rho \mathbf{u} \cdot \mathbf{g}, \end{aligned} \quad (3)$$

wobei hier  $k = |\mathbf{u}|^2/2$  die kinetische Energie pro Masseneinheit ist. Die effektive thermische Diffusivität  $\alpha_{eff}$  wird dabei gebildet aus

$$\alpha_{eff} = \frac{\rho v_t}{Pr_t} + \frac{\mu}{Pr} = \frac{\rho v_t}{Pr_t} + \frac{\lambda}{c_p}, \quad (4)$$

wobei  $\lambda$  die thermische Leitfähigkeit,  $c_p$  die Wärmekapazität bei konstantem Druck,  $Pr_t$  die (turbulente) Prandtl-Zahl und  $v_t$  die kinematische (turbulente) Viskosität. Die turbulenten Transportparameter in den Gleichungen (1) bis (3) müssen durch ein Turbulenzmodell zur Schließung der Gleichungen angenähert werden.

Um in diesen Strömungskonfigurationen nicht nur die Wandschubspannung, sondern auch den lokalen Wärmeübergang mit hoher Genauigkeit vorhersagen zu können, sind Modifikationen an den bestehenden Modellen für den turbulenten Wärmetransport notwendig. Bisher wurde mit dem isotropen Konzept der turbulenten Prandtl-Zahl gearbeitet, aber in den letzten Jahren werden zunehmend Tensormodelle verwendet. Eine erschöpfende Diskussion ist hier nicht möglich, aber letzte Erfolge in der RANS-Entwicklung waren z.B. das v2-f-Modell [5, 6] oder Skalenadaptive Verfahren [7]. In den letzten Jahren ist die Entwicklung von RANS-Modellen stark zurückgegangen, was zum einen an den mangelnden Erfolgen und zum anderen an den rasant wachsenden Rechenressourcen liegt, die den Einsatz aufwendigerer Verfahren erlauben.

Eine Lösung bietet hier die Grobstruktursimulation (large-eddy simulation, LES). Hier werden nur die energiereichen, aber grobskaligen Wirbel turbulenter

Strömungen direkt aufgelöst und nur die kleineren Skalen modelliert [8, 9, 10]. Dies reduziert den Rechenaufwand gegenüber der DNS enorm, erhöht aber die Genauigkeit gegenüber RANS-Verfahren ebenso stark, was diese Methode auch für industriellen Einsatz attraktiv macht. Eine Übersicht der generellen Verfahren ist in [11] gegeben. Diese Ansätze zur Turbulenzmodellierung haben sich für viele technische Anwendungen bewährt [12, 13, 14] und sind ebenfalls bei Mehrphasenströmungen einsetzbar, wie in [15, 16, 17] gezeigt wurde.

In der Grobstruktursimulation wird eine Filterfunktion mit einer charakteristische Filterweite  $\Delta$  auf die obigen Gleichungen angewendet, so dass die Feldgrößen in einen aufgelösten und einen nicht-aufgelösten Anteil aufgeteilt werden. Somit wird die Geschwindigkeit zu  $\mathbf{u} = \hat{\mathbf{u}} + \mathbf{u}'$ . Die oben dargestellte effektive Viskosität wird nun berechnet als  $\nu_{eff} = \nu + \nu_{SGS}$ , also der Summe aus molekularer und der scheinbaren Viskosität der Subgridskalen.

Diese wird gemäß der Boussinesq-Approximation als  $\nu_{SGS} = C_k \Delta \sqrt{k_{SGS}}$  modelliert, mit  $C_k = 0.07$  und  $k_{SGS}$  als kinetische Energie der Subgridskalen. Für  $k_{SGS}$  werden in dieser Arbeit zwei Modelle verwendet, der Smagorinsky-Ansatz [8] und der Eingleichungsansatz. Im Smagorinsky-Modell gilt

$$k_{SGS} = 2 \frac{C_k}{C_\epsilon} \Delta^2 |\mathbf{S}|^2, \quad (5)$$

mit  $C_\epsilon = 1.048$  und  $|\mathbf{S}| = (\mathbf{S} : \mathbf{S})^{1/2}$ .

Im Eingleichungsmodell wird eine zusätzliche Transportgleichung für die turbulente kinetische Energie der Subgridskalen  $k_{SGS}$  gelöst:

$$\begin{aligned} \frac{\partial k_{SGS}}{\partial t} + \nabla \cdot (k_{SGS} \mathbf{u}) - \nabla \cdot \left( \frac{\mu_{eff}}{\rho} \nabla k_{SGS} \right) \\ = 2 \nu_{SGS} |\mathbf{S}|^2 - C_\epsilon k_{SGS}^{3/2} / \Delta. \end{aligned} \quad (6)$$

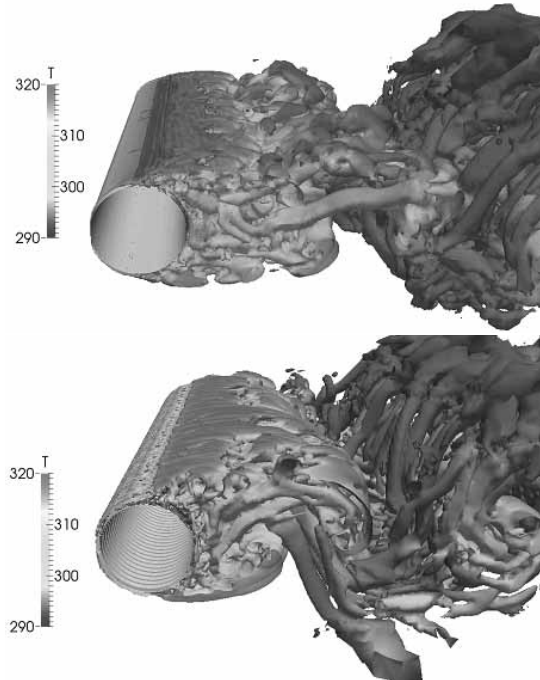
### 3 Ergebnisse

In diesem Kapitel sind die vorläufigen Ergebnisse der laufenden Projekte bei der Simulation des Wärmeübergangs an Rohren dargestellt. Diese Arbeiten sind Voruntersuchungen zur Validierung des eingesetzten Workflows.

#### 3.1 Simulation des Wärmeübergangs bei der Umströmung von Rohren

In Abbildung 1 sind die instationären Vorgänge beim Wärmeübergang mit erzwungener Konvektion an einem horizontalen Rohr visualisiert. Die Reynolds-Zahl gebildet mit dem Außendurchmesser des Rohrs  $D$  beträgt  $Re_D = 6900$ , d.h. die Strömung ist turbulent. Hier wurde eine konstante Wandtemperatur an der Außenwand des Rohres eingestellt und der Wärmeübergang an die umströmende Luft simuliert. In Abbildung 1 (oben) erkennt man die charakteristischen kohärenten Strukturen im turbulenten Strömungsfeld bei der periodische Ablösung stromab des Rohrs.

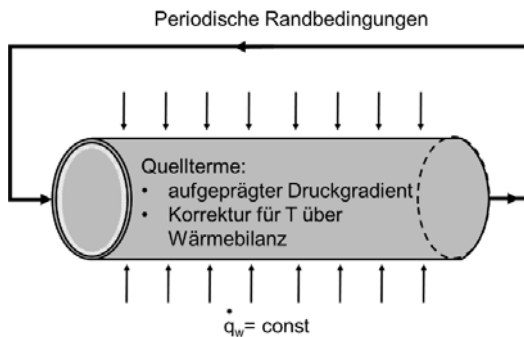
In Abbildung 1 (unten) sind die Ergebnisse der gleichen Simulation bei einer strukturierten Oberfläche aus Mikrorippen dargestellt. Man erkennt eine deutliche Beeinflussung des Ablöseverhaltens, und es wurde eine Verbesserung des Wärmeübergangs im Bereich von 20% festgestellt.



**Abbildung 1:** Grobstruktursimulation (LES) einer turbulenten Strömung  $Re_D = 6900$  mit erzwungener Konvektion an einem glatten Rohr (oben) und einem Rohr mit Mikrorippen (unten)

### 3.2 Simulation des Wärmeübergangs bei Durchströmung von Rohren

Die exakte Berechnung des Wärmeübergangs bei Innenströmungen mittels LES ist deutlich schwieriger als bei Außenströmungen. Das Problem ist die Erzeugung turbulenter Strömungsdaten am Eintritt in das Rohr. Da nur eine begrenzte Länge des Rohrs (ca. 12 Durchmesser) simuliert werden kann, ist der Einfluß der Eintrittsrandbedingung enorm. Es müssen turbulente Eingangsdaten für den Eintritt der Strömung in die Rechendomäne künstlich generiert werden. Unter der Annahme einer vollausgebildeten turbulenten Rohrströmung kann dies mit Hilfe von zyklischen Randbedingungen an Eintritt und Austritt erreicht werden, wobei ein Druckgradient künstlich aufgeprägt werden muss. Das Setup ist in Abbildung 2 dargestellt. Der Wärmestrom an der Wand wird als konstant angenommen oder als konstante Wandtemperatur, je nach Validierungsfall. Beides erfordert eine Korrektur des Temperaturprofils am Eingang, für das ebenfalls periodische Randbedingungen verwendet werden.

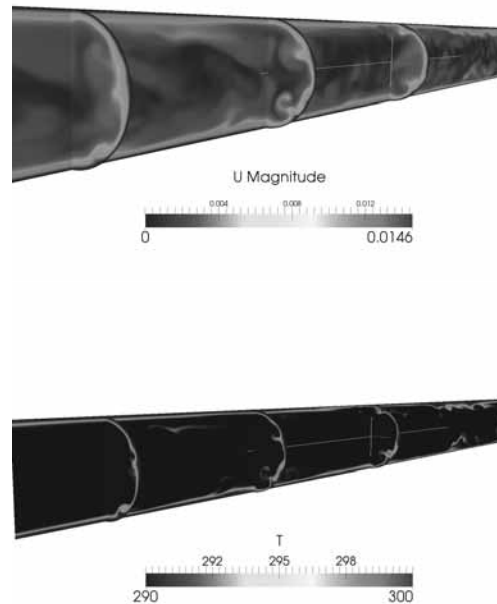


**Abbildung 2:** Skizze des numerischen Setups und der Randbedingungen für die LES-Simulationen der Rohrinnenströmung.

Die Rechengitter wurden mit dem OpenFOAM-Gittergenerierungswerkzeug *snappyHexMesh* erstellt. Eine Wandauflösung von  $y^+ = 0.1$  wird durch eine 12fache Prismaschicht erreicht. Das unstrukturierte Gitter besteht im Kern aus kubischen Zellen, was eine entsprechend hohe Zellanzahl für die Simulation mit sich bringt. Die Gitter haben um die 6 Millionen Zellen für die niedrige Reynoldszahl von  $Re_\tau = 180$  (Reynolds-Zahl gebildet mit der Schubspannungsgeschwindigkeit). Bei den in der Literatur üblichen strukturierten Gittern kann diese

Anzahl um den Faktor 10 niedriger liegen, da die Zellen in Strömungsrichtung stark gestreckt werden. Allerdings bietet dieses Vorgehen den entscheidenden Vorteil, das mit geringem Nutzaufwand die glatte Wand durch eine berippte ersetzt werden kann, wie im folgenden gezeigt wird.

In Abbildung 3 sind Ausschnitte aus dem 3D-Strömungsfeld an beliebigen Zeitpunkten dargestellt. Man erkennt die kohärenten Wirbelstrukturen im Geschwindigkeitsfeld (oben) und die transienten Transportprozesse beim Wärmeübergang (unten).



**Abbildung 3:**  $Re_\tau = 180$ , Geschwindigkeit (oben) und Temperatur (unten)

In Abbildung 4 sind die Ergebnisse nach der zeitlichen Mittelung des Strömungsfelds über 20 Durchläufe gezeigt. Das erwartete logarithmische Geschwindigkeitsprofil wird für zwei Simulationen mit unterschiedlichen Reynoldszahlen sehr gut getroffen. Der Vergleich des Wärmeübergangs in Form von Nusselt-Zahlen mit Werten aus der Literatur ist in Tabelle 1 gegeben.

In Abbildung 5 sind erste Ergebnisse einer Simulation bei einer Reynoldszahl von  $Re_\tau = 180$  mit Mikrorippen dargestellt. Das Gitter wurde automatisch erstellt und

Fall	LES	Gnielinski [18]
$Re_\tau = 180, Pr = 7$	45.2	48.8
$Re_\tau = 314, Pr = 7$	86.9	81.7

**Tabelle 1:** Vergleich der berechneten Nusselt-Zahlen aus der LES-Simulation mit den Literaturwerten (Gnielinski [18]).

hat eine Zellanzahl von 30 Millionen Zellen. In weiteren Untersuchungen wird der Effekt auf die Transportprozesse in der turbulenten Strömung genau untersucht.

## 4 Zusammenfassung

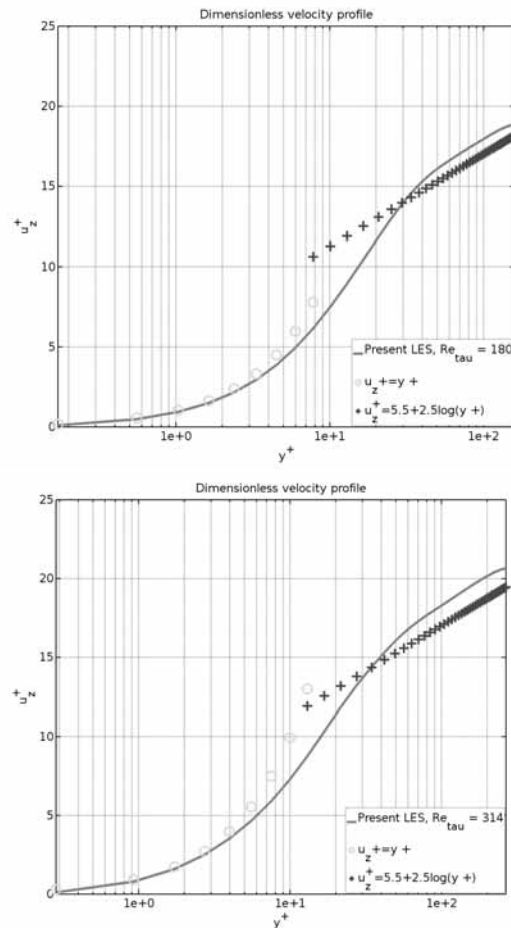
Die Softwarebibliothek OpenFOAM wird von einer großen Zahl Nutzern im akademischen und industriellen Umfeld zur Berechnung von Strömungen und Transportphänomenen verwendet. Sie ist frei verfügbar, auf vielen Hochleistungsrechenzentren vorinstalliert und kann für massiv paralleles Rechnen verwendet werden. Der Einsatz von OpenFOAM für die numerische Berechnung des Wärmeübergangs in technischen Apparaten wurde anhand einiger Beispiele demonstriert. Die bisherigen Ergebnisse der laufenden Forschungsarbeiten sind sehr vielversprechend. Die Ergebnisse der Simulation turbulenter Rohrrinnenströmungen mit Wärmeübergang konnten erfolgreich validiert werden. Es folgen Untersuchungen zur Verbesserung des Wärmeübergangs mit dieser Methode.

## Acknowledgements

Der Autor dankt dem Steinbuch Center for Computing (SCC) am Karlsruher Institut für Technologie (KIT) für die zur Verfügung gestellte Rechenzeit auf dem Forschungshochleistungsrechner ForHLR 2 [19].

## Literatur

- [1] A. E. Bergles, M. K. Jensen, B. Shome, *Bibliography on Enhancement of Convective Heat and Mass Transfer*. Heat Transfer Laboratory Report No HTL-23 Rensselaer Polytecnic Institute, 1995.



**Abbildung 4:**  $Re_\tau = 180, Pr = 7$  (oben)  $Re_\tau = 314, Pr = 7$  (unten)

- [2] R. L. Webb, N. H. Kim, *Principle of enhanced heat transfer*. 2nd Edition. Taylor Francis: New York, NY, USA, 2005.
- [3] H. Jasak H., H.G. Weller, *Interface tracking capabilities of the InterGamma differencing scheme*, Technical Report, Imperial College, University of London, 1995.
- [4] J. Ahrens, B. Geveci, C. Law, *ParaView: An End-User Tool for Large Data Visualization*, Visualization Handbook, Elsevier, 2005, ISBN-13: 978-0123875822.
- [5] Laurence, D.R., Uribe J.C., Utyuzhnikov, S.V. *A Robust Formulation of the v2-f Model*, Flow Turbulence and Combustion, 73, 169-185, 2004.

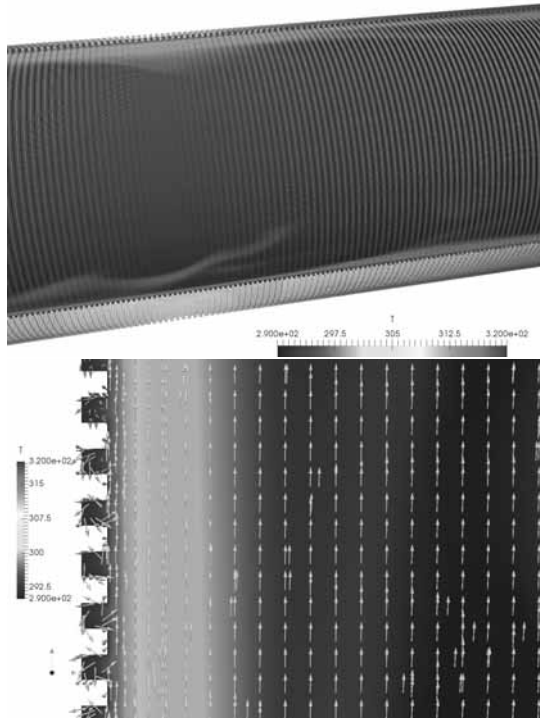


Abbildung 5:  $Re_\tau = 180$ ,  $Pr = 7$

- [6] Popovac, M., Hanjalic, K., *Compound Wall Treatment for RANS Computations of Complex Turbulent Flows and Heat Transfer*, Flow Turbulence and Combustion, Nr. 78, S. 177-202, 2007.
- [7] Menter, F. R., and Y. Egorov. *The scale-adaptive simulation method for unsteady turbulent flow predictions. Part 1: theory and model description*. Flow, Turbulence and Combustion 85.1 (2010): 113-138.
- [8] J. Smagorinsky. *General circulation experiments with the primitive equations, part I: The basic experiment*. Monthly Weather Review, 91:99–164, 1963.
- [9] Germano, M., Piomelli, U., Moin, P. and Cabot, W. H. (1991), *A dynamic sub-grid scale eddy viscosity model*, Physics of Fluids, A(3): pp 1760-1765, 1991.
- [10] You, D. and Moin, P. (2007), *A dynamic global-coefficient subgrid-scale eddy-viscosity model for large-eddy simulation in complex geometries*, Physics of Fluids, 19(6): 065110, 2007.
- [11] Sagaut, Pierre (2006). *Large Eddy Simulation for Incompressible Flows (Third ed.)*. Springer. ISBN 3-540-26344-6.
- [12] P. Renze, W. Schröder, M. Meinke, *Large-eddy simulation of film cooling flows at density gradients*, International Journal of Heat and Fluid Flow 29.1; 18-34, 2009.
- [13] P. Renze, W. Schröder, M. Meinke, *Large-eddy simulation of film cooling flows with variable density jets*, Flow, Turbulence and Combustion 80.1 (2008): 119-132, 2008.
- [14] P. Renze, W. Schröder, M. Meinke, *Large-eddy simulation of film cooling flow ejected in a shallow cavity*, AIAA Paper 927, 2007.
- [15] P. Renze, K. Heinen, M. Schönherr. *Experimental and Numerical Investigation of Pressure Swirl Atomizers*. Chem Eng Tech, 34 (7):1191–1198, 2011.
- [16] P. Renze, A. Buffo, D. L. Marchisio, M. Vanni, *Simulation of Coalescence, Breakup, and Mass Transfer in Polydisperse Multiphase Flows*, Chemie Ingenieur Technik, 86(7):1088–1098, 2014.
- [17] A. Buffo, D. L. Marchisio, M Vanni, P. Renze, *Simulation of polydisperse multiphase systems using population balances and example application to bubbly flows*, Chem. Eng. Res. Des., 91(10):1859–1875, 2013.
- [18] Verein Deutscher Ingenieure, VDI-Gesellschaft Verfahrenstechnik und Chemieingenieurwesen (VCI). VDI-Wärmeatlas: Berechnungsblätter für den Wärmeübergang. 11. Auflage, 2013.
- [19] Steinbuch Center for Computing. Forschungshochleistungsrechner ForHLR 2. <http://www.scc.kit.edu/dienste/forhlr2.php>, 2016. Accessed: November 16, 2016.

# Systemdesign eines Batteriemanagementsystems für Elektrofahrzeuge

Sven Jacobitz<sup>1</sup>, Sören Scherler<sup>2</sup>, Xiaobo Liu-Henke<sup>3</sup>

Ostfalia Hochschule für angewandte Wissenschaften, Fakultät Maschinenbau,

Institut für Mechatronik, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel

<sup>1</sup>sve.jacobitz@ostfalia.de, <sup>2</sup>so.scherler@ostfalia.de, <sup>3</sup>x.liu-henke@ostfalia.de

Lithium-Ionen-Batterien sind Grundlage der modernen Elektromobilität, da sie über eine verhältnismäßig hohe Energiedichte bei gleichzeitig hoher Leistungsfähigkeit verfügen. Aufgrund ihrer Empfindlichkeit gegenüber Verlassen des vorgegebenen Betriebsbereiches ist der Einsatz von Batteriemanagementsystemen (BMS) erforderlich. Dieser Beitrag fokussiert das ganzheitliche Systemdesign eines BMS, bestehend aus Hard- und Softwarekomponenten, für Lithium-Ionen-Batterien. Mittels durchgängig modellbasierter Methodik, basierend auf MiL-, SiL- und HiL-Simulationen erfolgt der Entwurf sowie die Validierung der Hard- und Softwarekomponenten. Beispielhaft für den Einsatz dieser Entwicklungsmethodik wird in diesem Beitrag die Auslegung eines Extended Kalman Filters zur Schätzung des Ladezustands dargestellt. Abschließend wird das aus Funktionsintegration entstandene BMS mittels HiL-Simulation unter Echtzeitbedingungen an einer realen Batteriezelle erprobt.

## 1 Einleitung

Derzeit stellen Batteriesysteme ein wesentliches Hindernis für die flächendeckende Verbreitung von Elektrofahrzeugen dar. Die im Vergleich zu fossilen Energieträgern wesentlich geringere Energiedichte führt, trotz des hohen Wirkungsgrades der Energiewandlung, zu einer erheblich geringeren Reichweite. Einen weiteren Nachteil stellen die hohen Investitionskosten in Verbindung mit der geringen Lebensdauer dar. Als Batterietyp höherer Energiedichte steigt die Bedeutung von Lithium-Ionen-Zellen (Li-Ionen-Zellen) in Kraftfahrzeugen stetig an [1], da mit ihnen diese Probleme gelöst werden sollen.

Um die benötigte Spannung bereitstellen zu können, werden mehrere Zellen als Batteriepaket in Reihenschaltung betrieben. Auch Batteriezellen derselben Baureihe variieren dabei in den maßgebenden Eigenschaften (Kapazität, Innenwiderstand, etc.), was unumgänglich zu unterschiedlichen Ladeständen im Betrieb führt. Da Li-Ionen-Zellen sowohl durch zu hohe als auch zu niedrige Spannungen beschädigt werden, können die unterschiedlichen Ladestände beim Laden und Entladen kritische Folgen haben [2].

Um einerseits kritische und Lebensdauer verkürzende Betriebszustände zu vermeiden sowie andererseits die gesamte Kapazität eines Batteriepakets auszunutzen, werden Batteriemanagementsysteme (BMS) eingesetzt [3]. In diesem Beitrag wird das Systemdesign eines solchen BMS, bestehend aus Hard- und Softwarekomponenten, für LiFePO<sub>4</sub>-Zellen dargestellt.

## 2 Methodik

Das Gesamtsystem BMS, bestehend aus Hard- und Softwarekomponenten, wird nach der durchgängig modellbasierten und verifikationsorientierten RCP Entwicklungsmethodik, basierend auf Model-in-the-Loop- (MiL), Software-in-the-Loop- (SiL) und Hardware-in-the-Loop-Simulationen (HiL), entworfen und validiert [4]. Softwareentwicklung und -test erfolgen offline an einem Batteriemodell anhand von MiL- und SiL-Simulationen. Mit steigender Softwaregüte werden HiL-Simulationen durchgeführt. Hierzu wird die Software auf der Zielhardware implementiert und unter Echtzeitbedingungen getestet, bevor die Implementierung, Validierung und Optimierung am realen System erfolgt.

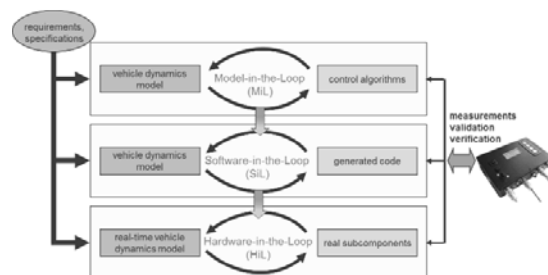


Abbildung 1. Systementwicklung mit modellbasiertem, verifikationsorientiertem Entwicklungsprozess.

Mittels dieser Methodik erfolgt ein ganzheitliches Systemdesign, wobei durch die durchgängig modellbasierte Vorgehensweise bereits in den Anfangsphasen des Entwurfsprozesses das Zusammenspiel von Hard-

und Software berücksichtigt wird. Um die Systemkomplexität zu reduzieren, erfolgt mittels Modularisierung zunächst die Zergliederung des Gesamtsystems in eine hierarchisch angeordnete Struktur (mechatronische Strukturierung). Nach Definition der Schnittstellen zwischen den Modulen erfolgt, angefangen bei der untersten Ebene, die Auslegung der einzelnen Funktionen (mechatronische Komposition) [4].

### 3 Systemkonzept des BMS

Das Batteriesystem ist ein typisches Embedded System bestehend aus den LiFePO<sub>4</sub>-Zellen, Sensoren, Aktoren und einer Informationsverarbeitungseinheit (vgl. Abbildung 2). Es nutzt Sensoren zur Messung von Spannungen, Strömen und Temperaturen. Von der Informationsverarbeitung werden diese Daten genutzt, um einen sichereren und alterungsverringenden Betrieb durch entsprechende Ansteuerung der Aktoren einzustellen.

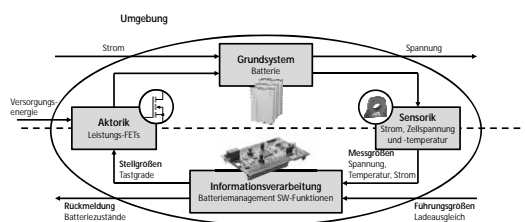


Abbildung 2. Systemkonzept des BMS als Embedded Control System [3].

Abbildung 3 zeigt die Topologie des BMS. Es besteht aus einer zentralen Control Unit, welche für die Ausführung anspruchsvoller Algorithmen wie Ladezustandsschätzung, Leistungsprädiktion oder dem Sicherheitskonzept eingesetzt wird, und dezentralen Zellmodulen auf jeder Batteriezelle. Die Zellmodule sorgen für die Messdatenerfassung, senden diese per CAN zur Control Unit und setzen den Ladeausgleich lokal um.

Das BMS kann durch seinen modularen Aufbau durch die Verwendung einer entsprechenden Anzahl Zellmodule flexibel auf Zellpakete variierender Zellanzahl skaliert werden, sodass es universell appliziert werden kann. Eine CAN-Schnittstelle ermöglicht zudem die Integration in einen Steuergeräteverbund eines Fahrzeugs.

Die Softwarestruktur des BMS lässt sich grob in vier Teilfunktionen (Module) gliedern [5]:

- Das **Ladungsmanagement** überwacht und steuert das Laden und Entladen der Zellen. Zu den Aufgaben

gehören z.B. das Freischalten des Batteriepaketes und der Ladeausgleich zwischen den Zellen.

- Im **Sicherheitsmanagement** werden gefährdende und kritische Zustände der Batterie erkannt und entsprechende Maßnahmen zur Gefahrenabwehr veranlasst.
- Das **Datenmanagement** organisiert die Speicherung, Verarbeitung und Kommunikation von Messwerten, Signalen und sonstigen Daten.
- Das Modul **Diagnose** berechnet die Systemzustände des Batteriepacks. Es kann in Zustandsschätzung (Bestimmung nicht messbarer Zustände) und -prädiktion (Vorhersage von Zuständen) unterteilt werden.

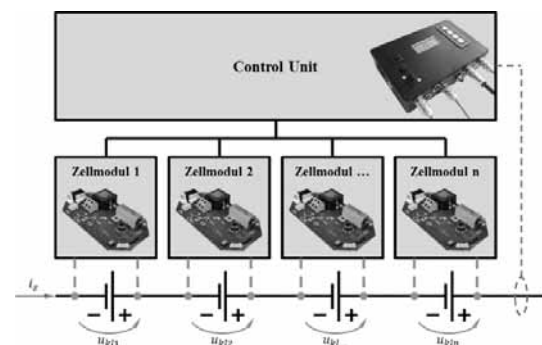


Abbildung 3. Dezentrale Topologie des BMS.

Eine Aufgabe des Softwaremoduls *Diagnose* ist die Schätzung des Ladezustands (*SoC*) im Betrieb. Aufgrund diverser nichtlinearer Einflüsse wie z.B. Temperatur und Alterung ist die einfache Integration des Stromes zur Bestimmung des *SoC* nicht ausreichend. Des Weiteren wird das System oft in einem unbekannten Zustand initialisiert, sodass der aktuelle *SoC* zunächst bestimmt werden muss. Der geschätzte *SoC* muss in diesem Fall mit ausreichender Geschwindigkeit gegen den realen Ladestand der Batterie konvergieren [3].

## 4 Systementwurf des BMS

### 4.1 Design der Softwaremodule

Zur exemplarischen Darstellung der durchgängigen Methodik wird in diesem Beitrag die Auslegung eines Extended Kalman Filter (EKF) zur Schätzung des Ladezustands von Lithium-Ionen-Zellen fokussiert.

#### 4.1.1 Modellierung der Batterie

Grundlage für die Entwicklung des EKF ist ein Modell der Batteriezelle, welches ihr stark nichtlineares Ver-

halten hinreichend genau abbildet. Die Klemmenspannung  $u_{kl}$  hängt hierbei sowohl vom Strom  $i$  als auch vom aktuellen  $SoC$ , der Temperatur  $T$  und dem State of Health ( $SoH$ ) ab. Im weiteren Vorgehen wird der Einfluss von  $T$  und  $SoH$  vernachlässigt.

Der  $SoC$  wird durch Strombilanzierung (Gl. 1) mit dem Startladestand  $SoC_0$ , dem Coulomb'schen Wirkungsgrad  $\eta_c$  und der Kapazität  $C_n$  der Batteriezelle berechnet:

$$SoC(t) = SoC_0 + \int \frac{\eta_c}{C_n} i(t) dt \quad (1)$$

Das dynamische Verhalten wird durch ein elektrisches Ersatzschaltbild (ESB, vgl. Abbildung 4) approximiert. Es besteht aus einer Spannungsquelle, einem seriellen Widerstand und vier RC-Gliedern.

Die Ruhespannung ( $OCV$ ) hängt stark nichtlinear vom  $SoC$  ab. Sie kann mithilfe mathematischer Funktionen angenähert oder als charakteristische Kennlinie in das Modell integriert werden. Um den Einfluss anderer Faktoren in der Berechnung zu berücksichtigen, wird die  $OCV$  als mehrdimensionales Kennfeld hinterlegt.

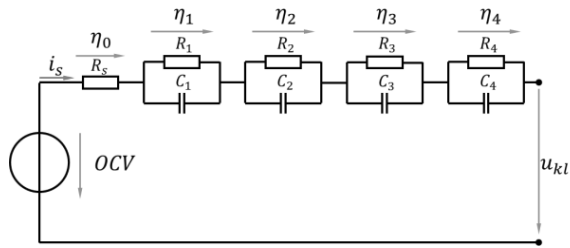


Abbildung 4. ESB der Batteriezelle.

Die Parameter für Widerstände und Kapazitäten des elektrischen Ersatzschaltbildes hängen ebenfalls nicht-linear vom  $SoC$  ab, was eine große Herausforderung bei der Identifikation darstellt. Aus dem physikalischen Ansatz des elektrischen Ersatzschaltbildes kann eine mathematische Beschreibung der Batterie hergeleitet werden, welche schließlich zu einer Übertragungsfunktion der Klemmenspannung  $u_{kl}$  bezogen auf den Strom  $i$  im Frequenzbereich führt (Gl. 2):

$$\frac{u_{kl}(s)}{i(s)} = R_s + \frac{R_1}{R_1 C_1 s + 1} + \dots + \frac{R_4}{R_4 C_4 s + 1} \quad (2)$$

Die Parameteridentifikation wird sowohl im Zeit- als auch im Frequenzbereich durchgeführt. Mittels Elektrochemischer Impedanzspektroskopie erfolgt zunächst die Bestimmung der Batterieparameter für verschiedene Ladestände im Frequenzbereich. Anschließend

wird das Modell mithilfe eines Dynamic Stress Tests (DST) im Zeitbereich validiert.

Die gemessenen (blau) und simulierten (rot) Spannungen sind oben in Abbildung 5 dargestellt und stimmen mit hoher Genauigkeit überein. Die Abweichung (unten) liegt in einem Bereich von  $\pm 2\%$ , wodurch die hohe Modellgüte bestätigt wird.

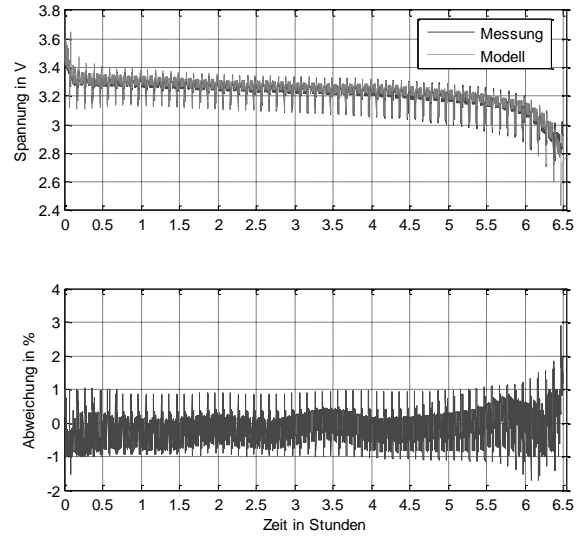


Abbildung 5. Validierung des Batteriemodells mit dem DST.

Eine detaillierte Beschreibung der Modellierung und Identifikation kann [6] entnommen werden.

#### 4.1.2 Auslegung des EKF anhand des Modells

Zur Auslegung des EKF wird das Batteriemodell in den Zustandsraum überführt. Thermische und alterungsbedingte Effekte werden zum aktuellen Stand vernachlässigt.

Der Zustandsvektor  $\underline{x}$  setzt sich aus den Überspannungen  $\eta_i$  an den RC-Gliedern und dem  $SoC$  zusammen. Die Eingangsvariable  $u$  ist der Strom  $i_s$  und die Ausgangsvariable  $y$  die Klemmenspannung  $u_{kl}$ .

$$\underline{x} = [SoC \quad \eta_1 \quad \eta_2 \quad \eta_3 \quad \eta_4]^T \quad (3)$$

$$u = i_s \quad (4)$$

$$y = u_{kl} \quad (5)$$

Die nichtlinearen Zustandsgleichungen lauten:

$$\dot{\underline{x}} = \underline{f}(\underline{x}, u) \quad (6)$$

$$y = \underline{h}(\underline{x}, u) \quad (7)$$

Die Struktur des diskreten EKF ist in Abbildung 6 dargestellt. Die Eingangsvariable  $u$  und die gemessene Variable  $y$  werden digitalisiert und dem digitalen Filter über einen Analog-Digital-Konverter (ADC) zugeführt. Durch einen Prädiktionsschritt erfolgt die Vorhersage des Zustandsvektors anhand der Eingangsgröße  $u$  und dem nichtlinearen Modell  $\underline{f}$ . Im Korrekturschritt erfolgt anhand der gemessenen Systemgrößen eine Anpassung der geschätzten Zustandsgrößen  $\underline{\hat{x}}$ .

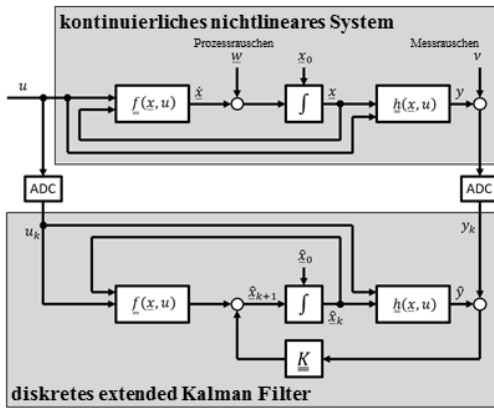


Abbildung 6. Diskretes EKF mit kontinuierlichem Systemmodell nach [7].

Der Algorithmus des EKF wird durch Abbildung 7 illustriert. Im Prädiktionsschritt wird, zusätzlich zum Zustandsvektor, anhand der Systemunsicherheit  $\underline{Q}$  die Schätzunsicherheit in Form der Fehlerkovarianzmatrix  $\underline{P}_t$  für den aktuellen Berechnungsschritt angepasst. Im Korrekturschritt erfolgt anhand der Kalman-Verstärkung  $\underline{K}$  und der Abweichung zwischen gemessenen und prädizierten Systemausgangsgrößen eine Anpassung der Zustandsgrößen sowie der Fehlerkovarianzmatrix. Die Berechnung der Kalman-Verstärkung stellt eine Minimierung des quadratischen Schätzfehlers dar. Hierzu wird die Unsicherheit der Messung in Form der Messrauschkovarianzmatrix  $\underline{R}$  benötigt.

Im Vergleich zum linearen Kalmanfilter arbeitet das EKF nicht mit den Systemmatrizen  $\underline{A}$  und  $\underline{C}$ , sondern mit den Jacobimatrizen der Funktionen  $\underline{f}$  und  $\underline{h}$ :

$$\underline{J}_A = \frac{\partial \underline{f}(x, u)}{\partial \underline{x}} \quad (8)$$

$$\underline{J}_C = \frac{\partial \underline{h}(x, u)}{\partial \underline{x}} \quad (9)$$

Für eine ausführliche Darstellung der Herleitung sei auf [8] oder [9] verwiesen.

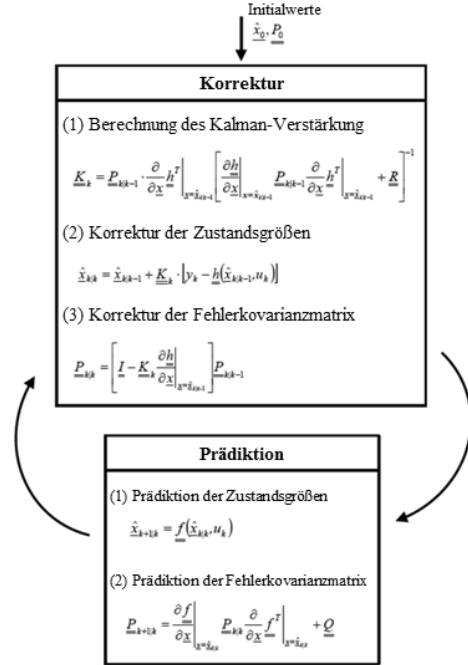


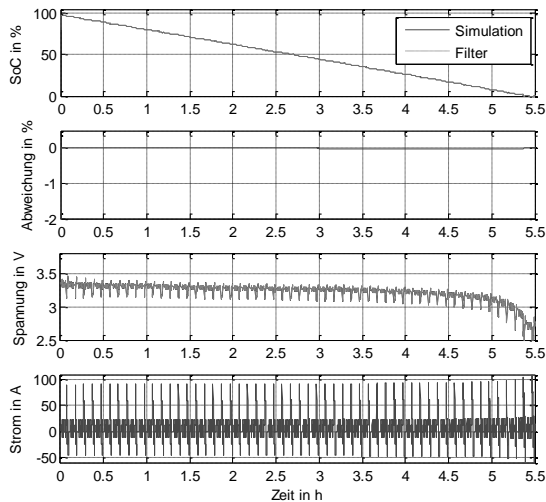
Abbildung 7. Algorithmus des EKF [7].

Über die Systemunsicherheit sowie das Messrauschen lässt sich das Verhalten des Kalmanfilters beeinflussen. Die Bestimmung der Messrauschkovarianzmatrix erfolgt, indem die Abweichung zwischen Messung und einem bekannten Verlauf der Messgröße in ein Histogramm eingetragen werden. Schlussendlich wird die Systemunsicherheit bestimmt, indem sie nach [10] als Gewichtungsmatrix einen Kompromiss zwischen Stabilität und Konvergenzgeschwindigkeit des EKF ermöglicht.

#### 4.1.3 Erprobung in der Offlinesimulation

Das anhand der theoretischen Zusammenhänge ausgelegte Filter wird durch eine Model-in-the-Loop-Simulation erprobt und validiert. Hierbei wird die Batterie mit dem DST Zyklus belastet. Das Filter ist mit einem SoC von 80% als Startwert parametrisiert, während das Batteriemodell mit einem SoC von 100% startet. Abbildung 8 illustriert das Ergebnis dieser Offlinesimulation. Oben ist in blau der durch das Batteriemodell simulierte und in rot der durch das Filter geschätzte SoC über der Zeit aufgetragen. In der zweiten Zeile wird die Abweichung zwischen simuliertem und geschätztem SoC, in Zeile drei die simulierte und die geschätzte Klemmenspannung sowie in der letzten Zeile der Strom dargestellt.

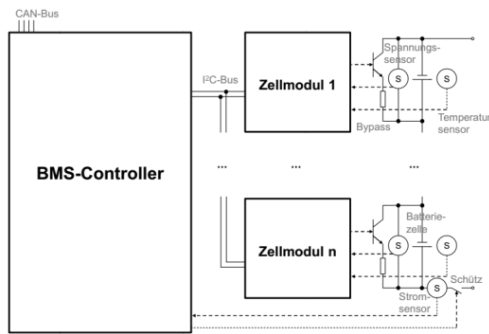
Innerhalb von 0,5 Sekunden konvergiert das Filter auf den *SoC* des Batteriemodells. Im weiteren Simulationsverlauf treten nur geringe Fehler auf.



**Abbildung 8.** Validierung des EKF mittels MiL-Simulation.

## 4.2 Design der Hardware

Das Hardwarekonzept des BMS stellt eine dezentrale Struktur, bestehend aus einem zentralen BMS-Controller und einer variablen Anzahl an Zellmodulen (vgl. Abbildung 9), dar. Diese modulare Struktur ermöglicht die Skalierung des Systems auf eine nahezu beliebige Anzahl an Batteriezellen. Diese ist lediglich durch die Auslastung der CAN-Bus Verbindung, welche die Zellmodule mit der zentralen Controllereinheit verbindet, begrenzt.



**Abbildung 9.** Hardwarekonzept des BMS [3].

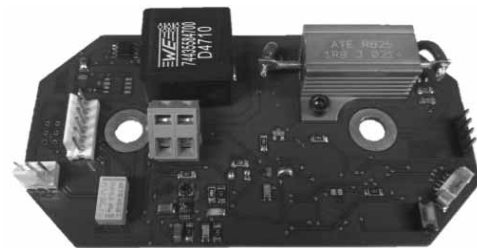
### 4.2.1 Zentraler BMS Controller

Der zentrale BMS-Controller stellt den Kern des Systems dar. Auf ihm sind die hochwertigen Softwarefunktionen und komplexen Algorithmen implementiert. Beim BMS-Controller handelt es sich um einen Mikrocontroller des Typs STM32, einer Signalverarbeitung und einer Benutzerschnittstelle.

Als einfache Benutzerschnittstelle sind ein Display und vier Taster vorgesehen, welche eine grundlegende Bedienung erlauben. Des Weiteren gibt es USB-Schnittstellen für die Kommunikation mit einem PC und die Programmierung des Controllers, zwei CAN-Schnittstellen für die Kommunikation mit anderen Steuergeräten sowie mit den Zellmodulen und drei Anschlüsse für Stromsensoren.

### 4.2.2 Dezentrale Zellmodule

Die dezentralen Zellmodule (vgl. Abbildung 10) dienen der Messwerterfassung sowie der Befehlsumsetzung. Sie messen die Klemmenspannung sowie die Temperatur der Zelle und senden diese Informationen an den zentralen BMS-Controller. Des Weiteren wird das Balancing der Zelle durch das Zellmodul lokal umgesetzt, sodass es einen Teil der Aktorik des Systems darstellt. Mit den Zellmodulen ist sowohl aktives als auch passives Balancing möglich, da entsprechende elektronische Bauelemente vorhanden sind.



**Abbildung 10.** Realisierung des Zellmoduls.

## 5 Erprobung des Gesamtsystems

Nach erfolgreicher Optimierung sämtlicher Komponenten in der Offlinesimulation bzw. in Einzelkomponententests erfolgt die Validierung des Gesamtsystems, bestehend aus den Hard- und Softwarekomponenten, mittels Onlinesimulation. In diesem Beitrag wird exemplarisch die Validierung des EKF mittels HiL-Simulation vorgestellt. Hierbei ist die Funktion der Ladezustandsschätzung auf dem zentralen BMS-Controller implementiert, welcher mittels Zellmodul an eine reale Batteriezelle angebunden ist. Der Systemtest erfolgt mit demselben DST Zyklus wie die Offlinesimulation.

In Abbildung 11 oben ist in blau der aufgrund des vom Prüfstand eingeregelter Stromes berechnete *SoC* und in rot der durch das EKF geschätzte *SoC* dargestellt. In der Mitte ist durch die Abweichung zwischen berechnetem und geschätztem *SoC* die Qualität des Filters illustriert. Im unteren Teil der Grafik ist die gemessene der geschätzten Klemmspannung gegenübergestellt.

Wie bereits in der MiL-Simulation festgestellt, konvergiert das Filter innerhalb kürzester Zeit gegen den berechneten *SoC*. Die Abweichungen im weiteren Verlauf sind minimal.

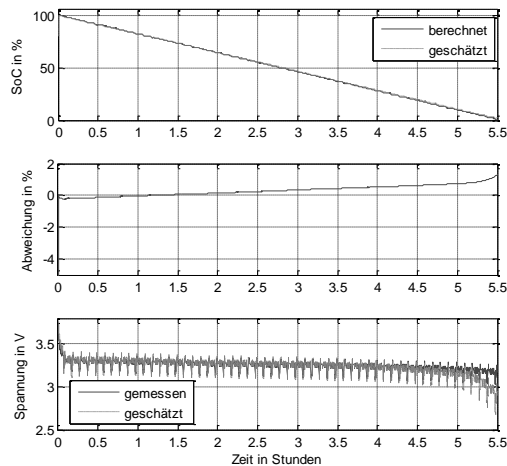


Abbildung 11. HiL-Simulation des EKF.

## 6 Zusammenfassung und Ausblick

Im vorliegenden Beitrag wurde das ganzheitliche Systemdesign eines flexibel skalierbaren Batteriemanagementsystems für Elektrofahrzeuge dargestellt. Mittels durchgängig verifikationsorientierter, modellgestützter Entwicklungsmethodik erfolgte die Auslegung der Hard- und Softwarekomponenten. Exemplarisch für den Entwurf der Softwarefunktionen wurde ein Ladezustandsschätzer ausgelegt und mittels MiL-, SiL- und HiL-Simulationen validiert. Die Hardwarekomponenten des BMS bestehen aus einem zentralen Controller und dezentralen Zellmodulen, was eine einfache Skalierung auf eine fast beliebige Anzahl an Batteriezellen ermöglicht.

In weiteren Arbeiten erfolgt die Auslegung und Verifikation weiterer BMS-Funktionen wie dem Balancing. Des Weiteren ist die Integration des BMS in den Forschungsträger FreDy geplant.

## 7 Danksagung

Der vorliegende Beitrag wurde im Rahmen des Forschungsprojektes *LoCoRCP* vom EFRE-Fonds der EU unter dem Zeichen ZW 6-85003460 gefördert. Die Verantwortung für den Inhalt liegt bei den Autoren, welche sich für die Förderung herzlich bedanken.



## 8 Referenzen

- [1] Neumeister, D., Wiebelt, A. u. Heckenberger, T.: *Systemeinbindung einer Lithium-Ionen-Batterie in Hybrid- und Elektroautos*. ATZ - Automobiltechnische Zeitschrift 112 (2010) 4, S. 250–255
- [2] Jossen, A. u. Weydanz, W.: *Moderne Akkumulatoren richtig einsetzen*. Ubooks 2006
- [3] Quantmeyer, F., Roch, M., Diehl, W. u. Liu-Henke, X.: *Ein skalierbares Echtzeitsystem zur Erprobung des Batteriemanagements in Elektrofahrzeugen*. Tagungsband ASIM-Workshop STS/GMMS 2014. 2014, S. 161–166
- [4] Liu-Henke, X.: *Mechatronische Entwicklung der aktiven Feder-Neigetechnik für das Schienenfahrzeug RailCab*. Fortschritt-Berichte VDI Reihe 12, Bd. 589. VDI-Verlag 2005
- [5] Fleischer, C. et al.: *Development of software and strategies for Battery Management System testing on HIL simulator*. Eleventh International Conference on Ecological Vehicles and Renewable Energies (EVER). 2016, S. 1–12
- [6] Quantmeyer, F. u. Liu-Henke, X.: Modeling the Electrical Behavior of Lithium-Ion Batteries for Electric Vehicles. Solid State Phenomena 214 (2014), S. 40–47
- [7] Quantmeyer, F. u. Liu-Henke, X.: *State of Charge Estimation for Lithium-ion batteries in Electric Vehicles using Extended Kalman Filtering*. 1st International Symposium on Energy Challenges & Mechanics (ECM). Aberdeen, UK 2014
- [8] Krebs, V.: *Nichtlineare Filterung. Methoden der Regelungstechnik*. Oldenbourg 1980
- [9] Grewal, M. S.: *Kalman Filtering*. In: Lovric, M. (Hrsg.): International Encyclopedia of Statistical Science. Springer Berlin Heidelberg 2011, S. 705–708
- [10] Föllinger, O. u. Dörrscheidt, F.: *Regelungstechnik. Einführung in die Methoden und ihre Anwendung*. Hüthig 2008

# Prädiktives Energiemanagement für Elektrofahrzeuge

Matthias Fritsch<sup>1</sup>, Sören Scherler<sup>1</sup>, Xiaobo Liu-Henke<sup>1</sup>

<sup>1</sup>Ostfalia Hochschule für angewandte Wissenschaften, Fakultät Maschinenbau,  
Institut für Mechatronik, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel  
matthias.fritsch@ostfalia.de

Dieser Beitrag fokussiert den modellbasierten, ganzheitlichen Entwurf eines energie- und zeitoptimalen Geschwindigkeitsprofils sowie dessen Integration in ein prädiktives Energiemanagement für Elektrofahrzeuge. Beginnend bei der mathematischen Beschreibung der Fahrzeuglängsdynamik als Grundlage für die nachfolgende Optimierung werden die notwendigen Teilfunktionen zur Berechnung einer optimierten Geschwindigkeitstrajektorie vorgestellt. Hierbei wird ein neuartiger Ansatz zur Diskretisierung des Lösungsraums des betrachteten Lösungsraums hergeleitet. Nachgelagert ermöglicht die berechnete Geschwindigkeitstrajektorie die Prädiktion zukünftiger Leistungsbedarfe seitens des Energiemanagements, wodurch beispielsweise Bordnetzverbraucher wirkungsgradoptimiert geschaltet werden können.

## 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.

Eine weitere Entwicklung zeichnet sich durch die zunehmende Intelligenz von Fahrerassistenzsystemen (ADAS, Advanced Driver Assistance System) infolge der höheren Verfügbarkeit von Daten innerhalb des Fahrzeugs und des Streckenverlaufs aus. Bei Berücksichtigung dieser zusätzlichen Informationen kann der Energieverbrauch des Fahrzeugs durch Anpassen der Fahrzeuggeschwindigkeit in Abhängigkeit aktueller Fahrzeugzustände und der vorausliegenden Strecke reduziert werden.

Deshalb liegt ein Forschungsschwerpunkt der Verfasser auf einem intelligenten, elektronischen Fahr-

zeugmanagement. Im vorliegenden Beitrag wird eine Teilfunktion dieses Fahrzeugmanagements, das prädiktive Energiemanagement (pEEM), vorgestellt. Als eine der Kernfunktionen des pEEM liegt der Fokus auf einem Algorithmus zum Berechnen eines energieoptimalen Geschwindigkeitsprofils für ein reines Elektrofahrzeug mit zwei radnahen Traktionsantrieben.

## 2 Stand der Technik

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 2012 entwickelte die Volkswagen AG die Fahrstrategie „Green Driving“ [2]. 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. Die Funktionsweise beschränkt sich jedoch auf Situationen in denen das Fahrzeug verzögert werden muss, beispielsweise vor einer Ortsdurchfahrt oder Geschwindigkeitsbegrenzungen. Fahrzeugspezifische Zustandsgrößen werden

vernachlässigt, welche insbesondere bei Elektrofahrzeugen einer Berücksichtigung bedürfen.

[3] entwickelt ein energieoptimales Geschwindigkeitsprofil mit äquidistanter Diskretisierung. Hieraus resultieren Vorteile hinsichtlich der benötigten Rechenleistung, da die Algorithmen lediglich für bestimmte Zustandsänderungen formuliert werden müssen. Die Diskretisierung wird in bestimmten Stellen des betrachteten Horizonts kleiner gewählt, beispielsweise in Kurven. Eine weitere Realisierung kann unter [4] nachgelesen werden.

Aufgrund der angesprochenen begrenzten Reichweite elektrisch angetriebener Fahrzeuge ist die bedarfsgerechte Aufteilung der gespeicherten chemischen Energie sehr wichtig. Insbesondere heuristische Betriebsstrategien werden in der Praxis häufig verwendet. Diese lassen sich mit geringem Aufwand auslegen, garantieren jedoch nicht zwingen einen optimalen Betrieb. Prädiktive Ansätze hingegen verwenden Informationen über die vorausliegende Strecke, um beispielsweise ein optimiertes Leistungsprofil zu berechnen [5].

Bisherige Arbeiten erstellten Geschwindigkeitsprofile ausschließlich für konventionelle, verbrennungsmotorisch angetriebene Fahrzeuge. Der Unterschied zu reinen Elektrofahrzeugen ist jedoch sehr groß, weshalb eine separate Betrachtung und Auslegung solcher Funktionen erfolgen muss. Ziel dieses Beitrags ist es deshalb, ein prädiktives Energiemanagement zu entwickeln, welches die Geschwindigkeit des Fahrzeugs in Abhängigkeit des Verbrauchs und der Fahrzeit optimiert und hierauf aufbauend die Leistungsflüsse im Fahrzeug koordiniert. Hierbei müssen sowohl die Akteure als auch die Straße berücksichtigt werden.

### 3 Entwicklungsmethodik

Bei der Entwicklung hochkomplexer Systeme ist es unabdingbar, eine systematische und klar strukturierte Entwicklungsmethode zu verwenden. Insbesondere bei der stark steigenden Anzahl der im Fahrzeug integrierten ADAS.

Die Entwicklungsmethode basiert auf einer Modularisierung und Hierarchisierung, wodurch große, komplexe Systeme in kleinere intelligente Untersysteme gegliedert werden können. Die einzelnen Systeme beinhalten mechatronische Komponenten und können über vorab definierte Schnittstellen mit der Umwelt kommunizieren. Ziel ist es, eine klare hierarchische

Struktur einzelner, in sich gekapselter Funktionen zu erreichen.

Angewandt auf die ganzheitliche Entwicklung eines prädiktiven, elektronischen Fahrzeugmanagement ergibt sich eine hierarchische Struktur nach

Abbildung 1. Fahrzeuge mit prädiktivem Fahrzeugmanagement können durch Car-2-Car- oder Car-2-Infrastructure-Kommunikation miteinander und ihrer Umwelt interagieren. Das Fahrzeugmanagement eines Fahrzeugs lässt sich in ein elektronisches Fahrwerkmanagement mit unterlagerten Fahrdynamikregelsystemen wie ABS oder ESP, Assistenzfunktionen und das pEEM unterteilen.

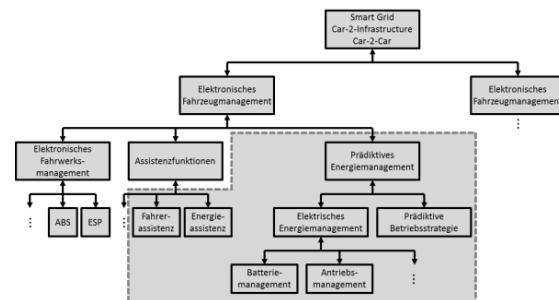


Abbildung 1. Hierarchische Struktur des elektronischen Fahrzeugmanagements

Im pEEM wiederum kann zwischen prädiktiver Betriebsstrategie und elektrischem Energiemanagement unterschieden werden. Die prädiktive Betriebsstrategie liefert Daten über die vorausliegende Fahrt, welche vom pEEM verarbeitet und in Sollgrößen für das Bordnetz umgesetzt werden. Diese Sollgrößen werden vom EEM und unterlagerten Funktionen eingeregt.

Die Entwicklung, Validierung und Optimierungen der einzelnen Funktionen erfolgt gekapselt, bevor sie mithilfe definierter Schnittstellen gekoppelt werden.

### 4 Modellbildung

Die Entwicklung eines Optimierungsalgorithmus setzt eine geeignete mathematische Beschreibungsform des Systems voraus.

#### 4.1 Fahrzeugmodell

Das vorgestellte ADAS zielt auf die Verbesserung des Energieverbrauchs der Fahrzeuglängsdynamik ab. Dem Newton-Euler-Ansatz folgend kann diese wie folgt beschrieben werden:

$$\frac{dv}{dt} = \frac{1}{m} \cdot F_M - \frac{1}{m} \cdot F_R(v)$$

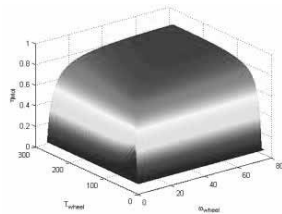
Mit der Fahrzeugmasse  $m$ , welche die zusätzlichen rotatorischen Trägheiten beinhaltet, den Antriebskräften  $F_M$  sowie den Fahrwiderständen  $F_R$ , welche von der aktuellen Geschwindigkeit abhängen. Dementsprechend hängt die Änderung der Fahrzeuggeschwindigkeit von den aufgetragenen Antriebskräften sowie den Fahrwiderständen ab. Hierbei werden der Roll-, Steigungs- und Luftwiderstand und somit die Fahrwiderstände mit dem größten Einfluss auf die Fahrdynamik berücksichtigt [6; 7]. Allerdings können die tatsächlich auftretenden Kräfte aufgrund von Modellungenauigkeiten oder Vernachlässigungen (z.B. Gegenwind, variierende Fahrbahnoberfläche) anders ausfallen. Aus diesem Grund ist eine zusätzliche Regelung notwendig, um auftretende Abweichungen zwischen der berechneten und realen Geschwindigkeit auszugleichen (vgl. Kapitel 5.4).

## 4.2 Antriebsstrang und Betriebsarten

### 4.2.1 Elektromotor

Der Hauptunterschied, verglichen zu einem konventionellen Fahrzeug, ist der Einsatz eines Elektromotors. Diese können, anders als Verbrennungsmotoren, sowohl motorisch als auch generatorisch betrieben werden. Aus diesem Grund kann dieser auf mehreren Wegen die Längsdynamik des Fahrzeugs beeinflussen. Nähere Informationen über die Modellbildung sind [7; 8] zu entnehmen.

Um den Betriebsbereich des Synchronmotors an den angestrebten Geschwindigkeitsbereich des Fahrzeugs anzupassen, wird ein Getriebe mit einer Übersetzung  $i = 7$  verwendet. Zusammen mit dem Getriebewirkungsgrad ergibt sich das folgende Wirkungsgradkennfeld für den Antriebsstrang:



**Abbildung 2.** Wirkungsgradkennfeld eines Elektromotors nach [8]

Die Abbildung zeigt den gesamten Wirkungsgrad von Batterie bis zum Rad, aufgetragen über der aktuellen Winkelgeschwindigkeit und Drehmoment des Rei-

fens. Je höhere Drehmoment und Drehzahl sind, desto höher ist der resultierende Gesamtwirkungsgrad. Insbesondere geringe Drehmomente zum Halten der Geschwindigkeit weisen einen geringen Wirkungsgrad auf und führen somit zu unverhältnismäßig hohen Leistungsbedarfen.

Hierauf aufbauend lassen sich die Leistungsbedarfe des Aktors für die verschiedenen Betriebsfälle berechnen:

- Motorischer Leistungsbedarf:

$$P_{Bat} = \frac{U \cdot I_{Mot}}{\eta}$$

- Generatorischer Leistungsbedarf:

$$P_{Bat} = U \cdot I_{Mot} \cdot \eta$$

Anhand der berechneten Leistung und der Zeitdauer kann auf den Energiebedarf der einzelnen Betriebsarten geschlossen werden.

$$W = \int_0^t P_{Bat} dt$$

### 4.2.2 Reibbremse

Das Vorsehen einer Reibbremse in einem Elektrofahrzeug garantiert ein sicheres und schnelles Abbremsen des Fahrzeugs, falls die Verzögerung durch Rekuperation nicht ausreicht. Hierbei wird eine verzögernde Kraft durch Zusammenpressen der Reibfläche und der Bremsscheibe durch einen hydraulisch generierten Druck erzeugt. Aus deren Fläche sowie dem angelegten Bremsdruck ergibt sich die Normalkraft, mit der die Reibfläche auf die Bremsscheibe gedrückt wird.

$$F_{Reib} = p \cdot A_{Brems} \cdot \mu_{Reib}$$

Aus dem Übersetzungsverhältnis zwischen effektivem Durchmesser der Bremsscheibe sowie dem statischen Reifenhalbmesser, kann die Wirkung der Bremse auf das Fahrzeug beschrieben werden.

### 4.2.3 Segeln

Hierbei wird keiner der vorhandenen Aktoren angesteuert. Lediglich die Fahrwiderstände und vergleichsweise geringe Reibung innerhalb der Radaufhängung bewirken in diesem Fall eine verzögernd wirkende Kraft.

Je nach Fahrbahnbeschaffenheit kann der Zustand segeln jedoch auch eine positive Geschwindigkeitsänderung als Folge haben.

#### 4.2.4 Traktionsbatterie

Da die Batteriespannung sehr stark vom aktuellen SoC abhängt, wird ein einfaches Batteriemodell in den Optimierungsalgorithmus integriert. Beginnend bei einem Startwert wird der SoC über die Höhe der benötigten Energie zum Erreichen der gewünschten Geschwindigkeitsänderung berechnet.

$$SoC = SoC_{start} - \int \frac{i_{Bat} \cdot \Delta t}{C_{Bat}} dx$$

Anschließend wird die Leerlaufspannung der Batterie über einen Look-up-Table berechnet.

$$U_{OCV} = f(SoC)$$

#### 4.3 Beschränkungen

Die betrachtete Fahrzeugkonfiguration ermöglicht eine hohe Anzahl verschiedener Aktorkombinationen. Allerdings können bestimmte Kombinationen wegen derer Wirkung sowie energetischer Aspekte ausgeschlossen werden.

Bei einer Beschleunigung kann beispielsweise lediglich der Elektromotor verwendet werden. Ist hingegen eine Verzögerung erforderlich, sollte die erste Wahl auf den generatorischen Einsatz des Elektromotors fallen, insofern die Batterie nicht vollgeladen ist. Wenn höhere Drehmomente erforderlich sind, wird die Bremse hinzugeschaltet.

Zuletzt könnte es sein, dass keine zusätzliche Fahrzeuglängskraft erforderlich ist, um die gewünschte Geschwindigkeitsänderung zu erzielen. Dann reichen die Fahrwiderstände aus. Gegenstrombremsen wird, wie bereits erwähnt, nicht berücksichtigt, da es energetisch nicht sinnvoll ist.

### 5 Energieoptimales Geschwindigkeitsprofil

Der Entwicklungsmethodik entsprechend wurde die Gesamtfunktionalität in kleinere, gekapselte Systeme unterteilt. Abbildung 3 illustriert den gewählten Aufbau.

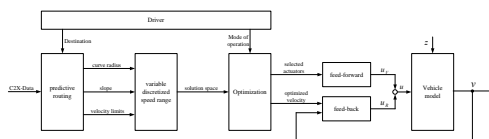


Abbildung 3. Gesamtübersicht des Geschwindigkeitsprofils

Insgesamt besteht das System aus vier gekapselten Funktionen, der prädiktiven Streckenplanung, dem variabel diskretisiertem Geschwindigkeitsschlauch, der Optimierung sowie einer nachgelagerten Regelung.

#### 5.1 Streckenplanung

Als Basis der Streckenplanung wird ein Navigationssystem angestrebt, welchem der Fahrer das Fahrtziel mitteilen kann. Anschließend wird die optimale Route unter Berücksichtigung der aktuellen Verkehrslage berechnet. Die gewählte Route wird im Anschluss über einen CAN-Bus an das zu entwickelnde Steuergerät gesendet. Dem ADASISv2-Protokoll entsprechend erfolgt die Übermittlung der Daten (Wegpunkt, Höhe, Geschwindigkeitsbegrenzungen etc.) abschnittsweise. Dies bedeutet, dass die Daten, beginnend beim Startpunkt bis zu einem gewissen Horizont im Voraus, einmalig übermittelt werden. Die nachgelagerte Auswertung kann beispielsweise durch die Nutzung eines Horizon Reconstructors unterstützt werden [9]. Durch die geplante Verwendung dieses Protokolls wird eine einheitliche Schnittstelle mit anderen Navigationsgeräten, die dieses Protokoll unterstützen, sichergestellt.

Erweiterungen hinsichtlich der aufkommenden Vernetzung durch Car-to-Infrastructure sind erweiterte Geschwindigkeitsbegrenzungen auf Basis von Wetterdaten oder Informationen hinsichtlich der Schaltzeiten von auf der Strecke befindlichen Lichtsignalanlagen [10].

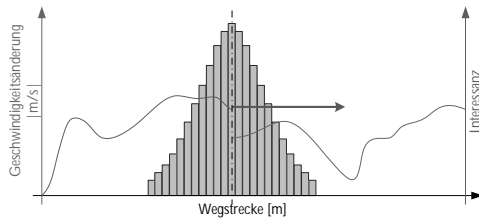
#### 5.2 Variabel diskretisierter Geschwindigkeitsschlauch

Im zweiten Schritt werden die übermittelten Informationen über die Streckenbeschaffenheit verwendet, um eine fundierte Diskretisierung der Strecke durchzuführen. Hierzu werden zunächst Größen definiert, welche einen Einfluss auf die zu wählende Diskretisierung haben könnten. Bei der Betrachtung einer Fahrt über eine Landstraße, bei der über längere Zeit eine hohe Geschwindigkeit erlaubt ist, kann zwar Energie gespart werden, indem eine niedrigere Geschwindigkeit gewählt wird, allerdings ist es nicht notwendig in einem solchen Bereich eine feine Diskretisierung zu wählen. Wohingegen bei der Betrachtung eines Hügels eine feine Diskretisierung sinnvoll ist, um die einzelnen Segmente der Steigung und Beschaffenheit korrekt abzubilden. Wird die Diskretisierung hier zu grob gewählt, könnte es sogar sein, dass der Hügel für die Optimierung nicht existiert.

Gleiches gilt für Geschwindigkeitsänderung in Folge von Begrenzungsschildern. Dementsprechend werden in diesem Beitrag die Änderung der zulässigen Geschwindigkeit sowie die Steigung der Strecke als Referenz verwendet, um die Diskretisierung festzulegen. Hierzu wird ein Hamming-Fenster

$$w(n) = 0,54 - 0,46 \cdot \cos\left(\frac{2\pi n}{M-1}\right)$$

mit einer Breite von  $M = 180 \text{ m}$  definiert, womit die einzelnen Streckenabschnitte bewertet werden (Abbildung 4). Die Laufvariable  $n$  wird beginnend bei 0 in Zehnerschritten hochgezählt. Demnach werden insgesamt 19 Wegstützstellen zur Bewertung herangezogen.



**Abbildung 4.** Faltung der Straßencharakteristiken zum Berechnen der Interessanz

Somit haben sowohl zurückliegende als auch vorausliegende Streckeneigenschaften einen Einfluss auf die Bewertung. Der berechnete Wert der Faltungsfunktion spiegelt sich in der *Interessanz* der einzelnen Wegpunkte wider. Die anschließende Einteilung der gewählten Diskretisierung erfolgt nach Tabelle 1:

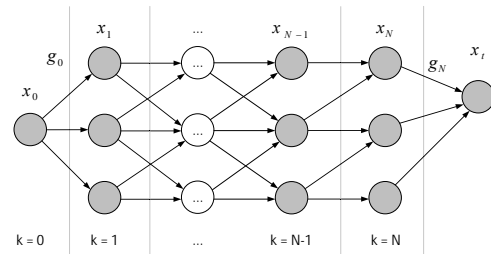
Interessanz	Skalierung $dx$ [m]	Skalierung $dv$ [km/h]
0 - 20	200	1,00
20 - 40	100	0,50
40 - 60	50	0,20
60 - 80	20	0,10
80 - 100	10	0,05

**Tabelle 1.** Zuordnung der Diskretisierung der Strecke

Hierdurch ist ein sehr breites Spektrum an verschiedenen Diskretisierungen möglich, wodurch der Rechenaufwand ohne größere Einbußen hinsichtlich der Genauigkeit gesenkt werden kann.

### 5.3 Numerisches Lösungsverfahren

Unter der Voraussetzung, dass es sich bei dem betrachteten System um eine Markov-Kette erster Ordnung handelt, können Zustandsänderungen unabhängig von der Vergangenheit mit jeder Betriebsstrategie erfolgen. Dies impliziert, dass die Zukunft des Systems nur von der Gegenwart und nicht von der Vergangenheit abhängt. Dementsprechend können alle Kombinationsmöglichkeiten der verfügbaren Betriebsarten über den betrachteten Horizont miteinander verglichen werden. Abbildung 5 illustriert beispielhaft die möglichen Zustände innerhalb einer vorgesehenen Wegstrecke.



**Abbildung 5.** Zustandsgraph eines deterministischen, endlichen Systems

Den einzelnen Zustandsübergängen werden Kosten zugeordnet, wodurch eine nachgelagerte Gegenüberstellung der insgesamt anfallenden Kosten möglich ist. Die Kosten ergeben sich aus zwei konkurrierenden Zielfunktionen – der Zeit und Energie. Ein Verbessern der einen hat demnach ein Verschlechtern der anderen Funktion zur Folge.

Die energetischen Kosten ergeben sich aus der zeitlichen Integration der Aktorleistung

$$k_{Energy} = \int_{t_0}^{t_1} P_{Act} dt,$$

wohingegen für die zeitlichen Kosten lediglich die Zeitdauer herangezogen wird.

$$k_{Time} = \Delta t$$

Als Gewichtung wird eine Stunde Fahrzeit gleichgesetzt mit zwei Stunden Laden bei einer Ladeleistung von  $P = 3000 \text{ W}$ . Zusätzlich wird ein Faktor eingesetzt, der die energetischen Aspekte gewichten kann.

$$k_{Total} = \frac{k_{Energy}}{6000 \text{ J}} \cdot k_{Mod} + \frac{k_{Time}}{1 \text{ s}}$$

Je kleiner  $k_{Mod}$  gewählt wird, desto geringer wird der Energieverbrauch gewichtet. Andersherum repräsen-

tiert ein hohes  $k_{Mod}$  eine stark energetisch optimierte Fahrweise.

Das Auffinden der optimalen Lösung des vorgestellten Problems fällt somit in das Fachgebiet der Kombinatorik. Die einfachste Lösung resultiert aus einer kompletten Enumeration der möglichen Zustände entlang der betrachteten Strecke. Das Berechnen aller möglichen Kombinationsmöglichkeiten führt jedoch bei linearer Zunahme des Zustandsraums zu einem exponentiellen Anstieg der möglichen Kombinationen.

Aus diesem Grund wird die dynamische Programmierung zur Reduzierung des Berechnungsaufwands eingesetzt [11]. Dem Bellmanschen Optimalitätsprinzip entsprechend setzt sich die optimale Lösung des Gesamtproblems aus den optimalen Teillösungen kleinerer Teilprobleme zusammen. Dies ermöglicht es, die optimalen Kosten zwischen jeder Wegstützstelle separat zu berechnen und trotzdem zu gewährleisten, dass die optimale Lösung gefunden wird. Die Realisierung dieser Funktion erfolgt durch drei ineinander verschachtelte Schleifen (Abbildung 6).

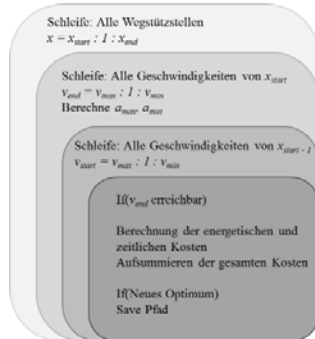


Abbildung 6. PAP des Geschwindigkeitsprofils

Die äußerste Schleife läuft über alle vorhandenen Wegstützstellen, beginnend bei  $x_{start}$ . Anschließend wird eine Schleife über alle Geschwindigkeitsstützstellen von  $x_{start}$  durchlaufen. Hierbei werden zunächst die möglichen positiven sowie negativen Beschleunigungen berechnet. Im Anschluss wird eine Schleife über alle Geschwindigkeitsstützstellen der vorherigen Stützstelle  $x_{start-1}$  durchlaufen. Sollte die betrachtete Geschwindigkeitsdifferenz erreichbar sein, werden die zeitlichen und energetischen Kosten für diesen Zustandswechsel berechnet. Addiert zu den zuvor kumulierten Kosten, wird überprüft, ob ein neues Optimum vorliegt und dieses zusammen mit der vorherigen Stützstelle gegebenenfalls gespeichert.

Durch das Abspeichern der vorherigen Stützstellen, die zum derzeit optimalen Ergebnis führen, wird eine direkte Rückverfolgung der optimalen Route mitsamt der notwendigen Stellgrößen erreicht. Eine anschließende Rekursionsrechnung ist somit nicht notwendig, vgl. [3].

## 5.4 Regelung

Nachgelagert erfolgt die Ansteuerung der Aktorik, bestehend aus zwei Teilfunktionen. Eine feed-forward Regelung, welche die berechneten Sollwerte an die Aktoren weitergibt sowie einem feed-back Regler, welcher überwacht, ob die korrekte Geschwindigkeit eingeregelt wird.

Diese Struktur ermöglicht es, zum einen die berechneten Stellgrößen für die Aktorik direkt als Vorsteuerung an die Steuergeräte zu leiten und zum anderen Modellungenauigkeiten oder nicht berücksichtigte Einflüsse auf die Längsdynamik durch den feed-back Regler zu kompensieren.

## 5.5 Simulationsergebnisse

Da die Berechnung des Geschwindigkeitsprofils sehr rechenintensiv ist und neue Informationen über den zukünftigen Streckenverlauf erst nach dem Passieren einer gewissen Wegstrecke vorliegen, wird das Geschwindigkeitsprofil nicht in jedem Simulationsschritt ausgeführt, sondern lediglich einmal pro Sekunde aufgerufen. Beispielhafte Simulationsergebnisse stellt Abbildung 7 vor.

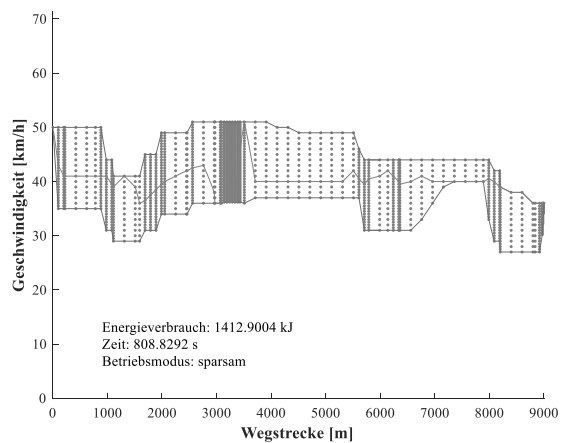


Abbildung 7. Energie- und zeitoptimales Geschwindigkeitsprofil bei Berücksichtigung ohmscher Verluste

Die obere und untere Linie repräsentieren die Geschwindigkeitsbeschränkungen und spannen den

zwischen ihnen liegenden Lösungsraum auf. Grau gepunktet werden die diskretisierten Weg- respektive Geschwindigkeitsstützstellen und in dunkelgrau das berechnete Geschwindigkeitsprofil dargestellt. Den vorherigen Beschreibungen entsprechend liegen die Wegstützstellen bei großen Geschwindigkeitsänderungen (u.a. bei 1500-2000 m) sowie Steigungen (u.a. bei 3000-3500 m) sehr eng beieinander. Bei dieser Simulation wurde eine energiesparende Betriebsart gewählt, weshalb eine insgesamt relativ niedrige Geschwindigkeit gewählt wird. Stärkere Geschwindigkeitsänderungen treten lediglich in der Nähe von Gefällen/Steigungen auf. Bei starken Gefällen (3000-3500 m) wird zunächst eine konstante Geschwindigkeit durch den Einsatz von Rekuperation eingehalten. Kurz vor Ende des Bereichs wird das Gefälle zum kurzzeitigen Erhöhen der Geschwindigkeit und somit zum Reduzieren der zeitlichen Kosten verwendet. Anschließend erfolgt wiederum der Übergang zu einer konstanten Fahrt.

## 6 Prädiktives Energiemanagement

Unter Verwendung des energie- und zeitoptimalen Geschwindigkeitsprofils kann eine Abschätzung des zukünftigen Energiebedarfs der Fahrzeuglängsführung erfolgen.

### 6.1 Elektrisches Energiemanagement

Ein elektrisches Energiemanagement sorgt generell für eine bedarfsgerechte und prioritätsbasierte Verteilung der im Elektrofahrzeug begrenzt verfügbaren Leistung auf die elektrischen Verbraucher bei gleichzeitiger Gewährleistung von Fahrsicherheit und möglichst nicht wahrnehmbaren Einbußen im Fahrkomfort. Weiterhin stellt es die Schnittstelle zu unterlagerten Systemen wie Batterie- oder Antriebsmanagement dar, verarbeitet deren Bedarfe und gibt Sollwerte vor (vgl. [12]). Bisherige Ansätze basieren auf aktuellen Bordnetz- und Fahrzeugzuständen ohne zukünftige Ereignisse zu berücksichtigen.

### 6.2 Erweiterung zum prädiktiven EEM

Die prädiktive Betriebsstrategie ergänzt das EEM, um einen optimierten Leistungsfluss im Fahrzeug unter Berücksichtigung der Wirkungsgrade von Batterie, Antrieben und Bordnetzverbrauchern zu erzielen

Hierzu werden zunächst die benötigten elektrische Leistungen für die Routenführung unter Verwendung eines inversen Antriebsstrangmodells prädiziert (Abbildung 8, rote Markierung). Dieser Leistungsver-

lauf bildet die Grundlage für eine Prädiktion des Ladestandverlaufs und die Planung der Leistungsflüsse aller anderen Komponenten während der Fahrt, um das Ziel zu erreichen und ggf. einen definierten Ziel-ladestand (insb. bei Einsatz eines Range Extenders, vgl. [13]) zu erreichen.

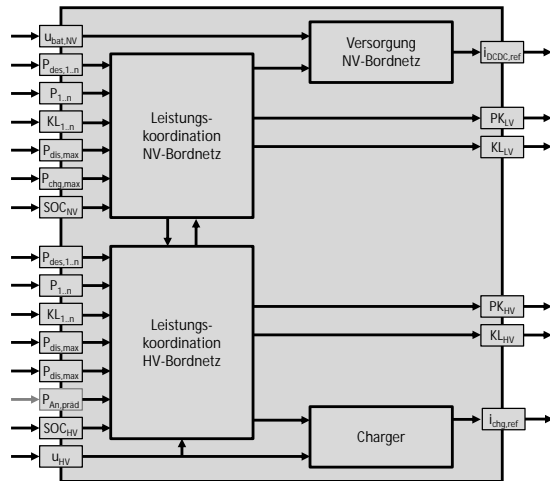


Abbildung 8. Struktur des prädiktiven EEM

So kann es bspw. sinnvoll sein, Rekuperationsphasen zur Versorgung von Bordnetzverbrauchern zu nutzen ohne die Rekuperationsleistung verlustbehaftet in die Batterie zwischenspeichern. Unter Kenntnis der Zeiträume von Rekuperationsphasen kann der Einsatz leistungsstarker Bordnetzverbraucher wie Klimatisierung frühzeitig auf die zusätzliche Leistung abgestimmt werden.

### 6.3 Simulationsergebnisse

In Abbildung 9 ist das Ergebnis einer Simulation zur Validierung des Antriebsleistungsverlaufs dargestellt. Im oberen Teil sind die exakt übereinstimmenden Geschwindigkeitsprofile aus Prädiktion und Simulation dargestellt.

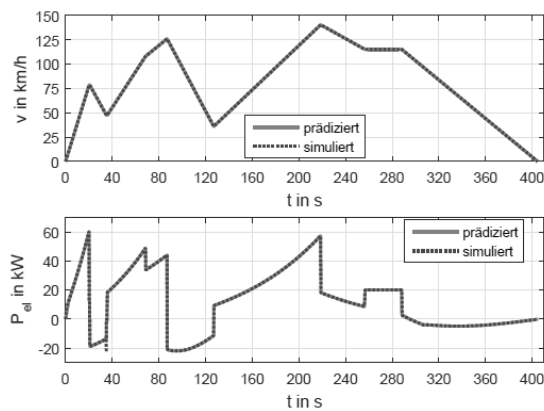


Abbildung 9. Ergebnis der Leistungsprädiktion

Unten ist in blau der Verlauf der Antriebsleistung bei Simulation des Fahrzeugs mit dem oben dargestellten Geschwindigkeitsprofil dargestellt. Dieser stimmt mit dem Verlauf des Prädiktionsalgorithmus überein, sodass er als validiert betrachtet wird. Aufbauend auf diesem Verlauf ist die zuvor beschriebene Koordination der Bordnetzkomponenten möglich.

## 7 Resüme und Ausblick

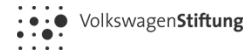
Im Rahmen dieses Beitrags wurden der systematische Entwurf eines energie- und zeitoptimalen Geschwindigkeitsprofils sowie die Erweiterung eines Energiemanagements um die generierten prädiktiven Streckendaten vorgestellt. Hierbei konnte gezeigt werden, dass eine mathematisch fundierte Diskretisierung der betrachteten Strecke den Rechenaufwand verringern kann, ohne die Optimierung zu verschlechtern. Darüber hinaus konnten die berechneten Geschwindigkeitswerte vom prädiktiven Energiemanagement verwendet werden, um einen optimierten Leistungsfluss im Fahrzeug zu erzielen.

Zukünftige Arbeiten beschäftigen sich neben einer tiefergehenden Integration von C2X-Daten, wie den genauen Schaltzeiten von Lichtsignalanlagen mit der Prädiktion der Leistungsbedarfe der Bordnetzverbraucher (Klimaanlage, Sitzheizung, etc.) sowie deren zeitlichen Optimieren.

## 8 Danksagung

Dieser Beitrag wurde im Rahmen des Projekts *Zukünftige Fahrzeugtechnologien im Open Region Lab* durch das MWK Niedersachsen und die VW-Stiftung unter dem Förderkennzeichen VWZN3236 gefördert.

Die Verantwortung für den Inhalt liegt bei den Autoren.



## 9 Literatur

- [1] Markschräger, P.; Wahl, H.-G.; Weberbauer, F.; Lederer, M.: Assistenzsystem für mehr Kraftstoffeffizienz. ATZ Automobiltech Z 114 (2012) Nr. 11, S. 850–855
- [2] Dornieden, B.; Junge, L.; Pascheka, P.: Vorausschauende Energieeffiziente Fahrzeuglängsregelung. ATZ Automobiltech Z 114 (2012) Nr. 3, S. 230–235
- [3] Gausemeier, S. F.: Ein Fahrerassistenzsystem zur prädiktiven Planung energie- und zeitoptimaler Geschwindigkeitsprofile mittels Mehrzieloptimierung, Dissertation, Paderborn 2013
- [4] Radke, T.: Energieoptimale Längsführung von Kraftfahrzeugen durch Einsatz vorausschauender Fahrstrategien. Karlsruher Schriftenreihe Fahrzeugsystemtechnik. 19. KIT Scientific Publishing, Karlsruhe 2013
- [5] Styler, A.; Nourbakhsh, I.; Sauer, A.; Rottengruber, H.: Learned Optimal Control of a Range Extender in a Series Hybrid Vehicle
- [6] Fritsch, M.; Liu-Henke, X.: Eine intelligente Fahrerassistenzfunktion zum energieoptimalen Fahrbetrieb für Elektrofahrzeuge: ASIM/GI STS/GMMS Workshop 2016, Lippstadt
- [7] Fritsch, M.; Liu-Henke, X.: Optimization of energy consumption by using an intelligent assistance system for an electric vehicle: IEEE 2017 – IEEE International Conference on Ecological Vehicles
- [8] Borchardt, N.: Modellierung, elektromechanische Auslegung und Validierung eines Radnabenmotors mit nutenloser Luftspaltwicklung und hoher gravimetrischer Leistungsdichte, Dissertation, Magdeburg 2014
- [9] dSPACE: Entwicklungen unter dem elektronischen Horizont. Eine durchgängige Entwicklungsumgebung für kartenbasierte Fahrerassistenzsysteme 2010
- [10] Themann, P.; Zlocki, A.; Eckstein, L.: Energieeffiziente Fahrzeuglängsführung durch V2X-Kommunikation. In: Siebenpfeiffer, W. (Hrsg.): Fahrerassistenzsysteme und Effiziente Antriebe. Springer Fachmedien Wiesbaden, Wiesbaden 2015, S. 63–67

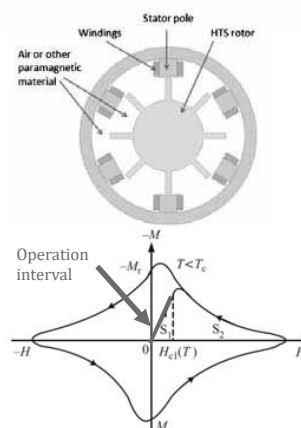
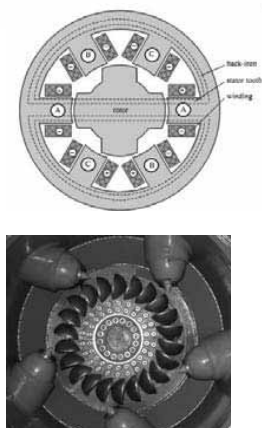
- [11] *Bellman, R.*: The theory of dynamic programming, Santa Monica 1954
- [12] *Scherler, S.; Quantmeyer, F.; Liu-Henke, X.*: Modellbasierte Entwicklung des elektrischen Energiemanagements für Elektrofahrzeuge mit einem Zweispannungsbordnetz:  
ASIM/GI STS/GMMS Workshop 2015
- [13] *Scherler, S.; Liu-Henke, X.*: Leistungsverteilung in Elektrofahrzeugen mit Range Extender:  
ASIM/GI STS/GMMS Workshop 2017

# Development and Design Optimization of a High-temperature Superconducting Quasi-Diamagnetic Motor Demonstration Unit

Marcell Baranyai – Dénes Kapi – István Vajda – Walter Commerell

Óbuda University / Hochschule Ulm

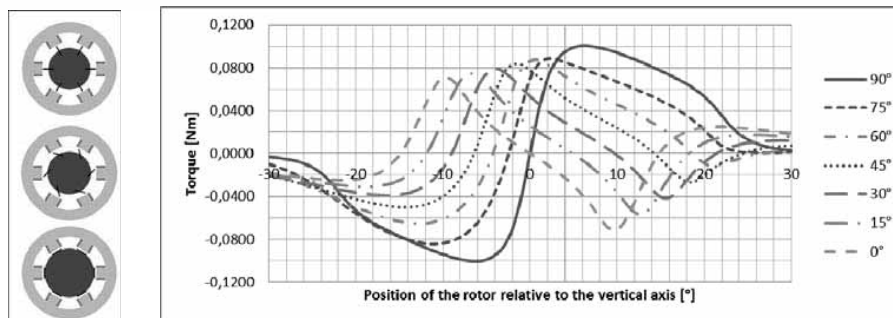
## PRELIMINARIES: ON OPERATING PRINCIPLE



- Similarities with switched reluctance motors (SRM) and Pelton-turbine
- Operation interval on the first magnetization curve of type-II superconductive materials

2

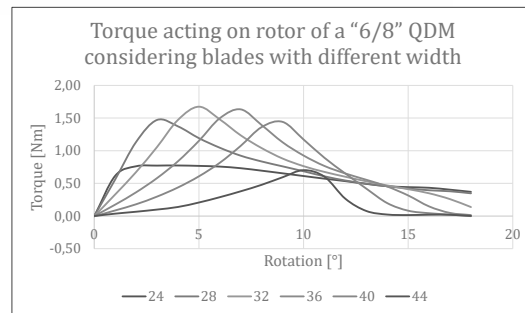
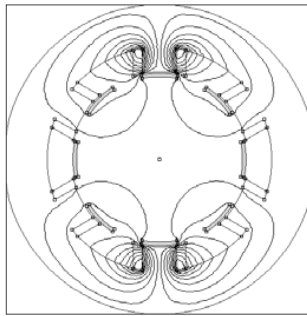
## PRELIMINARIES: PARAMETER SWEEP I. – SLANTED ROTOR BLADES (SENSITIVITY ANALYSIS)



- The angle of the rotor blades influenced the torque exerted on the rotor. "Tangential" blades gave best results (40% greater torque compared to "radial" blades)
- The sign of torque is reversed. (Attractive force instead of repulsive.)

3

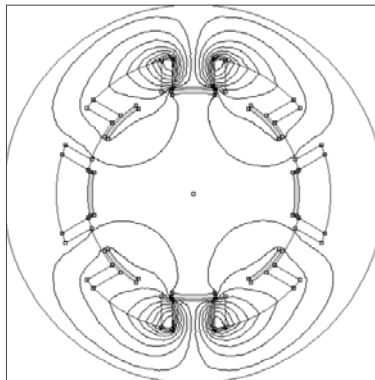
## PRELIMINARIES: PARAMETER SWEEP II. – WIDTH OF ROTOR BLADES (SENSITIVITY ANALYSIS)



- Torque production is also very sensitive to width of the blade attached to the surface of the rotor.
- There is an ideal range for the ratio of the width of pole to the width of blade.

4

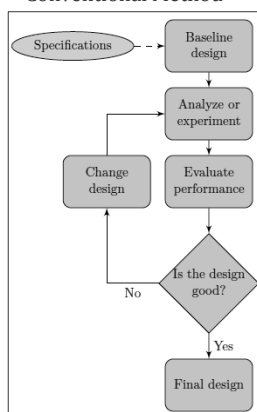
## PRELIMINARIES: FIRST EXPERIMENTAL SETUP



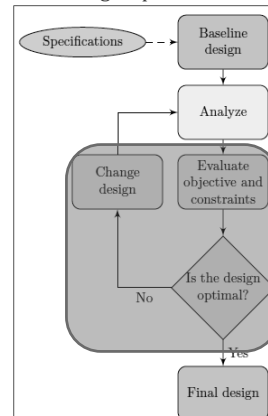
5

## CONVENTIONAL DESIGN PROCESS VS. DESIGN OPTIMIZATION

### Conventional Method



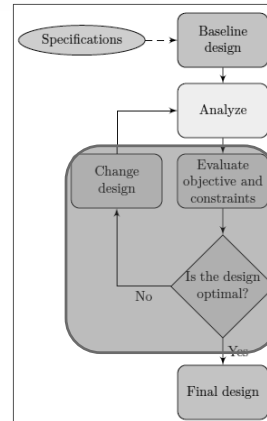
### Design Optimization



6

## DESIGN OPTIMIZATION: METHODOLOGY

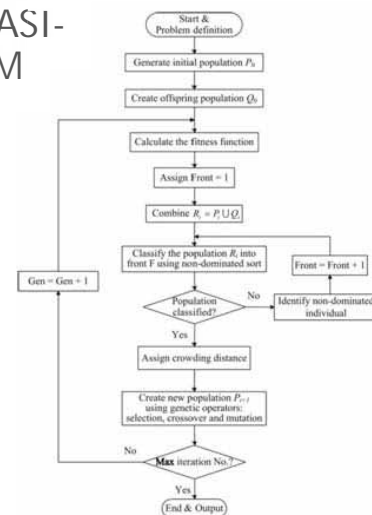
- Systematic approach of decision making is needed
- Need to decide on:
  1. Optimization algorithm
  2. Definition of variables
  3. Feasibility of model (geometrical constraints)
  4. Constraint functions
  5. Objective functions
  6. Stopping criterion



7

## DESIGN OPTIMIZATION OF THE QUASI-DIAMAGNETIC MOTOR: ALGORITHM

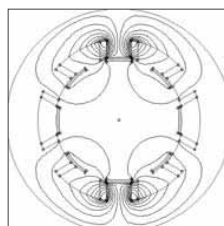
- Wide variety of optimization techniques can be used for motor design
- Complex design → explicit optimization methods cannot be used
- Mostly metaheuristic techniques used, such as evolutionary algorithms
- For the optimization of QDM, I used the NSGA-II algorithm, that is readily available in MATLAB (gamultiobj function)



8

## DESIGN OPTIMIZATION OF THE QUASI-DIAMAGNETIC MOTOR: DEFINITION OF VARIABLES

- Design variables are usually defined in two ways
  - directly (e.g. stator inner radius, number of stator poles, air gap etc.)
  - as ratios of model parameters (e.g. ratio of slot width to the slot depth)



Parameter and unit	Symbol
Thickness of superconductor blade [mm]	a
Width of superconductor blade [mm]	b
Position of the rotor blade measured from the vertical axis [°]	$\alpha$
Stator outer radius [mm]	$R_{so}$
Stator inner radius [mm]	$R_{si}$
Stator pole inner radius [mm]	$R_{pi}$
Width of stator pole [mm]	$w_p$
Width of stator pole with winding [mm]	$w_w$
Axial length of the machine [mm]	l
Peak current in stator winding [A]	$I_{pk}$
Number of stator poles	$N_{sp}$
Number of rotor blades	$N_{rb}$
Turns of stator winding	$N_{sw}$
Air gap [mm]	$\delta$

Model parameters of the QDM

9

## DESIGN OPTIMIZATION OF THE QUASI-DIAMAGNETIC MOTOR: OBJECTIVE FUNCTIONS, CONSTRAINTS

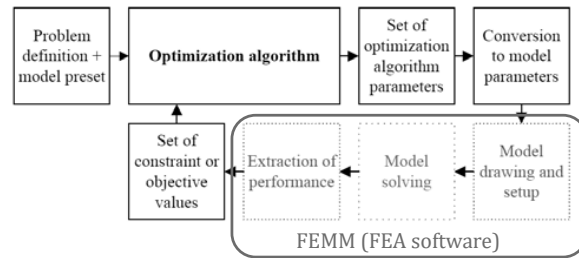
Minimize	$-M(a, b, \alpha, w_p)$
	$V_{sc}(a, b)$
Subject to	$a_{min} \leq a < R_{pi} - \delta$
	$b_{min} < b < \frac{2 \cdot (R_{pi} - \delta) \cdot \pi}{N_{rb}}$
	$\alpha_{min} < \alpha < \frac{360^\circ}{2 \cdot N_{sp}}$
	$w_{wmin} < w_w < 2 \cdot R_{pi} \cdot \sin\left(\frac{360^\circ}{2 \cdot N_{sp}}\right)$

Parameter and unit	Symbol
Thickness of superconductor blade [mm]	a
Width of superconductor blade [mm]	b
Position of the rotor blade measured from the vertical axis [°]	$\alpha$
Stator outer radius [mm]	$R_{so}$
Stator inner radius [mm]	$R_{si}$
Stator pole inner radius [mm]	$R_{pi}$
Width of stator pole [mm]	$w_p$
Width of stator pole with winding [mm]	$w_w$
Axial length of the machine [mm]	l
Peak current in stator winding [A]	$I_{pk}$
Number of stator poles	$N_{sp}$
Number of rotor blades	$N_{rb}$
Turns of stator winding	$N_{sw}$
Air gap [mm]	$\delta$

10

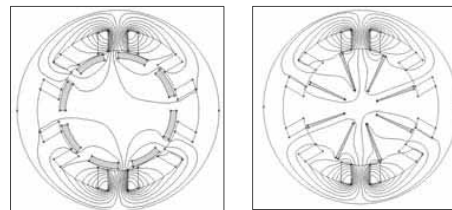
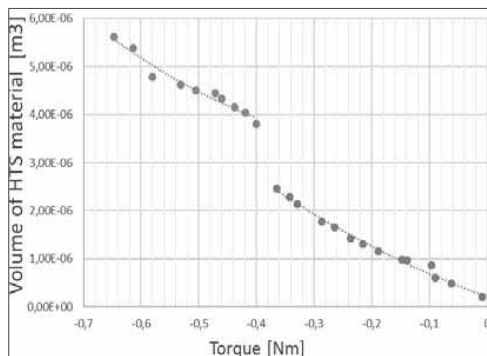
## DESIGN OPTIMIZATION OF THE QUASI-DIAMAGNETIC MOTOR: WORKFLOW OF OPTIMIZATION

1. Problem definition: setup of boundaries, objectives, constant model parameters (number of poles, winding etc.)
2. The NSGA-II algorithm in MATLAB generates variables/parameters
3. The model is drawn/setup in FEMM (by means of scripts in MATLAB)
4. Model is solved (in FEMM)
5. The performance of the model is extracted (post-processing in FEMM)
6. Values of objective functions are passed back to MATLAB.
7. Steps 2-6 performed iteratively, optimal solution is obtained after termination.



11

## DESIGN OPTIMIZATION OF THE QUASI-DIAMAGNETIC MOTOR: RESULTS



Two kinds of optimal designs:

- a) Wide blades along the circumference → better torque production
- b) Radial blades of the early design → less superconductive material

12

## DESIGN OPTIMIZATION OF THE QUASI-DIAMAGNETIC MOTOR: NEXT STEPS

- New inner stator – outer rotor design concept, cooling of superconducting wire made easier (the baseline design is ready)
- Design optimization process to be applied on new basic design
- New experimental setup to be built based on results
- Measurements to be carried out to determine operating characteristics



13

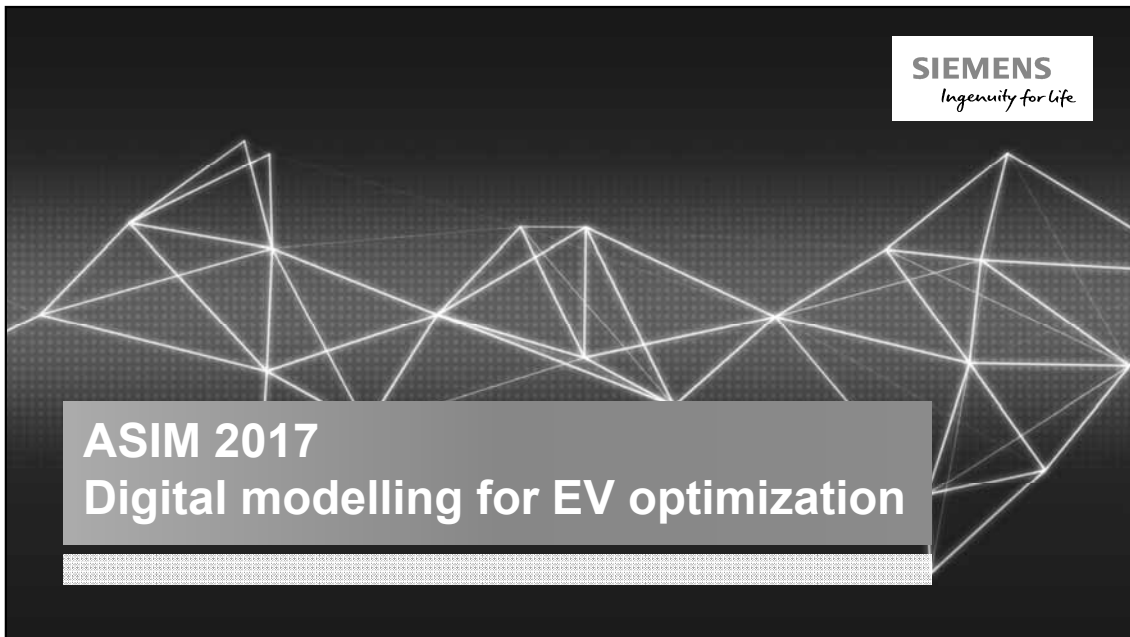
## REFERENCES

- Deb, Kalyanmoy (2001): Multi-objective optimization using evolutionary algorithms. 1st ed. New York: John Wiley & Sons (Wiley-Interscience series in systems and optimization).
- Lei, Gang; Zhu, Jianguo; Guo, Youguang (2016): Multidisciplinary design optimization methods for electrical machines and drive systems. Berlin: Springer (Power systems).
- Rebahi, Fares et al. (2016): Optimization design of a doubly salient 8/6 SRM based on three computational intelligence methods. In *Turk J Elec Eng & Comp Sci* 24, pp. 4454–4464. DOI: 10.3906/elk-1503-142.
- Stipetic, Stjepan; Miebach, Werner; Zarko, Damir (2015): Optimization in design of electric machines: Methodology and workflow. In H. Bülent Ertan (Ed.): *ACEMP-OPTIM-ELECTROMOTION 2015* Joint Conference, pp. 441–448.
- Vajda, Istvan; Semperger, Sandor; Baranyai, Marcell (2015): Analysing the characteristics of specific torque in HTS quasi-diamagnetic motor by variation of rotor blade geometry. In : 2015 9th International Conference on Compatibility and Power Electronics (CPE), pp. 230–233.

THANK YOU FOR  
YOUR  
ATTENTION!

Any questions?

14



## Digital modelling for EV optimization IMPROVE FP7 research project 2013-2017

SIEMENS

**Project goal**  
Research and develop new EV electric and electronic architecture for higher vehicle efficiency and range (>20% vs baseline), up to full scale demonstration (rolling demonstrator)

**Vehicle baseline:** Fiat Doblo EV

**Partners:** Continental, TOFAŞ TÖK OTOMOBİL FABRİKASI A.Ş., Jeep, SIEMENS, Applus, SIDA, SICI, BRUSA, virtual vehicle, Fraunhofer

**Siemens motivation:**

- End to end model-based vehicle architecture and sizing support
- Model-based EV optimal operations investigation: models integration to controls, route planning and eco-driving
- Access to experimental data for validating the approach.

Restricted © Siemens AG 2016  
Page 2 XX.XX.20XX

Continental, TOFAŞ TÖK OTOMOBİL FABRİKASI A.Ş., Jeep, SIEMENS, Applus, SIDA, SICI, BRUSA, virtual vehicle, Fraunhofer

Siemens PLM Software

## Digital modelling for EV optimization Vehicle Optimization

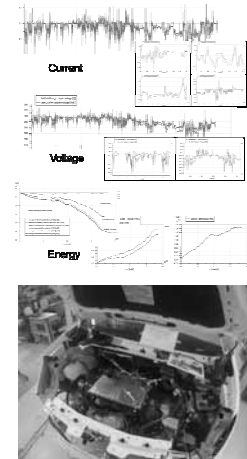
SIEMENS

A baseline vehicle model have been set up and validated vs experimental data, including real drive cycles.

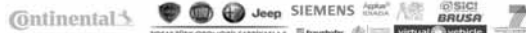
Good correlation have been achieved, with identified gaps due to identified uncertainties (wind, brake blending strategy).

The baseline vehicle model have then be used to support new components preliminary integration and validation: emotor (separated excitation), inverter, charger, batteries...

Manoeuvres virtual validation supported powertrain right sizing.



Restricted © Siemens AG 2016  
Page 3 XX.XX.20XX



Siemens PLM Software

## Digital modelling for EV optimization Advanced model-embedding controls

SIEMENS

Starting from the final vehicle configuration, Siemens supported EV optimal controls through:

- Providing models for Model in the loop controls development
- Providing exported models (S-functions) for **cloud-based range prediction** and high level energy management strategies
- Providing reduced models for **eco-driving and traction energy management** (integrated in a Simulink toochain/Woodward Motoshawk, low end target).
- Providing high complexity model reductions (Response surfaces networks) for **optimized thermal management** (integrated in a Simulink toochain/Woodward Motoshawk, low end target).



Restricted © Siemens AG 2016  
Page 4 XX.XX.20XX



Siemens PLM Software

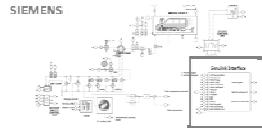
## Digital modelling for EV optimization

### Models for online range prediction, energy management and eco-driving

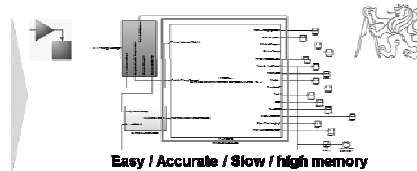
SIEMENS

Model for cloud-based range prediction & optimization

Direct simulink model export (BB)



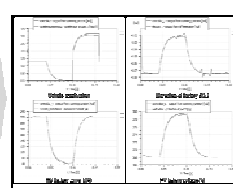
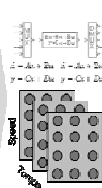
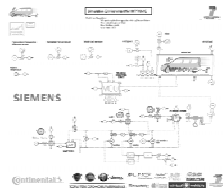
Grid of linear equivalent model



Easy / Accurate / Slow / high memory



Reduced model for on board eco-driving & energy management



Complex / accuracy is grid depend / Fast / Low memory

Restricted © Siemens AG 2016  
Page 5 XX.XX.20XX

Continental

Jeep

SIEMENS

Applus

IDIADA

BAUSA

virtual vehicle

Siemens PLM Software

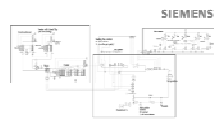
## Digital modelling for EV optimization

### Optimal thermal management support (Model Predictive Control)

SIEMENS

Model reduction (RSM networks) for thermal management controller inc complex heat pump for powertrain and cabin (low end ECU target)

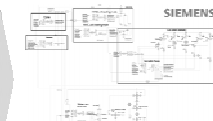
Model reduction methodology



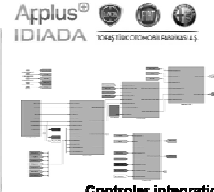
Model decomposition



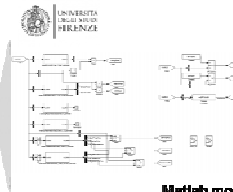
Reduced model generation



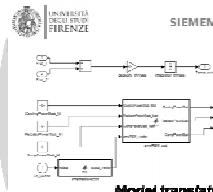
Reduced model assembly



Controller integration



Matlab model



Model translation

Restricted © Siemens AG 2016  
Page 6 XX.XX.20XX

Continental

Jeep

SIEMENS

Applus

IDIADA

BAUSA

virtual vehicle

Siemens PLM Software

## Digital modelling for EV optimization Current Status

SIEMENS

**New rolling demonstrator fully integrated**

**Tested and tuned (ESP, Brake blending) at IDIADA, Spain**

**Full performance testing in progress**



Restricted © Siemens AG 2016  
Page 7 XX.XX.20XX

Continental

Jeep

SIEMENS

Apple

BAUSA

virtual vehicle

Siemens PLM Software

## Digital modelling for ev optimization Conclusions

SIEMENS

Findings on the best export and physical models reduction methods for control purposes.

Demonstration of models integration benefits to multiple levels of IT systems and controls.

Demonstration of mandatory model-based methods for design and controls of complex EV thermal systems (mutualized heat pumps for thermal management and comfort functions).

Demonstration of the need of multi-disciplinary simulation to address complex combination of powertrain dynamics with auxiliaries impacting performances at first order and their controls.

Restricted © Siemens AG 2016  
Page 8 XX.XX.20XX

Continental

Jeep

SIEMENS

Apple

BAUSA

virtual vehicle

Siemens PLM Software

# 2Simulate - Werkzeug zur Online-Trimmung für die Simulation dynamischer Systeme

Jürgen Gotschlich

Deutsches Zentrum für Luft- und Raumfahrt (DLR)

Institut für Flugsystemtechnik

*juergen.gotschlich@dlr.de*

Bei der Simulation dynamischer Systeme ist es häufig notwendig Anfangsbedingungen herzustellen, die einem bestimmten Systemzustand entsprechen, um das System ausgetrimmt, also beschleunigungsfrei starten zu können. Im „Air VEHICLE Simulator“ (AVES) des DLR-Instituts für Flugsystemtechnik in Braunschweig wird das Werkzeug „2Simulate“ als Simulationsinfrastruktur benutzt. Damit wird sowohl die Modellintegration, die Prozessierung und Verteilung aller Signalflüsse als auch die Steuerung/Bedienung des Simulators durchgeführt. Der vorliegende Beitrag erläutert die Implementierung eines generischen Online-Trimmvorgangs mit dem beliebige dynamische Systeme getrimmt werden können in das „2Simulate“ Simulationswerkzeug.

## 1 Einleitung

In der Realität hat der Begriff Trimmung speziell für Luft- und Wasserfahrzeuge eine besondere Bedeutung. Durch geeignete Maßnahmen werden Kräfte und Momente so ausgeglichen, dass die Lage der Vehikel stationär, also unbeschleunigt ist. Bei der statischen Trimmung wird der Schwerpunkt des Körpers durch Verschieben oder Verändern von Masse variiert (z.B. Treibstoff, Ballast). Bei der dynamischen Trimmung werden zusätzliche Trimmkräfte durch die Verstellung von aero- bzw. hydrodynamischen Flächen erzeugt (z.B. Trimmruder, Trimmklappen) [8], [10].

Bei der Simulation dieser Systeme wird die Bedeutung des Begriffs Trimmung ausgeweitet. Hier ist es notwendig unterschiedliche Szenarien bereitzustellen, um gezielt Untersuchungen in einem konkreten Kontext durchführen zu können. Es sollen also unterschiedliche stationäre Betriebspunkte (Trimmpunkte) hergestellt werden, die in der englischen Literatur auch als Steady State oder Equilibrium Point bezeichnet werden. Die dafür notwendigen Anfangsbedingungen des dynamischen Modells sind neben den realen Trimmmöglichkeiten der Vehikel, wie im vorangegangenen Absatz beschrieben, weiterhin abhängig von dem Kontext des gesamten Szenarios (Masse, Position, Lage, Geschwindigkeiten, spez. Eigenschaften...).

Die Bereitstellung verschiedener Anfangsbedingungen einer Simulation kann durch unterschiedliche Methoden erfolgen. Im einfachsten Fall werden die Anfangswerte der Zustands- und Eingangsgrößen im Modell direkt auf die vorab berechneten Trimmwerte

gesetzt. Es werden mehrere parallele Varianten des Modells jeweils mit einem unterschiedlichen Trimpunkt parat gehalten und entsprechend gestartet. Jeder Modellwechsel kann jedoch in komplexen Simulatoren mit entsprechend komplexen Abhängigkeiten sehr aufwändig und lästig sein. Besser ist das Ablegen der unterschiedlichen Trimpunkte in Form von vorab berechneten Datensätzen, die in das Modell dann zur Laufzeit beliebig geladen und aktiviert werden können. Die Nachteile beider Methoden sind jedoch die geringe Flexibilität und der große Zeitaufwand sogar bei kleinsten Änderungen (Turnaround-Zyklus) sowie im Falle der vorab erstellten Trimpunktdatensätze die mögliche Diskrepanz der Modelle zum Zeitpunkt der Erstellung und zum Zeitpunkt des Aktivierens der Trimmzustände (Versionsdiskrepanz).

Ein weiterer Nachteil speziell bei der Echtzeitsimulation ist das zusätzliche Bereithalten einer zum Trimmen geeigneten Version des Modells (parallel zu dem echtzeitfähigen Modell), um damit neue, bzw. geänderte Trimpunkte zu berechnen. Auch hierbei ist als mögliche Fehlerquelle eine Versionsdiskrepanz zu bedenken. Wenn der Trimpunktdatensatz zu einem bestimmten Zeitpunkt erstellt wird und später mit der echtzeitfähigen Version des Modells gearbeitet wird, kann nur mit einem sehr aufwändigen konsequenten Qualitätsmanagementprozess verhindert werden, dass sich die Modelle unterscheiden und zuvor abgelegte Trimpunktdatensätze möglicherweise ungültig sind.

Eine wesentliche Optimierung bzgl. Flexibilität, Zeitaufwand und Qualität ist die hier vorgestellte Möglichkeit einer Online-Trimmung innerhalb einer Echtzeitsimulation unter Nutzung des echtzeitfähigen

Modells. Nur hiermit kann wirklich sichergestellt werden, dass der berechnete Systemzustand eines Trimpunktes mit dem verwendeten Modell zusammenpasst.

Das Trimmproblem lässt sich grundsätzlich für beliebige dynamische Systeme verallgemeinern, sofern für die Simulation des Systems ein passendes Modell zur Verfügung steht.

Einige Beispiele sind:

- schwingungsfähige Systeme (Feder-Masse-Systeme)
- gefederte Fahrwerke
- Hydraulik-/Pneumatiksysteme (z.B. elektrohydraulische Aktuatoren)
- Luftfahrzeuge (z.B. Flächenflugzeuge, Drehflügler, Schlagflügler)
- Wasserfahrzeuge (z.B. Schiff, Hovercraft)
- Wärmekraftmaschinen (z.B. Verbrennungsmotoren)

## 2 AVES-Simulationsinfrastruktur

Das Deutsche Zentrum für Luft- und Raumfahrt (DLR), Institut für Flugsystemtechnik betreibt in Braunschweig den Air VEHICLE Simulator (AVES). AVES ist als modulare, flexible Plattform ausgelegt für den Forschungseinsatz, speziell im Hinblick auf die dynamische Interaktion zwischen Mensch und Luftfahrzeug, insbesondere für Verkehrsflugzeuge und Hubschrauber sowie für neue Konfigurationen. AVES besteht aus einem elektromechanisch angetriebenen Bewegungssimulator und einem Festsitzsimulator. Zurzeit stehen sowohl ein Airbus-A320-Cockpit in Level-D-Qualität als auch ein Eurocopter-EC135-Cockpit zur Verfügung. Ein Roll-On, Roll-Off-Wechselsystem erlaubt einen Cockpitwechsel vom Festsitz- in den Bewegungssimulator und umgekehrt [1].



Abbildung 1: AVES Simulator

Für die vielfältigen Forschungsaufgaben, die mit AVES erfüllt werden sollen, ist ein äußerst flexibles Softwarekonzept notwendig. Dies teilt sich auf in Softwaremanagement und Softwaredesign. Für das Softwaremanagement ist das AVES Software Development Kit entwickelt worden [2]. Zum einen muss eine einwandfreie Funktionalität des Simulators von geprüften und jederzeit abrufbaren Softwareversionen sicher gestellt sein. Gleichzeitig muss jedoch die Anforderung erfüllt sein, jederzeit Zugriff auf alle Software zu haben, um sowohl kurzfristig auf aktuelle Problemstellungen in Form von Testsoftware reagieren zu können, als auch langfristig Komponenten und Techniken für neue Aufgaben zu integrieren. Hierzu gehören u.a. ein leistungsfähiges Sourcecode- und Konfigurationsmanagement.

Das Softwaredesign ist geprägt durch die Forderungen:

- Konsequente Wiederverwendbarkeit von Software
- Einheitliches Design aller Applikationen
- Vermeidung von Redundanz

Hierfür wurde für AVES das Simulationswerkzeug 2Simulate entwickelt [3]. 2Simulate ist in C++ programmiert und verfolgt konsequent den objektorientierten Ansatz. Es besteht aus drei Komponenten:

- 2Simulate Control Center (2SimCC)
- 2Simulate Real-Time Framework (2SimRT)
- 2Simulate Model Control (2SimMC)

Mit 2SimCC steht dem Anwender eine Applikation mit einer komfortablen graphischen Benutzeroberfläche (GUI) unter MS-Windows zur Bedienung und Steuerung der Simulation zur Verfügung. Die 2SimRT-Klassenbibliothek stellt den Echtzeitkern in Form einer API (Application Programming Interface) zur Verfügung und sorgt in Simulationsapplikationen für eine deterministische Ablaufsteuerung von diversen Echtzeit-Tasks. Eine individuelle Datenprozessierung wird durch das Bereitstellen von Callback-Aufrufen ermöglicht. Das Erstellen von 2Simulate-Applikationen ist unter MS-Windows, QNX und Linux möglich.

Eine 2Simulate-Applikation kann als minimale Standalone-Applikation betrieben werden oder bei Anbindung an 2SimCC mittels eines Connection-Tasks (ConTask) als 2Simulate-Targetapplikation. Zur Verbindung zwischen 2SimCC und einer 2Simulate-Targetapplikation wird das 2Simulate Control Protocol (TSCP) benutzt (siehe Abb. 2 und Abb. 4). Mit 2SimMC steht eine Modellschnittstelle für die implementierten Modelle zur Verfügung und erlaubt

deren Steuerung und Kontrolle innerhalb eines Modell-Tasks (MdlTask) [4].

Zurzeit können Modelle aus Simulink oder in der C++-basierten 2Simulate Modelling Language (TSML) geschriebene implementiert werden. In Planung befindet sich die Implementierung für FMI (Functional Mockup Interface).

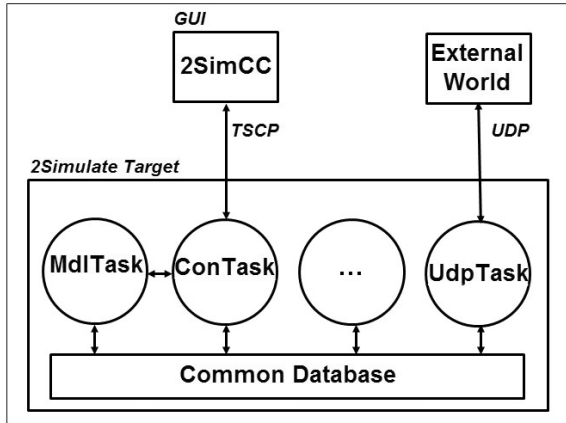


Abbildung 2. Typische 2Simulate-Targetapplikation

### 3 Grundlagen

In diesem Kapitel sollen die mathematischen Grundlagen für das implementierte Trimmverfahren aufgezeigt werden. Ein nichtlineares dynamisches System  $n$ -ter Ordnung kann als System von Differentialgleichungen erster Ordnung durch seine Vektordifferentialgleichung und Vektorausgangsgleichung beschrieben werden [7]:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), & \mathbf{x}(t_0) &= \mathbf{x}_0, \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)), \end{aligned}$$

wobei

- $\mathbf{x}$  der Zustandsvektor
- $\mathbf{u}$  der Eingangsvektor
- $\mathbf{y}$  der Ausgangsvektor
- $t$  die Zeit

ist.

Der stationäre Zustand eines dynamischen Systems entspricht der generellen Idee einer Ruhelage. Für die nähere Betrachtung eines stationären Betriebspunktes (Trimmpunktes) gilt es zu unterscheiden, ob das System eine absolute Ruhelage einnimmt oder lediglich einen beschleunigungsfreien Zustand. Ein einfaches

Feder-Masse-Dämpfer-System ist im Ruhezustand in absoluter Ruhe, d.h. nicht nur seine dynamischen Größen sind Null, sondern auch seine kinematischen.

Im Falle eines weitaus komplexeren Systems wie z.B. einem Flugzeug ist dies nur am Boden der Fall. Im Fluge sind zumindest einige kinematische Größen wie die translatorischen Geschwindigkeiten ungleich Null und nehmen einen festgelegten Wert ein, entsprechend dem definierten Trimmpunkt. Daher ist es notwendig, den Zustandsvektor  $\mathbf{x}$  in einen kinematischen Anteil  $\mathbf{x}_{kin}$  und einen dynamischen Anteil  $\mathbf{x}_{dyn}$  aufzuteilen [6]:

$$\mathbf{x}(t) = [\mathbf{x}_{dyn}(t)^T, \mathbf{x}_{kin}(t)^T]^T,$$

wobei

$$\dot{\mathbf{x}}_{dyn} = \mathbf{f}_{dyn}[\mathbf{x}_{dyn}(t), \mathbf{x}_{kin}(t), \mathbf{u}]$$

$$\dot{\mathbf{x}}_{kin} = \mathbf{f}_{kin}[\mathbf{x}_{dyn}(t), \mathbf{x}_{kin}(t)] .$$

Im dynamischen Anteil  $\mathbf{x}_{dyn}$  sind diejenigen Zustandsgrößen enthalten, deren Ableitung vom Eingangsvektor  $\mathbf{u}$  abhängig ist, während der kinematische Anteil  $\mathbf{x}_{kin}$  alle restlichen Elemente beinhaltet, deren Ableitung unabhängig von  $\mathbf{u}$  ist.

Verallgemeinert lässt sich das Trimmproblem zur Herstellung eines stationären Betriebspunktes (Trimmpunkt) eines zeitinvarianten dynamischen Systems wie folgt definieren:

$$\mathbf{f}[\dot{\mathbf{x}}_{soll}, \mathbf{y}_{soll}, \mathbf{x}_{trim}, \mathbf{u}_{trim}] = 0$$

Hierbei werden Sollwerte für die Vektoren  $\dot{\mathbf{x}}_{soll}$  und  $\mathbf{y}_{soll}$  im Rahmen der physikalischen Möglichkeiten des Modells vorgegeben. Diese werden als Trimmforderungen bezeichnet. Die Elemente des zur Erfüllung der Trimmforderungen notwendigen Zustandsvektors  $\mathbf{x}_{trim}$  und Eingangsvektors  $\mathbf{u}_{trim}$  werden als Trimmgrößen bezeichnet. Im allgemeinen Fall handelt es sich hierbei um ein nichtlineares Gleichungssystem, das nicht analytisch lösbar ist. Deshalb wird ein numerisches Iterationsverfahren zur Lösung dieses nichtlinearen Gleichungssystems benutzt. Dabei werden die Trimmgrößen solange variiert, bis die Trimmforderungen alle mit einer hinreichenden Genauigkeit erfüllt sind. Das hier verwendete Newton-Verfahren [9] gehört zur Klasse der gradientenbasierten Iterationsverfahren.

## 4 Das Trimmverfahren in der Praxis

In der Praxis kann das Verfahren zur Trimmung eines generischen dynamischen Systems in zwei Aspekte aufgeteilt werden:

1. Das Aufstellen des Trimmgesetzes.
2. Das Durchführen von Trimmrechnungen unter Anwendung dieses Trimmgesetzes mit variierenden Werten.

Unter der Aufstellung des Trimmgesetzes wird hier die Auswahl der Trimmgrößen und –forderungen verstanden [5]:

- Für die zu variierenden Trimmgrößen werden Elemente aus dem Zustandsvektor  $\mathbf{x}$  und dem Eingangsvektor  $\mathbf{u}$  ausgewählt.
- Für die Trimmforderungen werden Elemente aus dem Derivatvektor  $\dot{\mathbf{x}}$  und dem Ausgangsvektor  $\mathbf{y}$  ausgewählt.
- Die Anzahl der Trimmforderungen und Trimmgrößen muss gleich sein, damit das Trimmproblem eindeutig lösbar ist !

Zur Durchführung einer Trimmrechnung mit dem so aufgestellten Trimmgesetz für beliebige spezifische Trimpunkte gilt weiterhin:

- Für die Trimmforderungen müssen Sollwerte angegeben werden.
- Alle Elemente des Zustandsvektors  $\mathbf{x}$  und des Eingangsvektors  $\mathbf{u}$  müssen auf definierte Werte gesetzt werden. Falls es sich dabei um eine Trimmgröße handelt, wird der angegebene Wert als Startwert für die iterative Lösung des Gleichungssystems verwendet, falls nicht, wird der angegebene Wert in der Trimmrechnung direkt verwendet und ist auch der Anfangswert für die nachfolgende Simulation.

Durch die Trimmrechnung werden dann

- einerseits die gesuchten Trimmgrößen bestimmt,
- und andererseits auch alle Elemente des Derivatvektors  $\dot{\mathbf{x}}$  und des Ausgangsvektors  $\mathbf{y}$ , die keiner Trimmforderung unterliegen, automatisch mitberechnet.

## 5 Implementierung des Trimmverfahrens in 2Simulate

Die Implementierung des Trimmverfahrens in das Simulationswerkzeug 2Simulate ist in zwei Teilen realisiert. Zum einen erlaubt ein Trimm-Menü im 2Simulate Control Center (2SimCC), also der zentralen graphischen Bedienoberfläche (GUI), das Definieren des Trimmgesetzes sowie das Aktivieren der Trimmrechnung mit aktuell angegebenen Werten (siehe Abb. 6). Hierzu gehört auch die Festlegung der Ziel- bzw. Abbruchkriterien des Iterationsverfahrens (Fehlertoleranzgrenze, maximale Anzahl der Iterationsschritte), mit denen ein Ergebnis erzielt werden soll. Die GUI erlaubt das Überwachen des Trimmergebnisses und den anschließenden Start der Simulation von diesem ausgetrimmten Systemzustand.

Die Trimmrechnung selbst erfolgt auf dem 2Simulate-Target, auf dem das Modell implementiert ist (siehe Abb. 3). Das Modell wird innerhalb des Targets von der 2Simulate Model Control Schnittstelle (2SimMC) gesteuert. Die Verbindung von 2SimCC zu 2SimMC erfolgt über das 2Simulate Control Protocol (TSCP). Über TSCP werden dem Modell alle Informationen des 2SimCC über das Trimmgesetz und die aktuellen Parameter mitgeteilt.

**Abbildung 3:** Modellanbindung im 2Simulate-Target

Danach wird auf dem Target die Trimmrechnung ausgeführt und das Ergebnis wiederum über TSCP an das 2SimCC transferiert, um es dem Nutzer in adäquater Form zu präsentieren. Der Ablauf des Trimmverfahrens ist in Abbildung 4 skizziert.

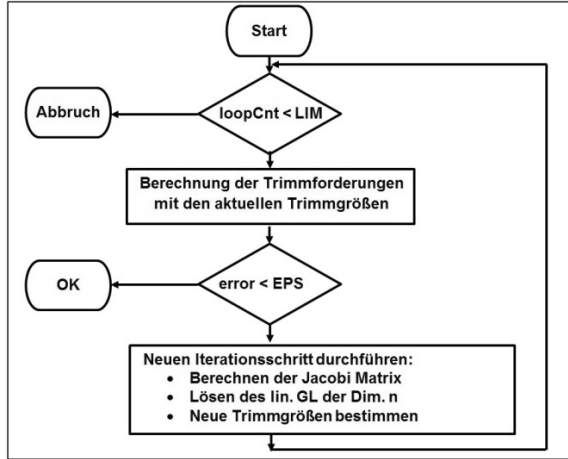


Abbildung 4. Ablaufsteuerung des Trimmverfahrens

## 5.1 Beispiel eines einfachen Zwei-Massen-Feder-Dämpfer-Systems

Die Implementierung und Anwendung soll an einem klassischen Zwei-Massen-Feder-Dämpfer-System gezeigt werden (siehe Abb. 5)

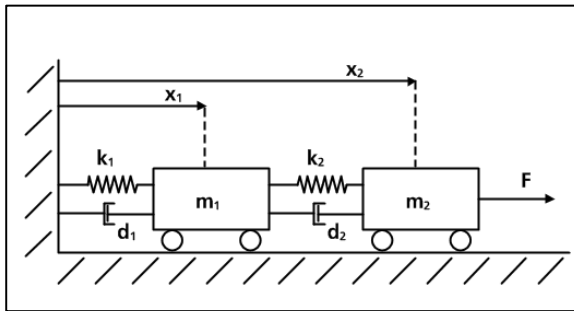


Abbildung 5. Zwei-Massen-Feder-Dämpfer-System

Die zugehörigen Bewegungsdifferentialgleichungen lauten:

$$\ddot{x}_1 = -\frac{k_1}{m_1}x_1 - \frac{d_1}{m_1}\dot{x}_1 + \frac{k_2}{m_1}(x_2 - x_1) + \frac{d_2}{m_1}(\dot{x}_2 - \dot{x}_1)$$

$$\ddot{x}_2 = -\frac{k_2}{m_2}(x_2 - x_1) - \frac{d_2}{m_2}(\dot{x}_2 - \dot{x}_1) + \frac{F}{m_2}$$

Mit Überführung in ein DGL-System erster Ordnung mit dem Zustandsvektor  $\mathbf{x}$  und seiner Ableitung  $\dot{\mathbf{x}}$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix} \Rightarrow \dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \ddot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_2 \end{bmatrix}$$

folgt die Zustandsraumdarstellung:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_1+k_2}{m_1} & -\frac{d_1+d_2}{m_1} & \frac{k_2}{m_1} & \frac{d_2}{m_1} \\ 0 & 0 & 0 & 1 \\ \frac{k_2}{m_2} & \frac{d_2}{m_2} & -\frac{k_2}{m_2} & -\frac{d_2}{m_2} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m_2} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Wie schon in der GUI von [11] wird das Trimmproblem im 2Simulate-Trim-Menü in einer 4-Quadranten-Struktur dargestellt (Abb. 6). Mit den beiden Quadranten auf der linken Seite können die zu variierenden Trimmgrößen (Trim Variables) festgelegt werden, während mit den beiden Quadranten auf der rechten Seite die Trimmforderungen (Trim Requirements) festgelegt werden.

Im oberen linken Quadranten sind die Elemente des Zustandsvektors und darunter die des Eingangsvektors dargestellt. Eine Checkbox erlaubt die Selektion der Trimmgrößen, in diesem Fall die Positionen  $x_1$  und  $x_2$ .

Auf der rechten Seite werden im oberen Quadranten die Elemente des Derivatvektors und darunter die des Ausgangsvektors dargestellt. Die Darstellung von Derivatgrößen erfolgt durch den Namen der Zustandsgröße mit einem abschließenden Hochkomma. Eine Checkbox erlaubt die Selektion von Größen für die Trimmforderungen, in diesem Fall die Beschleunigungen  $\ddot{x}_1$  und  $\ddot{x}_2$ , deren Sollwerte hier auf Null gesetzt sind. Eine Information über die Anzahl der ausgewählten Trimmgrößen und Trimmforderungen ist im unteren Bereich gegeben.

**Abbildung 6.** 2Simulate Trimm-Menü am Bsp. eines  
Zwei-Massen-Feder-Dämpfer-Systems

Mit den Parametern

$$\begin{aligned} m_1 &= 5 \text{ kg}, & k_1 &= 4 \text{ N/m}, & d_1 &= 1 \text{ Ns/m}, \\ m_2 &= 1 \text{ kg}, & k_2 &= 2 \text{ N/m}, & d_2 &= 3 \text{ Ns/m} \text{ und} \\ F &= 40 \text{ N} \end{aligned}$$

ergibt sich das Ergebnis einer ausgetrimmten Ruhelage des Systems bei folgenden Positionen der beiden Massen, welches aufgrund des linearen Zusammenhangs bereits nach einem Iterationsschritt erreicht wird:

$$\begin{aligned} x_1 &= 10 \text{ m} \\ x_2 &= 30 \text{ m.} \end{aligned}$$

## 6 Zusammenfassung / Ausblick

Die Funktionalität einer Online-Trimmung eines echtzeitfähigen dynamischen Systems ist im Simulationswerkzeug 2Simulate erfolgreich implementiert worden. Neben einfachen klassischen Testmodellen wie dem hier gezeigten Zwei-Massen-Feder-Dämpfer-System ist das Verfahren auch mit komplexeren dynamischen Systemen wie einem nichtlinearen 6-Freiheitsgrad-Flächenflugzeugmodell sowie einem äußerst komplexen 6-Freiheitsgrad-Helikoptermodell einer Eurocopter EC135 getestet worden. Der praktische Einsatz ist in den nächsten Monaten geplant.

## 7 Literatur

- [1] Duda, H., Gerlach, T., Advani S., Potter M. *Design of the DLR AVES Research Flight Simulator*. AIAA Modeling and Simulation Technologies Conference, Boston, MA, 2013
- [2] Gerlach, T., Durak, U. *AVES SDK: Bridging the Gap between Simulator and Flight Systems Designer*. AIAA Modeling and Simulation Technologies Conference, Dallas, TX, 2015
- [3] Gotschlich, J., Gerlach, T., Durak, U. *2Simulate: A Distributed Real-Time Simulation Framework*. ASIM STS/GMMS Workshop, Reutlingen, 2014
- [4] Gerlach, T., Durak, U., Gotschlich, J. *Model Integration Workflow for Keeping Models up to Date in a Research Simulator*. SIMULTECH, 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, 2014
- [5] Buchholz, J. *Regelungstechnik und Flugregler*. Vorlesungsmanuskript, Hochschule Bremen, 2016, <http://buchholz.hs-bremen.de/rtfr/skript/skript10.pdf>
- [6] De Marco, A., Duke, E., Berndt, J. *A General Solution to the Aircraft Trim Problem*. AIAA Modeling and Simulation Technologies Conference, Hilton Head, SC, 2007
- [7] Stevens, B., Lewis, F. *Aircraft Control and Simulation*, Wiley, 2. Auflage, 2003
- [8] Brockhaus, R. *Flugregelung*. Springer, 2. Auflage, 2001
- [9] Stör, J. *Einführung in die Numerische Mathematik*. Springer, 1979
- [10] Wikipedia, <https://de.wikipedia.org/wiki/Trimmung>
- [11] Buchholz, J. *trimmod*. <https://www.mathworks.com/matlabcentral/fileexchange/268-trimmod>

# Open Innovation/Sagitta – Implementation and Validation of a Real-Time Flight Dynamics model for Simulation, Integration Testing and Pilot Training

Dipl.-Ing. (Univ.) Richard O. Kuchar  
German Aerospace Center (DLR), Institute of System Dynamics and Control (SR)  
82234 Weßling, Germany  
*richard.kuchar@dlr.de*

The OpenInnovation/Sagitta project has its focus on the contribution of innovative technologies for the application within the UAV sector. Therein a number of research facilities and academic institutions – under the lead management of Airbus Defence and Space – are working together in order to resolve relevant questions: The Technical University of Munich (TUM), the University of the Armed Forces Munich (UniBW), the Technische Hochschule Ingolstadt (THI) and the TU Chemnitz (TUC), as well as the German Aerospace Center (DLR, being involved with a total of three institutes). Within project scope, a scaled demonstrator UAV aircraft has been designed for in-flight verification capabilities of the derived scientific project results. Aside the overall integration tasks in advance of its maiden flight, the aircraft is currently undergoing intensive simulation and integration testing activities on various levels. Within this paper, an overview of the general setup for these activities is given - specifically focusing on the required flight dynamics model (FDM) which depict the actual physical properties of the aircraft. Furthermore the respective implementation challenges due to project requirements are discussed in detail. Finally the performed validation tasks are described - as required to achieve clearance of the FDM for formal testing.

## 1 Introduction

During the various stages of aircraft design numerous questions and uncertainties regarding the dynamical behavior, controllability of the system-under-design as well as the embedded system interaction arise. This information gap can only be addressed by the implementation of a flexible and full-featured simulation infrastructure, that covers the various design stages with the most recent databases, models, soft- and hardware implementations. In order to fulfill these requirements within the scope of the Sagitta Demonstrator UAV project (see [1] and figure 1) the flexible simulation framework "**Sagitta Simulation and Integration Testing Environment**" (SIT) has been implemented. The SIT environment covers all relevant hard- and software elements, required to perform Model- (MiL), as well as Hardware-in-the-Loop (HiL) simulations - including the system-under-test (the Sagitta Demonstrator UAV).



Figure 1: Sagitta Demonstrator Aircraft (Courtesy of Airbus DS)

The presented paper aims at an introduction of this environment, as well as an in-depth presentation of the respective flight dynamics model (FDM). This model serves as the pivotal point in understanding the aircraft's dynamical properties: It receives commands (control surfaces, propulsion) by the Flight Control System (FCS) which result in an instantaneously al-

tered flight state. The resulting effects on the aircraft and system states are then fed back to the flight control system - closing the loop.

## 2 Simulation and Integration Testing environment

Based on top-level demands, the following requirements had to be facilitated within the SIT environment architecture:

- Provide system simulation capabilities under **real-time** conditions and with dedicated **dynamic behavior** for each embedded component as well as for the overall aircraft system
- All **component models** should be exchangeable with their corresponding **hardware counterpart**
- Allow for the utilization of **one** integrated, yet expandable SIT environment architecture: Changes of operational modes from engineering simulation, integration testing to pilot training should be possible with minimal effort
- Allow for **integration testing** of the aircraft and its components on system level (L2) and full-integration levels (L1/L0)
- Prepare the SIT environment for the flexible integration of engineering and **high-fidelity** (test-bench, wind-tunnel/CFD derived) models
- Allow for the insertion of defined **fault conditions** into the setup

The derived overall SIT architecture has been based on various experiences with simulation environments for UAV systems (e.g. see [2]). It was proposed by Airbus Defence and Space to utilize two in-house simulation and testing tools:

- **SIRIUS** (see [3]): Simulation framework with Airbus AP2633 [4] compliant model structure, providing a simulation state machine, scheduling and data-exchange services
- **AIDASS**: Hardware interfacing platform and simulation environment (e.g. MIL-STD-1553, ARINC 825, RS422/232, Ethernet etc.), test automation and execution, integrated data logging
- **SIRIUS-AIDASS Gateway**: Utility/Service layer for the optimized bi-directional data exchange between SIRIUS and AIDASS

Based on the requirements, the top-level architecture for the Sagitta L1/L0 SIT environment as depicted in figure 2 was developed. Two common network layers allow a flexible exchange of the simulation setup - the simulation network for data exchange, and the maintenance network for SIT internal data traffic (logging, xmlrpc commands). The core element is represented by the "simenv01" node - performing all relevant simulation computations and data exchange - including the flight dynamics model. Aside from this, there is the possibility to attach further equipment to the simulation network.

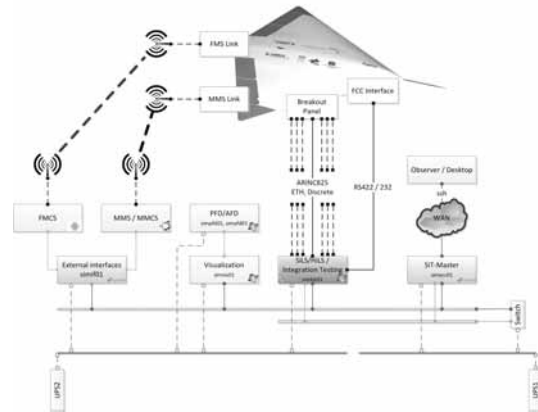


Figure 2: Sagitta L0/L1 SIT environment architecture

### 2.1 Simulation models

In general the Sagitta SIT environment consists of **ENVIRONMENT**-, **COMPONENT**- and **UTILITY**- models. The former resemble all relevant aspects of the "virtually flying" aircraft (e.g. the FDM with embedded aerodynamics, weight-and-balance, propulsion, atmosphere and landing gear model), the latter are similar to the actual on-board hardware components. Depending on the scope of the respective rig and the overall project progress, each component model can be replaced by its corresponding hardware.

*Within this paper the focus is set to the implemented flight dynamics model and its embedded subcomponent models - as it serves as the core model for the simulated setup.*

See figure 3 for the basic interaction between the FDM and its surrounding.

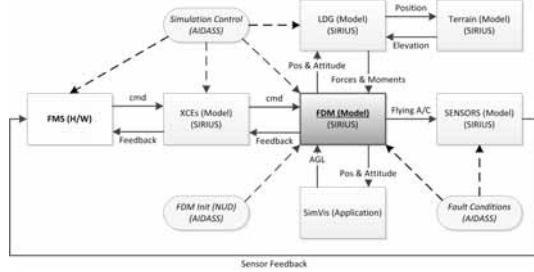


Figure 3: Sagitta FDM interaction

### 3 FDM implementation

This section describes all relevant steps taken for the assembly of the full-featured Sagitta flight dynamics model. In order to use well tested code and avoid potential mishaps, the model was built in close accordance to the Modelica Standard Library (see [5]) and the DLR-SR FlightDynamics Library (see [6]). Never the less some modifications had to be made, in order to take specific requirements and model conditions into account.

#### 3.1 Modelica FlightDynamics Library

The DLR-SR FlightDynamics Library has been utilized as the basic framework for the integration and set-up of the Sagitta Demonstrator UAV flight dynamics model: The DLR FlightDynamics library is used at the DLR-SR department for Aircraft System Dynamics for object-oriented multi-physics modeling of aircraft and their systems in Modelica [5]. Exploiting Modelica features, versatile models can be generated ranging from simple 3-DOF mass point models to extended rigid-body and elastic models. These can be combined with complex environments, energy and power system models. The library also provides environment models such as the ISA standard atmosphere, the World Magnetic model (WMM2015) and the Earth Gravitational model (EGM96), which are automatically embedded within the FDM scope (see figure 4). Data exchange between the various model components is provided via the utilization of generalized bus structures.

The following generalized Newtonian equations of motion for rigid aircraft apply for the Sagitta Demonstrator and are implemented within the flight dynamics model. The respective properties follow the general notation as stated within LN9300 and figure 5.

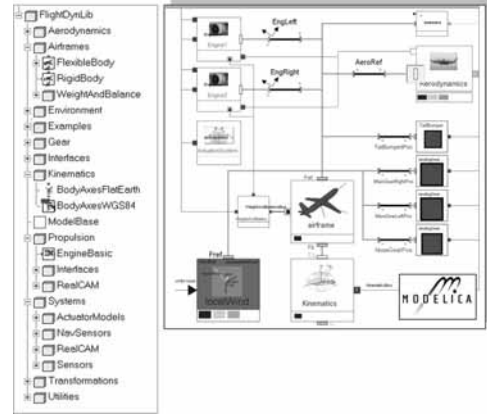


Figure 4: DLR-SR Flight Dynamics Library

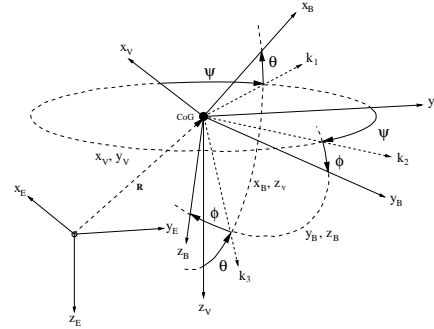


Figure 5: Flight mechanics axes system

First the equations for the respective external forces X, Y and Z can be denoted in the body fixed system as follows:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = m \begin{pmatrix} \dot{u} + qw - rv + g_0 \sin \Theta \\ \dot{v} + ru - pw - g_0 \cos \Theta \sin \Phi \\ \dot{w} + pv - qu - g_0 \cos \Theta \cos \Phi \end{pmatrix} \quad (1)$$

with

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_{aero} + T_x + X_{ldg} \\ Y_{aero} + T_y + Y_{ldg} \\ Z_{aero} + T_z + Z_{ldg} \end{pmatrix} \quad (2)$$

The respective moments L, M and N can be obtained in a similar fashion - taking an inherently symmetrical aircraft (XZ plane) and principal axes into account:

$$\begin{pmatrix} L \\ M \\ N \end{pmatrix} = \begin{pmatrix} I_x \dot{p} + (I_z - I_y) qr \\ I_y \dot{q} + (I_x - I_z) rp \\ I_z \dot{r} + (I_y - I_x) pq \end{pmatrix} \quad (3)$$

The respective share of the total external forces and moments are integrated into the flight dynamics model via specific disciplinary models. These are described in subsequent sections.

### 3.2 Aerodynamic model integration

The aerodynamic model (ADM) for the Sagitta demonstrator UAV has been compiled by TUM-AER and THI from wind-tunnel measurements, as well as supportive CFD computations (see [7] and [8]). The ADM provides aerodynamic forces and moments based on speed, altitude, angle of attack, side slip angle, respective rates and control deflections.

The nominal ADM has been issued as a Matlab script and was therefore not directly integrable as real-time database within the FDM - thus requiring a precise transformation of the database.

In order to obtain a real-time capable interpolation scheme, an **ANSI-C based linear, vectorized interpolation method** has been utilized. The respective C-code is directly embeddable in Modelica via the C-function call capability of Modelica.

```
function Stat_noVT_CX_
input Real alpha[1];
input Real beta[1];
output Real Stat_noVT_CX_Aero_A_B;

external "C" Stat_noVT_CX_Aero_A_B =
    Stat_noVT_CX_(alpha, beta);

annotation (Include="#include \"
    SagittaADM_ph03v02.c\"");
end Stat_noVT_CX_;
```

**Code 1:** Example - Modelica function wrapper for ANSI-C based ADM calls

The derived code proofed to be far more computational performant: Depending on the actual hardware, accelerations of up to a factor of 20 could be observed.

In order to verify the correct transformation of the nominal Matlab ADM into the Modelica/FDM version, extensive testing has been performed, evaluating all database dimensions with small step-size within the overall flight envelope.

### 3.3 Propulsion model integration

The propulsion model (PROP) has been compiled by TUM-LLS - based on the evaluation of test bench data of the respective jet engine - see [9]. The propulsion model provides forces and moments of the jet engine due to aerodynamic parameters and the actual throttle setting.

As the nominal propulsion model has been implemented in Simulink, it took some adaptations in order to integrate it into the Modelica context. Care has to be taken in order to **excavate all time dependent content** - such as transfer functions, integrators and delays - as the Simulink Coder generated surrogate ANSI-C code does not provide external timing input, but instead performs "tick" iterations. In order to provide exact external timing it is therefore essential to transfer all relevant time-dependent content up to the Modelica implementation layer (see figure 6).

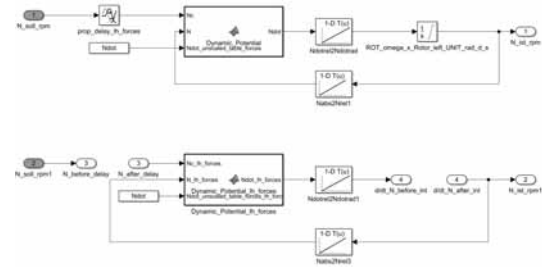


Figure 6: Example of Simulink model with time-dependent blocks (above) - and with external ports (below)

The following steps had to be taken in order to prepare for FDM integration:

1. **Remove all time-dependent content** from the underlying Simulink model and provide external interfaces (see figure 6)
2. Prepare a fitting **wrapper model in Modelica** for the reconstruction of the underlying dynamics (see figure 7)
3. Check, if the **original Simulink model has no time-dependencies** any more - e.g. compare solver settings ("discrete" solver by default)
4. **Export** the respective Simulink model with Simulink Coder
5. Integrate the exported code into the Modelica model and provide a relative path to the exported code collection

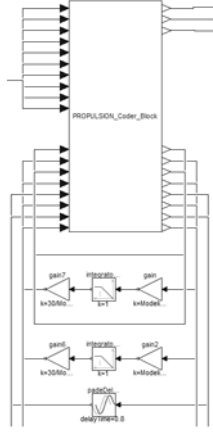


Figure 7: Exported PROP model embedded in Modelica with external ports

The following two API-functions are provided by Simulink Coder in order to utilize the exported model code within various application environments: The "InitModel" function is required for basic initialization of the model, whereas the "OneStep" function evaluates the underlying code.

```
function ENG_MOD_Coder_InitModel_  
external "C" ENG_MOD_Coder_InitModel_();  
  
annotation (Include="#include \"  
    ENG_MOD_Coder.c\"");  
end ENG_MOD_Coder_InitModel_;
```

**Code 2:** Modelica function wrapper for exported ANSI-C propulsion model - Initialize

```
function ENG_MOD_Coder_OneStep_  
  
input Real[18] dU;  
output Real[26] dY;  
  
external "C" ENG_MOD_Coder_OneStep_(dU, dY);  
  
annotation (Include="#include \"  
    ENG_MOD_Coder.c\"");  
end ENG_MOD_Coder_OneStep_;
```

**Code 3:** Modelica function wrapper for exported ANSI-C propulsion model - OneStep

In order to verify the correct transformation of the nominal Simulink model into the Modelica/FDM version, extensive testing has been performed, applying and evaluating numerous use-cases.

### 3.4 Actuator model integration

The actuator models (ACT) have been compiled by DLR-FT - based on the evaluation of test bench data for the respective actuator model taking the actual kinematic conditions into account. Therefore a dedicated model had to be exported for each of the eight present actuators and control surfaces of the Sagitta Demonstrator UAV. The actuator models provide the actual transfer function between the actuator command and the respective control deflections, taking the overall actuator behavior, hinge moments and backlash into consideration.

The method for the integration of the delivered Simulink ACT models into the Modelica FDM context is identical to the PROP model.

### 3.5 Landing Gear model integration

During early stages of the project, a preliminary landing gear (LDG) model has been directly embedded within the Sagitta Flight Dynamics model. The test-bench derived model parameters for the LDG have been supplied by DLR-FT, actual modeling has been conducted at DLR-SR. Unfortunately, the final model parameters showed limit cycle oscillations during initial ground operations (especially during the initialization and braking phases), therefore it has been decided to **separate the LDG content from the FDM** into a stand-alone model.

Basic benefits of this decision were:

- **Independent** (and higher) **sample rate** than FDM
- Better **simulation load distribution** (LDG computed by dedicated CPU core)
- Specific use of a qualified solver (Simulink ode4)
- **High sampling of dedicated terrain model**, delivering distance-to-ground for each wheel

Finally the only interaction between the FDM and the LDG models is an exchange of the acting forces and moments via external ports.

### 3.6 Weight and Balance model

The weight and balance model is based on static measurement of the aircraft weight, as well as estimated

data for the respective inertia. The weight and balance model is directly connected to the propulsion model, therefore the continuous fuel flow is constantly integrated - resulting in ever changing mass, inertia and center of gravity values in-flight.

In order to allow a performant interpolation of the respective center-of-gravity locations and inertias, a polynomial evaluation has been utilized.

## 4 The assembled Sagitta FDM

As soon as all relevant component models have been integrated in the overall FDM context, it was possible to export the resulting model, utilizing the embedded features of the FlightDynamics library.

The following figure 8 gives an impression on the assembled FDM, containing all previously discussed component models.

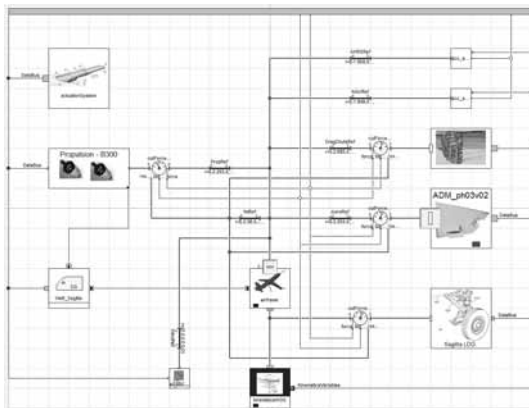


Figure 8: Sagitta Demonstrator FDM complete assembly of the model core (Modelica based)

## 4.1 Code Generation

Once the FDM has been assembled, the respective Modelica code can be automatically transformed, optimized and finally exported into ANSI-C code within the Dymola IDE – either for native simulation purposes, or for the use in external simulation and application environments. These external exports have been greatly simplified with the application of the standardized **Functional Mock-up Interface (FMI)** method (see [10]) - thus allowing the generation of

pure ANSI-C source code from the model with a standardized interface structure. For more information on general coding in the scope of flight dynamics modeling see [11].

See the following figures 9 and 10 for an impression of the actual code generation process and the final model software-structure within the SIRIUS simulation framework.

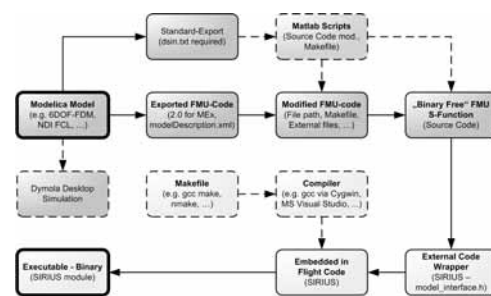


Figure 9: FDM Code generation process

As the FDM is exported as Functional Mock-up Unit (FMU) for Model Exchange, a dedicated solver has to be integrated into the respective model code. In order to aim for **enhanced precision in-air, a multi-step solver** (Adams-Bashforth, ode4) has been employed, **while for ground operations a Forward Euler solver** (ode1) is utilized. Multi-step methods gain efficiency by keeping and using the state-derivatives from previous steps rather than dropping it. The described solver switching is essential, as the FDM and LDG models are not exactly synchronized - therefore the danger of limit cycle oscillations on the ground had to be mitigated.

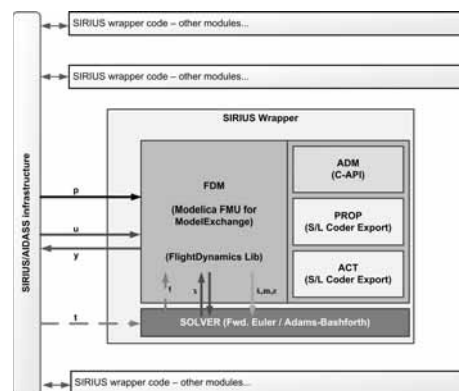


Figure 10: FDM model - SIRIUS module structure

The actual code generation and compilation is per-

formed by a highly automated toolchain, generating and manipulating all further required files (e.g. fmiEntries.h for input/output mapping), as well as the respective make file for the generation of SIRIUS module binaries. Within the SIT environment all code is compiled for the CentOS and MS Windows platforms, using GNU make and GNU gcc tool-chains, as well as MS Visual Studio.

## 5 FDM Validation

The implemented FDM required different levels of validation in order to be employed for formal testing: Starting with sanity checks and basic integration tests of the respective component models (aero database, propulsion model, actuator models) it was decided to **perform static and dynamic comparisons** with the TUM-FSD FDM derived for flight control law synthesis. Due to a project decision these two models were developed in a dissimilar way - furthermore, as the TUM-FSD FDM had a different application focus.

Starting with trim analysis - 240 trim cases were compared (mass, speed and altitude variations) - utilizing the TRIM FDM for the SIT part. Figure 11 shows an iso-line representation of the actual calculated differences between the TUM-FSD FDM and the SIT FDM. Numerous aerodynamic, kinematic and propulsion parameters have been compared for all trim cases, showing good accordance throughout the entire envelope.

Additionally, the models have been linearized along all trimmed points - resulting in lateral and longitudinal pole plots.

The Sagitta SIT environment has been prepared to import existent TUM-FSD L3 test cases. A subset of these test cases have been specifically generated for validation purposes between the two models. See figure 12 for typical comparison results - once for a discrete gust encounter, the other part showing turbulent flight. The results show in general good accordance - although perfect alignment is not achievable, due to a slightly different setup around the FDM models.

### 5.1 Visualization system

In the course of the project, it has been decided to add a dedicated visualization system to the setup in order to allow for visual inspection and pilot training. The

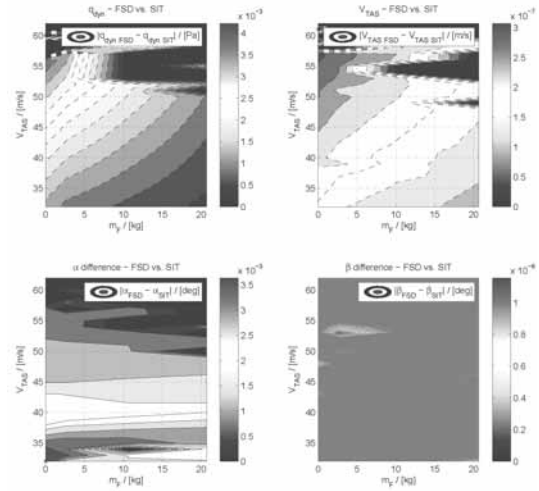


Figure 11: Example FDM trim result comparison - aerodynamic properties (iso-lines)

DLR-SR **visualization tool SimVis** (see [12] and figure 13) has been employed for this task.

As engineering judgment with respect to the visual impression adds to the overall validation, this system can be considered an integral part of the SIT setup.

Furthermore, some elevation measurements (AGL) are actually utilized by the FDM: Specifically for the determination of shear wind, turbulence levels and ground effect calculations.

## 6 Conclusion

The full path from the respective requirements to a full-featured and validated flight dynamics model for application within the Sagitta SIT environment has been outlined in this paper. Special focus has been given to the "lessons-learned" in the course of the implementation, as various component models were provided by project partners - resulting in differing implementation- and tool-approaches.

Finally, various validation measures have been employed in order to assure the integrity of the model for formal testing activities. As the respective results showed good agreement with independent reference solutions, the actual use of the FDM within the SIT environment proved to be valid.

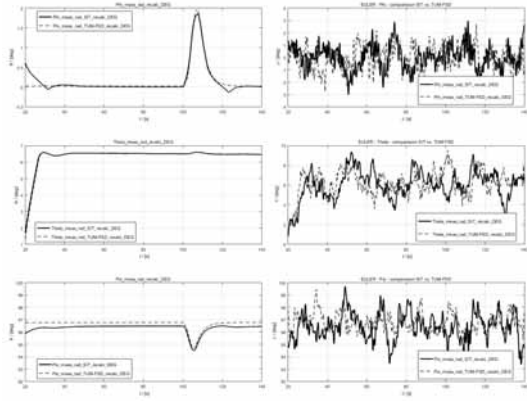


Figure 12: SIT L3 comparison - Euler Angles with discrete gust (left) and turbulence (right) - (Courtesy of TUM-FSD)



Figure 13: SimVis visualization of the virtual aircraft

## Acknowledgements

The author would like to thank Prof. E. Özger (THI) for testing the real-time ADM, A. Bierig (DLR-FT) for the preparation of the Sagitta actuator models, and L. Bougas (TUM-LLS) for supplying the Sagitta propulsion model.

The author would like to show gratitude to the team at TUM-FSD for their enduring assistance regarding the validation measures of the SIT FDM - especially to M. Geiser and V. Schneider. Furthermore, the author wishes to thank the team at Airbus Defence and Space for their continued support - especially S. Prechtel (AIDASS), G. Staudte (SIRIUS) and F. Westerbuhr.

## References

- [1] J. Seifert, “SAGITTA - Nationale Forschungskoooperation für fortschrittliche

UAV-Technologien im Rahmen der Open Innovation Initiative von Cassidian”, *DLRK 2012*, no. 1352, 2012.

- [2] C. Kulkarni, C. Teubert, G. Gorospe, D. Burgett, C. C. Quach, and E. Hogge, “Design, Development, and Testing of a UAV Hardware-in-the-Loop Testbed for Aviation and Airspace Prognostics Research”, *N/A*, 2016.
- [3] U. Sennes and R. Schabenberger, “Model Integration with the Common Simulation Framework SIRIUS”, in *AIAA Modeling and Simulation Technologies Conference*, 2011, p. 6587.
- [4] S. Val Serra, “AP2633 - Develop, Integrate and Validate Shared Simulation Models, Issue C”, Airbus France, Tech. Rep., 2008.
- [5] E. Hilding, M. Otter, and S. E. Mattsson, “Modelica Language Specification Version 3.3”, Modelica Association, Tech. Rep., May 9, 2012.
- [6] A. Klöckner, G. Looye, R. Müller, R. Kuchar, F. Re, and M. Leitner, “Object-Oriented Aircraft Modeling with the DLR FlightDynamics Library”, in *AIRTEC 2014*, Oct. 28, 2014.
- [7] A. Hövelmann and C. Breitsamter, “Aerodynamic Characteristics of the Sagitta Diamond Wing Demonstrator Configuration”, in *DLRK 2012*, Deutsche Gesellschaft für Luft- und Raumfahrt-Lilienthal-Oberth eV, 2012.
- [8] E. Özger, “Aerodynamic Model Validation of Unmanned Research Demonstrator SAGITTA”, in *DLRK 2012*, 2012.
- [9] L. Bougas and M. Hornung, “Propulsion system integration and thrust vectoring aspects for scaled jet UAVs”, *CEAS Aeronautical Journal*, vol. 4, no. 3, pp. 327–343, 2013.
- [10] T. Blochwitz, “Functional Mock-up Interface for Model Exchange and Co-Simulation Specification Version 2.0”, Modelica Association, Tech. Rep., Jun. 25, 2014.
- [11] R. Kuchar and A. Klöckner, “Automatic Flight Code Generation from Multi-Physics models”, in *ODAS 2015*, 2015.
- [12] T. Bellmann, “Interactive Simulations and Advanced Visualization with Modelica”, in *Proceedings of the 7th International Modelica Conference; Como; Italy; 20-22 September 2009*, Linköping University Electronic Press, 2009, pp. 541–550.

# Modellierung komplexer Flughafen-Ereignisse für Human-In-The-Loop Simulationen

Sebastian Schier<sup>1</sup>, Anne Papenfuß<sup>1</sup> und Nils Carstengerdes<sup>1</sup>

<sup>1</sup>Deutsches Zentrum für Luft- und Raumfahrt, Braunschweig

*sebastian.schier@dlr.de*

Der Flughafen ist eine der Engstellen im Luftverkehrssystem. Startbahnen, Stellplätze und Terminals haben nur begrenzte Kapazitäten und führen bei ineffizienter Nutzung zu Wartezeiten am Boden und in der Luft. Zusätzlich wird der Flughafenverkehr durch besondere Ereignisse (z.B. Schneefall, Streik, Bedrohungen der Sicherheit) gestört. Zahlreiche europäische Forschungsprojekte entwickeln derzeit neue Verfahren und Systeme zur effizienten Nutzung der Flughafenressourcen. Sie sollen in den Human-In-the-Loop Simulationen des Deutschen Zentrums für Luft- und Raumfahrt validiert werden. Die Komplexität der besonderen Ereignisse ist dabei eine große Herausforderung. Die Ereignisse müssen realitätsnah und wiederholbar, aber auch frei von Lerneffekten und im Simulator umsetzbar abgebildet werden. Vortests und Training sollen die Qualität der Validierung sicherstellen, gleichzeitig müssen Budget und Vorbereitungszeit im Rahmen bleiben.

Dieses Paper beschreibt die Entwicklung und Umsetzung eines Modellierungsverfahrens für komplexe Flughafen-Ereignisse in Human-In-The-Loop Simulationen. Hierfür wird analysiert, welche Anforderungen aus der Theorie an die besonderen Ereignisse gestellt werden und wie diese in vergangenen Simulationen abgebildet wurden. Darauf folgend wird ein Prozessmodell erstellt und die Anwendung kritisch diskutiert.

## 1 Motivation

Das Europäische Luftverkehrssystem steht vor einer großen Herausforderung. Allein im Jahr 2015 haben sich die am Flughafen erzeugten Verspätungen um ca. 60% erhöht (vgl. [1]). Diese Entwicklung soll sich laut einem Bericht der europäischen Organisation zur Sicherung der Luftfahrt EUROCONTROL sogar noch verstärken (vgl. [2] S.4). Zusätzlich erschweren „besondere Ereignisse“ (z.B. Gewitter, Streik, Sicherheitsbedrohungen) die Umsetzung der Flugplanung. Durch diese Störungen werden die knappen Ressourcen am Flughafen zusätzlich belastet. Die enge Verknüpfung des europäischen Luftverkehrssystems führt dann zu weitreichenden Folgen an zahlreichen europäischen Flughäfen (vgl. [2]).

Derzeit entwickeln und untersuchen eine Vielzahl an Forschungsprojekten neue Systeme und Verfahren für die europäischen Flughäfen. Die vorhandenen Ressourcen am Flughafen sollen so effizienter genutzt werden. Entsprechend der aktuellen Herausforderungen müssen diese neuen Systeme und Verfahren auch unter dem Einfluss besonderer Ereignisse die Akteure am Flughafen unterstützen.

Das Deutsche Zentrum für Luft und Raumfahrt (Abk. DLR) stellt für diese Projekte unter anderem

sogenannte Human-In-The-Loop Simulationen (Abk. HITL, vgl. [3]) bereit. Akteure aus dem Flughafenbereich (z.B. Piloten, Lotsen, Fluglinien-Disponenten) können in den Simulatoren die neuen Systeme und Verfahren testen (vgl. Abbildung 1). An Hand der aufgezeichneten Simulationsdaten nimmt das DLR anschließend eine Bewertung und Einschätzung der Nutzbarkeit vor.



**Abbildung 1.** DLR Tower-Simulator für Fluglotsen

In Bezug auf die besonderen Ereignisse bieten die HITL-Simulationen den Vorteil, dass komplexe Mensch-Maschine und Mensch-Mensch-Interaktionen dargestellt werden können. Das Risiko einer Untersuchung am realen Flughafen (Eingriff in den Luftverkehr) entfällt. Die Modellierung der besonderen Ereignisse in der Simulation ist aber ein

komplexes Verfahren. Realitätsnähe und technische Möglichkeiten müssen abgestimmt werden. Test- und Trainingsaufwand müssen im Rahmen bleiben.

Das vorliegende Paper analysiert die theoretischen Anforderungen und die praktischen Umsetzungen vergangener Simulationen. Auf dieser Basis wird ein Modellierungsprozess vorgeschlagen und davon ausgehend dessen Anwendung in zwei Projekten diskutiert.

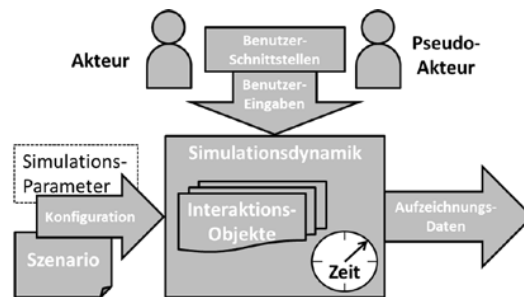
## 2 Theorie

### 2.1 Human-In-The-Loop Simulation

HITL-Simulationen werden in einer Vielzahl von Anwendungsbereichen verwendet und besitzen verschiedene Zielsetzungen und Funktionen (vgl. [3]). An dieser Stelle soll zusammenfassend dargestellt werden, welches Verständnis im vorliegenden Paper für HITL-Simulationen existiert.

**Zielsetzung:** Das Design und die Analyse neuer Systeme für den europäischen Luftverkehr werden durch eine entsprechende Validierungsrichtlinie strukturiert (vgl. [5]). Hier werden unter Anderem HITL-Simulationen als ein Forschungswerkzeug definiert. Systeme, bei denen Interaktionen mit Operateuren nötig sind, können mit Hilfe dieser Simulationen auf ihre Machbarkeit (engl. „Feasibility“ [6]) und ihre operationellen Vorteile untersucht werden. Die Verwendung als Trainingswerkzeug steht nicht im Fokus dieses Papers, wird aber abschließend diskutiert.

**Funktionsumfang:** Eine HITL-Simulation besteht nach der allgemeinen Definition von Narayanan et al. (vgl. [7]) aus Benutzerschnittstellen, einer Simulationsdynamik und Interaktionsobjekten. Die Interaktionsobjekte werden durch den Benutzer über die Benutzerschnittstellen verändert. Die Simulationsdynamik berechnet in Abhängigkeit der Eingaben den Zustand der Interaktionsobjekte. Aus der Luftfahrt-Perspektive verfeinern Dow et al. diese Definition (vgl. [8]). Sie beschreiben die Konfiguration der Simulation mit einem Szenario und trennen zwischen den Akteuren und Pseudoakteuren. Die Pseudoakteure stellen benötigte Rollen in den simulierten Prozessen dar. Ihre Arbeitsweisen und Benutzerschnittstellen entsprechen aber nicht der Realität. Die **Abbildung 2** stellt das resultierende Schema einer HITL-Simulation dar.



**Abbildung 2.** HITL-Simulation: Schematische Darstellung

### 2.2 Machbarkeitsstudie

HITL-Simulationen werden im Bereich der Luftfahrtforschung in der Regel in so genannten „Machbarkeitsstudien“ eingesetzt. Machbarkeitsstudien untersuchen generell die Erfolgsaussichten eines Projekts (vgl. [6]). Im Fall der Luftfahrtforschung wird die so genannte „operationelle Machbarkeit“ untersucht. Hier wird überprüft, ob neue Systeme oder Verfahren von den Akteuren verwendet oder umgesetzt werden können. Relevante Kriterien sind Leistung, Akzeptanz, sowie Anwendbarkeit. Für die Interaktion von menschlichen Akteuren mit technischen Systemen sind deshalb Maße wie Beanspruchung und Situationsbewusstsein wichtig (vgl. [5]). Bezieht sich die Machbarkeitsstudie auf besondere Ereignisse, so wird in diesem Paper von der **besonderen operationellen Machbarkeit** gesprochen.

Die besondere operationelle Machbarkeitsstudie in der HITL-Simulation fällt unter den Bereich der Laborexperimente (vgl. [9]). Die äußeren Rahmenbedingungen sind kontrollierbar, aber künstlich. Um die Ergebnisse der Studie in die Realität übertragen zu können ist eine der wichtigsten Anforderungen eine hohe Realitätsnähe der Simulation und abgebildeten Ereignisse.

Um trotz der häufig eher kleinen Probandengruppen (Lotsen und Piloten stehen nur sehr begrenzt zur Verfügung) inferenzstatistische Aussagen treffen zu können, wird in der Regel ein so genanntes „within-Design“ (vgl. [10]) verwendet. Ein Proband oder eine Probandengruppe durchlaufen alle Versuchsbedingungen nacheinander. Zur kontrollierten Bestimmung, welchen Einfluss ein neues System oder Verfahren hat, muss ein Proband mindestens einmal ein Ereignis ohne und einmal mit dem neuen System durchgeführt haben. Dieses Versuchsdesign beinhaltet den Nachteil, dass die Probanden unter anderem Übungseffekten und Ermüdungseffekten

unterlegen sein können. Diese lassen sich vor allem dann verringern, wenn man in der Lage ist die Ereignisse bei gleicher Auswirkung unterschiedlich abzubilden.

### 3 Methode

Die besonderen Ereignisse müssen nach den Vorgaben der Machbarkeitsstudie möglichst realitätsnah, gleich in ihren Auswirkungen, aber unterschiedlich in ihren Abbildungen dargestellt werden. Um diese Vorgaben zu erfüllen wurden die besonderen Ereignisse aus verschiedenen DLR HITL-Simulationen im Bereich Flughafen aus den letzten acht Jahren analysiert. In der **Tabelle 1** sind die Simulationsprojekte zusammengefasst.

Jahr	Beschreibung
2016	Effekt kooperativer Konzepte auf das Flughafenmanagement.
2013	Gleichzeitige Fernüberwachung von zwei Flughäfen.
2013	Einfluss von Bodenverkehr auf die Lotsenarbeit.
2011	Fernüberwachung von Flughäfen aus einer Flugsicherungszentrale.
2010	Gleichzeitige Fernüberwachung von zwei Flughäfen.
2009	Infrastrukturmaßnahme an einem deutschen Flughafen
2008	Textbasierte Verkehrsfreigabe am Flughafen

**Tabelle 1.** Analyisierte Simulationsprojekte

Die Analyse untersucht den Inhalt und die Auswirkung der Ereignisse. Auswirkung und Inhalt werden kategorisiert und auf die nötigen Daten für die Abbildung in der HITL-Simulation überprüft. Entsprechend der Vorgaben aus Kapitel zwei wird dann der Modellierungsprozess entworfen.

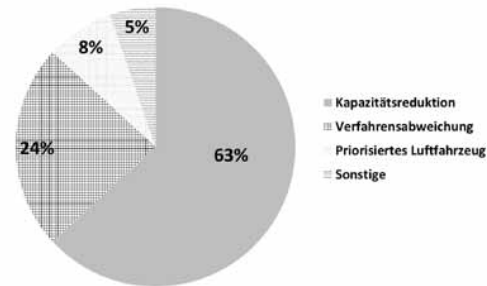
## 4 Ergebnisse

### 4.1 Analyse des Ereignisses

Aus den oben genannten Simulationsprojekten wurden 38 Ereignisse extrahiert. Die Beschreibung der Ereignisse wurde nach der Auswirkung des Ereignisses auf den Flughafenverkehr kategorisiert.

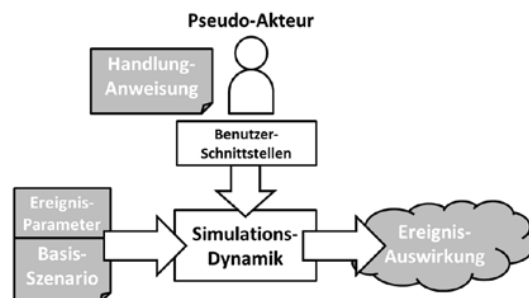
Die **Abbildung 3** zeigt, dass sich 95% der Ereignisse in drei Auswirkungskategorien einordnen lassen. Über 60% der Ereignisse beziehen sich auf die

Beeinträchtigung eines Teils der Flughafeninfrastruktur (z.B. Terminal, Rollweg, Startbahn). Etwa ein Viertel der Ereignisse bezieht sich auf die Abweichung von einem vorgegebenen Verfahren oder einer Anweisung (z.B. „versehentliches“ Rollen auf die Startbahn, Abweichung vom Endanflugpfad, etc.). Zehn Prozent der Ereignisse betreffen die Priorisierung eines oder mehrerer Luftfahrzeuge (z.B. Rettungsflug, Staatsflug, etc.).



**Abbildung 3.** Auswirkung der analysierten Ereignisse

Im Folgenden wurde untersucht, wie die einzelnen Ereigniskategorien in der Simulation umgesetzt werden. Hierbei wurde festgestellt, dass die Kapazitätsreduktion im Wesentlichen im Szenario verankert wird. Im Verlauf eines Basisszenarios mit Standardverkehr wird vorab definiert, welcher Flughafenteil betroffen ist und wie weit die Kapazität reduziert wird. Die Pseudo-Akteure müssen ggf. melden, dass die entsprechende Flughafenressource nur eingeschränkt zur Verfügung steht. Die Abweichung von Verfahren und die Priorisierung eines Luftfahrzeugs werden von den Pseudo-Akteuren eingeleitet. Sie benötigen hierfür eine Handlungsanweisung welches Luftfahrzeug von welcher Anweisung abweicht, bzw. wann sie Priorität erfragen sollen. Die **Abbildung 4** zeigt schematisch, wie die Ereignisse mit Hilfe von Basisszenario, Ereignisparametern und Handlungsanweisung generiert werden.

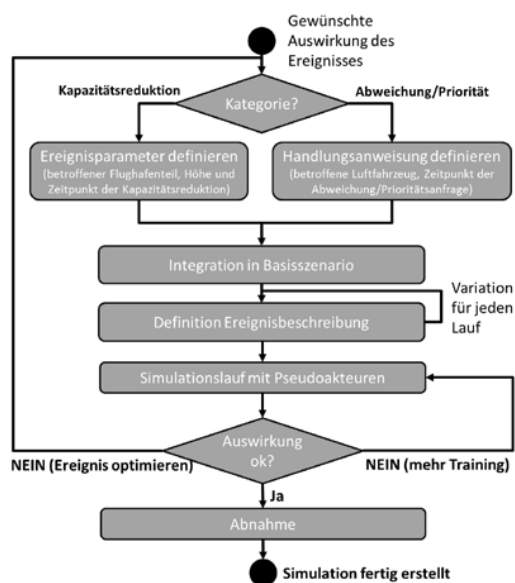


**Abbildung 4.** Schematische Ereignismodellierung

Zusätzlich zu diesen drei Quellen wurde festgestellt, dass immer eine Ereignisbeschreibung existiert, die dem Akteur den Grund für das Ereignis mitteilt (z.B. durch den Pseudoakteur oder als gedruckte Instruktion). Diese Ereignisbeschreibung ist nicht eindeutig mit der Auswirkung des Ereignisses, der Handlungsanweisung oder den Ereignisparametern verknüpft. Ein Beispiel hierfür ist das Ereignis „Gewitter“, dass mehrfach in der gleichen Simulation verwendet wurde, um einmal die Kapazität der Startbahn zu reduziert und einmal die Kapazitäten der Bodenabfertigung zu limitiert.

#### 4.2 Entwurf des Modellierungsprozesses

Die Anforderungen aus Kapitel 2.2 verlangen nach realitätsnahen Ereignissen, die in ihrer Auswirkung über die Simulationsläufe gleich bleiben, in ihrer Abbildung aber variieren müssen, um Lerneffekte zu verhindern. Die Analyse der 38 durchgeführten Ereignisse identifizierte Basisszenario, Ereignisparameter, Handlungsanweisung und Ereignisbeschreibung als Werkzeuge für die Modellierung. Der Modellierungsprozess muss diese Werkzeuge verknüpfen, um die Anforderungen zu erfüllen.



**Abbildung 5.** Modellierungsprozess für komplexe Ereignisse

Die **Abbildung 5** zeigt den abgeleiteten Modellierungsprozess. Am Anfang steht der Bedarf, dass neue System oder Verfahren mit einer Störung zu konfrontieren. Entsprechend der Kategorie werden dann Ereignisparameter definiert oder die Handlungsanweisungen für die Pseudoakteure verfasst. Nach

der Integration in ein Basisszenario muss die Ereignisbeschreibung definiert werden. Hier ist es wichtig, für jede Versuchsbedingung unterschiedliche Ereignisbeschreibungen zu generieren, um Übungs- und Ermüdungseffekten vorzubeugen. Der Realitätsgrad kann hier erhöht werden, in dem auf Ereignisbeschreibungen zurückgegriffen wird, die real stattgefunden haben. Nach dieser Definitionsphase muss in Simulationsläufen mit den Pseudoakteuren überprüft werden, ob die gewünschte Auswirkung eintritt. Abschließend ist eine Abnahme durch einen operationellen Akteur (z.B. Pilot oder Lotse) von großer Bedeutung, um die Realitätsnähe der Simulation zu sichern.

## 5 Diskussion

### 5.1 Anwendung im Flughafenmanagement

Im Rahmen des europäischen Forschungsprogramm SESAR wurde untersucht, ob ein Flughafenleitstand die Kooperation im Flughafenmanagement verbessert (vgl. [11]). Hierzu führte das DLR eine HITL-Simulation (vgl. [12]) durch, bei der das Flughafenmanagement mit drei verschiedenen Ereignissen konfrontiert wurde. Die Reaktion auf die Ereignisse musste jeweils einmal in separaten Räumen und einmal in einem gemeinsamen Leitstand erarbeitet werden. Alle drei Ereignisse stellten Kapazitätsreduktionen dar. Sie wurden über die Konfiguration in die Simulation gebracht. Vorab informierte der Supervisor die Akteure über die Ereignisbeschreibung und erweiterte diese immer wieder, wenn Änderungen auftraten (z.B. Abschwächung / Verstärkung des Gewitters)



**Abbildung 6.** Simulation des Flughafenmanagements Amsterdam

In der kritischen Betrachtung der Simulation kann festgestellt werden, dass alle Ereignisse eintraten und von den Akteuren als realistisch wahrgenommen wurden. Anzeichen für Ermüdungs- oder Übungseffekte konnten nicht gefunden werden. Viel Aufwand musste aber in die Überprüfung der Auswirkung investiert werden. Selbst kleine Änderungen an den Ereignissen mussten mit einer vollständigen Simulation auf ihre Auswirkung überprüft werden. Da insgesamt sechs Flugliniendisponenten, zwei Flughafenbetriebsleiter und ein Fluglotse für einen Testlauf nötig waren, war der Personalaufwand sehr hoch.

## 5.2 Anwendung im Tower-Simulator

Im Rahmen einer Infrastrukturoptimierung wurde das DLR von einem Flughafen beauftragt, einen neuen Rollweg auf seine operationelle Machbarkeit zu überprüfen. Im Verlauf der Vorortanalysen wurde klar, dass die Lotsen in Bezug auf die besondere operationelle Machbarkeit Schwierigkeiten sehen und diese im Simulator überprüfen wollen.



**Abbildung 7.** Simulation im DLR Tower-Simulator (vgl. [13])

Insgesamt wurden 13 Ereignisse definiert. Vier Ereignisse betrafen eine reduzierte Verfügbarkeit von Rollwegen und Standplätzen. Neun Ereignisse stellten Abweichungen von Verfahren dar. Die Pseudo-Akteure (hier: Pseudo-Piloten) erhielten entsprechende Handlungsanweisungen. Die Lotsen wurden vorab durch den Simulationsleiter mit der Ereignisbeschreibung konfrontiert.

In der Simulation konnten bis auf zwei Ereignisse alle realisiert werden und wurden von den Lotsen als realitätsnah wahrgenommen. Vorab mussten die Pseudopiloten aber über einen Zeitraum von mehreren Monaten trainiert werden, um die Ereignisse korrekt darzustellen. Schwierig hierbei

war vor allem die klare Beschreibung der Handlungsanweisungen, so dass die Pseudo-Piloten sie ohne mehrere Trainingsläufe umsetzen konnten. Zusätzlich wurden die Ereignisse während der Pseudo-Pilotentrainings immer wieder auf die gewünschten Auswirkungen adaptiert.

## 6 Zusammenfassung und Ausblick

Besondere Ereignisse können in einem Flughafensystem, das an der Kapazitätsgrenze betrieben wird, große Auswirkungen haben. Aus diesem Grund müssen neue Systeme und Verfahren auf ihre besondere operationelle Machbarkeit überprüft werden. Das DLR nimmt solche Validierungen in seinen HITL-Simulatoren vor. Die Herausforderung liegt hierbei in der Modellierung der besonderen Ereignisse. Die Ereignisse müssen realitätsnah, in ihrer Auswirkung gleich, aber in ihrer Abbildung unterschiedlich dargestellt werden.

Eine Analyse von 38 besonderen Ereignissen aus den Simulationen der letzten acht Jahre zeigt, dass sich die Ereignisse im Wesentlichen drei Kategorien zuordnen und über die Werkzeuge Basisszenario, Ereignisparameter und Handlungsanweisung darstellen lassen. Die zusätzliche Ereignisbeschreibung für die Akteure hat eine besondere Bedeutung. Sie ist nicht eindeutig mit dem Ereignis verknüpft und kann so durch Variationen Übungs- oder Ermüdungseffekten entgegenwirken. Die Werkzeuge wurden zu einem Modellierungsprozess verknüpft, der die oben genannten Anforderungen an die besonderen Ereignisse erfüllen soll.

Der Modellierungsprozess wurde in jeweils einem Projekt in der DLR Flughafenmanagement-Simulation und dem DLR Tower-Simulator angewendet. Hierbei konnten die Ereignisse vollständig und realitätsnah dargestellt werden. Im Bereich der Auswirkungsabschätzung und im Bereich des Trainings der Pseudoakteure wurden noch Optimierungspotentiale festgestellt.

Die Abschätzung der Ereignisauswirkung könnte durch eine Softwareunterstützung verbessert werden. Hierzu müsste die Software an Hand des Basisszenarios Kapazitätsreduktionen, Abweichung von Verfahren und Flüge mit Priorität berücksichtigen können. Die Auswahl, wann und mit welcher Stärke ein Ereignis modelliert wird, kann so erheblich vereinfacht werden.

Diese Methode kann auch den Trainingsaufwand für die Pseudoakteure deutlich verringern. Im Rahmen der Tower-Simulation wurde festgestellt, dass insbesondere die kontinuierlichen Anpassungen der Ereignisse im Verlauf des Trainings die Pseudoakteure verunsichert haben. Eine Trennung von Training und Auswirkungsabschätzung im Prozessmodell ist also notwendig. Darüber hinaus könnte eine vereinheitlichte, simulationsunabhängige Handlungsanweisung den Pseudo-Akteuren helfen sich schnell auf ein neues Ereignis einzustellen, Änderungen zu erkennen und diese umzusetzen.

Abschließend kann gesagt werden, dass sich der Prozess zur Modellierung komplexer Flughafenereignisse in HITL-Simulationen bewährt hat. In diesem Zusammenhang kann er nicht nur in Forschungssimulationen und Machbarkeitsstudien angewendet werden, sondern auch andere Einsatzarten von HITL-Simulationen unterstützen. Insbesondere in Trainingssimulationen, die häufig komplexe Ereignisse darstellen müssen, kann der Modellierungsprozess die Simulationserstellung vereinfachen.

## 7 References

- [1] European Organization for the Safety of Air Navigation (EUROCONTROL). *Annual Network Operations Report 2015*. version 1.0, Brussels, Belgium, März 2016
- [2] European Organization for the Safety of Air Navigation (EUROCONTROL). *Challenges of Growth 2013: Task 4 European Air Traffic in 2035*. Bussels, Belgium, Juni 2013
- [3] M. Schultz, S. Kaltenhäuser, Sven und M. Biella. *Human-in the-Loop Simulation in ATM*. International Conference on Research in Air Transportation, ICRAT 2016, Philadelphia, Pennsylvania, USA
- [4] D. J. Folds. *Human in the Loop Simulation*. In: *Modeling and Simulation in the Systems Engineering Life Cycle*. Springer London, 2015. S. 175-183.
- [5] N. Makins. *EUROPEAN Operational Concept Validation Methodology (EOCVM)*. EUROCONTROL, Version 3.0, Brussels, Belgium, 2005
- [6] G. I. M. Young. *Feasibility studies*. Appraisal Journal 38.3, p. 376-383, 1970
- [7] S. Narayanan, P. Kidambi. *Interactive Simulations: History, Fetures, and Trends*. In: "Human-in-the-Loop Simulations: Methods and Practice", Editors: L. Rothrock and S. Narayanan, ISBN: 978-0857298829, Springer Verlag, 2011
- [8] C. Dow und J. Histon. *An Air Traffic Control Simulation Fidelity Definition and Categorization System*. Proceedings of the Human Factors and Ergonomics Society 58th Annual Meeting, 2014
- [9] W. Hager. *Grundlagen einer Versuchsplanung zur Prüfung empirischer Hypothesen in der Psychologie*. In: G. Lüer. *Allgemeine experimentelle Psychologie*, 1987
- [10] B. J. Winer, D. R. Brown, und K. M. Michels. *Statistical principles in experimental design*. Vol. 2. New York, USA, McGraw-Hill, 1971.
- [11] F. Piekert, S. Schier, A. Marsden, N. Carstengerdes, und R. Suikat. *A high-fidelity artificial airport environment for SESAR APOC validation experiments*, Air Transport Research Society World Conference 2015, Singapore, 2015
- [12] S. Schier, F. Timmermann und T. Pett. *Airport Management in the box – A human in the loop simulation for ACDM and airport management*. Deutscher Luft- und Raumfahrt Kongress 2016, 13.-15. Sept. 2016, Braunschweig, Deutschland
- [13] S. Schier und S. Kaltenhäuser. *Validierungsumgebung Vorfeld- und Flugplatzkontrolle: State of the Art und Blick in die Zukunft*. Deutscher Luft- und Raumfahrtkongress 2014, 16.-18. September 2014, Augsburg

# Ein Flug durch Raum und Zeit – Entwurf und Evaluation einer Simulationsdynamik für das Flughafenmanagement

Sebastian Schier<sup>1</sup>, Yves Günther<sup>1</sup>, Sandro Lorenz<sup>1</sup>, Reiner Suikat<sup>1</sup> und Florian Piekert<sup>1</sup>

<sup>1</sup> Deutsches Zentrum für Luft- und Raumfahrt, Braunschweig

*sebastian.schier@dlr.de*

Der Luftverkehr in Europa steht vor einer Herausforderung: Die Verspätungen an den Flughäfen nehmen zu. Steigende Verkehrszahlen und beschränkte Ausbaumöglichkeiten werden dieses Problem in der Zukunft verstärken. Nur durch eine enge Abstimmung aller beteiligten Akteure (Fluglinien, Bodenabfertiger, Flugsicherung und Flughafenbetrieb) kann sichergestellt werden, dass die vorhandenen Ressourcen ausgenutzt werden. Hierfür hat EUROCONTROL das so genannte „Airport Collaborative Decision Making“ (ACDM) definiert. Nach festgelegten Prozessen werden Abflugzeiten vergeben und Ressourcen möglichst effizient verteilt. Aktuell werden auf der Basis von ACDM zahlreiche Optimierungsmöglichkeiten erforscht. Eine Plattform, auf der die neuen Systeme und Verfahren entwickelt, validiert und geschult werden können existiert aber nicht.

Das Institut für Flugführung des Deutschen Zentrums für Luft- und Raumfahrt (DLR) erforscht und validiert neue Systeme und Verfahren für den Luftverkehr. Seit den 1950er Jahren entwickelt und optimiert das DLR Human-In-The-Loop Simulationen (HITL) als ein Werkzeug für die Forschung. Für die Forschungs- und Trainingsvorhaben im Bereich Flughafenmanagement hat das DLR eine neue HITL-Simulation entworfen und implementiert. Das vorliegende Paper beschreibt wie das DLR die zugehörige Simulationsdynamik umgesetzt hat. Ausgehend von den aktuellen Herausforderungen im Luftverkehr wird dargestellt, wie auf Basis des ACDM-Standards eine Prozesssimulation mit Ressourcen- und Ereignismodell entwickelt und validiert wurde. Abschließend wird die Anwendung in einem Forschungsprojekt dargestellt und ein Ausblick auf zukünftige Entwicklungen gegeben.

## 1 Einleitung

Das europäische Luftverkehrssystem steht vor einer großen Herausforderung. Die Gesamtverspätung steigt seit 2013 moderat, der Anteil der am Flughafen generierten Verspätung nimmt aber einen immer größeren Anteil ein (vgl. [1]). In der letzten Auswertungsperiode von 2014 auf 2015 sind die Verspätungen am Flughafen um 60% gestiegen. Diese Entwicklung soll sich laut einem Bericht der europäischen Organisation zur Sicherung der Luftfahrt EUROCONTROL sogar noch verstärken (vgl. [2]).

Zahlreiche europäische Forschungsprojekte versuchen dieser Herausforderung mit verbesserten Kooperationsmethoden, Planungssystemen, Personalauswahl und Training zu begegnen (z.B. [3], [4], [5] und [6]). Zur Entwicklung und Validierung der Forschungsideen wird eine Plattform benötigt, auf der die Flughafenprozesse dargestellt werden können. Eine Validierung vor Ort auf den Flughäfen ist auf Grund des Risikos (Eingriff in den Luftverkehr), des Aufwands (Zugriff auf operationelle Systeme) und

dem fehlenden Einfluss auf die Randbedingungen (z.B. Wetter) nur bedingt sinnvoll. Die europäische Validierungsrichtlinie schlägt für diesen Fall Human-In-The-Loop Simulationen vor (Abk. HITL, vgl. [7]).

Das DLR hat aus diesem Grund eine Flughafenmanagement-Simulation entwickelt. Diese besteht aus einer zentralen Flughafen-Datenbank, Benutzerschnittstellen (vgl. [8]) und einer Simulationsdynamik. Die Simulationsdynamik wird hier als die zentrale Funktion zur Berechnung der Flugbewegungen verstanden. Das vorliegende Paper beschreibt die Entwicklung und Validierung der Simulationsdynamik. Dazu wird zuerst betrachtet, welche Anforderungen an eine Simulationsdynamik für das Flughafenmanagement gestellt werden. Danach wird ein geeignetes Modell ausgewählt und auf die Flughafenprozesse adaptiert. Abschließend erfolgen eine Validierung des Modells und ein Ausblick auf zukünftige Entwicklungen.

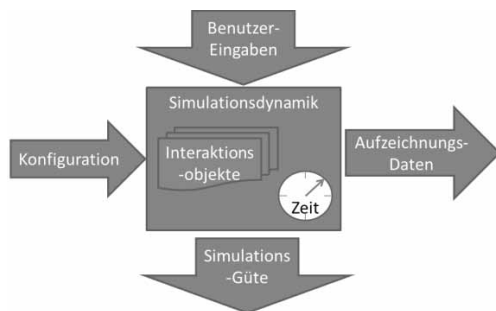
## 2 Theorie

### 2.1 Human-In-the-Loop Simulation

HITL-Simulationen werden in einer Vielzahl von Anwendungsbereichen verwendet und besitzen dementsprechend verschiedene Funktionen (vgl. [9]). Unabhängig vom Flughafenmanagement soll zunächst dargestellt werden, welche generischen Funktionen eine HITL-Simulation beinhalten muss. Hierfür wird sich auf die Definition von Narayan und Kidambi (vgl. [10]) gestützt. Sie definieren HITL-Simulationen als eine Verbindung der folgenden Funktionalitäten:

1. Abbildung von Interaktionsobjekten in einer Simulationsdynamik.
2. Abbildung der Zeit als dynamisches Element in der Simulationsdynamik.
3. Anpassungsmöglichkeit von Simulationsparametern und -Dynamik.
4. Schnittstellen über die Benutzer mit der Simulationsdynamik interagieren.
5. Aufzeichnung von Simulationsgütekriterien.
6. Methoden zur Analyse der Simulationsdaten und arbeitswissenschaftlicher Aspekte.

Die ersten beiden Funktionalitäten definieren den Inhalt der Simulationsdynamik, während drei und vier der Eingabe in die Simulationsdynamik entsprechen. Szenarien und Einstellungen werden vor dem Simulationslauf eingegeben (drei) und die Benutzereingaben erfolgen zur Laufzeit (vier). Die letzten beiden Funktionalitäten stellen die Ausgabe dar. Die Simulationsdynamik muss Gütekriterien ebenso wie Daten zu den Interaktionsobjekten aufzeichnen, um eine spätere Auswertung dieser zu ermöglichen. Schematisch sind diese Zusammenhänge in der **Abbildung 1** dargestellt.



**Abbildung 1.** Schematische Darstellung der Simulationsdynamik

### 2.2 Flughafenmanagement

Das Flughafenmanagement ist eine gemeinsame Entscheidungsebene der an den Flughafenprozessen beteiligten Akteure (z.B. Fluglinien, Flugsicherung, Flughafenbetreiber). Unter diesen Akteuren wird beraten, wie mit den vorhandenen Ressourcen (z.B. Standplätze oder Start-/Landebahnen) verfahren wird. Die Prozesse und Verantwortlichkeiten können sich hier von Flughafen zu Flughafen zum Teil stark unterscheiden. Die Fluglinien an US-Flughäfen sind in der Regel die Betreiber der Terminals und bestimmen, welches Luftfahrzeug welchen Standplatz belegt. In Europa übernimmt der Flughafenbetreiber diese Aufgabe (vgl. [11]).

Die DLR Flughafenmanagement Simulation soll vorrangig europäische Flughäfen darstellen. Für das europäische Flughafenmanagement wurde das so genannte „Airport Collaborative Decision Making“ (Abk. ACDM) definiert. ACDM beinhaltet die folgenden grundlegenden Elemente (vgl. [12]):

- **Informationsverteilung:** ACDM definiert eine Basissammlung an Flugdaten, zu der alle Beteiligten beitragen und auf die alle Beteiligten Zugriff haben (z.B. geplante und tatsächliche Startzeit).
- **Meilensteinansatz:** ACDM betrachtet jeden Flug als eine Abfolge von Prozessen, deren Ende jeweils mit einem Meilenstein gekennzeichnet ist (z.B. Rollen zum Start mit dem Meilenstein Abheben).
- **Variable Rollzeiten:** ACDM schreibt vor, dass die Rollzeiten abhängig von Start/Landebahn und Standplatz in die Berechnung der Flugzeit zu integrieren sind.
- **Vorsequenzierung der Abflüge:** ACDM definiert einen Prozess, über den die Abflugreihenfolge vor Verlassen des Standplatzes festgelegt wird. Statt mit an der Startbahn zu warten, werden die Triebwerke erst gestartet und der Standplatz verlassen, wenn ein Start ohne größere Wartezeit möglich ist.
- **Flugplan-Aktualisierungen:** Alle Änderungen am Flugplan müssen an das Europäische Netzwerkmanagement weitergegeben werden, um andere Flughäfen und die verschiedenen Flugsicherungsstellen vorab zu informieren.

Die Flugplan-Aktualisierung und die Vorsequenzierung der Anflüge müssen nicht von der Simulationsdynamik generiert werden. Sie werden über Daten-Schnittstellen (Flugplan-Aktualisierung), bzw. über die Benutzerschnittstelle der Flugsicherung (vgl. [13]) realisiert. Die Informationsverteilung betrifft eher die bereits existierende Flughafen-Datenbank und gibt deren Design vor. Folglich muss die Simulationsdynamik im Wesentlichen alle Meilensteine generieren können und dabei die variablen Rollzeiten auf dem Flughafen berücksichtigen.

### 3 Methode

Die Umsetzung der Simulationsdynamik kann nur erfolgen, wenn vorab ein geeignetes Simulationsmodell ausgewählt wird. Für den Luftverkehr existieren eine Vielzahl an Modellen (z.B. [14] und [15]). Diese Modelle konzentrieren sich auf unterschiedliche Abschnitte eines Fluges und stellen diesen in unterschiedlichem Detailgrad dar.

Das benötigte Modell hängt dabei stark vom untersuchten System oder Verfahren ab. Ein neues Anflugplanungssystem benötigt eine detaillierte Darstellung des aerodynamischen Luftfahrzeug-Verhaltens, während ein neues Standplatzvergabesystem eher Abfertigungsprozesse abbilden muss. Die zu untersuchenden Systeme und Verfahren für die Flughafenmanagement-Simulation werden erst in zukünftigen Forschungsprojekten entwickelt. Fest steht nur, dass nach ACDM ein Flug vom Start am Ausgangsflughafen bis zum Start auf dem eigenen Flughafen betrachtet wird. Die Abbildung muss mindestens entsprechend der ACDM-Meilensteine erfolgen. Diese werden mit einer Minutengranularität gesetzt.

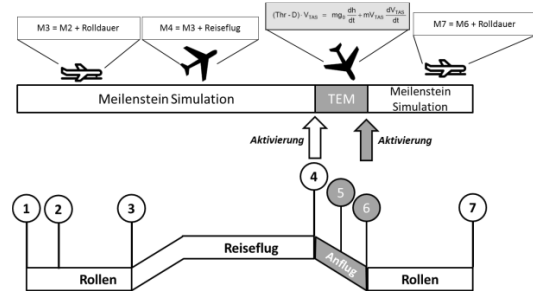
Aus diesem Grund wurde für die ACDM-Simulation beschlossen, einen flexiblen Ansatz zu wählen: Ein grundlegendes Modell soll die ACDM-Mindestanforderungen erfüllen. Wird eine detaillierte Darstellung benötigt, sollen Teilbereiche der Simulation an andere Modelle delegiert werden können.

## 4 Ergebnisse

### 4.1 Meilensteinmodell

Der ACDM-Standard beschreibt implizit ein Prozessmodell. Die Flüge sind die Interaktionsobjekte und werden ausschließlich als Prozess über die Zeit

wahrgenommen. Folglich wurde für die grundlegende Simulationsdynamik das ACDM-Prozessmodell umgesetzt. Dieses „Meilensteinmodell“ kann vorab mit Flugplänen, Roll- und Standardflugzeiten konfiguriert werden. Mit Start der Simulation werden die ACDM-Meilensteine gesetzt. Wird der Zustand eines Fluges (z.B. über eine Benutzerschnittstelle) verändert, so wird diese Änderung in den Berechnungen berücksichtigt.



**Abbildung 2.** Übergabe vom Meilensteinmodell (weiß) an das Totale Energien Modell ([14], grau)

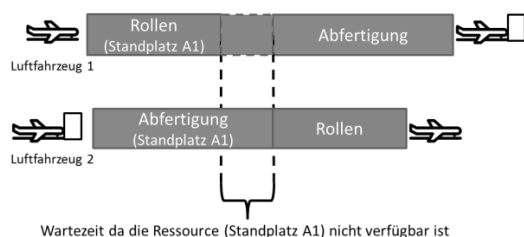
Wie gefordert wurde das Meilensteinmodell mit einer Möglichkeit versehen Teilprozesse an andere Simulationen zu delegieren. Die Übergabe erfolgt jeweils an einem Meilenstein. Der Meilenstein dient dabei als Synchronisationszeitpunkt. Die **Abbildung 2** zeigt dieses Verfahren exemplarisch. Die Meilensteinsimulation bildet den Flug bis zum Ende des Reiseflugs ab (Meilenstein vier). Die Zeit des Meilenstein vier wird als Startzeitpunkt an das detaillierte Simulationsmodell (hier: Totales Energien Modell) übermittelt. Dieses Modell übernimmt die Simulation bis zum Aufsetzen (Meilenstein sechs) und übergibt hier zurück an das Meilensteinmodell. So werden der Reiseflug und das Rollen am Boden eindimensional als Prozess über die Zeit dargestellt, während der Anflug vierdimensional in Raum und Zeit simuliert wird.

### 4.2 Ressourcenmodell

Die Abbildung der Meilensteine und der dahinter liegenden Prozesse stellen den Flugverlauf vollständig aber noch nicht realitätsgetreu dar. Es wird missachtet, dass die Ressourcen des Flughafens (z.B. Standplätze oder Start/Landebahnen) endlich sind. Überschreitet die Nachfrage die Kapazität einer Ressource kann es zu Wartezeiten kommen.

Die **Abbildung 3** zeigt an Hand der Standplatzbelegung, wie das Ressourcenmodell der Meilensteinsimulation diesen Aspekt umsetzt. Das Ressourcenmodell überprüft ob eine Ressource zu

dem Zeitpunkt an dem der Meilenstein gesetzt werden soll verfügbar ist. Ist die Ressource durch ein anderes Luftfahrzeug besetzt, so wird der Meilenstein verzögert. Im Beispiel des belegten Standplatzes wartet das Luftfahrzeug also so lange, bis die Abfertigung des blockierenden Luftfahrzeugs abgeschlossen ist.



**Abbildung 3.** Ressourcenmodell: Besetzter Standplatz

Das Ressourcenmodell wurde für den Endanflug, für die Start- und Landebahnen und die Standplätze in die Simulationsdynamik integriert.

### 4.3 Ereignismodell

Meilenstein- und Ressourcenmodell stellen die Flugverläufe bereits valide dar und können so einen Tagesflugplan simulieren. Ein wichtiger Faktor im Flughafenmanagement sind aber Störungen, die sich zusätzlich auf den Verkehr auswirken. In diesem Fall soll das Flughafenmanagement aktiv werden und Ressourcen neu verhandeln. Bereits heute, aber insbesondere für die Zukunft werden diese Störungen die Flughäfen intensiver beanspruchen (vgl. [2]).

Die Simulationsdynamik muss in der Lage sein diese Ereignisse darzustellen. Die Herausforderung ist hier die große Bandbreite möglicher Störungen (z.B. Gewitter, Streik, Sicherheitsbedrohung). Alle Störungen stellen sich auf der Prozessebene aber in der gleichen Weise dar: Sie verzögern einen oder mehrere Prozesse.

Das Ereignismodell verfährt aus diesem Grund ähnlich wie das Ressourcenmodell: Es prüft, ob ein bestimmter Meilenstein von einer Störung betroffen ist und verzögert dann entsprechend den Meilenstein.

### 4.4 Datenaufzeichnung und Simulationsgüte

Die Simulationsdynamik wurde zusätzlich mit einer Datenaufzeichnung ausgestattet. Diese sichert alle relevanten Flugdaten für später durchzuführende Auswertungen. Des Weiteren berechnet und speichert die Datenaufzeichnung Gütekriterien für die Simulation.

Als Gütekriterium für die Simulation wurden drei Werte definiert:

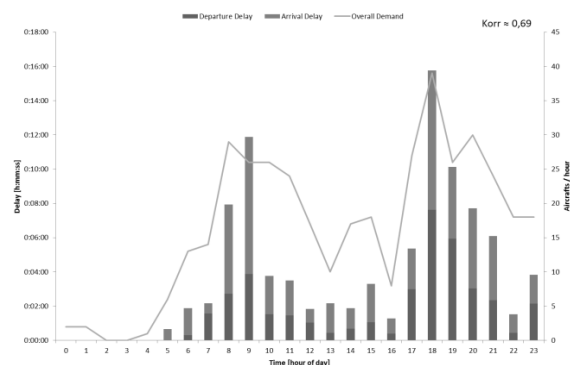
- Rechendauer pro Zeitschritt
- Verhältnis Realzeit zu Simulationszeit
- Speicherverbrauch

## 5 Diskussion

Eine vollständige Simulationsdynamik für das Flughafenmanagement bildet nach den in Kapitel zwei erhobenen Anforderungen die Flüge als Interaktionsobjekte über die Zeit ab. Zusätzlich muss die Simulationsdynamik (z.B. in Bezug auf die Rollzeiten) konfigurierbar sein und Eingaben der ACDM-Akteure verarbeiten können. Des Weiteren muss die Aufzeichnung von Simulationsdaten und Gütekriterien möglich sein. Diese Anforderungen werden von der Meilensteinsimulation in Verbindung mit Ressourcen- und Ereignismodell erfüllt.

Eine Validierung soll zeigen, dass sich das erstellte Modell für den beabsichtigten Einsatzbereich eignet. Hierfür wurde in einem ersten Schritt ein Referenzszenario der „Airport Simulation and Performance Assessment Group“ (Abk. ASPAG, vgl. [16]) simuliert. Die Simulation wurde ohne Ressourcen- oder Ereignismodell durchgeführt. An Hand der aufgezeichneten Daten wurde überprüft, ob die Meilensteine in der logisch richtigen Reihenfolge und mit dem vordefinierten Abstand gesetzt werden. Darüber hinaus wurden die Werte mit den Erwartungswerten aus dem Flugplan verglichen. Hier zeigte sich keine Abweichung.

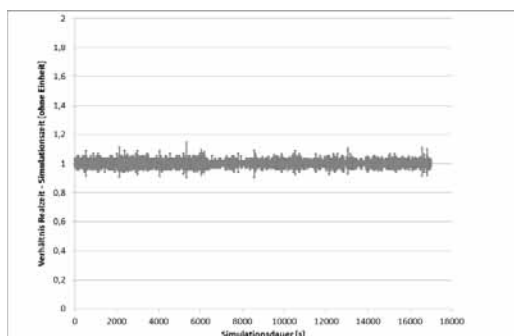
Als nächstes wurde das Ressourcenmodell überprüft. Hierzu wurde der grundlegende Zusammenhang genutzt, dass sich mit geringerer Ressourcen-Kapazität mehr Wartezeiten der Luftfahrzeuge ergeben.



**Abbildung 4.** Verlauf von Verkehr und Verspätung.

Auf Basis dieser Hypothese wurden fünf Szenarien mit geringer werdenden Kapazitäten im Bereich der Start/Landebahn untersucht. **Abbildung 4** zeigt exemplarisch das Ergebnis einer Berechnung mit einer Start/Landebahnpazität von 37 Luftfahrzeugen pro Stunde. Hierbei ergab sich eine Korrelation von 0.69 zwischen Durchsatz und Wartezeit. Es kann also in diesem Beispiel auf einen hohen Zusammenhang von Wartezeit und Durchsatz geschlossen werden. In den anderen Szenarien ergab sich das erwartete Verhältnis: Je geringer die Ressourcen, desto mehr korrelieren Wartezeit und Durchsatz.

Abschließend wurde die Simulationsgüte untersucht. Die Simulation wurde hierzu auf ein Berechnungsintervall von einer Sekunde konfiguriert und der Tagesflugplan eines großen europäischen Flughafen mit über 1200 Flügen geladen und über eine Zeitdauer von fünf Stunden simuliert. Das Verhältnis Realzeit zu Simulationszeit sollte dabei im Mittel nicht von eins abweichen. Speicherverbrauch und Rechenzeit sollten konstant sein und nicht mit der Zeit korrelieren.



**Abbildung 5.** Verhältnis Real- zu Simulationszeit

Die **Abbildung 5** zeigt, dass das Verhältnis von Real- zu Echtzeit in geringem Maße um den Wert eins pendelt. Eine konstante Verschiebung des Verhältnisses ist nicht zu beobachten. Dies wird auch durch die Berechnungsdauer bestätigt. Sie ist im Mittel 37ms groß mit einer Standardabweichung von 11ms. Mit 141ms liegt die maximale Berechnungsdauer weit unter der verfügbaren Sekunde Realzeit. Der Speicherverbrauch zeigt ebenfalls stabile Tendenzen mit einem Mittelwert von 13Mb. Er korreliert nicht mit der Zeit (Korr. 0,1).

## 6 Fazit und Ausblick

Die Entwicklung, Validierung und Schulung neuer Systeme und Verfahren für das Flughafenmanagement benötigt eine HITL-Simulationsplattform. Als ein Bestandteil der Plattform wurde die hier beschriebene Simulationsdynamik umgesetzt. Alle erhobenen Anforderungen wurden erfüllt. Die Flüge werden als Prozess über die Zeit dargestellt, können durch die Übergabe an andere Modelle aber auch im dreidimensionalen Raum berechnet werden. Die durchgeführte Validierung bestätigt die Güte der generierten Daten. Angesichts der stabilen und geringen Berechnungsdauer kann davon ausgegangen werden, dass die Simulationsdynamik nicht nur Echtzeit-, sondern auch Schnellzeitfähig ist. Zusammen mit den Benutzerschnittstellen und der zentralen Datenbank bildet die Simulationsdynamik so eine flexible Entwicklungs- und Validierungsplattform für das Flughafenmanagement.



**Abbildung 6.** DLR HITL-Simulation des Flughafenmanagements

Im Rahmen einer Validierung wurde die neue Flughafenmanagement-Simulation bereits eingesetzt (siehe **Abbildung 6**). Hier sollten die Vorteile eines gemeinsamen Flughafenleitstands und unterstützender Systeme analysiert werden (vgl. [17]). Drei Fluglinien, die Bodenabfertigung, die Flugsicherung und der Flughafenbetreiber wurden mit verschiedenen Ereignissen konfrontiert und mussten den Verkehr unter den Ereigniseinflüssen möglichst effizient abarbeiten. Hierzu wurde die Meilensteinsimulation mit einem realen Flugplan eines europäischen Flughafens konfiguriert. Die Simulation des Anflug- und Rollverkehrs erfolgte nach einer Übergabe im Endanflug durch den Simulator NARSIM (vgl. [18]) der auf dem totalen Energien Modell (vgl. [14]) basiert.

In der Zukunft soll die Flughafenmanagement-Simulation auch dazu eingesetzt werden die Zusammenarbeit mehrerer Flughäfen darzustellen. Zu diesem Zweck ist es geplant die Meilenstein-Simulation so zu erweitern, dass die Prozesse an anderen Flughäfen ebenso abgebildet werden. Des Weiteren soll der Einfluss des europäischen Netzwerkmanagements für den Luftverkehr (z.B. durch Sektoren mit limitierter Kapazität) dargestellt werden.

## 7 Literatur

- [1] European Organization for the Safety of Air Navigation *Annual Network Operations Report 2015*. Version 1.0, Brussels, Belgium, März 2016
- [2] European Organization for the Safety of Air Navigation. *Challenges of Growth 2013: Task 4 European Air Traffic in 2035*. Brüssel, Belgien, Juni 2013
- [3] D. Schulze-Kissing, H. Eißfeldt. *ConCent: Eine Simulationsplattform zur Untersuchung kollaborativer Entscheidungsprozesse in Leitzentralen*. In: DGLR-Bericht 2015-01, S. 157-170. 57. FAS Anthropotechnik, Nov. 2015, Rostock. ISBN 978-3-932182-83-9
- [4] K.-H. Keller, Y. Günther, M. Schaper und F. Piekert. *Total Airport Management - A holistic approach towards airport operations optimization*. Séminaire d'anglais professionnelle, Toulouse, France, 2011
- [5] A. Papenfuss, N. Carstengerdes und Y. Günther. *Konzept zur Kooperation in Flughafen-Leitständen*. 57. FAS DGLR L6.4 Anthropotechnik, Rostock, 2015
- [6] S. Helm, S. Loth, und M. Schultz. *Advancing Total Airport Management – An Introduction of Performance Based Management in the Airport Context*. 19th ATRS World Conference, Singapore, 2015
- [7] N. Makins. *EUROPEAN Operational Concept Validation Methodology (EOCVM)*., EUROCONTROL, Version 3.0, Brussels, Belgium, 2005
- [8] S. Schier, T. Pett, O. Mohr, und S.J. Yeo *Design and Evaluation of User Interfaces for an Airport Management Simulation*. AIAA Modelling and Simulation Conference, Washington D.C., USA, June 2016.
- [9] D. J. Folds, *Human in the Loop Simulation*. In: Modeling and Simulation in the Systems Engineering Life Cycle. Springer London, 2015. S. 175-183.
- [10] S. Narayanan, P. Kidambi. *Interactive Simulations: History, Fetures, and Trends*. In: "Human-in-the-Loop Simulations: Methods and Practice", Editors: L. Rothrock and S. Narayanan, ISBN: 978-0857298829, Springer Verlag, 2011
- [11] Z. Zhu, N. Okuniek, I. Gerdes, S. Schier, H. Lee, und J.C. Jung. *Performance Evaluation of the Approaches and Algorithms for Hamburg Airport Operations*. 35th DASC, September 2016, Sacramento, CA, USA
- [12] EUROCONTROL. *Airport CDM Implementation – The Manual*, version 4.0, Brussels, Belgium, 2012
- [13] M. Schaper. *Operational Improvements in the Context of DMAN, A-SMGCS AND A-CDM*. In: CEAS Journal, Seiten 1-9. CEAS 2009, Manchester, UK.
- [14] A. Nuic. *User manual for the Base of Aircraft Data*, EUROCONTROL, Brussels, Belgium, Atmosphere 2010 (2010): 001.
- [15] P. Förster. *A discrete event modelling and simulation environment for applications in the TAM context*. 60. DLRK 2011, 27.-29. September 2011, Bremen
- [16] M. Schultz, S. Lorenz, F. Knabe, Y. Günther. *Airport in a Lab*. Airside Simulation and Performance Assessment Group, 2015, London
- [17] F. Piekert, S. Schier, A. Marsden, N. Carstengerdes, und R. Suikat. *A high-fidelity artificial airport environment for SESAR APOC validation experiments*. Air Transport Research Society World Conference 2015, Singapore, 2015
- [18] J. M. Ten Have. *The development of the NLR ATC Research Simulator (Narsim): Design philosophy and potential for ATM research*. Simulation Practice and Theory 1.1 (1993): 31-39.



## Design exploration process for aerospace industry

Author: Peter Hoffmann, Chiatek, [peter.hoffmann@chiatek.com](mailto:peter.hoffmann@chiatek.com)

### Introduction

It is not really surprising that design optimization is a more commonly used simulation method than design exploration as an optimal design is in many cases good enough, or?

Even though the outcome of a new design is unknown to the design engineer in the beginning, optimization methods are more frequently used approaches within engineering. This needs a radical change as time to market, risky designs and overall costs are too much of a pain. To accomplish the best design for a new aerospace part, system or aircraft, design exploration beyond co-simulation within extended enterprise processes is one of the best and intuitive future proof method to evolve and investigate multi-disciplinary design-spaces and –discoveries. As it is a much wider and open approach, it better supports the decision making throughout all of the design phases.

As it is impossible to formulate and characterize all problems before hand, let us explore all of them (known and unknown) during the research process. Anyhow it is the necessary step to constantly improve engineering efficiency, which do addresses the pain points of our industry today!

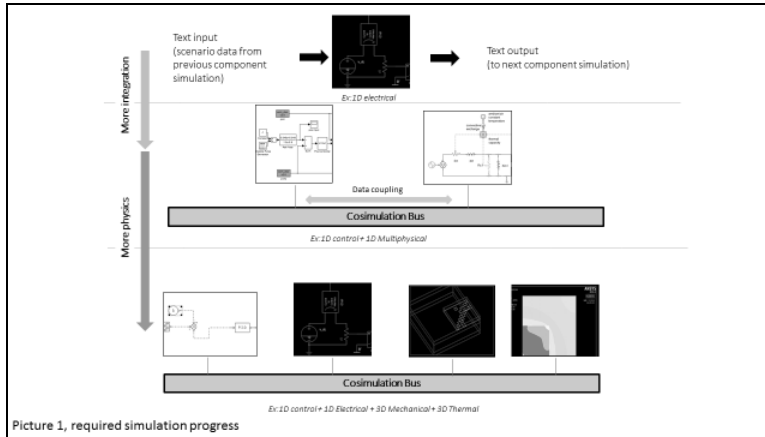
### Why is design exploration so important nowadays?

The simulation world has made progress in supporting increased complexity of products and processes. Even though there is still a long way to become true multi-domain & multi-physic co-simulation oriented.

The need to provide simulation environments for physical systems with interaction of all components and parts has significantly grown. By supporting a system made of multi-components and using multi-physics (mechanical, electrical, hydraulic, thermal, pneumatic) describes well what multi-domain & multi-physics is. To accomplish non monolithic solutions, which are required in a world of distributed components, electronics and networked software, sharing intellectual property (IP) with an interdisciplinary mindset needs to happen in research, design and engineering. The simulation strategy needs to change and adapt from:

- 1) simulation to co-simulation
- 2) mono-physics to multi-physics
- 3) design optimization to design exploration

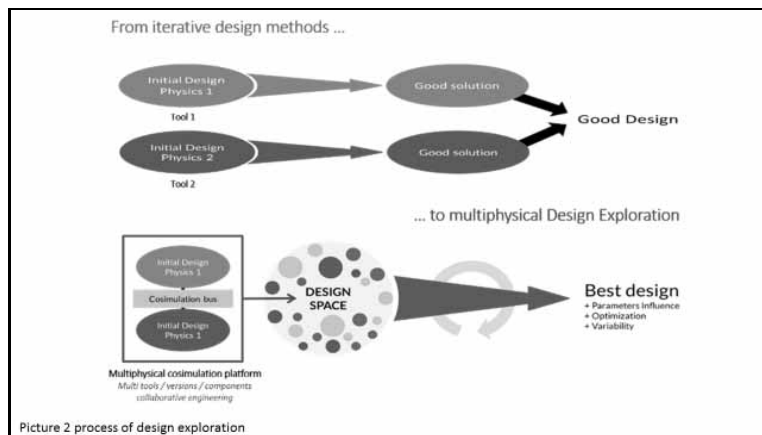
Following these 3 steps will make simulation stay tuned with industry requirements and will help to add significant value to engineering. Picture 1 illustrates the required simulation progress.



Specifically in the aerospace industry, where development phases are longer than anywhere else, latest available technologies are out dated before a new airplane hits the clouds. In using optimization methods as the basis for research & design approaches are wrong as those rely on known technologies, known components and known rules only. Therefore any optimization approach will ultimately end up with a compromise in design, inadequately addressing development steps big enough to accomplish competitive edges of tasks like weight reduction, aerodynamic buoyancy and energy efficiency.

As in all industries of mobility, less combustion and more electrification as well as less manual and more autonomous functions are daily engineering challenges. Not only that engineering products are more and more complex, the pressure for time to market, better reliability and performance as well as optimized maintenance costs are engineering guidelines to follow. We need a choice of alternative designs to choose from and that is where design exploration helps with.

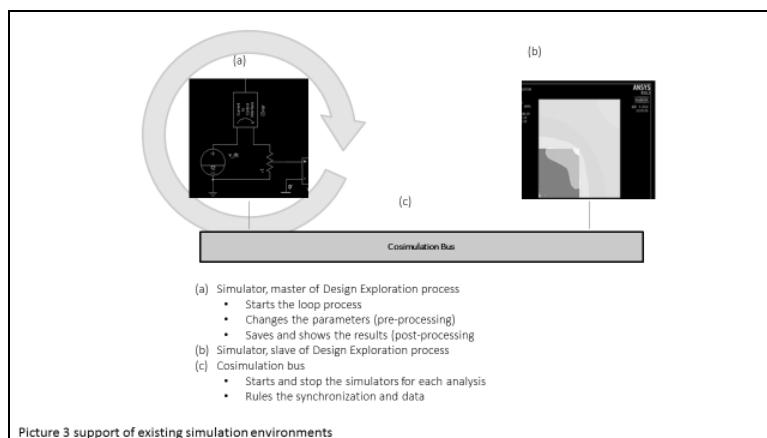
Today in research of new designs and under the engineering circumstances detailed above, the time is there to use design exploration within multi-physics co-simulation as it leads to more desirable solutions than design optimization does. The design exploration process shown in picture 2, as a formal or semi-formal method, is a gradual approaching method heading for the optimal best design. It is used to evolve and investigate multidisciplinary design space with the intent of design discovery. The strategies are based on the belief that the problem formulation evolves during the process of searching. A design space process creates parametric topologies followed by selecting varied parameters within associated ranges. A selection of new parameter configurations is used to explore the entire design space and the performance feedback to find the most promising areas.



The need for the best design of parts, systems and aircrafts is essential for a successful aerospace business. In being able to explore all problems (known and unknown) during the research & design process will make the outcome of engineering far more future proof and therefore much longer competitive. The need for competitive edges and best designs are accomplished herewith.

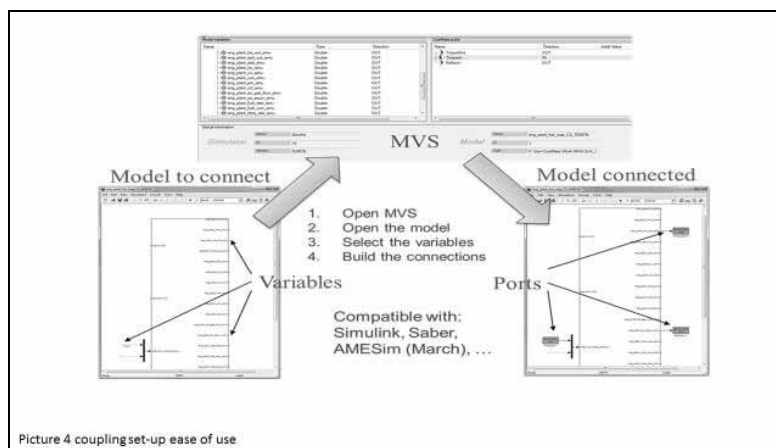
### What is needed and how it can be accomplished within existing engineering environments?

It is a commonly known that simulation needs to be the first step in all kind of development activities. It is also understood that engineers do have a preference for simulation tools they have used or familiar with. Furthermore it is convenient to use existing simulation models, as they have been a good basis for perfecting accuracy over time. Last but not least dedicated simulation tools are optimized for very specific tasks, physics and domains. In being able to choose the best matching tool out of a wide range of available simulators (new & historical once) is advantageous as it enforces constant tool improvements satisfying customer requirements and stimulates competition.



Those existing engineering environment, picture 3, need to be supported and unchanged as much as possible even though constant enhancements are required to stay tuned with existing engineering requirements as well as upcoming new challenges. It is our understanding to support all simulation parties, to enrich their system application capabilities and to enable future-oriented tool utilization thanks to native coupling.

Coupling simulators can be accomplished in many ways. For a real benefit of enabling virtual integration to assess an assembly consistence before any prototype or even computer added design (CAD) is available, dynamic data- and signal-exchange as well as control through native coupling supports it. Processes like design exploration can be accomplished. The required co-simulation architecture in mind needs to be a native based & open bus architecture, allowing to manage all kind of activities such as communication speeds, timed or unsynchronized traffics, and so on.



Required set-up of the bus communication exemplary drawn in picture 4 just means filling in parameter values into so called netlist template per simulator underlines ease of use. Once this is done, it manages everything fully automatically and acts as a background task. Adding more tools to the co-simulation bus is just repeating netlist value entrances accordingly. As soon as a netlist is completed, co-simulation is ready to use. At that point of time user can run more than one simulator at time to solve:

- 1) compatibility issues between outdated models and unsupported software versions
- 2) system-simulation across heterogeneous tools, teams and geographies
- 3) multi-physic simulation supporting design optimization
- 4) design exploration shooting for the best design

The common and heavily used method of traditional design optimization is easy to use as experts will concentrate on their area of expertise from a known solution. Needless to say that this is a widely used approach in all kind of design phases. It is an efficient approach even by taken into account multi-physical constrains through textual of static data, but does not take into account how other possible design variable combinations will impact the designed solution.

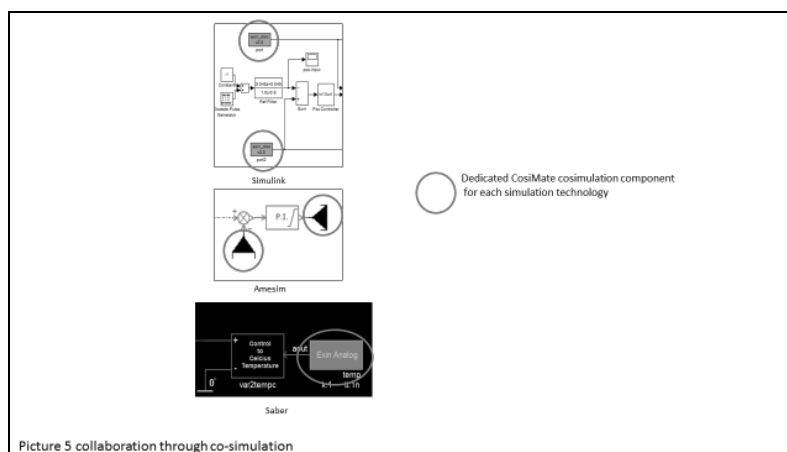
The real need to extend design of experiment has become so important due to the dramatic increase of networked electronics and coupled software driven by the fact of far more electrification and autonomy. This requires other simulation approaches and should therefore be based on multi-domain co-simulation. The simple example of fast current and slow heat in any power electronic design, requires at least 2 dedicated simulation tools natively coupled. Taking heat recovery, usage or dissipation additionally into consideration adds more simulation tools. This is not a conflict, it is a must for new designs. Optimization methods are inadequate addressing the complexity of networked hardware, software and applications. Furthermore they cannot quickly evaluate changes in design, reducing development costs and preventing late-stage design changes.

This is where design exploration goes far beyond and drives for perfection. It takes all combinations of parameters to evaluate the outcome on the design performance into real world scenarios.

### How it supports engineering tools familiarity already in place?

The foundation to ensure support of engineering tool familiarity already in place is the co-simulation bus architecture (Picture 4). The unique part of data exchange and control is native tool coupling, where no translation of simulation data and/or control data can mislead to inaccuracy or latency issues impacting performance. This simplistic approach requires permanent adaption of tools already implemented, new tools coming available and simulation workflows supported. Due to native data exchange and control, this unique bus architecture does not require to be the master instance running multi-domain or multi-physics co-simulation; obviously it can be, but it doesn't need to be.

It is structured to delegate master instances according the simulation workflow required. Therefore an engineer, who is per example an Amesim expert can run other simulation tools through his own application. The flexibility of master/slave operation can be even go beyond all of the simulation tools to management solutions such as product life cycle or any other very specific tool, which could run design exploration workflows automatically.



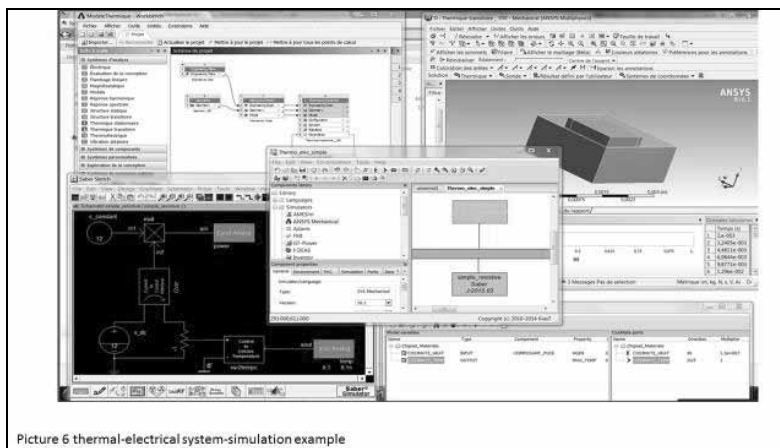
Picture 5 collaboration through co-simulation

Such an open & independent approach not only enriches any simulation tool available, it also enables needed collaboration between vendors, suppliers and all contributors involved in the process picture 4. Historically co-simulation has first taken place in aerospace industry, followed by early adaption of what if studies known as Monte Carlo analysis. With this background design exploration is the next logical step advancing simulation processes, unlocking knowledge and helping to support a next revolutionary step in aerospace design.

## Conclusion

Establishing a design exploration process as part of multi-physics co-simulation helps the engineer to find alternative designs and to evaluate system integration. Furthermore he can collaborate with colleagues and stay with his favorite platform.

The advantageous usage of design exploration as part of the research & development workflow within aerospace industry, targeting best designs, is out of question. Next generation aircrafts, components and systems cannot longer rely on optimized designs. The aerospace industry needs revolutionary ideas to stay in sync with the fast pace of economic changes, business trends and environmental restrictions. In the world of simulation, the aerospace industry is leading and will greatly benefit from their system simulation know-how already in place as picture 6 example shows.



This industry is very familiar with design optimization processes and therefore well prepared for a wide usage of design exploration. To address existing industry technology challenges like

- 1) new hypersonic solution
- 2) high energy efficiency
- 3) full electrification
- 4) wide autonomous functions
- 5) connected aircrafts

The need to virtualize an all-in-one solution of a next generation component, system or aircraft is there. We are just in the beginning of adapting design exploration processes and published use cases are rare. As it is impossible to formulate and characterize all problems before hand, let us explore all of them (known and unknown) during the research process.

# Optimaler Flug eines Drachenfliegers durch eine Wolkenstraße

Andreas Britzelmeier<sup>1</sup>, Anna-Lena Klingler<sup>2</sup>, Kurt Chudej<sup>3</sup>

<sup>1</sup>Fakultät für Luft- und Raumfahrttechnik, Universität der Bundeswehr München

<sup>2</sup>Frauenhofer-Institut für System- und Innovationsforschung ISI, Karlsruhe

<sup>3</sup>Lehrstuhl für Ingenieurmathematik, Universität Bayreuth

*kurt.chudej@uni-bayreuth.de*

Drachenfliegen bzw. Hängegleiten ist ein populärer Sport in den Alpen. Lange Flüge, die große Reichweiten erreichen, sind dank geeigneter Thermiken möglich. Wir präsentieren neue numerische Ergebnisse für den Reichweitenflug durch eine Wolkenstraße.

## 1 Einleitung

Drachenfliegen ist in den Alpen dank geeigneter Thermikwinde über Berge und Täler ein populärer Sport (siehe Abb. 1 und z.B. [1]). Die Flugbahnen enthalten üblicherweise spiralförmige Aufstiegsbögen. Wir untersuchen einige Thermikszenerarien mit einem neuen mathematischen 3D Modell [2]. Dieses neue Modell basiert auf einem Vorgängermodell von Bulirsch, Nerz, Pesch, von Stryk [3], welches auf den Flug in einer Vertikalebene beschränkt war und deshalb spiralförmige Aufstiegsbögen nicht fassen konnte. In [3] wurde die Lösung mittels der direkten Kollokations Software DIRCOL [4] und der Mehrzielmethode [5, 6] berechnet. Eine direkte Lösung des Vorgängermodells mittels der Modellierungssprache AMPL und einem Innere Punkte Löser ist in [7] beschrieben.

## 2 Flugbahnoptimierung

Für Flüge über eine kurze Distanz, wie es typischerweise bei Hängegleitern der Fall ist, kann man die Bewegungsgleichungen bzgl. einer flachen nicht-rotierenden Erde benutzen [8]. Die  $xy$ -Ebene modelliert die flache Erdoberfläche. Mit  $h$  wird die Höhe bezeichnet. Der Hängegleiter zusammen mit dem Piloten ist als Punktmasse an der Position  $(x, y, h)$  modelliert. Folgende Kräfte wirken: Gewichtskraft

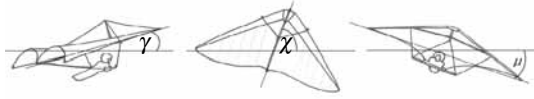
$W = mg$ , Auftrieb  $L(v, h; c_L) = c_L \rho S v^2 / 2$  und Widerstand  $D(v, h; c_D) = c_D(c_L) \rho S v^2 / 2$ . Wir benutzen die quadratische Widerstandspolare  $c_D(c_L) = 0.034 + 0.069662 c_L^2$  eines speziellen Drachenfliegers aus [3]. Folgende Konstanten treten auf:  $m = 100$  [kg] (Masse des Piloten und des Drachenfliegers),  $S = 14$  [m<sup>2</sup>] (Bezugsfläche),  $\rho = 1.13$  [kg m<sup>-3</sup>] (Luftsdichte in einer Höhe von etwa 1000 [m] über dem Meeresspiegel),  $g = 9.81$  [m s<sup>-2</sup>] (Gravitationskonstante). Der Geschwindigkeitsvektor des Drachenfliegers wird in sphärischen Koordinaten beschrieben, vgl. [8] und die Abb. 1:  $v$  bezeichnet die Geschwindigkeit,  $\gamma$  den Bahnneigungswinkel und  $\chi$  den Azimutwinkel. Zustandsvariablen sind deshalb  $(x, y, h, v, \gamma, \chi)$ .

In diesem einfachen Modell wirken der Auftriebsbeiwert  $c_L$  und der Querneigungswinkel  $\mu$  als (math.) Steuerungen. In Wirklichkeit werden  $c_L$  und  $\mu$  indirekt durch Bewegungen des Piloten bzgl. des Drachenfliegers beeinflusst.

Zur Vereinfachung ist angenommen, dass alle betrachteten Varianten der Thermikwinde nur eine vertikale Komponente  $U$  besitzen.

Damit erhalten wir die Bewegungsgleichungen [8, 9]:

$$\begin{aligned} \dot{x} &= v \cos \gamma \cos \chi, & \dot{y} &= v \cos \gamma \sin \chi, \\ \dot{h} &= v \sin \gamma + U(x, y), \\ \dot{v} &= \frac{-D(v, h; c_L)}{m} - g \sin \gamma - \dot{U}(x, y) \sin \gamma, \end{aligned} \quad (1)$$



**Abbildung 1:** Bahnneigungswinkel  $\gamma$  (Zustand), Azimutwinkel  $\chi$  (Zustand), Querneigungswinkel  $\mu$  (Steuerung).

$$\begin{aligned}\dot{\gamma} &= \frac{L(v, h; c_L) \cos \mu}{m v} - \frac{g}{v} \cos \gamma - \frac{1}{v} \dot{U}(x, y) \cos \gamma, \\ \dot{\chi} &= \frac{L(v, h; c_L) \sin \mu}{m v \cos \gamma}.\end{aligned}$$

Die komplizierten Start- und Landephase sind nicht im Modell enthalten. Stattdessen nehmen wir an, das der Drachenflieger zur Startzeit  $t = 0$  und zur Endzeit  $t_f$  sich in der Geschwindigkeit besten Gleitens  $v_G$  und dem zugehörigen Gleitwinkel  $\gamma_G$  befindet. Diese Daten können aus der Widerstandspolare berechnet werden, siehe [10] S. 467–471. Damit ergeben sich die Randbedingungen:

$$\begin{aligned}x(0) &= 0 \text{ [m]}, y(0) = 0 \text{ [m]} \\ h(0) &= 1000 \text{ [m]}, h(t_f) = 900 \text{ [m]} \\ v(0) &= v(t_f) = v_G := 13.29 \text{ [m/s]} \\ \gamma(0) &= \gamma(t_f) = \gamma_G := -0.017 \text{ [rad]} \\ \chi(0) &= 0 \text{ [rad]}\end{aligned} \quad (2)$$

Um einen Strömungsabriss zu vermeiden muss der Auftriebsbeiwert  $c_L$  (Steuerung) eingeschränkt werden:

$$0 \leq c_L \leq c_{L, \max} := 1.4 \quad (3)$$

Die Endzeit  $t_f$  ist frei. Wir betrachten verschiedene Szenarios der Thermikwinde.

### 3 Thermikschlauch mit innerer Wirbelströmung

An einem sonnigen Tag wird die Bodenluft erwärmt. Aufgrund der geringeren Dichte lösen sich die erwärmten Luftpakete. Schließlich formt sich eine Thermikblase, die aufsteigt (Abb. 2). Unter günstigen Voraussetzungen bildet sich ein Thermikschlauch, siehe Abb. 3 und [11, 12, 13]. Ein math. Modell für die vertikale Geschwindigkeit  $U$  eines Thermikschlauchs geht zurück auf Klingler (vgl. [2]) und

wird durch die folgende Funktion mit  $r = 100$  [m] beschrieben:

$$U_a(x, y) = 2.5 \cdot \left[ 1 - \left( \frac{x}{r} - 2.5 \right)^2 - \left( \frac{y}{r} - 2.5 \right)^2 \right] \cdot \exp \left[ - \left( \frac{x}{r} - 2.5 \right)^2 - \left( \frac{y}{r} - 2.5 \right)^2 \right], \quad (4)$$

Im Bereich von  $150 \text{ [m]} \leq R = \sqrt{x^2 + y^2} \leq 350 \text{ [m]}$  ist die Geschwindigkeit positiv mit einem Maximalwert von  $U_{\max} = 2.5 \text{ [m/s]}$ . Dies ist der Thermikschlauch. Am Rand gibt es Abwinde.

## 4 Wolkenstraße

Oft sind Thermikschläuche sehr nahe benachbart. Dies nennt man eine Wolkenstraße.

Im folgenden wird ein Modell einer Wolkenstraße mit zwei Thermikschläuchen nach Klingler

$$U_{cl,2}(x, y) = 2.5 \sum_{i=1}^2 \exp \left[ \beta \left( (x - r_i)^2 + (y - r_i)^2 \right) \right] \quad (5)$$

mit  $r_1 = 250 \text{ [m]}$ ,  $r_2 = 600 \text{ [m]}$ ,  $\beta = -0.0002 \text{ [m}^{-1}]$  vorgestellt, siehe Abb. 5. Die hier verwendete Funktion besitzt ausschließlich positive Werte, auf die Modellierung von Turbulenzen am Rande der Thermik wird hier verzichtet. Eine Thermik einer Wolkenstraße mit drei Thermikschläuchen nach Klingler ist gegeben durch

$$U_{cl,3}(x, y) = 2.5 \sum_{i=1}^3 \exp \left[ \beta \left( (x - r_i)^2 + (y - r_i)^2 \right) \right] \quad (6)$$

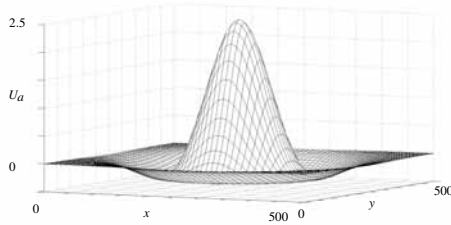
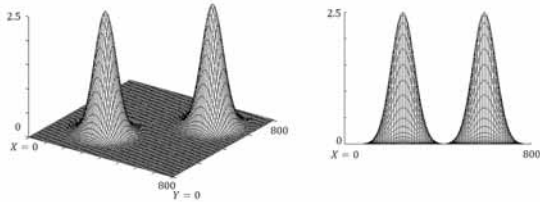
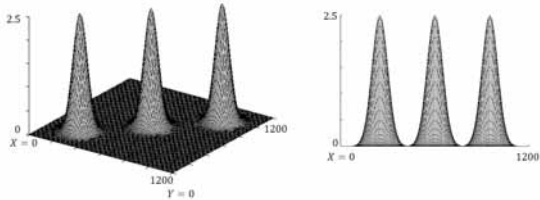
mit  $r_1 = 250 \text{ [m]}$ ,  $r_2 = 600 \text{ [m]}$ ,  $r_3 = 950 \text{ [m]}$ ,  $\beta = -0.0002 \text{ [m}^{-1}]$ , siehe Abb. 6.



**Abbildung 2:** Bildung und Aufstieg einer Thermikblase.



Abbildung 3: Thermikschlauch.


 Abbildung 4: Windgeschwindigkeit  $U_a(x, y)$  des Thermikschlauchs.

 Abbildung 5: Wolkenstraße  $U_{cl,2}(x, y)$ .

 Abbildung 6: Wolkenstraße  $U_{cl,3}(x, y)$ .

## 5 Numerische Ergebnisse

Die optimalen Steuerungsaufgaben werden mit einer direkten Methode gelöst. Zunächst wird ein gemeinsames Zeitgitter für Zustände und Steuerungen eingeführt. Die Steuerungen werden als Treppenfunktionen über dem Zeitgitter approximiert. Die Zustände werden z.B. mit dem impliziten Euler oder der impliziten Trapezregel als stückweise lineare Funktionen approximiert. Dies führt auf ein nichtlineares Optimierungsproblem (NLP) [14, 7]. Numerische Lösungen werden mittels der Modellierungssprache AMPL [15] und dem SQP Löser SNOPT [16] oder dem Innere Punkte Löser IPOPT [17] berechnet. Alternativ wird auch die direkte Kollokations Software DIRCOL [4] zusammen mit dem SQP Löser SNOPT benutzt. DIRCOL benutzt die Hermite-Simpson Methode zur Approximation der Differentialgleichungen und erlaubt eine automatische Gitterverfeinerung sowie eine Schätzung der adjungierten Variablen.

### 5.1 Maximale Durchschnittsgeschwindigkeit in einem Thermikschlauch

Als Zielfunktional verwenden wir die Durchschnittsgeschwindigkeit

$$I_{av}[c_L, \mu] = \frac{\sqrt{[x(t_f)]^2 + [y(t_f)]^2}}{t_f} \rightarrow \max \quad (7)$$

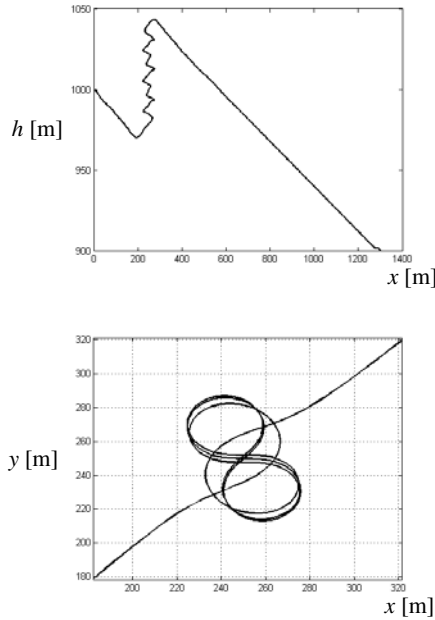
zusammen mit zwei Zustandsbeschränkungen die eine untere Schranke für die Endreichweite vorgeben

$$x(t_f) \geq 1300 \text{ [m]}, \quad y(t_f) \geq 1300 \text{ [m]}. \quad (8)$$

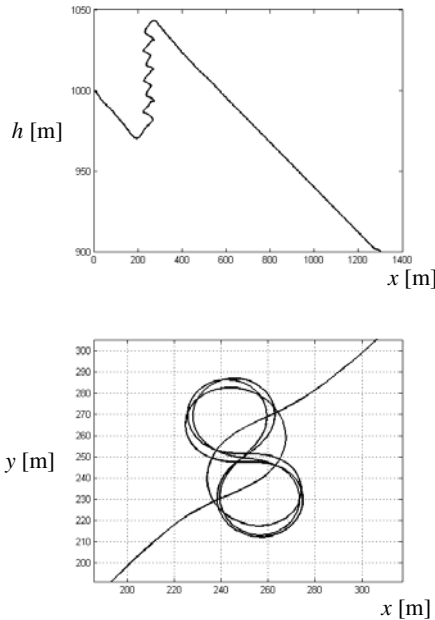
Als Thermikwind verwenden wir das Modell  $U_a(x, y)$  des Thermikschlauchs. Zusätzlich verwenden wir eine Beschränkung an die Beschleunigung:

$$|\dot{v}(t)| \leq 3 \text{ [m/s}^2\text{]} \quad (9)$$

Numerische Ergebnisse die mittels der impliziten Trapezregel mit 500 Gitterpunkten und AMPL/SNOPT berechnet wurden finden sich in Abb. 7 zusammen mit einem Zoom. Eine etwas andere Lösung wurde vom Optimierer mit einer anderen Startschätzung der Optimierungsvariablen des NLPs berechnet. Der Drachenflieger spiralt mehrmals über



**Abbildung 7:** Flugbahn über dem Thermikschlauch als Optimale Steuerungsaufgabe (1–4,7–9).



**Abbildung 8:** Andere Flugbahn über dem Thermikschlauch als Optimale Steuerungsaufgabe (1–4,7–9).

dem Zentrum (250 [m], 250 [m]) des Thermikwinds. Beide Zielfunktionalwerte unterscheiden sich nur wenig:  $I_{av} \doteq 9.02783$  [m/s] bzw. 9.03036 [m/s]. Nach einer Flugdauer von  $t_f \doteq 203.19$  [s] wird die untere Schranke der Reichweite  $x(t_f) = y(t_f) = 1300$  [m] exakt angenommen.

## 5.2 Wolkenstraße

Als Zielfunktion wird die maximale Durchschnittsgeschwindigkeit

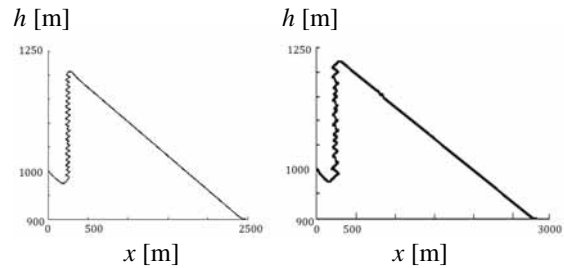
$$I_{av}[c_L, \mu] = \frac{\sqrt{[x(t_f)]^2 + [y(t_f)]^2}}{t_f} \rightarrow \max \quad (10)$$

zusammen mit einer unteren Schranke für die Endzeit  $t_f \geq 500$  [s] benutzt.

Um einen guten Vergleich zwischen einfacher und mehrfacher Bodenthermik zu erhalten benutzen wir eine einfache Bodenthermik ohne Randturbulenzen nach Klingler

$$U_{cl,1}(x,y) = 2.5 \exp[\beta((x-r_1)^2 + (y-r_1)^2)] \quad (11)$$

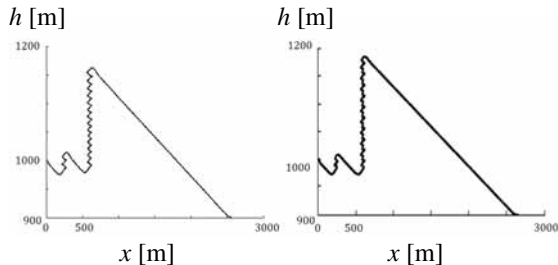
mit  $r_1 = 250$ ,  $\beta = -0.0002$ . Die berechneten Flugbahnen des Drachenfliegers finden sich in Abb. 9. Hier liefert die Lösung mit DIRCOL/SNOPT



**Abbildung 9:** Flugbahn mit  $U_{cl,1}$ . Links: AMPL/IPOPT Rechts: DIRCOL/SNOPT

eine größere Reichweite mit höherer Durchschnittsgeschwindigkeit, siehe Tabelle 1.

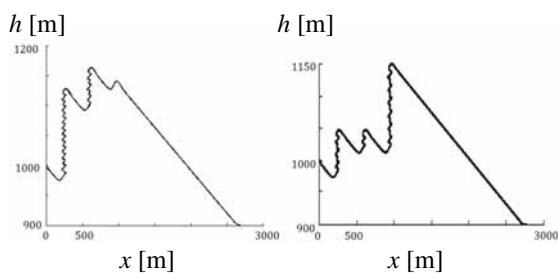
Die Flugbahnen des Drachenfliegers über der Wolkenstraße  $U_{cl,2}$  finden sich in Abb. 10. Bei beiden Lösungen wird dieselbe Strategie benutzt: Eine einfache Spirale über dem ersten Maximum der Thermik und mehrere Spiralen über dem zweiten Maximum der Thermik. Der Drachenflieger erreicht eine größere



**Abbildung 10:** Flugbahn mit  $U_{cl,2}$ . Links: AMPL/IPOPT Rechts: DIRCOL/SNOPT

Reichweite bei der Lösung mit DIRCOL/SNOPT, siehe Tabelle 2.

Unterschiedliche Strategien zeigen sich in der Lösung über der Wolkenstraße  $U_{cl,3}$ , siehe Abb. 11. Der



**Abbildung 11:** Flugbahn mit  $U_{cl,3}$ . Links: AMPL/IPOPT Rechts: DIRCOL/SNOPT

Drachenflieger benutzt hauptsächlich das erste Maximum der Thermik in der numerischen Lösung, die mit AMPL/IPOPT berechnet wurde, dagegen wird hauptsächlich das letzte Maximum der Thermik in der Lösung benutzt, die mit DIRCOL/SNOPT berechnet wurde. Da alle Thermikmaxima gleich stark modelliert sind, sind die Lösungen gleichwertig. Hier sieht man deutlich, dass mehrere lokale Optima existieren, weitere Details siehe Tabelle 3.

**Tabelle 1:** Ergebnisse für die Thermik  $U_{cl,1}$

	AMPL/IPOPT	DIRCOL/SNOPT
Gitterpunkte	300	239
$I_{av}$	6.92 m/s	7.30 m/s
$t_f$	500 s	500 s
$x(t_f)$	2803.57 m	2490.32 m
$y(t_f)$	2334.22 m	2399.97 m
Iterationen	1073	445
CPU	15.54 s	45.08 s

**Tabelle 2:** Ergebnisse für die Thermik  $U_{cl,2}$

	AMPL/IPOPT	DIRCOL/SNOPT
Gitterpunkte	300	200
$I_{av}$	7.18 m/s	7.50 m/s
$t_f$	500 s	500 s
$x(t_f)$	2558.14 m	2652.53 m
$y(t_f)$	2520.99 m	2649.49 m
Iterationen	1407	578
CPU	48.29 s	35.38 s

**Tabelle 3:** Ergebnisse für die Thermik  $U_{cl,3}$

	AMPL/IPOPT	DIRCOL/SNOPT
Gitterpunkte	300	336
$I_{av}$	7.60 m/s	7.81 m/s
$t_f$	500 s	500 s
$x(t_f)$	2688.37 m	2763.74 m
$y(t_f)$	2688.38 m	2758.48 m
Iterationen	898	1528
CPU	118.49 s	193.55 s

## 6 Zusammenfassung

Realistische numerische Lösungen wurden für diese interdisziplinäre Aufgabe, die vom Sport über Flugmechanik, Meteorologie, Optimale Steuerung, nicht-lineare Optimierung bis zur Numerischen Mathematik viele Bereiche berührt, berechnet. Abhängig von der Startnäherung der Optimierungsvariablen und der genauen Formulierung als nichtlineares Optimierungsproblem wurden verschiedene Lösungen berechnet. Zudem sind wegen der Vernachlässigung der Erdrotation im Modell Spiralen im Uhrzeigersinn bzw. Gegenuhrzeigersinn meist gleichwertig. Dies legt nahe, dass es hier meist verschiedene lokale Optima gibt. Weitere numerische Resultate, z.B. für einen Berg- und Talwind, finden sich in [2].

## Literatur

- [1] P. Dees, "Hang Glider Design and Performance," in *Proc. 10th AIAA Aviation Technology, Integration, and Operations Conference*, Fort Worth, Texas, 2010, Paper-No. AIAA 2010-9300.

- [2] K. Chudej, A.-L. Klingler, A. Britzelmeier, "Flight Path Optimization of a Hang-Glider in a Thermal Updraft," *IFAC-PapersOnLine* 48 (2015) 1, pp. 808-812.
- [3] R. Bulirsch, E. Nerz, H.J. Pesch, O. von Stryk, "Combining Direct and Indirect Methods in Optimal Control: Range Maximization of a Hang Glider". In: *Optimal Control*, R. Bulirsch, A. Miele, J. Stoer, K.H. Well, Eds. Basel: Birkhäuser, 1993, pp. 273–288.
- [4] O. von Stryk, *Numerische Lösung optimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der adjungierten Variablen*. Düsseldorf: Fortschrittberichte VDI, Reihe 8, Nr 441, VDI-Verlag, Düsseldorf. 1995
- [5] R. Bulirsch, *Die Mehrzielmethode zur numerischen Lösung von nichtlinearen Randwertproblemen und Aufgaben der optimalen Steuerung*. Report der Carl-Cranz-Gesellschaft 251, Oberpfaffenhofen, 1971.
- [6] H.J. Oberle, W. Grimm, *BNDSCO: A program for the numerical solution of optimal control problems*. Inst. für Angewandte Mathematik, Universität Hamburg, 2001.
- [7] R.E. Vanderbei, "Case Studies In Trajectory Optimization: Trains, Planes, And Other Pastimes." *Optimization and Engineering*, vol. 2, pp. 215–243, 2000.
- [8] A. Miele, *Flight Mechanics – Theory of Flight Paths*. Reading: Addison-Wesley, 1962.
- [9] A. Miele, T. Wang, H. Wang, W.W. Melvin, "Optimal Penetration Landing Trajectories in the Presence of Windshear," *Journal of Optimization Theory and Applications*, vol. 57, 1988.
- [10] D.P. Raymer, *Aircraft Design: A Conceptual Approach*. Washington, D.C.: AIAA, 1989.
- [11] P. Janssen, K. Tänzler, *Drachenfliegen*. München: Nymphenburger, 1993.
- [12] P. Janssen, K. Tänzler, *Drachenfliegen für Meister*. München: Nymphenburger, 1993.
- [13] B. Klose, *Meteorologie*. Berlin: Springer, 2008.
- [14] J.T. Betts, "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, 21, 2, pp. 193–207, 1998.
- [15] R. Fourer, D.M. Gay, B.W. Kernighan, (2003) *AMPL: A Modeling Language for Mathematical Programming*. Pacific Grove, CA: Thomson-Brooks/Cole, 2004.
- [16] P.E. Gill, W. Murray, M.A. Saunders, "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, 47, 1, pp. 99–131, 2005.
- [17] A. Wächter, L.T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, 106, pp. 25–57, 2006.

# Interaktion stochastischer Filter für unterschiedlich dimensionierte, heterogene Zustandsräume

Edmond Skeli<sup>1</sup>, Dirk Weidemann<sup>1</sup>

<sup>1</sup>Institut für Systemdynamik und Mechatronik

FH Bielefeld

*edmond.skeli@fh-bielefeld.de*

*dirk.weidemann@fh-bielefeld.de*

Im Kontext der Diagnose technischer Systeme ist neben der reinen Fehlererkennung auch das Ermitteln des Fehlerortes (Fehlerlokalisierung) und der Fehlergröße (Fehleridentifikation) von Bedeutung. Vielfach finden für die Fehlerlokalisierung und -identifikation modellbasierte Diagnoseverfahren Verwendung, welche auf der gleichzeitigen Nutzung mehrerer mathematischer Modelle basieren, sodass Fehlerlokalisierung und -identifikation im Sinne einer hybriden Zustandsschätzung durchgeführt werden können. Bekannte Ansätze zur hybriden Zustandsschätzung wie Generalized-Pseudo-Bayesian-Ansätze oder der Interacting-Multiple-Model-Ansatz verwenden für jedes der betrachteten Modelle ein stochastisches Filter, wobei die Ergebnisse der einzelnen stochastischen Filter geeignet aggregiert werden. Allerdings gilt es zu berücksichtigen, dass die einzelnen mathematischen Modelle unterschiedlich dimensionierte, heterogene Zustandsräume aufweisen können. Folglich kann das Aggregieren, d.h. die gewichtete Kombination der Schätzergebnisse nicht ohne entsprechende Modifikation durchgeführt werden. Anhand eines hydraulischen Zylinders werden mögliche Modifikationsansätze für die Interaktion der Filter für unterschiedlich dimensionierte, heterogene Zustandsräume erläutert.

## 1 Einleitung

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 ist bspw. der Schutz des Betreibers bzw. Benutzers des technischen Systems. 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. Der bei komplexen Sys-

temen vielfach angewendete objektorientierte Modellierungsansatz führt im Allgemeinen auf nichtlineare differential-algebraische Gleichungen. Konsequenterweise findet im vorliegenden Beitrag ein modellbasiertes Diagnoseverfahren für nichtlineare DAE-Systeme Verwendung. Dieses Diagnoseverfahren basiert auf der gleichzeitigen Nutzung mehrerer mathematischer Modelle und ermöglicht die Fehlerlokalisierung und -identifikation im Sinne einer hybriden Zustandsschätzung durchzuführen.

Da bei der optimalen hybriden Zustandsschätzung der benötigte Rechenaufwand exponentiell mit jedem Abtastschritt steigt, lassen sich nur suboptimale hybride Zustandsschätzer in der Praxis realisieren. Bekannte und vielfach eingesetzte suboptimale Verfahren, wie bspw. der Generalized-Pseudo-Bayesian-Ansatz 1. bzw. 2. Ordnung oder der Interacting-Multiple-Model-Ansatz, verwenden für jedes der betrachteten mathematischen Modelle ein stochastisches Filter und

aggregieren deren Ergebnisse geeignet. Im Rahmen der modellbasierten Diagnose gilt es jedoch zu berücksichtigen, dass die Modellierung der Fehler mittels einer Zustandserweiterung erfolgt. Folglich weist der Zustandsraum des Modells des fehlerfreien Falls eine geringere Dimension als die Zustandsräume der Modelle für die einzelnen Fehlerfälle auf. Dies ist bei der Interaktion der stochastischen Filter, d.h. beim Aggregieren der Ergebnisse der einzelnen Filter, zu berücksichtigen. Das Aggregieren entspricht dem Mischen der mit Hilfe der Filter bestimmten Wahrscheinlichkeitsdichtefunktionen und ist nur dann möglich, wenn alle Dichtefunktionen dieselbe Dimension aufweisen. Somit müssen die Dichtefunktionen von Modellen mit niedrigerer Zustandsdimension vor dem Mischen geeignet modifiziert werden.

Im Rahmen des vorliegenden Aufsatzes werden verschiedene Ansätze für das Modifizieren der Wahrscheinlichkeitsdichtefunktionen vorgestellt. Anhand eines hydraulischen Zylinders, der durch ein nichtlineares DAE-System vom Index 1 beschrieben wird, werden die Auswirkungen der Modifikationen auf die hybride Zustandsschätzung erläutert. Zur hybriden Zustandsschätzung wird eine Interacting-Multiple-Model-Ansatz eingesetzt, wobei als stochastische Filter skalierte Unscented-Kalman-Filter Verwendung finden.

## 2 Modellierung fehlerbehafteter Systeme

Hinsichtlich der Modellierung von fehlerbehafteten Systemen ist es notwendig sowohl zwischen den einzelnen Orten, an denen Fehler auftreten können, als auch zwischen den Charakteristiken dieser Fehler zu unterscheiden. Da Fehler neben den Aktoren und Sensoren auch das Verhalten des zugrunde liegenden Prozesses verändern können, klassifiziert man die Fehler in Abhängigkeit des Fehlerortes als Aktor-, Sensor- oder Prozessfehler. Während somit nur zwischen drei verschiedenen Fehlerorten zu unterscheiden ist, gelingt eine derart einfache Unterscheidung bei den einzelnen Fehlercharakteristiken nicht, da unendlich viele unterschiedliche Charakteristiken existieren. Ungeachtet dessen, dass folglich die Fehleramplitude prinzipiell einen beliebigen Verlauf aufweisen kann, lassen sich die Fehleramplituden in vielen Fällen im Sinne eines

abrupten Fehlers mit konstanter Amplitude oder eines konstant ansteigenden Fehlers modellieren.

Verfolgt man im Kontext der modellbasierten Diagnose nicht nur das Ziel aufgetretene Fehler zu erkennen sondern auch die Fehler zu lokalisieren und darüber hinaus deren Fehleramplitude zu identifizieren, bietet es sich an das zeitliche Verhalten der Fehleramplituden durch das Einführen zusätzlicher Zustandsgrößen zu modellieren. Im Kontext von DAE-Systemen sind dies zusätzliche differentielle Zustandsgrößen. Den Ausgangspunkt bildet das Modell des fehlerfreien Systemverhaltens, das durch das nichtlineare DAE-System vom Index 1

$$\dot{x}_d^{(\text{ff})} = \phi_d^{(\text{ff})}(x_d^{(\text{ff})}, x_a, u, t), \quad (1)$$

$$g_a^{(\text{ff})}(x_d^{(\text{ff})}, x_a, u, t) = 0 \wedge \det\left(\frac{\partial g_a^{(\text{ff})}}{\partial x_a}\right) \neq 0, \quad (2)$$

$$y = h^{(\text{ff})}(x_d^{(\text{ff})}, x_a, u, t) \quad (3)$$

beschrieben wird. Hierbei kennzeichnen  $x_d^{(\text{ff})} \in \mathbb{R}^{n_d}$ ,  $x_a^{(\text{ff})} \in \mathbb{R}^{n_a}$  die differentiellen bzw. algebraischen Zustandsgrößen,  $u \in \mathbb{R}^m$  die Stellgrößen,  $y \in \mathbb{R}^v$  die Messgrößen und  $\phi_d^{(\text{ff})}$ ,  $g_a^{(\text{ff})}$  und  $h^{(\text{ff})}$  die korrespondierenden Differentialgleichungen, algebraischen Gleichung und Ausgangsgleichungen des fehlerfreien Systems.

Bei dem im Abschnitt 5 betrachteten hydraulischen Zylinder werden als Fehlerfälle eine interne und externe Leckage betrachtet. Da in beiden Fällen von abrupt auftretenden Fehler mit konstanter Fehleramplitude ausgegangen wird, lassen die Fehlerdynamiken durch  $\dot{x}_{d,\text{erw}}^{(\text{in})} = 0$  und  $\dot{x}_{d,\text{erw}}^{(\text{ex})} = 0$  beschreiben. Im Weiteren seien  $x_d^{(\text{in})} = [(x_d^{(\text{ff})})^T, x_{d,\text{erw}}^{(\text{in})}]^T$ ,  $x_d^{(\text{ex})} = [(x_d^{(\text{ff})})^T, x_{d,\text{erw}}^{(\text{ex})}]^T$  mit  $x_{d,\text{erw}}^{(\text{in})}, x_{d,\text{erw}}^{(\text{ex})} \in \mathbb{R}$  die differentiellen Zustände für den Fall der internen und der externen Leckage. Somit charakterisiert

$$\dot{x}_d^{(\text{in})} = \phi_d^{(\text{in})}(x_d^{(\text{in})}, x_a, u, t) = \begin{bmatrix} \phi_d^{(\text{ff})}(x_d^{(\text{in})}, x_a, u, t) \\ 0 \end{bmatrix}, \quad (4)$$

$$g_a^{(\text{in})}(x_d^{(\text{in})}, x_a, u, t) = 0 \wedge \det\left(\frac{\partial g_a^{(\text{in})}}{\partial x_a}\right) \neq 0, \quad (5)$$

$$y = h^{(\text{in})}(x_d^{(\text{in})}, x_a, u, t) \quad (6)$$

das Verhalten im Fall der internen Leckage und

$$\dot{x}_d^{(\text{ex})} = \phi_d^{(\text{ex})}(x_d^{(\text{ex})}, x_a, u, t) = \begin{bmatrix} \phi_d^{(\text{ff})}(x_d^{(\text{ex})}, x_a, u, t) \\ 0 \end{bmatrix}, \quad (7)$$

$$g_a^{(\text{ex})}(x_d^{(\text{ex})}, x_a, u, t) = 0 \wedge \det\left(\frac{\partial g_a^{(\text{ex})}}{\partial x_a}\right) \neq 0, \quad (8)$$

$$y = h^{(\text{ex})}(x_d^{(\text{ex})}, x_a, u, t) \quad (9)$$

im Fall der externen Leckage.

Die Gesamtheit des hier betrachteten Systemverhaltens lässt sich folglich im Sinne eines hybriden Systems mit dem hybriden Zustandsvektor  $\xi = [z, x_d^T, x_a^T]^T$ , der die drei diskretwertigen Zuständen (Moden)  $z \in \mathcal{M} = \{m^{(\text{ff})}, m^{(\text{in})}, m^{(\text{ex})}\}$  sowie die wertkontinuierlichen differentiellen und algebraischen Zustände umfasst, beschreiben. Während die wertkontinuierliche Systemdynamik in den Moden jeweils durch das korrespondierende nichtlineare DAE-System repräsentiert wird, nutzt man zur Beschreibung der wertdiskreten Dynamik eine Markovkette, s. [3].

### 3 Mischen der Wahrscheinlichkeitsdichtefunktionen

Grundlage des Interacting-Multiple-Model-Ansatzes (IMM-Ansatz), aber auch der Generalized-Pseudo-Bayesian-Ansätze, ist das Mischen einzelner Wahrscheinlichkeitsdichtefunktionen. Der Übersichtlichkeit halber werden die differentiellen und algebraischen Zustände zu einem Vektor  $x = [x_d^T, x_a^T]^T \in \mathbb{R}^n$  mit  $n = n_d + n_a$  zusammengefasst. Seien  $p^{(1)}(x), \dots, p^{(q)}(x)$  Dichtefunktionen und  $\mu_1, \dots, \mu_q \in [0, 1] \subset \mathbb{R}$ ,  $q \in \mathbb{N}$  mit  $\sum_{i=1}^q \mu_i = 1$ , dann ist die Mischung

$$p(x) = \sum_{i=1}^q \mu_i p^{(i)}(x) \quad (10)$$

ebenfalls eine Dichtefunktion.

Offensichtlich gelingt das Mischen der Wahrscheinlichkeitsdichtefunktionen nur dann, wenn die einzelnen Dichtefunktionen dieselbe Dimension aufweisen. Darüber hinaus ist zu berücksichtigen, dass die im

Rahmen der Fehlermodellierung eingeführten erweiterten Zustandsgrößen typischerweise unterschiedliche physikalische Effekte beschreiben, was im Rahmen des Mischens zu berücksichtigen ist. Um das Mischen und das damit verbundene Anpassen der zugrunde liegenden Dichtefunktion bei unterschiedlich dimensionierten, heterogenen Zustandsräumen zu verdeutlichen, soll im Folgenden davon ausgegangen werden, dass das hybride System die Moden  $m^{(1)}$ ,  $m^{(2)}$  und  $m^{(3)}$  umfasst. Für die wertkontinuierlichen Zustandsgrößen der einzelnen Moden gelte  $x^{(1)} \in \mathbb{R}^n$ ,  $x^{(2)} = [x, x_{d,\text{erw}}^{(2)}]^T \in \mathbb{R}^n \times \mathbb{R}^{n_{d,\text{erw}}^{(2)}}$  und  $x^{(3)} = [x, x_{d,\text{erw}}^{(3)}]^T \in \mathbb{R}^n \times \mathbb{R}^{n_{d,\text{erw}}^{(3)}}$  und  $p^{(1)} : \mathbb{R}^n \rightarrow [0, \infty) \subset \mathbb{R}$ ,  $p^{(2)} : \mathbb{R}^n \times \mathbb{R}^{n_{d,\text{erw}}^{(2)}} \rightarrow [0, \infty) \subset \mathbb{R}$  sowie  $p^{(3)} : \mathbb{R}^n \times \mathbb{R}^{n_{d,\text{erw}}^{(3)}} \rightarrow [0, \infty) \subset \mathbb{R}$  seien die korrespondierenden Dichtefunktionen. Um das Mischen der einzelnen Dichtefunktionen zu ermöglichen, können die zugrunde liegenden Dichtefunktionen  $p^{(1)}, p^{(2)}, p^{(3)}$  entsprechend

$$p_e^{(1)}(x, x_{d,\text{erw}}^{(2)}, x_{d,\text{erw}}^{(3)}) = p^{(1)}(x) p^{(1,2)}(x_{d,\text{erw}}^{(2)}) p^{(1,3)}(x_{d,\text{erw}}^{(3)}) \quad (11)$$

$$p_e^{(2)}(x, x_{d,\text{erw}}^{(2)}, x_{d,\text{erw}}^{(3)}) = p^{(2)}(x, x_{d,\text{erw}}^{(2)}) p^{(2,3)}(x_{d,\text{erw}}^{(3)}) \quad (12)$$

$$p_e^{(3)}(x, x_{d,\text{erw}}^{(2)}, x_{d,\text{erw}}^{(3)}) = p^{(3)}(x, x_{d,\text{erw}}^{(3)}) p^{(3,2)}(x_{d,\text{erw}}^{(2)}) \quad (13)$$

erweitert werden. Hierbei können die Dichtefunktionen  $p^{(1,2)}$ ,  $p^{(1,3)}$ ,  $p^{(2,3)}$  und  $p^{(3,2)}$ , analog zu der Vorgehensweise in [1], beliebig gewählt werden.

### 4 Interacting-Multiple-Model-Ansatz

Da beim IMM-Ansatz für jeden Mode ein geeignetes stochastisches Filter zu verwenden ist, findet das in [3] beschriebene skalierte Unscented-Kalman-Filter (UKF) Verwendung, welches für nichtlineare DAE-Systeme vom Index-1 einsetzbar ist und auf den in [2] beschriebenen Filteransätzen basiert. Hierbei schätzen die in [2] und [3] beschriebenen Ansätze jedoch nur die differentiellen Zustandsgrößen im Sinne eines Kalman-Filters. Demgegenüber werden die algebraischen Zustandsgrößen nach jedem Korrekturschritt so

angepasst, dass die algebraischen Zwangsbedingungen erfüllt sind. D.h. zu jeden Abtastzeitpunkt  $t = kT$  werden, ausgehend von bekannten differentiellen Zustandsgrößen, konsistente Anfangswerte durch das Lösen der korrespondierenden algebraischen Zwangsbedingung  $g_a^{(\cdot)}(x_d^{(\cdot)}, x_a, u, t) = 0$  ermittelt. Berücksichtigt man, dass die algebraischen Zustandsgrößen bei den hier betrachteten Index-1 Systemen in Abhängigkeit der differentiellen Zustandsgrößen ermittelt werden, genügt es den IMM-Ansatz nur für die differentiellen Zustandsgrößen aufzusetzen.

Obgleich sich der IMM-Ansatz aus insgesamt vier Schritten zusammensetzt, ist im Rahmen der Erweiterung der Dichtefunktionen nur die Reinitialisierung der einzelnen UKFs zu Beginn eines jeden Abtastintervalls von Bedeutung. Die Reinitialisierung resultiert aus dem Mischen der Dichtefunktionen der einzelnen Filter, sodass sich die Erwartungswerte und die Kovarianzmatrizen für jeden Mode zu

$$\hat{x}_{e,d,k|k}^{(j)} = \sum_{i=1}^q \hat{x}_{e,d,k|k}^{(i)} \mu_{k|k}^{(ij)} \quad (14)$$

$$\tilde{P}_{x_e, d, x_e, d, k}^{(j)} = \sum_{i=1}^q \mu_{k|k}^{(ij)} \left( P_{x_e, d, x_e, d, k}^{(i)} + \Delta \Delta^T \right) \quad (15)$$

mit  $\Delta = \hat{x}_{e,d,k|k}^{(i)} - \hat{x}_{e,d,k|k}^{(j)}$  und  $j = 1, \dots, q$  ergeben. Der vollständige IMM-Ansatz für nichtlineare Index-1 Systeme, der auch die Bestimmung der Mischwahrscheinlichkeiten  $\mu_{k|k}^{(ij)}$  umfasst, findet sich in [3].

**Erweiterung der Dichtefunktionen mit Normalverteilungen.** Geht man im Weiteren davon aus, dass nur drei Moden zu berücksichtigen sind, ergibt eine Erweiterung der Dichtefunktionen mit  $p^{(1,2)}(x_{d,erw}^{(2)}) = p^{(3,2)}(x_{d,erw}^{(2)}) = \mathcal{N}(x_{d,erw}^{(2)}; e^{(2)}, P^{(2)})$  und  $p^{(1,3)}(x_{d,erw}^{(3)}) = p^{(2,3)}(x_{d,erw}^{(3)}) = \mathcal{N}(x_{d,erw}^{(3)}; e^{(3)}, P^{(3)})$  die folgenden erweiterten Erwartungswerte

$$\hat{x}_{e,d,k|k}^{(1)} = \left[ \left( \hat{x}_{d,k|k}^{(1)} \right)^T, \left( e^{(2)} \right)^T, \left( e^{(3)} \right)^T \right]^T, \quad (16)$$

$$\hat{x}_{e,d,k|k}^{(2)} = \left[ \left( \hat{x}_{d,k|k}^{(2)} \right)^T, \left( \hat{x}_{d,erw,k|k}^{(2)} \right)^T, \left( e^{(3)} \right)^T \right]^T, \quad (17)$$

$$\hat{x}_{e,d,k|k}^{(3)} = \left[ \left( \hat{x}_{d,k|k}^{(3)} \right)^T, \left( e^{(2)} \right)^T, \left( \hat{x}_{d,erw,k|k}^{(3)} \right)^T \right]^T \quad (18)$$

und Kovarianzmatrizen

$$P_{x_e, d, x_e, d, k}^{(1)} = \begin{bmatrix} P_{x_d, x_d, k}^{(1)} & 0 & 0 \\ 0 & P^{(2)} & 0 \\ 0 & 0 & P^{(3)} \end{bmatrix}, \quad (19)$$

$$P_{x_e, d, x_e, d, k}^{(2)} = \begin{bmatrix} P_{x_d, x_d, k}^{(2)} & P_{x_d, x_{d,erw}, k}^{(2)} & 0 \\ P_{x_{d,erw}, x_d, k}^{(2)} & P_{x_{d,erw}, x_{d,erw}, k}^{(2)} & 0 \\ 0 & 0 & P^{(3)} \end{bmatrix} \quad (20)$$

$$P_{x_e, d, x_e, d, k}^{(3)} = \begin{bmatrix} P_{x_d, x_d, k}^{(3)} & 0 & P_{x_d, x_{d,erw}, k}^{(3)} \\ 0 & P^{(2)} & 0 \\ P_{x_{d,erw}, x_d, k}^{(3)} & 0 & P_{x_{d,erw}, x_{d,erw}, k}^{(3)} \end{bmatrix} \quad (21)$$

der differentiellen Zustandsgrößen, welche für die Reinitialisierung in (14) und (15) benötigt werden.

**Erweiterung der Dichtefunktionen mit Gleichverteilungen.** Erweiterung der Dichtefunktionen mit  $p^{(1,2)}(x_{d,erw}^{(2)}) = p^{(3,2)}(x_{d,erw}^{(2)}) = \prod_{i=1}^{n_{d,erw}^{(2)}} \mathcal{U}(x_{d,erw}^{(2)}; u_i^{(2)}, o_i^{(2)})$  und  $p^{(1,3)}(x_{d,erw}^{(3)}) = p^{(2,3)}(x_{d,erw}^{(3)}) = \prod_{j=1}^{n_{d,erw}^{(3)}} \mathcal{U}(x_{d,erw}^{(3)}; u_j^{(2)}, o_j^{(2)})$  ergibt die erweiterten Erwartungswerte

$$\hat{x}_{e,d,k|k}^{(1)} = \left[ \left( \hat{x}_{d,k|k}^{(1)} \right)^T, \left( \tilde{e}^{(2)} \right)^T, \left( \tilde{e}^{(3)} \right)^T \right]^T, \quad (22)$$

$$\hat{x}_{e,d,k|k}^{(2)} = \left[ \left( \hat{x}_{d,k|k}^{(2)} \right)^T, \left( \hat{x}_{d,erw,k|k}^{(2)} \right)^T, \left( \tilde{e}^{(3)} \right)^T \right]^T, \quad (23)$$

$$\hat{x}_{e,d,k|k}^{(3)} = \left[ \left( \hat{x}_{d,k|k}^{(3)} \right)^T, \left( \tilde{e}^{(2)} \right)^T, \left( \hat{x}_{d,erw,k|k}^{(3)} \right)^T \right]^T \quad (24)$$

und Kovarianzmatrizen

$$P_{x_e, d, x_e, d, k}^{(1)} = \begin{bmatrix} P_{x_d, x_d, k}^{(1)} & 0 & 0 \\ 0 & \tilde{P}^{(2)} & 0 \\ 0 & 0 & \tilde{P}^{(3)} \end{bmatrix}, \quad (25)$$

$$P_{x_e, d, x_e, d, k}^{(2)} = \begin{bmatrix} P_{x_d, x_d, k}^{(2)} & P_{x_d, x_{d,erw}, k}^{(2)} & 0 \\ P_{x_{d,erw}, x_d, k}^{(2)} & P_{x_{d,erw}, x_{d,erw}, k}^{(2)} & 0 \\ 0 & 0 & \tilde{P}^{(3)} \end{bmatrix}, \quad (26)$$

$$P_{x_e, d, x_e, d, k}^{(3)} = \begin{bmatrix} P_{x_d, x_d, k}^{(3)} & 0 & P_{x_d, x_{d,erw}, k}^{(3)} \\ 0 & \tilde{P}^{(2)} & 0 \\ P_{x_{d,erw}, x_d, k}^{(3)} & 0 & P_{x_{d,erw}, x_{d,erw}, k}^{(3)} \end{bmatrix} \quad (27)$$

der differentiellen Zustandsgrößen mit

$$\tilde{e}^{(\cdot)} = \frac{1}{2} [o_1^{(\cdot)} + u_1^{(\cdot)}, \dots, o_{n_{d,erw}}^{(\cdot)} + u_{n_{d,erw}}^{(\cdot)}]^T \quad (28)$$

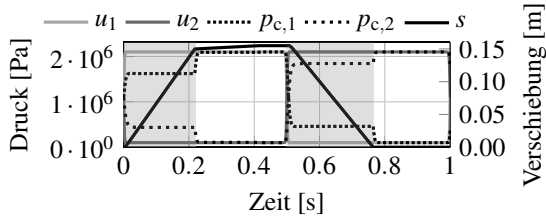
und

$$\tilde{P}^{(\cdot)} = \frac{1}{12} \text{diag}((o_1^{(\cdot)} - u_1^{(\cdot)})^2, \dots, (o_{n_{d,erw}}^{(\cdot)} - u_{n_{d,erw}}^{(\cdot)})^2). \quad (29)$$

**Triviale Erweiterung.** Der einfachste Ansatz (s. [1]) besteht in der Wahl  $e^{(1)} = 0, e^{(2)} = 0, P^{(1)} = 0, P^{(2)} = 0$  und  $\tilde{e}^{(1)} = 0, \tilde{e}^{(2)} = 0, \tilde{P}^{(1)} = 0, \tilde{P}^{(2)} = 0$ . Die entspricht einer Erweiterung der zugrundeliegenden Dichtefunktionen mit der diskreten Verteilung  $p^{(1,2)}(x_{d,erw}^{(2)}) = p^{(3,2)}(x_{d,erw}^{(2)}) = \delta(x_{d,erw}^{(2)})$  bzw.  $p^{(1,3)}(x_{d,erw}^{(3)}) = p^{(2,3)}(x_{d,erw}^{(3)}) = \delta(x_{d,erw}^{(3)})$ .

## 5 Numerische Ergebnisse

Betrachtet wird der in [3] beschriebene hydraulischer Zylinder, bei dem zwischen dem fehlerfreien Verhalten ( $m^{(ff)}$ ) und dem Verhalten bei interner ( $m^{(in)}$ ) sowie externer Leckage ( $m^{(ex)}$ ) zu unterscheiden ist. Die mathematische Modellierung sowohl des fehlerfreien Verhaltens als auch des Verhaltens bei Leckagen erfolgt, entsprechend der Vorgehensweise in Abschnitt 2, durch nichtlineare DAE-Systeme vom Index 1, wobei das Modell für den fehlerfreien Fall insgesamt 5 Differentialgleichungen und 41 algebraische Gleichungen umfasst. Eine detaillierte Beschreibung der Modellierung findet sich in [3]. Der Arbeitszyklus des Zylinders unterteilt sich in vier unterschiedliche Phasen, die für den fehlerfreien Fall in Fig. 1 dargestellt sind. Hierbei kennzeichnen  $u_1$  und  $u_2$  die Drücke in



**Fig. 1:** Verhalten des hydraulischen Zylinders im fehlerfreien Fall.

den Druckleitungen,  $p_{c,1}$  und  $p_{c,2}$  die Drücke in den korrespondierenden Kammern des Zylinders und  $s$  die Verschiebung der Kolbenstange.

Für acht aneinandergereihte Arbeitszyklen sind die Ergebnisse des IMM-Ansatzes bei der Modeschätzung in Fig. 2 und bei der Schätzung der Fehleramplitude in Fig. 3 dargestellt. Hierbei wurden die zugrunde liegenden Dichtefunktionen wie nachfolgend aufgeführt erweitert:

**Triviale Erweiterung.**

$$p^{(ff,in)}(x_{d,erw}^{(in)}) = p^{(ex,in)}(x_{d,erw}^{(in)}) = \delta(x_{d,erw}^{(in)}) \quad (30)$$

$$p^{(in,ex)}(x_{d,erw}^{(ex)}) = p^{(in,ex)}(x_{d,erw}^{(ex)}) = \delta(x_{d,erw}^{(ex)}) \quad (31)$$

**Erweiterung mit Normalverteilung.**

$$p^{(ff,in)}(x_{d,erw}^{(in)}) = p^{(ex,in)}(x_{d,erw}^{(in)}) = \quad (32)$$

$$\mathcal{N}(x_{d,erw}^{(in)}; \hat{x}_{d,erw,k|k}^{(ex)}, P_{x_{d,erw}^{(ex)} x_{d,erw}^{(ex)}}^{(ex)})$$

$$p^{(in,ex)}(x_{d,erw}^{(ex)}) = p^{(in,ex)}(x_{d,erw}^{(ex)}) = \quad (33)$$

$$\mathcal{N}(x_{d,erw}^{(ex)}; \hat{x}_{d,erw,k|k}^{(in)}, P_{x_{d,erw}^{(in)} x_{d,erw}^{(in)}}^{(in)})$$

**Erweiterung mit Gleichverteilung.**

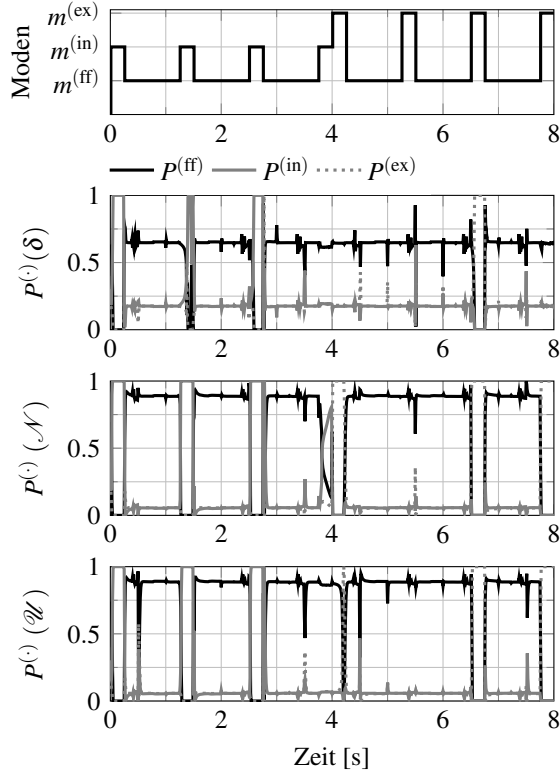
$$p^{(ff,in)}(x_{d,erw}^{(in)}) = p^{(ex,in)}(x_{d,erw}^{(in)}) = \quad (34)$$

$$\mathcal{U}(x_{d,erw}^{(in)}; \hat{x}_{d,erw,k|k}^{(ex)} - 0.5, \hat{x}_{d,erw,k|k}^{(ex)} + 0.5)$$

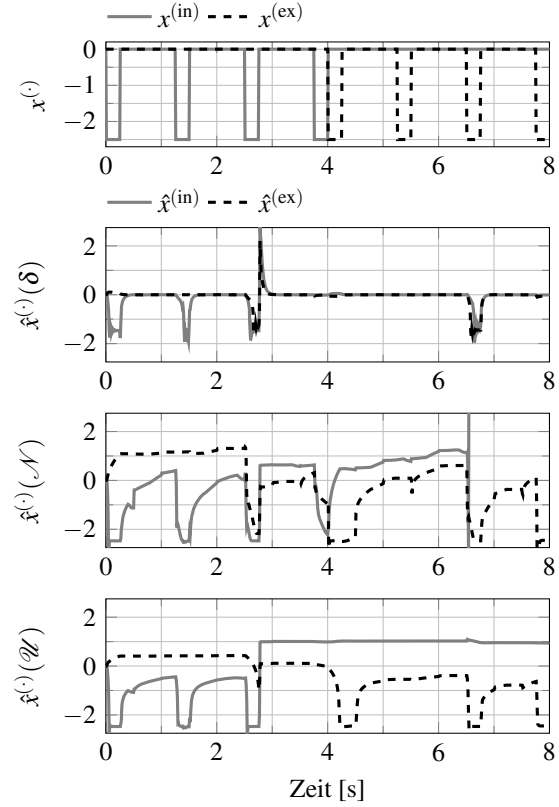
$$p^{(in,ex)}(x_{d,erw}^{(ex)}) = p^{(in,ex)}(x_{d,erw}^{(ex)}) = \quad (35)$$

$$\mathcal{U}(x_{d,erw}^{(ex)}; \hat{x}_{d,erw,k|k}^{(in)} - 0.5, \hat{x}_{d,erw,k|k}^{(in)} + 0.5)$$

Sowohl in Fig. 2 als auch Fig. 3 ist deutlich zu erkennen, dass die Erweiterung mit einer diskreten Verteilung weder zu einer zuverlässigen Moden- noch Amplitudenschätzung führt. Obgleich in [1] bei Erweiterung mit einer Gleichverteilung tendenziell bessere Ergebnisse erzielt werden, zeigen die Ergebnisse in Fig. 2 und Fig. 3, denen eine Erweiterung mit einer Normalverteilung bzw. Gleichverteilung zugrunde liegt, nur geringfügige Unterschiede bei der Moden- und Amplitudenschätzung. Dies resultiert aus der von [1] abweichenden Parametrierung der Gleichverteilung in (34) und (35). Während in [1] obere und untere Grenze der Gleichverteilung als Konstanten vorgegeben werden, erfolgt in (34) und (35) die Parametrierung über jeweils einen der Erwartungswerte der erweiterten differentiellen Zustandsgrößen. Mit konstanter Parametrierung kann nur dann eine zuverlässige Schätzung gewährleistet werden, wenn die Fehleramplitude in den Grenzen der Gleichverteilung liegt. Berücksichtigt man, dass im Kontext der Fehlerdiagnose vielfach keine a-priori Kenntnis der Fehleramplitude vorliegt, bietet die Parametrierung über die Erwartungswerte eine sinnvolle Alternative.



**Fig. 2:** Modewahrscheinlichkeiten  $P^{(\cdot)}(\delta)$  bei Erweiterung mit diskreter Verteilung,  $P^{(\cdot)}(\mathcal{N})$  bei Normalverteilung und  $P^{(\cdot)}(\mathcal{U})$  bei Gleichverteilung.



**Fig. 3:** Fehleramplituden  $\hat{x}^{(\cdot)}(\delta)$  bei Erweiterung mit diskreter Verteilung,  $\hat{x}^{(\cdot)}(\mathcal{N})$  bei Normalverteilung und  $\hat{x}^{(\cdot)}(\mathcal{U})$  bei Gleichverteilung.

## 6 Zusammenfassung

Wendet man klassische Ansätze zur hybriden Zustandsschätzung, wie den IMM-Ansatz, im Rahmen der Fehlerdiagnose an, weisen die einzelnen verwendeten mathematischen Modelle in der Regel unterschiedlich dimensionierte, heterogene Zustandsräume auf. Hinsichtlich einer Interaktion der einzelnen Filter bedarf es daher einer geeigneten Erweiterung der einzelnen, von den Filtern berechneten Wahrscheinlichkeitsdichtefunktionen (bzw. der daraus resultierenden Momente). Während die triviale Erweiterung mit einer diskreten Verteilung zu keiner zuverlässigen Schätzung führt, lassen sich bei einer Erweiterung mittels Normal- oder Gleichverteilung mit Hilfe des IMM-Ansatzes sowohl Mode als auch Fehleramplitude hinreichend präzise schätzen.

## Literatur

- [1] K.. Granstroem, P. Willett, and Y. Bar-Shalom. Systematic approach to imm mixing for unequal dimension states. *IEEE Transactions on Aerospace and Electronic Systems*, 51(4):2975–2986, 2015.
- [2] R. K. Mandela, R. Rengaswamy, and S. Narasimhan. Nonlinear state estimation of differential algebraic systems. *Advanced Control of Chemical Processes*, 7:792–797, 2009.
- [3] D. Weidemann and I. Alkov. Fault diagnosis of nonlinear differential-algebraic systems using hybrid estimation. In A. Rauh and L. Senkel, editors, *Variable-Structure Approaches*, pages 283–307. Springer, 2016.

# Simulation als epistemologische Grundlage für intelligente Roboter

Andreas Tolk<sup>1</sup>, Umut Durak<sup>2</sup>

<sup>1</sup>The MITRE Corporation

<sup>2</sup>Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)

*atolk@mitre.org*

Simulation spielt eine entscheidende Rolle für die Realisierung intelligenter Roboter. Ein Roboter muss seine Umgebung wahrnehmen, eine sinnvolle Perzeption aus der zu dem vorhandenen Wissen und der neuen Information bilden, unter möglicher Berücksichtigung von Beiträgen weiterer Roboter oder Anwender zu einer Entscheidung kommen, und diese Entscheidung in eine Aktivität umsetzen. Der vorliegende Beitrag zeigt auf, an welchen Stellen in diesem Kontrollkreis die Simulation zum Einsatz kommt, welche epistemologischen Einschränkungen den Einsatz der Simulation beeinflussen, und was dies für intelligente Roboter bedeutet.

## 1 Einleitung

Der vorliegende Beitrag fasst die wesentlichen Erkenntnisse aus zwei Gastvorträgen zusammen, die für die internationalen Arbeitstreffen der NATO zum Thema *“Modelling and Simulation for Autonomous Systems”* gehalten wurden [1, 2].

Zunächst wird die Topologie von Robotern vorgestellt, die einen einfachen Vergleich zu intelligenten Softwareagenten erlaubt, deren Topologie in [3] in gleicher Weise definiert wurde. Dieser Vergleich verdeutlicht die topologische Nähe von autonomen Robotern und intelligenten Softwareagenten, da sowohl die Komponenten, wie auch die durchzuführenden Schritte sehr ähnlich sind. Das Ergebnis dieses Vergleichs ist, dass die Algorithmen, die einen intelligenten Softwareagenten implementieren, auch die Implementierung eines intelligenten Roboters erlauben.

Kern dieser Algorithmen bilden die Eigenschaften, eine Situation zu erkennen, und auf Grund dieser Situation und eines gegebenen Zieles, eine Aktivität auszuwählen, die durchgeführt werden soll. Deshalb werden die Grundlagen der Maschinenerkennung beschrieben, die das Erkennen einer Situation und die Notwendigkeit zum Handeln erlauben [4]. Von besonderer Bedeutung ist hierbei, dass nicht nur die aktuelle Situation erkannt wird, sondern dass von dieser

auch auf zukünftige Entwicklungen extrapoliert werden kann. Dieser Abschnitt verdeutlicht, in welchen Schritten dieses Prozesses Simulationstechnik eine entscheidende Rolle spielt.

Im letzten Abschnitt werden die epistemologischen Einschränkungen der Simulation generell und in Bezug auf ihren Einsatz zur Implementierung von intelligenten Robotern verdeutlicht. Die Grenzen definieren, was grundsätzlich epistemologisch mit Simulationen verwirklicht werden kann. Auch wenn Simulation die beste Methode ist, welche die Implementierung eines intelligenten Roboters erlaubt, so gibt es dennoch Grenzen, die dem Wissenschaftler bekannt sein müssen.

## 2 Topologie von Robotern

In diesem Beitrag soll gezeigt werden, warum Simulationstechnik für die Implementierung intelligenter Roboter notwendig ist. Eine der unterstützenden Thesen in [1] ist, dass autonome Roboter und intelligente Softwareagenten topologisch und konzeptionell verwandt sind. Wenn diese Beziehung motiviert werden kann, dann können die wissenschaftlichen Erkenntnisse beider Anwendungsgebiete kombiniert werden.

Das *National Institute of Standards and Technology (NIST)* definiert Autonomie für Roboter wie folgt:

“The condition or quality of being self-governing. When applied to unmanned autonomous systems (UAS), autonomy can be defined as UAS’s own ability of integrated sensing, perceiving, analysing, communicating, planning, decision-making, and acting/executing, to achieve its goals as assigned by its human operator(s) through designed human-robot interface (HRI) or by another system that the UAS communicate with.” [5].

Autonomie beinhaltet somit die Fähigkeiten, seine Umwelt wahrzunehmen, eine Perzeption der aktuellen Situation zu erstellen, diese zu analysieren und mit anderen System zu kommunizieren, zu planen, zu einer Entscheidung zu kommen, und diese Entscheidung in Aktivitäten umzusetzen.

Ein Roboter benötigt damit die in Abbildung 1 gezeigten Komponenten, welche in [6] im Detail erklärt werden.

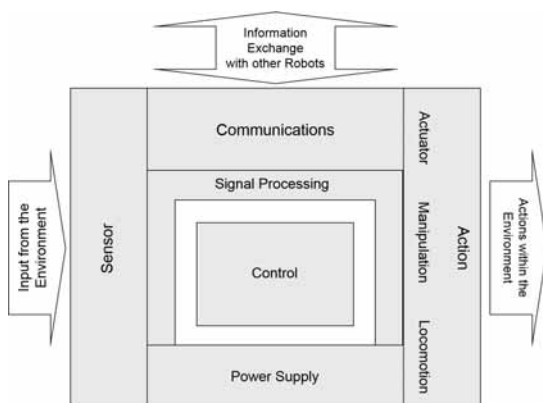


Abbildung 1: Topologische Struktur eines Roboters

- Roboter benötigen *Sensoren*, um ihre Umgebung wahrnehmen zu können.
- Sie benutzen *Kommunikationsmittel*, z.B. Radios, um mit anderen Systemen in Verbindung treten zu können.
- Sie setzen ihre Entscheidungen mit Hilfe von *Aktuatoren*, *Manipulatoren*, und *Lokomotoren* um. Aktuatoren platzieren Komponenten des Roboters in die notwendige Position, z.B. richten sie Antennen aus, platzieren einen Greifarm, usw. Manipulatoren interagieren mit Objekten der

Umwelt. Lokomotoren bewegen den Roboter. Dies können einfache Räder sein, oder die komplexen Elemente eines unbemannten, fliegenden Roboters.

- Alle diese Elemente sind über *Signalprozessoren* mit der *Steuerung* in der Kontroll- und Steuerungseinheit verbunden.
- Die *Energieversorgung* (Power Supply) versorgt alle Komponenten mit der notwendigen Energie, in der Regel in Form von Batterien, die Strom zur Verfügung stellen.

Intelligente Softwareagenten existieren in der virtuellen Welt, haben aber ähnliche Eigenschaften. Tolc und Uhrmacher [3] haben die wesentlichen Attribute eines Agenten nach einer umfangreichen Literaturrecherche zusammengefasst. Die resultierende Topologie ist in Abbildung 2 dargestellt.

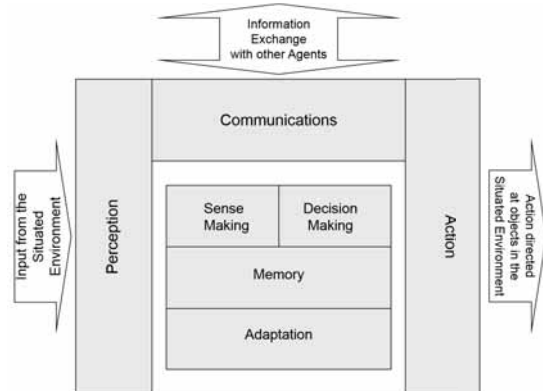


Abbildung 2: Topologische Struktur eines intelligenten Softwareagenten

- Intelligente Softwareagenten nehmen ihre Umgebung durch ein *Perzeptionsmodell* wahr.
- Über das *Kommunikationsmodell* können Softwareagenten mit anderen Agenten – oder auch mit dem Benutzer – Information austauschen.
- Das *Aktionsmodell* agiert mit der Umwelt.
- Das *Steuerungsmodell* eines intelligenten Softwareagenten ist etwas detaillierter erfasst.
  - Ein genereller *Speicher* (Memory) beinhaltet die Nutzwertfunktionen, deren Parameter, und andere Daten und Algorithmen, die benötigt werden.

- Dies erlaubt das Erkennen und Feststellen einer aktuellen *Situation* (Sense Making). Die Beurteilung der festgestellten Situation in Bezug auf die Ziele ist Teil des Prozesses.
- Das *Entscheidungsmodell* (Decision Making) wählt eine Aktion aus, die die erkannte Situation näher an die gewünschte Situation bringt. Nichts zu tun ist eine erlaubte Handlung.
- Ein wesentlicher Bestandteil der Lagebeurteilung der folgenden Zyklen ist festzustellen, ob die zuvor durchgeführten Aktionen erfolgreich waren. Ist dies der Fall, so wird in einer ähnlichen Situation die Auswahl der gleichen Aktion mehr wahrscheinlich. Sollte dies nicht der Fall sein, ist die Auswahl eher unwahrscheinlich. Dies wird über das Adaptionsmodell geregelt.

Sowohl autonome Roboter, als auch intelligente Softwareagenten, nehmen somit ihre Umgebung über Sensoren wahr, kommunizieren mit anderen über die Perzeption und können an einem koordinierten Plan arbeiten, und setzen diesen Plan in Form von Aktivitäten um.

### 3 Grundlagen der Maschinenerkennung

Zeigler [4] beschreibt die Grundlagen der Maschinenerkennung als einen Prozess mit vier Phasen, die in Abbildung 3 schematisch erfasst sind. Ziel für das beobachtende System auf der rechten Seite ist, das beobachtete Objekt auf der linken zu erkennen, und dessen Möglichkeiten zu Handeln richtig einzuschätzen.

- In der Phase (1) muss das beobachtende System zunächst mit Hilfe der Sensoren seine Umwelt erkennen. Um das zu erkennende Objekt mit den Sensoren erfassen zu können, muss dieses Attribute besitzen, deren Eigenschaften von den Sensoren aufgenommen werden können. Weiterhin müssen diese Eigenschaften hinreichend verschieden von denen der Umgebung sein, so dass eine Diskriminierung zwischen Objekt und Umwelt möglich ist.

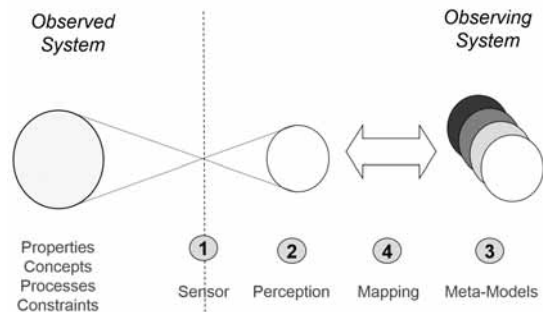


Abbildung 3: Logische Schritte der Maschinenerkennung

- In Phase (2) werden die beobachteten Werte in eine Perzeption eingebettet. Alle beobachteten Attribute werden in das Perzeptionsmodell aufgenommen. Wenn das Modell nicht alle Beobachtungen unterstützt, dann können nur die abgebildeten Daten berücksichtigt werden. Das der Perzeption zugrundeliegende Datenmodell ist daher sehr wichtig.
- In der Phase (3) wird nun ein Metamodell ausgewählt, das die beobachteten attributierten Konzepte am besten widerspiegelt. Dieses Metamodell beinhaltet nicht nur die statischen Eigenschaften der Interpretation der Beobachtung, sondern auch die dynamischen Eigenschaften.
- Die letzte Phase (4) stellt eine Verbindung zwischen dem beobachteten Objekt und seiner Interpretation durch das Metamodell her. Die Maschine hat nun ein Objekt erkannt, und kann vorhersagen, was dieses Objekt in der aktuellen Situation tun kann, da diese Optionen mit dem Metamodell verknüpft sind.

Maschinenerkennung ist somit die Verknüpfung einer Beobachtung über ein statisches Datenmodell mit einem dynamischen Metamodell.

Eine Maschine kann lernen, wenn die Familie der Metamodelle modifiziert werden kann. Einfache Versionen erlauben die Kalibrierung der Metamodelle, um neue Objekte erkennen zu können. Ein Beispiel hierfür ist die generelle Beschreibung eines Autos mit beobachtbaren Attributen, und dann die Population verschiedener Automarken mit den charakteristischen Parametern. Kompliziertere Lernalgorithmen erlauben, auch neue Metamodelle zu definieren, was aber in der Regel einen kreativen Prozess erfordert.

Diese Grundlagen der Maschinenerkennung spiegeln die wissenschaftliche Methode wieder. Die Metamodelle repräsentieren bekannte Theorien. Die Beobachtung der Objekte entspricht einem Experiment. Wenn die Beobachtung durch eine der Theorien erfasst ist, dann kann diese dem entsprechenden Metamodell zugeordnet werden. Ist dies nicht der Fall, so muss entweder eine bestehende Theorie erweitert, oder eine neue Theorie entwickelt werden.

Roboter erkennen Objekte in ihrer Umwelt nach dem gleichen Prinzip. Oftmals wird dabei übersehen, dass die Metamodelle in einer Form implementiert sein müssen, die den Prinzipien der Simulation entsprechen, wenn die dynamischen Eigenschaften der erkannten Objekten berücksichtigt werden sollen. Simulation spielt damit eine wesentliche Rolle, um intelligente Roboter zu ermöglichen.

Wann immer eine Vorhersage der zukünftigen Entwicklung der Situation notwendig ist, muss eine Simulation durchgeführt werden. Wann immer der Vergleich zweier Alternativen durchgeführt werden sollen, müssen diese Alternativen simuliert werden. Interpolation und Extrapolation werden durch Anwendung von Simulationstechniken ausführbar.

Simulationstechnik ermöglicht es, dynamisches Wissen für die Roboterkontrolle zur Verfügung zu stellen. Es ist deshalb besonders wichtig, sich über die epistemologischen Grenzen im Klaren zu sein, die in dem nächsten Abschnitt behandelt werden.

## 4 Epistemologische Grenzen

Epistemologie – auch Erkenntnistheorie genannt – ist der Zweig der Wissenschaftsphilosophie, der sich mit der Frage beschäftigt, wie Erkenntnis und Wissen abgeleitet werden können. In Zusammenhang mit Simulationstechniken ist die Frage der Epistemologie: Kann die Anwendung von Simulation zu neuen Erkenntnissen oder neuem Wissen beitragen? Da wir im letzten Abschnitt die Wichtigkeit der Simulationstechnik zur Erfassung der dynamischen Aspekte der Umweltwahrnehmung und -einschätzung dargestellt haben, ist diese Frage unmittelbar für die Implementierung intelligenter Roboter von Bedeutung.

Die Hierarchie von Daten (**D**ata) zu Information (**I**nformation) zu Wissen (**K**nowledge) zu Weisheit (**W**isdom) wurde von Ackoff in form der DIKW Pyramide vorgestellt [7]. Daten sind faktische und punk-

tuelle Beobachtungen von Attributen. Durch das Verständnis der Beziehungen – generell in Form eines Datenmodells, z.B. das vorgestellte Perzeptionsmodell – werden die Daten attribuierten Konzepten zugeordnet, was zu Information führt. Die Zuordnung von Metamodellen entspricht dem Verständnis von Mustern in einem ausführbaren Kontext, was in Wissen resultiert. Wenn nun mehrere Alternativen untersucht werden können, so führt dies zum Verständnis von Grundsätzen, oder auch Weisheit. Modellbildung und Simulation spielen somit eine entscheidende Rolle um vom Boden der DIKW Pyramide bis an ihre Spitze zu kommen.

Nichtsdestotrotz muss bedacht werden, dass Simulation nur einen Teil der Realität repräsentiert, da sie auf einem Modell basiert, welches eine zielgerichtete, bewusste Vereinfachung und Abstraktion einer Perzeption der Realität ist. Die Perzeption ist durch physikalische, kognitive, und legale/moralische Einschränkungen definiert:

- *Zielgerichtet:* Eine Simulation soll eine Aufgabe erfüllen, z.B. eine wissenschaftliche Frage beantworten, oder eine Trainingsvorgabe erfüllen.
- *Bewusst:* Modellierung ist eine kreative, gezielte Tätigkeit, um die Simulationsaufgabe zu ermöglichen.
- *Vereinfachung:* Alle Konzepte und Beziehungen, die zur Erreichung der Simulationsaufgabe nicht als nötig erachtet werden, werden im Rahmen der Modellbildung identifiziert und eliminiert.
- *Abstraktion:* Alle Konzepte und Beziehungen, die zur Erreichung der Simulationsaufgabe notwendig sind, werden auf dem angemessenen Abstraktionslevel modelliert.
- *Perzeption der Realität:* Unser Verständnis der Realität ist von vielen Faktoren abhängig. Dies gilt insbesondere auch für Teams und individuelle Teammitglieder. Zwei Experten können sehr unterschiedliche Ansichten von einem zu modellierenden Problem haben.
- *Physikalische Einschränkungen:* Nicht alle Attribute, die wir beobachten möchten, können von unseren Sensoren erfasst werden.
- *Kognitive Einschränkungen:* Die Ausbildung und Erfahrungen des modellierenden Experten bestimmt, wie dieser das Problem versteht. Auch

die Modellierungsmethode kann beeinflussen, was der Experte wie wahrnimmt, bevor er es in das Modell umsetzt.

- *Legale/moralische Einschränkungen:* Anwendbare Gesetze oder auch moralische Wertvorstellungen können der Erfassung von anderweitig beobachtbaren Werten entgegenstehen.

Das resultierende Modell wird als Simulation implementiert. Es bildet die konzeptionelle Grundlage, was die Simulation später erkennen und anwenden kann. Ein Objekt, das nicht als Konzept modelliert wurde, ist nicht Teil der Realität, wie sie sich der Simulation darstellt. Eine Kausalität, die im Model als Funktion von Eingabewerten zu Ausgabewerten dargestellt ist, bildet die Basis aller kausalen Zusammenhänge, deren sich die Simulation bewusst sein kann. Als solches kann eine Simulation als ausführbare Theorie angesehen werden. Ist die Theorie solide, sollte sie zu keinen Widersprüchen beim Einsatz in der Realität führen.

Desweiteren ist zu beachten, dass Computersimulationen den Grenzen allgemeiner Computerprogramme unterliegen. Der Gödelsche Unvollständigkeitssatz zeigt die Grenzen der Ableitbarkeit von Aussagen in formalen Systemen auf [8]. Zur gleichen Zeit zeigte Turing die Grenzen der Probleme auf, die mit einem Computerprogramm gelöst werden können [9]. Deterministisches Chaos impliziert die grundsätzliche Unmöglichkeit, langfristige Vorhersagen zum Zustand eines chaotischen Systems mit einer Simulation machen zu können. Oberkampff und Kollegen [10] geben zusätzliche Beispiele, in denen computergenerierte Probleme die Vertrauenswürdigkeit von Computersimulationsergebnissen einschränken.

Computersimulationen können keine zusätzlichen Probleme lösen, die mit einem anderen Computerprogramm nicht lösbar sind.

## 5 Abschließende Diskussion

Nach all diesen kritischen Anmerkungen stellt sich die Frage, ob und in wie weit Simulation eine epistemologische Grundlage für Roboter bilden können? Die folgende Abbildung wurde zuerst in [3] veröffentlicht, um intelligente Softwareagenten in ihrer virtuellen Umgebung zu verdeutlichen.

Wie bereits in der topologischen Diskussion gezeigt, ist eine Anwendung auf intelligente Roboter möglich:

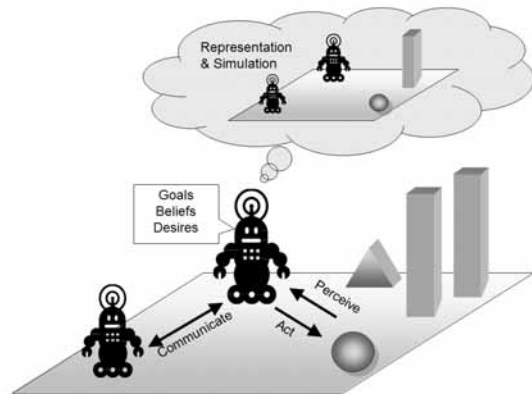


Abbildung 4: Simulation und intelligente Roboter

- Roboter nehmen über ihre Sensoren die Umwelt wahr (perceive).
- Sie kommunizieren mit anderen Robotern – oder auch Benutzern – und tauschen Informationen aus.
- Mit Hilfe der Sensorendaten und der Kommunikation erstellen Sie eine interne Interpretation der Situation. Diese enthält
  - eine Abbildung der beobachteten Objekte und ihrer Attribute,
  - eine Interpretation der Objekte mit Hilfe der Metamodelle, die dem beobachteten Objekt am ähnlichsten sind, und
  - eine Interpretation, wie sich die Situation entwickeln kann, über Simulation verschiedener Möglichkeiten des Handelns.
- Die Perzeption kann unvollständig sein, da nicht alle Objekte gesehen werden, oder auch wenn das Perzeptionsmodell nicht mächtig genug ist.
- Der Roboter agiert in seiner Umgebung (act), so dass er seine Ziele erreichen kann.

Simulationen spielen eine entscheidende Rolle in allen diesen Schritten. Das Perzeptionsmodell ist das Datenmodell, auf dem die internen Simulationen arbeiten. Nur was perzipiert wird, kann auch simuliert werden. Die Metamodelle sind Objektmodelle, die neben den Daten, die die Konzepte beschreiben, auch Methoden bereitstellen, die die Dynamik der Konzepte beschreiben. Um eine Aktion auszuwählen, werden die Auswirkungen und Effekte intern simuliert. Die internen Bewertungs-

oder Nutzwertfunktionen werden auf die Ergebnisse der Simulationen angewandt, um die beste Option auszuwählen.

Intelligente Roboter reagieren nicht nur, sie agieren, getrieben von ihren Zielen. Dies ist ohne Simulation nicht möglich. Lernen entspricht der Optimierung von Metamodellen zu den Beobachtungen, Verbesserung der Metamodelle und der Nutzwertfunktionen durch Kalibrierung, oder durch zufügen von neuen Metamodellen oder Nutzwertfunktionen. Die letzte Option verlangt allerdings einen kreativen Prozess, oder die vollständige, parametrische Abdeckung des Lösungsraumes, was nur in trivialen Umgebungen möglich ist.

Trotz der epistemologischen und computerbedingten Einschränkungen bieten Computersimulationen die beste Unterstützung für die Realisierung intelligenter Roboter. Solange das zugrundeliegende Modell eine plausible und angemessene Theorie darstellt, gibt es zur Zeit keine bessere Alternative. Allerdings muss sichergestellt sein, dass alle relevanten Attribute nicht nur über die Sensoren erfasst werden, sondern auch in das Perzeptionsmodell abgebildet werden können. Die Zahl der Metamodelle muss hinreichend groß sein, und das Abstraktionslevel muss der Problemstellung angemessen sein. Die implementierte Kausalität muss mit empirischen Beobachtungen korrelieren. Die Nutzwertfunktionen müssen die Ziele angemessen darstellen. Diese Voraussetzung sind nicht neu, sondern stellen die allgemeinen Voraussetzung für wissenschaftliche Nutzung von Modellen dar. Verifizierte Simulation von validierten Modellen ist derzeit die mächtigste Form der ausführbaren Wissensdarstellung.

Zusammenfassend kann festgestellt werden, dass die Implementierung einer Kontrollfunktion für intelligente Roboter in jedem Fall Simulationstechniken nutzen muss. Da die Kontrollmodule oftmals von Experten geschrieben werden, die keine Simulationsexperten sind, ist es durchaus möglich, dass Begriffe wie Interpolation oder Extrapolation genutzt werden, und damit der Einsatz von Simulation verschleiert wird. Nichtsdestotrotz sind dies Simulationstechniken, und der Experte sollte sich der Möglichkeiten, wie auch der Einschränkungen, wie sie in diesem Artikel beschrieben wurden, bewusst sein.

## References

- [1] A. Tolk. *Merging two Worlds: Agent-based Simulation Methods for Autonomous Systems*. In: Systems with Autonomous Capabilities: Issues for Defence Policy Makers, edited by A. P. Williams, and P. D. Scharre, NATO Headquarters Supreme Allied Command Transformations (HQ SACT), Norfolk, VA: Innovations in Capability Development Publication Series, pp. 291-317, 2015.
- [2] A. Tolk. *Modeling and Simulation Interoperability Concepts for Multidisciplinarity, Interdisciplinarity, and Transdisciplinarity – Implications for Computational Intelligence enabling Autonomous Systems*. Proceedings of the Modelling and Simulation for Autonomous Systems Workshop 2015, Lecture Notes in Computer Science, Vol. 9055, Springer Verlag, pp. 60-74, 2015.
- [3] A. Tolk, A. M. Uhrmacher. *Agents: Agenthood, Agent Architectures, and Agent Taxonomies*. In: Agent-Directed Simulation and Systems Engineering, edited by L. Yilmaz and T. Ören. Wiley-Berlin, pp. 75-109, 2009
- [4] B. P. Zeigler. *Toward a Simulation Methodology for Variable Structure Modeling*. In: Modeling and Simulation Methodology in the Artificial Intelligence Era, edited by M. S. Elzas, T. Ören, and B. P. Zeigler, North Holland, Amsterdam, pp. 195-210, 1986.
- [5] F. Kendoul. *Towards a Unified framework for UAS Autonomy and Technology Readiness Assessment (ATRA)*. In: Autonomous Control Systems and Vehicles: Intelligent Unmanned Systems, edited by K. Nonami, M. Kartidjo, K.-J. Yoon, and A. Budiyo, Springer, Tokyo, pp. 55-72, 2013.
- [6] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza. *Introduction to Autonomous Mobile Robots, 2nd Edition*. MIT Press, Cambridge, MA, 2011.
- [7] R. L. Ackoff. *From data to wisdom*. Journal of Applied Systems Analysis 15 (1989):3-9.
- [8] K. Gödel. *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*. Monatshefte für Mathematik und Physik 38 (1931):173–198.

- [9] A. Turing. *On Computable Numbers, with an Application to the Entscheidungsproblem*. Proceedings of the London Mathematical Society 2 (42) (1936):230–265.
- [10] W. L. Oberkampff, S. M. DeLand, B. M. Rutherford, K. V. Diegert, and K. F. Alvin. *Error and Uncertainty in Modeling and Simulation*. Reliability Engineering & System Safety 75 (3) (2002): 333-357.

## Bemerkungen

Die Autoren danken Thorsten Pawletta für die Diskussionen und redaktionellen Unterstützungen im Rahmen der Erstellung dieses Beitrages.

Dieses Dokument wurde von der MITRE Corporation zur Veröffentlichung und uneingeschränkten Verteilung freigegeben. Die entsprechende MITRE PRS Vorgangsnummer ist **17-0085**.

Andreas Tolk's Angestelltenverhältnis zur MITRE Corporation ist ausschließlich für Identifikationszwecke angeben und impliziert keinerlei Zustimmung der MITRE Corporation zu Positionen, Meinungen, oder Ansichten, die in diesem Beitrag dokumentiert sind.

# Entwurf, Simulation und Implementierung ereignisdiskreter Steuerungen mit PDEVS<sub>RCP</sub> Version 2.0

Birger Freymann<sup>1</sup>, Artur Schmidt<sup>1</sup>, Sven Pawletta<sup>1</sup>, Sven Hartmann<sup>2</sup>, Thorsten Pawletta<sup>1</sup>

<sup>1</sup>Hochschule Wismar - University of Applied Sciences, Research Group CEA, Wismar, Germany

<sup>2</sup>Clausthal University of Technology, Clausthal-Zellerfeld, Germany

*birger.freymann@hs-wismar.de*

In der Steuerungsentwicklung haben sich Entwurfsmethodiken wie der *Rapid Control Prototyping* (RCP) Ansatz etabliert. Die Steuerungssoftware (CS) wird durchgehend modellbasiert in möglichst einer Softwareumgebung entwickelt. Beginnend in der frühen Entwurfsphase, wird ein *Simulationsmodell* (SM) schrittweise zu einer CS ausgebaut. Die Überführung in eine ausführbare CS erfolgt beim RCP Ansatz entweder durch eine automatische Codegenerierung für die Zielplattform oder durch den Einsatz eines Steuerungsrechners mit entsprechenden Schnittstellen zur Prozessumgebung (implizite Codegenerierung). Letzteres ist auch als *Software in the Loop* (SiL) Ansatz bekannt. Im Beitrag wird die RCP gestützte Entwicklung ereignisdiskreter Steuerungen auf Basis des *Parallel Discrete Event System* (PDEVS) Formalismus untersucht. Der PDEVS Formalismus definiert eine systemtheoretische Spezifikation sowie Ausführungsalgorithmen für ereignisdiskrete, dynamische Systeme. Hinsichtlich des durchgängigen Einsatzes der Modelle bis zum operativen Betrieb unter Echtzeitbedingungen wurde der PDEVS konforme PDEVS<sub>RCP</sub> Formalismus definiert. Jedoch zeigte sich im Rahmen verschiedener Forschungsprojekte, dass bei komplexen Problemen umfangreiche Adaptionen der Modelle beim Übergang von der Entwurfs- zur Automatisierungsphase erfolgen müssen. Aus diesem Grund wurde aufbauend auf PDEVS<sub>RCP</sub> eine neue PDEVS<sub>RCP</sub> Version 2.0 entwickelt. Im Beitrag wird der theoretische Ansatz von PDEVS<sub>RCP</sub> 2.0 vorgestellt und die Vorteile anhand eines Demonstrationsbeispiels aufgezeigt.

## 1 Einleitung

Die Entwicklung komplexer Steuerungen ist zumeist aufwändig und fehleranfällig, sodass Ad-hoc-Implementierungen nicht möglich sind. Um eine durchgängige Steuerungsentwicklung zu ermöglichen, haben sich Entwurfsmethoden wie der *Rapid Control Prototyping* (RCP) Ansatz nach [1] etabliert. Das Ziel von RCP ist die durchgängige Steuerungsentwicklung von der frühen Entwurfsphase, über die Automatisierungsphase, bis zum operativen Betrieb, unter frühzeitigem Einsatz von Simulationsmodellen. Im Fokus stehen die möglichst durchgehende Verwendung kompatibler Softwarepakete sowie theoretisch fundierter und umfassend modellbasierter Entwicklungsmethoden. Die manuelle und fehleranfällige Re-Implementierung von Steuerungsalgorithmen wird vermieden und eine frühzeitige Fehlererkennung ermöglicht. Steuerungsentwürfe können sukzessive simulationsbasiert getestet und um weitere Anforderungen ergänzt werden. Dafür erfordert RCP aufeinander abgestimmte Softwarewerkzeuge oder eine integrierte Entwicklungsumgebung. Für die Entwurfsphase stehen zumeist unterschiedliche Softwarewerkzeuge zur Verfügung, während die Implementierung der Algorithmen auf der ausgewählten Zielhardware durch eine automatische

explizite Codegenerierung erfolgen kann. Eine Alternative zur expliziten Codegenerierung bildet der *Software in the Loop Ansatz* (SiL). Beim SiL Ansatz wird der Entwicklungsrechner oder ein Industrie-PC unmittelbar zum Steuern realer Prozesse verwendet. Dazu ist ein entsprechendes Prozessinterface zum realen Prozess zu implementieren. Diese Form der Kommunikation mit einem realen Prozess wird auch als *implizite Codegenerierung* [2] bezeichnet. Im Folgenden soll die durchgängige Steuerungsentwicklung nach dem SiL Ansatz, mit dem *Discrete Event System* (DEVS) Formalismus nach [3] untersucht werden. Der DEVS Formalismus definiert eine systemtheoretische Spezifikation und Ausführungsalgorithmen für ereignisdiskrete, dynamische Systeme. Eine Erweiterung der DEVS Spezifikation bezüglich Echtzeitfähigkeit und Prozessanbindung ist der *Real Time DEVS* (RT-DEVS) Formalismus. Dieser unterscheidet sich bezüglich DEVS durch eine modifizierte Modellspezifikation und -abarbeitung [3,4]. Weitere DEVS basierte Steuerungsansätze werden z.B. in [5] erwähnt. Bei den meisten Ansätzen ist für eine Prozessanbindung im Entwicklungsprozess ein Austausch des Simulators notwendig, was wiederum umfangreiche Modella-daptionen erfordert. Wie auch bei RT-DEVS erfolgt die

Interaktion mit der zu steuernden Prozessumgebung durch Erweiterungen im Simulator.

Hinsichtlich eines durchgängigen Einsatzes der Modelle bis zum operativen Betrieb unter Echtzeitbedingungen wurde in [6] der PDEVSRCP Formalismus definiert. Dieser entspricht einer Erweiterung des *Parallel DEVS* (PDEVs) Formalismus [7], um RCP Fähigkeit und ermöglicht eine durchgängige Steuerungsentwicklung über alle Entwicklungsphasen. PDEVs Modelle werden hierbei schrittweise erstellt, getestet und um Anforderungen ergänzt. Beim Übergang zur Automatisierungsphase werden PDEVs Modelle gemäß dem PDEVSRCP Formalismus um ein Prozessinterface zum realen Prozess erweitert. Die Interaktion mit der Prozessumgebung ist vollständig auf Modellebene spezifiziert, so dass PDEVSRCP Modelle mit einem klassischen PDEVs Simulator ausführbar sind. Somit kann ein entwickeltes *Simulationsmodell* (SM) direkt nach dem SiL Ansatz als *Steuerungssoftware* (CS) verwendet werden. Der Einsatz von PDEVSRCP im Rahmen verschiedener Forschungsprojekte [8,9] zeigte, dass bei komplexen Problemen umfangreiche Adaptationen der Modelle beim Übergang von der Entwurfs- zur Automatisierungs- und Betriebsphase erfolgen müssen. Aus diesem Grund wurde aufbauend auf PDEVSRCP eine neue Version 2.0 entwickelt. Im Beitrag wird der theoretische Ansatz des PDEVs Formalismus und des hierauf aufbauenden PDEVSRCP Formalismus, vorgestellt. Anschließend erfolgt eine Diskussion offener Probleme, des PDEVSRCP Formalismus sowie die Weiterentwicklung zu einer neuen Version 2.0. Abschließend werden beide PDEVSRCP Entwicklungen anhand eines Demonstrationsbeispiels gegenübergestellt und die Ergebnisse zusammengefasst.

## 2 Grundlagen des PDEVs Formalismus

Der *Parallel DEVS* (PDEVs) Formalismus [7] ist eine Weiterentwicklung des DEVS Formalismus und definiert neue Mechanismen zur Verarbeitung zeitgleicher Ereignisse. DEVS wurde bereits 1976 von Zeigler eingeführt und ist eine ereignisorientierte Modellierungs- und Simulationstechnik [3]. DEVS basiert auf modular hierarchischen Modellspezifikationen und zugehörigen Simulatoralgorithmen. Der DEVS Formalismus geht von einer eindeutigen Trennung zwischen Modellspezifikation und Modellbearbeitung aus. Bei der Modellspezifikation wird zwischen zwei DEVS Systemtypen unterschieden. Das dynamische Verhalten wird mit atomaren (atomic) DEVS Systemen abgebildet. Daneben gibt es gekoppelte (coupled) DEVS

Systeme, welche eine Komposition aus atomic beziehungsweise coupled DEVS Systemen beschreiben. Beide Systemtypen, atomic und coupled DEVS, definieren kompatible Ein- und Ausgänge. Die Spezifikation der Systemdynamik in atomic DEVS erfolgt mit einer Menge festgelegter Funktionen.

Ein atomic PDEVs ist in [3] formal definiert als:

$$\text{PDEVs} = (X, Y, S, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta) \quad (1)$$

Dabei entspricht  $X$  der Eingangsmenge,  $S$  der Zustandsmenge und  $Y$  der Ausgangsmenge. Weiterhin definiert PDEVs drei Zustandsüberföhrungsfunktionen.  $\delta_{int}$  entspricht der *internen* und  $\delta_{ext}$  der *externen Zustandsüberföhrungsfunktion*.  $\delta_{con}$  definiert die *confluente Zustandsüberföhrungsfunktion*, welche beim zeitgleichen Auftreten von internen und externen Ereignissen ausgeföhrt wird.  $\lambda$  repräsentiert die *Ausgabefunktion* und  $ta$  die *Zeitfortschrittsfunktion*.

Ein coupled PDEVs oder auch PDEVs-Netzwerk (PDEVN) ist in [3] formal definiert als:

$$\text{PDEVN} = (X, Y, D, \{M_d\}, \{I_d\}, \{Z_{i,d}\}) \quad (2)$$

$X, Y$  sind analog PDEVs definiert.  $D$  entspricht der Indexmenge der Subkomponenten.  $\{M_d\}$  spezifiziert die Menge der atomaren oder gekoppelten Subsysteme mit  $d \in D$ .  $I_d$  ist die Indexmenge (Namen) der auf  $d \in D$  einwirkenden Systeme, mit  $I_d \subseteq D \cup \{N\}$ , wobei  $\{N\}$  den Namen des coupled PDEVs repräsentiert.  $Z_{i,d}$  ist die Menge der *i-zu-d*-Kopplungsbeziehungen mit  $i \in I_d$ .

Basierend auf dem PDEVs Formalismus wird nachfolgend eine RCP konforme Erweiterung nach [6], zur durchgängigen Steuerungsentwicklung vorgestellt.

## 3 Der PDEVSRCP Formalismus zur Umsetzung des SiL Ansatzes

Die RCP gestützte durchgängige Steuerungsentwicklung nach dem SiL Ansatz erfordert, dass ein SM in der Automatisierungsphase um die Möglichkeit zur Interaktion mit externen Soft- oder Hardwarekomponenten erweitert wird. Um ein entsprechendes Prozessinterface zu implementieren wurde der PDEVSRCP Formalismus entwickelt. Da PDEVSRCP unmittelbar auf PDEVs aufbaut, können PDEVSRCP Modelle direkt mit einer PDEVs-Simulationsumgebung ausgeföhrt werden, wobei diese aufgrund der eingeföhrten Erweiterungen auch als Echtzeitumgebung für SiL genutzt werden kann. Dadurch ist es möglich, PDEVs Modelle schrittweise von der Entwurfsphase bis zum operativen Steuerungsbetrieb zu erweitern und mit einer durchgehenden Simulationsumgebung auszuföhren.

Diese Durchgängigkeit wird von der RT-DEVS-Erweiterung in [4] nicht erfüllt, da hier zur Umsetzung des SiL Ansatzes sowohl die Modellspezifikation als auch die Simulatoralgorithmen modifiziert werden.

Ein atomic PDEVS<sub>RCP</sub> ist nach [6] formal definiert als:

$$\text{PDEVS}_{\text{RCP}} = (X, Y, S, \delta_{\text{int}}, \delta_{\text{ext}}, \delta_{\text{con}}, \lambda, ta, A) \quad (3)$$

$$X = X_{\text{model}} \cup X_{\text{clock}}$$

$$X_{\text{model}} = \{(p, v) | p \in \text{InPorts}, v \in X_P\}$$

$$X_{\text{clock}} = \{('clock', v) | v \in \mathbb{R}^+\}$$

$$\lambda: S \rightarrow Y \times A$$

Die Definitionen von  $Y, S, \delta_{\text{int}}, \delta_{\text{ext}}, \delta_{\text{con}}, ta$  sind analog PDEVS. Die neu eingeführte Menge  $A$  spezifiziert die Interaktionen mit der Umgebung. Für jede Interaktion ist ein zulässiges Zeitintervall  $[t_{\text{min}}, t_{\text{max}}]$  anzugeben, welches die Echtzeitanforderungen festlegt. Aufgrund des Ausführungsintervalls wird eine Interaktion als Aktivität  $a \in A$  bezeichnet. Die Menge der Eingangsereignisse  $X$  setzt sich aus den Mengen  $X_{\text{model}}$  und  $X_{\text{clock}}$  zusammen. Dabei umfasst  $X_{\text{model}}$  die Menge der Eingangsereignisse gewöhnlicher PDEVS Modelle. Die Eingangs Menge  $X_{\text{clock}}$  wird von einer *Echtzeituhr* (Real Time Clock, RTC) generiert. Die Ausgabefunktion  $\lambda$  definiert die Bindung der aktuellen Aktivität  $a \in A$  an den aktuellen Zustand  $s \in S$ . Analog dazu wird auch das zulässige Zeitintervall der aktuellen Aktivität  $a \in A$  im aktuellen Zustand  $s \in S$  abgebildet. Das dynamische Verhalten eines atomic PDEVS<sub>RCP</sub> Modells zeigt Abbildung 1.

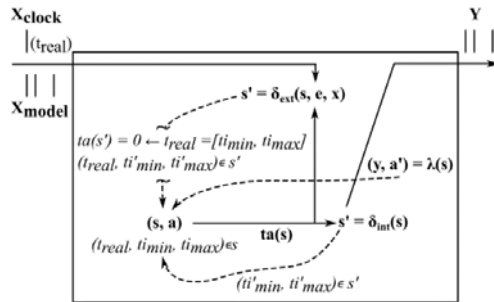


Abbildung 1: Dynamik eines PDEVS<sub>RCP</sub> nach [6].

Demgemäß muss die als externes Ereignis  $('clock', v) \in X_{\text{clock}}$  empfangene Realzeit durch  $\delta_{\text{ext}}$  als Zustand gespeichert werden. Auf Basis der so gespeicherten Realzeit kann das zulässige Zeitintervall der aktuellen Aktivität  $a \in A$  in Realzeit umgerechnet werden, welches ebenfalls im aktuellen Zustand  $s \in S$  gespeichert wird. Weiterhin zeigt Abbildung 1, dass eine Aktivität  $a \in A$  durch die Ausgabefunktion  $\lambda$  beendet wird und diese eine neue Aktivität  $a' \in A$  startet. Das zugehörige Zeitintervall wird in Echtzeitwerten durch  $\delta_{\text{int}}$  definiert. Alle externen Ereignisse  $x \in X$  werden

bei PDEVS<sub>RCP</sub> analog zu PDEVS durch die externe Zustandsüberföhrungsfunktion  $\delta_{\text{ext}}(s, e, x)$  ausgewertet. Diese bestimmt auf Grundlage des aktuellen Zustands  $s$ , der verstrichenen Zeit  $e$  seit dem letzten Ereignis und den aktuellen externen Ereignissen  $x$  den nachfolgenden Zustand  $s'$ . Darüber hinaus wird innerhalb der externen Zustandsüberföhrungsfunktion geprüft, ob die aktuellen Ereignisse  $x \in X_{\text{clock}}$  innerhalb ihres Zeitintervalls  $[t_{\text{min}}, t_{\text{max}}]$  eintreten. Dabei werden nach [6] zwei Fälle unterschieden.

1. Falls externe Ereignisse innerhalb des Zeitintervalls eintreten, muss der Folgezustand  $s'$  zum virtuellen Zeitfortschritt  $ta(s)=0$  führen und somit unmittelbar ein internes Ereignis auslösen.

2. Falls externe Ereignisse außerhalb des Zeitintervalls eintreten, wird durch  $\delta_{\text{ext}}$  der Folgezustand  $s'$  berechnet und somit auch die Zeitwerte  $[t'_{\text{min}}, t'_{\text{max}}]$  aktualisiert. Anschließend wird der virtuelle Zeitfortschritt  $ta(s)$  neu berechnet.

Die Definition gekoppelter PDEVS<sub>RCP</sub> ist identisch mit PDEVN in Gleichung 2.

Im Rahmen verschiedener Forschungsprojekte [8,9] zeigte sich, dass bei komplexen Problemen umfangreiche Adaptionen der Modelle beim Übergang von der Entwurfs- zur Automatisierungs- und Betriebsphase erfolgen müssen. Nachfolgend werden offene Probleme von PDEVS<sub>RCP</sub> aufgezeigt.

#### 4 Offene Probleme von PDEVS<sub>RCP</sub>

Bei der Simulation eines in der Entwurfsphase entwickelten PDEVS Modells, kann jedes atomare PDEVS den Zeitpunkt für sein nächstes internes Ereignis mit  $t_{\text{next}} = ta(s)$  bestimmen, sowie beliebig Ereignisse empfangen oder versenden. Eine Zeitfortschaltung erfolgt hierbei auf Basis *logischer Zeit* (Virtual Time, VT). Beim Übergang zur Automatisierungsphase wird in der Regel eine Interaktion mit der Prozessumgebung erforderlich. Hierfür wird in [6] der Begriff Aktivität eingeföhrt und jeder Aktivität zugleich ein zulässiges Ausführungsintervall auf Basis *realer Zeit* (Wall Clock Time, WCT) zugeordnet. Zur Ermittlung der WCT wird eine *Echtzeituhr* (Real Time Clock, RTC) benötigt. Eine formale Definitionen der RTC und von Aktivitäten, wird jedoch nicht angegeben. Beim Übergang von der Entwurfsphase zur Automatisierungsphase muss die Dynamik bereits entwickelter atomarer PDEVS Komponenten zumeist überarbeitet werden. Dies folgt direkt aus der Forderung, dass atomare PDEVS<sub>RCP</sub> Modelle nur eine Zeitfortschaltung mit

$ta(s) \in \{0, inf\}$  definieren dürfen. Daraus folgt, dass es bei der Simulation von PDEVSRCP Modellen keine virtuelle Zeitfortschaltung gibt. Somit muss die Dynamik aller atomaren PDEVSRCP Modelle überarbeitet werden, welche eine Zeitfortschaltung  $ta(s) > 0$  definieren. Dies kann zu Fehlern führen und die Steuerungsentwicklung maßgeblich verzögern. Nur atomare PDEVSRCP Komponenten mit einer Zeitfortschaltung  $ta(s) \in \{0, inf\}$  sind hiervon nicht betroffen und können unverändert weiterverwendet werden. Dies zeigt, dass die Forderung einer durchgängigen Steuerungsentwicklung teilweise verletzt wird. Somit lassen sich vier wesentliche Probleme identifizieren: (i) Es existiert keine konkrete Spezifikation der Echtzeituhr. (ii) Eine formale Spezifikation von Aktivitäten fehlt. (iii) Es fehlt jegliche Information zur formalen Spezifikation von Aktivitäten durch atomare PDEVSRCP Modellkomponenten. (iv) Ereignisse lassen sich durch Modellkomponenten nur eingeschränkt einplanen ( $ta(s) \in \{0, inf\}$ ). PDEVSRCP in der Version 2.0 behebt diese Probleme und wird nachfolgend vorgestellt.

## 5 Der PDEVSRCP 2.0 Formalismus

Aufbauend auf den offenen Problemen der ersten PDEVSRCP Version werden zunächst Aktivitäten betrachtet. Hierbei kann man, wie in Abbildung 2 gezeigt, zwischen synchronen und asynchronen Aktivitäten unterscheiden.

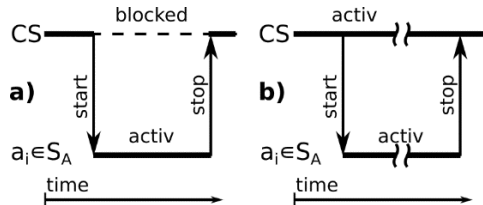


Abbildung 2: a) Syn- und b) asynchrone Aktivitäten

Bei synchronen Aktivitäten wird die Ausführung der *Steuerungssoftware* (CS) solange verzögert, bis die gestartete Aktivität  $a_i \in S_A$  beendet ist. Bei der Ausführung einer synchronen Aktivität muss zwingend sichergestellt werden, dass sie sich von selbst beendet, da ansonsten ein *DeadLock* in der CS auftritt. Bei asynchronen Aktivitäten wird die Ausführung der CS nicht durch das Starten der Aktivität  $a_i \in S_A$  blockiert. Die CS sollte die Ausführung der Aktivität überwachen und bei einer Verletzung des zulässigen Ausführungszeitintervalls die Aktivität definiert beenden. Mit der Einführung von PDEVSRCP 2.0 wird eine Aktivität  $a_i \in S_A$  als dynamisches System gemäß Gleichung 4 definiert.

$$a_i = (\mathbf{M}, \mathbf{X}, \mathbf{S}, \mathbf{Y}, f_{ext}, f_{int}, f_{out}) \quad (4)$$

$\mathbf{M}$  Set of activity methods

$\mathbf{X}$  Set of input values

$\mathbf{S}$  Set of states

$\mathbf{Y}$  Set of output values

$f_{ext}$   $f_{ext}: S \times X \rightarrow S$

$f_{int}$   $f_{int}: M \times S \rightarrow S$

$f_{out}$   $f_{out}: S \rightarrow Y$

Eine Aktivität ist charakterisiert durch eine Menge von Methoden  $M$ , Eingangswerten  $X$ , Zuständen  $S$  sowie Ausgangswerten  $Y$ . Weiterhin definiert eine Aktivität drei Dynamikfunktionen  $f_{ext}$ ,  $f_{int}$  und  $f_{out}$ . Die Funktion  $f_{ext}$  bildet die Eingangswerte  $x \in X$  der Aktivität auf ihren Zustand  $s \in S$  ab. Die Funktion  $f_{int}$  führt die Methoden  $m \in M$  der Aktivität aus und aktualisiert den Zustand  $s \in S$  der Aktivität. Beispielsweise kann eine Fehlerbehandlungsroutine  $m \in M$  gestartet werden, falls ein zuvor definiertes Ausführungszeitintervall  $t_{lim} \in S$  nicht eingehalten wird. Die dritte Funktion  $f_{out}$  spezifiziert die Abbildung des internen Zustandes  $s \in S$  auf Ausgabewerte  $y \in Y$  der Aktivität. Diese Definition von Aktivitäten ermöglicht eine einheitliche Spezifikation von synchronen und asynchronen Aktivitäten.

Aufbauend auf der neu eingeführten Spezifikation von Aktivitäten und der Definition von PDEVSRCP im Kapitel 3 erfolgt die neue Definition eines atomaren PDEVSRCP 2.0 Modells gemäß Gleichung 5.

$$\text{PDEVSRCP2.0} = (\mathbf{X}, \mathbf{Y}, \mathbf{S}_{RCP}, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda_{RCP}, ta) \quad (5)$$

$$S_{RCP} = S \cup S_A$$

$$S_A = \{a_1, \dots, a_i, \dots, a_n\}$$

$$\lambda_{RCP}: S_{RCP} \rightarrow Y \times S_A$$

$X$ ,  $Y$ ,  $\delta_{ext}$ ,  $\delta_{int}$ ,  $\delta_{con}$  und  $ta$  folgen dem klassischen PDEVSRCP Formalismus. Angemerkt sei, dass die Menge der Eingangsereignisse  $X$ , im Gegensatz zu [6] nicht vom allgemeinen PDEVSRCP Formalismus in Gleichung 1 abweicht.  $S_{RCP}$  ist eine Vereinigung der zwei Mengen  $S$  und  $S_A$  zusammen. Hierbei ist  $S$  die klassische Zustandsmenge gewöhnlicher atomarer PDEVSRCP.  $S_A$  ist eine Menge zur Abbildung von Aktivitäten nach Gleichung 4. Die Ausgabefunktion  $\lambda_{RCP}$  definiert analog zu PDEVSRCP die Berechnung von Ausgangsereignissen. Diese können an andere Modellkomponenten oder die Prozessumgebung versendet werden. Weiterhin erfolgt durch die Ausgabefunktion  $\lambda_{RCP}$  die Ausführung und Überwachung einer Aktivität  $a_i \in S_A$ .

Jedes Modell mit Prozessinteraktion benötigt eine *Echtzeituhr* (RTC). Die Spezifikation der RTC gemäß

PDEVSRCP 2.0 unterscheidet sich grundlegend von der in [6]. Bei PDEVSRCP 2.0 synchronisiert die RTC die virtuelle und die reale Zeitfortschaltung auf Modellebene. Ein Versenden von WCT Werten als externe Ereignisse  $X_{clock}$  an andere PDEVSRCP Komponenten, wie in [6], erfolgt nicht. Die Spezifikation der RTC als atomares PDEVSRCP 2.0 ist in Gleichung 6 dargestellt.

$$\begin{aligned}
& \mathbf{RTC} = (X, Y, S_{RCP}, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda_{RCP}, \mathbf{ta}) \quad (6) \\
& X = \emptyset, Y = \emptyset \Rightarrow \delta_{ext}, \delta_{con} \text{ not defined} \\
& S_{RCP} = S \cup S_A; S = \{\sigma, t_{Last}, t_{WCT}\}; S_A = \{getWCT\} \\
& s_{RCP,0} = (0,0,0, getWCT) \quad // \text{initial state} \\
& getWCT = (M, X, S, Y, f_{ext}, f_{int}, f_{out}) \\
& M = \{readSystemClock\} \\
& X = \emptyset \Rightarrow f_{ext} \text{ not defined} \\
& S = \{t_{value}\} \text{ with } s_0 = 0 \\
& Y = \{t_{wct}\} \\
& f_{int}(readSystemClock, t_{value}) \\
& \quad t_{value} := readSystemClock \\
& \quad f_{out}(t_{value}) \\
& \quad t_{WCT} := t_{value} \\
& ta(\sigma, t_{Last}, t_{WCT}, getWCT) := \sigma \\
& \lambda_{RCP}(\sigma, t_{Last}, t_{WCT}, getWCT) \\
& \quad getWCT.f_{int}(getWCT.M, getWCT.S) \\
& \delta_{int}(\sigma, t_{Last}, t_{WCT}, getWCT) \\
& \quad t_{WCT} := getWCT.f_{out}(getWCT.S) \\
& \quad \text{if } t_{WCT} - t_{Last} < 0.1 \\
& \quad \text{yes: } \sigma := 0 \quad // \text{VT advance} = 0 \\
& \quad \text{no: } \sigma := 0.1; t_{Last} := t_{WCT} \quad // \text{VT advance} = 0.1
\end{aligned}$$

Beim Start der Simulation beziehungsweise CS ist die RTC aktiv und plant mit  $ta(\sigma, \dots) = 0$  ein internes Ereignis ein. Dies führt zur Ausführung der Ausgabe-funktion  $\lambda_{RCP}$ . Hierbei wird die Funktion  $f_{int}$  der Aktivität  $getWCT$  gerufen, welche den aktuellen WCT Wert der Systemuhr auf den Zustand  $t_{value} \in S$  der Aktivität speichert. Anschließend erfolgt gemäß dem PDEVSRCP Formalismus die Ausführung der internen Zustandsüberföhrungsfunktion  $\delta_{int}$ . Hierfür wird der zuvor durch die Aktivität  $getWCT$  bestimmte WCT Wert durch die Funktion  $f_{out}$  auf  $t_{WCT} \in S$  gespeichert. Basierend auf  $t_{WCT}$  und  $t_{Last}$  wird die vergangene Zeitdauer gemäß der WCT bestimmt und mit einem zuvor definierten Schwellwert verglichen. In Gleichung 6 wurde exemplarisch der Schwellwert 0.1 gewählt. Wenn der Schwellwert unterschritten wird, folgt mit  $\sigma = 0$  sofort ein weiteres internes Ereignis und der zuvor beschriebene Ablauf wiederholt sich ( $ta(\sigma, \dots) = 0$ ). Wenn hingegen die vergangene Zeitdauer größer oder gleich dem Schwellwert ist, wird  $t_{Last} = t_{WCT}$  aktualisiert und mit  $ta(\sigma, \dots) > 0$  (in diesem Fall  $\sigma = 0.1$ ) ein neues internes Ereignisses eingeplant.

Ein entscheidender Vorteil von PDEVSRCP Version 2.0 ist, dass die Dynamik von zuvor entwickelten PDEVSRCP

Komponenten beim Übergang in die Automatisierungsphase nicht angepasst werden muss. Ein SM kann durch das Hinzufügen einer RTC Komponente gemäß Gleichung 6 als CS nach dem SiL Ansatz benutzt werden. Dabei bleibt die virtuelle Zeitfortschaltung (VT) auf Modellebene erhalten und automatisch durch die RTC mit der WCT synchronisiert. Die Abarbeitung von PDEVSRCP 2.0 Modellen erfolgt durchgehend mit einem klassischen PDEVSRCP Simulator.

Auf Modellebene kann der Proportionalitätsfaktor zwischen VT und WCT frei gewählt werden, sodass die Umsetzung von skaliertem Echtzeit ( $WCT \approx c * VT; c > 0$ ) möglich ist. Dies kann die Fehlersuche unterstützen. Weiterhin kann die Einhaltung einer oberen Zeitgrenze ( $t_{elapsed} > t_{kritisch}$ ) durch die RTC geprüft werden und somit die Einhaltung *weicher Echtzeit* sichergestellt werden.

Nachfolgend werden beide PDEVSRCP Versionen an einem Anwendungsbeispiel gegenübergestellt.

## 6 Anwendungsbeispiel

Abbildung 3a, zeigt das Modell eines einfachen Prozesses bestehend aus einem *Generator* (GEN), einem *Prozessor* (PROC) und einem *Transducer* (TRA). Der GEN erzeugt aufeinanderfolgende Aufträge (JOBS) für den PROC. Dieser verarbeitet jeweils einen JOB und erzeugt bei Beendigung ein Ausgangsereignis DONE. Der TRA stoppt (STP) die Erzeugung weiterer JOBS durch den GEN, wenn eine zuvor definierte Anzahl an JOBS durch den PROC abgearbeitet wurde. Bearbeitet der PROC einen JOB und erhält einen weiteren, so wird dieser verworfen. Im Teilbild 3b wird das Modell um ein *Interface* (INTF) erweitert. Dieses dient der späteren Interaktion mit der Prozessumgebung. Somit muss das INTF in der Automatisierungsphase als PDEVSRCP Modell spezifiziert werden. Die Umsetzung mit PDEVSRCP nach Kapitel 3 ist im Teilbild 3c zu sehen. Hierbei wurde das Modell um eine *Echtzeituhr* (RTC) ergänzt, welche die WCT als Ausgangsereignis verschickt. Auffällig ist, dass nicht nur INTF dieses Ereignis benötigt, sondern auch GEN. Folglich muss GEN um einen neuen Eingangsport ergänzt werden. In der Entwurfsphase wurde die Dynamik des GEN auf Basis *virtueller Zeit* (VT) entwickelt. PDEVSRCP nach Teilbild 4c unterstützt keine Synchronisation zwischen VT und WCT, denn nach Kapitel 3 gilt:  $ta(s) \in \{0, inf\}$ . PDEVSRCP 2.0 behebt dieses Problem mit einer Neudefinition der RTC gemäß der Spezifikation in Gleichung 6. Wie Teilbild 3d zeigt, muss das SM aus Teilbild 3b nur um eine RTC ergänzt

werden. Diese synchronisiert die VT und die WCT. Damit entfällt das explizite Versenden der WCT als externe Ereignisse. Eine Modifikation des GEN ist nicht erforderlich. PDEVs und PDEVs<sub>RCP</sub> 2.0 Komponenten können in einem Modell gemeinsam genutzt werden. Das INTF kann als PDEVs<sub>RCP</sub> 2.0 Komponente nach Kapitel 5 entwickelt werden und Aktivitäten starten, überwachen und beenden.

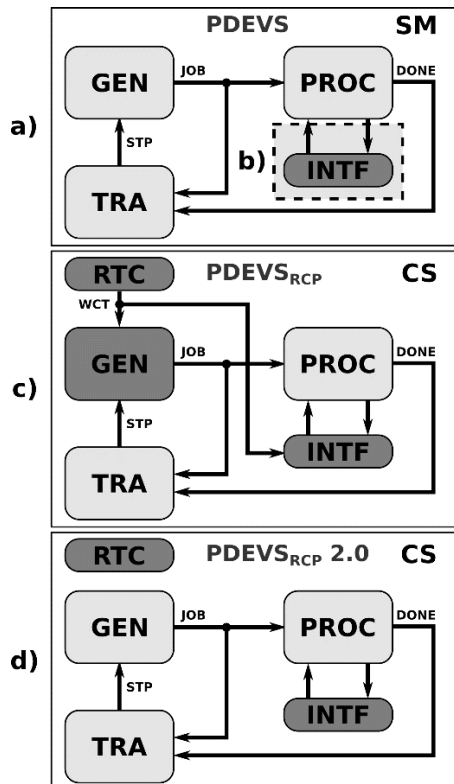


Abbildung 3: Entwicklungsphasen einer SiL basierten Steuerung: (a, b) Entwurf, (c) Automatisierung mit PDEVs<sub>RCP</sub>, (d) Automatisierung mit PDEVs<sub>RCP</sub> 2.0

## 7 Zusammenfassung

Die Umsetzung der RCP Entwurfsmethodik unter Nutzung des *Software in the Loop* Ansatzes auf Basis des PDEVs Formalismus wurde untersucht. In der Literatur sind verschiedene DEVS Formalismen zur Interaktion mit einer Prozessumgebung bekannt. Den meisten Ansätzen ist gemeinsam, dass es im Entwicklungsprozess zum Austausch des Simulators und damit einhergehenden Modellmodifikationen kommt. Dies kann zu Fehlern führen und widerspricht der durchgängigen Steuerungsentwicklung nach dem RCP Ansatz. Hieraus folgte die Motivation zur Entwicklung des PDEVs<sub>RCP</sub> Formalismus, welcher direkt auf PDEVs

basiert und eine durchgängige Simulatornutzung erlaubt. Nachteilig war bisher, dass beim Übergang von der Entwurfsphase in die Automatisierungsphase erhebliche Modellanpassungen notwendig waren. Mit der Entwicklung von PDEVs<sub>RCP</sub> 2.0 entfallen diese Modellanpassungen. Anhand eines Demonstrationsbeispiels wurden beide Versionen verglichen und die Vorteile der neuen Version aufgezeigt.

## 8 References

- [1] D. Abel, A. Bollig, Rapid Control Prototyping, Methoden und Anwendung, Springer, 2006.
- [2] G. Maletzki, Rapid Control Prototyping komplexer und flexibler Robotersteuerungen auf Basis des SBC-Ansatzes. Diss., Uni. Rostock/HS Wismar, 2014.
- [3] B.P. Zeigler, H. Praehofer, T.G. Kim, Theory of modeling and simulation 2nd ed., Acad. Pr, 2000.
- [4] S.M. Cho, T.G. Kim, Real Time Simulation Framework for RT-DEVS Models, Trans. Soc. Comput. Simul. Int. 18 (4) (2001) 203–215.
- [5] J.L. Risco-Martin, S. Mittal, J.C. Fabero, P. Malagón, J.L. Ayala, Real-time Hardware/Software Co-design Using Devs-based Transparent M&S Framework, in: Proc. of the Summer Computer Conf., 45:1-45:8.
- [6] T. Schwatinski, T. Pawletta, S. Pawletta, C. Kaiser: Simulation-based development and operation of controls on the basis of the DEVS formalism. Proc. 7th EUROSIM Conf., Vol.2, Prag, Czech Rep., 2010, 8 pages.
- [7] A.C.-H. Chow, Parallel DEVS: A Parallel, Hierarchical, Modular Modeling Formalism and Its Distributed Simulator, Trans. Soc. Comput. Simul. Int. 13 (2) (1996) 55–67.
- [8] B. Freymann, T. Pawletta, S. Pawletta, Multi-Robotersteuerungen mit variablen Interaktionsprinzipien auf Basis des Simulation Based Control Frameworks und dem Discrete Event System Specification Formalismus, Proc. ASIM STS/GMMS WS, ARGESIM Rep. No.50, 2015, 67–77.
- [9] B. Freymann, T. Pawletta, T. Schwatinski, S. Pawletta, Modellbibliothek für die Interaktion von Robotern in der MATLAB/DEVS-Umgebung auf Basis des SBC-Frameworks, Proc. ASIM STS/GMMS WS, ARGESIM Rep. No.42, 2014, 199–208.

# Conceptional problems of transaction-based modeling and its implementation in SimEvents 4.4

Lars Austermann, Peter Junglas, Jan Schmidt, Christian Tiekmann  
PHWT Vechta/Diepholz/Oldenburg  
*peter@peter-junglas.de*

Transaction-based modeling is a widely used graphical method for modeling discrete event systems, a recent implementation being SimEvents from Mathworks. Though it is applicable to a wide range of problems, it has its specific drawbacks. Some of them are connected to the basic abstractions of the method, others are related to the specific program used. By implementing a few different standard examples, we will show some of these shortcomings together with possible workarounds. This should be pointed out, when teaching this method to new users, but additionally has to be taken into account, when building a new transaction-based library or corresponding blockset.

## 1 Introduction

The modeling of discrete systems is a broad and difficult subject that has created a wealth of very different paradigms. Concentrating on graphical methods only, widely used techniques range from highly abstract ones such as Petri nets [1] and state graphs [2] to concrete material flow applications, e. g. PlantSimulation [3].

At a medium level of abstraction one finds process-based and transaction-based modeling [4], which both describe entities that are handled by fixed components. A widely used implementation of the former method is Arena [5], of the latter SimEvents [6]. The basic difference between them is the way how entities are transported: In process-based methods the components are the active parts and “seize” the entities, in transaction-based systems the entities play an active role and move automatically, until they are blocked. These two methods are used in many industrial applications, since they are sufficiently abstract to be universally applicable, but at the same time concrete enough to be comprehensible by users from very different disciplines.

But until now there is no well-established set of basic features and components for both methods. As a consequence users have to stick to a given simulation environment or even to a fixed version, if they don’t want to reimplement and partially redesign their mod-

els. As the huge conceptional changes between the latest releases of SimEvents indicate [7], especially the definition of the transaction-based modeling approach presently seems to be unclear. Therefore one should reconsider the basic design of the method and try to answer questions such as the following: What are the shortcomings of current implementations? Which concepts or components are missing? How could a reasonable set of components be defined?

The aim of this paper is to provide first steps to answer these questions. To this end it examines a few standard example problems and their implementations in SimEvents 4.4. It shows the conceptual problems that have been encountered during the implementation and provides possible workarounds. It concludes with some proposals related to applying and teaching the transaction-based approach to modeling.

## 2 Example models

To study the range of applicability of the transaction-based approach, one can draw on a huge stock of practical examples. For the purpose of finding the weaknesses of the method, four models will be presented in the following, that clearly indicate the points that are of interest here. The first three come from Law’s textbook [8], the last one from the Argesim benchmark C14 [9]:

**timeshared** Model of a time-shared computer, where several terminals send jobs of varying computing time demands, which are processed in time slices using a round-robin scheduler.

**multiteller** Model of a multiteller bank with several queues and jockeying, i.e. customers are allowed to change to a shorter queue.

**jobshop** Model of a factory with five workstations, where variable kinds of jobs are processed, which require different paths through the stations.

**supplychain** Model of a supply chain consisting of wholesalers, who order different products from distributors, which in turn order from several factories. The distributors use special strategies to comply with the demand.

The models are described in full detail in their references. The following section will concentrate on those parts of the implementations that are relevant for the discussion.

### 3 Problems implementing the examples

In the following some of the conceptual problems that have been encountered during implementation of the examples will be presented in detail. In addition it will be shown how to cope with them, mainly by introducing components that help to implement new abstractions.

#### 3.1 Handling of concurrent events

The central element of many discrete event systems is a global event queue that contains all events in the proper order. But in a transaction-based model the events are defined locally by the individual blocks, so that the proper order of events that originate from different blocks at the same time instant is not always well-defined.

An example is shown in Fig. 1: A generator component creates entities with increasing id's, which will be routed afterwards to one of two outputs depending on whether the id is even or odd. For this purpose the Get Attribute block extracts the id of an

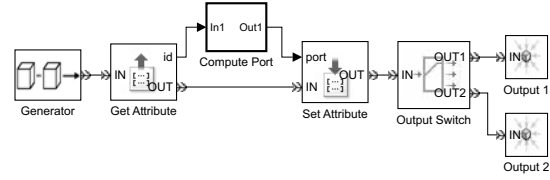


Figure 1: Example with concurrent events

entity, the Compute Port component uses this value to compute the corresponding output port number and the Set Attribute block sets the port number as an attribute of the entity.

Simulating the model leads to an error message stating that a race condition has been detected; it is not defined which of the two inputs of Set Attribute arrives first: the new entity or the new attribute value. If one ignores this message by changing it into a warning, the entities leave the Output Switch at the wrong port. But if one follows the recommendation given in the error message and inserts a *null server*, i.e. a Server component with service time  $t_s = 0$ , after the entity output of Get Attribute, the model works as planned.

This problem is inevitable when working with local components and is of course well known for a long time: Already the textual modeling language GPSS had a BUFFER command that reorders concurrent events [10].

The insertion of a null server to delay entities seems to be a simple solution, but it has a serious drawback: If its output is blocked, the server stores one entity. This has to be taken into account properly and complicates model design, as can be seen in the implementation of the CPU in the “timeshared” example (Fig. 2).

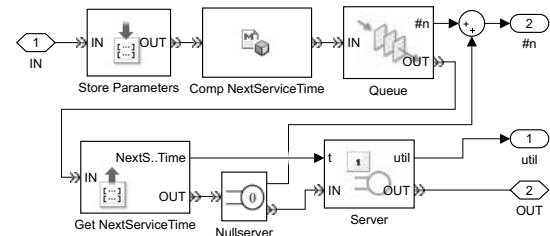


Figure 2: CPU component of the “timeshared” example

When jobs leave the queue to enter the server block that represents the CPU proper, their service time has to be fetched from an attribute, which makes the in-

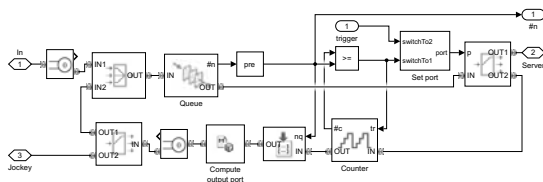
section of a null server necessary. If one is interested in statistical data about the length of the queue, one has to add the additional job that may be stored in the null server. Therefore the average queue length that the queue component provides, is of no use, it has to be computed “manually” instead.

The workaround of using a null server to implement an “infinitesimal” delay is conceptually wrong, because the implicit storage that it adds, is not related to the problem it solves. This can even lead to more serious problems as will be seen in section 3.4.

### 3.2 Separation of entities from a queue

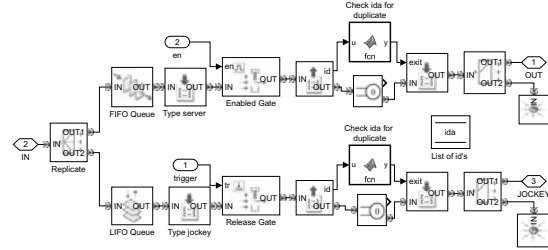
The “multiteller” example allows for the well known phenomenon of jockeying in a queue, i.e. customers at the end of a queue can switch to another shorter one. Modeling this behaviour with SimEvents turned out to be much more difficult than expected. The reason is that although an entity can leave a queue, when its waiting time exceeds a threshold (a behaviour known as *reneging*), there is no way to detach the last entity of a FIFO queue on arrival of an input signal (e. g. when another queue has become shorter).

To model such a *jockey queue* two very different schemes have been devised: In the *shuffle queue* (Fig. 3) the signal opens up a path from the exit to the beginning of the queue. All entities walk around the circle and get back into their old position, except for the last one, which leaves the block through the extra jockeying output. From here it is routed to the shorter queue.



**Figure 3:** Shufflequeue component of the “multiteller” example

The *clone queue* (Fig. 4) creates duplicates of all incoming entities and routes them into a FIFO and a LIFO queue. When the queue exit is opened, an entity is taken from the FIFO queue, but when the jockey signal arrives, the LIFO queue is used. A book-keeping device destroys clones, whose partner has already left the queue, when they appear at the end.



**Figure 4:** Clonequeue component of the “multiteller” example

Both schemes are quite complicated and lead to a lot of additional events. Though they worked at last, their implementations had to cope with a lot of difficult timing problems and appear to be cumbersome and error prone. What one needs instead, is a simple mechanism to remove a given entity from a queue, maybe similar to the concept of *user chains* in GPSS [10].

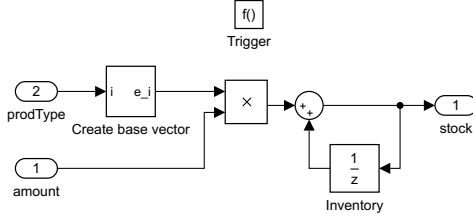
### 3.3 Storing entities

A basic ingredient of the “supplychain” example is a storage component that stores incoming entities denoting products of several types. On arrival of an order it emits the corresponding products at its output port. Scanning through the SimEvents library to find blocks that can store elements, one comes up with the queue, the server and the resource blocks, but none of them seems appropriate for the task at hand: A server is not well suited to deliver a certain type of product on demand, the resource pool provides a fixed amount of resources. And the queue block doesn’t scale well with the number of different product types, since one needs one queue per type to emit a product entity of given type on request.

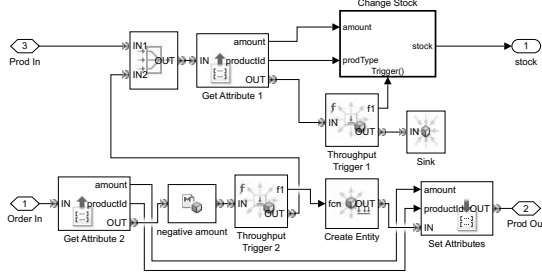
Instead of trying to use some of these blocks together with complicated gates and logic to coerce them into a non-fitting scheme, one can use a simple 1/z block from Simulink’s basic discrete library. It contains the inventory (Fig. 5), which is just a vector with the amounts of the different product types in the stock.

The Storage component (Fig. 6) registers incoming products in the inventory and destroys the corresponding entities. When an order arrives the inventory is reduced and the proper entities are recreated at the output port.

Of course this is only a trick, because the entities



**Figure 5:** Change Stock component of the “supplychain” example



**Figure 6:** Storage component of the “supplychain” example

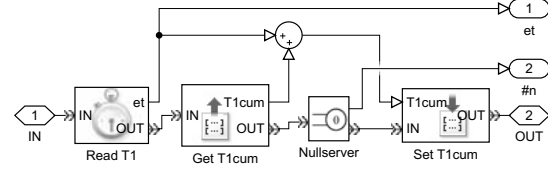
themselves are not stored at all. It worked for the example program, since the structure of the product entities is simple and always the same. What one really needs, is a more general component that actually stores the incoming entities and can release them on demand. The actual design of such a block is open to discussion and could be guided on example models and implementations in other environments. A simple first idea would be to use a special queue, where selected entities can move to the front of the line. Again, the old idea of “user chains” could come in handy here.

### 3.4 Time measurements across several blocks

In the “jobshop” example entities use different paths through workstations with associated queues. One is interested in statistics for the total waiting time of the entities over all those queues. To measure this value, one needs timers that can be paused after a queue and resumed before the next one to add up the single waiting times of passing entities.

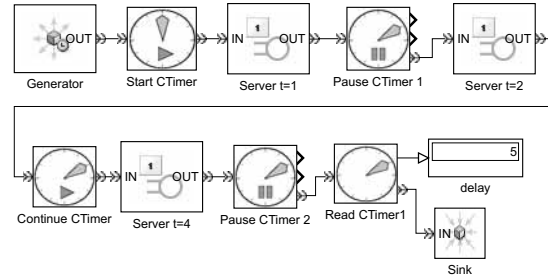
Unfortunately SimEvents only provides simple timers that measure between two fixed points. Therefore one has to add up the waiting times of the individual

queues inside the entity using an attribute (Fig. 7).



**Figure 7:** Accumulation of times in a PauseCTimer component

Based on this idea one can easily build components to start, pause, continue and read such accumulating timers and test them in simple models (Fig. 8).



**Figure 8:** Simple test model for accumulating timers

But the problem is, they don’t work if the accumulating block `PauseCTimer` sits between a queue and a server – which is exactly the most interesting configuration in general and especially in the “jobshop” example. The reason is of course the null server block that is necessary inside `PauseCTimer`: When the server is busy, the first entity in the preceding queue moves into the null server and waits there instead in the queue. The additional waiting time is not accounted for in the timer. This is a serious problem: One could measure the time that an entity stays in the null server, but to accumulate it, one needs another null server!

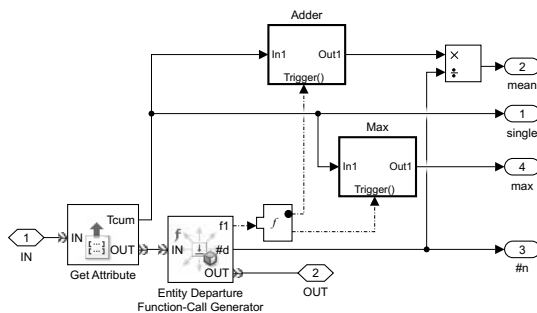
As always there is a workaround: Accumulation between a server and an (unlimited) queue is no problem, as the queue never blocks and the preceding null server doesn’t store an entity. Therefore one can measure the time  $t_{Q+S}$  between entering the queue and leaving the server and the time  $t_S$  inside the server and accumulate the time  $t_Q = t_{Q+S} - t_S$  after the server. But this is awkward and only shows again that the basic design of a null server is seriously flawed.

### 3.5 Statistical analysis

An important aspect of discrete simulation is the gathering of statistical data, often in the form of a final report. In the “jobshop” example for instance, one is interested in the queue delay, queue length and server utilisation for the different workstations, as well as the total waiting time per job and per jobtype.

In a transaction-based environment there is no instance to collect such data but the blocks themselves. SimEvents provides mean values, utilisations and similar data for individual components, but no additional statistical tools. This is unfortunate, because due to the abundance of null servers one often has to combine individual values and can not use the statistical block data itself. The actual length of a queue for example has to be enlarged by the occasional inhabitant of a subsequent null server, as well as the corresponding waiting time. To compute mean values per entity or per time one has therefore to build own blocks, which admittedly can be done easily with standard Simulink methods.

Another challenge is the collection of statistics connected to entities, not blocks, such as the accumulated waiting times through several queues. The basic idea here is to store data in the entities themselves and collect them at the end. Fig. 9 shows how mean or maximal values can be computed with the help of simple 1/z blocks that are hiding in the Adder and Max blocks.



**Figure 9:** Component **EntityMeanMax** for accumulating entity data

Combining all these auxiliary blocks in a final statistics subsystem not only helps to unclutter the model, but brings together all statistical data. They now can easily be pipelined into a Matlab script that creates a proper report file, if such is desired.

## 4 Conclusions

The preceding analysis has unfold four different areas, where transaction-based modeling and its implementation in SimEvents show conceptual difficulties:

- timing of concurrent events;
- implementation of alternative queueing policies;
- storing and retrieving of entities;
- gathering and processing of statistical data.

Of these the first one is by far the most serious, and a generally working solution instead of the defective null server workaround is not available in SimEvents. This is especially annoying since already GPSS had a better solution with its **BUFFER** command.

All other problems could be solved by introducing appropriate subsystems using blocks available in SimEvents or the underlying Simulink. Adding such components to a user library, one can simplify the modeling of a wide range of applications. Though this may be sufficient for the practitioner in the industry, it is a real drawback of the transaction-based method: A corresponding library should provide the basic abstractions that are necessary to model all problems that the method addresses.

Considering the queueing and storing problems the GPSS construction of “user chains” seems to be a promising idea: Instead of adding ever more specialized components, it provides an underlying mechanism that may be able to cope with some of the difficulties presented above. To gain further insight into possible solutions, we propose to add a new benchmark to the Argesim suite [11] that requests the modeling of several jockeying queues and the collection of statistical data including the delay over several queues, similar to the multiteller example. To complicate matters one could ask for implementations with a large number of queues, which would address the problem of “vectorising” components.

Mathworks has realized that SimEvents 4.4 has still basic problems and came up with a complete redesign of its SimEvents library with version 5. A very interesting feature is that the design is based on a unifying theoretical description [7]. Unfortunately, Mathwork has chosen a new design instead of relying on the well-known DEVs formalism [4]. Many basic aspects have

been changed with the new release, often by substituting graphical elements with Matlab code. As a consequence there is no simple migration path from the old to the new version. Whether this has led to a satisfactory implementation, needs further investigation.

In any case this only adds to the central point made here: For the advancement of transaction-based modeling it is vital that it is based on a thorough theoretical analysis to reveal the fundamental abstractions and basic components that are necessary. The goal is to find stable designs, that don't change with every new tool or release, to get models that are better understandable, because they don't rely on tricky workarounds, and to reach high quality solutions, since they have a sound foundation. If we don't care for the basic concepts, we have to live with redesigning our models and rewriting our lectures every other year.

## Acknowledgements

The second author (P. J.) likes to thank Thorsten Pawletta for introducing him to the rich history of discrete modeling tools.

## References

- [1] C. G. Cassandras, S. Lafortune. *Introduction to Discrete Event Systems*. Springer, New York, 2. ed. 2008.
- [2] D. Harel. *Statecharts: A Visual Formalism for Complex Systems*. Science of Computer Programming, 8, pp. 231-274, 1987.
- [3] S. Bangsow. *Tecnomatix Plant Simulation: Modeling and Programming by Means of Examples*. Springer, Cham, 2015.
- [4] B. P. Zeigler, H. Praehofer, T. G. Kim. *Theory of Modeling and Simulation*. Academic Press, San Diego, 2nd ed. 2000.
- [5] W. D. Kelton, R. P. Sadowski, N. B. Zupick. *Simulation with Arena*. McGraw-Hill, New York, 6. ed. 2015.
- [6] The MathWorks. *SimEvents: Model and simulate discrete-event systems*. Online: [www.mathworks.com/products/simevents.html](http://www.mathworks.com/products/simevents.html) (called 2017-01-30).
- [7] W. Li, R. Mani, P. J. Mosterman. *Extensible discrete-event simulation framework in SimEvents*. Proc. 2016 Winter Simulation Conference, Arlington, pp. 943-954, 2016.
- [8] A. M. Law. *Simulation Modeling and Analysis*. McGraw-Hill, New York, 5. ed. 2014.
- [9] S. M. Tauböck. *C14 Supply Chain Management Definition*. Simulation News Europe, 32/33, pp. 42-43, 2001.
- [10] T. J. Schriber, *On the application of user chains in GPSS*. Proc. 1973 Winter Simulation Conference, San Francisco, pp. 140-158, 1973.
- [11] F. Breiteneker, S. Wassertheurer, N. Popper, G. Zauner. *Benchmarking of Simulation Systems — the Argesim Comparisons*. Proc. First Asia Int. Conference on Modelling & Simulation, Washington, pp. 568-573, 2007.

# Comparison of Approaches for Modelling and Simulation of Structural-dynamic Systems - ARGESIM Benchmark C21 'State Events and Structural-dynamic Systems'

Felix Breitenecker, Andreas Körner

TU Wien, Institut für Analysis und Scientific Computing,

Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria

*Felix.Breitenecker@tuwien.ac.at*

Modelling and simulation of structural-dynamic systems is getting more and more important in advanced modelling theory and application. Up to now the description of the structural dynamic changes and of the state events scheduling these changes are defined in a grey zone between mathematical model description and model implementation. For comparison of proper approaches to modelling and simulation of structural-dynamic systems, the journal SNE – Simulation Notes Europe (EUROSIM's and ASIM's membership journal) has set up in its benchmark series a new benchmark dealing with this topic: SNE ARGESIM Benchmark C21 'State Events and Structural Dynamic Systems'. Simulationists are invited to propose and publish template solutions for the three case studies of this benchmark: 'Bouncing Ball', 'RLC Circuit with Diode, and 'Rotating Pendulum with Free Flight Phase' This contribution commemorates the benchmark definition in SNE 26-2, concentrating on tasks and experiments.

## 1 Introduction

Since 2000, the old CSSL standard for simulation systems has become obsolete, and new standards and techniques for system simulation are arising. At mathematical modelling level, *Physical Modelling* or *Component-based Modelling* has introduced a new era for multidomain modelling and system simulation. The 'components' may be part of textual or graphical libraries in various domains. With Modelica and with competitive VHDL-AMS, modelling languages with a certain standard have emerged.

At structural modelling level, new approaches have been driven by computer science. While object oriented programming became the quasi standard of software development, the object-oriented modelling paradigm, has emerged as very useful for modelling of simulation models and object-oriented techniques have been introduced in simulation modelling.

Especially state chart diagrams as part of the *Unified Modelling Language* (UML), a set of graphical modelling techniques have shown to be really useful for event modelling. With hybrid state charts — the era of *Dynamic Statechart Modelling* started which allows for relatively transparent modelling of structural-dynamic systems.

But still a gap between mathematical state space descriptions and unfortunately various description methods for state events on the one side and structural description of change of model and of model sequence can be observed.

Within Modelica association, a working group is still discussing proper description for hybrid systems, state events and structural dynamic systems, see e.g. [1]. Many authors are expanding hybrid state charts, but only few in context with mathematical state space descriptions, e.g. [2] with event classification and mathematical mapping for change of states.

## 2 Benchmarks in SNE

Simulation Notes Europe (SNE), EUROSIM's membership journal, features a series on comparisons of simulation software. Based on not too complicated, easily comprehensible general model descriptions, modelling approaches and implementation methods are compared by means of selected simulations tasks.

Usually, the benchmarks emphasize on specific application areas. Up to now 20 benchmarks have been defined, and about 250 'solutions' have been submitted by simulationists, following a given template [3]:

- C 1 - Lithium-Cluster Dynamics
- C 2 - Flexible Assembly System
- C 3 - Analysis of a Generalized Class-E Amplifier

- C 4 - Dining Philosophers I
- C 5 - Two State Model
- C 6 - Emergency Department - Follow-up Treatment
- C 7 - Constrained Pendulum
- C 8 - Canal-and-Lock System
- C 9 - Fuzzy Control of a Two Tank System
- C 10 - Dining Philosophers II
- C 11 - SCARA Robot
- C 12 - Collision Processes in Rows of Spheres
- C 13 - Crane Crab with Embedded Control
- C 14 - Supply Chain
- C 15 - Clearance Identification
- C 16 - Restaurant Business Dynamics
- C 17 - Spatial Dynamics of SIR-Type Epidemic
- C 18 - NN vs Transfer Functions Modelling
- C 19 - Pollution in Groundwater Flow
- C 20 - Complex Production System
- C 21 - *State Events and Structural Dynamic Systems*

C5 - *Two State Model* with state-dependent switching, C7 - *Constrained Pendulum* with impact, C11 - *SCARA Robot* with obstacle prevention, and C12 - *Collision Processes in Rows of Spheres* already sketched modelling and handling of state events, but mainly from the viewpoint of state events within a model.

More recently, the benchmarks developed towards a comparison of different modelling approaches – e.g. [4]. The new ARGESIM Benchmark C21 *State Events and Structural Dynamic Systems* [5] deals with three case studies, which incorporate structural dynamic changing systems and phenomena associated with state events, like Zeno effect, change of degree of freedom, and coordinate transformation.

### 3 Benchmark C21: State Events and Structural-dynamic Systems

After two years preparation, ARGESIM is releasing the new benchmark C21 *State Events and Structural Dynamic Systems*. The benchmark is intended to review to set up a basis for a basis for proper standardisation and description of structural dynamic systems and associated state events in system simulation. Within three case studies, different approaches to modelling and implementation of structural changes in the dynamics are to be discussed. Simulationists are invited to submit a contribution – a ‘benchmark solution’ describing their model description approach and their implementation.

### 4 C21 - Case Study ‘Bouncing Ball’

Playing table tennis is that kind of ‘experiment’ everybody has already done, watching the ball jump on the table. First, the ball is falling and jumping quite high, but bit by bit, the observer recognizes can observe decreasing amplitude and increasing frequency until it stops jumping and changes to the coasting phase.

This physical process is well known as *Bouncing Ball Dynamics*, which is met also in other application.

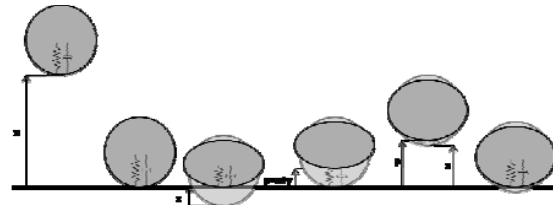


Figure 1. Bouncing ball dynamics with compliant contact region

#### 4.1 Modelling Approaches for ‘Bouncing Ball’

The benchmark requires two general modelling approaches:

- Modelling with contact phase as event, with and without air resistance
- Modelling of contact phase as dynamic movement, with and without air resistance

The first approach is not realistic, but it involves the Zeno effect: an infinite number of bounces in finite time - requiring special consideration in implementation.

The second more realistic approach results in a structural dynamic system, where the degree of freedom changes (Figure 1), and as refinement a first phase with zero deformation.

#### 4.2 Tasks in Modelling / Handling State Events with Event Contact Model

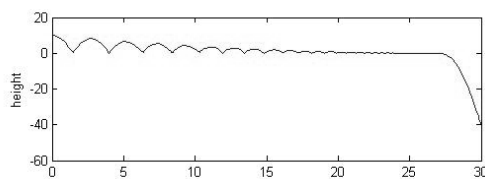
Generally, the tasks are model description, especially of event functions and event action, time domain analysis with model comparison and parameter studies, and especially of event handling, and comparison.

**Description of model implementation.** Document implementation of continuous model parts and of the event *Bounce* (textual model code snippets, (parts of) graphical model diagrams, etc.)

**'Last' Bounce Analysis.** As the model is linear, event times can be calculated directly, and a recursion allows to determine the event times for the infinite number of bounces and consequently also the the 'last' bounce (infinite number of bounces in finite time !). Simulate as near as possible to this final bounce and compare event time with limit formula.

**Scattering Prevention by Maximal Height Check.** Straightforward implementation of events often have problems with event scattering – which happens definitely near to the 'last' bounce (see Figure 2 – the ball 'falls' into the ground).

Workaround is to stop bouncing before 'last' event, e.g. if maximal height of the flight period becomes too small.



**Figure 2.** Scattering of *Bounce* events near the 'last' bounce due to missing error prevention

In each flight period, an additional event could determine the maximal height, and below a critical maximal height further *Bounce* events are stopped by parameter change or by model change, or by change of solver parameters or accuracy of event detection, etc.

**Testing accuracy of event handling.** Simulate the linear model without air resistance by an DE solver with event handling, and compare numerical bounce times with analytical bounce times given by recursion. Document results by comparison of entry times of the first 100 bounces by plotting bounce time differences over number of bounces.

**Compensation of linear model deviation.** In any case, air resistance is evident – but very small. Due to missing air resistance in the linear model, the event times are 'too late'. Simulate nonlinear and linear model and try to compensate the 'too late' bounce times in the linear model by giving an initial velocity

#### 4.3 Tasks in Modelling/Handling State Events with Continuous Contact

These tasks investigate modelling approaches for events, tests cooperation of event location with different ODE solvers, and performs parameter studies.

**Description of model implementation.** Document implementation of continuous model parts and of the events *Contact* and *Fly Restart* (textual model code snippets, (parts of) graphical model diagrams, etc.). Discuss the general modelling approach – maximal state space, hybrid decomposition, or switching model parts.

**Dependency of results from algorithms.** It might be necessary to choose specific ODE solvers or to tune ODE solver parameters and event detection parameters.

Simulate the *dynamic contact model* using different ODE solvers, and / or tune parameters for ODE solvers and event detection. Document of simulation results with plots or with tables indicating deviations for different algorithms, and discuss appropriateness of specific algorithmic properties (e.g. stepsize control vs event detection, or scattering prevention).

**Investigation of contact phase.** The proper implementation of the linear and nonlinear model ( i.e. without and with air resistance) and of the events *Contact* *Fly Restart* can be seen in detail in the contact phase which is much shorter than the flying phase. Simulation results for first contact phase, the second flight phase, and the second contact phase should be given in separate time plots over proper small observation windows.

**Parameter studies.** Variation of the spring constant  $k$  has big influence on the systems behaviour. Calculate simulation studies varying the stiffness-parameter while concurrently setting the damper constant. Afterwards, vary damping and finally document the relation of the parameters by appropriate time plots and / or parameter plots (use standard parameters before).

**Bouncing ball on Mars.** It might be nice, to now about bouncing ball behaviour on Mars and to compare with behaviour on Earth. Let's simulate for 30 seconds, whereby for Mars we must use the different parameter values for gravity constant and air resistance. Document results as comparative plots for position and velocity.

## 5 C21 - Case Study ,RLC Circuit with Diode'

The second cased study is based on a class E-amplifier (Figure 3), used already used in ARGESIM Benchmark C3. It aims for investigation of modelling and efficient calculation of switching elements (i.e. time events and state events) and for physical modelling of circuits.

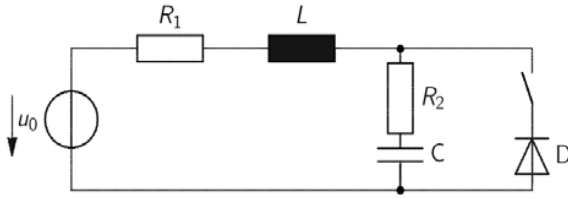


Figure 3. Serial RLC circuit with diode in parallel

### 5.1 Modelling approaches for RLC circuit and diode

Special emphasis is given to diode modelling, from simple switching to nonlinear switching behaviour. Modelling of the diode behaviour by Shockley equation causes a structural dynamic change from a 'simple' linear ODE system to a nonlinear DAE system. Challenge is a proper modelling and implementation of diode modes – from tricky switching to 'sound' model change.

Kirchhoff's laws and node equation first allow setup the physical model equations for the voltages, and inductor and capacitor have classical linear constitutive equations. The diode is described by a 'switching' functional relation between current and voltage, in general. The diode has a locking phase and a conducting phase depending on zero crossings of diode voltage. The benchmark suggests to compare the following diode models (Figure 4)

**Shortcut diode model.** The *shortcut diode model*, a simple diode model, mimics the dynamic behaviour as an ideal switch for the current depending on diode voltage, so that the diode functional description is simple (Figure 4), and also the overall model description becomes a simple linear ODE system, in *conducting phase* and in *locking phase*. The zero crossing of the diode voltage simply switches between different parameters in a linear system.

**Shockley diode model.** A diode is a nonlinear element, and indeed the switching dynamics evolve nonlinear dynamics.

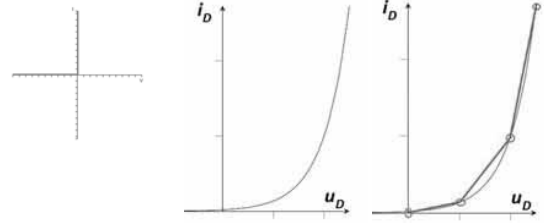


Figure 4. Diode models – functional descriptions

One nonlinear model is known as *Shockley diode model*. The mathematical description is given by an exponential-like functional relation between diode current and diode voltage.

In conducting phase, to the linear RLC ODE's now a nonlinear algebraic equation for diode voltage (or diode current) is added – resulting in a DAE system of index 1.

The zero crossing of the diode voltage now controls a structural-dynamic system – adding or neglecting a nonlinear algebraic equation.

**Interpolated Shockley diode model.** Characteristic curves are a classic way-around for algebraic equations. The 'Shockley' curve for the diode's operation can be made a linear interpolated table function, which allows for explicit resolving with respect to the diode voltage.

As result, the state equations in *conducting phase* become a piecewise linear explicit system, and no algebraic equation is necessary. The model description in *locking phase* is again the classic RLC model.

The zero crossing of the diode voltage switches between a fully linear system and a piecewise linear system.

**Explicit Shockley diode model.** In Shockley diode model, in *conducting phase* a DAE system has to be solved. DAE solvers require iteration and state event detection requires backstepping in time, which is not suitable in case of real time simulation.

The resulting DAE system is an index-1 system. Index reduction can transform the algebraic equation to an explicit ODEs. A straightforward method is to differentiate the algebraic equation directly with respect to time, resulting in a relatively complicated ODE for diode voltage (or diode current).

The zero crossing of the diode voltage now controls a structural-dynamic system – adding or neglecting an additional ODE.

## 5.2 Tasks for modelling / simulation the RLC and diode models

**Description of model implementations.** Document implementation of the RLC model and especially of the various diode models (shortcut diode model, Shockley diode model, interpolated Shockley diode model, and explicit Shockley diode model with all phase changes and events (textual model code snippets, (parts of) graphical model diagrams, etc.). Discuss the general modelling approach – maximal state space, hybrid decomposition, or switching model parts, and possible model modifications for efficient modelling e.g. for comparing different diode models.

**Dependency of results from algorithms.** It might be necessary to choose specific ODE solvers or to tune ODE/DAE solver parameters and event detection parameters.

Simulate the *diode shortcut model* and the *Shockley diode model* using different ODE/DAE solvers, and / or tune parameters for ODE solvers and event detection, with standard parameters. Give plot results, indicate sensible solver parameters.

**Comparison of shortcut and Shockley diode model.** Compare results for *diode shortcut model* and for *Shockley diode model*, with standard parameter, but timespan only two switching periods. Document results with plots, and give a relative comparison of computation times.

**Approximation of Shockley diode model.** Investigate the approximation of the *Shockley diode model* by the *interpolated Shockley diode model* with different numbers of breakpoints for the interpolation

**Relevance of choice of algebraic state.** In case of *Shockley diode model*, either an equation for diode voltage or for diode current can be used as algebraic state equation.

Simulate *Shockley diode model* with both variants, and check eventual differences – documented by plots or numeric deviations (standard parameters).

**Investigation for real-time simulation.** For real-time simulation, fixed step sizes and simplified event detection (without backstepping) must be used. With these premises, compare results for *diode shortcut model*, for *interpolated Shockley diode model*, and for *explicit Shockley diode model* by appropriate simulations. Document the implementation of the ODE for the diode voltage.

## 6 C21 – Case Study ‘Rotating Pendulum with Free Flight Phase’

The classical idealized pendulum on a rope can evolve two movements. If the rope is tight, the body is moving along a circular path. If the rope is loose, the body is free falling - until the rope is tight again. State events (length of rope and centrifugal force; fig. 3) cause structural changes of the model – with increase or loss of degree of freedom.

Challenge is a proper modelling and implementation of the two different phases of dynamics. Use of different coordinates cause more complex model description, while shared coordinates result in an DAE model of index 3 !

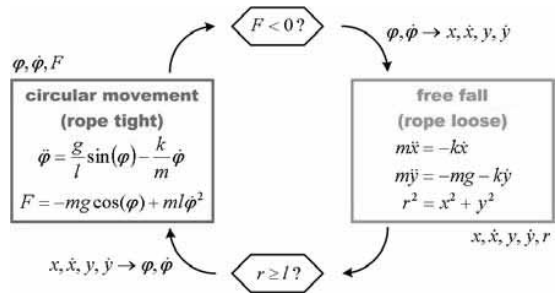


Figure 5. Hybrid decomposed model for rotating pendulum with free flight phase

### 6.1 Modelling approaches pendulum movement

Equations for movement in phase *swinging* can be derived by using the angular momentum balance, resulting in the classic pendulum equation. The pendulum is swinging, as long as the force on the rope is bigger than zero. If this force becomes lower than zero, the gravitational force outweighs the centrifugal force: the pendulum is changing into phase *falling*.

In phase falling, the movement of the body has two degrees of freedom. The motion of the mass is derived by conservation of momentum in *x*- and *y*-direction. The distance from rotation center indicates whether the rope is loose or whether it gets tight again, forcing the body back on the circular path: the mass switches to phase *swinging*.

This structural-dynamic model can be implemented by hybrid decomposition, or by a maximal state space. A hybrid decomposition for this case study is sketched in Figure A. A maximal state space approach would require a state space of dimension 6.

From physical modelling an alternative approach is suggested. The movement in the phase *swinging* can also be described in Cartesian coordinates: the mass is moving in x- and y- direction, but a force 'bounds' the movement on a circle.

The resulting DAE system indeed describes the swinging of the mass, and it could be used instead of the ODE model in polar coordinates. The events could now switch easier between the phases, because the differential state space is almost the same – so one state space with internal switching could be used. Unfortunately this DAE system is difficult to solve, because it has a differential index of 3 – which makes index reduction necessary.

## 6.2 Tasks for modelling / simulating the rotating pendulum with free flight phase

Tasks in this case study concentrate on the modelling approach and model implementation, and on few simulation.

**Description of model implementations.** Document implementation of the structural-dynamic system with phase *swinging* and phase *falling* and the state events which switch the phases. Alternatively describe the implementation of the 'common' DAE system (textual model code snippets, (parts of) graphical model diagrams, etc.). Discuss the general modelling approach – maximal state space, hybrid decomposition, DAE system with switching model parts, and possible modifications for efficient modelling.

**Basic simulation of phases.** Calculate and visualize a basic simulation run. Simulate until the maximal oscillation does not exceed a threshold – could be determined by adding an output event!

**Dependency of results from algorithms.** It might be necessary to choose specific ODE solvers or to tune ODE/DAE solver parameters and event detection parameters.

Perform simulations with standard parameters, timespan until begin of second phase *swinging* using different ODE/DAE solvers, and / or tune parameters for ODE solvers and event detection. Give plot results and indicate sensible solver parameters.

**External energy supply.** Due to physical constraints, only one phase *falling* can occur because of energy loss.

In order to 'restart' the alternating movements, energy is supplied – as increase of the angular velocity (a 'kick'). Following the basic simulation of the second task, the angular velocity is increased by a factor, so that the pendulum reaches again the phase falling.

At the first zero crossing after angle did not exceed the threshold, the angular velocity multiplied by a factor, resulting in three variants.

## 7 Summary

The Benchmark C21 - 'State Events and Structural-dynamic Systems' is a challenging one. We hope, that 'solution' sent in enrich the variety of modelling approaches and clarify some inconsistencies in state event modelling. We invite simulationist to provide a 'solution' – with publication of a *Technical Benchmark Note* (up to 10 pages) in SNE.

Authors are also invited to add better modelling approaches, or to suggest interesting experiments.

## 8 References

- [1] M. Otter, M. Malmheden, H. Elmqvist, S. E. Mattson, and C. Johnsson, "A New Formalism for Modeling of Reactive and Hybrid Systems". In: Proc. 7th Int. Modelica Conference, Linköping Univ. Electronic Press, ISBN 978-91-7393-513-5, 2009, pp 364-377.
- [2] A. Körner, "Classification for State Event Modelling of Hybrid Systems", submitted Proc. EUROSIM Congress 2016, 2 pages.
- [3] F. Breiteneker, "ARGESIM Benchmarks on Modelling and Simulation: Revised Definitions, Extended Solutions, and Supplemental Information", Simulation Notes Europe SNE 16/3-4, Dec. 2006, pp. 57-57.
- [4] G. Zauner and F. Breiteneker, "Three Structural Different Modelling Approaches to ARGESIM Comparison C7 'Constrained Pendulum' using the Modelica-Simulator MOSILAB", Simulation News Europe SNE 16/3-4, Dec. 2006, 67-68
- [5] F. Breiteneker and A. Körner, "ARGESIM Benchmark C21 – State Events and Structural Dynamic Systems", Simulation News Europe SNE 26/2, June 2016, 117 – 122. Doi: 10.11128/sne.26.bn21.10339 in print.

# Die Wissensgebiete der Simulationstechnik

Umut Durak, Torsten Gerlach  
DLR Institut für Flugsystemtechnik  
*umut.durak@dlr.de*

In den letzten Jahrzehnten hat sich die Simulationstechnik als ein eigenständiges Fachgebiet mit traditionellen, aber auch neuen und unkonventionellen Bereichen weiterentwickelt. Im Gegensatz zu den vergangenen Jahren ist der Bedarf an Ausbildung im Bereich Simulationstechnik erheblich gestiegen. Es wird deshalb zunehmend notwendig, ein gemeinsames Verständnis darüber zu erlangen, was die Wissensgebiete der Simulationstechnik sind. Dieser Beitrag präsentiert einen Überblick über bisherige Studien bezüglich der Wissensgebiete Modellierung und Simulation und gibt einen Ausblick auf das Wissensgebiet der Simulationstechnik. Der Ausblick beinhaltet die Vorstellung seiner Untergebiete und der behandelten Themen. Eine Auflistung der wichtigsten Referenzen dieser Themen vervollständigt diese Veröffentlichung.

## 1 Einleitung

Die Simulation als Begriff existiert seit dem 14. Jahrhundert und bedeutete ursprünglich Nachahmung oder Fälschung. Heutzutage definieren wir Simulation im technisch-naturwissenschaftlichen Bereich als die Anwendung dynamischer Modelle, um Experimente durchzuführen, Erfahrungen zu sammeln oder um zu unterhalten. Ein Modell stellt wiederum eine abstrakte Darstellung der Realität dar. Wie Tolk et al. in [1] schreiben, besteht das Fachgebiet „Modellierung und Simulation“ (M&S) aus den drei Komponenten Simulationswissenschaft (M&S Science), Simulationstechnik (M&S Engineering) und Simulationsanwendungen (M&S Applications). Die Simulationswissenschaft beschäftigt sich mit den theoretischen, allgemeinen Grundlagen der Simulation. Die Simulationstechnik stellt allgemeine, theoretisch fundierte Methoden und anwendungsunabhängige Lösungen und Heuristiken bereit. Die Simulationsanwendungen beinhalten spezifische Lösungen, oft basierend auf den allgemeinen Methoden der Simulationstechnik, für konkrete, praktische Probleme. Dieser Beitrag beleuchtet die Komponente Simulationstechnik und versucht eine erweiterte Definition zu schaffen.

Die Simulationstechnik kann als Disziplin betrachtet werden, die durch die Evolution von Modellierung und Simulation entstanden ist. Sie ist somit ein Teil des Wissensgebietes der M&S. Ein Wissensgebiet kann als strukturiertes Wissen verstanden werden, welches von einem Individuum gemeistert werden muss, um dort als Spezialist verstanden zu werden. Betrachtet man andere Ingenieursdisziplinen, so sind bereits viele Wissensgebiete definiert worden, z. B.

- Systems engineering (SEBoK) [2]
- Software engineering (SWEBOK) [3]
- Project management (PMBOK) [4]

In den späten 1990er und frühen 2000er Jahren wurden einige Workshops über die Wissensgebiete der M&S durchgeführt [5-8]. Es wurde erörtert, ob M&S ein eigenständiges Wissensgebiet ist und was wichtige Teilgebiete sind. Später veröffentlichte Ören, als primärer Unterstützer der Idee, seine Vorstellung zur Entwicklung einer Klassifizierung für die Wissensgebiete der M&S [9]. Er definiert elf Kategorien, die eine genaue Unterscheidung und Spezifikation erlauben. Dies sind u. a. Technik, Ethik und Zuverlässigkeit [10]. Eine weitere Verankerung in der M&S-Gemeinschaft erreichte Ören durch verschiedene Veröffentlichungen, die eine detaillierte Ausarbeitung der elf Kategorien enthalten [11-14].

Wir greifen die Grundlagen von Ören auf und betrachten diese aus dem Blickwinkel der Simulationstechnik.

## 2 Grundlagen der Simulationstechnik

Die Evolution der Simulation kann durch eine zeitliche Abfolge dargestellt werden. Am Anfang des 20. Jahrhunderts war die Simulation nicht rechnergestützt, sondern sie wurde meistens als ein Gedankenexperiment [15] oder mit miniaturisierten Modellen durchgeführt. Mit der Einführung der Rechentechnik entstand die Computersimulation. In den sechziger Jahren begann die Ära der formalen Simulation, die die Verwendung von formalen Modellen vorantrieb. Die intelligente Simulation kann als nächster Schritt der Evolution aufgefasst werden, die durch die Sy-

nergien zwischen Simulation und künstlicher Intelligenz möglich gemacht wurde [16]. Sie beinhaltet u. a. agentenbasierte, neuronale Netzwerk-, Fuzzy- und Schwarmsimulation. Letzten Endes kann Simulationstechnik als das Ergebnis der Synergien zwischen Simulation und Systems Engineering begriffen werden. Sie kann als interdisziplinärer Ansatz zur Entwicklung, Wartung und Nutzung von Simulationen betrachtet werden [17]. Sie wird durch ihren Prozess und die damit verbundenen Aktivitäten definiert. Diese Aktivitäten sind konzeptionelle Analyse, Design, Entwicklung, Integration, Verifikation und Validierung, Bereitstellung und Betrieb, Wartung, Qualität und Verwaltung. Diese Schlüsselwörter werden in dem folgenden Abschnitt als Teil der Wissensgebiete der Simulationstechnik betrachtet und ihre Bedeutung erläutert.

### 3 Wissensgebiete der Simulationstechnik

Auf Grundlage von Örens Klassifikation [10] können die Wissensgebiete der Simulationstechnik auf der ersten Ebene in die Kategorien Grundlagen, Technik und Praxis gegliedert werden (Abbildung 1).

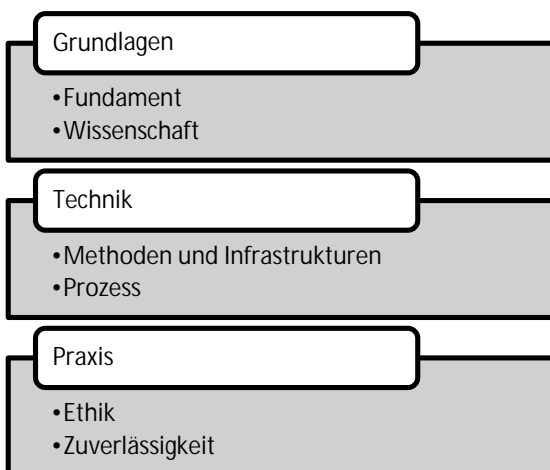


Abbildung 1. Wissensgebiete der Simulationstechnik

#### 3.1 Grundlagen

Die Teilgebiete der Grundlagen sind „Fundament“ und „Wissenschaft“ (Abbildung 2). Mit dem Fundament wird die Definition und Kategorisierung der Disziplin Simulationstechnik angesprochen. Die Arbeiten von Ören [9, 10, 13, 14, 18] und Birta [8, 19] sind dazu wertvolle Referenzen. In [18] hat Ören eine

umfangreiche Sammlung von Definitionen erstellt und bestimmt drei Hauptkategorien:

- Simulation als zielgerichtetes Experiment
- Simulation als vermittelte Erfahrung unter kontrollierten Bedingungen zum Training
- Simulation als vermittelte Erfahrung zur Unterhaltung

Des weiteren werden 400 verschiedene Simulationsarten beschrieben, z. B. Man-in-the-Loop-Simulation.

Die „Wissenschaft“ gehört einerseits zu den Grundlagen der Simulationstechnik, andererseits ist sie eine der drei Komponenten von M&S [20]. Sie befasst sich mit Daten, Modellen und Experimenten.

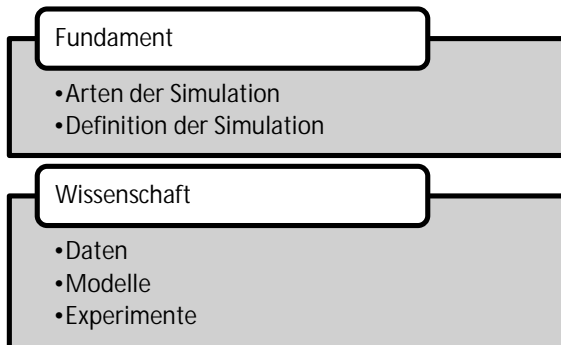


Abbildung 2. Wissens- und Teilgebiete der Grundlagen

Daten werden u. a benötigt, um ein Modell auszuführen und die Ergebnisse der Simulationsexperimente zu analysieren. Als Teilgebiet entspricht es beispielsweise in der ereignisorientierten Simulation den wissenschaftlichen Grundlagen der Eingangsdatenmodellierung, z. B. der Verteilungsfunktion der Intervallzeiten oder der Ausfallrate, und der Ausgangsanalyse, z. B. den Standardabweichungen der Wartezeiten. Unter den zahlreichen Lehrbüchern ist das Buch von Law und Kelton [21] eines der meistreferierten, die diese Themen ausführlich erklären.

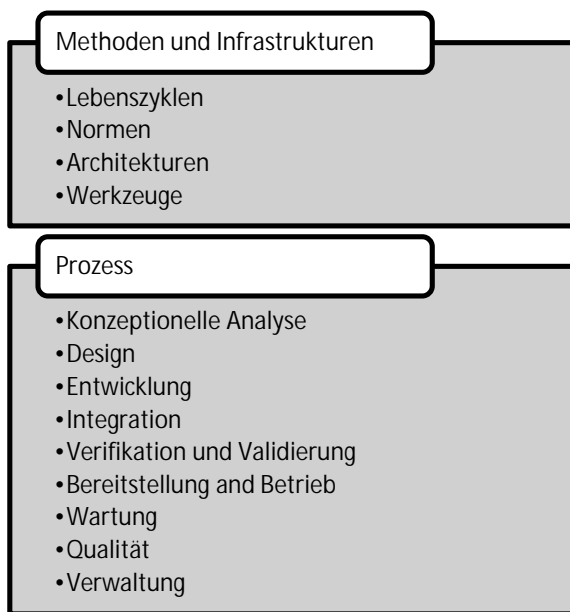
Modelle sind als Abstraktion der Realität ein Kernelement der Simulation. Die Modellierung und modellbasierte Entwicklung sind seit langem Gegenstand wissenschaftlicher Untersuchungen [22, 23]. Zu dem Thema „Modelle“ gehören die akademischen Grundlagen der Modellierung, der Modellverwaltung und der Modellbearbeitung. Maßgebliche Referenzen des Themas befassen sich z. B. mit Discrete Event System Specification (DEVS) [24], Modelica (mit objektorientierten Konzepten) [25] oder System Enti-

ty Structure/Model Base (als einen Ansatz für die Modellverwaltung) [26].

Die letzte Kategorie der „Wissenschaft“ sind die Experimente, welche eine Schlüsselstellung in jeder wissenschaftlichen Studie einnehmen. Hierbei geht es um das virtuelle Experimentieren mit Modellen bzw. die Ausführung eines Simulationsprogramms. In [18] hat Ören eine Kategorisierung verschiedener Experimentarten, die mit der Simulation verwandt sind, erstellt.

### 3.2 Technik

Die Kategorie „Technik“ beinhaltet Lösungsansätze auf Basis des Teilgebiets „Wissenschaft“ [20]. Auch hier legen wir Örens Klassifikation [10] zu Grunde und definieren die Wissensgebiete „Methoden und Infrastrukturen“ sowie „Prozess“ (Abbildung 3).



**Abbildung 3.** Wissens- und Teilgebiete der Technik

„Methoden und Infrastrukturen“ kann in vier Teilgebiete unterteilt werden: die Lebenszyklen, die Normen, die Architekturen und die Werkzeuge.

Der Lebenszyklus einer Simulation wurde ursprünglich von Balci in den neunziger Jahren definiert als ein aus zehn Phasen bestehender Prozess [27]. Später wurde ein Lebenszyklus (FEDEP - Federation Development and Execution Process) insbesondere für verteilte Simulationen, welche die High-Level-Architektur nutzen, entwickelt und als IEEE-Standard 1516.3-2003 veröffentlicht [28]. FEDEP wurde danach verallgemeinert, um einen Engineering-Prozess

für alle Arten der verteilten Simulationen zu entwickeln. Diese neue Norm, IEEE Std 1730-2010 IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP) [29], setzt sich aus sieben Schritten zusammen, u. a. „Konzeptionelle Analyse“ und „Design“. DSEEP ist gleichzeitig die Basis für das simulationstechnische Wissensgebiet „Prozess“.

Normen werden genutzt, um akzeptierte Methoden und Infrastrukturen zu dokumentieren. Tolk bietet einen umfassenden Überblick der Simulationsnormen in [30]. Die bekanntesten sind „IEEE 1278 Standard for Distributed Interactive Simulation (DIS)“ und „IEEE 1516 Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)“.

Unter „Architekturen“ wird in diesem Zusammenhang im Wesentlichen die Softwarearchitektur für Simulationen verstanden. Bekannte und weit verbreitete Simulationsarchitekturen umfassen z. B. HLA [31-33] und Test and Training Enabling Architecture (TENA) [34].

Das letzte Teilgebiet von „Methoden und Infrastrukturen“ ist „Werkzeuge“, deren Klassifizierung auf Grundlage der Ausführungsstrategie, z. B. Echtzeitsimulation, der Simulationsmechanik, z. B. agentenbasierte Simulation oder der Anwendungsdomäne, z. B. Supply-Chain-Simulation [35] oder Mehrkörpersimulation erfolgt [36].

Wie bereits erwähnt, ist DSEEP ein akzeptierter Prozess für die Simulationstechnik, der mit Systementwicklungsprozessen erweitert werden kann [37]. Die vorgeschlagenen Teilgebiete für das Wissensgebiet „Prozess“ sind die „Konzeptionelle Analyse“, „Design“, „Entwicklung“, „Integration“, „Verifikation und Validierung“, „Bereitstellung und Betrieb“, „Wartung“, „Qualität“ und „Verwaltung“.

Zwei primäre Aktivitäten der konzeptionellen Analyse sind die konzeptionelle Modellierung und die Szenarioentwicklung. Eine der wichtigsten Referenzen für die konzeptionelle Modellierung ist [38] von Pace. Er stellt die konzeptionelle Modellierung als eine Modellierungsaktivität vor, die sich am beabsichtigten Verwendungszweck der Simulation orientiert. Die Szenarioentwicklung ist eine umfassende Aktivität, die mit der Beschreibung des geforderten Simulationsszenarios durch den Nutzer beginnt und mit der Entwicklung der entsprechenden ausführbaren Szenariospezifikation endet [39].

Die Teilgebiete „Design“ und „Entwicklung“ können gemeinsam betrachtet werden. Sie beschreiben das Softwaredesign der Simulationsumgebung und z. B. für eine verteilte Simulation die Entwicklung von Datenaustauschmodellen und deren Implementierung. Gängige Praxis ist es, modellbasierte Methoden zu nutzen, die aus der Entwicklung von softwareintensiven Systemen übernommen wurden [40]. Modellbasierte Methoden unterstützen die Nutzung von Modellierung, Meta-Modellierung und Modelltransformation, um Modelle von einem technischen Kontext in einen anderen zu überführen, z. B. von einem Simulink-Modell zu C++-Quellcode.

„Integration“, als logischer Arbeitsschritt nach der Entwicklung, ist durch die Überführung aller Simulationsbestandteile in eine vereinheitlichende Betriebsumgebung definiert [29]. Ein sinnvolles Zusammenspiel von integrierten Simulationsbestandteilen, die sogenannte Interoperabilität, ist dabei eine der großen Herausforderungen [41].

„Verifikation und Validierung“ stellen einen wichtigen Schritt zur Qualitätssicherung dar. Während die Verifikation die reine Implementierung der Simulation betrachtet (Ist die Simulation richtig implementiert?), prüft die Validierung wiederum die hinreichende Erfüllung der Anforderungen an eine Simulation (Ist die richtige Simulation implementiert?) [42].

Für eine erfolgreiche Nutzung der Simulation muss eine „Bereitstellung“ durchgeführt werden. Sie bezieht sich auf die Aktivitäten zwischen der Entwicklung und der Freigabe der Simulation für den „Betrieb“ [43], und umfasst z. B. die rechtliche Prüfung der genutzten Lizenzen. Der eigentliche Betrieb befasst sich damit, die Simulation durchzuführen sowie Ergebnisse zu sammeln, zu analysieren und auszuwerten [29].

Während des Betriebes der Simulation ist ihre Evolution unvermeidlich. Gründe dafür sind beispielsweise die ständige Veränderung des Umfelds, z. B. durch andauernden technologischen Fortschritt, oder die durch zusätzliche Anforderungen weitergehende Entwicklung. „Wartung“ weist auf die Aktivitäten hin, die sicherstellen, dass die Simulation im laufenden Betrieb funktionsfähig bleibt, z. B. durch Fehlerbehebung, und den sich ändernden Anforderungen weiterhin entspricht [40].

Die Teilgebiete „Qualität“ und „Verwaltung“ umfassen klassische Managementthemen, wie Projektmanagement, Risikomanagement, Konfigurationsma-

nagement, Informationsmanagement und die Qualitätssicherung. Bisher wurden diese Teilgebiete in vielen technischen Bereichen systematisch betrachtet [2], aber noch nicht speziell für die Simulationstechnik.

### 3.3 Praxis

Die letzte Kategorie der Wissensgebiete der Simulationstechnik ist die simulationstechnische Praxis. Sie wird untergliedert in die Wissensgebiete „Ethik“ und „Zuverlässigkeit“. Ören schlägt vor, dass ein Kodex professioneller Ethik für Simulierer die Themengebiete a) persönliche und berufliche Entwicklung, b) fachliche Kompetenz, c) Vertrauenswürdigkeit, und d) Eigentumsrechte abdecken sollte [44]. Zuverlässigkeit ist wesentlich für die Akzeptanz einer Simulationsstudie [45], die sich aus der Akzeptanz von Simulationsergebnissen, Daten, Modellen und Parametern, Testspezifikationen, Programmen und Methoden zusammensetzt.

## 4 Ausblick

Die Kategorisierung der Wissensgebiete der Simulationstechnik ist einer der wichtigsten Pfeiler um eine etablierte Disziplin zu schaffen. Die vorgeschlagene Kategorisierung nutzt verschiedene bestehende Wissensgebiete, um eine eigene Einordnung zu ermöglichen. Es wurden kurze Definition und Referenzen der Kategorien und Wissens- und Teilgebiete gegeben, jedoch sind umfangreichere Ausarbeitungen erforderlich. Die Schaffung einer klaren Definition der Wissensgebiete der Simulationstechnik kann die Grundlagen für eine professionelle Ausbildung legen, z. B. durch zertifizierte Lehrgänge oder individuelle Lizenzierung. Wir hoffen, dass diese Publikation einen guten Ausgangspunkt für weitere Bemühungen zu diesem Thema bildet.

## 5 References

- [1] A. Tolk, O. Balci, C.D. Combs, R. Fujimoto, C. Macal, B. Nelson, und P. Zimmerman. *Do We Need a National Research Agenda for Modeling and Simulation?* Winter Simulation Conference, Huntington Beach, CA, 2015.
- [2] BKCASE Editorial Board. *The Guide to the Systems Engineering Body of Knowledge (SE-BoK)*, v. 1.6. R.D. Adcock. Hoboken, NJ, 2016. Webseite: [www.sebokwiki.org](http://www.sebokwiki.org). 03 Januar 2017.

- [3] P. Bourque, R.E. Fairley. *Guide to the Software Engineering Body of Knowledge (SWEBOK®): Version 3.0*. IEEE Computer Society Press, Los Alamitos, CA, 2014.
- [4] Project Management Institute, Inc. *Guide to the Project Management Body of Knowledge (PMBOK®)—Fifth Edition*. PMI Publications, Newtown Square, PA, 2013.
- [5] R. Rogers. *What Makes A Modeling and Simulation Professional? The Consensus View from one Workshop*. Winter Simulation Conference (WCS), Atlanta, GA, 1997.
- [6] H. Szczerbicka, J. Banks, T.I. Ören, R.V. Rogers, H.S. Sarjoughian, und B.P. Zeigler. *Conception of Curriculum for Simulation Education (panel)*. Winter Simulation Conference (WCS), Orlando, FL, 2000.
- [7] B. Waite. *M&S Professional Body of Knowledge / Code of Ethics*. Summer Computer Simulation Conference (SCSC), Orlando, FL, 2001.
- [8] L.G. Birta. *The Quest for the Modelling and Simulation Body of Knowledge*. Proceedings of the 6th Conference on Computer Simulation and Industry Applications, Tijuana, Mexico, 2003.
- [9] T.I. Ören. *Body of Knowledge of Modeling and Simulation (M&SBOK): Pragmatic Aspects*. Proceedings of the 2nd European Modeling and Simulation Symposium, Barcelona, Spain, 2006.
- [10] T.I. Ören. *Modeling and Simulation Body of Knowledge (M&S BoK) - Index*. Webseite: <http://www.site.uottawa.ca/~oren/MSBOK/MSBOK-index.pdf>. 03 Januar 2017.
- [11] T.I. Ören, und B. Waite. *Modeling and Simulation Body of Knowledge Index: An Invitation for the Final Phases of its Preparation*. SCS M&S Magazine, 1(4), 2010.
- [12] L.W. Lacy, D.C. Gross, T. Ören, und B. Waite. *A Realistic Roadmap for Developing a Modeling and Simulation Body of Knowledge Index*. Proceedings of Fall Simulation Interoperability Workshop, Orlando, FL, 2010.
- [13] T.I. Ören. *A Basis for a Modeling and Simulation Body of Knowledge Index: Professionalism, Stakeholders, Big Picture, and Other BoKs*. SCS M&S Magazine, 2(1):40-48, 2011.
- [14] T.I. Ören. *The Richness of Modeling and Simulation and an Index of its Body of Knowledge*. In: M.S. Obaidat, J. Filipe, J. Kacprzyk und N. Pina (eds) *Simulation and Modeling Methodologies, Technologies and Applications, Advances in Intelligent Systems and Computing*, Vol. 256, Springer, 2014.
- [15] J.R. Brown. *Thought Experiments*. Stanford Encyclopedia of Philosophy. Webseite: <http://plato.stanford.edu/entries/thought-experiment/>. 03 Januar 2017.
- [16] L. Yilmaz, und T.I. Ören (Eds.) *Agent-Directed Simulation and Systems Engineering*. Wiley-VCH Verlag, Weinheim, Germany, 2011.
- [17] U. Durak, und T. Ören. *Towards an Ontology of Simulation Systems Engineering*. SpringSim-ANSS, Pasadena, CA, 2016.
- [18] T. Ören. *A Critical Review of Definitions and About 400 Types of Modeling and Simulation*. SCS M&S Magazine, 2(3):142-51, 2011.
- [19] L.G. Birta. *A Perspective of the Modeling and Simulation Body of Knowledge*. SCS M&S Magazine, 2(1), 2003.
- [20] J.J. Padilla, S.Y. Diallo, und A. Tolks. *Do We Need M&S Science?*. SCS M&S Magazine, 8, 2011.
- [21] A.M. Law, und W.D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York, NY, 1991.
- [22] B.P. Zeigler, M.S. Elzas, J.G. Klir, und T.I. Ören. *Methodology in System Modelling & Simulation*. North-Holland, Amsterdam, 1979.
- [23] T.I. Ören. *Model-Based Activities: A Paradigm Shift*. In: *Simulation and Model-Based Methodologies: An Integrative View*. (eds T.I. Ören, B.P. Zeigler, M.S. Elzas) Springer-Verlag, Heidelberg, Germany, 1984.
- [24] B.P. Zeigler, H. Praehofer, und T.G. Kim. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, San Diego, CA, 2000.

- [25] P. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. John Wiley & Sons, Piscataway, NJ, 2010.
- [26] B.P. Zeigler, C.J. Luh, und T.G. Kim. *Model Base Management for Multifaceted Systems*. ACM Trans. Model. Comput. Simul., 1(3):195-218, 1991.
- [27] O. Balci. *Guidelines for Successful Simulation studies*. 22nd Winter Simulation Conference, New Orleans, LA, 1990
- [28] IEEE STD 1516.3-2003. *IEEE Recommended Practice for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP)*, 2003.
- [29] IEEE STD 1730TM-2010. *IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)*, 2010.
- [30] A. Tolk. *Standards for Distributed Simulation*. In: Engineering Principles of Combat Modeling and Distributed Simulation. John Wiley & Sons, Inc., Hoboken, NJ, 2012, pp. 209-241
- [31] IEEE STD 1516-2010. *IEEE Standard for Modeling and Simulation High Level Architecture (HLA) – Framework and Rules*, 2010.
- [32] IEEE STD 1516.1-2010. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification*, 2010.
- [33] IEEE STD 1516.2-2010. *IEEE Standard for Modeling and Simulation High Level Architecture (HLA) – Object Model Template (OMT) Specification*, 2010.
- [34] J.R. Noseworth. *The Test and Training Enabling Architecture (TENA) Supporting the Decentralized Development of Distributed Applications and LVC Simulations*. 12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications, British Columbia, Canada, 2008.
- [35] J.P. Kleijnen. *Supply Chain Simulation Tools and Techniques: A Survey*. International Journal of Simulation and Process Modelling, 1(1-2):82-9, 2001.
- [36] W. Schiehlen (ed). *Advanced Multibody System Dynamics: Simulation and Software Tools*. Springer Science & Business Media, 2013.
- [37] A. D'Ambrogio, und U. Durak. *Setting Systems and Simulation Life Cycle Processes Side by Side*. IEEE International Symposium on Systems Engineering, Edinburgh, Scotland, 2016.
- [38] D.K. Pace. *Ideas about Simulation Conceptual Model Development*. Johns Hopkins APL Technical Digest, 21(3):327-36, 2000.
- [39] U. Durak, O. Topçu, R. Siegfried, und H. Oğuztüzün. *Scenario Development: A Model-Driven Engineering Perspective*. 2014 International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH), Wien, Austria, 2014.
- [40] O. Topcu, U. Durak, H. Oğuztüzün, und L. Yilmaz. *Distributed Simulation - A Model Driven Engineering Approach*. Springer International Publishing, Cham, Germany, 2016.
- [41] A. Tolk. *Interoperability, Composability, and Their Implications for Distributed Simulation: Towards Mathematical Foundations of Simulation Interoperability*. IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications, 2013.
- [42] R.G. Sargent. *Verification and Validation of Simulation Models*. 37th Winter Simulation Conference, Orlando, FL, 2005.
- [43] U. Durak, A. Öztürk, und M. Katircioglu. *Simulation Deployment Blockset for MATLAB/Simulink*. SpringSim-TMS/DEVS, Pasadena, CA, 2016.
- [44] T.I. Ören, M.S. Elzas, I. Smit, und L.G. Birta. *A Code of Professional Ethics for Simulationists*. 2002 Summer Computer Simulation Conference, San Diego, CA, 2002.
- [45] T.I. Ören. *Concepts and Criteria to Assess Acceptability of Simulation Studies: A Frame of Reference*. Communications of the ACM, 24(4):180-189, 1981.

# Parameter der arteriellen Pulswellenanalyse zur modellbasierten Charakterisierung eines reduzierten Herzauswurfs

Andreas Bauer<sup>1,2</sup>, Bernhard Hametner<sup>1</sup>, Felix Breitenecker<sup>2</sup>, Thomas Weber<sup>3</sup>, Siegfried Wassertheurer<sup>1</sup>

<sup>1</sup>Center for Health & Bioresources, AIT Austrian Institute of Technology, Wien, Österreich

<sup>2</sup>Institut für Analysis und Scientific Computing, Technische Universität Wien, Wien, Österreich

<sup>3</sup>Kardiologie, Klinikum Wels-Grieskirchen, Wels, Österreich

*andreas.bauer@ait.ac.at*

In dieser Arbeit werden Methoden der Pulswellenanalyse präsentiert, die speziell zur vorklinischen Erkennung von systolischem Herzfehler aus arteriellen Pulswellenmessungen verwendet werden können. Mithilfe der Auswurfdauer und der Wellenintensität kann potenziell ein reduzierter Herzauswurf, der diese Art von Herzfehler charakterisiert, bestimmt werden. Für die Bestimmung der Auswurfdauer wurde ein Algorithmus entwickelt. Dieser wurde auf Pulswellenmessungen von Patienten einer Vergleichsstudie mit zwei Messgeräten angewandt und die Ergebnisse wurden miteinander sowie mit Ergebnissen eines in einem der Geräte implementierten Algorithmus verglichen. In weiterer Folge soll auf Basis der hier dargestellten Parameter ein Entscheidungsmodell entwickelt werden, das eine Klassifizierung von Patienten hinsichtlich eines potenziell vorliegenden systolischen Herzfehlers durchführt.

## 1 Einleitung

Ein Herzfehler mit reduzierter Auswurfleistung ist eine Erkrankung des Herzens, die vorwiegend bei Menschen fortgeschrittenen Alters verbreitet ist. Dabei wird durch eine Schwächung des Herzmuskels ein geringerer Teil des im linken Ventrikel befindlichen Bluts ausgestoßen als bei gesunden Menschen. Dies führt zu einer krankhaften Vergrößerung des Herzens. Diese Erkrankung wird auch Systolischer Herzfehler (SHF) genannt. Die herkömmlichen Methoden der Pulswellenanalyse, die sich nur der arteriellen Druckwelle bedienen, liefern für Patienten mit erhaltener Herzfunktion gute Ergebnisse, jedoch sind sie für Patienten mit SHF nur bedingt gültig, da diese allgemein ein normales Blutdruckniveau aufweisen. Zusätzlich sind auch die Parameter der Pulswellenanalyse für diese Patienten in einem ähnlichen Bereich wie bei gesunden Menschen. Deshalb sind andere Wege notwendig, um systolischen Herzfehler vorklinisch zu erkennen und dann auch Therapien begleiten zu können. Eine Analyse der Form der Blutdruckkurve kann Aufschlüsse über die Existenz eines systolischen Herzfehlers geben. Aufgrund der Ergebnisse vorangegangener Arbeiten wird einerseits auf die Bestimmung der Auswurfdauer, andererseits auf die sogenannte Wellenintensitätsanalyse, bei der Pulswellen in einzelnen Wellenfronten zerlegt und analysiert werden, spezielles Augenmerk gelegt [1-5].

## 2 Methoden der Pulswellenanalyse

### 2.1 Auswurfdauer

Die Auswurfdauer entspricht genau jener Zeitdauer während der die Herzklappen geöffnet sind und Blut in das Arteriensystem ausgeworfen wird. Dies entspricht auch der Länge der Systole. Eine verkürzte Auswurfdauer ist ein möglicher Indikator für einen systolischen Herzfehler.

Im Zuge dieser Arbeit soll die Auswurfdauer aus einem gemessenen Blutdrucksignal bestimmt werden. Dazu gibt es in der Literatur verschiedene Ansätze. Auch über die genaue Lage des Endpunkts der Systole herrscht keine Einigkeit [6-8].

Für diese Studie wurde folgender Ansatz verwendet. Zuerst wurde für die Auffindung des Endpunkts der Systole ein Intervall definiert, welches vom Zeitpunkt des maximalen Blutdrucks und der Herzrate abhängt. Dadurch soll die Fehleranfälligkeit des Algorithmus reduziert werden. Innerhalb dieses Intervalls wurde die erste Nullstelle der dritten Ableitung vom positiven in den negativen Bereich bestimmt und als Endpunkt der Systole definiert.

Bei 105 Patienten wurden im Rahmen einer Vergleichsstudie in direkter Abfolge 216 Messungen der Pulswellen der Aorta mittels Mobil-O-Graph (IEM, Stolberg, Deutschland) und SphygmoCor system (AtCor Medical Pty. Ltd., West Ryde, Australien)

durchgeführt. Ersteres Gerät führt automatisierte, oszillometrische Messungen am Oberarm durch, letzteres manuelle, tonometrische Messungen am Unterarm. Von SphygmoCor wurden auch die Ergebnisse des implementierten Algorithmus bezogen.

## 2.2 Wellenintensitätsanalyse

Die Wellenintensitätsanalyse (WIA) ist ein Konzept, das zur Analyse von Wellen in Blutgefäßen eingesetzt werden kann. Dabei werden die Wellen als aufeinanderfolgende Wellenfronten beschrieben, die die Änderung bestimmter Größen während eines Zeitschritts  $\Delta t$  beschreiben. Die Analyse findet im Zeitbereich statt, wodurch sich Auffälligkeiten bei den Ergebnissen zeitlich verorten lassen [9].

Ausgangspunkt für die Herleitung der Wellenintensität sind die eindimensionalen Euler Gleichungen für inkompressible und nichtviskose Fluide. Drei Zustandsgrößen sind dabei involviert. Die Querschnittsfläche der Arterie  $A$ , die über den Querschnitt gemittelte Geschwindigkeit  $U$  und der über den Querschnitt gemittelte Druck  $P$ .

Die Gleichungen beruhen auf Massen- und Impulserhaltung. Durch die Massenerhaltung muss die Änderung des Volumens eines differenziellen Elements gleich der Differenz der Volumsflussraten in das Element hinein und aus dem Element heraus sein. Die Zeit wird mit  $t$  bezeichnet und  $x$  ist die Strecke entlang des Rohrs.

$$A_t + (UA)_x = 0 \quad (1)$$

Für die Gleichung zur Impulserhaltung wird die Beschleunigung im Fluid dem Impulsstrom in das Element und der Kraft, die durch den Druck auf das Element ausgeübt wird, gleichgesetzt.  $\rho$  bezeichnet dabei die Blutdichte.

$$U_t + UU_x = -\frac{P_x}{\rho} \quad (2)$$

Um das System von zwei Gleichungen mit drei Zustandsgrößen lösbar zu machen, wird ein Zusammenhang zwischen der Querschnittsfläche und dem Druck in einem Rohr verwendet. Daraus resultiert folgendes System:

$$\begin{pmatrix} P \\ U \end{pmatrix}_t + \begin{pmatrix} U & \frac{A}{A_p} \\ \frac{1}{\rho} & U \end{pmatrix} \begin{pmatrix} P \\ U \end{pmatrix}_x = \begin{pmatrix} \frac{UA_x}{A_p} \\ 0 \end{pmatrix} \quad (3)$$

Die Eigenwerte der Systemmatrix sind  $\lambda_{1,2} = U \pm c$ , wobei  $c = \sqrt{\left(\frac{A}{\rho A_p}\right)}$  die Ausbreitungsgeschwindigkeit der Welle im Rohr beschreibt.

Unter der Annahme, dass  $c$  konstant im ganzen Rohr ist, kann das gegebene System partieller Differentialgleichungen mittels der Methode der Charakteristiken gelöst werden [10]. Daraus ergeben sich die sogenannten Waterhammer-Gleichungen für Vorwärts- und Rückwärtswellen ( $f, b$ ):

$$dP_f = \rho c dU_f \quad (4)$$

$$dP_b = -\rho c dU_b \quad (5)$$

Unter der Annahme der Linearität kann die gemessene Welle ( $m$ ) als Summe der vorwärts- und der rückwärtslaufenden Wellen dargestellt werden:

$$dP_m = dP_f + dP_b \quad (6)$$

$$dU_m = dU_f + dU_b \quad (7)$$

Unter der Annahme, dass zu Beginn der Systole nur vorwärtslaufende Wellen vorhanden sind, können die Waterhammer-Gleichungen folgendermaßen formuliert werden:

$$dP_m = \rho c dU_m \quad (8)$$

Dies führt zu einem linearen Zusammenhang zwischen Druck und Fluss und damit kann ein Wert für die Wellengeschwindigkeit  $c$  berechnet werden [11].

Die Intensität der Vorwärtswelle kann folgendermaßen definiert werden:

$$dI_f = dP_f dU_f \quad (9)$$

Die Intensität der Rückwärtswelle wird analog definiert.

Die Einheit dieser Größe ist Arbeit pro Fläche ( $W/m^2$ ). Das Vorzeichen von  $dI$  gibt an, ob die Wellenfront sich vorwärts oder rückwärts ausbreitet.

Die vorwärtslaufenden Wellenintensitäten werden durch eine reduzierte Auswurfleistung beeinflusst und können daher potenziell zur Bestimmung eines systolischen Herzfehlers herangezogen werden.

## 3 Resultate

Im Folgenden werden die Ergebnisse des Algorithmus zur Bestimmung der Auswurfleistung für die Datensätze von Mobil-O-Graph (Mobil) und SphygmoCor (Sphygmo) präsentiert und miteinander

sowie mit den Werten des SphygmoCor Systems (SphygmoInt) verglichen.

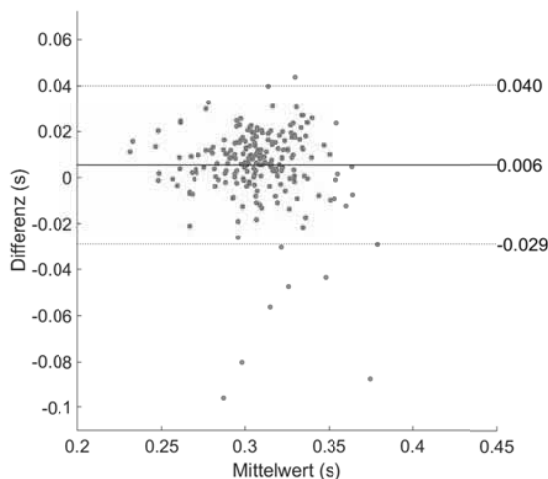
In Tabelle 1 sind der mittlere Fehler und die Standardabweichung der Ergebnisse für Mobil und Sphygmo bezüglich der von SphygmoCor berechneten Auswurfdauer dargestellt.

Abweichung zu SphygmoInt von	Sphygmo	Mobil
Mittlerer Fehler (s)	0,010	0,014
Standardabweichung (s)	0,004	0,013

**Tabelle 1.** Absoluter mittlerer Fehler und Standardabweichung der berechneten Auswurfzeiten gegenüber den Ergebnissen von SphygmoInt in s

Die Werte in Tabelle 1 zeigen, dass die Ergebnisse in einem ähnlichen Bereich liegen, wobei der Algorithmus für den SphygmoCor-Datensatz leicht bessere Ergebnisse liefert. Dies kann unter anderem an der Qualität des Signals liegen.

Zur weiteren Analyse wurde die Ergebnisse der Datensätze Sphygmo und Mobil in einem Bland-Altman-Plot einander gegenübergestellt, siehe Abbildung 1.



**Abbildung 1.** Bland-Altman-Plot der Auswurfdauer für die Datensätze Sphygmo und Mobil.

In Abbildung 1 ist zu sehen, dass die Ergebnisse für die SphygmoCor Daten im Schnitt 6 ms größer sind. Die Abweichungen sind weitgehend unabhängig von der Größe der Werte.

## 4 Conclusio und Ausblick

In dieser Arbeit wurde ein Ansatz präsentiert um systolischen Herzfehler anhand von nicht-invasiven Blutdruckmessungen zu detektieren. Weiters wurde ein funktionierender Algorithmus zur Berechnung der Auswurfdauer des Herzens implementiert und an verschiedenen Datensätzen getestet. Der Vergleich zeigte zufriedenstellende Ergebnisse. Die Abweichung zwischen den Methoden ist im Vergleich zu den physiologischen Unterschieden zwischen gesunden und SHF-Patienten klein.

Basierend auf der Auswurfdauer und Parametern der Wellenintensitätsanalyse wird ein Entscheidungsmodell entwickelt werden, das eine Klassifizierung von Patienten hinsichtlich eines potenziell vorliegenden systolischen Herzfehlers durchführt.

## 5 Referenzen

- [1] S. Parragh, B. Hametner, M. Bachler, J. Kellermair, B. Eber, S. Wassertheurer und T. Weber. *Determinants and covariates of central pressures and wave reflections in systolic heart failure*. International Journal of Cardiology, 308-314, 2015.
- [2] S. Parragh, B. Hametner, M. Bachler, B. Eber, S. Wassertheurer und T. Weber. *Non-invasive wave reflection quantification in patients with reduced ejection fraction*. Physiological Measurement, S. 179-190, 2015.
- [3] B. Hametner, M. Bachler, S. Parragh, C. Mayer, T. Weber und S. Wassertheurer. *Pulse Wave Intensity und ECG: a Multisensor Approach for the Risk Assessment in Systolic Heart Failure*. Tagungsband IEEE Sensors Applications Symposium, 2015.
- [4] B. Hametner, S. Wassertheurer, S. Parragh, B. Eber und T. Weber. *Diagnostic Value of pressure waveform based analysis for impaired systolic function*. Journal of Hypertension. S. e97, 2015.
- [5] S. Parragh. *Mathematical Models for Pulse Wave Analysis Considering Ventriculo-Arterial Coupling in Systolic Heart Failure*. Dissertation. TU Wien, Österreich, 2017.
- [6] M. J. Oppenheim und D. F. Sittig. *An innovative dicrotic notch detection algorithm which combines rule-based logic with digital signal*

- processing techniques*. Computers and biomedical research, S. 154-170, 1995.
- [7] C. F. Starmer, P. A. McHale und J. C. Greenfield. *Processing of arterial pressure waves with a digital computer*. Computers and biomedical research. S 90-96, 1973.
- [8] R. Blazek und C. Lee. *Multi-resolution linear model comparison for detection of dicrotic notch and peak in blood volume pulse signals*. Analysis of Biomedical Signals and Images, S. 378-386, 2010.
- [9] K.H. Parker. *An introduction to wave intensity analysis*. Medical & Biological Engineering & Computing, S. 111-118, 2009.
- [10] B. Riemann. *Über die Fortpflanzung ebener Luftwellen von endlicher Schwingungsweite*. Abhandlungen der Königlichen Gesellschaft der Wissenschaften zu Göttingen. S. 43-65, 1860.
- [11] A. Zambanini, S. Cunningham, K. Parker, A. Khir, S. Thom und A. Hughes. *Wave-energy patterns in carotid, brachial, and radial arteries: a noninvasive approach using wave-intensity analysis*. American Journal of Physiology – Heart and Circulatory Physiology, S. H270-H276, 2005.

# A Microscopic Model for Simulation of Time Efficiency Concerning the Supply in Steep Terrain

Johannes Tanzler<sup>1</sup>, Martin Bicher<sup>2</sup>, Felix Breiteneker<sup>1</sup>

<sup>1</sup>Inst. Analysis and Scientific Computing, TU Wien, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria

<sup>2</sup>dwh GmbH, dwh simulation-services, Neustiftgasse 57-59, 1070 Vienna, Austria

*johannes.tanzler@tuwien.ac.at*

In this contribution a microscopic model is used for getting a better understanding of how an inhabited institution in steep terrain, which can only be accessed by walking, could be supplied efficiently and if a second inhabited supplying settlement would make sense. The simulation is applied to the prehistoric salt mining village of Hallstatt. This village is under close investigation by archaeologists and one method for this investigation is the computer simulation. To get a better idea of how these Bronze Age people have lived especially how they could have managed the farming and the supply of the salt mine and how much time they may have used for this, this project was realised. Accommodating the steep terrain and a high share in travelling time an A\* algorithm is used for the path calculation. Different tasks for farming and the wood supply are modelled and because of the relatively great distance between the settlement and the agriculturally used area the possibility of a second farming settlement is taken into account.

## 1 Introduction

The motivation for this contribution was how the prehistoric settlement in Hallstatt, a settlement which had the purpose of salt mining could have been supplied with food and wood. This settlement is located close to the salt mine up on a mountain and surrounded with partially very steep terrain. Because of this impracticable and unfertile areas close by, the main agricultural cultivable land is assumed to be located in the valley at the bottom of the mountain. This leads to a long travel to the agriculturally useable fields and therefore the path calculation is an important part of this project.

Besides the path calculation also the tasks which had to be executed on the fields like seeding, mowing and harvesting as well as the tasks concerning the wood work are modelled. The wood was important for maintaining the salt mining process especially to lighten the mine and to build stairs and other construction for the mine.

Another important part is the population size. A primary assumption is that all supply is produced locally and in a previous work [1] the maximum of persons which could have been supported by the surrounding land was calculated. A result of this previous work is that 72 people could have lived in Hallstatt and therefore this is the number of people which in this model has to manage the mine as well as all the supporting tasks.

The main focus of this project lies on the time consumption for all the tasks and how much time was left to work in the salt mine. Therefore the population size is very important because it directly effects the time left for working in the mine.

This model built as an agent-based model [2] and is implemented in Anylogic [3]. Every Person is represented as an agent and has its own attributes like age and gender and is placed in a kind of a three-dimensional environment.

## 2 Environment

As environment a predefined rectangular map of the surrounding of Hallstatt is used which archaeologists think that could have been counted to the area of Hallstatt. The environment consists of two parts, on the one hand of a three-dimensional elevation model and on the other hand of a two-dimensional shapefile [4]. The elevation model is only used for calculating the paths and the speed of the agents but for the representation in the simulation only the two-dimensional shapefile is used.

Archaeologists provided the shapefiles which contain the information about the shapes of the different fields as well as the types of the fields. Unfortunately it is not possible to get real data from the Bronze Age of Hallstatt so the oldest map available was used, a map created for fiscal purposes in the 19<sup>th</sup> century.

This map consists of the attributes ‘wood’, ‘rock’, ‘water’, ‘grassland’, ‘garden’ and ‘field’. The area with the attribute ‘rock’ is not relevant for this model just like the areas with the attribute ‘water’, but in contrast to water persons can walk on rocks. Fields with the attribute ‘garden’ are used for the production of beans and the total size of all fields with this attribute is 16 hectare. Fields with the attribute ‘field’ are used for two different crops; half of every field is tilled with sorghum and the other half with barley. The total area tilled with sorghum and barley is about 13 hectare. Concerning the grassland just a part of its total size is used for production because its total size would be about 650 hectare this could provide much more animals as needed to feed a population of 72 persons. Further more data provided by archeologists suggest a balanced nutrition which means that about a third of the nutrition was meat. Therefore just about 100 hectare of grassland is included to this simulation.

Figure 1 shows the total area considered to be part of Hallstatt at this time.



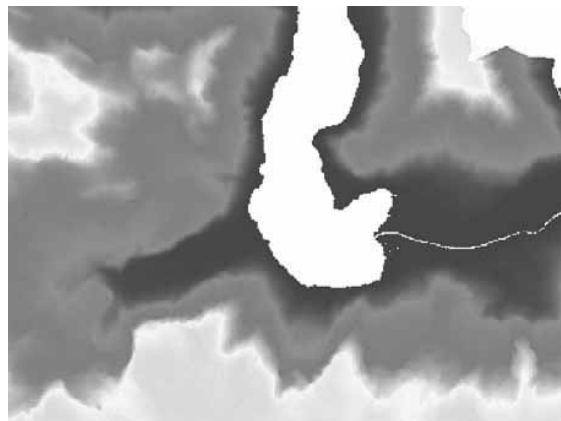
**Figure 1.** Area used for the simulation

### 3 Path Calculation

As mentioned before the surrounding of the mine is very steep and therefore a simple two-dimensional path calculation would not be suitable and therefore an A\* algorithm is used. This algorithm is an extension of the Dijkstra algorithm and works with weighted graphs. It returns the path with the lowest weigh which means the fastest path. A path which is a little bit downward slopping is the fastest one and the others are proportionally slower.

The downside of this powerful algorithm is that the calculation is very time as well as memory consuming. Although the calculated paths are saved into a database the grid of the elevation map is reduced to 25 meters.

The following Figure 2 shows the elevation map. The white area in the middle is the lake and the different colors show the height of the terrain. Figure 3 shows agents on their path from the hill down to the valley and back up. The putative loop way gives an impression how steep the direct way must be and why a two dimensional path would not be sufficient.



**Figure 2.** Elevation map



**Figure 3.** Path example

### 4 Population

The population size as mentioned above is taken from a previous work and is set to 72. On the other hand the age structure of the population is basically taken from another project called ‘mining with agents’ [5]. The important information provided by the age structure is how many persons are children and too young to work. According to archaeological findings also young children were working in the mine but in this model it is assumed that children younger than 7 years were too young to work.

The count of children younger than 7 years is according to this age structure 13, which means that only 59 people had to do the work. Each one of these 59 people is modelled as an agent and can work off assigned tasks.

## 5 Tasks

Five tasks are modelled which is done in a quite simple way to keep the model relatively clear and to keep the count of uncertain parameters on a lower level. All the tasks are modelled by a working rate, how much could have been produced in a time step of one minute.

Concerning ‘seeding’, ‘mowing’ and ‘harvesting’ the working rate has to be calculated because it is much more intuitive to estimate the time which is needed to process one hectare of land. For these three tasks it is assumed that it takes 10 hours to process one field. In contrast to seeding and mowing harvesting not just consists of processing the land but also of transporting the harvest back to the village. Therefore a transport capacity has to be set and is assumed to be 20 kg for every agent. To get an impression on how much work the transport is it should be mentioned that every field is depending on its crop yields of 300 to 760 kg per hectare.

Another task is the ‘wood production’ which is simplified in this way that all the wood is harvested from one spot in the woods and then drawn to the mine. Archaeologists found evidence that wood was drawn with a special method and therefore the transport capacity concerning wood is set to 50 kg. The harvest rate for wood is set to 20 kg per minute.

The wood which was transported to the mine has to be chipped to be used for lighting purposes and this task is assumed to be done with a rate of 1 kg chips per minute.

### 5.1 Scheduling of the workload

The scheduling of the workload is driven by the agricultural tasks and the supply needs of the mine. The agricultural tasks are scheduled by specific dates when it is assumed to be the optimal day to process the tasks. At this date the specific task is scheduled and assigned to the agents. On the contrary the tasks of providing wood and chopsticks are need driven, if the stock falls below a certain mark the task ‘wood production’ is scheduled and the same with ‘wood chip production’.

Every morning each agent gets a scheduled task assigned. The amount of agents working on one task is dependent on how many agents are needed for a task to be processed in one day if the task is of agricultural purposes. This is implemented to minimize the traveling time between fields. If a task cannot be finished anyway before the evening it gets scheduled again for the next day. In case there is no agricultural task scheduled and there are enough wood chips available every agent is sent to work in the mine.

### 5.2 Supply settlement

Because of the fact that all the agricultural land beside the grassland is located in the valley it raises the question if it would make sense to have a second settlement installed in the valley. Therefore food which has been produced by the people living in the valley only has to be transported to this settlement and will be brought to the uphill settlement when the people living in the valley go to work in the mine.

## 6 Results

The main result of the simulation is the time which could have been spent on working in the mine. But concerning the comparison between the scenario of one and two settlements the time used for food production seems more useful.

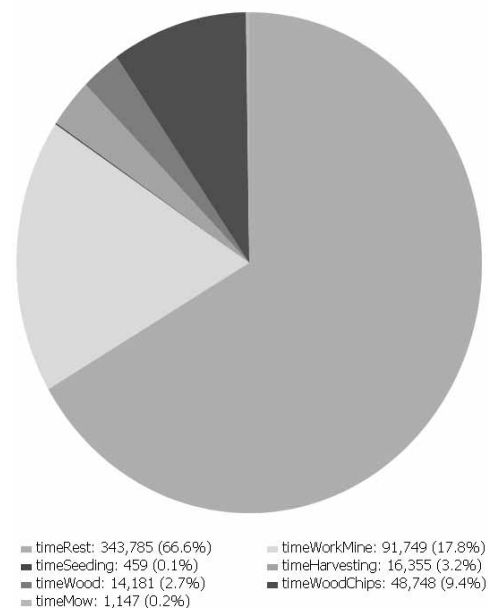


Figure 4. Time usage

### 6.1 Results on scenario - one settlement

The simulation results of the standard model with one settlement are shown in Figure 4. It can be seen that the seeding time is 459 hours which includes the travelling time to the fields the actual time used for seeding at the field is easily calculated and is 290 hours which means that about 40% of the time used for seeding is caused by the traveling time. The time used for harvesting itself is also 290 hours but because of the time needed for transport the altogether time consumption is 16355 hours.

### 6.2 Results on Scenario – two settlements

The results of this scenario are shown in Figure 5 by comparing the harvest time and the amount of people living in the valley. This result shows that the harvest time decreases rapidly if more people would live in the valley. The raise when more than 45 people live in the valley is caused by the fact that then the people living in the valley have to go uphill to harvest the grass beside the mine because the grass in the valley is exhausted and not enough people are living uphill to harvest this grass. Despite the overall time which could be spent in the mine may decrease the more people live in the valley there are other factors like that the running to seed of crops which are not harvested at time which may legitimate a second settlement.

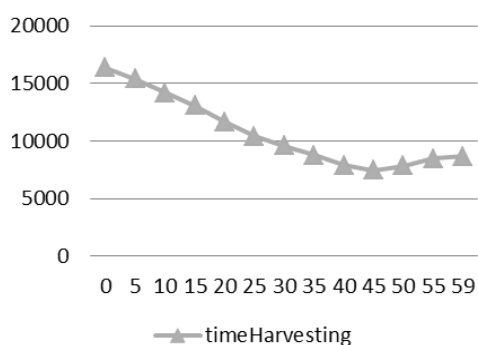


Figure 5. Harvest time versus people

## 7 Conclusion

This model is meant to be the basis for analysing different archaeological thoughts on the prehistoric mining village of Hallstatt. The model is quite easy adjustable in terms of including further tasks to improve the relation to reality. Concerning the tasks itself the level of detailedness is low but therefore easily adjustable and it seems more useful to use other methods like experimental archaeology to im-

prove the accuracy of the modelled tasks. Although the used parameters for modelling the single tasks are not yet driven from experimental archaeology the results of the basic model should give a rough impression on how much time was used for the tasks especially the share of the time consumption for transport in harvesting.

The second scenario where a second settlement is included is an example on how this model can be used for analysing archaeological thoughts. Archaeologists think that there could have been a second settlement for supporting the mining settlement and the results of the second scenario show that from an agricultural point of view it would actually make sense.

Like mentioned before the model is quite easily adjustable and could therefore maybe also be used for other applications for instance by enlarging the size of the map a route for transporting the salt to other villages could be investigated or it could maybe even be used in some todays applications.

## 8 References

- [1] J Tanzler, P Pichler, K Kowarik, H Reschreiter, G Wurzer, A Bacher, F Breiteneker. *Modelling agricultural constraints for population size in prehistoric Hallstatt*. ERK - International Electrotechnical and Computer Science Conference. 2013.
- [2] F Breiteneker, M Bicher, G Wurzer. *Agent-based modeling and simulation in archaeology, chapter: Agent-Based Simulation in Archaeology: A Characterization*. [S.l.]: Springer, 2013.
- [3] Grigoryev I. *AnyLogic 6 in three days: a quick course in simulation modeling*. AnyLogic North America, Hampton, NJ, 2012.
- [4] ESRI Shapefile Technical Description, An ESRI White Paper-July 1998.
- [5] K Kowarik, H Reschreiter, G Wurzer, R Tot-schnig, A Rausch. *Mining with Agents - Agent-based Modeling of the Bronze Age Salt Mine of Hallstatt*, Talk: Archäologie und Computer, Wien; 11-03-2008 - 11-05-2008; in: "Workshop 13 - Archäologie und Computer 2008", CD der Stadt Wien, Stadtarchäologie, Wien (2009), ISBN: 978-3-85161-016-1; 1 - 19.

# Fallstudien für die aggregierte Analyse von agentenbasierten Modellen mithilfe von Markovketten

Florian Kitzler<sup>1</sup>, Martin Bicher<sup>1</sup>, Niki Popper<sup>2</sup> Felix Breiteneker<sup>1</sup>

<sup>1</sup>TU Wien, Institut für Analysis und Scientific Computing,  
Wiedner Hauptstraße 8-10, 1040 Wien, Österreich

<sup>2</sup>dwh GmbH, dwh simulation-services,  
Neustiftgasse 57-59, 1070 Wien, Österreich  
*florian.kitzler@tuwien.ac.at*

Gerade für Fragestellungen aus den Sozialwissenschaften hat sich in den letzten Jahren die Anwendung sogenannter agentenbasierter Modelle (ABM) stark etabliert. Einerseits lässt die flexible “Bottom-Up” Modellierung eines ABM die Simulation von fast beliebig komplexen, heterogenen Systemen zu, was für moderne Fragestellungen im Sozialbereich unerlässlich ist. Andererseits macht der geringe Abstraktionsunterschied zwischen ABM und Realsystem die Kommunikation zwischen Modellierer und Entscheidungsträger deutlich einfacher. Unglücklicherweise bringt gerade die angesprochene Flexibilität gewaltige Probleme auf technischer Seite mit sich, denn gerade bei der formalen Analyse, d.h. insbesondere bei der Validierung, von ABMs fehlt es im Unterschied zu klassischen Modellierungsmethoden an geeigneten Tools. In diesem Paper wird eine Methode präsentiert, die hier für gewisse Fragestellungen hilfreich sein kann. Das Konzept führt über die korrekte Darstellung eines ABM mithilfe sogenannter Markovketten, die, im Gegensatz zu ABMs, über mathematische Analyseverfahren verfügen. Die Methode wird anschließend durch ausgewählte Beispiele demonstriert.

## 1 Einleitung

Agentenbasierte Modelle, d.h. Modelle in denen das Gesamtmodell aus vielen Einzelmodellen, den sogenannten Agenten, besteht, kommen heutzutage in nahezu allen “Soft-Sciences” wie z.B. in der Informatik, Gesundheitsversorgung, Soziologie, Ökonomie, Epidemiologie und vielen weiteren zum Einsatz [1].

Einer der Gründe für den Erfolg für ABMs in diesen Bereichen ist deren niedriges Abstraktionsniveau: Da Fachexperten aus den angesprochenen Anwendungsdisziplinen zumeist über ähnlich wenig Know-How im Modellierungsbereich verfügen, wie Modellierer über die jeweilige Fachdisziplin, ist die Person des Modellierers klassischerweise nicht mit jener des Anwenders ident und die Kommunikation zwischen Techniker und Fachexperte von entscheidender Wichtigkeit (Das stellt einen starken Kontrast zu Simulationen im “Hard-Science” Bereichen wie Physik oder Elektrotechnik dar). Agentenbasierte Modelle bieten die Möglichkeit das vorliegende Realsystem fast

ohne jegliche Abstraktion im Modell abzubilden – “Mensch” = “Agent” oder “Tier” = “Agent”. Demnach sind sie auch für Nicht-Techniker nachvollziehbar, was natürlich die angesprochene Kommunikation erleichtert.

Des Weiteren sind ABMs eine der wenigen Modellierungstechniken die es gestattet komplexe, heterogene Systeme abzubilden, denn der Bandbreite der Eigenschaften und Verhaltensweisen der Agenten sind fast keine Grenzen gesetzt. Im Speziellen ist auch jegliche Art der Kommunikation zwischen Agenten möglich womit sich der faszinierende, chaotisch anmutende Effekt der Gruppendynamik damit besonders gut modellieren lässt.

Einen weiteren Vorteil im Vergleich zu klassischeren Modellierungsmethoden bietet die hohe Flexibilität von ABMs im Hinblick auf Modifikationen und Erweiterungen. Während bei vielen klassischen Modellierungsmethoden (vgl. Modellierung mit Differentialgleichungen) eine für den Fachexperten/Entscheidungsträger “minimal” wirkende Modifikation/Er-

weiterung der Fragestellung, z.B. die Einführung neuer Systemzustände oder Zustandsereignisse, eine völlig neuen Abstraktionsprozess und eine gewaltige Modellerweiterung zufolge haben kann, erweisen sich ABMs hierbei als (vergleichsweise) sehr flexibel.

Trotz dieser Vorteile, die agentenbasierte Modellierung eigentlich als das optimale Modellierungskonzept im Soft-Science Bereich unterstreichen, sollte der Modellierer dennoch vor einem großen Nachteil von ABM gewarnt sein: Gerade die erwähnte Flexibilität, die vorwiegend durch eine fehlende mathematisch-formale Definition der Modellierungsmethode entsteht, führt zu einem direkten Trade-Off mit der Möglichkeit das Modell formal analysieren zu können – es ist mathematisch einfach nicht mehr greifbar. Insbesondere hat das Auswirkung auf die Validierbarkeit und Verifizierbarkeit des Modells: Da das Modell nicht mathematisch beschreibbar ist, gibt es keine (kaum) Formalismen um das Modell korrekt zu parametrisieren, validieren, die Parameter auf Sensitivität zu testen, oder dessen korrekte Implementierung zu überprüfen. D.h. man kann im Allgemeinen sehr schlecht abschätzen, ob ein unerwartetes Simulationsergebnis ein bislang unbekanntes Verhalten im Realsystem beschreibt, oder auf einen Modellierungsfehler zurückzuführen ist (es sei denn man verfügt über eine so große Menge an Validierungsdaten, dass das Modell eigentlich schon wieder obsolet wäre).

In den folgenden Kapiteln wird eine Methode beschrieben, die es zumindest teilweise gestattet, einfachere ABMs mit mathematischen Mitteln zu formulieren und damit zu analysieren. Hierfür werden sogenannte Markovketten, d.h. spezielle stochastische Prozesse mit mathematisch betrachtet sehr nützlichen Eigenschaften, verwendet um gewisse Teile eines stochastischen ABMs zu beschreiben. Man erhält damit präzise Aussagen über Grenzverteilungen und kann mathematisch fundierte Prognosen über den zeitlichen Verlauf von aggregierten Größen des agentenbasierten Modells machen. Hiermit ist man nicht nur in der Lage, die korrekte Umsetzung (Implementierung) des Modells zu überprüfen, sondern hat durch die Übergangsmatrix der Markovkette einen Zugewinn an Systemwissen welcher zur Validierung verwendet werden kann.

## 2 Einführung in Agentenbasierte Modelle und Markovketten

In diesem Abschnitt präsentieren wir eine kleine Einführung in agentenbasierte Modellierung. Zudem werden einige Grundlagen zu Markovketten (MC) erklärt.

### 2.1 Agentenbasierte Modelle

Wie erwähnt bestehen ABMs aus der Komposition einer Vielzahl an einzelnen Submodellen, die auch Agenten genannt werden. Im Sinne ihrer Wortherkunft aus dem Lateinischen – “agere” für handeln – agieren diese autonom und “bewohnen” eine gemeinsame Umgebung die im Wesentlichen beschreibt wie die Agenten zueinander stehen. Jeder Agent kann für sich gewisse Verhaltensweisen an den Tag legen und diese ggf. sogar ändern. Diese können im Wesentlichen beliebig komplex (zielorientiert, selbstlernend, ...) sein. Die Umgebung kann einer natürlichen Topologie entspringen (z.B. einer geographischen Lagebeziehung) aber sie kann auch abstrakt (wie z.B. ein Kontaktnetzwerk) aufgebaut sein.

Man merkt sofort, dass obige “Definition”, die Sinngemäß [2] entspricht, keinen formalen Charakter hat und auch keinen haben kann, ohne entweder zu stark einzuschränken oder unbrauchbar allgemein zu sein. Darum müssen wir für die folgenden Abschnitte ein paar Einschränkungen und Begriffe einführen, die für irgendeine formale Erfassbarkeit eines ABMs notwendig sind:

Zunächst wollen wir uns in dieser Arbeit auf zeitgetaktete, stochastische ABMs fokussieren, was zwar im Sinne der allgemeinen Beschreibung einschränkt, aber den Großteil der existierenden ABMs abdeckt. D.h. das Modell wird in festen Zeitschritten mit einer globalen Aktualisierungsregel, die sich aus den Verhaltensweisen aller Agenten zusammensetzt, aktualisiert die zudem zufällige Elemente beinhaltet. Als “Zustand” wollen wir ab sofort die Zusammenfassung aller Eigenschaften bezeichnen, die ein Agent zu einem gewissen Zeitpunkt besitzt. Die Aktualisierungsregel bildet somit in jedem Zeitschritt die Gesamtheit aller einzelnen Agentenzustände auf neue ab.

## 2.2 Markovketten

Trotz obigen Versuchen die Definition von ABMs ein wenig formaler zu machen, ist das im Vergleich mit der rein mathematischen Definition von Markovketten (englisch Markov-Chain, abgekürzt MCs) noch richtiggehend “schwammig”. Eine MC ist ein zeitdiskreter stochastischer Prozess auf einem Zustandsraum  $S$ , der die Eigenschaft der Gedächtnislosigkeit besitzt. D.h. der Übergang zu seinem nächsten Zustand wird ausschließlich vom aktuellen Zustand beeinflusst aber keinem früheren. Da der Prozess zeitdiskret ist, kann er als Folge  $\{X_n, n = 0, 1, 2, \dots\}$  angesehen werden, wobei alle  $X_i \in S$ . Durch die Gedächtnislosigkeit ist der Prozess schlussendlich über seine sogenannten Übergangswahrscheinlichkeiten  $p_{i,j} := P(X_{n+1} = j | X_n = i)$ ,  $i, j \in S$  eindeutig definiert. Ist  $S \cong \{1, \dots, m\}$  endlich, lassen sich diese in einer sogenannten Übergangsmatrix  $P$  zusammenfassen.

Wir wollen nun ein wenig auf die Analyse von Markovketten eingehen. Es ist nicht verwunderlich, dass im Wesentlichen alle Eigenschaften des Prozesses aus dessen Übergangsmatrix ablesbar sind. Bezeichne  $\mu^{(t)} \in \mathbb{R}^m$  den Vektor der Wahrscheinlichkeiten, dass der Prozess zum Zeitpunkt  $t$  einen der  $m$  Zustände aus  $S$  annimmt, so gilt  $\mu^{(n)} = \mu^{(0)} P^n$ . Unter gewissen Bedingungen an die Matrix  $P$  gibt es folglich eine sogenannte Grenzverteilung des Prozesses

$$\mu^* := \mu^{(0)} \lim_{n \rightarrow \infty} P^n. \quad (1)$$

Erfüllt  $P$  erwähnte Bedingung, lässt sich die Grenzverteilung als Lösung eines linearen Gleichungssystems finden (genauer in [3]).

Dieses ist nur ein Beispiel für die Unzahl an Methoden, die es gestattet MCs mithilfe einfachere analytischer Operationen zu untersuchen. Eine Darstellung eines ABMs durch eine MC hat also viele Vorteile.

## 3 Agentenbasierte Modelle als Markovketten

In diesem Kapitel wollen wir erklären wie man unter gewissen Umständen ABMs mithilfe von Markovketten formalisieren kann. Wir folgen in diesem Kapitel [4, 5], und beschreiben das Vorgehen in schrittweise.

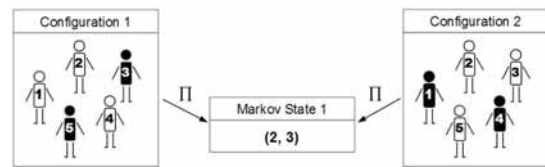
### 3.1 Schritt 1: Identifikation des Zustandsraumes

Im ersten Schritt muss das gesamte ABM im Hinblick auf unterschiedliche Zustände analysiert werden. Hierbei verwenden wir den Begriff der “Konfiguration”, der die Zusammensetzung aller individuellen Zustände aller Agenten eines ABMs beschreibt. Es sei  $\sigma$  jener individuelle Zustandsraum der alle möglichen Zustände aller Agenten beinhaltet, und  $N$ , wie auch in Folge, die Gesamtanzahl aller Agenten, so beinhaltet  $\sigma^N$  alle möglichen Konfigurationen des ABM.

Die nächste Aufgabe besteht nun darin, eine Abbildung  $\Pi$  zu finden, die den Raum  $\sigma^N$  auf den Zustandsraum  $S = \{1, \dots, m\}$  einer Markovkette abbildet. Hierfür gibt es im Allgemeinen keine eindeutige Lösung und je “injektiver” die Abbildung  $\Pi$  gewählt wird, umso unwahrscheinlicher ist es, dass die damit abgebildeten Zustände die Markov-Eigenschaft erfüllen, wenn das ABM aktualisiert wird. Existenz ist jedenfalls gewährleistet, denn im schlimmsten Fall, erhält man mit  $\Pi$  konstant einen gedächtnislosen Prozess. Zumeist findet sich über die klassische Aggregationsabbildung

$$\Pi: \sigma^N \rightarrow (\mathbb{N} \cup \{0\})^{|\sigma|}: \{a_1, \dots, a_N\} \mapsto \left( \sum_{i=1}^N \mathbb{1}_x(a_i) \right)_{x \in \sigma}, \quad (2)$$

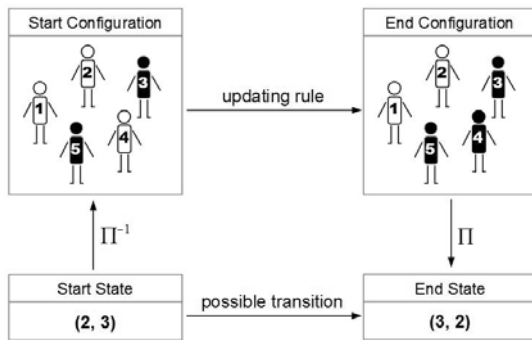
mit der Indikatorfunktion  $\mathbb{1}$ , eine Abbildung die auf gedächtnislose Zustände führt. Diese kann man sich über “Zählen aller Agenten im selben Zustand” erklären. Ein Beispiel dafür, wie diese Funktion arbeitet, findet sich in Abbildung 1. Die über  $\Pi$  abgebildeten Konfigurationen des ABM wollen wir fortan als Markov-Zustände bezeichnen.



**Abbildung 1:** Zwei Konfigurationen die auf den selben Markov-Zustand abbilden. Die erste Zahl des Markov-Zustandes bezeichnet die Anzahl der Agenten im Zustand “schwarz” während die zweite die Anzahl jener im “weißen” Zustand beschreibt. Der Zustandsraum der Markov-Zustände wäre somit durch  $S = \{(N_b, N_w) | N_b + N_w = 5, N_b, N_w \geq 0\}$  gegeben, wobei  $N_b$  die Anzahl der “schwarzen” Agenten beschreibt und  $N_w$  jene der “weißen”.

### 3.2 Schritt 2: Berechnung der Übergangsmatrix

Hat man eine geeignete Abbildung  $\Pi$  gefunden, geht es im zweiten Schritt an die Berechnung der Übergangsmatrix der Markovkette. Hierfür müssen zunächst alle möglichen Übergänge eines Markov-Zustandes in einen anderen bestimmt werden. Wir wollen diesen Vorgang "Erreichbarkeitsanalyse" nennen. Genauer muss hierfür jede Konfiguration die einen Markov-Zustand  $A \in S$  hervorruft darauf analysiert werden, ob die Aktualisierungsregel (Updating Rule) des ABM einen Übergang (Transition) in eine andere Konfiguration zulässt, die einen Markov-Zustand  $B \in S$  bewirkt (siehe Abbildung 2). In vielen



**Abbildung 2:** Schematische Beschreibung der Analyse der Markov-Übergänge. Die Umkehrfunktion  $\Pi^{-1}$  sei als Urbildabbildung zu betrachten.

Fällen kann man diese Analyse auf Basis der Modellbeschreibung durchführen. Kann man das nicht, muss ein Zeitschritt mit dem agentenbasierten Modell (öfters) simuliert werden um alle erreichbare Konfigurationen zu finden. Grundsätzlich ist es nicht trivial welche(s) Element aus  $\Pi^{-1}(A)$  man hierbei als Startkonfiguration für das ABM verwendet um die Erreichbarkeit zu analysieren. Diese Fragestellung würde jedoch den Rahmen dieses Papers sprengen. Zudem wird sich das in den Fallbeispielen als irrelevant herausstellen.

Sind alle möglichen Übergänge bekannt, d.h. kennt man, welche Einträge der Übergangsmatrix  $P$  einen Wert ungleich 0 einnehmen, verbleibt es diese zu quantifizieren. Hierfür geht man ähnlich vor, wie bei der Erreichbarkeitsanalyse: Entweder die Modellbeschreibung lässt es zu die Werte mit stochastischen Mitteln direkt zu berechnen, oder man führt einen Zeitschritt des agentenbasierten Modells so oft

aus, bis man mit statistischen Methoden quantitative Werte bestimmen kann.

## 4 Resultate aus Fallstudien

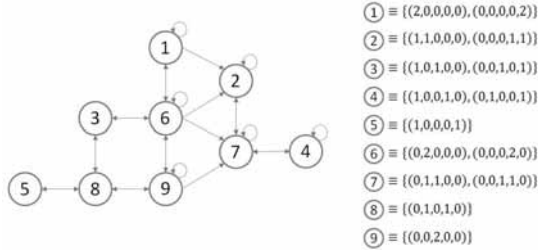
Schlussendlich wollen wir einige ABMs mit den, im vergangenen Kapitel beschrieben, Methoden hergeleiteten MC vergleichen. In allen Fallbeispielen wird der Vergleich auf Basis einer Monte-Carlo Analyse des ABM durchgeführt. Im Sinne der Vergleichbarkeit wird stets eine feste Startverteilung für die Anfangskonfiguration des ABM verwendet.

### 4.1 1D Random Walk

Als erstes Fallbeispiel wird ein agentenbasiertes Modell analysiert, in welchem die Agenten einen eindimensionalen Zufallspfad (Random Walk) auf einem diskretem endlichem Gitter durchführen. Sofern der Agent nicht am Rand des Gitters steht kann er pro Zeitschritt stets einen Gitterpunkt nach links oder rechts rücken. Steht er am Rand entsprechen "links" und "rechts" jeweils der einzig möglichen Richtung. Steht er alleine auf seinem Gitterpunkt, rückt er mit Wahrscheinlichkeit 0.5 nach links und eben selber nach rechts. Teilt er seine aktuelle Position mit mindestens einem anderen Agenten, gibt es zusätzlich eine Wahrscheinlichkeit von 0.6, dass er auf seinen Gitterpunkt verbleibt. Für die Vergleichsresultate betrachten wir dieses Modell mit 2 Agenten und ein Gitter mit 5 Gitterpunkten  $\{1, 2, 3, 4, 5\}$ . Als Zustand eines Agenten ergibt sich seine Position am Gitter, d.h. eine natürliche Zahl zwischen 1 und 5.

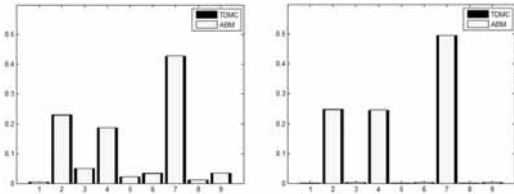
Zunächst definieren wir einen Zustandsraum der Markovkette und damit eine Projektionsabbildung  $\Pi$ . Da das ABM selbst schon gedächtnislos arbeitet, stellt die Markoveigenschaft keinerlei Einschränkung für  $\Pi$  bzw.  $S$  dar. Man könnte z.B. auch  $\Pi = Id$  wählen, womit  $S = \{1, \dots, 5\}^2$  insgesamt 25 Elemente besäße. Im Sinne der Praktikabilität (man denke z.B. daran die Anzahl der Agenten zu erhöhen) macht es aber Sinn den Raum weiter einzuschränken. Die klassische Aggregationsabbildung (2) führt auf 15 Zustände – jeweils 5-elementige Vektoren mit Einträgen zwischen 0 und 2 je nach Anzahl der Agenten auf dem jeweiligen Gitterpunkt. Unter Ausnützung von Symmetrien lässt sich dieser Raum noch zusätzlich auf schlussendlich 9 Elemente einschränken lassen. Abbildung 3 zeigt das sogenannte Übergangsdiagramm der Markovkette in

dem auch ersichtlich ist, wie die Symmetrie hier zur Einschränkung verwendet wurde.



**Abbildung 3:** Übergangsdiagramm zu 1D Random Walk Modell mit 2 Agenten und 5 Gitterpunkten. Rechter Hand sieht man, welche aggregierte Zustände über Symmetrien zusammengefasst wurden.

Aus der Abbildung ersichtlich ergibt sich ein mit  $C := \{\textcircled{2}, \textcircled{4}, \textcircled{7}\} \subset S$  eine absorbierender Zyklus den man als geschlossene kommunizierende Klasse (Closed Communicating Class) bezeichnet. Sobald die Markovkette einen dieser Zustände einnimmt, wird sie die Klasse nicht mehr verlassen. Da  $C$  mit positiver Wahrscheinlichkeit erreichbar ist, kann eine Grenzverteilung ( $t \rightarrow \infty$ ) nur für Elemente aus  $C$  positive Werte annehmen. In Abbildung 4 ist leicht zu



**Abbildung 4:** Vergleich der Verteilung des ABM (hell) mit jener der MC (dunkel) nach 10 (links) bzw. 20 Zeitschritten. Das ABM startet hier stets mit der Ausgangskonfiguration (1,5) was  $\Pi((1,5)) = \textcircled{5}$  bedeutet. Um die Verteilung des ABM statistisch zu bestimmen wurden 100000 Durchläufe simuliert.

erkennen, dass scheinbar schon nach 20 Schritten die Grenzverteilung erreicht wird. Die Werte der Übergangsmatrix, die hierfür verwendet wurden, lassen sich ähnlich bestimmen wie das Übergangsdiagramm, nämlich direkt aus der Modellbeschreibung.

Die exakte Grenzverteilung der Markovkette  $X_t$  lässt sich, wie erwähnt, mithilfe eine linearen Gleichungssystems berechnen. Man erhält  $\lim_{t \rightarrow \infty} P(X_t = \textcircled{2}) = 0.25$ ,  $\lim_{t \rightarrow \infty} P(X_t = \textcircled{5}) = 0.25$  und  $\lim_{t \rightarrow \infty} P(X_t = \textcircled{7}) = 0.5$ . Zudem lässt sich auch der

Erwartungswert für die Anzahl der Schritte bis zum Ersteintritt in  $C$  mithilfe der Übergangsmatrix berechnen. Man erhält

$$\mathbb{E}(\min_t (X_t \in C, \forall k < t : X_k \notin C)) \approx 7.08, \quad (3)$$

wenn im Zustand  $\textcircled{5}$  gestartet wurde.

## 4.2 Entscheidungsmodell

Das zweite ABM, welches wir zwar hier als Fallbeispiel anführen aber nicht mehr so rigoros behandeln wollen, ist ein Entscheidungsmodell (Opinion Dynamics Modell). Hierbei haben Agenten stets eine Meinung zu einem gewissen Thema. In unserem einfachen Fall wird diese mit einer boolesche Variable modelliert. Diese Meinung kann sich durch Kommunikation mit einem andersdenkenden Agenten ändern, wenn dieser ‘‘Gesprächspartner’’ überzeugend war. Dieser Vorgang wird mithilfe von Wahrscheinlichkeiten modelliert.

Wir analysieren hierbei unterschiedliche Netzwerktypen, die für die Auswahl der Gesprächspartner verantwortlich sind und betrachten jeweils 20 Agenten. Wir verwenden die normierte Aggregationsabbildung (d.h. (2) dividiert durch die Agentenzahl) was auf eine MC mit einem zweidimensionalen Zustand führt.

Nach dem Bestimmen der Übergangsmatrix erhält man ein MC deren Ergebnisse die Verteilungen des ABM nahezu perfekt ( $\|error\|_2 \leq 0.0028$  für die Test-szenarien) widerspiegelt, sondern zusätzlich noch Aussagen über die Auswirkungen von Netzwerkkonstanten zulässt.

## 5 Conclusio

Die beiden angeführten Testfälle in Kapitel 4 unterstreichen, dass die Methode aus Kapitel 3 nicht nur funktioniert, sondern auf welche Art und Weise die erarbeitete MC analysiert werden kann um Rückschlüsse auf das ABM treffen zu können. Klarerweise wird das Verfahren sehr umständlich, sobald das Modell ‘‘groß’’(hohe Agentenzahlen/Zustandsdimensionen) wird. Hier ist weitere Forschungsarbeit notwendig.

## Acknowledgments

K-Projekt DEXHELPP is supported by BMVIT, BMWFW and the state of Vienna via COMET ? Competence Centers for Excellent Technologies. Programme COMET is processed by FFG

## References

- [1] R. Axelrod. *Chapter 33 Agent-Based Modeling as a Bridge Between Disciplines*. In: Handbook of Computational Economics, Elsevier, pp. 1565-1584, 2006.
- [2] C. Macal, M., North. *Tutorial on Agent-Based Modeling and Simulation Part 2: How to Model with Agents*. In: Proceedings of the 2006 Winter Simulation Conference, Monterey, California, pp. 73-83, 2006.
- [3] R. Serfozo. *Basics of Applied Stochastic Processes*. Berlin, Springer, 2009.
- [4] S. Banish, R. Lima, R. Araujo. *Agent Based Models and Opinion Dynamics as Markov Chains*. In: Social Networks, 34(4), pp. 549-561, 2012.
- [5] J. Izqueierdo, S. Segismundo, S. Izquierdo, J. Galan, I. Santos. *Techniques to Understand Computer Simulations: Markov Chain Analysis*. In: Journal of Artificial Societies and Social Simulation, 12(1), p. 6, 2009.

# Softwarebasierte Analyse von Simulationsdaten durch automatisierte Berechnung von Kennzahlen

Jakob Krieg<sup>1</sup>; Wolfgang Schlüter<sup>1</sup>

<sup>1</sup>Hochschule Ansbach

[jakob.krieg@hs-ansbach.de](mailto:jakob.krieg@hs-ansbach.de)

Beim Postprocessing von Simulationen werden gewöhnlicherweise die bereits vorhandenen Tools der Simulationsumgebung eingesetzt, um Plots zu erstellen oder Kennzahlen zu berechnen. Diese Vorgehensweise reicht für die Simulation von einzelnen Komponenten zwar aus, ist aber für die Auswertung von umfangreichen Fertigungssimulationen über einen großen Zeitraum hinweg zunehmend aufwändig, unübersichtlich und funktional nicht ausreichend.

Aufgrund dieser Problematik wird für das Projekt *Smart Melting*, in dem ein Schmelz- und Druckgussbetrieb simuliert wird, die Auswertesoftware *Melting Analytics* entwickelt. Im ersten Schritt wird dafür eine Datenbankanbindung der Simulation implementiert, die alle Simulationsdaten in einer Datenbank auf einem Server sammelt. Die in Matlab entwickelte Auswertesoftware *Melting Analytics* verarbeitet diese Simulationsdaten, um daraus mithilfe einer graphischen Benutzeroberfläche Kennzahlen flexibel zu berechnen, zu visualisieren und zu vergleichen. Die Berechnung der Kennzahlen erstreckt sich auf absolute, relative und abgeleitete Kennzahlen, die wiederum Einzelwerte, zeitliche Summen und zeitliche Mittelwerte darstellen können.

Durch das Auswertetool wird das Postprocessing der Fertigungssimulation optimiert, indem das Schreiben oder Ändern von unübersichtlichen Auswerteskripten durch die intuitive Bedienung einer graphischen Benutzeroberfläche ersetzt wird. Die Simulation kann dadurch wesentlich effizienter ausgewertet werden. Das Client-Server Modell bietet zudem die Möglichkeit, dass verschiedene Projektmitarbeiter Simulationen auswerten können, ohne dass dabei die Simulationsdaten lokal verfügbar sind. Durch die plattformunabhängige Speicherung der Prozessdaten kann die Analysesoftware ohne große Anpassungsmaßnahmen für andere Simulationsumgebungen oder den realen Betrieb genutzt werden, falls die Fertigungssimulation eine virtuelle Inbetriebnahme darstellt.

## 1 Einleitung

Im Forschungsverbund *Green Factory Bavaria* wird im Teilprojekt *Smart Melting* ein Prozessmanagementsystem (PMS) entwickelt, um die Energieeffizienz in der Schmelz- und Druckgussindustrie zu steigern. Dabei wird ein gasbetriebener Schmelzbetrieb, der Aluminium als Betriebsmedium nutzt, betrachtet. Der reale Betrieb wird dabei durch eine Material- und Energieflusssimulation in *Matlab* ersetzt, um das PMS zu entwickeln. Die Fertigungssimulation wird über einen Zeitraum von einer Woche durchgeführt, wobei eine Milliarde an Prozessdaten anfallen, die für die Auswertung der Simulation relevant sind. Abbildung 1 stellt die verschiedenen Maschinentypen eines Schmelz- und Druckgussbetriebes dar und verdeutlicht dessen Produktionsablauf. In dieser Arbeit sind insbesondere die Maschinentypen „Schmelzofen“ und „Druckgussanlage“ relevant, die für das Schmelzen beziehungsweise Gießen des Aluminiums sorgen.

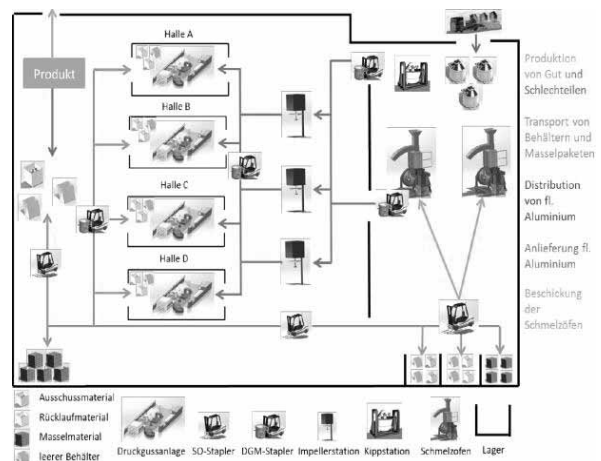


Abbildung 1: Schmelz- und Druckgussbetriebe [1]

Die Auswertung der Material- und Energieflusssimulation besitzt eine hohe Relevanz für die Entwicklung des PMS, da zur Optimierung des PMS verschiedene Steuerungsmechanismen untersucht werden müssen.

Zudem können durch verschiedene Simulationskonfigurationen Energieeinsparpotentiale für die Schmelz- und Druckgussindustrie abgeschätzt werden. Bei einem jährlichen Verbrauch von  $1,1 \cdot 10^6$  t Nicht-Eisen-Werkstoffen ergibt sich in der Annahme von 2000 kWh pro Tonne guter Guss ein Energieverbrauch von 2,2 Mrd. kWh jährlich [2]. In der Schmelz- und Druckgussindustrie besitzt die Produktionssicherheit höchste Priorität. Dementsprechend werden die Kennzahlen hauptsächlich hinsichtlich Energieverbrauch und Produktionssicherheit betrachtet. Die Berechnung der Kennzahlen resultiert aus einer großen Menge von unterschiedlichen Prozessdaten und erstreckt sich auf absolute, relative und abgeleitete Kennzahlen. Diese können wiederum als Einzelwerte sowie Summen oder Mittelwerte über eine bestimmte Zeit dargestellt werden. Die verschiedenen Kennzahlentypen sind in Abbildung 2 dargestellt.

Prozessdaten:  $t \rightarrow y$

↓

	Absolute Kennzahlen:	Relative Kennzahlen:	Abgeleitete Kennzahlen:
Einzelwerte:	$f(t) = y(t)$	$f(t) = \frac{y(t)}{k}$	$f(t) = \frac{y(t) * z(t)}{k}$
Summiert:	$f(t_1, t_2) = \sum_{t=t_1}^{t_2} y(t)$	$f(t_1, t_2) = \sum_{t=t_1}^{t_2} \frac{y(t)}{k}$	$f(t_1, t_2) = \sum_{t=t_1}^{t_2} \frac{y(t) * z(t)}{k}$
Gemittelt:	$f(t_1, t_2) = \frac{\sum_{t=t_1}^{t_2} y(t)}{t_2 - t_1}$	$f(t_1, t_2) = \frac{\sum_{t=t_1}^{t_2} \frac{y(t)}{k}}{t_2 - t_1}$	$f(t_1, t_2) = \frac{\sum_{t=t_1}^{t_2} \frac{y(t) * z(t)}{k}}{t_2 - t_1}$

Abbildung 2: Kennzahlentypen

In der Fertigungssimulation des Schmelz- und Druckgussbetriebes im Projekt *Smart Melting* resultieren aus den verschiedenen Kennzahlentypen mehr als 80 verschiedene Kennzahlen. Diese Kennzahlen sollen wiederum für einzelne Maschinen, Maschinengruppen oder Zeiträume bestimmt werden. Die deskriptive Datenanalyse, die die Gewinnung von Informationen durch Grafiken, Tabellen und Kennzahlen aus einer Grundgesamtheit beinhaltet, gestaltet sich also dementsprechend umfangreich [3].

## 2 Manuelles Postprocessing

Die Auswertung der Fertigungssimulation wird im manuellen Postprocessing in *Matlab* durch das Schreiben von Befehlsabfolgen ohne Kopplung zu einer Datenbank vorgenommen. Da Simulationen auf verschiedenen Rechnern durchgeführt und ausgewertet werden, müssen die Auswerteskripte und Simulationsdaten unter einem kontinuierlichen Austausch zwischen den Projektmitarbeitern stehen. Wird die Simulation weiterentwickelt, müssen die Auswerteskripte für jede Kennzahl angepasst werden.

Für den häufigen Fall, dass Kennzahlen für einzelne Simulationen, Maschinen, Maschinengruppen oder Zeiträume berechnet werden sollen, müssen wiederum Auswerteskripte angepasst werden. Daraus resultiert beim Postprocessing ein erhöhter Zeitaufwand. Aufgrund des benötigten Wissens über *Matlab* und das Kennzahlensystem entsteht zudem eine große Fehleranfälligkeit. Die Auswerteskripte beschränken sich beim manuellen Postprocessing in *Matlab* auf das Berechnen und Plotten von Kennzahlen. Weitere Aufgaben müssen mit den graphischen Standardfunktionen in *Matlab* gelöst werden. Das Auswerten der Simulation kann also insgesamt betrachtet nicht effizient durchgeführt werden. Eine automatisierte Berechnung und Visualisierung der Kennzahlen wird benötigt.

## 3 Automatisiertes Postprocessing

Ziel ist es, mithilfe einer Auswertesoftware Kennzahlen automatisiert zu berechnen und mithilfe einer grafischen Benutzeroberfläche schnell, flexibel und methodenreich zu visualisieren. Dafür wird eine Auswertesoftware benötigt, die speziell für das Projekt entwickelt wird und als Datenbasis eine Datenbank besitzt. Die Auswertesoftware, genannt *Melting Analytics*, muss ausnahmslos alle benötigten Funktionalitäten zur Verfügung stellen, die im vorliegenden Projekt benötigt werden und dabei dennoch intuitiv bedienbar sein. Dabei werden vom Benutzer keine Programmierkenntnisse oder Kenntnisse über die genaue Berechnung der Kennzahlen abverlangt. Um die Analysesoftware unabhängig von der Simulationsumgebung zu entwickeln, sollen die Simulationsdaten in einer Datenbank gespeichert werden. Dadurch kann die Software *Melting Analytics* ohne große Anpassung auch für reale Schmelz- und Druckgussbetriebe genutzt werden, die die Maschinendaten in Datenbanken speichern. Indem die Datenbank auf einem Server angelegt wird, können verschiedene Projektmitarbeiter Simulationen parallel auswerten. Die Speicherung und Auswertung der Simulationsdaten erfolgt übersichtlich, selbst wenn verschiedene Projektmitarbeiter an der Durchführung und Analyse der Simulationen beteiligt sind.

## 4 Umsetzung der Auswertesoftware

Die Software wird gemäß dem Vorgehensmodell *Unified Process (UP)* entwickelt, dessen Workflows die Disziplinen Anforderungsanalyse, Entwurf, Realisierung, Testen und Auslieferung vorsehen [4].

Nachfolgend wird näher auf den Entwurfsschritt eingegangen, da dies auch auf andere Projekte mit derselben Problemstellung anwendbar ist, während die Ausführung der anderen Disziplinen stark vom einzelnen Projekt abhängt.

#### 4.1 Datenbank

Im ersten Schritt muss ein Datenbankmanagementsystem gewählt werden. Bei *Melting Analytics* handelt es sich trotz der großen Datenmengen nicht um eine *Big Data* Anwendung, da diese sich durch unstrukturierte Datenformate, hunderte Terabytes, maschinelles Lernen und datenbasierte Produkte kennzeichnen [5]. Da keine *Big Data* Anwendung vorliegt, eignet sich eine relationale Datenbank [6]. Das Open-Source Datenbankmanagementsystem *MySQL* erfüllt die benötigten funktionalen und sicherheitsrelevanten Anforderungen, zum Beispiel eine Serveranbindung und eine Zugriffskontrolle.

Die nachfolgend aufgelisteten Relationen reichen aus, um die Simulationsdaten für die Auswertung ausreichend zu strukturieren.

Legende:

**Relation** (Primärschlüssel, *Fremdschlüssel*, Feld)

- **Simulation** (ID-Simulation, Zeit, Dauer, Beschreibung)

Neben der Entität „Simulation“ wird für jeden Maschinentyp nach Abbildung 1 eine Tabelle der Konfigurationsdaten sowie eine Tabelle mit den Prozessdaten angelegt. Dies wird nachfolgend am Beispiel des Schmelzofens durchgeführt

- **Konfigurationsdaten\_Schmelzofen**(ID-Maschine, *ID-Simulation*, Kapazität, Füllstand, Verbrauch, ...) *ID-Simulation* referenziert **Simulation**
- **Prozessdaten\_Schmelzofen**(ID, *ID\_Simulation*, *ID\_Maschine*, Zeitpunkt, Füllstand, Temperatur, ...) *ID-Simulation* referenziert **Simulation**, *ID\_Maschine* referenziert **Konfigurationsdaten\_Schmelzofen**

#### 4.2 Datenbankanbindung

Für die Datenbankanbindung der Fertigungssimulation kann in *Matlab* die *Database Toolbox* verwendet werden, welche als Kommunikationsschnittstelle *ODBC* (Open Database Connectivity) benutzt, die von allen gängigen Datenbankmanagementsystemen unterstützt wird [7][8].

Bei der Speicherung von Simulationsdaten in einer Datenbank sollte vor jedem Beginn der Simulation die Datenbankstruktur umfassend geprüft werden, um die Integrität der Simulationsdaten sicherzustellen und Programmfehler zu vermeiden. Dies wird gelöst, indem eine Konfigurationsdatei die Datenbankstruktur vorgibt, die vor Beginn der Simulation mit der vorhandenen Datenbankstruktur abgeglichen wird. Die Konfigurationsdatei wird gleichermaßen für die Auswertung der Simulation mit *Melting Analytics* verwendet. Da sich die Struktur der Prozessdaten in der Entwicklung einer Fertigungssimulation häufig ändert, ist es wichtig, die Struktur der Datenbank möglichst frei und unkompliziert ändern zu können.

Wählt der Benutzer beim Starten der Simulation die Datenbankanbindung und wird die Datenbankstruktur erfolgreich überprüft, werden zu Beginn der Simulation die Konfigurationsparameter und die Beschreibung der Simulation in der Datenbank gespeichert. Anschließend werden in jedem Zeitschritt der Simulation die aktuellen Prozessdaten in die Datenbank geschrieben.

#### 4.3 Auswertesoftware Melting Analytics

Der Benutzer interagiert über eine objektorientiert aufgebaute grafische Benutzeroberfläche (GUI) durch Bedienelemente mit einer Klasse, die alle vom Benutzer angeforderten Daten aus der Datenbank lädt, entsprechende Kennzahlen berechnet und auf den Anzeigeelementen der GUI darstellt. Die *SQL*-Abfragen in *Matlab* sorgen dafür, dass ausschließlich vom Benutzer angeforderte Daten geladen werden, um größtmögliche Performance zu gewährleisten.

Für jeden Maschinentyp nach Abbildung 1 wird aus der Vorlage ein Fenster (Tab) erstellt. Zusätzlich werden Fenster für den Vergleich von Maschinentypen sowie die Auswertung von Maschinengruppen und des Gesamtbetriebes erstellt. Durch den identischen Aufbau der Fenster müssen jegliche Funktionen der GUI nur einmalig für die Vorlage implementiert werden. Die GUI von *Melting Analytics* ist in Anhang 1 ersichtlich. Jeder Tab besitzt drei voneinander abgegrenzte Bereiche: Die Darstellung der Onlinekennzahlen durch Plots im Bereich *Onlinekennzahlen*, die Darstellung der zeitlich gemittelten und summierten Kennzahlen durch Tabellen im Bereich *Zeitlich summierte/gemittelte Kennzahlen* und auf der rechten Seite die Möglichkeit durch Buttons alle nötigen Benutzereingaben zu treffen und Funktionen von *Melting Analytics* zu nutzen.

Die Struktur der Auswertesoftware *Melting Analytics* wird in Abbildung 3 dargestellt.

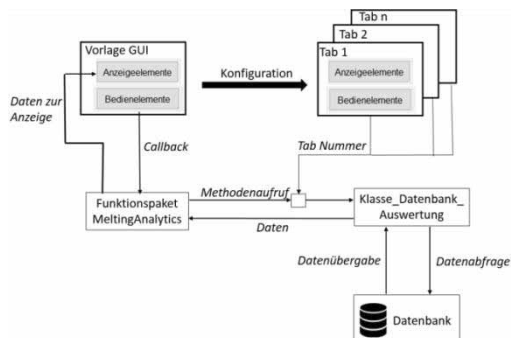


Abbildung 3: Struktur *Melting Analytics*

Zusammenfassend zeigt das Komponentendiagramm in Abbildung 4 den Entwurf des Softwarepaketes, das die Nutzung einer automatisierten Auswertung einer Fertigungssimulation ermöglicht.

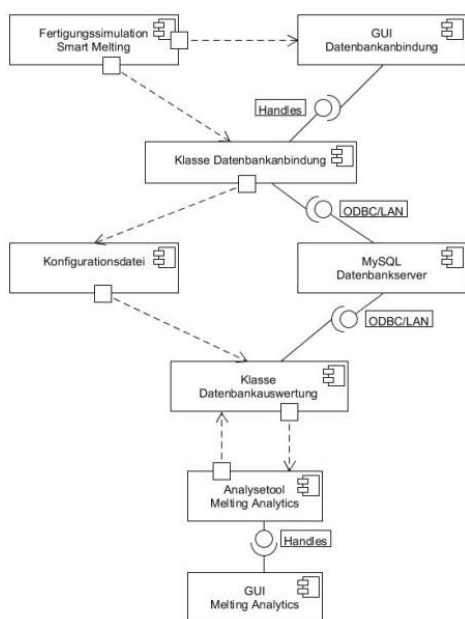


Abbildung 4: Komponentendiagramm Analysetool

## 5 Gegenüberstellung

Nachfolgend wird der Vergleich des manuellen zum automatisierten Postprocessing am Beispiel der Kennzahlen Gesamtanlageneffektivität (OEE) und Ausbringung der Druckgussmaschinen durchgeführt.

### 5.1 Gesamtanlageneffektivität (OEE)

Die OEE bestimmt den zeitlichen Wirkungsgrad von Maschinen, wobei Qualitätsverluste, geplante und ungeplante Störungen, Rüstungsaufwände sowie Geschwindigkeitsverluste berücksichtigt werden.

Die Kennzahl OEE wird wie in Abbildung 5 dargestellt berechnet. Es handelt sich um eine abgeleitete Kennzahl, die das Produkt von drei anderen Kennzahlen ist.

Betriebszeit			
Verfügbarkeit	A	Planbelegungszeit	
	B	Hauptnutzungszeit	Verfügbarkeitsverluste: - Störungen - Warten/Umrüsten - Linienbeschränkungen
Effektivität	C	Mögliche Ausbringung	
	D	Tatsächliche Ausbringung	Leistungsverluste: - Kurzstillstände - reduzierte Geschwindigkeit
Qualitätsrate	E	Tatsächliche Ausbringung	
	F	Einwandfreie Produkte	Qualitätsverluste: - Ausschuss - Nacharbeit
$OEE = \text{Verfügbarkeit} \cdot \text{Effektivität} \cdot \text{Qualitätsrate} = B/A \cdot D/C \cdot F/E$			

Abbildung 5: Bestimmung OEE in Anlehnung an [9]

Anhand der OEE lässt sich feststellen, welcher prozentuale Zeitanteil dafür genutzt wird, tatsächlich Teile zu produzieren. Dementsprechend besitzt die Kennzahl bei der Auswertung von Fertigungssimulationen eine hohe Relevanz. Die OEE kann ausschließlich als zeitlich gemittelter Wert über einen beliebigen Zeitraum berechnet werden.

#### 5.1.1 Manuelle Kennzahlenanalyse

Um die OEE manuell zu bestimmen, wird ein Skript benötigt, das aufgrund der Komplexität der Kennzahl über 100 LOC (Lines of Code) besitzt. Ergebnis ist ein Wert, der über den gesamten Simulationszeitraum für alle Maschinen berechnet wurde. Soll die OEE für bestimmte Maschinen und Zeiträume bestimmt werden, muss an mehreren Stellen in den Quellcode eingegriffen werden. Voraussetzung für die fehlerfreie Auswertung ist eine genaue Kenntnis über das Auswerteskript und die Berechnung der Kennzahl.

#### 5.1.2 Automatisierte Kennzahlenanalyse

Nachdem die Software *Melting Analytics* gestartet wird, finden folgende Auswahlen statt:

1. Datenbank
2. Simulation
3. Zeitraum / Zeiträume
4. Maschine/n

Für die getroffene Auswahl kann nun jede verfügbare Kennzahl berechnet werden.

Die OEE kann folglich in weniger als einer Minute für verschiedene Simulationen, Maschinen, Maschinengruppen und Zeiträume verglichen werden.

## 5.2 Ausbringung Druckgussmaschine

Die Ausbringung der Druckgussmaschine stellt eine absolute Kennzahl dar, die als Einzelwert und über einen Zeitraum  $[t_1, t_2]$  gemittelten oder summierten Wert berechnet werden kann, wie in Formel (1) - (3) dargestellt ist. Die Kennzahl beschreibt die Produktion der Druckgussmaschinen in Form der produzierten Masse und unterliegt dem Prinzip je größer desto besser.

- Einzelwert:

$$\text{Ausbringung}(t) = \text{Nettoschussgewicht}(t) \quad (1)$$

- Summiert:

$$\text{Ausbringung}(t_1, t_2) = \sum_{t=t_1}^{t_2} \text{Nettoschussgewicht}(t) \quad (2)$$

- Gemittelt:

$$\text{Ausbringung}(t_1, t_2) = \frac{\sum_{t=t_1}^{t_2} \text{Nettoschussgewicht}(t)}{t_2 - t_1} \quad (3)$$

Die Ausbringung wird dafür verwendet, die Produktivität von verschiedenen Druckgussmaschinen oder Gruppen von Druckgussmaschinen über verschiedene Zeiträume bei unterschiedlichen Simulationen zu vergleichen.

### 5.2.1 Manuelle Kennzahlenanalyse

Die Kennzahl kann für einzelne Maschinen mit wenig Aufwand geplottet, summiert oder gemittelt werden. Der Vergleich zwischen verschiedenen Simulationen, Zeiträumen und Maschinengruppen ist jedoch mit zusätzlichem Aufwand verbunden.

### 5.2.2 Automatisierte Kennzahlenanalyse

Nachdem die Auswahlen wie in Kapitel 5.1.2 getroffen wurden, wird für verschiedene Gruppen, Zeiträume und Simulationen die Ausbringung automatisch geplottet und der zeitliche Mittelwert und die zeitliche Summe tabellarisch angeordnet. Anschließend kann auf weitere von der Auswertesoftware zur Verfügung gestellte Funktionen zurückgegriffen werden und Plots oder Tabellen interaktiv gestaltet werden.

## 6 Bewertung

Nachfolgend wird bewertet, welchen Nutzen und welche Schwachstellen die softwarebasierte Analyse von Simulationsdaten durch die in dieser Arbeit beschriebene automatisierte Berechnung von Kennzahlen mit sich bringt.

### 6.1 Nutzen

- Kennzahlen können zeiteffizient berechnet, visualisiert und verglichen werden.
- Die Simulationsauswertung erfordert keine Programmierkenntnisse.
- Kenntnisse über die genaue Definition der Kennzahlen sind nicht erforderlich.
- Berechnungsfehler im Postprocessing werden vermieden.
- Vorgefertigte Funktionen zur Simulationsauswertung können verwendet werden.
- Parallele und übersichtliche Auswertung von Simulationen durch verschiedene Projektmitarbeiter.
- Auswertesoftware kann für andere Simulationsumgebungen oder reale Betriebe ohne große Anpassungen übernommen werden.

### 6.2 Nachteile

- Die Entwicklung einer projektspezifischen Auswertesoftware benötigt einen großen Zeitaufwand und ist dementsprechend nur für umfangreiche Projekte geeignet.
- Der Benutzer kann nur die Funktionen zum Postprocessing benutzen, die die Auswertesoftware unterstützt.
- Durch die Datenbankbindung erhöht sich die Simulationsdauer erheblich und beim Postprocessing wird mehr Zeit benötigt, die Prozessdaten aus der Datenbank zu laden.
- Simulationsdaten werden durch verschiedene Softwareebenen geführt, wobei stets die Datenintegrität sichergestellt werden muss.

### 6.3 Fazit

Infolge der Digitalisierung der Industrie im Rahmen von *Industrie 4.0* werden Fertigungssimulationen vor allem bei virtuellen Inbetriebnahmen eine immer größere Rolle spielen. In dieser Arbeit wurde ein Ansatz geliefert und für Schmelz- und Druckgussbetriebe umgesetzt, wie Daten von Fertigungssimulationen durch die automatisierte Berechnung von Kennzahlen effizient ausgewertet werden können. Die plattformunabhängige Speicherung der Simulationsdaten ermöglicht es zudem, *Melting Analytics* für Prozessdaten des realen Betriebes einsetzen zu können.

## 7 Quellen

- [1] Jeckle, D. (2015): *Dokumentation der Software zur Simulation des Materialflusses und Untersuchung der Energieeffizienz eines Schmelz- und Druckgussbetriebes*, Studienarbeit, Fakultät Ingenieurwissenschaften, Hochschule Ansbach, Ansbach, 2015, S.8
- [2] Herrmann, C., Priesund, H., Hartmann, G (2013): *Energie- und ressourceneffiziente Produktion von Aluminiumdruckguss*. Springer Vieweg, Berlin, 2013.
- [3] Cleff, T. (2015): *Deskriptive Statistik und Explorative Datenanalyse: Eine computergestützte Einführung mit Excel, SPSS und STATA*, 3. Auflage, Wiesbaden: Springer Verlag, 2015, S. 4.
- [4] Jacobson, I., Booch, G., Rumbaugh, J. (1999): *The Unified Software Development Process*, Reading, Massachusetts: Addison-Wesley
- [5] Davenport, T. (2014): *big data @ work*, Massachusetts: Harvard Business Review Press, S.4
- [6] Meier, A., Kaufmann, M. (2016): *SQL- & no-SQL- Datenbanken*, 8. Auflage, Heidelberg: Springer Vieweg, S. 18
- [7] MathWorks: *Choosing Between ODBC and JDBC Drivers*, abgerufen am 02.06.2016 von Documentation: <http://de.mathworks.com/help/database/ug/choosing-between-odbc-and-jdbc-drivers.html>
- [8] Schulz, K. (2002): *Java professionell programmieren*, Heidelberg: Springer, S. 465
- [9] C. May, C., Koch, A.: *Overall Equipment Effectiveness (OEE): Werkzeug zur Produktivitätssteigerung*, Zeitschrift der Unternehmensberatung - ZUB, Seite 245–250.

## 8 Anhang

### Anhang 1: GUI Melting Analytics

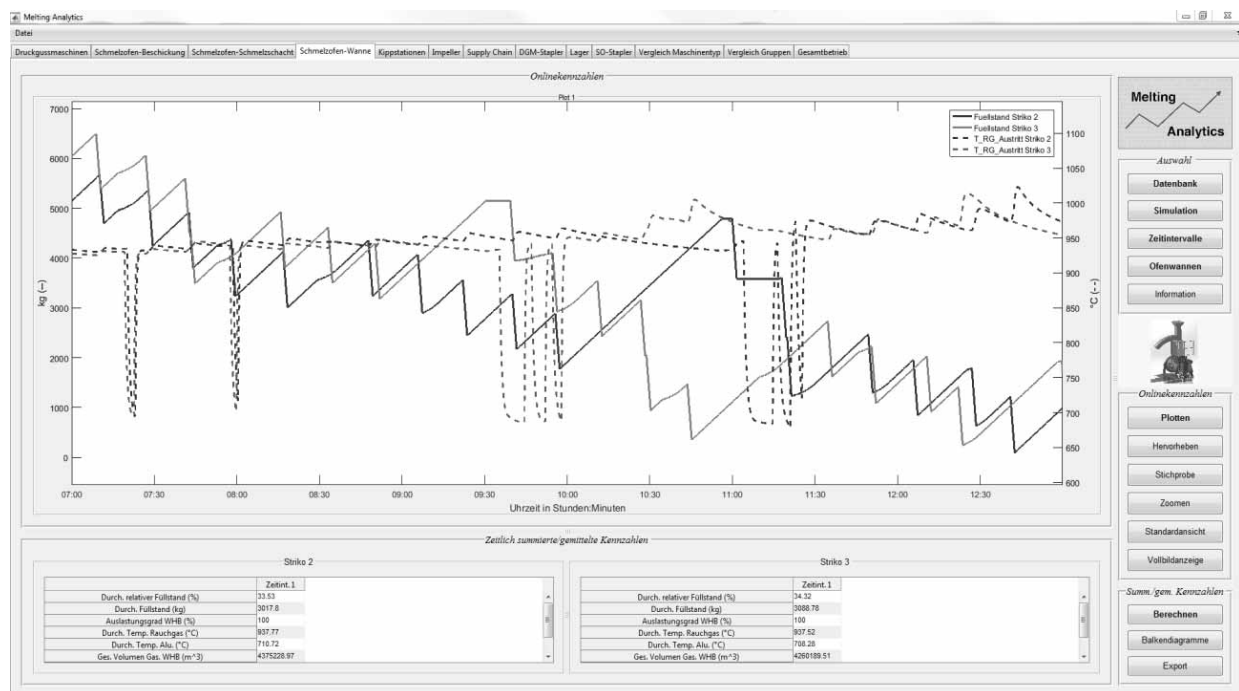


Abbildung 6: GUI Melting Analytics

## Automation-Data-Exchange (ADEX) - Durchgängiges Austauschsystem für Simulation und Visualisierung

David Weigert<sup>1</sup>, Paul Aurich<sup>1</sup>

<sup>1</sup>Otto-von-Guericke-Universität Magdeburg

david.weigert@ovgu.de

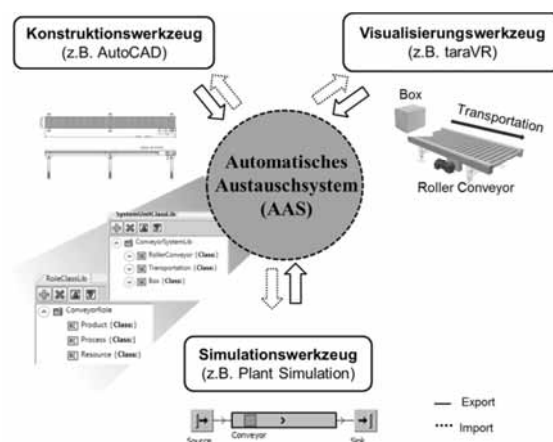
Die automatisierte sowie teilautomatisierte Generierung von Materialflusssimulationsmodellen wird bereits seit Jahrzehnten diskutiert und entwickelt. Mit AutomationML (AML) ist ein offenes, objektorientiertes, XML-basiertes Speicherungs- und Austauschformat gegeben, welches zusätzlich ein Austausch zwischen Visualisierungs-, Konstruktions- und Simulationswerkzeugen ermöglichen soll. Die Anwendung von AML innerhalb der unterschiedlichen Werkzeuge soll eine durchgängige Gestaltung innerhalb des Produkt- und Produktionsentstehungsprozesses von Produktions- und Intralogistiksystemen vorantreiben. Durch das neutrale Austauschformat soll es ermöglicht werden die unterschiedlichen Kernfunktionen der Werkzeuge sowie deren proprietäre Schnittstellen aufzuheben. Die Zielstellung des FuE-Kooperationsprojektes „ADEX - AutomationDataEXchange“ ist die Entwicklung dieser einheitlichen, digitalen Planungsmethoden und -werkzeuge durch AML. Es werden quelloffene automatisierte Import- und Exportlösungen der unterschiedlichen Werkzeuge entwickelt. Die Kernaufgabe der derzeit laufenden Umsetzungsphase ist die detaillierte Entwicklung des Automatischen Austauschsystems (AAS). Der Fokus liegt hierbei zunächst im Datenaustausch von Visualisierungs- und Simulationswerkzeugen. Erste prototypische Umsetzungen skizzieren den Nutzen aber auch die Grenzen der automatischen Modellgenerierung via AML. Bisher können einfache Fördersysteme aus Geraden-, Kurvenförderbändern, Drehtischen und Einzelstationen aus dem Visualisierungswerkzeug Tarakos - taraVR automatisch im Simulationswerkzeug Siemens - Plant Simulation modelliert werden. Hierbei kann auf die Verwendung kostenintensiver und optionaler Schnittstellenbibliotheken verzichtet werden.

### 1 ADEX

Auf Grund immer kürzer werdender Produktlebenszyklen wird in der Fabrikplanung zunehmend Wert auf die Wandlungsfähigkeit von Produktions- und Logistikprozessen gelegt. Die Wandlungs- und Wettbewerbsfähigkeit einer Fabrik kann durch flexible Produktions- und Logistikprozesse sowie variable Anlagen- und Einrichtungsplanung verbessert werden. Der Einsatz digitaler Werkzeuge bestehend aus Simulation, Visualisierung und Konstruktion unterstützt die Fabrikplanung und wirkt positiv auf den gesamten Produktentstehungsprozess, als Teil des Produktlebenszyklus.

Die Zielstellung des FuE-Kooperationsprojektes „ADEX - AutomationDataEXchange“ ist die Entwicklung einheitlicher, digitaler Planungsmethoden und -werkzeuge für eine durchgängige Gestaltung innerhalb des Produkt- und Produktionsentstehungsprozesses von Produktions- und Intralogistiksystemen. Kernaufgabe des „ADEX“-Projekts ist die Entwicklung und Erprobung eines automatischen Austauschsystems (AAS) zwischen Visualisierungs-, Konstruktions- und Simulationswerkzeugen. Hierbei

steht die verlustfreie und beschleunigte Konvertierung und Modellierung innerhalb der verschiedenen Werkzeuge im Fokus (s. **Abbildung 1**).



**Abbildung 1.** Funktionen des automatischen Austauschsystems

Auf Grund der unterschiedlichen Kernfunktionen jedes Werkzeuges sowie deren proprietärer Schnittstellen besteht das Ziel quelloffene automatisierte

Import- und Exportlösungen zu entwickeln. Folgende Vorteile ergeben sich aus dem Einsatz von einem automatischen Austauschsystem:

- Vorhandene Simulations-, Visualisierungs- und Konstruktionswerkzeuge im Unternehmen bleiben bestehen, dies verhindert kostenintensive Neuinvestitionen
- Produktivität und Kostenreduktion können durch den Einsatz eines AAS erreicht werden, da die individuellen Vorteile der Werkzeuge kombiniert werden
- Visualisierung, Modellierung und Simulation von realen intralogistischen Systemen wird beschleunigt, da der überwiegend manuelle und kostenintensive Aufwand bei einer erneuten Modellerstellung reduziert wird

Die Kernaufgabe der derzeit laufenden Umsetzungsphase ist die detaillierte Entwicklung des Automatischen Austauschsystems (AAS). Der Fokus liegt hierbei zunächst im Datenaustausch von Visualisierungs- und Simulationswerkzeug.

## **2 Stand der Technik und Wissenschaft**

### **2.1 Automatisierte/Teilautomatisierte Modellgenerierung**

Die automatisierte bzw. teilautomatisierte Generierung von Materialflusssimulationsmodellen wird bereits seit zwei Jahrzehnten diskutiert und entwickelt. Erste Ansätze kommen von Lorenz und Schulze [1], welche layoutbasiert Modelle generieren oder von Splanemann [2], dessen Ansatz sowohl strukturelle als auch Daten aus einem PPS-System nutzt um Modelle zu generieren. Eine erste Klassifizierung von automatischen Modellgenerierungsansätzen wurde von Eckhardt [3] bereitgestellt. Unterschieden werden hierbei parametrische, strukturelle und hybride Ansätze. Bei parametrischen Ansätzen erfolgt die Auswahl und Konfiguration von Standardsimulationsbausteinen durch Parameter. Strukturelle Ansätze basieren dagegen auf der Struktur des abzubildenden Systems, welches vornämlich durch CAD-Daten verkörpert wird.

Eine weitere Möglichkeit zur Klassifizierung ist der morphologische Kasten von Straßburger [4]. Basierend auf diesem kann ADEX folgendermaßen eingeordnet werden:

- Einsatzfall: planungsbegleitend (taktisch)
- Fokussiertes Gewerke: Fördertechnik

- Grad der Automatisierung: teilautomatisch
- Ansatz: direkter generischer Aufbau
- Unterstützung bei Modellerstellung
- Schnittstelle: Text- und XML-basiert

Weitere Ansätze für die kontinuierliche digitale Planung und Kontrolle beschreiben die Enterprise Application Integration (EAI) oder Unternehmensanwendungsintegration (UAI) und Service Oriented Architecture (SOA) oder Serviceorientierte Architektur. Die EAI stellt integrierte Geschäftsprozesse entlang der Wertschöpfungskette dar. Unternehmensanwendungen verschiedener Generationen und Architekturen können somit über ein gemeinsames Netzwerk interagieren [5][6]. Die SOA beschreibt eine Methode die vorhandene EDV-Komponenten wie Datenbanken, Server und Websites in Dienstleistungen abkapselt und koordiniert, somit werden die bereitgestellten Dienste zu höheren Diensten zusammengefasst und anderen Abteilungen der Organisation zur Verfügung gestellt [7][8]. Ziele sind die langfristige Senkung der Kosten bei der Entwicklung der Produktionspläne und eine größere Flexibilität der Geschäftsprozesse durch die Wiederverwendung von bestehenden Diensten. Die Kosten für zukünftige Entwicklungen werden reduziert, da bereits alle notwendigen Dienstleistungen verfügbar sind und diese nur kombiniert werden müssen. Grund für die schleppende Weiterentwicklung ist die hohen Anforderungen an die Datensicherheit, Kontinuität der Werkzeugentwicklung sowie unsichere Systeme und Produktentwicklungsprozesse [9][10].

Im Rahmen der Diskussionen um die „Digitale Fabrik“ wurde vermehrt zum Thema automatisiert Modellgenerierung publiziert. Straßburger [4] identifizierte in diesem Zusammenhang die Implementierung von Steuerungs- und Routingstrategien sowie von komplexem Systemverhalten als Hauptproblem der automatisierten Modellgenerierung. Hierfür führt er diverse Gründe an. Zum einen können Steuerungsregeln vom Planer oftmals nicht innerhalb der Layoutwerkzeuge in die Planungsdaten integriert werden, zum anderen fehlen den Layoutplanern häufig entsprechende Kompetenz zur Festlegung der Steuerungsregeln. Rooks [11] schlägt vor die fehlenden Informationen im Rahmen der digitalen Prozessplanung zu erzeugen und mit den Layoutdaten zusammenzuführen. Bei bausteinorientierten Simulatoren sind 10-30 % der Steuerungs- und Routingstrategien nicht durch eine direkte Parametrisierung der Stan-

dardbausteine abbildbar [4]. Um diese komplexen Systemprozesse implementieren zu können, ist eine manuelle Modellierung notwendig, welche typischerweise durch Skriptsprachen ausgeführt wird. Die automatische Generierung solcher Logiken ist mit den Skriptsprachen der meisten Simulatoren kaum umsetzbar [4].

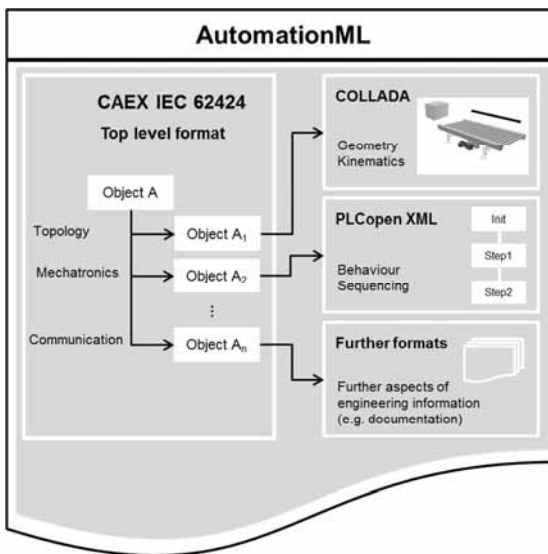
Bei der automatisierten Modellgenerierung, welche auch als „datengetriebene Modellgenerierung“ bezeichnet wird [12], spielt die Herkunft und die Art der Informationen eine wesentliche Rolle. Bekannte Datenstandards, die bisher zur automatischen Modellgenerierung eingesetzt werden, sind SDX und CMDS. Das SDX-Format von Siemens dient ausschließlich der Bereitstellung von Layoutinformationen. Mit dem offenen, XML-basierten CMDS-Format können sowohl layout- als auch prozessbezogene Informationen übermittelt werden. [4] Die Problematik der Abbildung von komplexen Verhaltenslogiken, ist jedoch auch mit diesem nicht vollständig lösbar (vgl. Bergmann und Straßburger [13] und Bergmann [12]). Mit dem im ADEX-Projekt verwendeten AutomationML gibt es einen weiteren Datenstandard für die automatische Modellgenerierung, dieser wird im nächsten Kapitel ausführlicher beschrieben.

ADEX kann im Themenfeld als weiterer Ansatz zur teilautomatisierten Modellgenerierung eingeordnet werden. Die Problematik der Abbildung von Steuerungsprozessen ist bis zu diesem Zeitpunkt allerdings noch nicht ausreichend konzeptioniert, um sie als gelöst zu betrachten.

## 2.2 AutomationML

AutomationML (AML) ist ein offenes, freiverfügbares, objektorientiertes, XML-basiertes Speicherungs- und Austauschformat, welches vom AutomationML e.V. entwickelt wird. AML dient dem Austausch von Engineering-Daten zwischen den heterogenen Werkzeugen der „Digitalen Fabrik“. Ziel des AutomationML e.V. ist es die Transferkosten zu reduzieren, da diese einen erheblichen Anteil der Engineeringkosten ausmachen. [14]

Bisher können mit AML Topologie, Geometrie, Kinematik und Verhalten von Systemkomponenten beschrieben werden (s. **Abbildung 2**).



**Abbildung 2.** Inhalt und Funktion von AML in Anlehnung an [15]

Die hierarchische Abbildung der Topologie eines Planungsgegenstandes erfolgt mittels CAEX (Computer Aided Engineering Exchange). Das CAEX-Bibliothekskonzept umfasst drei Bibliotheksarten [16]:

- Die SystemUnitClass-Bibliothek entspricht einem Katalog von konkreten physikalischen oder logischen Anlagenobjekten oder deren Kombination. Den Elementen sind Attribute, Schnittstellen und verschachtelte interne Elemente sowie deren Verbindungen zugewiesen.
- Die RoleClass-Bibliothek definiert abstrakte physikalische oder logische Anlagenobjekte, unabhängig von einer konkreten technischen Realisierung. Rollen beschreiben die Funktionsweise von Anlageobjekten.
- Die InterfaceClass-Bibliothek beschreibt die Art der Schnittstellen zwischen den Anlageobjekten. Es werden die Relationen zwischen Anlageobjekten abgebildet.

Geometrie und Kinematik können durch Collada-Dateien einzelnen Systemkomponenten zugeordnet werden. Ebenso wird die mit PLCopenXML definierte Steuerung bzw. das Verhalten referenziert. AML ist anpassbar und flexibel gestaltet, es bietet die Möglichkeit weitere XML-Formate einzubinden. [15]

AutomationML besitzt außerdem eine inhärente verteilte Datenstruktur. Die Informationen werden statt in einem monolithischen XML-Dokument, als Einzeldokumente gespeichert.

Die Wiederverwendbarkeit einzelner Systemkomponenten und der Aufbau von Element-Bibliotheken werden somit erleichtert. [16]

### 3 Konzept und Umsetzung

#### 3.1 Automatisches Austauschsystem

Das automatische Austauschsystem wird durch ein System von Abbildungsregeln und einem dazugehörigen Benutzerinterface definiert. Innerhalb des Austauschsystems wird das zu übertragende Modellelement mit bestehenden oder neuen Rollenprofilen, aus der hierfür zu entwickelnden AutomationML Rollenbibliothek belegt. Hierdurch wird eine einheitliche Modellgenerierung innerhalb der verschiedenen Werkzeuge angestrebt.

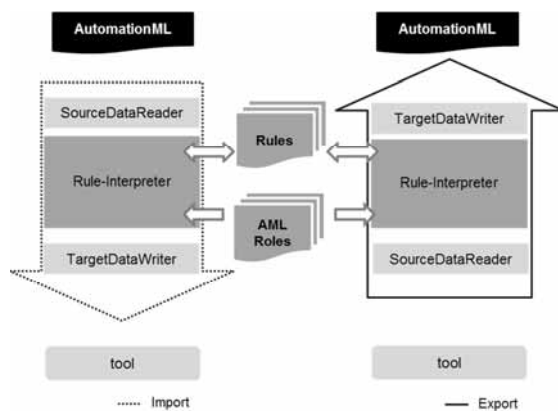
Sollte es bei der Zuordnung der unterschiedlichen Modelle aus den verschiedenen Werkzeugen zu Konflikten kommen kann dies über ein Interface separat durch eine manuelle Auswahl behoben werden. Neben der Übertragung von Modellelementen ist es von Bedeutung deren Attribute wie Lage, Transportrichtung, Geschwindigkeit und weitere Zustandsbeschreibungen zu übernehmen. Des Weiteren müssen die Beziehungen zwischen den Elementen übertragen werden. In der Anfangsphase des Projektes erfolgte die Konzeptionierung der Schnittstellen für den Import und Exporter von Visualisierung zu Simulation. Hierfür wurden die zu übertragenden Parameter analysiert und die verfügbaren Schnittstellen verglichen. Der Ansatz eines quelloffenen Standards über die AML-Funktion grenzt die vorhandenen Schnittstellenpakete und kaufbaren Bibliotheken für den Datenaustausch aus. Ziel ist es unabhängige Modellerstellung über geschlossene Softwarepakete hinweg zu realisieren. Hierdurch soll eine umständliche und oftmals zeitintensive als auch fehlerbehaftete Neumodellierung reduziert werden.

Die Kernaufgabe der derzeit laufenden Umsetzungsphase ist die detaillierte Entwicklung des Automatischen Austauschsystems (AAS). Der Fokus liegt zunächst in der Verbindung von Visualisierungs- und Simulationswerkzeug.

#### 3.2 Datenübertragung vom taraVRbuilder zu Plant Simulation

Die Visualisierungssoftware „taraVRbuilder“ der Firma Tarakos verfügt bereits über einen AML-Exporter. Beim Export werden die in „taraVRbuilder“ modulierten Objekte mit ihren Parametern und Beziehung als AML-Datei gespeichert. Anschließend

wird die AML-Datei vom Regelinterpreter verarbeitet. Basierend auf der Funktion eines Objektes erfolgt die Zuweisung zu einer der vordefinierten AutomationML-RoleClass. Die RoleClass ist wiederum einem „Plant Simulation“-Baustein zugewiesen. Somit wird sichergestellt, dass verschiedene Objekte mit gleichartiger Funktionsweise in „Plant Simulation“ identisch abgebildet werden. Band-, Rollen-, Kettenförderer des „taraVRbuilder“ werden beispielsweise allesamt als „Plant Simulation“ Förderstrecke interpretiert. Eine allgemeingültige Darstellung der Lösung für den Import und Export vom und in das Simulationswerkzeug verdeutlicht **Abbildung 3**.



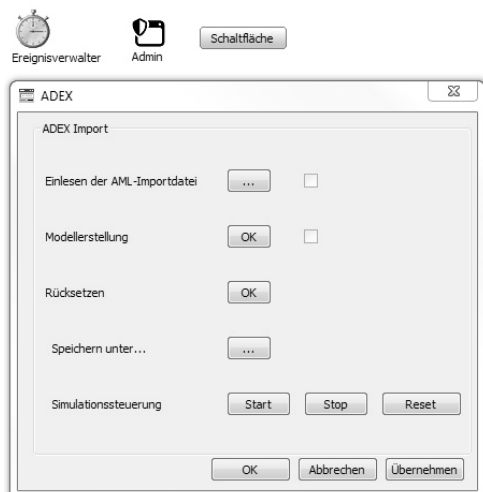
**Abbildung 3.** Funktion und Struktur von Import und Export

Die Konzeptionierung des Importers zu „Plant Simulation“ wurde maßgeblich durch die Forderung nach Allgemeingültigkeit bestimmt. Ziel ist es im FuE-Projekt ADEX das entwickelte AAS sämtlichen „Plant Simulation“-Anwendern lizenzunabhängig bereit zu stellen. Schnittstellen, die nur durch den Erwerb der Lizenzen „Professionell“ oder „Interface Package“ zur Verfügung stehen, wurden dementsprechend nicht berücksichtigt. Der Import zu „Plant Simulation“ erfolgt über die Dateischnittstelle. Die strukturierte Textdatei kann darüber hinaus von anderen Simulatoren (z.B. ExtendSim, AnyLogic) verarbeitet werden. Die Erweiterbarkeit von ADEX auf diese Simulatoren dient dem Projektziel, heterogene Planungswerkzeuge durch das AAS miteinander verknüpfen zu können.

#### 3.3 Automatisierte Modellgenerierung

Die automatisierte bzw. teilautomatisierte Modellgenerierung erfolgt mit einem standardisierten „Mastermodell“. Dieses beinhaltet ein Userinterface und Methoden, welche die Modellgenerierung ausführen.

Durch das in **Abbildung 4** gezeigte Userinterface kann eine Textdatei ausgewählt werden. Durch Betätigung eines Buttons wird die Modellgenerierung ausgelöst. Des Weiteren kann das „Mastermodell“ zurückgesetzt, gespeichert oder ein Simulationslauf gestartet und gestoppt werden. Das Userinterface soll im späteren Projektverlauf um die Exportfunktionen erweitert werden.



**Abbildung 4.** Userinterface des Importers im Simulationswerkzeug

Die Modellgenerierung ist in eine „Plant Simulation“-Methode implementiert (s. **Code 1**). Zur Erstellung eines statischen Systemelementes wird zu nächste aus der Textdatei der standardisierte Objekttyp ausgelesen. Die Methode springt zum für diesen Objekttyp definierten Teil des Codes und führt diesen aus. Zu Beginn werden die Parameter des Objekts ausgelesen und in Variablen gespeichert. Typische Parameter sind:

- Objektname
- ID
- x-/y-Position
- Drehwinkel
- objektspezifische Attribute  
z.B. Länge, Breite, Geschwindigkeit, Zeitverbrauch
- In- und Output-ID

Anschließend wird das Materialflussobjekt entsprechend der x-/y- Position in das Modell-Netzwerk eingefügt. Es folgt die Benennung, Etikettierung und Parametrisierung durch Definierung der Attribute des Materialflussobjekts.

Abschließend wird das Materialflussobjekt entsprechend der In- und Output-ID mit den bereits vorhandenen Vorgängern und Nachfolgern durch Kanten verbunden. Sollte im Datensatz keine Information zu In- oder Output-ID vorhanden sein, wird automatisch eine Quelle bzw. Senke erstellt.

```
OPEN textfile
READ objecttype
WHILE NOT textfile.end
  CASE objecttype OF
    CASE "conveyor"
      READ: name, ID, length, width,
            speed, capacity, angle, x-pos.,
            y-pos., input-ID, output-ID
      CREATE conveyor AT x-pos., y-pos.
      SET name, ID, length, width, speed,
            capacity, angle
      IF input-ID <> ""
        CONNECT TO input-ID
      ELSE
        CREATE source AND CONNECT TO
          source
      END IF
      IF output-ID <> ""
        CONNECT TO output-ID
      ELSE
        CREATE sink AND CONNECT TO
          sink
      END IF
    CASE "curve"
      ...
  END CASE
ENDWHILE
CLOSE textfile
```

**Code 1.** Modellgenerierung in Plant Simulation

Exemplarisch wird die Umsetzung eines Visualisierungsmodells in das Simulationswerkzeug durch das automatische Austauschsystem mit dem Quelltext aus der formatierten Textdatei in **Abbildung 5** gezeigt. Das prototypische Beispiel für eine erste Möglichkeit der automatischen Modellgenerierung skizziert das Potential aber auch die Grenzen der Entwicklung. Bisher können einfache Fördersysteme aus Geraden-, Kurvenförderbändern, Drehtischen und Einzelstationen automatisch modelliert werden. Weitere Materialflussobjekte sollen in Kürze folgen. Diskutiert wird momentan die Zweckmäßigkeit und Möglichkeit benutzerdefinierte „Plant Simulation“-Bausteine generieren und parametrisieren zu können. Grundsätzlich müssten die Elemente von benutzerdefinierte Klassenbibliotheken ausgelesen werden können. Wobei sowohl die neuen Objekttypen, mit den vor-

handenen Rollen zu verknüpfen wären, als auch eine Möglichkeit gefunden werden müsste, die Attribute dieser Objekte anzusprechen.

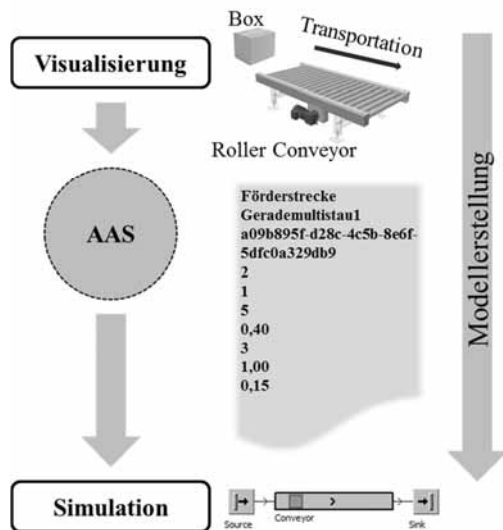


Abbildung 5. Exemplarische Modellumsetzung in Plant Simulation

Ein benutzerdefiniertes Objekt wird durch ein Subnetzwerk implementiert. Diese können mehreren Standardbausteinen und Methoden enthalten. Fraglich ist, welche Parameter als Attribut welchen Objekts einzustellen sind oder ob die Attribute durch Methoden an die Objekte übergeben werden. Letzteres entspräche in der Umsetzung einer automatische Durchsuchung und Modifikation des Codes von Methoden.

Ein weiterer Schritt ist die Generierung von beweglichen Materialflussobjekten, mit Attributen und entsprechenden Arbeitsplänen. Der Datenaustausch wird identisch zu den statischen Materialflussobjekten erfolgen. Die Steuerung bzw. das Routing der beweglichen Objekte durch das Materialflusssystem anhand der Arbeitspläne muss noch konzeptioniert werden.

#### 4 Zusammenfassung und Ausblick

Die hier beschriebene Entwicklung „ADEX“ stellt eine erste Entwicklung für die automatische, quelloffene sowie frei konfektionierbare Modellerstellung und -übertragung zwischen unterschiedlichen Werkzeugen dar. Dies entspricht den Anforderungen an ein integriertes Planungstool. Es ist eine Möglichkeit geschaffen individuelle AML-Bibliotheken für Simulation, Visualisierung und Konstruktion einer wiederholten Nutzung zuzuführen. Die Entwicklung hilft somit den Austausch von Informationen und Modell-

komponenten durch die Verknüpfung verschiedener Werkzeuge zu erleichtern. Die ersten Ergebnisse zeigen, dass die Umsetzung erfolgreich möglich ist. Ebenso kommen hierbei keine proprietären Softwarelösungen zum Einsatz. Mit der Möglichkeit die Modellerstellung innerhalb der unterschiedlichen Werkzeuge zu vereinen kann es in Zukunft möglich sein, genaue und schnelle Modelle aus einer Hand zu produzieren. Dadurch werden kostspielige, fehleranfällige und zeitintensive erneute Modellerstellung vermieden.

Der bisherige Stand der Konzeptionierung und Umsetzung bildete die Grundlage für die zeitnahe Implementierung der automatischen Generierung weiterer statischer Materialflussobjekte. Hierzu müssen noch die zu übertragenden Attribute und die Struktur der Textdatei abgestimmt werden. Die parallel laufende Entwicklung der automatischen Modellgenerierung für AutoCAD befindet sich auf einem ähnlichen Stand. Schnittstellen wie ActiveX und .NET sind bei AutoCAD ohne zusätzliche Lizenzen verfügbar. Dadurch können die Informationen aus der AML-Datei direkt ausgelesen und verarbeitet werden. Bisher werden die statischen Materialflussobjekte durch einfache geometrische Körper dargestellt. Problematisch ist auch hier die Anforderung benutzerdefinierte „Blöcke“ automatisch zu Parametrisieren.

Die beschriebene Entwicklung ist ein erster Schritt für die automatische Modellerstellung. Dies entspricht den Anforderungen für ein integriertes und automatisierbares Planungstool. Es ist nun möglich, gemeinsame und individuelle AML-Bibliotheken für Simulation, Visualisierung und Konstruktion zu erstellen, zu teilen und wiederholend zu verwenden. Weitere Schritte sind die zunehmende Detaillierung und Standardisierung des automatisierten Austauschsystems. Zusätzlich steht die stärkere Dynamisierung bei der Erzeugung benutzerdefinierte Materialflussobjekte im Simulationswerkzeug im Vordergrund. Hierbei geht es um die Identifizierung der Bausteine sowie deren Verknüpfung mit vorhandenen Rollen und die notwendige Parametrisierung. Zukünftig sollen auch die Frage nach der Implementierung und Übermittlung der Arbeitspläne und die damit einhergehende Steuerung und das Routing der Elemente im Simulationswerkzeug erforscht werden.

## 5 Literaturverzeichnis

- [1] P. Lorenz und T. Schulze. *Layout Based Model Generation*. Proceedings of the 27th Conference on Winter Simulation, IEEE Computer Society, Washington, DC, USA, pp. 728–735, 1995.
- [2] R. Splanemann, M. Roth, und S. Soravia. *Einsatz der Materialflußsimulation zur Planung, Analyse und Optimierung von verfahrenstechnischen Produktionsanlagen*. Chemie Ingenieur Technik, vol. 67, no. 9, pp. 1107–1108, 1995.
- [3] F. Eckardt. *Ein Beitrag zu Theorie und Praxis datengetriebener Modellgeneratoren zur Simulation von Produktionssystemen*. Shaker, Aachen, 2002.
- [4] S. Straßburger, S. Bergmann, und H. Müller-Sommer. *Modellgenerierung im Kontext der Digitalen Fabrik - Stand der Technik und Herausforderungen*. Integrationsaspekte der Simulation: Technik, Organisation und Personal, G. Zülch und P. Stock, Eds., KIT Scientific Publ, Karlsruhe, pp. 37–44, 2010.
- [5] S. Aier and M. Schönherr. *Enterprise application integration - Flexibilisierung komplexer Unternehmensarchitekturen (Enterprise Architecture)*. Gito-Verlag, Berlin, 2004.
- [6] M. Kaib. *Enterprise Application Integration - Grundlagen, Integrationsprodukte, Anwendungsbeispiele*. Dt. Univ.-Verl., Wiesbaden, 2004.
- [7] N. Bieberstein. *Executing SOA - A practical guide for the service-oriented architect*. IBM Press/Pearson plc, Upper Saddle River, N.J., 2008.
- [8] D. Liebhart. *SOA goes real - Service-orientierte Architekturen erfolgreich planen und einführen*. Hanser, München, 2007.
- [9] A. Fay. *Reduzierung der Engineering-Kosten für Automatisierungssysteme*. Industrie Management, no. 2, pp. 29–32, 2006.
- [10] G. Rauprich, C. Haus, und W. Ahrens. *PLT-CAE-Integration in gewerkeübergreifendes Engineering und Plant-Maintenance*. Automatisierungstechnische Praxis, vol. 44, no. 2, pp. 50–62, 2002.
- [11] T. Rooks. *Rechnergestützte Simulationsmodellgenerierung zur dynamischen Absicherung der Montagelogistikplanung bei der Fahrzeugneutypplanung im Rahmen der Digitalen Fabrik*. Dissertation, Technische Universität Clausthal, 2009.
- [12] S. Bergmann. *Automatische Generierung adaptiver Modelle zur Simulation von Produktionssystemen*. Dissertation, Technische Universität Ilmenau, 2013.
- [13] S. Bergmann, A. Fiedler, und S. Straßburger. *Generierung und Integration von Simulationsmodellen unter Verwendung des Core Manufacturing Simulation Data (CMSD) Information Model*. Integrationsaspekte der Simulation: Technik, Organisation und Personal, G. Zülch und P. Stock, Eds., KIT Scientific Publ, Karlsruhe, pp. 461–468, 2010.
- [14] AutomationML e.V. *AutomationML - story*. <https://www.automationml.org/o.red.c/story.html>.
- [15] A. Lüder, L. Hundt, und N. Schmidt. *AutomationML – Fachexperten erklären das Format*. <http://www.sps-magazin.de/downloads/WhitepaperAutomationML.pdf>.
- [16] R. Drath, ed. *Datenaustausch in der Anlagenplanung mit AutomationML - Integration von CAEX, PLCopen XML und COLLADA*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010.

# Simulationskonzepte in der Brandschutzausbildung

Prof. Dr-Ing. Dieter Wloka, M.Eng.<sup>1</sup>,

<sup>1</sup>Fachgebiet Technische Informatik

FB 16 Elektrotechnik/Informatik

Universität Kassel

Wilhelmshöher Allee 71-73

34128 Kassel

*dwloka@uni-kassel.de, www.katie-katastrophensimulation.de*

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. Ein bedeutender Bereich ist die Prävention von Bränden. Ein Teilprojekt befasst sich mit der Entwicklung von Ausbildungsmedien zur Brandschutzunterweisung. In diesem Beitrag wird ein Überblick über das Teilprojekt sowie die Entwicklung der verschiedenen 3D Modelle gegeben.

## 1 Einleitung

Für die Entwicklung von Katie werden verschiedene Technologien, u.a. aus dem Bereich der Simulation eingesetzt und verknüpft, wie

- E-learning,
- Virtuelle Realität,
- 3D-Modellierung und Simulation,
- Game-Engines,
- Darstellung virtueller Menschen und
- Immersion durch Techniken der Augmented - und Virtuellen Realität.

Das Projekt Katie unterstützt die Ausbildung, indem Handlungen und Lösungsstrategien aufgezeigt werden.

## 2 Technischen Regeln für Arbeitsstätten

Die Technischen Regeln für Arbeitsstätten (ASR) konkretisieren die Anforderungen der Arbeitsstättenverordnung (ArbStättV). Die Regelungen der ASR A2.2 umfassen:

- Grundschutz
- Erhöhte Brandgefahr
- Berücksichtigung anderer Verordnungen
- Verkürzte Reaktionszeit bei der Brandmeldung
- Regelmäßige Unterweisung, Brandschutzhelfer

Wichtig ist es regelmäßig unterweisen. Natürlich genügt eine ausreichende Anzahl an tragbaren Feuerlöscher 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 Verhaltes im Gefahrenfall zu unterweisen.

## 3 Brandschutzhelfer

Zusätzlich muss der Arbeitgeber eine ausreichende Anzahl von Beschäftigten zu Brandschutzhelfern ausbilden. Ausreichend heißt: ein Anteil von 5 % der Beschäftigten ist als Brandschutzhelfer zu qualifizieren.

Ausbildungsinhalte sind:

- Grundzüge des vorbeugenden Brandschutzes

- Gefahren durch Brände
- Verhalten im Gefahrfall
- Kenntnisse der betrieblichen Brandschutzorganisation
- Funktion und richtiger Einsatz der Löschgeräte
- praktische Übung im Umgang mit Löschgeräten

## 4 Aktuelle Ausbildungssituation

Viele Unternehmen versuchen ihrer Pflicht zur Unterweisung nachzukommen. Hierzu werden oft externe Trainer engagiert, die eine Schulung durchführen. Diese Präsenzsulung 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 verschiedener Elemente durchzuführen:

- E-Learning Phase für das Erlernen der Theorie
- Praxisphase durch Simulation in einer Präsenzsulung
- Praxisphase durch Simulation über Browser

## 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 Arbeits-Umfeldes
- Immersion mittels VR/ AR -Technologien
- Natürliche Interaktion in 3D-Welten

## 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, Feuerlöcher, Fluchtschilder, etc.

## 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. Für Ausbildungszwecke der Feuerwehr wurde ein Modell des Krankenhauses erstellt, wie in Bild 1 gezeigt.

Das Training der Mitarbeiter wird auf Stationsebene, also im Rahmen von Patientenzimmern durchgeführt. Bild 2 zeigt das Photo eines tatsächlichen Patientenzimmers.



Abbildung 1: Modell Krankenhaus



Abbildung 4: 3D-Patientenzimmer



Abbildung 2: Beispiel: reales Patientenzimmer

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 zudem nur für wenige Mitarbeiter das persönliche Arbeitsumfeld abbilden.

Durch die Nutzung der verfügbaren Technologien einer Game-Engine ergeben sich aber wesentliche Vorteile, wie verfügbare Navigation, Partikelfeuer, etc., gegenüber einer eigenen Programmierung. Bild 3 zeigt einen Ausschnitt des 3D-Stationsmodells, Bild 4 zeigt exemplarisch das 3D-Modell des Patientenzimmers.



Abbildung 3: 3D-Stationsmodell

Interessante Fragen sind nun, welche Art von Training durchgeführt werden sollte und kann und ob eine Wiedererkennung seiner "eigenen" Station wichtig ist.

Bild 5 zeigt exemplarisch die Sicht in den Flur der Station. Man erkennt die wichtige Integration von brand-schutztechnischen Komponenten, wie Brandmelder, Fluchtwegkennzeichnung und Wandhydranten.



Abbildung 5: Blick in den Flur

## 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 6 zeigt als Beispiel

ein Photo mit überlagertem Partikelfeuer als Konzeptbild. Für Löschübungen ist dieses Konzept weniger geeignet.



**Abbildung 6:** 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-Photomodellworkflow 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 7 zeigt das 3D Modell, welches aus einer Vielzahl von Bildern zusammengesetzt wurde.



**Abbildung 7:** Simulation mit 3D Photomodell

## 11 Browser basiertes 3D-Modell

Möchte man ein grosse Zahl von Nutzern erreichen, ist der Zugriff über Browser eine denkbare Version. Idealerweise sollte eine solche Lösung auf die Installation von Plug-Ins für den Browser verzichten, da gerade bei Firmen-Nutzern dies nicht möglich ist.

Die Nutzbarkeit eines 3D-Modells im Browser hängt nun von der Größe des Modells, Optimierungen und der verfügbaren Netzbandbreite ab.

Ein erster Prototyp der Krankenstation wurde mit Hilfe von WEBGL geschaffen. Dank zahlreicher Optimierungen konnte die Modellgröße deutlich reduziert werden.

## 12 Immersives Modell

Den Nutzer in eine 3D-Simulation hineinzusetzen erscheint als optimale Trainingslösung. Ein erster Schritt ist eine stereoskopische Projektion, die mittels PC oder Beamer genutzt werden kann. Systeme wie Powerwalls bieten hohe Auflösungen, CAVes eine Rundumsicht.

Einen großen Grad an Immersion bietet ebenfalls ein Head-Mounted Display. Diese sind schon seit den 1990iger Jahren bekannt. Seit kurzem besteht ein neuer VR Hype und viele neuartige Geräte sind verfügbar.

Im Rahmen des Projektes KATIE wurde ein Labor als Holodeck umgerüstet. Hier steht eine VR-Anlage zur Verfügung, die einen begehbaren Raum von ca 20 m<sup>2</sup> ermöglicht. Diese kann mit einer zweiten Anlage verbunden werden, die in ein MOCAP Labor integriert ist.

Die Nutzung von HMDs für Ausbildungszwecke bietet jedoch einige Überraschungen hinsichtlich des Bedienkonzeptes der Software. Ein HMD ist keine übliche Rechner-Komponente, wie Maus oder Tastatur und deshalb typischerweise nicht an jedem Rechner-Arbeitsplatz vorhanden. Um HMDs zu nutzen ist eine leistungsstarke Graphikkarte erforderlich, die an typischen Arbeitsplätzen nicht verfügbar ist.

Betrifft der Nutzer die VR-3D Welt will er sich wie in der realen Welt verhalten, Fortbewegen durch z.B. Laufen, Greifen von Gegenständen durch Einsatz seiner Hände, Auslösen von Tasten durch Drücken der

3D-Tasten, etc.

Hierzu ist ein komplettes Umdenken bei der Gestaltung der Szene erforderlich.

Um die hierfür erforderlichen Konzepte zu entwickeln, wurde das Modell als VR Modell umgestaltet und für den Test mittels Smartphones (Gear) und HMD (Oculus, Vive) vorbereitet.

Im Holodeck werden auch Ausbildungsmedien für die Feuerwehr entwickelt [3]. Einen aktuellen Einblick gibt ein Beitrag des HR [4].

## 13 Ausbildung und 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](http://www.e-brandschutz.de) zu finden sind, siehe auch [1]. Verfügbar sind Kurse für die Brandschutzunterweisung sowie zum Brandschutzhelfer. Bisher wurden mehrere tausend Personen mit Hilfe der e-learning Kurse im Brandschutz unterwiesen.

Aktuell werden die Kurse mit den verschiedenen Simulationskonzepten verknüpft, um das theoretisch Erlernte mit realem Tun zu verknüpfen.

## 14 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 von Strategien, die in der Realität in einem Arbeitsumfeld nicht durchgeführt werden können.

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.

## Acknowledgements

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.

## References

- [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: *Projekt KATIE:Einsatz - Simulation bei der Feuerwehr; Brandschutz Deutsche Feuerwehr-Zeitung, Verlag W. Kohlhammer GmbH, Stuttgart;* (4/16), S. 278 – 281; April 2016.
- [4] Dieter Wloka: *Virtuelle Feuerwehr; TV, HR 3, Hessenschau;* Januar 2017.

# Projektierung mechatronischer Anlagen in der Hochschullehre am Beispiel von Labormodellen

Tobias Loose<sup>1</sup>

<sup>1</sup>Hochschule Heilbronn, Max-Planck-Str. 39, 74081 Heilbronn  
*tobias.loose@hs-heilbronn.de*

Dieser Beitrag zeigt Ideen für eine industriepraktische Interpretation von Laborarbeiten in der Hochschullehre auf. Damit sollen Studierende einen hohen Bezug zu realen mechatronischen Industrieanlagen erhalten. Zudem werden Methoden aus der Industriepraxis demonstriert, um Anlagen zu projektieren, z. B. Bewertung von Maschinen-Eigenfrequenzen. Die Methoden werden anhand relativ einfacher Laborausstattung gezeigt, wie z. B. LEGO® MINDSTORMS®. Die Ausrüstung wird zudem online mit MATLAB® / Simulink® eingebunden. Damit werden Simulationen und Methoden zur Modellbildung mechatronischer Systeme praxisnah angewendet.

## 1 Einleitung

In diesem Beitrag werden Anregungen gegeben, wie mit kostengünstigen Laborausstattungen industriennahe Anwendungen und Interpretationen realisierbar sind. Damit soll eine *Ergänzung* zu klassischen Laborarbeiten aufgezeigt werden.

### 1.1 Klassische Laborarbeit

Es gibt viele unterschiedliche “klassische” Laborarbeiten, wie z. B. Lernfabriken, Simulationen, Versuche über ausgewählte Komponenten / Subsysteme (Messung von Motorkennlinien, Auswertung von Sensordaten, usw.). Schwierigkeiten dabei sind u. a. hohe Anschaffungskosten, hoher Betreuungsaufwand, fehlende mechatronische Gesamtbetrachtung, fehlender Bezug zur Industriepraxis, siehe z. B. [1].

### 1.2 Ziel projektbasierter Laborarbeiten

Bei einer projektbasierten Laborarbeit sollen Studierende selbständig eine mechatronische Anlage planen, projektieren, bauen und in Betrieb nehmen, d. h. ein komplettes Entwicklungsprojekt selbst realisieren. Dabei sollen Vorlesungsinhalte angewendet werden, z. B. Projektmanagement, systematisches Entwickeln bzw. Kreativitätsmethoden, Konstruktionsoptimierung, Auslegung von

Antrieben, Technische Mechanik, Schwingungslehre, Regelungstechnik. Die Anlagen können von Studierenden selbst vorgeschlagen werden oder aus einem “Katalog” gewählt werden, siehe Tabelle 1.

#### Industrieanlagen und Aufgaben

---

Schwingungsdämpfung Regalbediengerät (RBG)

---

Ventilgesteuerter Fluidantrieb, z. B. hydraulische Hebebühne, Anbaugeräte mobiler Arbeitsmaschinen wie Traktoren

---

Spurgeführtes, fahrerloses Transportsystem (FTS) oder autonomer, mobiler Roboter

---

Kran-Pendeldämpfung

---

Aktive Gebäude-Schwingungsdämpfung, z. B. im Taipei Financial Center / Taipeh 101

---

Roboter zur Bestückung einer Werkzeugmaschine

---

Automatisches Einparken mit / ohne Anhänger

---

Regelung Hexapod Plattform mit mehreren Antrieben

---

Gleichlaufregelung, z. B. einer Werkzeugmaschine, oder Hebebühne

---

**Tabelle 1:** Beispiele von Industrieanlagen i. S. eines (unvollständigen) “Katalogs” für eine Laborarbeit

Dabei soll der Fokus auf Methoden gelenkt werden, die in der Industrie von Relevanz sind, z. B. Positions- und Geschwindigkeitsregelung, Gleichlaufregelung,

Teachen von Robotern, Trajektorienplanung, aktive sowie passive Schwingungsdämpfung, Umgang mit Messfehlern und indirekte Messung.

In diesem Beitrag wird beispielhaft eine ventilgesteuerte, fluidtechnische Hebebühne diskutiert, siehe Abbildungen 1 und 2 rechts. Bei der Projektierung fluidtechnischer Anlagen ist u. a. die Analyse von Eigenfrequenzen oder passende Auswahl einer Ventilkennlinie wichtig, was durch Labormodelle ebenso gezeigt werden kann [9].

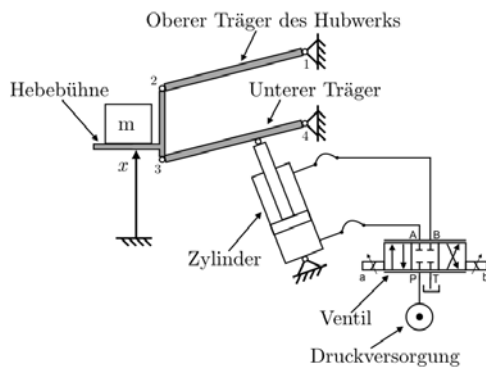


Abbildung 1: Prinzip einer fluidtechnischen Hebebühne [9]

Des Weiteren werden hier Schwingungsprobleme bei Regalbediengeräten (RBG) gezeigt. RBG haben bedingt durch ihre Konstruktion (große bewegte Masse, hoher und damit biegeelastischer Mast) eine niedrige Eigenfrequenz, siehe Abbildungen 2 links und 3. Durch die Verzögerung wird das System zum Schwingen angeregt, so dass an den Ein- und Auslagerungspunkten wertvolle Taktzeit verloren geht. Um dies zu lösen bietet die Regelungstechnik viele Lösungen, siehe [2, 3, 4, 8, 12].

Zudem wird die Modellbildung fahrerloser Transportsysteme (FTS) gezeigt, die auf einer festgelegten Spur geregelt fahren sollen, siehe [15, 16].

## 2 Methoden der mechatronischen Gesamtbetrachtung

In diesem Kapitel werden exemplarisch Projektierungsmethoden aus der Industriepraxis gezeigt, die bei der modellbasierten, mechatronischen Entwicklung von Anlagen angewendet werden. Diese Methoden lassen sich im Rahmen einer Laborarbeit anhand kleiner Modelle diskutieren bzw. sollen durch

Studierende erarbeitet werden. Theoretische Ansätze aus der "klassischen Lehre" sollen pragmatisch-heuristischen Ansätzen gegenüber gestellt werden, wie z. B. die Notwendigkeit der exakten Kenntnis über die Regelstrecke inkl. Parametern. In realen Anlagen sind Regelstrecken nie exakt bekannt oder nicht immer durch lineare, mathematische Modelle zu beschreiben, dennoch ist der Regler passend zu entwerfen. Daher wird bei vielen "Standard-Industrieanwendungen" ein pragmatischer, heuristischer Reglerentwurf gewählt, mit dem Ziel einer akzeptablen Regelgüte.

### 2.1 Kinematische und kinetische Betrachtung zur Antriebsauslegung

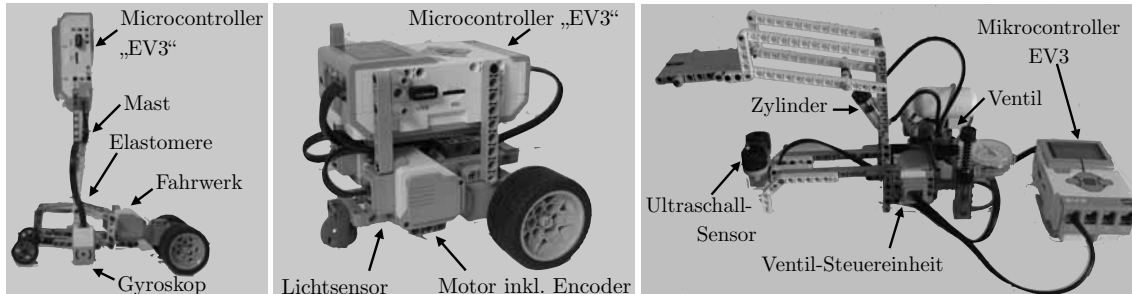
Die Konstruktion einer Anlage ist nach kinematischen und kinetischen Aspekten zu bewerten, ob sie den Anforderungen aus den Kundenvorgaben / dem Lastenheft genügen. Beispielsweise müssen vorgegebene Stellwege oder Geschwindigkeiten erreicht werden, letztere sind idealerweise in einem v-t-Diagramm darzustellen (=Geschwindigkeitsprofil  $v(t)$  über der Zeit  $t$ ). Die Kräfte auf die Anlage — und insbesondere auf den Aktor — müssen ebenso berechnet werden, was hier beispielhaft an einer Hebebühne und einem Fluidaktor (Zylinder) gezeigt wird. Wird die Anlage dabei nicht als Ganzes betrachtet, dann können Projektierungsfehler die Folge sein.

Bei hydraulisch betriebenen Anlagen kommt ggf. zur Vergrößerung der Stellwege ein Hebelarm als Übersetzungsverhältnis zum Einsatz. Dabei sitzt die "starke Hydraulik am kurzen Hebel", siehe  $l_2$  in Abbildung 4. Beispiele für eine derartige Kinematik sind Hubwerke bei Anbaugeräten von Traktoren, Hebebühnen von Lastkraftwagen oder Schwenkzylinder in Walzwerken. Bei einer solchen Konstruktion wirkt der Zylinder als sog. Pendelstütze, was einem einwertigen Lager entspricht. Somit wird eine senkrecht zum Hebel wirkende Lagerreaktion  $F_{Zs}$  in diesem Punkt übertragen, um ein statisches Gleichgewicht mit dem zweiwertigen Lager A zu erreichen.

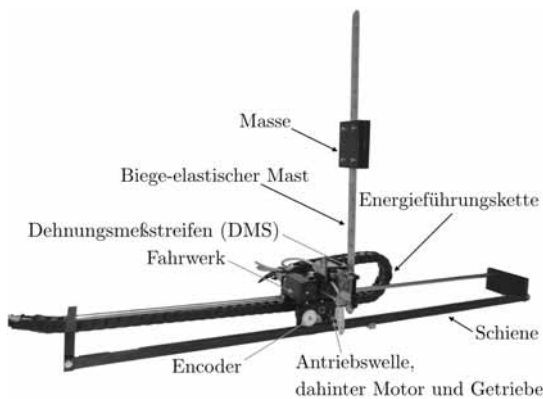
Durch die geometrische Anordnung des Zylinders steht dieser jedoch meistens nicht entlang der Kraft  $F_{Zs}$ , sondern schließt den Winkel  $\beta$  mit

$$\cos(\beta) = \frac{l_3 \cdot \cos(\alpha)}{\sqrt{l_2^2 + l_3^2 - 2 \cdot l_2 \cdot l_3 \cdot \sin(\alpha)}} \quad (1)$$

ein. Dadurch wird der Zylinder sowohl axial als



**Abbildung 2:** Durch LEGO® nachgebildete Industrieanwendungen, Regalbediengerät (links), fahrerloses Transportsystem (Mitte), ventileregeltes Fluidantriebs als Hebebühne (rechts)



**Abbildung 3:** Labormodell eines RBG

auch radial belastet, siehe Abbildung 4. Die radiale Kraftkomponente

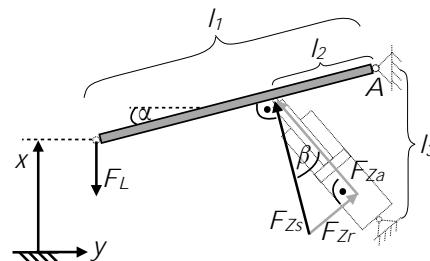
$$F_{Zr} = F_{Zs} \cdot \sin(\beta) = \frac{l_1}{l_2} \cdot \cos(\alpha) \cdot \sin(\beta) F_L \quad (2)$$

bewirkt ein „Verkannten“, was für die Funktion des Zylinders nachteilig ist, z. B. zusätzliche Lager- und einseitige Dichtungsbeanspruchung. Hohe radiale Zylinderkräfte sind in der Industriepraxis zu vermeiden, siehe [5, 9].

Wie aus Gleichung (2) zu erkennen ist, wird die radiale Kraftkomponente  $F_{Zr}$  klein für  $\beta \rightarrow 0$ . Damit kann eine Optimierung der Konstruktion angestoßen werden, so dass z. B. der Zylinder genau dann senkrecht steht, wenn die Last  $F_L$  maximal wird, siehe [9].

## 2.2 Analyse der Eigenfrequenz

Durch die Kompressibilität von Fluiden wirkt im Antriebssystem eine zusätzliche relativ „weiche“



**Abbildung 4:** Exemplarisch gezeigtes Prinzip eines hydraulisch betriebenen Hebels

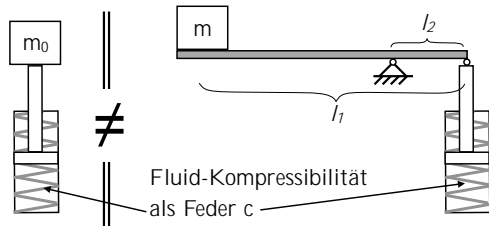
Feder. Stahl ist sehr viel „steifer“ als Öl, d. h. ein Stahl-Elastizitätsmodul ist im Vergleich zu einem Öl-Ersatz-Kompressionsmodul ca. um den Faktor 175 größer. Zwischen Luft in einer Pneumatik- und Öl in einer Hydraulik-Anlage existiert ein weiterer Faktor von ca. 12.000 in Bezug auf die Steifigkeit, siehe [5, 7, 9]. Die durch Kompression hervorgerufene Feder stützt sich auf dem Stellglied ab, z. B. am Stetigventil. Durch diese zusätzliche Elastizität bzw. Feder ist bereits eine gewisse Schwierigkeit bei der Regelung zu erkennen, da zusammen mit einer bewegten Masse  $m$  ein schwingungsfähiges System vorliegt. In diesem Beispiel ist dann die Last  $F_L = m \cdot g + m \cdot \ddot{x}$ .

Ein typischer Projektierungsfehler von Industrieanlagen besteht nun darin, dass ein Übersetzungsverhältnis bzw. hier der Hebelarm  $l_2/l_1$  nicht mit in die Berechnung der Eigenfrequenz mit einbezogen wird. In diesem Beispiel wird die Eigenfrequenz

$$f_{0,\text{Sys}} = \frac{1}{2\pi} \sqrt{\frac{l_2^2 \cdot c}{l_1^2 \cdot m}} \quad (3)$$

des ungedämpften Systems mit der Federsteifigkeit  $c$  durch die Fluid-Kompressibilität berechnet, siehe [9]. Da der Hydraulik-Zylinder „am kurzen Hebel“ an-

greift, reduziert sich die tatsächliche Eigenfrequenz im Vergleich zu einem Direktantrieb ohne Übersetzungsverhältnis, siehe Abbildung 5. Für den

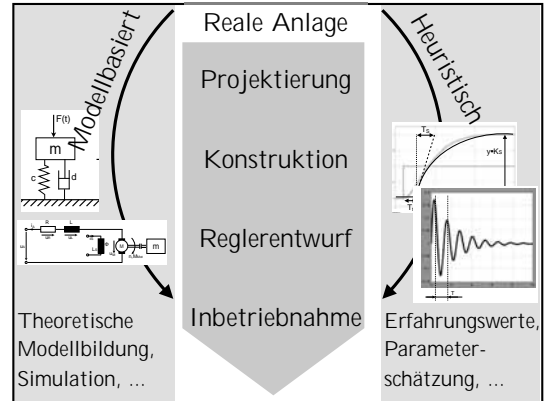


**Abbildung 5:** Vergleich der Eigenfrequenzen zur Schwingungsanalyse von Anlagen als Direktantrieb (links) bzw. als Antrieb mit Übersetzungsverhältnis (rechts)

“Standard-Maschinenbau” hat sich eine minimale Eigenfrequenz von  $f_0 > 15\text{ Hz}$  in der Industriepraxis bewährt. Damit lässt sich die Maschine regelungstechnisch hinreichend gut in Betrieb nehmen, da typische Anregungen unterhalb der Eigenfrequenz statt finden. Bei einer Inbetriebnahme kann das andernfalls zu unerwünschten Schwingungsproblemen führen, die im Vorfeld hätten vermieden werden können, z. B. durch Vergrößerung des Zylinderdurchmessers, um die “Federsteifigkeit”  $c$  zu vergrößern und somit auch die Eigenfrequenz, siehe [9].

## 2.3 Modellbasierter Entwurf

Der industriepraktische Entwurf einer Regelung inkl. Inbetriebnahme der *realen* Anlage lässt sich prinzipiell in zwei Vorgehensweisen untergliedern. Auf der einen Seite wird der Regler durch theoretische Modellbildung bzw. Simulationen, die die physikalischen Gesetze abbilden, entworfen, siehe linke Seite in Abbildung 6. Anschließend werden über Versuche an der realen Anlage ggf. noch Parameter feinjustiert. Diese Vorgehensweise kommt tendenziell im Seriengeschäft zum Einsatz, bei dem eine sehr hohe Regelgüte im Vordergrund steht. Auf der anderen Seite gibt es noch das Projektgeschäft bei dem einmalige Anlagen projektiert und gebaut werden. Die Inbetriebnahme kann ggf. nur teilweise durch Simulationen unterstützt werden. Der Regler wird aber letztlich heuristisch vor Ort mit einer ausreichenden Regelgüte eingestellt. Grundlage für die Einstellung kann noch eine vorgelagerte Parameteridentifikation sein, bei der wesentliche Maschinenparameter ermittelt werden, wie z. B. die Eigenfrequenz oder der Dämpfungsgrad der Maschine, siehe rechte Seite in



**Abbildung 6:** Vorgehen des regelungstechnischen Entwurfs bei realen Industrieanlagen von der Projektierung zur Inbetriebnahme, links: Theoretische Modellbildung durch physikalische Gesetze, rechts: Pragmatischer Entwurf durch Erfahrungswerte

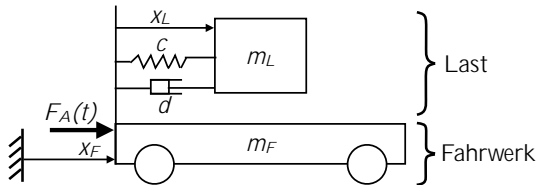
Abbildung 6. Diese beiden Vorgehensweisen lassen sich anhand kleiner Labormodelle sehr schön demonstrieren, so dass auch die jeweiligen Vor- und Nachteile beider Varianten ersichtlich werden, wie nachfolgend exemplarisch gezeigt. In der Industriepraxis wird je nach Anlage eine Kombination aus beiden Varianten angestrebt. Ebenso ist durch die Labormodelle schön zu zeigen, welchen großen positiven Einfluss eine gute Konstruktion und Projektierung auf eine spätere regelungstechnische Inbetriebnahme hat, z. B. indem eine passende Getriebeübersetzung und Eigenfrequenz oder Auflösung und Platzierung eines Sensors ausgewählt wird.

Um Schwingungen eines Regalbediengerätes (RBG) zu dämpfen, kann ein Zustandsregler zum Einsatz kommen, siehe Labormodelle in den Abbildungen 2 links und 3. Zwingende Voraussetzung für den Entwurf eines Zustandsreglers ist eine vorgelagerte Modellbildung. Durch z. B. ein vereinfachtes Modell lässt sich das Differentialgleichungssystem

$$\ddot{x}_F = \frac{1}{m_F} (F_A + d \cdot \dot{x}_L + c \cdot x_L) \quad (4)$$

$$\ddot{x}_L = -\frac{d}{m_L} \cdot \dot{x}_L - \frac{c}{m_L} \cdot x_L - \ddot{x}_F, \quad (5)$$

herleiten, siehe Abbildung 7 und [8]. Dadurch lassen sich die Dynamik bzw. System- sowie Ein- und Ausgangsmatrix im Zustandsraum beschreiben, um die Reglerparameter zu berechnen. Das entspricht dem reinen modellbasierten Vorgehen, siehe linke Seite



**Abbildung 7:** Vereinfachtes Modell eines Regelbediengerätes, siehe [8]

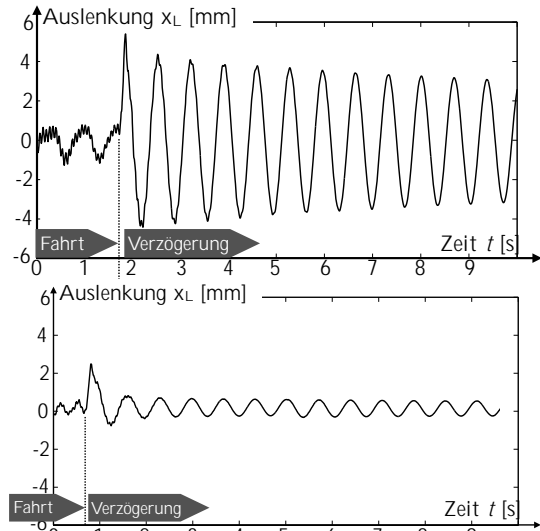
in Abbildung 6 und Ergebnis in Abbildung 8 unten sowie Einzelheiten zur Auslegung in [8]. Auf Basis eines Modells lassen sich auch weitere Reglerentwurfverfahren nutzen, wie z. B. im Zeit- oder Frequenzbereich, siehe [6, 11, 14].

Eine weitere Vorgehensweise besteht in der Vorabauswahl eines möglichen Regelstreckenmodells, um strukturell das Verhalten nachzubilden. Die Parameter werden danach aus Messwerten erfasst. Beispielsweise lassen sich aus den Messwerten des ungedämpften Mastes eines RBG der Dämpfungsgrad  $D$ , die Eigenkreisfrequenz  $\omega_0 = 2 \cdot \pi \cdot f_0$  und der Verstärkungsfaktor  $K$  ermitteln, siehe Abbildung 8 oben, und in der Übertragungsfunktion

$$F_S = \frac{K \cdot \omega_0^2}{s^2 + D \cdot \omega_0 \cdot s + \omega_0^2} \quad (6)$$

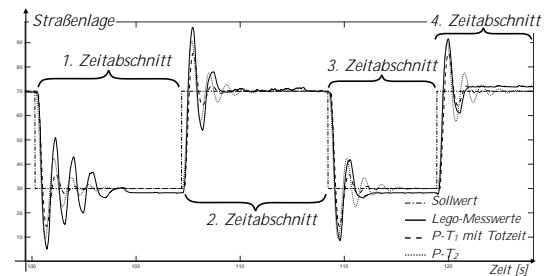
abbilden. Mit diesem Ansatz kann wiederum mit unterschiedlichen Entwurfverfahren ein Regler auslegt werden, wie oben beschrieben. Weitere Entwurfverfahren sind natürlich ebenso einsetzbar, z. B. nach Ziegler-Nichols oder daraus abgeleitete rein "heuristische" Entwürfe vor Ort an den Anlagen.

Diese Gegenüberstellung und Auslegung durch verschiedene Methoden lassen sich sehr schön durch die oben beschriebenen Labormodelle veranschaulichen. Eine Schwierigkeit verlässliche Modelle zu ermitteln besteht allerdings darin, dass die Realität nicht immer durch lineare Gleichungen erfassbar ist oder auch einzelne Parameter nicht immer bekannt sind, wie z. B. Hysterese-Effekte, Reibungskoeffizienten in Abhängigkeit der Temperatur, Spiel, Fertigungstoleranzen. Dazu passt ein berühmtes Zitat von Albert Einstein, auch wenn es aus einem anderen Zusammenhang stammt: *"Insofern sich die Sätze der Mathematik auf die Wirklichkeit beziehen, sind sie nicht sicher, und insofern sie sicher sind, beziehen sie sich nicht auf die Wirklichkeit."*, siehe S. 133 in [13]. Diese "Unsicherheit" kann durch LEGO® MINDSTORMS® Labormodelle gut demonstriert werden. Bei der Mes-



**Abbildung 8:** Messwerte eines RBG-Labormodells, ohne Dämpfung (oben) und mit Zustandsregelung (unten)

sung eines Einschwingvorgangs auf die Straßenlage eines fahrerlosen Transportsystems sind nichtlineare und stochastische Einflüsse gut erkennbar, siehe Abbildungen 2 Mitte und "Lego-Messwerte" in 9. Unter



**Abbildung 9:** Messwerte zur Modellbildung, siehe [10]

gleichen Bedingungen zeigen die Messwerte teilweise unterschiedliche Ergebnisse. Gründe hierfür sind u. a. verhältnismäßig hohe Abtastzeiten des Microcontrollers, Lagerspiel und Toleranzen, Nichtlinearitäten in der Mechanik (hier z. B. durch Gummireifen) sowie Hysterese-Effekte durch die Reibung. Als Laborarbeit werden damit die Schwächen jeder Modellbildung gut demonstriert. Natürlich lässt sich durch rein theoretische Modellbildung anschließend ein Regler entwerfen (linke Spalte in Abbildung 6), was aber in diesem Beispiel sehr aufwändig ist und die Realität dennoch nicht erfasst werden kann, insbesondere durch stochastische Einflüsse der Spiele. Eine weitere

Möglichkeit besteht darin die Messwerte durch Übertragungsfunktionen zu approximieren, hier z. B. mit Gleichung (6) oder

$$F_{S2}(s) = \frac{K_2}{T \cdot s + 1} \cdot e^{-T_i \cdot s}. \quad (7)$$

Mit diesen Modellen lässt sich ein Regler parametrieren, z. B. mit der MATLAB® / Simulink® Funktion "Autotuner". Es kann aber auch gezeigt werden, dass bei derartigen "Standard-Aufgaben" mit einer heuristischen Herangehensweise eine sehr gute Regelgüte erzielt werden kann, siehe [10].

### 3 Zusammenfassung

Dieser Beitrag zeigt industriepraktische Interpretationen von Laborarbeiten. Damit erfahren Studierende einen hohen Praxisbezug inkl. Anwendung an realen mechatronischen Anlagen, z. B. praxisgerechte Auslegung von Reglern, Umgang mit fehlender Kenntnis von Regelstrecken oder Streckenparametern. Zudem wird dabei eine *gesamte* Anlage von der Projektierung bis zur Inbetriebnahme betrachtet, z. B. Einfluss der Konstruktion oder Auflösung und Platzierung eines Sensors auf die spätere Inbetriebnahme.

### References

- [1] A. Behrens, L. Atorf, R. Schwann, B. Neumann, R. Schnitzler, J. Ballé, T. Herold, A. Telle, T. Noll, K. Hameyer, T. Aach. *MATLAB Meets LEGO Mindstorms—A Freshman Introduction Course Into Practical Engineering*. In: IEEE Transactions on Education Vol. 53, Iss. 2. p. 306 - 317, 2010.
- [2] DIN 15350. *Regalbediengeräte, Grundsätze für Stahltragwerke, Berechnungen*. Beuth, Berlin, 1992.
- [3] Deutsches Nationalkomitee der FEM, VDMA. *Berechnungsgrundlagen für Regalbediengeräte Tragwerke*. FEM 9.311, 1978.
- [4] H. Dresig, A. Fidlin. *Schwingungen mechanischer Antriebssysteme*. Springer, Berlin, Heidelberg, 2014.
- [5] D. Findeisen, S. Helduser. *Ölhydraulik: Handbuch der hydraulischen Antriebe und Steuerungen*. Springer Vieweg, Berlin, Heidelberg, 2015.
- [6] O. Föllinger. *Regelungstechnik: Einführung in die Methoden und ihre Anwendung*. VDE Verlag, Berlin, Offenbach, 2013.
- [7] H.-W. Grollius. *Grundlagen der Pneumatik*. Carl Hanser Verlag, München, 2012.
- [8] T. Loose. *Benchmark-Anwendung zur Schwingungsanalyse und -dämpfung von Regalbediengeräten am Beispiel eines Labormodells*. Proc., 25. Workshop Computational Intelligence, Dortmund, p. 127-144, 2015.
- [9] T. Loose. *Projektierung mechatronischer Anlagen als Laborarbeit in der Hochschullehre am Beispiel LEGO® MINDSTORMS®*. Proc., 2. IFToMM D-A-CH Konferenz, Innsbruck, p. 1-8 (Tagungsband auf CD-ROM), 2016.
- [10] T. Loose. *Reglerentwurf mit klassischen und Computational Intelligence Methoden in der Hochschullehre am Beispiel von Modellen*. Proc., 26. Workshop Computational Intelligence, Dortmund, p. 261-280, 2016.
- [11] W. Oppelt. *Kleines Handbuch technischer Regelvorgänge*. Verlag Chemie, Weinheim / Bergstr., 1972.
- [12] M. Schipplick und W. A. Günthner. *Potentiale einer flachheitsbasierten Antriebssteuerung am Beispiel Regalbediengerät*. In: at – Automatisierungstechnik Band 59, Heft 4. p. 242–247. 2011.
- [13] C. Seelig (Hrsg.), A. Einstein. *Mein Weltbild*. Ullstein, 2015.
- [14] D. Schröder. *Elektrische Antriebe – Regelung von Antriebssystemen*. Springer, Berlin, Heidelberg, 2014.
- [15] G. Ullrich. *Fahrerlose Transportsysteme*. Springer, Wiesbaden, 2014.
- [16] VDI 2510. *Fahrerlose Transportsysteme*. Beuth, Berlin, 2005.

# Simulationsgestützte Schwingungsstabilisierung hydraulischer Systeme

Mathias Niebergall<sup>1</sup>, Rudolf Mönkle<sup>1</sup>

<sup>1</sup>Hochschule Ulm, Institut für Fahrzeugsystemtechnik

Kompetenzzentrum Hydraulik an der Hochschule Ulm

*niebergall@hs-ulm.de*

Wesentliches Ziel simulationsgestützter Entwicklung technischer Systeme ist es, kosten- und zeitintensiven experimentellen Aufwand zu reduzieren. Hydraulische Systeme zeigen häufig ein instabiles dynamisches Verhalten, das meist erst während der Inbetriebnahme sehr zeit- und kostenintensiv stabilisiert werden kann. Dynamisches Systemverhalten hängt ab von steifigkeits-, dämpfungs- und trägheitsbedingten Wechselwirkungen der beteiligten Komponenten. Im Rahmen der hier vorgestellten Forschungsaufgabe wird eine Methode zur simulationsgestützten, systematischen Schwingungsstabilisierung am Beispiel eines hydraulischen Steuerungssystems entwickelt. Aufbauend auf einer sorgfältigen Modellbildung des hydraulischen Systems wird die systemtypische hydraulische Schwingungscharakteristik mittels Simulationsrechnungen und Eigenwert-/Eigenformanalysen transparent gemacht. Dabei werden die dominanten schwingungssensitiven Systemkomponenten ermittelt und dort gezielt das Gesamtsystem durch konstruktive Änderungen stabilisiert.

## 1 Hydraulisches Steuerungssystem

Die Methode zur Schwingungsstabilisierung wird beispielhaft an dem in Abbildung 1 dargestellten hydraulischen System zur Ansteuerung einer Kupplung für ein Fahrzeuggetriebe vereinfacht dargestellt. Der über eine Pumpe erzeugte Volumenstrom wird hier, über Druckventile geregelt, einer Kupplung zugeführt.

Die dynamischen Wechselwirkungen der beteiligten Komponenten werden im Wesentlichen von folgenden Größen bestimmt:

- Federsteifigkeiten ( $c$ ), Kompressibilitätsmodul ( $K$ ) des Fluids, Dichte ( $\rho$ ) des Fluids
- Dämpfung ( $d$ ) durch Reibung, Viskosität sowie Spalte und Blenden sowie von
- Massenträgheiten der beweglichen Massen ( $m$ ) und des Fluids

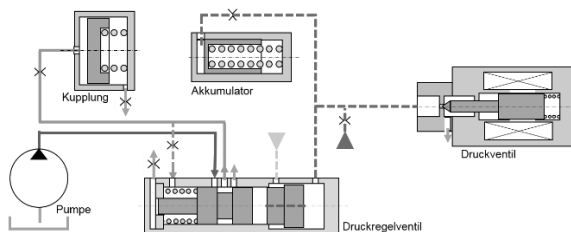


Abbildung 1. Hydraulisches Steuerungssystem

## 2 Modellbildung und Abbildung des hydraulisch instabilen Verhaltens

Im ersten Schritt wird das betrachtete System mittels eines geeigneten Simulationstools als mathematisches Modell abgebildet. Idealerweise werden dabei die dynamischen Eigenschaften des Simulationsmodells mit bereits gemessenen Daten validiert und ggf. angepasst.

Die mathematische Beschreibung der dynamischen Eigenschaften des Systems beruht im Wesentlichen auf den folgenden vereinfachten Modellgleichungen, die aus einschlägiger Fachliteratur, z.B. [1], [2], entnommen bzw. abgeleitet werden können:

a) Kontinuitätsgleichung (1) einer kompressiblen Strömung durch einen Behälter variablen Volumens (hier in der Raumrichtung  $x$ )

$$\rho \cdot A \cdot \dot{x} + \frac{\rho}{K} \cdot \dot{p} \cdot A \cdot x + \rho \cdot \sum_{i=1}^n q_i = 0 \quad (1)$$

Hierin sind  $A$  der Querschnitt des Behälters,  $p$  der Druck im Behälter und  $q_i$  ein Volumenstrom über die Behältergrenze.

b) Bewegungsgleichung (2)

$$m \cdot \ddot{x} + d \cdot \dot{x} + c \cdot x = \sum_{j=1}^m F_j \quad (2)$$

Hierin ist  $F_j$  eine an der beweglichen Masse angreifende Kraft, z.B. aufgrund eines Drucks.

c) Blendengleichung

$$q \cong \alpha_D \cdot A_0 \cdot \sqrt{\frac{2}{\rho} \cdot \Delta p} \quad (3)$$

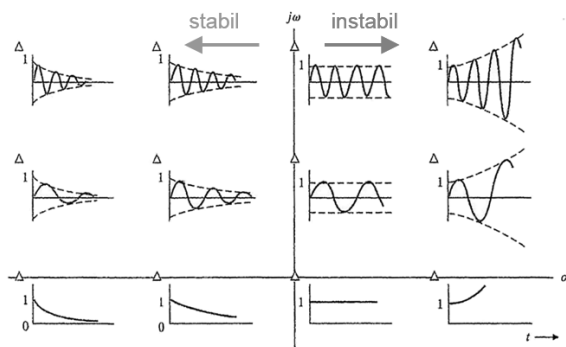
Hierin sind  $\alpha_D$  der Durchflusskoeffizient der Blende,  $A_0$  der Durchflussquerschnitt,  $\Delta p$  die Druckdifferenz über der Blende und  $q$  der Volumenstrom durch die Blende.

Die simulierten Zeitsignale, beispielsweise des Drucks in der Kupplung, repräsentieren den vollständigen dynamischen, meist nichtlinearen Informationsgehalt des Simulationsmodells. Eine Zerlegung dieser Signale in ihre Frequenzanteile (FFT) zeigt die dominanten Frequenzen der auftretenden Schwingungen.

### 3 Eigenwertanalyse

Neben der Darstellung der Simulationsergebnisse im Zeitbereich können auch die Eigenwerte des Schwingungssystems berechnet werden, was eine Linearisierung des Modells in einem geeigneten Arbeitspunkt voraussetzt.

Die Eigenwerte beschreiben sehr anschaulich geometrisch den dynamischen Charakter des Systems. Die Lage der Eigenwerte in der komplexen Ebene zeigt insbesondere deren Eigenfrequenz, Dämpfung, Abklingzeitkonstante und Überschwingverhalten (vgl. Abbildung 2).



**Abbildung 2.** Systemdynamik und Eigenwertposition (abgewandelt aus [3])

### 4 Systemstabilisierung

Werden die bereits aus den Zeitsignalen bekannten dominanten Frequenzen durch konjugiert komplexe Eigenwertpaare bestätigt, können mithilfe einer Eigenformanalyse die schwingungssensitiven Berei-

che des Systems ermittelt und systematische Maßnahmen zur Systemstabilisierung durch geeignete konstruktive Änderungen eingeleitet werden.

### 5 Referenzen

- [1] D. Will und N. Gebhardt (Hrsg.). *Hydraulik*. Springer Vieweg, Berlin Heidelberg, Deutschland, 2014.
- [2] D. Findeisen und S. Helduser. *Ölhydraulik*. Springer Vieweg, Berlin Heidelberg, Deutschland, 2015.
- [3] R.C. Dorf und R.H. Bishop. *Modern Control Systems*. Prentice Hall, USA, 2011.

# Modelling of friction in mechatronics systems

Dr.Sergey Petkun

Brose Fahrzeugteile GmbH & Co. Kommanditgesellschaft, Bamberg

*Sergey.Petkun@Brose.com*

The mechatronics systems are tight connected with a mechanical movement. The losses are present always in reality during it. The most important losses are the friction. The friction modelling both in physics and in mathematics are devoted the great attention. It helps to understand the physical processes. In addition, it makes essential mathematical contribution in theory of discontinuous function. However, there is some problem for usage such models in modelling. The parameters of these physical or mathematical models are very often not available during modelling process. The discontinuities bring bad convergence on simulation. The requirement from mechatronics is very simple and namely is to optimize the consisting components of whole system to each other. The optimization criteria can be contradictory especially under strong economic pressure. That is why the importance of setting the simple behaviour frictional models are very great. Two essential questions arise by development of such models. First deals with physical effects, which reflect, in basic structure of friction model. Second question occurs in connection with content of every model in set. The set is better to make as little as possible on the one hand. On the other hand, the set should cover the usual industrial needs. The Amontons-Coulomb friction laws build the base for the behaviour of the proposed models. The main difficulty on the modelling the friction is that mathematical treatment of their laws, because the mathematical formulation for friction force has form of inequality. The most frequent mechanical cases contain the aim for the suggested set. The friction model set was developed in VHDL AMS what does not restrict the similar implementation in other modelling languages.

## 1 Introduction

The friction is one of the most complex phenomena in mechanics. It is connected with principal description of the friction on behavioural level through its action. There is a wide spreading mistake that the friction plays only negative role in our life. At the first glance it will be thinking about losses due to friction. Nevertheless, it is forgotten about dynamical and static stability, which is possible only due to friction. The friction from physical point of view is nothing else as contact problem between at least two participants. Not only the physical properties define the acting friction forces but also the dynamical conditions make the enormous contribution in full result. Depending on chosen criterion, media and studying conditions there are different types of it: dry, damping, viscous, aerodynamic, static, and dynamic and so on. The determination of art the friction depends also on scientific domain. The viscous one, used in fluid

or gas dynamics, has the same mathematical meaning like damping in dynamics of solids. By that, the linear dependence rubbing velocity is assumed between counterparts. Sometimes in mechanics of solid under viscous friction is understood the continuity by crossing over the zero velocity.

Very interesting feature of friction is ability to transform. By means of it, the mechanical energy can be transform in heat and at the same time the constructive form of both or one participant can be changed. Due to the friction is possible to transform a translational movement in rotational one. As physical consequences of its action are as positive also negative. Wear, corrosion, overheating can be ordered to negative side, but there are also positive effects - breaking ability, burnishing, invention of fire.

The situation with friction in mathematics is not essential better. The classical mathematics become so powerful thanks differential calculus where continuity and differentiability, which are built the basic of

it. The direction of friction forces is always aligned against velocity. That means by the changing of velocity direction (crossing over zero point) the friction force changes its direction. Only single point (zero velocity) makes the friction force analysis enormous complex. All mathematical difficulties go over in simulation field also. The classical laws of friction reflect principal behaviour of it. The simple formulations of this laws can be found in [1]. On the modelling of friction, the focus is not the on complex physical phenomena of it, but rather on proper behaviour by the movement. Only one-dimensional movement along the generalized coordinate will be discussing in this paper. The important role by that plays reasonable implementation of models by means of available constructions of modelling language. The science and engineering study on friction is named the tribology.

## 2 Dynamical effects by friction

In the dynamics, the friction is nothing else like some force. It is a little bit tautological because the force itself is interaction between two objects according to determination in classical mechanics. The force term is more common as friction: by the friction is always meant the force (moment), but not arbitrary force is friction. The force is usually represented like vector in space on some acting line. The distributed frictional action is usually summarized in some concentrated force. A force vector has a little difference from free vector. The action line of force vector is fixed in space in as distinct from free vector. That means the force is defined with three parameters: magnitude, direction and action line. For modelling, the friction only magnitude remains clear to define, because it directed against velocity. The action line is not so important by one-dimensional analysis. The transition from distributed interactions to concentrated forces is out consideration in this paper. It is very important that the resulting effects in mechanics consist not only from resultant force but also from moment (torque). The moment in rotational movement is an analog of force in translation one. From Coulomb friction model, which reflects reality relative good,

$$F_{fric} \leq \mu_{fric} \cdot N$$

where  $F_{fric}$  - maximal friction force before the movement is starting,  $\mu_{fric}$  - friction coefficient, dependent

on the contacting surfaces,  $N$  -normal force between both bodies (mutual pressure).

The Coulomb model is pure experimental one [2]. That is why important to understand that friction coefficients could be obtained only from measurement, but do not calculated from material properties. In contrast that, normal force can be gained from calculation on the base of balances for forces and moments. From the law follows that friction value of force can be calculate during the movement besides the velocity zero - stop condition. Next peculiarity is that input in motionless state and output from it are not the same. This is described by means of static and kinetic coefficients. There is the difference between them almost always. Yet another question arises about velocity zero (rest state). Absolute value like zero exists only in mathematics as some abstraction. In calculus the zero-problem has been understood and solved with introduction small  $\varepsilon$ -neighborhood. The same problem exists in real life if the vibration, existing always and everywhere, are taken into consideration. This consideration applying to simulation leads to introduction the term of velocity threshold for motionless state.

The friction by modelling is usually interpreted like force source. At least two points are needed for its modeling what follows from their definition as interaction between a pair. Yet another remark to general consideration about friction deals with mechanical constraints. In mechanics under constraints are thought the conditions on a connection of elements but not only like geometrical joining. This is the simplest type of constraints. The friction, as connection way, belongs to dynamic constraints what makes the analysis of possible simulation friction models yet more difficult. In reality, the dynamical constraints are more often occurred than geometrical. Gravitational or electromagnetic fields are great examples for it. Now the requirements on behaviour model can be summarized in form different tasks (questions) like that

Choice of motionless limits (velocity thresholds)

Transition from motion to stoppage state

Behaviour in motionless state

Beginning of the movement (startup)

Behaviour description for motion phase

Variability of normal force

Anisotropy of frictional action

The algorithmic oscillations can arise by crossing over the value for threshold velocities, because the static friction (stiction) is not equal of kinematic (kinetic) one. It depends on quality the solver's algorithms of simulator.

### 3 In the neighborhood of zero

Classical Coulomb friction force looks like at Figure 1(solid). To overcome the undefined zero

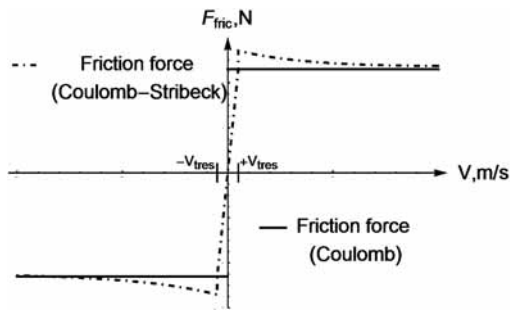


Figure 1: Friction models

range, the linear connection between positive and negative velocities is often used. At the same time, the difference was introduced between static and kinetic forces.

After that, the huge number of friction models [3] look like at the Figure 1(dot-dashed). This model (dot-dashed) is pseudo Coulomb one due to that fact that in his law there is no information about force value by velocity in zero value. The necessary step for complete determination of dry friction is to describe the transition from static value to kinetic one. For this purpose, the Stribeck method is used for example in form exponential transition [4]. This model is very pretty for application: for every value of velocity can be found the suitable friction force value. The only problem consists in inconsistency between zero velocity and force. According to this representation of friction, it follows that the arbitrary force causes at once some movement what does not have place in reality. The next representation looks like more realistic, but it is not enough for complete mathematical definition and hence for simulation modelling because it is not defined behaviour in zero point. It can be make better with introduction velocity threshold like  $\pm V$  at the

Figure 1. There are two areas of velocities with different definitions. If velocity is out of zero areas, there is precise definition for force according physical laws (Coulomb with or without Stribeck addition). The definition for force in rest are is based on requirement for velocity to remain in zero area if the necessary force does not exceed the stiction force. The needed force for motionless can be determined only from statical equilibrium state. It is undefined which magnitude will be reached friction force but if it is less than stiction force the velocity remains in zero area, i.e. motionless state. Such model is very often used as dry friction model where the classical Coulomb model is combined with Stribeck transition.

### 4 Effects included in model

According to definition the friction like some physical behaviour in the nature [5], it is necessary clear to determine the effects are to embed in model. Some compromise should be made between many possible interesting effects and ability to applicate them in this model. Almost every included effect brings with it also additional parameters, which describe it. At the end, this model is very difficult proper to parametrize because of the big numbers of parameters. The basic friction model should describe at least the different behaviour on the motion phase and in motionless state. The difference between stiction and kinetic friction is also obligatory. The influences of small and big velocities can be included but not obligatory. The anisotropy in motion direction can be switch off if the left and right coefficients have the same value. The most simple friction model needs the static, kinetic coefficients, magnitude of normal friction and definition of motionless state, i.e. velocity thresholds. The information about acting forces during the motionless phase are needed for proper simulation. The minimal parameter set is to include the following values  $\mu_{stic}^{\pm}$ ,  $\mu_{kin}^{\pm}$  - stiction and kinetic coefficients,  $V_{tres}^{\pm}$  - threshold for motionless state,  $N$  - normal force. If the assumption of isotropy of movement is valid that means the values with index  $\pm$  are equal to each other. As a matter of convenience in the following, it is assumed isotropic movement: all parameters with  $+$  are equal to  $-$ . Taking into account the pairwise nature of friction, the two pins (inputs) are obligatory.

## 5 Translational friction models

The question, if the model describes the real behaviour more or less satisfactory, is not answerable before the comparison criteria with real behaviour are not precise defined. There is not proper suitable simple analytical model for the comparison with simulation results. The classical analytical model of mass-spring system with friction uses damping and is never stopped according to their solution. The velocity on stopping phase is very small and do not have almost any friction action by damping. However, the dry friction plays the essential role by the real stop. The estimations of the stop time are given in the [6]. The simple simulation test on example mass-spring model was made with different values of velocity thresholds ( $V_{tres}$ ), with different precisions for simulation for testing of stability the results. The results remain the same (stable solution) in wide range of tolerances for velocity thresholds and calculation precision. The units are in MKS

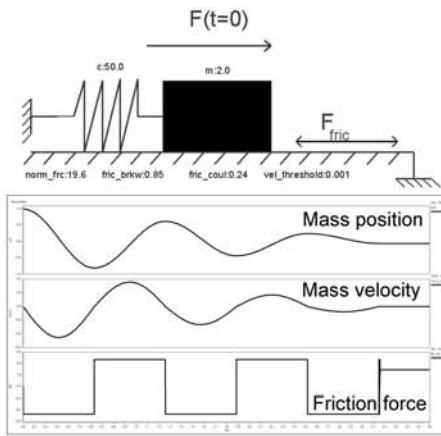


Figure 2: Mass-spring under dry friction

system in all examples. The simulation begins on time  $t = zero$  the spring with stiffness  $c = 50$  which strained with force  $F = 50$ . In time  $t = 0.001$  the initial force  $F$  is vanished. With it, the mass begin to oscillate under action of elastic spring and breaking with dry friction. The system has the next parameters  $m = 2.0$ ,  $\mu_{stic} = 0.35$ ,  $\mu_{kin} = 0.24$ ,  $N = 19.6$ ,  $V_{tres} = 10^{-7}$ . The result of simulation is a transient stopping process giving also the stop time  $t_{stop} = 3.14s$ . The plausibility of model behaviour has not last place on it. Some results are presented on Figure 2. There are no essential differences in results when the threshold

$V_{tres} = 10^{-7}$  is increased to  $V_{tres} = 10^{-3}$ . The simple mass-spring model with dry friction represent some oscillation of the mass-spring at horizontal surface with friction. Normal force was set manually in friction model as parameter:  $N = m * g$ , where  $g$ - gravity acceleration. From this example, the next extension of friction model is wanted. The great advantage of previous model is the simplicity its parametrization. More complex models can be found in [7]. The disadvantage of this friction model is bad connection of components with each other: if the mass is changed in mass model, the normal force is to change also, but normal force is only some parameter in friction model. The better solution were to add a pin as external input for normal force. For example if the normal force or pressure between the rubbing bodies is known from some measurement.

The next friction model is extended for some external input. The input for normal force no arbitrary function, because normal force is some pressure, which is to be always positive. A test for positive values should be performed inside of friction model or "normal force as external function" has to restrict only on positive range. In VHDL AMS, quantity is determined as pure function without physical energy flow. The practical realization of such situation can be represented like the oscillations the mass-spring system at horizontal surface with the vertical perturbations. The mass oscillates on horizontal plane. The vertical perturbations have influence only on pressure the mass at the plane and hence on normal force. With introduction input for normal force, one parameter vanishes from friction model and appears as quantity function. A practical example for physical realization of this case can be imagined like Figure 3 (top). The all parameters are taken from previous example with addition to sinusoidal force  $P(t) = Sin(Amp * t)$ , where  $Amp = 30$  is amplitude of perturbations. The zero friction phases, which correspond the negative part of sinusoidal perturbations, can be seen at the Figure 3 (bottom). Next step for extension the model is to transfer the quantity in some physical input. It is also a three-pins model, but all pins are already physical pins. That means the energy balance in model must be valid. The main difference between these both three-pins model consists in balance of energy for whole system. In previous case (normal force as quantity function), the whole system energy is change for exporting or importing energy from/to some external source.

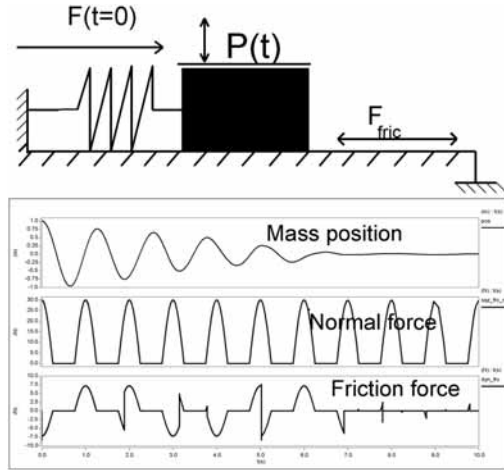


Figure 3: External variable normal force

Another three-pins model redistributes only the internal energy of whole system. The additional pa-

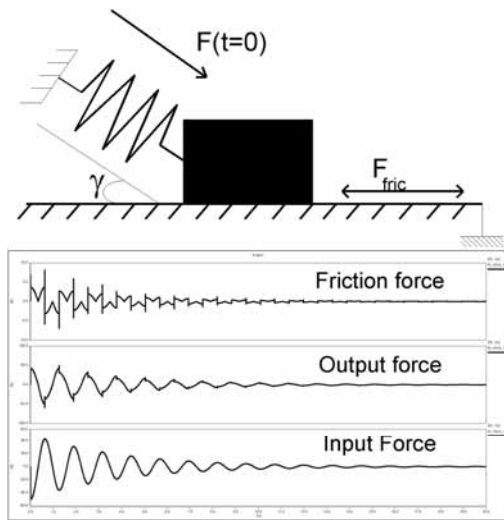


Figure 4: Normal force dependable from input

parameter has to include in the model: which part of inputs force has to be transferred into normal force:  $norm\_ratio = \sin(\gamma)$ . The simulation result is shown at the Figure 4. The parameter  $norm\_ratio$  was set to 0.8 what corresponds  $\gamma = 53.13^\circ$ . With replacing the forces to moments  $F \mapsto M$  and velocity  $v \mapsto \omega$ , it will be the analog description the friction in the rotational movement. If it is possible, the number of parameter should be held as less as possible. With setting param-

eter to zero value, it is possible to switch off the influence of some types of friction. The combination of the different friction models occurs from wish to cover the most frequent cases by the modeling of mechatronics system. The mechatronics devices needs the mechanical energy for its movement. The most energy comes by means of electrical motors. Their rotational velocities vary in very wide ranges. The main role by low velocity range plays Coulomb friction (dry friction). The contribution of viscous friction comes at first place during fast rotation. If the rotation speed comes over some limit, it is already necessary to take into account also aerodynamic part. Start-stop process demands the proper transfer from motion to stopping. There are big number of effects involved in this process. It is possible all time to improve the model to other fine effects [8]. Nevertheless, every further effects is connected with introduction the next series of parameters what makes model almost usefulness because its proper parameterization will be fast impossible or very difficult because of many unknown parameters. The most of usual situation can be simulated with proposal three models.

How it is already mentioned, the typical energy supplier in system is the electrical motor. This requires the same friction model but in rotational domain. The straightforward transfer is not so simple because for the normal force is not direct transferable in normal torque. That is why the rotational friction with variable pressure contact has not only rotational inputs but also translational one. When the translational or rotational movements are mentioned, it is very important to understand that is meant movement along some generalized coordinate but not only like strait line movement (geometrical dimension). At the schema, the proposal set is shown at the Figure 5. The model 2 contains the model 1 as particular case. The model 1 is very useful because of the frequency of usage this model is very high. The model 1 is the same as 2 if the quantity normal force is constant value. However, it has influence on speed of simulation. The number of solving equations are different for model 1 and 2. As result of that is the increasing of simulation time for model 2. Models 1 and 3 are used usually if the friction force (torque) is more or less constant, what means the pressure between contacting bodies doesn't change essentially during the movement. The model 2 and 5 are recommended to use if the rubbing pressure is changing and information about its dynamics is

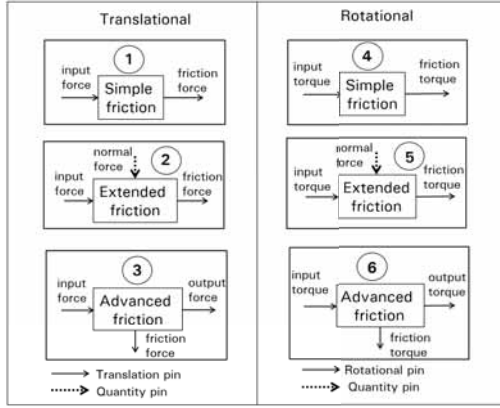


Figure 5: Set of friction models

available. This model is to use very careful because of the losses due to friction in this case are determined by user itself. The models 3 and 4 correspond the case if the friction is implicit dependable from input load. There is the difficulty to estimate of part of energy transferring in losses. The help for this estimation can be output, as information source, which can be better measure or estimate. All what is lost was going down in friction. It is relative crude estimation but for judgment about dynamical behaviour is good enough. Despite of simplicity of description of functional properties of basic model, the implementation of such requires taking into account the particular peculiarities of simulation system and modeling language. The robust of model is to proof on capability of multiply usages of models. The result should be good stable regard to simulation accuracy. The wish for a symmetry in usage of pins is not easy technical to realize despite of their physical equivalence.

## 6 Pendulum under gravity

As simulation example was chosen the classical model of the oscillating pendulum about some horizontal axis under action of gravitational force. The model with its parameter is shown at the Figure 6. All parameters are self explaining. The effective mass of pendulum is concentrated at the end of massless non-deformable rod. By means the simulation, the question "How many oscillations for given parameters will make the pendulum till full stop?" is to answer. The

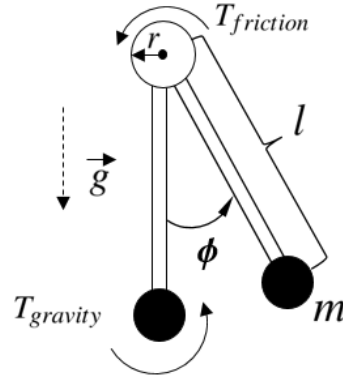


Figure 6: Pendulum under gravity with friction

assumption about smallness the oscillation angle does not take a place. The normal force for calculation the friction torque is equal of tension of rod. There are two component, influenced on the rod: part of gravity and centrifugal forces of rotating mass. The mass is assumed as point one. After some transformation, the differential equation of movement is easy to get for given parameters:

where  $\dot{\phi}$  - angular velocity of pendulum,

$\ddot{\phi}$  - angular acceleration of it

$Sign$  - signum function.

$$\ddot{\phi} = -\frac{g}{l} \cdot \sin(\phi) - Sign(\dot{\phi}) \cdot \frac{\mu \cdot r}{l^2} (g \cdot \cos(\phi) + l \cdot \dot{\phi}^2)$$

There is no exact analytical solution of this differential equation. It can be solved only numerical. For using the simulation models, the problem is to reformulate using existing models from mechanical libraries. For the rotating mass, there is the model like moment of inertia. For both torques, it needs to develop the torque sources. Some additional model is required for realization of term  $-\frac{g}{l} \cdot \sin(\phi)$ . The normal force  $\frac{\mu}{l^2} (g \cdot \cos(\phi) + l \cdot \dot{\phi}^2)$  is calculated in this model and is used as input information for friction model. The model with number 5 from Figure 5 is proper suitable for this case. The simulation structure is shown at Figure 7 (top). The parameter was set to next values:  $m = 1$ ,  $l = 1$ ,  $g = 9.81$ ,  $r \cdot \mu_{stat} = 0.226$ ,  $r \cdot \mu_{kin} = 0.2$ ,  $c_s = 100$ ,  $v_{tres} = 10^{-5}$ . The viscous and aerodynamic part were switched off. Initial deflection angle  $\phi_0 = 0.7rad = 40.107^\circ$ . The simulator solution is shown at the Figure 7 (bottom). The simulating result seems to be plausible. The number of oscillation is easy to count, which for this input is equal 8. It is use-

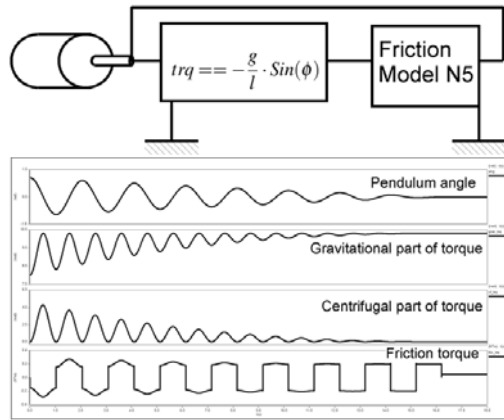


Figure 7: Simulation of the pendulum oscillation

ful take a look a transient behaviour of both components, giving contribution in normal force and therefore in friction moment. It also interesting, that the visual analysis of transient position gives almost no information about force processes behind it.

## 7 Conclusion

The modeling of friction is very strong dependent from problem formulation. The focus at friction modeling lays on dynamics for mechatronics. The interests the physics, mathematics and computer science are faced in this modeling art together. The modelling languages like VHDL AMS or Modelica were created for modeling of dynamics in mechatronics. The standardization of terminology of generic models can accelerate the modelling process and force model content more undependable from modeling languages. The choice of model membership in generic set is not simple task, because of the wide range of application field. The proposal for friction set is arisen from analysis the practical situations, occurring very often in daily life. The simulation results shows that our knowledge about forces especially friction in dynamics is very restrict up to now. The simulation result of oscillating pendulum leads to next question: Is the simulation result correct? To answer at this question the validation and verification plays the important role and also in future by the development of basic models. The offering set is only first trying to build some "alphabet" by means of the mechatronics can describe

their stories. The creation of such generic set is common task of the simulator manufacture and simulation language development in cooperation with physicians and mathematicians together with industrial engineering. The quality, robustness and simplicity are required from basic generic unit. By simplicity is meant not simple implementation but rather clear physical content and as easy as possible its parametrization.

## References

- [1] Wikipedia *Friction*  
<https://en.wikipedia.org/wiki/Friction>
- [2] Coulomb C. A., *Théorie des machines simples, en ayant égard au frottement de leurs parties, et la roideur des cordages*. In Mém.Math. Phys., vol. X, pp. 161–342, 1785.
- [3] Olsson H., Åström K.J., Candus de Win C., Gävert m., Lischinsky P. *Friction models and friction compensation* IEEE Transactions on Automatic Control, 40:3, 1995
- [4] Stribeck R. *Die Wesentlichen Eigenschaften der Gleit- und Rollenlager* Z. Verein. Deut. Ing. Vol. 46 Seite 38ff. 1341–1348, 1902
- [5] Armstrong-Hélouvy B., Dupont P., Canudas de Wit C. *A Survey of Models, Analysis Tool and Compensation Methods for Control of Mashines with Friction* Automatica, Vol30, No.7, pp1083–1138, 1994
- [6] Marchewka A., David S. A., Robert J. B. *Oscillator damped by a constant-magnitude friction force*. Am. J. Phys., Vol. 72, No. 4, pp.477–483, 2004
- [7] Berger E. *Friction modeling for dynamic system simulation* journal Appl.Mech Vol.55, No.6 pp.535–577, 2002
- [8] Karnopp. D. *Computer simulation of slipstick friction in mechanical dynamic systems* Journal of Dynamic Systems, Measurement and Control, 107:1, pp. 100–103, 1985.

# Bestimmung unbekannter Simulationsparameter durch eine mehrstufige Monte-Carlo-Simulation unter Einbeziehung gemessener Daten eines Referenzmanövers

Univ.-Prof. Dr.-Ing. Dr. h. c. Albert Albers<sup>1</sup>, Jan-Michael Veith<sup>1</sup>, Dr.-Ing. Matthias Behrendt<sup>1</sup>,  
Dr.-Ing. Achim Romer<sup>2</sup>

<sup>1</sup>Institut für Produktentwicklung, Kaiserstr. 10, 76131 Karlsruhe

<sup>2</sup>Dr. Ing. h.c. F. Porsche AG, Porschesstraße 911, 71287 Weissach

*jan-michael.veith@porsche.de*

Eine Simulation basiert auf der vereinfachten Abbildung der Realität in einem Modell in Abhängigkeit der Zielstellung der Simulation. Bei der Simulation eines bekannten Manövers können Teile des Antriebstrangs vereinfacht abgebildet werden – beispielweise kann die Abbildung eines Schaltvorgangs entfallen, wenn die (bezüglich der betrachteten Belastungsgröße) relevanten Teile eines Manövers nur in einem Gang durchfahren werden. Durch die Modellbildung kann das Modell Parameter enthalten, welche unbekannt sind und nicht mit physikalischen Größen eines Fahrzeugs übereinstimmen. Üblicherweise werden diese unbekannten Simulationsparameter manuell eingestellt, bis das Modell das gewünschte Verhalten zeigt, analog zu den Messdaten einer Referenzmessung. Das vollständig parametrisierte Modell kann nun beispielsweise zur systematischen Untersuchung der Abhängigkeit auftretender Belastungen von Versuchsrandbedingungen genutzt werden.

Dieser Beitrag zeigt, wie unbekannte Simulationsparameter durch eine mehrstufige Monte-Carlo-Simulation systematisch ermittelt werden können. Das Simulationsmodell basiert auf dem IPEK-X-in-the-Loop-Ansatz bei dem durchgängig die Interaktion von Fahrer, Fahrzeug und Umwelt unter realitätsgetreuen Bedingungen sichergestellt wird. Zur Ermittlung der unbekannten Simulationsparameter werden diese hinsichtlich zu definierender Belastungsgrößen eines Referenzmanövers optimiert. Das Vorgehen wird anhand eines konkreten Beispielmanövers aufgezeigt. Die Automatisierung ermöglicht eine zeitoptimierte Modellparametrisierung und steigert die Qualität der Simulationsergebnisse.

## 1 Einleitung

Innerhalb eines simulierten Manövers können kritische Belastungszustände einzelner Bauteile eines Fahrzeug-Antriebstrangs ermittelt werden [1]. Abhängig von der betrachteten Belastungsgröße wird das Simulationsmodell in der dafür notwendigen Detailierungstiefe aufgebaut, um eine effiziente Nutzung der vorhandenen Ressourcen darzustellen. Unbekannte Simulationsparameter (oder Parameter deren exakte Berechnung bzw. Ermittlung nicht möglich oder zu aufwändig wäre) sind beispielsweise die Parametrisierung eines Totzeit-Glieds oder ein Dämpfungsfaktor. Häufig werden diese Parameter aus Erfahrungswerten abgeschätzt. Hierdurch wird meist nicht der optimale Parametersatz gefunden. Insbesondere bei komplexen Simulationsmodellen mit vielen unbekannten Parametern ist die Ermittlung dieser Parameter ein aufwändiger und zeitintensiver Prozess. Im Folgenden wird gezeigt, wie anhand einer mehrstufigen Monte-Carlo-Si-

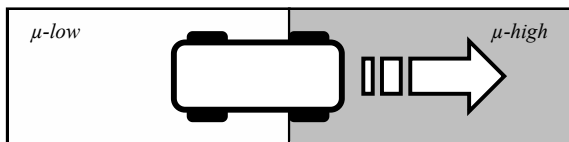
mulation (MCS) unter Einbeziehung von Fahrzeugmessdaten des betrachteten Manövers eine automatisierte Bestimmung der unbekannten Simulationsparameter erfolgen kann.

## 2 Getriebeerprobung am Beispiel des Manövers „Abfahrt $\mu$ -Sprung“

Ein Teil des Entwicklungsprozess eines Fahrzeugs ist die Getriebeerprobung [2]. Bei der Funktionserprobung werden Grundfunktionalitäten, wie die Schaltbarkeit, sichergestellt. Die Dauererprobung stellt das Erreichen einer Mindest-Lebensdauer sicher. Fahrmanöver, welche durch unsachgemäßen Gebrauch sehr hohe Belastungen in kurzer Zeit erzeugen, werden „Missbrauchsmanöver“ genannt [3].

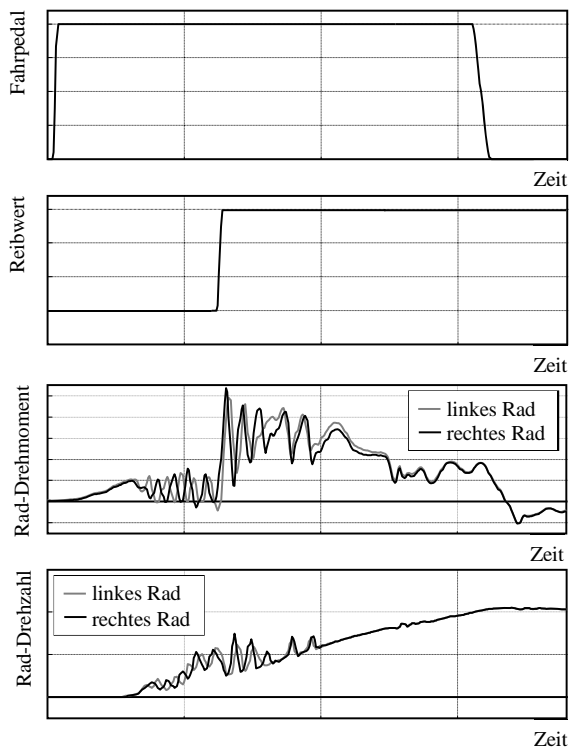
In diesem Beitrag wird das Missbrauchsmanöver „Abfahrt  $\mu$ -Sprung“ betrachtet. Es stellt eine Vollast-Anfahrt auf achsweise unterschiedlichen Reibwerten zwischen Reifen und Straße dar. Im Alltag kann dies durch eine teilweise verschmutzte oder vereiste Fahrbahn

auftreten. Wie in **Abbildung 1** zu sehen, steht das Fahrzeug (Hinterradantrieb bzw. heckbetonter Allradantrieb) zu Beginn des Manövers mit den Rädern der Hinterachse auf einem Niederreibwert-Bereich ( $\mu$ -low, circa 0,1), mit den Vorderrädern auf einem Hochreibwert-Bereich ( $\mu$ -high, circa 1). Der Fahrer betätigt das Fahrpedal, die Hinterräder beschleunigen stark. Bei Erreichen des Reibwert-Übergangs erleidet der Antriebsstrang eine Stoßbelastung in Form eines sprunghaften Anstiegs der beiden Rad-Drehmomente der Hinterräder. Die Abbildung dieser Stoßbelastung in einem Simulationsmodell ist die Zielstellung dieser Modellierung.



**Abbildung 1.** Manöver „Abfahrt  $\mu$ -Sprung“

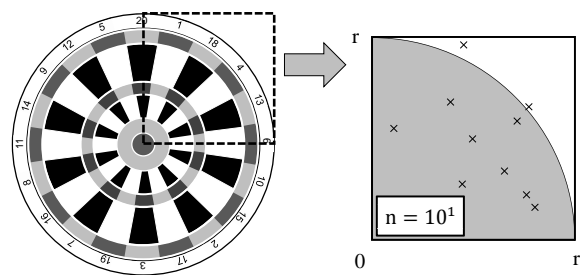
In **Abbildung 2** ist der exemplarische Verlauf dieses Manövers dargestellt, gezeigt werden beide Rad-Drehmomente und Rad-Drehzahlen der Hinterräder, der Reibwertverlauf und das Fahrpedal über die Dauer des Versuchs.



**Abbildung 2.** Verlauf der Rad-Drehzahlen / der Rad-Drehmomente / des Reibwerts der Hinterräder und des Fahrpedals

### 3 Monte-Carlo-Simulation als Optimierungsverfahren

Mit einer Monte-Carlo-Simulation können komplexe Optimierungsprobleme durch das wiederholte Simulieren mit zufällig erzeugten Simulationsparametern gelöst werden [4]. So kann beispielsweise die Kreiszahl  $\pi$  durch eine MCS – in Form von dem Werfen von Dartpfeilen auf eine Scheibe – bestimmt werden, siehe **Abbildung 3**:



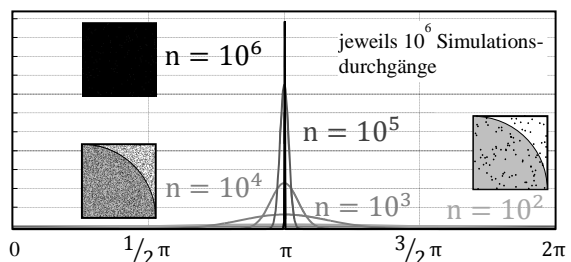
**Abbildung 3.** Bestimmung der Kreiszahl  $\pi$  durch das Werfen von Dartpfeilen

In einem Simulationsdurchgang werden  $n$  Dartpfeile zufällig auf eine Viertelscheibe geworfen, dies entspricht  $n$  gleichverteilten Punkten innerhalb eines Quadrats mit der Seitenlänge  $r$ . Hierbei kann die Kreiszahl  $\pi$  aus dem Verhältnis der Treffer innerhalb des Viertelkreises zu der Anzahl an Würfeln geschätzt werden:

$$\frac{n_{\text{Treffer}}}{n} \approx \frac{\frac{1}{4} \cdot A_{\text{Kreis}}}{A_{\text{Quadrat}}} = \frac{\frac{1}{4} \pi r^2}{r^2} \quad (1)$$

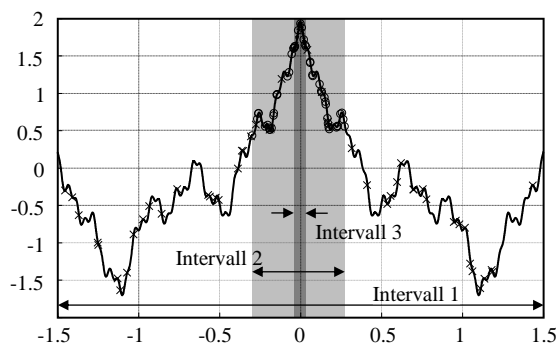
$$\pi \approx 4 \cdot \frac{n_{\text{Treffer}}}{n} \quad (2)$$

Für eine gleiche Anzahl an Simulationsdurchgängen ( $10^6$  Durchgänge in **Abbildung 4**) wird die Streubreite der Ergebnisse für eine steigende Anzahl an Dartpfeilen  $n$  je Durchgang geringer. Eine MCS benötigt also eine hohe Anzahl an Berechnungsschritten für gute Simulationsergebnisse.



**Abbildung 4.** Verteilungen der Ergebnisse in Abhängigkeit der Anzahl an Würfeln

Mehrstufige Monte-Carlo-Simulationen sind gegenüber einfachen MCS effizienter. Soll beispielsweise das Maximum einer Weierstraß-Funktion (an keiner Stelle differenzierbar) in einem Intervall bestimmt werden, führen mehrstufige MCS schneller zum Ziel. In der ersten Stufe werden zufällig Funktionswerte aus dem gesamten Wertebereich berechnet und zwischengespeichert. In der nächsten Stufe wird das Suchintervall auf den Bereich eingeengt, in dem zuvor die höchsten Werte ermittelt wurden. Mit jeder Stufe wird das Suchintervall weiter eingeengt, wie in **Abbildung 5** dargestellt.



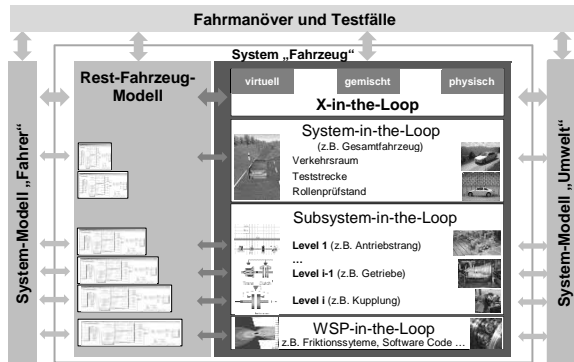
**Abbildung 5.** Dreistufige MCS zur Maximum-Suche einer nicht differenzierbaren Weierstraß-Funktion

#### 4 IPEK-X-in-the-Loop-Ansatz

Der IPEK-X-in-the-Loop-Ansatz (IPEK-XiL) nach Albers integriert die jeweiligen Vorteile bisheriger Ansätze (Model-in-the-Loop, Software-in-the-Loop und Hardware-in-the-Loop) und ergänzt diese um die Belange der Mechanik und Mechatronik sowie die der Entwickler verschiedener Fachdisziplinen [5].

Das (Teil-) System welches sich in der Entwicklung befindet – das „System-in-Development“ (SiD) – kann beispielsweise ein komplettes Fahrzeug (Vehicle-in-the-Loop), ein Getriebe (Subsystem-in-the-Loop) oder ein Reibbelag einer Lamellenkupplung (Working-Surface-Pairs-in-the-Loop) sein, wie in **Abbildung 6** dargestellt.

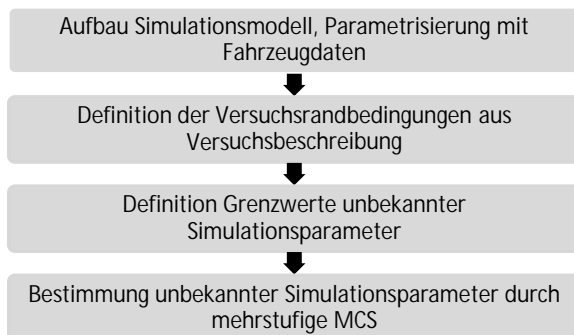
Dabei werden definierte Fahrmanöver und Testfälle des SiDs in Wechselwirkung mit dem System „Fahrer“, dem System „Umwelt“ und dem Restfahrzeugmodell gefahren. Alle Systeme können hierbei virtuell oder real in die Simulation eingebunden sein. Dieser Ansatz ist Grundlage für die in diesem Beitrag dargestellte Simulation.



**Abbildung 6.** IPEK-X-in-the-Loop-Framework [5]

#### 5 Bestimmung unbekannter Simulationsparameter durch eine mehrstufige Monte-Carlo-Simulation

Nach Erstellung des Simulationsmodells erfolgt dessen Parametrisierung, siehe **Abbildung 7**. Dabei werden Fahrzeugparameter verwendet, soweit diese bekannt sind und auf das Modell übertragen werden können. Das Fahrer- und Umweltmodell kann anhand der Versuchsbeschreibung parametrisiert werden, beispielsweise die Vorgabe des Reibwertverlaufs über der Zeit. Die verbleibenden unbekannten Simulationsparameter – wie die Dämpfungskonstanten im Antriebsstrangmodell – werden durch eine mehrstufige MCS bestimmt, wobei zuvor Minimal- und Maximalwerte der Parameter abgeschätzt werden müssen.



**Abbildung 7.** Prozess zur Bestimmung unbekannter Simulationsparameter durch eine mehrstufige MCS

Das Ziel ist die Erstellung eines Modells für anschließende Untersuchungen, so kann beispielsweise das Modell innerhalb eines Design of Experiments (DoE)-Ansatzes [6] genutzt werden.

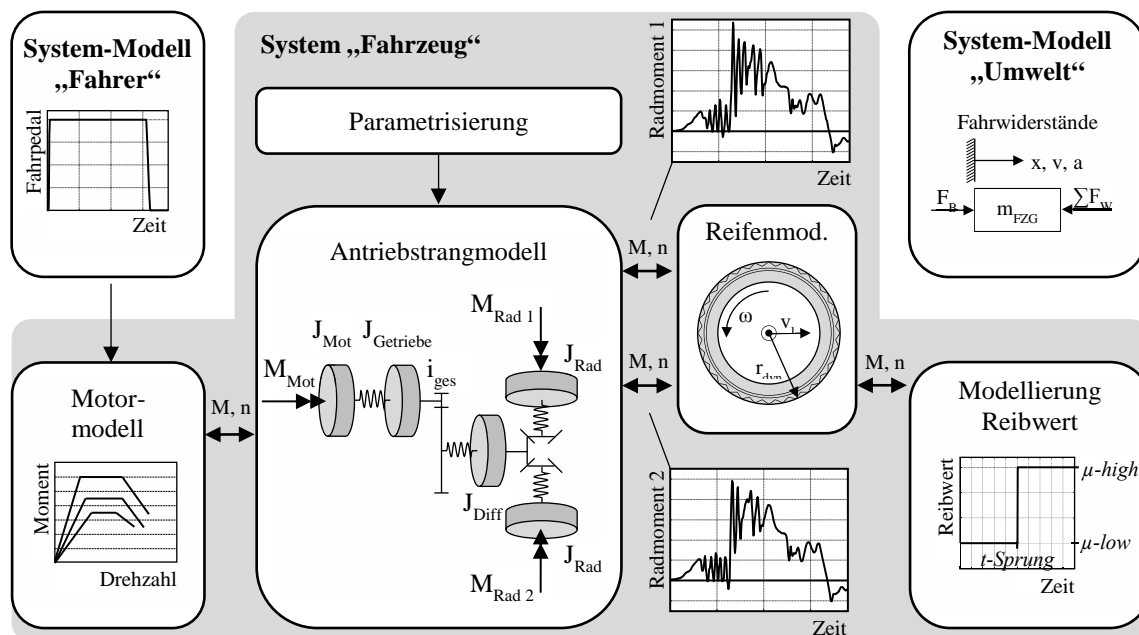


Abbildung 8. Simulationsmodell basierend auf dem IPEK-X-in-the-Loop-Ansatz

### 5.1 Aufbau des Simulationsmodells

Das Simulationsmodell besteht aus Fahrer-, Fahrzeug- und Umweltmodell und basiert auf dem IPEK-X-in-the-Loop-Framework, siehe **Abbildung 8**. Das SiD ist der Fahrzeug-Antriebsstrang ab dem Motor. Das Fahrermodell besteht aus der Vorgabe des Fahrpedals über den Manöververlauf.

Das Fahrzeugmodell setzt sich aus einem Motormodell, der Modellierung des Antriebsstrangs und einem Reifenmodell zusammen. Das Motormodell liefert für einen gegebenen Fahrpedalwert und eine anliegende Motordrehzahl das Motordrehmoment. Das Antriebsstrangmodell ist ein gedämpfter Fünf-Massen-Drehschwinger, welcher auch die Übersetzung des Antriebsstrangs abbildet. Das Reifenmodell berechnet die übertragbaren Rad-Drehmomente aus dem anliegenden Schlupf nach Pacejka [7]. Das Umweltmodell berechnet zum einen die Fahrwiderstände und gibt zum anderen den Reibwert zwischen Reifen und Straße vor

### 5.2 Definition der Versuchsrandbedingungen

Aus der Versuchsbeschreibung können die Versuchsrandbedingungen abgeleitet werden. Dazu gehört die

Vorgabe, das Manöver mit voll ausgelenktem Fahrpedal bis zum Erreichen des Reibwertübergangs zu fahren. Außerdem sind die Werte für den Niedrig- und Hochreibwert analog zu der Versuchsstrecke des Referenzfahrzeugs zu wählen.

### 5.3 Definition der Grenzwerte der unbekannten Simulationsparameter

Nachdem nun das Simulationsmodell aufgebaut und soweit möglich parametrisiert ist, erfolgt die Ermittlung der unbekannten Simulationsparameter. Hierzu muss im ersten Schritt eine Unter- und Obergrenze jedes Parameters gewählt werden. Dies kann anhand von Erfahrungswerten geschehen. Alternativ kann auf Literaturwerte zurückgegriffen werden, wie bei der Parametrisierung des Pacejka-Reifenmodells.

Im Allgemeinen wäre auch die Wahl eines sehr großen Intervalls – wie  $[0, 10^{12}]$  – für alle gesuchten Parameter möglich, jedoch steigt dadurch die Anzahl der benötigten Simulationsdurchläufe bis zur Entdeckung eines brauchbaren Ergebnisses enorm an. Eine Wahl der Parametergrenzen innerhalb einer Größenordnung lieferte hier gute Ergebnisse.

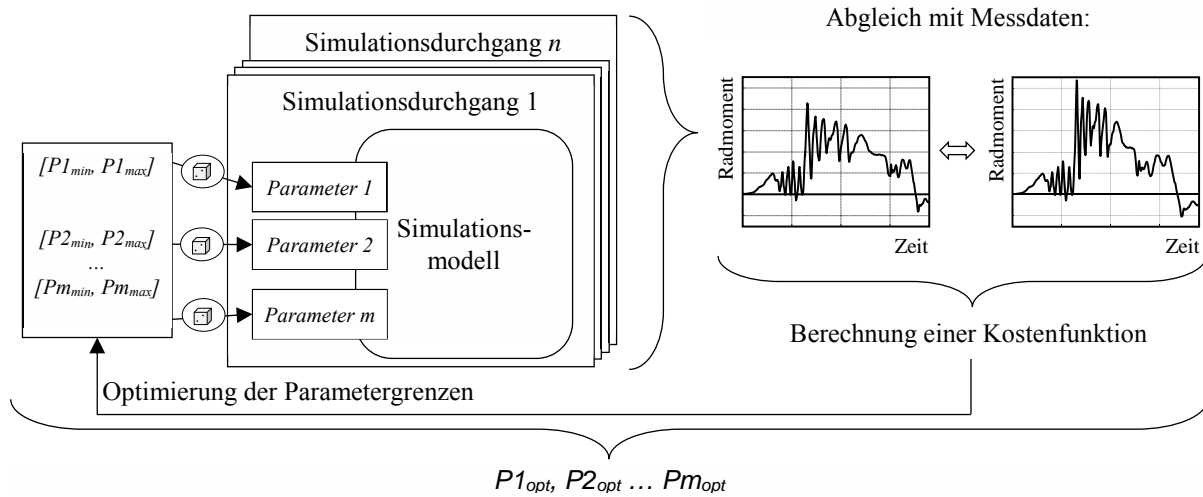


Abbildung 9. MCS zur Bestimmung unbekannter Simulationsparameter

#### 5.4 Optimierung der unbekannten Simulationsparameter

Die Bestimmung eines optimalen Parametersatzes erfolgt durch eine mehrstufige Monte-Carlo-Simulation [8], siehe **Abbildung 9**. In einem Simulationsdurchlauf wird für jeden unbekannten Parameter zufällig ein Wert innerhalb der vorher festgelegten Grenzen ermittelt. Mit diesem Parametersatz wird die Simulation nun durchgeführt, Ergebnis ist ein Verlauf der gesuchten Referenzgröße (hier das Drehmoment des linken Hinterrads). Dieser Verlauf wird mit den Daten der Referenzmessung verglichen und daraus eine Kostenfunktion berechnet und zwischengespeichert. Es findet also ein Gesamtmodell-Abgleich zwischen Simulationsmodell und Messdaten eines Versuchsfahrzeugs statt. Die Kostenfunktion ergibt einen hohen Wert für eine schlechte Übereinstimmung zwischen Messung und Simulation und einen niedrigen Wert für eine gute Übereinstimmung. In einer Stufe werden nun viele Simulationsläufe durchgeführt und zu jedem Lauf eine Kostenfunktion errechnet. Zu Beginn der nächsten Stufe der MCS werden die Parametergrenzen optimiert. Die Grenzen der einzelnen Parameter werden eingengt, um die Effizienz der Simulation zu steigern. Die Einengung der Parametergrenzen erfolgt anhand der berechneten Kosten: Die neue Untergrenze eines Parameters  $P1$  ergibt sich aus dem Minimum der besten Parametersätze, also denen mit den niedrigsten Kosten.

$$P1_{min}^{neu} = \min(P1_1, P1_2 \dots P1_m) \quad (3)$$

Analog hierzu berechnet sich die neue Obergrenze aus dem Maximum der besten Parametersätze:

$$P1_{max}^{neu} = \max(P1_1, P1_2 \dots P1_m) \quad (4)$$

Die Anzahl der Parameter  $m$ , aus denen das Minimum oder das Maximum gewählt wird, ist von der Anzahl der Optimierungsschritte – der Stufen – und den einzelnen Simulationsdurchgängen je Stufe abhängig. In dieser Simulation hat sich der folgende Zusammenhang bewährt:

$$m \approx 5 \cdot \frac{\text{Anzahl Stufen}}{\text{Anzahl Durchgänge je Stufe}} \quad (5)$$

Ergebnis der MCS ist ein optimierter Parametersatz  $P1_{opt}, P2_{opt} \dots Pm_{opt}$  der den bestmöglichen Verlauf der Referenzgröße erzeugt

#### 5.5 Aufstellung einer gewichteten Kostenfunktion für die Optimierung

Abhängig von der Zielstellung einer Simulation können verschiedene Kostenfunktionen Anwendung finden. Im Normalfall bietet sich die Berechnung der Summe der Fehlerquadrate an:

$$K_{qF} = \frac{1}{n} \sum_{i=1}^n (y_i^{Messung} - y_i^{Sim})^2 \quad (6)$$

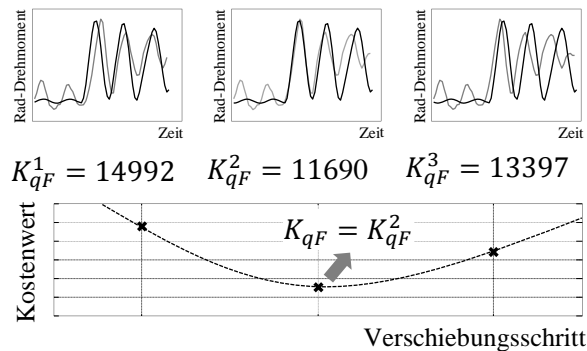
Ist dagegen nur der Maximalwert einer Messgröße (auch innerhalb eines bestimmten Bereichs) von Interesse, so kann dies auch bei dem Aufstellen der Kostenfunktion berücksichtigt werden:

$$K_{max} = (\max(y_{Messung}) - \max(y_{Sim}))^2 \quad (7)$$

Generell können verschiedene Kostenfunktionen auch gewichtet aufsummiert werden:

$$K_{ges} = a_1 K_1 + a_2 K_2 + \dots + a_n K_n \quad (8)$$

Oftmals führen Vereinfachungen bei der Modellbildung zu einem geringen Zeitversatz zwischen dem simulierten und dem wahren Verlauf der Referenzgröße. Bei der Berechnung des quadratischen Fehlers führt dieser Zeitversatz auch bei eigentlich guten Simulationsergebnissen zu einem hohen Kostenwert. Um dies zu vermeiden, wird bei der Berechnung von  $K_{qF}$  der simulierte Verlauf der Referenzgröße schrittweise auf der Zeitachse verschoben und zu jedem Schritt eine Kostenfunktion berechnet, siehe **Abbildung 10**.



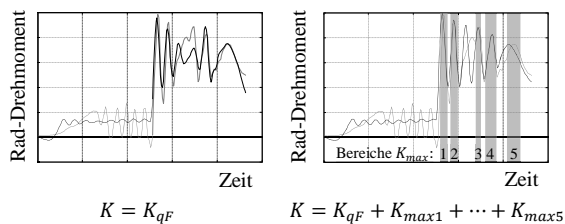
**Abbildung 10.** Kostenwert über Verschiebungsschritt

Der Kostenwert des Simulationsdurchgangs ist dann das Minimum der Kosten je Verschiebungsschritt:

$$K_{qF} = \min(K_{qF}^1, K_{qF}^2 \dots K_{qF}^n) \quad (9)$$

## 5.6 Ergebnis der Optimierung

**Abbildung 11** zeigt das Ergebnis der Optimierung, entsprechend der gewählten Kostenfunktion. Sollen die Maximalwerte des Rad-Drehmoments erreicht werden, bietet sich die Verwendung einer zusammengesetzten Kostenfunktion an:



**Abbildung 11.** Einfluss der gewählten Kostenfunktion

## 6 Zusammenfassung und Ausblick

In diesem Beitrag wurde ein Vorgehen entwickelt, mit dem unbekannte Parameter einer Simulation automatisiert abgeschätzt werden können. Bisher bekannte Ansätze (wie [6]) wurden um gewichtete, zusammengesetzte Kostenfunktionen erweitert, welche auch eine

Zeitverschiebung des Simulationsergebnisses zulassen. Die Optimierung der Parameter durch eine mehrstufige MCS hinsichtlich dieser Kostenfunktionen erlaubt eine genaue Anpassung der Parametrisierung an die vorliegende Zielstellung. Mit der Modellierung eines Missbrauchsmanövers können nun äußere Einflussgrößen auf die entstehenden Bauteilbelastungen untersucht werden. Zukünftig kann die Methode dahingehend erweitert werden, dass auch die Ober- und Untergrenzen der unbekannten Parameter – durch Vorgabe eines Startwerts – automatisiert bestimmt werden. Dies ermöglicht eine weiter automatisierte Bestimmung unbekannter Simulationsparameter und steigert somit die Effizienz bei der Erstellung von Simulationsmodellen.

## 7 References

- [1] K. Matros, F. Schille, M. Behrendt und H. Holzer, „Manöverbasierte Validierung von Hybridantrieben,“ *ATZ - Automobiltechnische Zeitschrift*, Februar 2015.
- [2] B. Bertsche und G. Lechner, *Zuverlässigkeit im Fahrzeug- und Maschinenbau*, Stuttgart: Springer Verlag, 2004.
- [3] H. Brandhoff, W. Pfau und F.-J. Tögel, „Extrembelastungen im PKW-Antriebsstrang durch Missbrauch,“ *ATZ Automobiltechnische Zeitschrift* 101, pp. 1006-1015, 1999.
- [4] H. Nahrstedt, *Die Monte-Carlo-Methode*, Wiesbaden: Springer Vieweg, 2015.
- [5] A. Albers, M. Behrendt, S. Klinger und K. Matros, „Verifikation und Validierung in der Produktentwicklung,“ in *Handbuch Produktentwicklung*, München, Carl Hanser Verlag, 2016, pp. 541-569.
- [6] A. Schwarz, „Integration von Messdaten in die Simulation zur multikriteriellen, zeiteffizienten, versuchsbasierten Optimierung technischer Systeme,“ Karlsruhe, 2013.
- [7] A. Wiesebröck, *Ein universelles Fahrbahnmodell für die Fahrdynamiksimulation*, Wiesbaden: Springer Fachmedien, 2016.
- [8] W. Wohlers, *Mehrstufige Optimierung komplexer strukturmechanischer Probleme*, Göttingen: Cuvillier Verlag, 2005.
- [9] A. Romer, „Anwendungsspezifischer Zuverlässigkeitsnachweis auf Basis von Lastkollektiven und Vorwissen,“ Institut für Maschinenelemente der Universität Stuttgart, Dissertation, Stuttgart, 2014.

# Funktionale Sicherheit (ISO 26262) und Simulation

Prof. Dr. Walter Commerell<sup>1</sup>

<sup>1</sup>Hochschule Ulm

commerell@hs-ulm.de

## Zusammenfassung

Die Entwicklung von Fahrzeugen oder Fahrzeugsystemen erfordert die Anwendung der Norm ISO 26262. Das System wird dabei einer Gefahren- und Risikoanalyse unterzogen. Ergebnis dieser Analyse ist ein Automotive Safety Integrity Level (ASIL) im Bereich QM, A bis D. D bedeutet, dass ein System vorliegt bei dem ein Fehler zu einer hohen Schadensschwere (z.B. Tod) führt, die Verweildauer in dieser Situation sehr Wahrscheinlich ist und bei dem die Situation nicht oder nur sehr schwer kontrolliert werden kann.

Anhand des ASIL's werden bestimmte Maßnahmen während des ganzen Lebenszyklus des Systems gefordert. Ein Entwicklungsprozess der durch die Simulation unterstützt wird, kann hier zu einer wesentlichen Kostenreduktion durch frühzeitige Analysemöglichkeiten und effiziente Testmöglichkeiten führen. Entscheidend dabei ist die Auswahl des Modelldetaillierungsgrad, des Simulators und die Modellarchitektur.

## 1 Einleitung

Bei der Entwicklung von Fahrzeugen hat die Sicherheit einen hohen Stellenwert. Die Norm ISO 26262 (Road vehicles – Functional Safety-) [1] kurz auch FUSI genannt, definiert hierbei Anforderungen an die Entwicklungs-, Fertigungs- und Betriebsprozesse sicherheitsrelevanter Funktionen.

Speziell in der Fahrzeugentwicklung mit seinen hohen Stückzahlen, muss ein System sicher entwickelt werden, da der Test aller Fahrzeuge oder Nachbesserungen im Betrieb nicht angewendet werden können.

Die Simulation ist dabei ein fester wichtiger Bestandteil des Entwicklungsprozesses und hilft Fahrzeugfunktionen sicher zu gestalten.

## 2 Funktionale Sicherheit bei Fahrzeugen

### 2.1 Ziele der ISO 26262

Die ISO 26262 standardisiert das Vorgehen bei sicherheitsrelevanten Fahrzeugsystemen. Sie beschreibt wie die Sicherheitsanforderungsstufe (Safety Integrity Level SIL) ermittelt wird. In der Automotive Norm wird hierbei der Begriff ASIL für „Automotive SIL“ verwendet.

Ziel ist es

- systematische Fehler vermeiden

- Zufällige Fehler minimieren
- Restrisiko minimieren

Neben dem im automobilen Bereich angewendeten V-Prozessmodell werden in der Norm erweiterte Prozessschritte abhängig vom Automotive Safety Integrity Level (ASIL) gefordert. Speziell im Bereich Konzeption und Validierung sind modellbasierte Validierungsschritte erforderlich um die sichere Funktion des Systems zu gewährleisten.

### 2.2 Maßnahmen

Die Norm zielt nicht nur auf die Entwicklungsprozesse ab sondern betrachtet auch die Sicherheitskultur, Sicherheitskompetenzen und den kompletten Lebenszyklus des Produktes. Abbildung 1 zeigt die Kapitel der Norm auf.

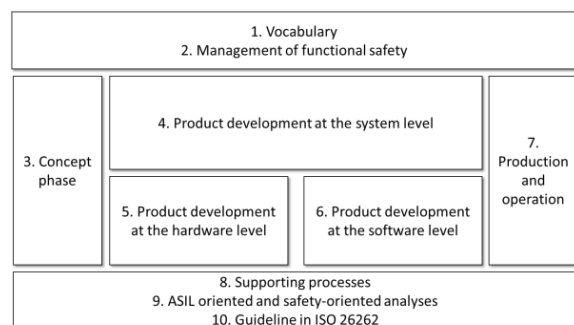


Abbildung 1. Übersicht und Teile der ISO26262

Die Norm verfolgt dabei das Prinzip der unabhängigen Prüfung. Abhängig vom ASIL Level, ist ein “Vier-Augen-Prinzip” oder sogar ein “Sechs-Augen-Prinzip” erforderlich. Zufällige Fehler sollen so oft entdeckt und in einer frühen Phase der Entwicklung behoben werden.

Klare Verantwortlichkeiten vermeiden Missverständnisse. Nachvollziehbarkeit (traceability, configuration management) erleichtern es im Fehlerfall Fehler schnell zu beheben.

Ein Augenmerk der Norm liegt auf der Systemarchitektur und deren Komplexitätsgrad. Dieser sollte so ausgeprägt sein, dass es die Architektur überhaupt erlaubt Fehler zu entdecken.

## 2.3 Vorgehen

Schon bei den ersten Überlegungen zu einer Funktion greift die Norm ein und fordert eine klare Definition mit Grenzen des sogenannten Item (System oder Array von Systemen für die Sicherheitsfunktion im Fahrzeug). Abbildung 2 zeigt das Vorgehensmodell mit den zugehörigen Unterkapitel der Norm.

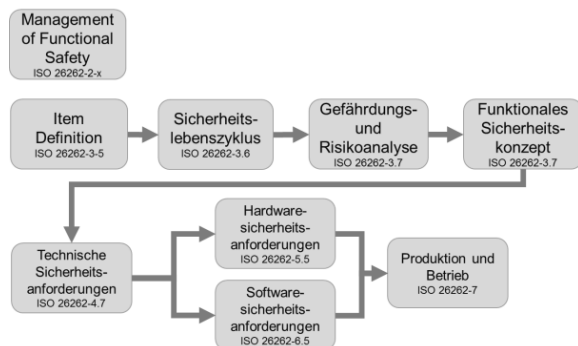


Abbildung 2. Vorgehen

Zentrales Element der geforderten Prozessschritte ist die Zuordnung des “Item” zu einem ASIL Level. Tabelle 1 zeigt auf wie die Risikoanalyse durchzuführen ist. Das “Item” wird dabei je einer Klasse zugeordnet.

Severity class				
	S0	S1	S2	S3
Description	No injuries	Light and moderate injuries	Serve and life-threatening injuries (survival probable)	Life-threatening injuries (survival uncertain) fatal injuries

	Probability class				
	E0	E1	E2	E3	E4
Description	Incredible	Very low probability	Low probability	Medium probability	High probability

Controllability class				
	C0	C1	C2	C3
Description	Controllable in General	Simply Controllable	Normally Controllable	Difficult to control or uncontrollable

Tabelle 1. Severity, Probability und Controllability Klasse

Aus der Zuordnung lässt sich nun der ASIL Level, wie in Tabelle 2 dargestellt, bestimmen.

Severity class	Probability class	Controllability class		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

Tabelle 2. ASIL Level Festlegung

## 3 Simulative Anforderungen der Norm

### 3.1 Systemebene

Bereits im Systementwurf ist die Simulation als Verifikationsmethode für ASIL Level A und B empfohlen und für Level C und D sehr empfohlen wie Tabelle 3 aufzeigt. Alternative zur Simulation können Prototypen aufgebaut und Fahrzeugtest durchgeführt werden.

7.4.8	Verification of system design	ASIL			
7.4.8.1	The system design shall be verified for compliance and completeness with regard to the technical safety concept using the verification methods listed in Table 3.	A	B	C	D
	1a System design inspection	+	++	++	++
	1b System design walkthrough	++	+	o	o
	2a Simulation	+	+	++	++
	2b System prototyping and vehicle tests	+	+	++	++
	3 System design analyses see Table 1)	Inductive and/or deductive analysis			

++ highly recommended  
+ recommended  
o non recommendation for or against

Tabelle 3. Simulation beim Systementwurf

### 3.2 Hard- und Softwareebene

Im Hardwareentwurf wird bei der Verifikation der Hardware Simulation von ASIL B bis D “nur” empfohlen. Während im Software Design bei der Validierung der Architektur die Simulation der dynamischen Teile als Alternative empfohlen und bei Level D stark empfohlen werden (Teil 6: 7.4.18 [1]). Damit verbunden, ist ein ausführbares Modell der Software. Die Verifikation der SW Sicherheitsanforderungen kann mit HiL Simulation nachgewiesen werden (Teil 6: 11.4.2 [1]). Annex B von Teil 6 widmet sich dem Modellbasierten Entwurf.

### 3.3 Hard- und Softwareintegration

Bei der Integration und dem Test der technischen Sicherheitsanforderungen auf Hardwar-Software Ebene wird der sog. “Back-to-back-test” für ASIL Level A und B empfohlen und für Level C und D sehr empfohlen (Teil 4 8.4.2.2.2 [1]). Mithilfe der Simulation soll vor allem die korrekte Funktion, die Genauigkeit und das Zeitverhalten des Sicherheitsmechanismus auf Hardware-Software ebene nachgewiesen werden (Teil 4 8.4.2.2.3 [1]).

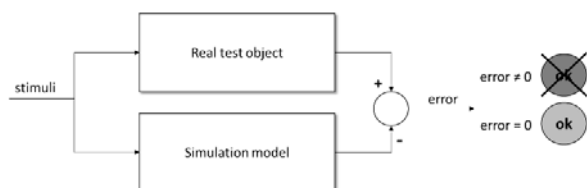


Abbildung 3. Back-to-back Testmethode

Auch auf System-Integrationsebene (Teil 4 8.4.3.2.2 und 8.4.3.2.3 [1]) wird diese Methode als Alternative zu Requirements-based test oder Fault injection test empfohlen um die korrekte Implementierung und Funktion nachzuweisen.

Der Nachweis für die Erfüllung der Sicherheitsziele wird erreicht durch wiederholende Tests mit spezifischen Testprozeduren, Testfällen und pass/fail Kriterien. Beispielsweise, ist hier als Methode die Simulation genannt.

### 3.4 Verifikationsplanung

In der Verifikationsplanung kann eine Simulationsumgebung eingesetzt werden (Teil 8: 9.4.1.1 [1]). Als Methode sind Simulationsszenarien in 9.4.2.1 genannt.

## 4 Optimaler Einsatz von Simulationsmodellen

### 4.1 Simulatorauswahl

Die vielfältigen Anforderungen nach Validierung und Verifikation der Sicherheitsfunktionen verlangt eine sorgfältige Auswahl der Simulationsumgebung.

Soll das Simulationsmodell umfassend für den Systementwurf und die Validierung verwendet werden, ist eine Simulationsumgebung die mehrere Domänen abdeckt zu empfehlen. Entweder wird ein

a) Multidomain Simulator z.B auf Basis von “Modelica”, “VHDL-AMS”,

b) Simulatorkopplung oder

c) Modellaustausch z.B. FMI

eingesetzt. Für kleinere Unternehmen stellt dies oft eine große Herausforderung dar in Bezug auf Invest, Know-how und Betrieb dieser Simulationssysteme.

### 4.2 Modellaufbau

Der Einsatz der Simulation sollte möglichst umfassend und durchgängig gewählt werden. Ein wesentliches Einsparpotential liegt in der Modellarchitektur. Diese sollte so gestaltet sein, dass vor allem zu Beginn der Entwicklung sicherheitsgefährdeter Systeme eine Systemarchitektur gefunden wird, die beherrschbar wird.

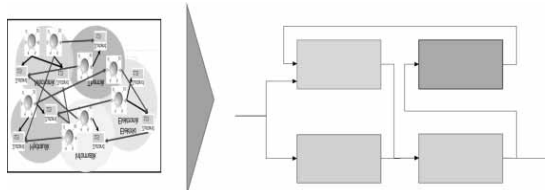


Abbildung 4. Modellarchitektur

Die Beherrschbarkeit drückt sich unter anderem darin aus, dass die Anzahl der Signale zwischen den Modulen begrenzt und messbar ist.

Beherrschbarkeit ist jedoch auch innerhalb eines Moduls erforderlich. Interne relevante Zustandsgrößen sollten vorbereitend nach außen geführt werden.

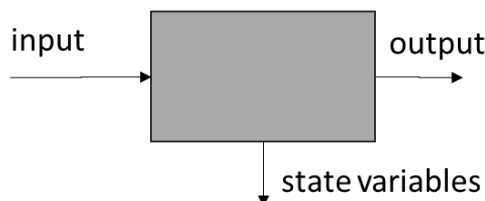
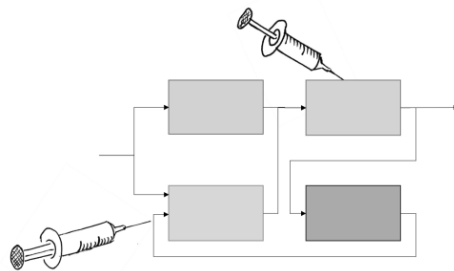


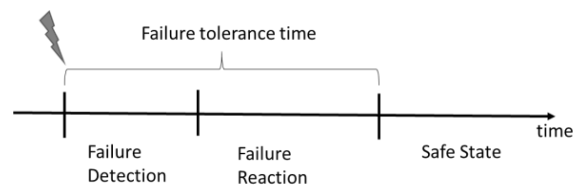
Abbildung 5. Zugriff auf Zustandsvariablen

Eine weitere Modellanforderung ergibt sich aus der Verifikation des Sicherheitskonzeptes. Durch geeignete Wahl der Modellvariablen und –signale können Fehler in das Modell injiziert werden, um die Sicherheitsmechanismen in einem frühen Stadium abzusichern. Das Sicherheitsverhalten des realen Systems kann dann mit dem modellierten Verhalten verglichen werden.



**Abbildung 6.** Fehlerinjektion

Beim Entwurf eines Sicherheitskonzeptes ist das Zeitverhalten (Failure tolerance time wie in Abbildung 7 dargestellt) entscheidend.



**Abbildung 7.** Fehlertoleranz Zeit

Im Speziellen erfordert die Norm den Nachweis des Sicherheitszeitverhaltens. Dies lässt sich geeigneter Weise mithilfe eines dynamischen Modells frühzeitig untersuchen.

## 5 References

- [1] ISO26262 International Standard. *Road vehicles – Functional safety* - . Beuth Verlag, Berlin, 2011

# Modellbasierte Entwicklung für sichere Software bei MAGIRUS GmbH

Kai Kriegl

<sup>1</sup>MAGIRUS GmbH

<sup>2</sup>CNH Industrial

*Kai.Kriegl@cnhind.com*

Eine Fallstudie über die Entwicklung, das Testen und die Integration von sicheren und nicht sicheren Softwareanteilen in einem Steuergerät.

Neben der rein funktionalen Software eines Steuergerätes sind in den vergangenen Jahren zahlreiche Anforderungen bezogen auf die Sicherheitsfunktionen im gesamten Lebenszyklus einer Software hinzugekommen. Die Sicherheitsfunktionen sind aber nicht nur bei der Softwareentwicklung zu betrachten, sondern ziehen sich vielmehr durch alle Abteilungen des Unternehmens. Im Rahmen der Steuerungstechnik-Neuentwicklung bei MAGIRUS wurde diesem Fakt besondere Aufmerksamkeit gewidmet. Neben der Einführung eines Product Lifecycle Management Tools (PLM-Tools) wurde auch bei der Auswahl der Hardware besonderes Augenmerk auf die Erfüllung der Sicherheitsanforderungen gelegt. Bei der Definition der Toolchain zur Softwareerstellung wurden der Sicherheitsgedanke fort- und moderne Methoden wie Modellierung, Simulation und Codegenerierung eingeführt. Auch das Schichtenmodell innerhalb der Software wurde stark von den Sicherheitsfunktionen beeinflusst und hat dazu geführt, dass neben dem Betriebssystem (CodeSys) weitere Abstraktionsschichten wie Middleware, Adaptation Layer und Application erstellt wurden.

Die Fallstudie beschreibt die Zusammenhänge der einzelnen Tools und insbesondere die Interaktion zwischen sicherer und nicht sicherer Software auf einem Steuergerät und die dazu notwendigen Eigenschaften der Funktionsmodelle und einzelnen Softwareschichten.

## 1 Einleitung

Im Rahmen einer Überarbeitung der Steuerungstechnik in den Fahrzeugen der MAGIRUS GmbH begann auch die Neuentwicklung eines Entwicklungsprozesses für Sicherheitsfunktionen im gesamten Unternehmen und mit besonderem Augenmerk auf die Elektrik/Elektronik Abteilung. Neben der besonderen Interaktion zwischen sicherer und nicht sicherer Software auf einem Steuergerät wurde auch der gesamte Entwicklungsprozess mit Unterstützung eines PLM –Tools (Polarion) grundlegend verändert und neu aufgesetzt. Eine große Rolle in der Betrachtung spielten die einzelnen, zur Anwendung kommenden Normen, wie z.B. die DIN EN 14043 für automatische Drehleiter und der anzuwendenden Norm EN 13849 mit Ihren Verweisen auf die EN 61508 für die Funktionale Sicherheit. Während die Drehleiter bereits über eine Einstufung von Funktionen mit einem ‚Performance Level‘ verfügte, musste für andere Fahrzeuge der MAGIRUS GmbH diese Einstufung nach der ‚Maschinenrichtlinie‘ (2006/42/EG) neu erstellt und dokumentiert werden. Der Einsatz eines PLM-Tools unterstützte genauso wie bei der Anwendung von standardisierten Entwicklungsmethoden in

der Softwareerstellung, die Dokumentation und Nachvollziehbarkeit von Funktionen und deren Sicherheitseinstufung. Nachfolgend werden einige Ideen und die Lösungsansätze für die Realisierung beschrieben, welche bei der MAGIRUS GmbH eingeflossen sind.

## 2 Polarion als PLM-Tool

Die Einführung des PLM-Tools Polarion und die Anpassung des Entwicklungsprozesses sowohl für die Elektrik/Elektronik Abteilung als auch für das gesamte Projekt hat viel Überzeugungsarbeit erfordert. Jedoch sind die Erfolge dieser Bemühungen bereits nach kurzer Zeit in einzelnen Bereichen zu spüren und die Vorteile werden sich zukünftig noch deutlicher zeigen.

### 2.1 Requirementsengineering

Wichtigste Grundlage für den erfolgreichen Einsatz eines PLM-Tools für die Software oder das Engineering sind die Requirements. MAGIRUS hat die Einstufung der Requirements in System- und Software requirements als Detaillierungsgrad getroffen. Mit den Systemrequirements wird die allgemeine

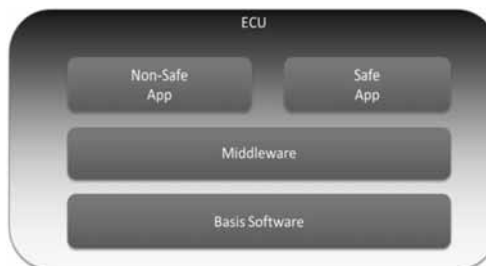
Funktion der Maschine, in unserem Fall der Drehleiter, beschrieben. Die Beschreibungstiefe ist dabei so gewählt, dass jeder, mit technischem Grundwissen über die Drehleiter die Funktion nachvollziehen kann. Die Softwarespezifikation dagegen ist so detailliert gehalten, dass ein Entwickler in der Lage ist, alleine aus der Beschreibung heraus die Software für diese Funktion zu erstellen. Die Systematik besteht in der Verknüpfung der feiner detaillierten Softwarerequirements mit den Systemrequirements. Ebenso sind alle für die Umsetzung benötigten Applikationssignale, Sensorsignale, Aktuatoransteuerungen, sowie CAN-Nachrichten mit den Requirements über eine entsprechende Verlinkung verbunden. Damit ist die Forderung der EN 13849 nach einer Nachvollziehbarkeit der einzelnen Elemente voll erfüllt. Mit Hinzunahme verlinkter Testrequirements und den dokumentierten Ergebnissen deren Ausführung, enthält Polaron alle wichtigen Informationen rund um das entwickelte Produkt und ist somit für alle Beteiligten transparent.

## 2.2 Datenexport und Codegenerierung

Die Detaillierung der Software Requirements mit Ihrer Verlinkung zu allen Sensoren/Aktoren, Signalen und CAN-Nachrichten hat neben der Nachvollziehbarkeit auch den Vorteil, dass für die Softwareerstellung alle Daten aus Polaron extrahierbar sind. Diesen Umstand hat MAGIRUS dazu genutzt, um aus Polaron Daten wie z.B. die K-Matrix für die CAN Kommunikation oder die Variablenamen und auch die Sensor-/Aktorkonfigurationen zu extrahieren und über Codegeneratoren in einen automatischen Build-Prozess zu integrieren. Damit muss MAGIRUS sich bei der Softwareintegration nicht mehr um die korrekte Einstellungen von EEPROM Speicherplätzen, CAN Konfigurationen oder Sensorumrechnungen kümmern, da alles bereits aus Polaron fertig extrahiert in einem ‚One Button Build‘ einfließt, auf die Steuerung übertragen und dort von der Middleware Schicht ausgewertet wird. Der Softwareentwickler bei MAGIRUS ist für die korrekte Modellierung seiner Applikation zuständig.

## 3 Softwarearchitektur

Ein weiterer wichtiger Bestandteil einer Software ist die interne Architektur. Wie in der Einleitung schon beschrieben setzt sich die Softwarearchitektur aus 3 Schichten zusammen, wobei nur in der Applikationsschicht zwischen sicherer und nicht sicherer Software unterschieden wird.



**Abbildung 1:** Schichtenmodell ECU Software bei MAGIRUS

### 3.1 Schichtenmodell

Die Middleware als Abstraktionsschicht zwischen Betriebssystem und den Applikationen ist darauf ausgelegt, mit unterschiedlichen Applikationen beider Einstufungen zu kommunizieren. Dazu ist die Middleware als sichere Softwareschicht deklariert und bearbeitet damit direkt alle Aktoren und Sensoren. Auch die Umrechnung der direkten Sensor- und Aktorwerte in die physikalischen Einheiten übernimmt die Middleware.

Der Adaptation Layer ist in der Softwarearchitektur von MAGIRUS notwendig geworden, damit keine spezifischen Anpassungen an das Modell gemacht werden müssen, um Sensorvariablen vom richtigen Eingang wie auch Aktuatoransteuerungen auf den richtigen Pin des Steuergeräts zu schreiben.

Die Applikationsschicht besteht einzig aus der Funktionsmodellierung der Software. Dazu ist im Vorfeld in Polaron bereits festgelegt, welche Ein- und Ausgangswerte das Modell besitzt, sowie die Parameter zur Einstellung der Funktionen. Eine der wichtigsten Festlegungen bei der Softwarearchitektur ist die Trennung von sicheren und nicht sicheren Softwareanteilen auf Applikationsebene. Damit kann MAGIRUS seine Funktionssoftware weiterhin, wie gewohnt, an den Bedürfnissen der Kunden ausrichten, ohne jedes Mal nach einer Änderung eine neue Zertifizierung der Gesamtsoftware anstoßen zu müssen. Dadurch kann MAGIRUS sehr zügig Softwarewünsche, die durch einen Kundenauftrag getrieben sind, umsetzen.

### 3.2 Interaktion von sicherer Software mit nicht sicherer Software

Bei der Betrachtung des Softwareschichtenmodells und der expliziten Trennung von sicherer und nicht sicherer Applikation fällt auf, dass sowohl die Funktionssoftware (nicht sicher) als auch die Sicherheitssoftware auf Eingangs- bzw. Ausgangswerte

zugreifen müssen und damit sich gegenseitig stören könnten. Auch die gemeinsame Verwendung eines Prozessorkerns, EEPROMs und weiterer Hardware-Ressourcen erfordern, dass die Rückwirkungsfreiheit explizit nachgewiesen wird. Für diesen Nachweis sind bei der Interaktion beider Softwareteile besondere Vorkehrungen in der Middleware, dem zentralen sicheren Element der Software, getroffen. Im Speziellen sollen die Ein- und Ausgänge als primäre Störeinheit für die sichere Software nun genauer betrachtet werden.

**Abbildung 2** zeigt das generelle Zusammenspiel aller Softwareschichten miteinander.



**Abbildung 2:** Zusammenspiel der Softwareschichten

BSW: Betriebssystem

MW: Middleware

Jacking: Non-safe App: Funktionssoftware

Jacking-safe: Safe App: Sicherheitssoftware

### Eingangsbetrachtung

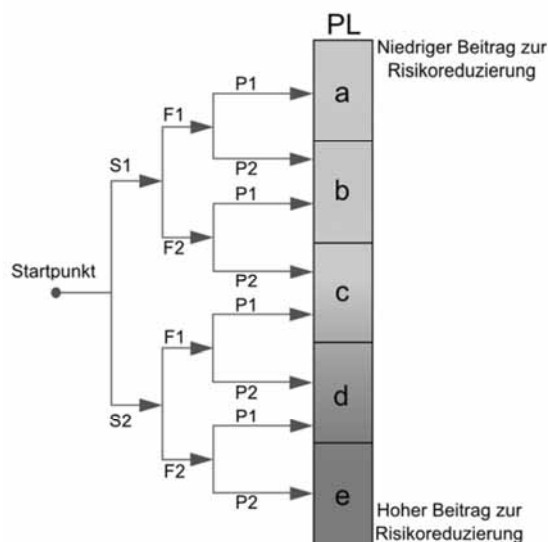
Für die Eingangsbetrachtung, (wie auch die Ausgangsbetrachtung) ist es einfacher, wenn man sich die Problemstellung an einem expliziten Beispiel ansieht. MAGIRUS ist ein Hersteller von Feuerwehrfahrzeugen und damit auch von Drehleitern. Bei Drehleitern ist die Abstützung ein zentrales Funktionselement, ohne das ein weiterer Betrieb der Drehleiter erst gar nicht möglich ist. In der Abstützungsfunktion ist es sehr wichtig, zu wissen, wie weit jede einzelne Stütze ausgefahren ist. Eine Sicherheitsfunktion der Drehleiter lautet:

**SF\_1.1** Vermeidung eines Ausfalls der Stützbreitenerfassung

Mit folgender Sicherheitseinstufung:

- Verletzungsgrad: S2  
S2: schwere, irreversible Verletzung
- Häufigkeit: F2  
F2: Häufige oder dauernde Benutzung der Funktion bei Maschinenbenutzung.
- Vermeidung: P1  
P1: Vermeidung der Gefährdung unter bestimmten Umständen möglich

Für die Sicherheitsfunktion ergibt sich nach der DIN EN 13849 ein geforderter Performance Level von d (PL d). **Abbildung 3** zeigt den Entscheidungsbaum aus der Norm.



**Abbildung 3:** Entscheidungsbaum nach EN 13849

Zur Vereinfachung der Betrachtung wird beispielhaft nur eine Stütze herangezogen. Die beteiligten Sensoren an der Stützbreitenerfassung sind folgende:

- Inkrementalsensor Spur A/B
- Endlagenschalter

Der Inkrementalsensor erkennt über die Flankenwechsel von Spur A und B die Bewegungsrichtung der Stütze, da die Spuren 90° phasenverschoben zueinander sind. Der Endlagenschalter wird dazu benutzt, um die Nulllage der Stütze zu erkennen.

In dem Zusammenspiel mit einer Ansteuerung der Stütze und der Bewegung der Stütze kann es folgende Fehlerszenarien neben dem eines Sensordefektes geben:

- Bewegung wird erkannt, aber eine Ansteuerung liegt nicht vor.
- Ansteuerung liegt vor, aber es wird keine Bewegung erkannt.

In beiden Fällen ist davon auszugehen, dass die tatsächliche Stützbreite nicht bekannt ist und damit im weiteren Verlauf ein Kippen der Leiter riskiert wird. Hier muss nun die Sicherheitsfunktion greifen und für die Funktionssoftware Vorgaben machen, die ein Kippen der Leiter verhindert. Zu diesem Zweck wird

die Sicherheitsfunktion der Funktionssoftware den ungünstigsten Wert für die Stützbreite weitergeben.



Abbildung 4: Sensordaten durch die Softwareschichten

**Abbildung 4** zeigt das neue Zusammenspiel der Softwareschichten welches für Sensoren verwendet wird, die in Sicherheitsfunktionen eingesetzt werden.

Detailliert betrachtet bedeutet es, dass die Weitergabe des Sensorwertes aus dem Betriebssystem über die Middleware an die Funktionssoftware erst nach der Sicherheitssoftware stattfindet. Das heißt, dass die Middleware in diesem Fall wissen muss, dass bei einer Fehlererkennung für diesen Sensor nicht der wirklich gemessene Sensorwert weitergegeben werden darf, sondern ein vorher parametrierter Ersatzwert. Der Ersatzwert wird für die weitere Berechnung in der Funktionssoftware verwendet und die Fehlerinformation bleibt bis zu einer Fehlerücksetzung bestehen. Polarion als Dokumentationstool beinhaltet eine Markierung aller Sensoren, die über entsprechende Fehlererkennungsmechanismen diesem Funktionsprinzip folgen. Auch dieser Teil wird über Datenexport und Codegeneration dem Steuergerät als Konfiguration mitgeteilt. In einer speziellen Fehlerreaktionstabelle ist auf Sensorebene festgehalten, welche Auswirkungen auf die Drehleiterfunktion eine Fehlererkennung für den Sensor hat, sowie die Bedingungen für ein Rücksetzen der Fehlerinformation.

### Ausgangsbetrachtung

In oben genannter Sicherheitsfunktion (SF\_1.1) spielen Aktoren nur dahingehend eine Rolle, als dass die Ansteuerinformation als Vergleich herangezogen wird. In anderen Sicherheitsfunktionen ist die Sensorvalidierung ohne Ansteuerinformation der zugehörigen Hydraulikachse möglich. Für eine Sicherheitsfunktion ist nicht nur die Validierung der Sensorsignale Bestandteil, sondern auch Hydraulikachsen oder Aktoren haben Anteil daran. Bei jeder Sicherheitsfunktion wird auch der sichere Zustand der Maschine definiert. Per se ist es bei fast allen Maschinen so, dass die Abschaltung jeglicher Energie für die Maschine der sichere Zustand bedeutet. Im Falle einer Drehleiter steht die Abschaltung im großen Widerspruch zur Verfügbarkeit, weshalb meistens nur der zweite Abschaltweg die Trennung von der Energiequelle ist. Im ersten Schritt wird immer versucht, den entsprechenden Aktor in ‚Nullposition‘ zu bewegen

und damit den sicheren Zustand: keine Bewegung, herzustellen. Für die Ausgangsbetrachtung des Zusammenspiels von sicherer und nicht sicherer Applikation wird eine andere Sicherheitsfunktion herangezogen:

**SF\_1.4** Vermeidung einer ungewollten Bewegung der Abstützung

Mit folgender Sicherheitseinstufung:

- Verletzungsgrad: S2  
S2: schwere, irreversible Verletzung
- Häufigkeit: F1  
F1: Seltene Benutzung der Funktion bei Maschinenbenutzung.
- Vermeidung: P1  
P1: Vermeidung der Gefährdung unter bestimmten Umständen möglich

Der Entscheidungsbaum von **Abbildung 3** gibt Performance Level c (PL c) für diese Sicherheitsfunktion aus.

Die beteiligten Aktoren an der Sicherheitsfunktion sind:

- Hydraulikventil Stütze Ausfahren
- Hydraulikventil Umschaltung Abstützbetrieb / Leiterbetrieb

Die Bewegung der Abstützung erfolgt, wenn das Umschaltventil auf Abstützbetrieb steht und das Hydraulikventil Stütze Ausfahren bestromt wird. Zusätzlich ist Hydraulikdruck notwendig für die Bewegung.

Im Betrieb der Drehleiter sind folgende Fehlerszenarien möglich, die das Ausführen der Sicherheitsfunktion SF\_1.4 anstoßen:

- Verklebtes Ventil an der Achse Stütze Ausfahren
- Leckage

In beiden Fällen ist, nachdem die Sicherheitsfunktion aktiviert wurde, der sichere Zustand anzufahren. Dies heißt, dass die Umschaltachse Abstützbetrieb / Leiterbetrieb auf Mittelstellung gestellt wird und dass die Achse Stütze Ausfahren auf ‚keine Ansteuerung‘ gestellt wird. Aus **Abbildung 2** ist ersichtlich, dass beide Softwareanteile über die Middleware die Ausgänge setzen dürfen und damit im ungünstigen Fall die Sicherheitsfunktion umgehen können. **Abbildung 5** zeigt die geänderte Interaktion zwischen sicherer und nicht sicherer Software.



Abbildung 5: Aktordaten durch die Softwareschichten

Die Middleware als zentrales Element, welche zwischen „Abbildung 4: Sensordaten durch die Softwareschichten“ und „Abbildung 5: Aktordaten durch die Softwareschichten“ umschalten muss erhält ihre Konfigurationsinformationen (analog zu den Eingängen) direkt aus Polarion. In Polarion muss der Entwickler festlegen, welche Ergebnisse ein ‚Überschreiben‘ der Ausgangssteuerung durch die sichere Software erforderlich machen. Des Weiteren hat MAGIRUS zwei Kategorien der Überschreibung eingeführt, um bei laufenden Bewegungen ein sanftes Anfahren des sicheren Zustandes zu ermöglichen. Ein abruptes Anfahren des sicheren Zustandes kann in den Benutzungsgrenzen der Maschine zu einem Schwingungsverhalten und letztendlich zum Kippen der Drehleiter führen.

### 3.3 Testkonzept

Mit der Einführung des neuen Entwicklungsprozesses bei MAGIRUS ist auch das Testkonzept überarbeitet worden. Basis des Testkonzeptes ist es, die Software so früh als möglich im Entwicklungsprozess zu testen. Neben der hohen Qualifizierung der Software durch entwicklungsbegleitende Tests, soll auch die Testdauer am Fahrzeug verringert werden. MAGIRUS hat sich das Ziel gesetzt, Kundenwünsche wie gewohnt in kurzer Zeit zu realisieren und dabei die Qualität der Software auf bekannt hohem Maß zu erhalten. **Abbildung 6** zeigt schematisch das Testkonzept bei MAGIRUS.

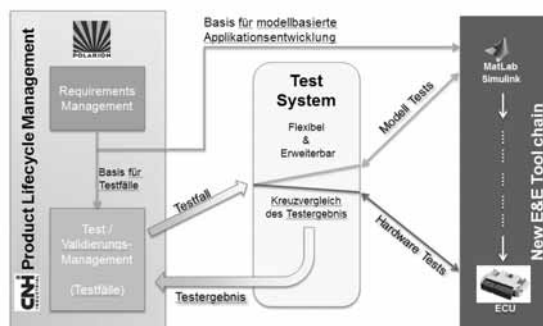


Abbildung 6: Testkonzept MAGIRUS im Überblick

Die Grundlage für das Testkonzept bildet die Definition der Testfälle in Polarion und ein gemeinsames Testsystem. Mit dem Testsystem können sowohl Modultests auf Modellebene durchgeführt werden, als

auch Gesamttests auf der ECU (Electronic Control Unit). Bei der Modellebene steuert das Testsystem die Entwicklungsumgebung während die Hardware-Tests auf einen HiL (Hardware in the Loop) Prüfstand durchgeführt werden. Durch die gemeinsame Verwendung des Testsystems sind die Ergebnisse der einzelnen Tests automatisch vergleichbar. Somit können zu einem sehr frühen Zeitpunkt in der Testphase Probleme in dem Zusammenspiel der einzelnen Softwareelemente erkannt werden. Dabei ist die Klassifizierung der Software (sicher oder nicht sicher) beim Testen unbedeutend. Für beide Softwarearten werden die Tests durchgeführt und dokumentiert. In Kapitel 3.2 ist das Zusammenspiel von sicherer und nicht sicherer Software beschrieben worden. Hier soll das Testsystem mit dem HiL v.a. die Rückwirkungsfreiheit bei Änderungen der nichtsicheren Applikation auf das Gesamtsystem nachweisen.

Die Möglichkeiten des Testsystems mit der tiefen Testung der kompletten Software, darf aber nicht über die Tatsache hinweg täuschen, dass abschließende Validierungstests am Fahrzeug sinnvoll, notwendig und auch durch die Norm EN 13849 gefordert sind. Der gemeinsame Zugriff auf die Testfalldefinition in Polarion bietet dabei die Grundlage für den Vergleich der Ergebnisse.

## 4 Fazit

Mit Einführung des neuen Entwicklungsprozesses bei MAGIRUS können die normativen Regelungen für sicherheitsgerichtete Software aktuell und auch in Zukunft eingehalten werden. Die Trennung der Applikationssoftware zwischen sicherer und nicht sicherer Software gibt MAGIRUS zudem die Möglichkeit, in der Funktionssoftware Änderungen durchzuführen, ohne dass eine Re-Zertifizierung der gesamten Software durch eine externe Instanz notwendig ist. Über das Testkonzept ist jederzeit eindeutig nachweisbar, dass die nicht sichere Software rückwirkungsfrei auf dem Steuergerät ausgeführt werden kann, da alle Testfälle, sowohl für die Funktions- als auch für die Sicherheitssoftware, durchgeführt und rückgespeichert werden.

Zusammengefasst kann gesagt werden:

MAGIRUS ist fit für die Zukunft.

# Echtzeitsimulation hochdynamischer Fahrzeugantriebe

Häckh, Jakob<sup>1</sup>, Willmerding, Günter<sup>1</sup>

<sup>1</sup>Steinbeis Transferzentrum Verkehrstechnik.Simulation.Software

[www.stz-verkehr.com](http://www.stz-verkehr.com)

## Abstract

Die Anforderungen an Fahrzeugantriebe sind vielfältig und es wird ein sehr großer Aufwand betrieben, um alle Anforderungen hinsichtlich Fahrleistung, Emissionen, Lebensdauer, Kraftstoffverbrauch zu erfüllen. Um eine effiziente Entwicklung in kurzer Zeit zu ermöglichen, werden Computersimulationen, Prüfstands-simulationen und Fahrversuche durchgeführt. Bei den Fahrversuchen werden reale Streckendaten erfasst, die dann Grundlage für die Prüfstands- und Computersimulationen sind.

Das Nachfahren von den, in der Realität gemessenen Fahrzyklen auf einem Antriebsstrang-Prüfstand, ist eine regelungstechnisch sehr anspruchsvolle Aufgabe, da die Elemente des Antriebsstrangs relativ große Totzeiten und Nichtlinearitäten aufweisen. Zur Lösung des Problems wurde eine Kombination von Steuerung und Regelung gewählt. In der Prüfstandssteuerung wird dazu eine Echtzeitsimulation des Antriebsstrangs, unter den Prüfstandsbedingungen durchgeführt und die Stellgrößen (Gaspedalstellung, Bremse, Gang, etc.) werden an den Prüfstand übergeben. Die Reaktionen des Prüfstands werden mit denen des Simulationsmodells verglichen und Abweichungen zwischen Prüfstandsverhalten und Simulationsmodell werden verwendet, das Simulationsmodells anzupassen.

Dieses System wird seit Jahren bei führenden Automobilfirmen zur Antriebsstrang-Simulation und Prüfstands-Steuerung eingesetzt und hat sich sowohl bei Pkw als auch bei Busantrieben bewährt.

## 1 Einleitung

Prüfstandszyklen für Gesamtantriebsstrangerprobungen werden überwiegend als Blockprogramme oder Konstantfahrzklen (z.B. unter Überlast) durchgeführt. In [6] wurde gezeigt, dass das kritische Bauteilkollektiv für die Auslegung im Hinblick auf Lebensdauer hilfreich ist, wozu jedoch reale Fahrzyklen – im Folgenden als On-Line-Simulation bezeichnet – zu Grunde zu legen sind. Die Möglichkeit reale Fahrzyklen nachzufahren stellt einen Meilenstein für die Antriebsstrangerprobung dar, da praxisnähere Ergebnisse erzielt werden können.

Anfängliche Versuche zur Online-Simulation führten jedoch zu starken Verspannungen im Antriebsstrang und damit zu unbrauchbaren Ergebnissen.

Die Realisierung eines Prüfstands zur Online-Simulation, bei dem die Probleme der Verspannung gelöst wurden, gelang für Bus-Antriebsstränge vor der Jahrtausendwende für einen BBC-Prüfstand, der seitdem erfolgreich bei einem Bus-Getriebehersteller eingesetzt wird.

Auch für hochdynamische Pkw-Fahrzeugantriebe konnte eine Prüfstandssteuerung für die On-Line-Simulation zusammen mit der Daimler AG entwickelt werden [4]. Arbeiten zur Optimierung und robusten Regelung folgten [3] und das System wurde als wichtiger Teil der Entwicklungsmethodik von der Daimler AG bewertet [1].

Auf Grund positiver Ergebnisse für die Reproduzierbarkeit und Genauigkeit der Prüfstandssimulationen im Vergleich mit Fahrversuchen [17], wurde die Prüfstandssimulation mit winEVA bei der Daimler AG ausgerollt und seitdem eine große Zahl von Prüfständen so ausgestattet.

Die weitere Entwicklungsarbeiten konzentrieren sich auf die einfache Datenermittlung für die Parameterierung des Prüfstands, insbesondere der dynamischen Kenngrößen durch Lernzyklen, die robuste Fehlerbehandlung bei dem Betrieb und die Einbeziehung von Hybrid-Topologien [18] in die Simulation.

Folgende wesentlichen Elemente sind für Prüfstandssteuerung mittels Online-Simulation erforderlich:

- ✓ Detailliertes Antriebsstrangmodell mit allen wesentlichen Antriebsstrangkomponenten
- ✓ Wegbasierte Simulation der Wunschgeschwindigkeit (Wunschgeschwindigkeitssimulation unterscheidet sich hierbei durch größere Geschwindigkeitsintervalle vergleichbar mit den Geschwindigkeitsangaben bei einer Fahrzeugfahrt auf öffentlichen Straßen)
- ✓ Das Fahrermodell strebt unter Berücksichtigung von Streckendaten wie Steigung, Krümmung, und Geschwindigkeit, sowie Fahrereingriffen wie Gangbegrenzung oder Wahl geänderter Fahrprogrammen die Wunschgeschwindigkeit an
- ✓ Rückkopplung von Prüflingsreaktionen in die Online-Simulation
- ✓ Durch Lernzyklen lernt die Online-Simulation automatisiert die Besonderheiten des Prüfaufbaus
- ✓ Übertragbarkeit der Prüfprogramme auf unterschiedliche Prüfstandssteuerungen

Potenziale unter Nutzung des Systembaukastens:

- ✓ Verkehrsflussabhängige Simulationen können einen Beitrag zur Ermittlung der Streuung der Einsatzbedingungen leisten [9]
- ✓ Systeme zur Erfassung von Streckenvorgaben beschleunigen den Transfer von Straßenerprobungen auf den Prüfstand
- ✓ Schadensakkumulationsanalysen zur Ermittlung ein-satzbezogener Beanspruchungen der Antriebsstrangkomponenten sowie der Fokussierung auf die kritischen Bauteilkollektive [6]
- ✓ Effektive realitätsnahe Lebensdauerabschätzung von Antriebsstrangkomponenten [7]
- ✓ Analyse von hochdynamischen Vorgängen wie Schaltvorgängen in Automatgetrieben erlauben Aussagen über Funktion, Komfort und Lebensdauer [5]
- ✓ Vergleich von neuen Antriebskonzepten unter realitätsnahen Betriebsbedingungen [4]

## 2 Echtzeitsystem für die Online-Simulation

Die für die Prüfstandssteuerung mittels Online-Simulation verwendete Antriebsstrangsimulation winEVA entspricht aufbauseitig den objektorientierten Simulationssystemen, die im vorliegenden Fall nach dem Ursache-Wirkungs-Prinzip arbeitet. Traditionell sind die Rechenzeiten dieser Systeme nicht echtzeitsynchron, so dass bei der Verwendung von unterschiedlichen Systemen auf Prüfständen die Echtzeitfähigkeit über eine Echtzeitbox hergestellt werden muss, hierfür wird die Rechenzeit der Online-Simulation entsprechend „heruntergebrems“.

### 2.1 Prüfstandssteuerung

Die Aufgaben der klassischen Prüfstandssteuerungen sind unter anderem die Steuerung, Aktivierung und Überwachung,

- ✓ der Sicherheitssysteme wie Warn- und Abschaltbedingungen oder in kritischen Situationen die Aktivierung der Löschsysteme,
- ✓ der Hilfsbetriebe wie Kraftstoff, Kühlwasser, Luft, el. Versorgung...,
- ✓ der Kommunikations- und Bussysteme incl. Restbus-simulation, der Messsysteme,
- ✓ die Übertragung und Regelung der Vorgabewerte, usw..

Ein Teil der Prüfstandssteuerung stellen die Vorgabesysteme dar, die den Programmablauf das sog. Prüfprogramm abbilden und auf den Prüfaufbau übertragen.

Im vorliegenden Fall ist unter der Prüfstandssteuerung mittels Online-Simulation der Teil der Prüfstandssteuerung für die Vorgabesysteme zu verstehen. Verglichen mit der Straßenerprobung stellt die Prüfstandssteuerung mittels Onlinesimulation im Wesentlichen die Aufgabe des Fahrers dar. Ziel des Vorgabesystems und somit der Online-Simulation ist, unter Berücksichtigung vieler Parameter, einem vorgegebenen Geschwindigkeitsverlauf zu folgen. Hierfür muss die Simulation ständig ein Kräftegleichgewicht zwischen den Fahrwiderständen, die sich aus den Steigungswiderständen, den Luftwiderständen, den Rollwiderständen und den Beschleunigungswiderständen ergeben und dem Zugkraftangebot aus dem Antriebsstrang über den Pedalwert des Fahrers herstellen. Das Prinzip der Antriebsstrangprüfstandssteuerung ist in Abbildung 1 vereinfacht (Bremsen fehlt) dargestellt.

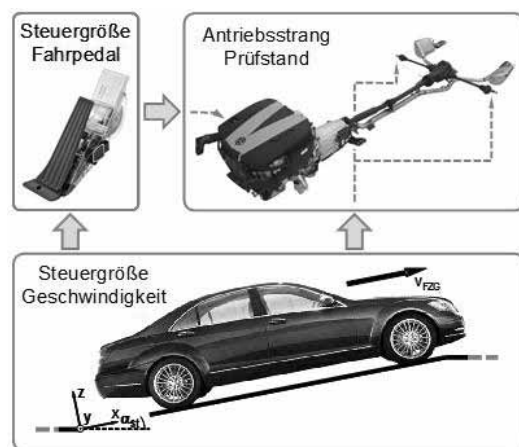


Abbildung 1: Schematische Darstellung der Aufgabe der Prüfstandssteuerung

## 2.2 Prüfstandssteuerung als Regler

Der nachfolgende Abschnitt verdeutlicht die Begrifflichkeit der Prüfstandssteuerung mittels Online-Simulation im Gegensatz zur Prüfstandsregelung und zeigt die möglichen Ausführungen von Reglern und deren Auswahl.

### Klassifizierung des Reglers

Die Eigenschaften eines Reglers, der für die vorliegende Aufgabe geeignet ist, muss folgende Funktionen aufweisen, [3]:

- ✓ **Zeitinvarianz:** Ein System ist zeitinvariant, wenn es auf eine Erregung unabhängig davon reagiert, wann das Ereignis eintritt. Das vorliegende System ist nicht zeitinvariant.
- ✓ **Folgeregelung:** Bei der Prüfstandsregelung handelt es sich um eine Folgeregelung. Das geht aus der Forderung hervor, dass ein bestimmter Geschwindigkeitsverlauf nach gefahren werden muss. Bei einer Folgeregelung empfiehlt sich eine Vorsteuerung.
- ✓ **Linearität:** Bei dem Prüfstand handelt es sich um ein nicht lineares System, da einzelne Komponenten nicht-linear sind. Das vorliegende System lässt sich aufgrund ständiger Arbeitspunktverschiebungen nicht sinnvoll linearisieren.
- ✓ **Kausalität:** Die Kausalität besagt, dass ein bestimmter Wert einer Eingangsgröße das Verhalten des Systems nur in den folgenden Zeitschritten beeinflusst. Bei diesem System ist die Kausalität gegeben.
- ✓ **Totzeitsystem mit sehr großer Totzeit.** Wichtig bei der Beurteilung der Totzeit ist das Verhältnis zwischen Totzeit und Anstiegszeit der Sprungantwort.
- ✓  $TT_{\text{Totzeit}} / T_{\text{Anstiegszeit}} \ll 1$
- ✓ Bei diesem System handelt es sich um ein Totzeitsystem

### Einfache Regler

Um die in Kapitel 2.1 genannten Anforderungen gerecht zu werden, könnte die Prüfstandssteuerung als typischer Regler [13], wie in Abbildung 2 dargestellt, ausgeführt werden.



Abbildung 2: Prüfstandssteuerung als Regler

- $w(t)$  Führungsgröße, in diesem Fall der vorgegebene Geschwindigkeitsverlauf.
- $e(t)$  Regelabweichung.
- $u(t)$  Stellgröße, bei der Prüfstandssteuerung die Fahrpedalstellung.
- $d(t)$  Störgröße, bei der Prüfstandssteuerung sind es

die Summe der vorgegebenen Fahrwiderstände und die zu beschleunigende reduzierte Masse. Dies sind keine unbekannten Störgrößen im eigentlichen Sinn, sondern sie könnten aus den Fahrzeugdaten, dem Geschwindigkeitsverlauf und den Streckendaten berechnet werden.

- $y(t)$  Regelgröße, in diesem Fall die Fahrgeschwindigkeit.

### Regler mit Vorsteuerung

Da es sich bei der Prüfstandsregelung um eine Folgeregelung (Geschwindigkeitsverlauf) und nicht um eine Festwertregelung handelt, ist oft ein schnelles Umsteuern des Systems notwendig, was durch eine Vorsteuerung wie in Abbildung 3 dargestellt erreicht werden kann [14].

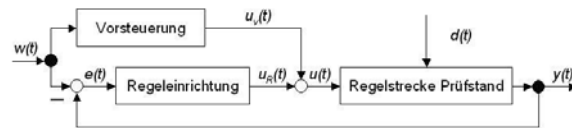


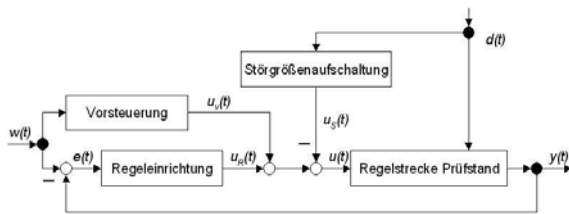
Abbildung 3: Prüfstandssteuerung als Regler mit Vorsteuerung

- $u_v(t)$  Anteil der Stellgröße aus der Vorsteuerung.
- $u_R(t)$  Anteil der Stellgröße aus der Regelung

Damit wären die wünschenswerten Eigenschaften der Robustheit und Störkompensation der Regelung mit der schnellen Umsteuerung kombiniert. Dabei sollte die Vorsteuerung so gewählt werden, dass die Prüfstandstrecke möglichst schnell der Führungsgröße  $w(t)$  nachgeführt wird. Die Regelung sorgt dafür, dass die Störung  $d(t)$  sowie die Wirkung der Modellunsicherheit ausgeglichen wird.

### Regler mit Vorsteuerung und Störgrößenaufschaltung

Als Erweiterung der Prüfstandsregelung bietet sich eine Störgrößenaufschaltung an. Denn bei der Prüfstandsregelung ist die Störgröße bekannt. Die Störgröße besteht aus den berechneten Fahrwiderständen. Diese Information kann also als Stellgröße benutzt werden und fließt nicht erst in die Regelung ein, wenn die Auswirkungen am Ausgang der Regelstrecke messbar sind. Ein Regler mit Störgrößenaufschaltung ist in Abbildung 4 dargestellt.



**Abbildung 4:** Prüfstandsteuerung als Regler mit Vorsteuerung und Störgrößenaufschaltung

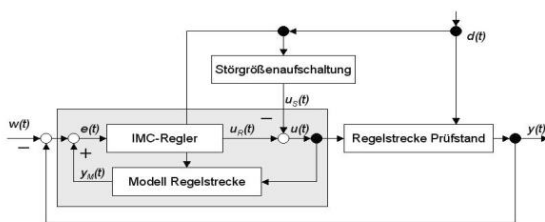
$u_s(t)$  Anteil der Stellgröße, welcher über die Störgröße kompensiert wird

Beim Regler mit Störgrößenaufschaltung, muss der Regler nur während des dynamischen Übergangsvorgangs eingreifen, denn die Störgröße wird vollständig über die Störgrößenaufschaltung kompensiert.

### IMC-Regler

Als weitere Erweiterung bietet sich an, die Prüfstandsregelung als IMC-Regler (Internal Model Control) [13] auszuführen. Bei den bisher beschriebenen Regelungen würden die im Modell enthaltenen Informationen über das Verhalten der Regelstrecke in die Wahl der Regelparameter nicht eingehen und die Regleinrichtung würde das Modell nicht enthalten.

Es bietet sich jedoch an, das Prüfstandsmodell als Modell im Regelkreis zu verwenden und als Regelparameter die Antriebsstrangparameter zu verwenden. Der innere Regelkreis kann dynamische Eigenschaften haben und die Regelstrecke realistisch als Modell abbilden, so dass eine Vorsteuerung dadurch überflüssig wird. Die Prüfstandsteuerung als IMC-Regler ist in Abbildung 5 dargestellt.



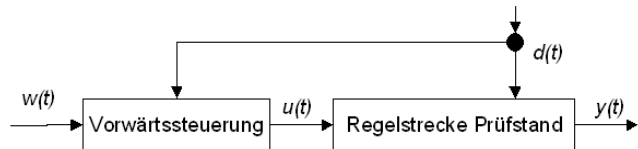
**Abbildung 5:** Prüfstandsteuerung als IMC-Regler

$y_M(t)$  Regelgröße des Modells der Regelstrecke (Prüfstandsmodell).

Würde nun das Prüfstandsmodell das gleiche Verhalten aufweisen wie der Prüfstand,

ModellRegelstrecke= RegelstreckePruefstand

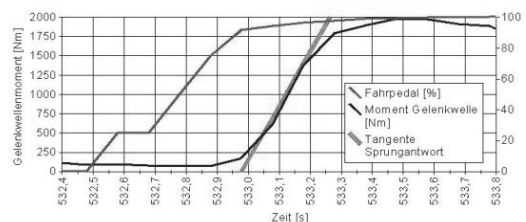
so würde die äußere Regelabweichung  $y(t) - w(t)$  verschwinden, da der innere Regelkreis die Regelabweichung komplett kompensiert. Der innere Regelkreis könnte somit als Vorwärtssteuerung betrachtet werden. Die Störgrößenaufschaltung bleibt natürlich wie beim vorherigen Modell erhalten. Dieser Aufbau ist in Abbildung 6 dargestellt.



**Abbildung 6:** Prüfstandsteuerung als Vorwärtssteuerung

### 2.3 Auswahl des geeigneten Regelverfahrens

Wie bereits in Abschnitt 2.2.1 beschrieben, erfüllt das vorliegende System nicht alle Anforderungen die an eine stabile Regelung gestellt werden. Vor allem die großen Totzeiten des Systems stellen eine besondere Herausforderung dar. Hierbei gelten Systeme bis zu einem Verhältnis von Totzeit zur Anstiegszeit in der Größenordnung von 0,3 noch als regelbar [15]. Das Verhältnis beim vorliegenden System ist teilweise über 1. Das bedeutet, dass dieses System mit herkömmlichen Methoden nicht regelbar ist.



**Abbildung 7:** Sprungantwort: Gelenkwellenmoment als Reaktion auf einen Fahrpedalsprung

In Abbildung 7 ist die Antwort des Prüfstandes auf eine schnelle Gaspedaländerung von Null auf Volllast dargestellt. Obwohl es sich eigentlich um ein rampenförmiges Signal handelt, wird im Folgenden von Sprungantwort gesprochen, da die Änderungsgeschwindigkeit sehr hoch ist. Zur Beurteilung der Sprungantwort wird nicht die Geschwindigkeit welche die Regelgröße darstellt verwendet, sondern das Gelenkwellenmoment. Die Geschwindigkeit wäre in diesem Falle nicht sehr aussagekräftig, da sie eine integrierte Größe ist.

Bei einer Vollastbeschleunigung ist wichtig, mit welcher Beschleunigung der Antriebsstrang eine bestimmte Zielgeschwindigkeit erreicht. Das Gelenkwellenmoment wird als Regelgröße verwendet, da es im direkten Zusammenhang mit der Beschleunigung steht und später entscheidend in die Lebensdauerberechnung der Bauteile eingeht. Das Verhältnis zwischen Totzeit und Anstiegszeit der Sprungantwort in Abbildung 7 wird folgendermaßen berechnet.

$$\begin{aligned} TT_{\text{Totzeit}} &= 532,97 \text{ [s]} - 532,65 \text{ [s]} \\ T_{\text{Anstiegszeit}} &= 533,26 \text{ [s]} - 532,97 \text{ [s]} \\ TT_{\text{Totzeit}}/T_{\text{Anstiegszeit}} &= 1,1 \end{aligned}$$

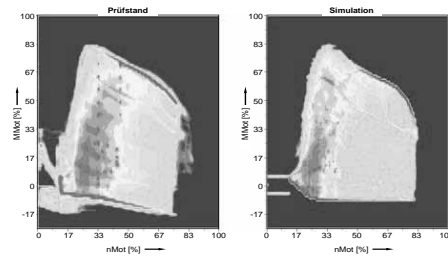
Das Verhältnis 1,1 ist viel zu groß und erlaubt keine Prüfstandssteuerung mit einem herkömmlichen Regler.

### 3 Validierung der Regelungsgüte

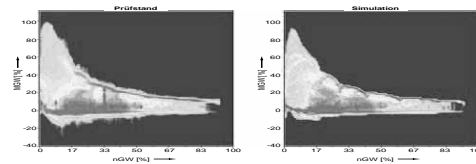
Die Regelungsgüte wird anhand statistischer Verfahren und deren Kenngrößen durchgeführt. Hierbei werden die Prüflaufdaten eines kompletten Dauerlaufs ausgewertet. Dieser Dauerlauf besitzt eine Gesamtlänge von mehreren 10.000 km und stellt belastungsseitig einen durchmischten Kundenbetrieb unter verschärften Bedingungen dar.

#### 3.1 Verweildauerkollektive

Grundsätzlich zeigen die Verweildauerkollektive von Motor und der Gelenkwelle sehr gute Übereinstimmung zwischen der Simulationsvorgabe und der Prüfaufbaureaktion. Hierbei ist in Abbildung 8 u. 9 auf der x-Achse die zugehörige Drehzahl und auf der y-Achse das zugehörige Moment aufgetragen, die Farbgebung kennzeichnet die Verweildauer der jeweiligen Lastpunkte in logarithmischer Darstellung. Für die Bewertung ist die Farbgebung entscheidend, aus der die Häufigkeiten der gefahrenen Lastpunkte hervorgeht. Je roter die Farbgebung, desto häufiger werden die Lastpunkte angefahren, je grüner desto seltener, blaue Bereiche liegen außerhalb des Kennfelds und werden somit überhaupt nicht gefahren. Aus dem direkten Vergleich kann beispielsweise festgestellt werden, dass die Modellierung des Schubmomentenbereichs des Motors, vgl. Abbildung 8 der Simulation verbesserungsfähig ist.



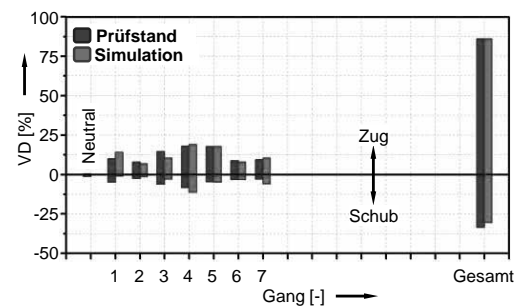
**Abbildung 8:** Verweildauerzählung der Motordrehzahl ( $n_{\text{Mot}}$ ) und -moment ( $M_{\text{Mot}}$ )



**Abbildung 9:** Verweildauerzählung der Gelenkwellendrehzahl ( $n_{\text{GW}}$ ) und -moment ( $M_{\text{GW}}$ )

#### 3.2 Gangverteilung

Die Gangverteilung zeigt die Verweildauer im jeweiligen Gang während des Prüflaufs von der Simulationsvorgabe und der Prüfaufbaureaktion. Hierbei werden Abweichungen im Schaltprogramm zwischen der Simulationsvorgabe und der Prüfaufbaureaktion deutlich. Grundsätzlich zeigen die Gangverteilungen brauchbare Übereinstimmung, vor allem unter dem Gesichtspunkt, dass die Anteile der Neutralstellung und des ersten Gangs des Prüfaufbaus addiert werden müssen, um die Verweildauern des ersten Gangs der Simulationsvorgabe zu erhalten. Der Grund hierfür liegt am Stopp/Start-System, bei dessen Implementierung in der Simulation der erste Gang eingelegt bleibt, der reale Aufbau aber auf Neutral schaltet. Ansonsten sind größere Abweichungen im 3. Gang festzustellen, vgl. Abbildung 10. Dies erklärt sich durch geänderte Softwarestände beim Prüfaufbau ohne Anpassung der Parametrierung des Simulationsmodells.



**Abbildung 10:** Gangverteilung Verweildauer (VD)

### 3.3 Überrollungsschädigung

Bei der Überrollungsschädigung werden wegen des logarithmischen Zusammenhangs zwischen Lastamplitude und Schädigung Unterschiede der Signalgrößen verstärkt erkennbar. Im Gegensatz zu den vorangegangenen Darstellungen wird hier eine relative Auswertung verwendet, in der die 100 %-Linie Schädigungsgleichheit zwischen der Simulationvorgabe und der Prüfaufbaureaktion darstellt. Werte größer 100 % bedeuten, dass das Prüfkollektiv schärfer als die Vorgabe ist und umgekehrt. Nennenswerte Unterschiede sind ausschließlich im 7. Gang festzustellen, vgl. Abbildung 11. Gemessen am Gesamtkollektiv werden die Unterschiede als unbedeutend bewertet.

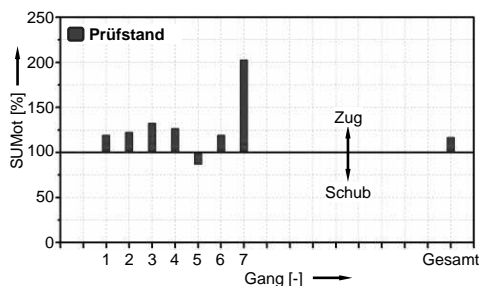


Abbildung 11: Überrollungsschädigung am Motor (SUMot) in relativer Darstellung

## 4 Weiterentwicklung der Prüfstandsteuerung

Der Vergleich der Modelldaten mit der Regelstrecke zeigt, dass zwar Abweichungen existieren, diese sind jedoch ausreichend klein. Entscheidend für die Güte der Prüfstandssimulation im Alltagsbetrieb ist, dass die Modelldaten einfach und kostengünstig ermittelt werden können.

Im ersten Schritt können die Modelldaten auf Grund der beim Automobilhersteller umfangreich ermittelten Komponentenmessungen angegeben werden. Es gelingt daher, ein brauchbares Startmodell zu erzeugen. Durch standardisierte Lernzyklen, die auf dem Prüfstand gefahren werden, werden die Unterschiede zwischen Modell und Regelstrecke erkannt und die Modelldaten ggf. korrigiert. Insbesondere der für die Simulationsgüte sehr wichtige Zusammenhang zwischen Pedalwert und Drehmoment wird auf diese Weise sehr einfach und realistisch ermittelt.

## 5 Ausblick und Zusammenfassung

Die hier beschriebene Prüfstandssteuerung mittels Online-Simulation wurde im Jahr 2007 als Prototyp für einen bestimmten Triebstrang eingeführt. In der Zwischenzeit hat sich dieses System bewährt und wird auf verschiedensten Ein- und Mehrmaschinenprüfständen für unterschiedlichste Triebstränge eingesetzt.

Die in Abschnitt 3 und 4 beschriebene Validierung des Vorgabesystems zwischen der Simulationvorgabe und der Prüfaufbaureaktion zeigt das Ergebnis ohne die Verwendung entsprechender Lernzyklen. Die Erweiterung der vergleichsweise einfachen Vorwärtssteuerung, durch die Rückführung von Zustandsgrößen des Prüfaufbaus in den IMC-Regler, reduzieren nochmals deutlich die hier diskutierten Unterschiede. Elementar hierbei ist, dass sich die Simulation dem Verhalten des realen Aufbaus, ohne dessen Betriebsstrategien zu verändern, automatisiert anpasst.

Durch die Entwicklung eines firmeninternen Vorgabesystems für die Antriebsstrangprüfstände unter Berücksichtigung, deren Anforderungen und dessen Integration in den Arbeitsablauf, konnten nicht nur bedeutend bessere Erprobungsergebnisse für die Antriebsstrangkomponenten erzielt werden, sondern auch Effizienzen durch deutlich reduzierte Abschaltungen gehoben werden.

Da in der Zwischenzeit diese Art der Prüfstandssteuerung sehr erfolgreich ist, wurden verschiedenste Schnittstellen zu anderen Prüfstandsdienstleistern aufgebaut.

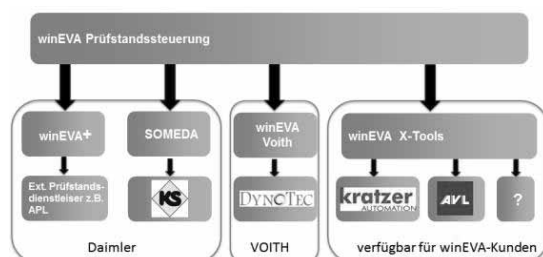


Abbildung 12: Übersicht Prüfstandssteuerung mit winEVA

## 6 References

- [1] Klos, W.; Schenk, M.; Schwämmle, T.; Müller, M.; Bertsche, B.: Antriebsstrangerprobung bei der Daimler AG, Internationales Symposium für Entwicklungsmethodik, Stuttgart 29.8.2011
- [2] Willmerding, G.; Häckh, J.: Ein System zur Untersuchung von Fahrzeugantrieben zur Vorhersage von Fahrleistung, Kraftstoffverbrauch und Lebensdauer unter realistischen Einsatzbedingungen, Vortrag auf der VDI-Tagung Erprobung und Simulation in der Fahrzeugentwicklung, 24.6.-25.6.2009, Würzburg
- [3] Häckh, J.: Untersuchung und Optimierung der Steuerung von Prüfständen zur Streckensimulation von Pkw-Antriebssträngen unter realistischen Einsatzbedingungen, Masterarbeit an der Hochschule Ulm, Fakultät Informatik, Nov. 2007.
- [4] Greiner, J., Dörr, C., Klos, W., Schwämmle, T.: Lastkollektive 7-Gang Automatikgetriebe W7A700, durchgängige Bewertung und Betrachtung im Entwicklungsprozess bei Mercedes-Benz, Getriebetagung Friedrichshafen, VDI-Berichte
- [5] Klos, W.: Gruppenschaltungsansteuerung von Nutzfahrzeuggetrieben, Dissertation 2004, Universität Stuttgart, Institut für Maschinenkonstruktion und Getriebebau Bericht Nr. 508, ISBN 3-922 823-60-2
- [6] Körner, T.; Depping, H.; Häckh, J.; Willmerding, G.: Fatigue Life Prognosis for Transmissions based on critical Component Spectrum; World Automotive Congress FISITA 2002, Helsinki, Paper Nr.F02V091
- [7] Körner, T.; Depping, H.; Häckh, J.; Willmerding, G.; Klos, W.: Rechnerische Lebensdauerabschätzung unter Berücksichtigung realer Belastungskollektive für die Hauptwelle eines Nutzfahrzeuggetriebes, DVM-Tagung Fulda vom 5. bis 6.6. 2002, VDI-Berichte N2. 1689, 2002 Seite 275 - 285
- [8] Willmerding, G.: A Simulation System to study the Working Conditions of Vehicles and to develop fuel efficient Drivetrains, Publication on the FISITA-Congress 1992, Institution of Mechanical Engineers, London 1992
- [9] Willmerding, G.: Ein Simulationsmodell für den Autobahnverkehr, Teil 1; Automobiltechnische Zeitschrift Heft 5/1992
- [10] Willmerding, G.: Ein Simulationsmodell für den Antriebsstrang, Teil 2. ATZAutomobiltechnische Zeitschrift, Heft 6, 1992
- [11] Willmerding, G.; Jehlicka, E.: Verbrauchsuntersuchungen an Stadtbussen, Zeitschrift der Nahverkehr 1986, Heft Seite
- [12] Willmerding, G.; Hoffmann, M.; Rasch, W.; Schicker, R.: Ermittlung der Einsatzverhältnisse militärischer Radkraftfahrzeuge durch Auswertung militärischer Übungen; Teil 4: Erfassung der Einsatzverhältnisse hinsichtlich Beanspruchungen bei der Übung "Große Rochade", Battelle-Bericht 1976
- [13] Jan Lunze. Regelungstechnik 1, Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelung. Springer –Verlag Berlin, 6., korrigierte Auflage, 2007.
- [14] G. Schlüter. Regelung technischer Systeme – interaktiv. Carl Hanser Verlag, München 2001.
- [15] ABB Automation Production. Einführung in die Regelungstechnik. Technische Information, 30/60-110 DE, 1(1):3, April 2003
- [16] W. Beitz und K.-H. Küttner. Dubbel Taschenbuch für den Maschinenbau. Springer –Verlag Berlin, 17, neu bearbeitete Auflage, 1990.

# Systemkonzept eines modularen HiL-Systems für modellbasierte Funktionsentwicklung fahrzeugmechatronischer Systeme

Xiaobo Liu-Henke<sup>1</sup>, Marian Göllner<sup>1</sup>, Robert Buchta<sup>2</sup>, Florian Quantmeyer<sup>2</sup>

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

<sup>2</sup>Volkswagen...

*x.liu-henke@ostfalia.de*

In der vorliegenden Arbeit wird ein modularer Closed-Loop Fahrsimulator mit Hilfe des mechatronischen Entwicklungsprozesses konzipiert. Mittels dieses Fahrsimulators kann der reale Fahrereinfluss im Regelkreis Fahrer-Fahrzeug-Umwelt berücksichtigt werden. Darüber hinaus ermöglicht dieser Fahrsimulator eine Echtzeitsimulation des Gesamtfahrzeugs mit sechs Freiheitsgraden sowie die Integration vorhandener HiL-Prüfstände über definierte Echtzeitschnittstellen. Ebenso können Sicherheitsfunktionen in kritischen Fahrsituationen gefahrenlos mit realem Fahrer validiert werden. Der Simulator ist so ausgelegt, dass er ebenfalls als Anregungseinheit zur Erprobung weiterer Fahrdynamiksysteme genutzt werden kann. Diese Eigenschaft wird beispielsweise bei der Entwicklung aktiver Fahrzeugfederungen genutzt.

## 1 Einleitung und Motivation

Eine modellgestützte Funktionsentwicklung basierend auf Model-in-the-Loop, Software-in-the-Loop und Hardware-in-the-Loop mit anschließenden Fahrversuchen ist der Standard in der Automobilindustrie. Neben der objektiven Beurteilung der entwickelten Systeme anhand definierter Kriterien spielt auch die subjektive Beurteilung verschiedener Fahrer eine wichtige Rolle. Problematisch ist hieran, dass vor den Fahrversuchen im realen Fahrzeug während der Entwicklung kein Feedback vom Fahrer berücksichtigt werden kann. Dies wiederum führt zu einer großen Anzahl Iterationsschritte und somit zu hohen Entwicklungskosten.

Um den fehlenden Schritt zwischen Hardware-in-the-Loop-Simulation und Fahrversuch zu schließen, soll ein modular erweiterbarer Fahrsimulator zur Durchführung von Closed-Loop-Simulationen entwickelt werden. Dieser Fahrsimulator besteht aus einer Bewegungsplattform und einem Hexapoden mit sechs Freiheitsgraden, der eine realistische Nachbildung der Dynamik des Fahrzeugaufbaus ermöglicht. An die Informationsverarbeitung des Hexapods wird ein dSPACE-Echtzeitsystem angekoppelt, welches die Steuer- und Regelaufgaben auf globaler Ebene übernehmen kann.

Es wird ein Konzept entwickelt, das zum einen die Abbildung der Anregung der Fahrzeugbewegung mit

sechs Freiheitsgraden erlaubt und darüber hinaus den Fahrereinfluss einbezieht und somit realistische Closed-Loop-Manöver durch alle Ebenen des Entwicklungsprozesses ermöglicht.

## 2 Grundlagen der Fahsimulation

Um eine Fahrsimulation aufzubauen ist es unabdinglich zuerst zu verstehen, wie der Fahrer seine Umgebung während der Fahrt wahrnimmt und auf sie reagiert. In Abbildung 1 ist das Gesamtsystem Fahrer-Fahrzeug-Fahrumgebung dargestellt. Der Fahrer wirkt über Lenkung, Bremsen und Antrieb auf das Fahrzeug. Dieses reagiert auf die Fahrervorgabe mit definierter Dynamik und Fahrverhalten. Aus der Fahrumgebung wirken verschiedene Störungen und Vorgaben, die unterschiedliche Aspekte des Regelkreises beeinflussen, auf Fahrer und Fahrzeug. So wird die Navigation des Fahrers anhand des ihm bekannten Straßennetzes durch Umleitungen geschwächt. Witterung und Straßenschäden wirken signifikant auf das Fahrzeug und beeinflussen die Fahrdynamik und Fahrsicherheit in allen sechs Freiheitsgraden. Der Fahrer nimmt mittels optischer, akustischer, haptischer sowie vestibulärer Signale Informationen über den Fahrzustand auf und passt darauf basierend in seiner Funktion als „Regler“ durch die ihm zur Verfügung stehenden Stellgrößen das Verhalten des Fahrzeugs an [1].

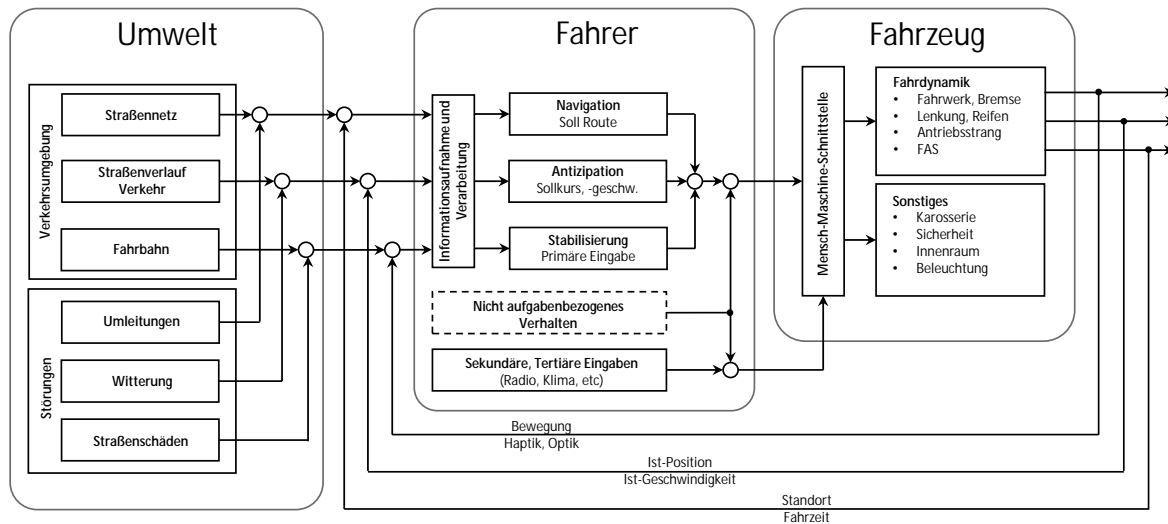


Abbildung 1. Regelkreis aus Umgebung, Fahrer und Fahrzeug

Um eine Fahrsimulation aufzubauen, müssen diese Signale künstlich nachgebildet werden. Es ist also das Ziel zunächst optische und akustische Signale z.B. mit modernen Grafiksystemen und Surround Audioanlagen, zu imitieren. Durch die stationäre Art des Simulators ist die Abbildung vestibulärer Signale dagegen schwieriger. Das Regelsystem der Bewegungsplattform muss Beschleunigungskräfte mithilfe der Erdfallbeschleunigung nachbilden, um ein realistisches Fahrgefühl zu vermitteln [2]. Ergo sind besonders hohe Anforderungen an die Leistungsfähigkeit des aktorischen System, der Visualisierung, des Fahrdynamikmodell und der haptischen Bedienelemente gestellt, welche im komplexen Wechselspiel die Fahrsimulation bilden. Es wird eine Methode benötigt, mit der diese hohen Anforderungen umgesetzt werden können.

### 3 Mechatronische Entwicklungsmethodik

Zur Entwicklung des hoch komplexen Closed-Loop Fahrsimulators wird die mechatronische Entwicklungsmethodik herangezogen. Zur Gewährleistung eines systematischen Entwurfs ist die Beherrschung der Systemkomplexität elementar. Nach [3] hat sich hierbei folgende mechatronische Methodik mittels Modularisierung und Hierarchisierung bewährt: „Ein komplexes System wird mittels „Top-down-Verfahren“ in Teilsysteme mit mechatronischen Komponenten, sog. Funktionsmodule aufgeteilt.“

Zum Entwurf jedes Funktionsmoduls wird der in Abbildung 2 dargestellte mechatronische Entwick-

lungskreislauf herangezogen. Am Beginn der theoretischen Untersuchung steht die Modellbildung. Das technische System wird nach Funktionsprinzipien im Hinblick auf die Anforderung in ein physikalisches bzw. mathematisches Ersatzmodell aus unterschiedlichen Fachdisziplinen wie Mechanik, Hydraulik und Elektronik abgebildet. Es repräsentiert das Systemverhalten. Die Regelstrategie, die vom einfachen Regler bis hin zu hierarchisch angeordneten Mehrgrößenregelstrukturen aufgebaut sein kann, wird anhand des Systemverhaltens festgelegt.

Die anschließende modellbasierte Komposition erfolgt gemäß des Rapid Control Prototypings (RCP) in einem durchgängig verifikationsorientierten Prozess aus Model-in-the-Loop (MiL), Software-in-the-Loop (SiL) und Hardware-in-the-Loop (HiL) [4].

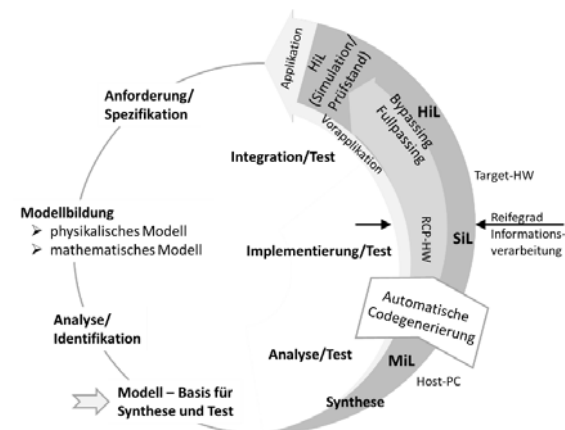


Abbildung 2. Mechatronischer Entwicklungskreislauf

## 4 Systemkonzept

Es ist zunächst notwendig zu spezifizieren welche Anforderungen genau an den Simulator gestellt werden. Anschließend wird, basierend auf diesen Anforderungen, ein Konzept für den Fahrsimulator auf Basis der mechatronischen Komposition entwickelt. Es folgt dann die exemplarische Modellbildung des Simulators und des Fahrdynamikmodells. Abschließend wird die resultierende Struktur der Echtzeit-Datenverarbeitung vorgestellt.

### 4.1 Einsatzaspekte des Fahrsimulators

Zur Entwicklung und Erprobung mechatronischer Systeme wie Fahrdynamikregelsysteme, Fahrkomfortsysteme und auch Fahrerassistenzsysteme ist eine realitätsnahe Erprobung unter Einbeziehung des Fahrers und komplexer Verkehrsszenarien notwendig. Eine Erprobung dieser Systeme ist im Fahrversuch oft aus Sicherheitsgründen nicht vollständig möglich. Der geplante Fahrsimulator soll dem Fahrer mit Hilfe einer Fahrsimulation ermöglichen Closed-Loop Manöver im vollen Spektrum der physikalisch möglichen Fahrdynamik zu untersuchen und somit die reaktionen von Regelsystemen auch in Grenzbereichen zu testen.

Neben der Horizontaldynamik soll eine realistische Nachbildung der Vertikaldynamik durch ein detailliertes Modell des Fahrzeugaufbaus erfolgen. Hierdurch können die verschiedenen Reglerfunktionen unter Echtzeitbedingungen integriert und mittels Gesamtfahrzeugsimulation unter realistischen Bedingungen erprobt sowie optimiert werden.

Es soll eine Kombination des vorhandenen HiL-Systems „E-Traktion“ mit dem Fahrsimulator und damit die Erweiterung der Einsatzmöglichkeiten von aktuellen Open-Loop-Manövern zu Closed-Loop-Manövern durch Einbindung eines realen Fahrers vorgenommen werden. Somit können die Energieflüsse im elektrischen Antriebstrang realitätsnah nachgebildet werden.

Weiterhin soll auch der vorhandenen HiL-Prüfstand „Aktive Fahrzeugfederung“ in den Fahrsimulator integriert werden. Hierzu soll die Bewegungsplattform des Simulators mit sechs Freiheitsgraden als Anregungseinheit für das Fahrzeugmodell dienen.

Zur Untersuchung von Fahrerpräferenzen soll mithilfe haptischer Bedienelemente ein realitätsnahes Fahrerverhalten nachgebildet werden. Es können einflüsse der MMI auf kognitive Funktionen des Fahrers

analysiert und eine Personalisierung fahrzeugmechatronischer Systeme vorgenommen werden. Ausserdem kann die Mensch-Maschine-Schnittstelle (aktive Haptik) in der Elektromobilität mit Fokus auf das kooperative Bremsen (Steuerung der Rekuperationsintensität durch den Fahrer) untersucht werden.

Allgemein ist so die realitätsnahe Erprobung verschiedenster Regelsysteme und Failsafe-Funktionen, die sich auch mit einem prototypischen Fahrzeug nicht oder nur in Verbindung mit Gefährdungssituationen darstellen lassen würden, auch in Grenzbereichen möglich.

### 4.2 Resultierende Hardwarekomponenten

Aus den vorliegenden Anforderungen wird ein Konzept entwickelt, das zum einen die Abbildung der Anregung der Fahrzeugbewegung mit sechs Freiheitsgraden ermöglicht und darüber hinaus den Fahrereinfluss einbezieht und somit realistische Closed-Loop-Manöver durch alle drei Ebenen (MiL, SiL und HiL) des Entwicklungsprozesses darstellen kann [5]. Das Konzept sieht eine Integration in die bestehende HiL-Technologie vor, wobei die Bewegungsplattform auch als Anregungseinheit genutzt wird.

Abbildung 3 stellt den Fahrsimulator und seine Integration in die bestehende HiL-Technologie dar. Der Fahrsimulator gliedert sich in fünf Module:

- *Bewegungsplattform mit Hexapod* zur Nachbildung der Einflüsse der Fahrdynamik und Fahrzeuganregung in sechs Freiheitsgraden. Die Regelung dieses Moduls übernimmt das lokale Regelsystem des Hexapoden.
- *MMI Interface mit haptischen Bedienelementen* zur Nachbildung eines realitätsnahen Fahrerumfelds. Dieses Modul wird von Human-Machine-Interface Reglern koordiniert.
- *Echtzeitinformatonsverarbeitung*: Hier werden leistungsfähige RCP-Systeme der Firma dSPACE eingesetzt, um eine Echtzeit-Kommunikation mit anderen angehängten Systemen aufbauen zu können.
- *Visualisierung (Sichtsimulation)*: Ein separater Visualisierungsrechner soll sich nur um das Rendering der Graphischen Umfeldausgabe kümmern. Die gerenderten Einzelbilder werden dann an Bildschirme vor dem Fahrer gesendet.
- *Koordinationssystem*: Um die Module miteinander zu synchronisieren und Nutzereingriffe zuzulassen, ist ein Master in Form des Leitstellenrechners notwen-

dig. Dieser koordiniert auch die Kommunikation aller Module über die Ethernetschnittstelle.

rer Latenz über vorhandene I/O-Boards der dSPACE-Echtzeitverarbeitung eingelesen werden. Insbesondere

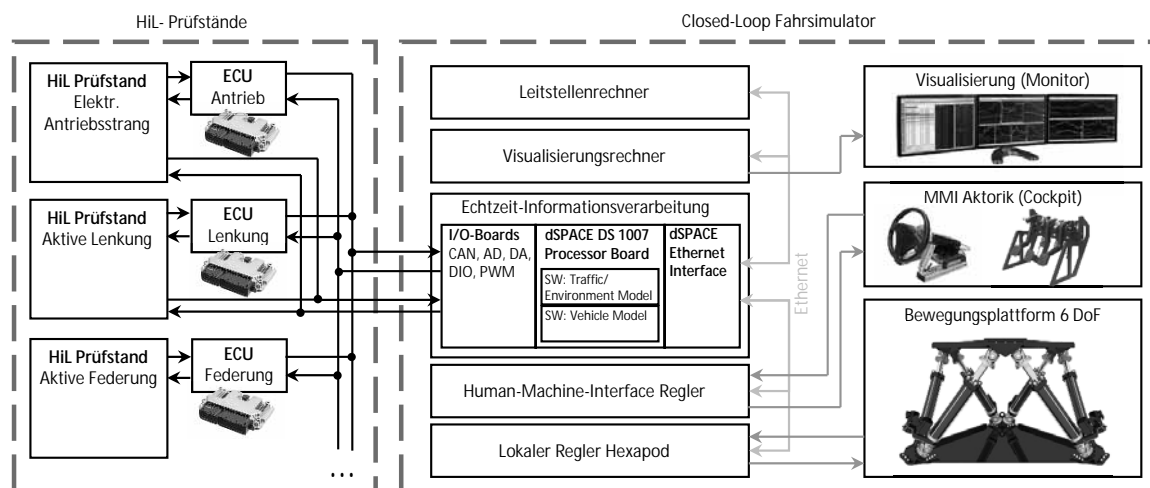


Abbildung 3. Mechatronische Strukturierung

Die Echtzeithardware des Fahrsimulators und die Systeme anderer HiL-Prüfstände lassen sich über vordefinierte, offene Schnittstellen kombinieren und flexibel konfigurieren. Über die Echtzeitanbindung der dSPACE Hardware ist eine schnelle Kommunikation zwischen Fahrsimulator und HiL-Prüfständen gegeben.

Durch die Kopplung der vorhandenen HiL-Prüfstände mit dem Fahrsimulator und dem realen Fahrer soll eine möglichst vollständige Repräsentation des Gesamtverhaltens und der Wechselwirkungen von Fahrer-Fahrzeug-Fahrumgebung erreicht werden [6].

### 4.3 Struktur der Echtzeitdatenverarbeitung

Der Echtzeitprozess zur Berechnung des Fahrzeughaltens, des Verkehrs und der Umwelt erfolgt auf Basis der vorhandenen Automotive Simulation Models (dSPACE ASM) mit einer Visualisierung über MotionDesk. In den Prozess können, ergänzend zum ASM, eigene Modellbibliotheken mit alternativen Fahrzeugmodellen, unterschiedlichen Antriebstechnologien und diversen Antriebsstrangkfigurationen einbezogen werden. Die Schnittstelle, über die die Führungsgrößen an den Hexapod gesendet und Feedbackgrößen von der Bewegungsplattform erhalten werden, ist durch eine Ethernetverbindung realisiert. Diese Schnittstelle dient auch zur Betriebsartensteuerung und Überwachung der Bewegungsplattform. Weiterhin können die Ein- und Ausgabesignale der Mensch-Maschine-Schnittstelle über diese Ethernetverbindung ausgetauscht oder alternativ mit geringe-

re bei der Funktionsauslegung der Haptikaktorik wird diese Signalverbindung bevorzugt. Die offenen Schnittstellen dienen zur Integration von vorhandenen HiL-Prüfständen.

### 4.4 Herleitung des Fahrdynamikmodells

Das Fahrzeugmodell, das für die Nachbildung der Fahrdynamik im Simulator eingesetzt wird, ist ein auf physikalischen Gegebenheiten basierendes mathematisches Ersatzmodell. Es soll das kinematische, statische und dynamische Verhalten des nachgebildeten Fahrzeugs repräsentieren. In einem durchgängigen Entwicklungsprozess wird dieses Modell von Beginn an zur Analyse, Synthese und zum Test mittels MiL, SiL und HiL verwendet.

Abbildung 4 skizziert die Struktur des Fahrzeugmodells. Diese besteht aus dem Fahrzeugaufbau und vier Radmodulen. In den Radmodulen werden die zwischen Fahrbahn und Reifen wirkenden Kräfte über ein semi-empirisches Reifenmodell (Magic-Formula) ermittelt. Die Dynamik der eingesetzten Aktoren wird über die entsprechenden mechatronischen Funktionsmodule (MFM) beachtet.

Darauf aufbauend wird die räumliche Bewegung des Fahrzeugs unter Berücksichtigung der Kinematik berechnet. Anschließend wird ein nichtlineares MKS-Modell ganzheitlich nach Funktionsprinzipien entwickelt. Das Gesamtfahrzeugmodell umfasst das kinematische, statische und dynamische Verhalten des Fahrzeugs mit seinen Teilsystemen Reifen, MFM

Antrieb, MFM Federung und MFM Lenkung sowie dem Energiespeicher.

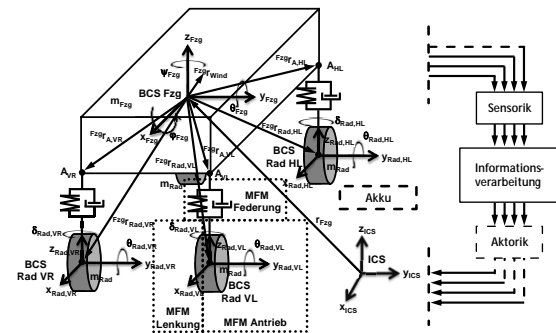


Abbildung 4. Modellstruktur des Fahrzeugmodells

Die auf den Aufbau wirkenden Kräfte werden dem Newton-Euler-Ansatz entsprechend für die einzelnen Freiheitsgrade in Differentialgleichungen überführt. Anschließend kann die Bewegung des Fahrzeugs vom körperfesten (BCS) in das ortsfeste (ICS) Koordinatensystem transformiert werden.

## 5 Realisierung und Anwendung

Die folgende Abbildung 5 zeigt den Fahrsimulator mit seinen Komponenten.

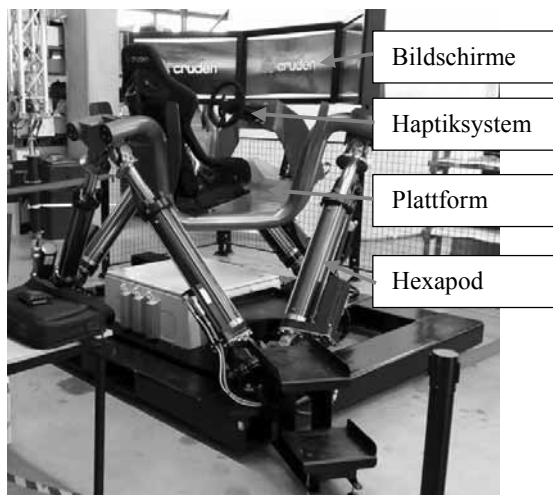


Abbildung 5. Closed-Loop Fahrsimulator

Die konkrete Funktionsweise des Simulators soll exemplarisch am Beispiel der „Aktive Federung“ näher erläutert werden.

Die Regelstruktur zur aktiven Fahrzeugfederung (ABC) hat zwei Ziele: verbesserte Fahrdynamik der drei modalen Aufbaubewegungen (Hub, Wanken, Nicken) und guter Fahrkomfort, der durch Isolation

des Fahrzeugaufbaus von den fahrbahninduzierten Schwingungen erreicht wird.

Durch die physikalischen Kopplungen werden durch die Hubbewegungen auch eine Wankbewegung und eine Nickbewegung des Aufbaus hervorgerufen. Kern der Regelung der Aufbaudynamik ist die Entkopplung der drei Freiheitsgrade des Aufbaus, um z.B. eine robuste Regelung für jeden Freiheitsgrad auszulegen, ohne, dass das Verhalten von anderen Freiheitsgraden beeinflusst wird. Die Isolation der fahrbahninduzierten Schwingungen und die verbesserte Dynamik erfolgen durch die aktive Fußpunktverstellung des Federbeins.

Bisher wird die Regelstruktur modellbasiert ausgelegt und am vorhandenen HiL-Prüfstand Aktive Federung mit einem Freiheitsgrad hinsichtlich der Vertikalbewegung erprobt. Die Hubbewegungen rufen jedoch beim Gesamtfahrzeug auch ein Wanken und ein Nicken des Aufbaus hervor. Folglich können die Algorithmen, z.B. die Entkopplung, am vorhandenen HiL-Prüfstand nicht unter Echtzeitbedingungen validiert werden. Eine weitergehende Optimierung kann mithilfe des Hexapods realisiert werden.

Zur Verkopplung des Hexapods und des HiL-Prüfstands „Aktive Federung“ sind grundsätzlich zwei Anwendungsfälle möglich:

Im vorhandenen HiL-Prüfstand „Aktive Federung“ sind derzeit die verschiedenen Aktoren sowie ein Teil der Regelstrecke (Räder, Aufbaumasse) als Realteile vorhanden. Diese werden auf den Hexapod an einer speziellen modular austauschbaren Bewegungsplattform installiert. Über verschiedene Sensoren werden fahrdynamische Größen erfasst und den Regelalgorithmen zur Verfügung gestellt. Der Regelalgorithmus berechnet die Stellgrößen und gibt sie an die Aktoren weiter. Die Bewegungsplattform regt dabei das System nicht nur durch Huben, sondern in allen sechs Freiheitsgraden an. Somit kann die Regelstruktur mithilfe des Fahrsimulators mit Hexapod für alle Freiheitsgrade unter Echtzeitbedingungen weitergehend optimiert und erprobt werden.

Anders herum können die Algorithmen der aktiven Federung auch in das Fahrdynamikmodell des Simulators geladen werden. Der Fahrer erfährt die Beschleunigung ähnlich wie sie auch im Fahrzeug auftritt. Somit kann eine subjektive Bewertung des durch die aktive Fahrzeugfederung verbesserten Fahrkomforts erfolgen. Gleichzeitig können Informationen über den Fahrbahn- und Fahrzeugzustand der aktuel-

len Simulation an den vorgestellten HiL- Prüfstand für aktive Federung weitergegeben werden. So ist es möglich mit dem Prüfstand die Regelung und die eingetzten Aktoren und Sensoren unter realen Bedingungen zu testen und nicht nur einen definierten Zyklus abfahren zu lassen.

## 6 Resümee

Im Zuge eines Forschungsprojektes wird ein modular erweiterbarer Fahrsimulator zur Durchführung von Closed-Loop-Simulationen entwickelt. Der Fahrsimulator verfügt über flexibel konfigurierbare und erweiterbare Schnittstellen zu einer Echtzeithardware der Firma dSPACE, sodass beliebige Systeme und Prüfstände ankoppelbar sind. Es sind fünf Anwendungsfälle vorgesehen:

1. Test von Regelsysteme in virtueller Umgebung mit Driver-in-the-Loop zur Durchführung von Closed-Loop-Manöver. Dabei ist der Fahrer über die MMI-Schnittstellen des Simulators mit der Simulation verbunden.
2. Visualisierung (Sichtsimulation) von Zuständen einer Open-Loop Simulation über einen Visualisierungsrechner, der die Simualtionsergebnisse als realitätsnahes, gerendertes Bild darstellen kann.
3. Einsatz leistungsfähigster Echtzeitsysteme der Fa. dSPACE zur schnellen Echtzeitverarbeitung. So können viele zuvor nur in einer Simulation abgedeckte Komponenten in der Realität gemeinsam mit dem Einflusss des Fahrers betrachtet werden.
4. Nutzung des Hexapoden als Anregungseinheit in die bestehende HiL-Technologie. Zu diesem Zweck kann die Bewegungsplattform des modular aufgebauten Simualtors ausgetasucht werden um die Anbindung an z.B. einen Prüfstand für aktive Federung zu realisieren.
5. Untersuchung von Fahrerpräferenzen mithilfe von haptischen Bedienelementen zur Nachbildung eines realitätsnahen Fahrerverhaltens. Es können einflüsse der MMI auf konginitive Funktionen des Fahrers analysiert und eine Personalisierung fahrzeugmechatronischer Systeme vorgenommen werden.

## Danksagung

Diese Arbeit wurde durch die Fördeung des DFG, Antrag Großgeräte des Landes Niedersachsen Ermög-

licht. Für die Förderung bedanken sich die Autoren herzlichst.

## 7 Referenzen

- [1] H. Proff. *Radikale Innovationen in der Mobilität: Technische und betriebswirtschaftliche Aspekte*. Springer Verlag, Duisburg, 2013
- [2] W. Tomaske und M. Meywerk. *Möglichkeiten zur Vermittlung von subjektiven Fahreindrücken mit Fahrsimulatoren*. Beitrag in: Brill, U. (Hrsg.): *Subjektive Fahreindrücke sichtbar machen III*. Expert Verlag, Renningen, 2006
- [3] X. Liu-Henke, R. Buchta, F. Quantmeyer. *Simulation eines mechatronischen Lenkungsmoduls für ein Elektrofahrzeug mit dezentralen Direktantrieben*. ASIM STS/GMMS Workshop 2011 - Simulation Technischer Systeme, Krefeld, 2011.
- [4] R. Buchta. *Mechatronische Entwicklung eines Forschungselektrofahrzeugs zur Erprobung von Fahrdynamikregelungen und Fahrerassistenzsystemen*. Dissertation, Universität Magdeburg, 2016
- [5] U. Kiencke und L. Nielsen. *Automotive Control Systems*. Springer Verlag, Berlin Heidelberg New York, 2005
- [6] A.Schmidt. *Modellierung von Fahrzeugantrieben anhand von Messdaten aus dem Kopplbetrieb zwischen Fahrsimulator und Antriebsstrangprüfstand*. Springer Verlag, Stuttgart, 2015.

# Effiziente Absicherung von Systemen der Fahrzeugelektronik mit System-HiLs in der Elektronikintegration

Dr.-Ing. Robert Buchta

Elektronikentwicklung, Volkswagen AG, Wolfsburg

*robert.buchta@volkswagen.de*

Dieser Beitrag zeigt auf, wie mit System-HiLs als Ergänzung die Integration der Fahrzeugelektronik effizient abgesichert wird. Anhand einer mechatronischen Entwurfsmethodik werden zunächst die domänenspezifischen Funktionen hierarchisch strukturiert. Damit wird eine Übersichtlichkeit zusammen mit allen Schnittstellen in horizontaler und vertikaler Richtung geschaffen. Bei der Konzeption von System-HiLs gilt es den grundsätzlichen Zielkonflikt aus hoher Testabdeckung und geringer Komplexität der Prüfinfrastruktur zu entschärfen. Systematisch wurde hierzu die Methodik erweitert, um einen Kompromiss hinsichtlich einer effizienten Absicherung zu finden. Im Kontext eines Vielmarkenkonzerns mit markenübergreifenden Bauteilbaukästen wird aufgezeigt, wie die Durchgängigkeit und Testabdeckung von Varianten mit dem Vorgehen systematisch bereits in frühen Phasen erhöht wird.

## 1 Einleitung

Steigende Komplexität, kürzere Entwicklungszyklen und eine hohe Variantenvielfalt erfordern Absicherungsprozesse für elektrische Funktionen in Fahrzeugen, die ein Frontloading für eine frühe Fehlerfindung und eine hohe Durchgängigkeit zur Zeit- und Kosteneffizienz sicherstellen. Dazu setzt Volkswagen bei der Elektronikintegration auf eine zunehmende Anzahl von System-HiLs in der Fahrzeugintegration als Ergänzung zu den Gesamtintegrations-HiLs. Neben einer frühen Funktionserlebbarkeit der einzelnen Systeme wird auf diese Weise auch eine zeitgleiche Absicherung dieser ermöglicht, was zum Frontloading beiträgt. Weiterhin erlaubt die vergleichsweise geringe Komplexität einen effizienten, hochautomatisierten Betrieb zur Variantenabdeckung der Systeme. Am Beispiel der Fahrzeugklimatisierung fokussiert dieser Beitrag die Konzeption der System-HiLs und den effizienten Testbetrieb.

## 2 Mechatronische Entwurfsmethodik

Die Heterogenität bedingt durch das Zusammenwirken unterschiedlicher Domänen resultiert in einer hohen Systemkomplexität und erschwert den Überblick über das Gesamtsystem. Um das gewünschte integrative Verhalten der Teilsysteme zu erlangen, ist eine ganzheitliche und strukturierte Vorgehensweise von Beginn der Entwicklung an notwendig.

Zur Konzeption der System-HiLs wurde eine verifikationsorientierte und modellbasierte Entwicklungs-

methodik angewandt. Dabei erfolgt zunächst eine Strukturierung und Hierarchisierung des Gesamtsystems in einem Top-Down-Prozess in Module, sodann die modellbasierte Auslegung der Module in einem durchgängigen Prozess bis hin zur sukzessiven Integration zum Gesamtsystem (Mechatronische Komposition) in einem Bottom-Up-Prozess [1, 2]. Der durchgängige und modellbasierte Charakter dieses Vorgehens ermöglicht auch durch Parallelisierbarkeit des Entwurfs der einzelnen Module eine zeiteffiziente Entwicklung.

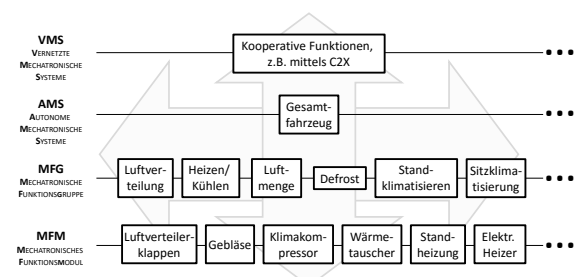
In einem Top-Down-Prozess werden zur Strukturierung des Gesamtsystems die abgeleiteten Teilfunktionen modularisiert und hierarchisiert. Für eine übersichtliche Darstellung des Gesamtsystems wird in [3] eine hierarchische Darstellung auf vier Ebenen vorgeschlagen:

- VMS – Vernetztes mechatronisches System
- AMS – Autonomes mechatronisches System
- MFG – Mechatronische Funktionsgruppe
- MFM – Mechatronisches Funktionsmodul

Die mechatronischen Funktionsmodule (MFM) auf der untersten Ebene stellen die vitalsten Elemente des Gesamtsystems dar. Mit ihrer mechanischen Tragsstruktur, Sensorik, Informationsverarbeitung und Aktorik weisen diese die kleinsten Zeitkonstanten im System auf, weshalb deren Informationsverarbeitung teilweise harte Echtzeitbedingungen erfüllen muss. Mit ihrer Aktorik dienen die MFM zur Umsetzung der geforderten Dynamik für das gewünschte Verhal-

ten. Auf der nächsten Hierarchieebene verfügen die mechatronischen Funktionsgruppen (MFG) über eigene Sensorik und Informationsverarbeitung und greifen zur Funktionserfüllung auf die unterlagerten MFM zurück.

Die MFG verfügen über keine eigene Aktorik. Häufig dienen MFG der hierarchischen Strukturierung der Informationsverarbeitung. Das mechatronische Gesamtsystem bildet die nächste Hierarchieebene mit der Bezeichnung autonome mechatronische Systeme (AMS). Bestehend aus mechanischer Tragstruktur, Sensorik und Informationsverarbeitung verfügt es über Informationen, die das Gesamtsystem betreffen, und erteilt entsprechende Befehle an unterlagerte MFG und MFM. Werden mehrere Gesamtsysteme nebeneinander betrieben, z.B. zur kooperierenden Aufgabenerfüllung, so bedarf es einer höheren Instanz zur Koordination. Diese wird durch die Ebene vernetzte mechatronische Systeme (VMS) abgebildet. Die VMS verfügen über die hierarchisch globalsten Informationen, die zur Erfüllung einer Kooperation von AMS notwendig sind. Bei der hierarchischen Strukturierung unterstützt eine Orientierung an den Zeitkonstanten der Teilsysteme. Teilsysteme mit kleinen Zeitkonstanten sind in den unteren Ebenen aufzufinden, Teilsysteme mit großen Zeitkonstanten und weitreichenden Informationen gehören zu den höheren Ebenen.



**Abbildung 1.** Hierarchische Struktur am Beispiel Fahrgastraumklimatisierung

Ergebnis der Modularisierung und Strukturierung ist eine funktional hierarchische Anordnung der Teilfunktionen mit definierten Schnittstellen in horizontaler und vertikaler Richtung. Damit wird eine Übersicht über das Gesamtsystem geschaffen und die Schnittstellen sind definiert. Diese sind wesentlich, um eine holistische Simulationsumgebung, die einen closed-loop Betrieb im Gesamtsystem ermöglicht, zu realisieren. In Abbildung 1 ist ein Ausschnitt der hierarchischen Strukturierung mit Fokus auf die

Funktionen der Fahrgastraumklimatisierung abgebildet.

Auf der untersten Ebene MFM finden sich die Akteure wieder, die für ein kontrolliertes Systemverhalten mit gewünschter Dynamik im Gesamtsystem sorgen. Hierzu zählen exemplarisch die Luftverteilungsklappen, das Gebläse zur Steuerung der zugeführten Luftmenge, der Klimakompressor, der Heizungs-wärmetauscher, die Standheizung und der elektrische Zuheizer zur Standklimatisierung.

Für eine ganzheitliche Wirkung mangelt es den MFM an überlagerten Informationen. Über solche Informationen verfügen die MFG, die von umfangreicher Sensorik zur Schätzung des Systemverhaltens und Betriebszustands unterstützt werden. Mit ihren weitreichenden Informationen erteilen sie den MFM Stellbefehle. Exemplarisch werden hier die Luftverteilung, das Heizen/ Kühlen, die Luftmengenbestimmung, die Defrostfunktion, das Standklimatisieren und die Sitzklimatisierung als MFG genannt. Über weitere, das Gesamtfahrzeug betreffende Informationen verfügen die AMS. Für Aufgaben, die erst mit einer Vernetzung ermöglicht werden, verfügt das VMS über das einzelne Fahrzeug hinausgehende Informationen.

Vorteilhaft an der mechatronischen Strukturierung sind die damit geschaffene Übersichtlichkeit, die auch durch die bewusst gewählte Anzahl von vier Hierarchieebenen gewahrt bleibt, und die Definition von Schnittstellen in horizontaler und vertikaler Richtung. Die grauen Pfeile in Abbildung 1 deuten die Schnittstellen in diesen beiden Richtungen an. Dies erlaubt die simultane modellbasierte Auslegung aller Funktionsmodule auf allen Ebenen und damit einen hocheffizienten Entwurf. Bei der Konzeption der modularen System-HiLs unterstützte die hierarchische Strukturierung mit den aufgezeigten Schnittstellen bei der Auslegung der Simulationsmodelle. Aufgrund der Modularität ist der Prüflingsverbund erweiterbar und damit verschieben bzw. verändern sich die Schnittstellen zu den Simulationsmodellen.

Für die Funktionsentwicklung erfolgt nach der Strukturierung der Entwurf der einzelnen Module bis hin zur Integration zum Gesamtsystem. Beginnend mit der untersten und vitalsten Ebene erfolgt ein ganzheitlicher, durchgängiger, modellbasierter und verifikationsorientierter Entwurf mittels Model-in-the-Loop (MiL), Software-in-the-Loop (SiL) und Hardware-in-the-Loop (HiL) in einem mechatronischen

Entwicklungskreislauf [2]. Dabei werden die Teilsysteme sukzessiv zum Gesamtsystem in einem Bottom-Up-Prozess integriert.

### 3 Konzeption des System-HiLs

Bei der Definition des Steuergeräteverbunds für den System-HiL besteht ein grundlegender Zielkonflikt. Einerseits ist aus Sicht der Prüfinfrastruktur der Prüfingsverbund zur Reduzierung der Komplexität möglichst klein zu halten, da eine geringe Komplexität ein hohes Potential für einen zeiteffizienten Betrieb darstellt. Eine geringe Anzahl der am Testbetrieb beteiligten Komponenten reduziert den Analyseaufwand und trägt zur schnellen Ursachenfindung bei Auffälligkeiten bei, woraus eine Zeitersparnis resultiert. Andererseits wird ebenso zur Aufwandsreduzierung eine hohe Testabdeckung des Systems mit seinen Varianten angestrebt. Das Ziel der Aufwandsreduzierung besteht hierbei mit möglichst wenig unterschiedlicher Prüfinfrastruktur alle Varianten bei Volllastabdeckung abzusichern. Verschiedene, unausgelastete Prüfplätze, die z.T. dieselben Prüffähigkeiten aufweisen, sollten aus wirtschaftlichen Gründen vermieden werden.

Unter Anwendung und Erweiterung der beschriebenen mechatronischen Entwurfsmethodik wird der Zielkonflikt entschärft. Die hierarchische Strukturierung der Klimafunktionen wird um die an den Teilfunktionen beteiligten Steuergeräte ergänzt. Weiterhin wird an jedes beteiligte Steuergerät die Testfallanzahl der Teilfunktion dargestellt, siehe Tabelle 1.

Teilfunktionen einer Domäne	beteiligte Steuergeräte	Anzahl Testfälle	exemplarisch gewichtete Testfallanzahl
Teilfkt 1	SG 1	TF_Fkt1_SG1	500
Teilfkt 1	SG 2	TF_Fkt1_SG2	
Teilfkt 1	SG 3	TF_Fkt1_SG3	
Teilfkt 1	SG 4	TF_Fkt1_SG4	100
Teilfkt 2	SG 1	TF_Fkt2_SG1	300
Teilfkt 2	SG 2	TF_Fkt2_SG2	
Teilfkt 2	SG 4	TF_Fkt2_SG4	50
Teilfkt 3	SG 1	TF_Fkt3_SG1	200
Teilfkt 3	SG 2	TF_Fkt3_SG2	
Teilfkt 3	SG 3	TF_Fkt3_SG3	
Teilfkt 4	SG 2	TF_Fkt4_SG2	100
Teilfkt 4	SG 5	TF_Fkt4_SG5	

**Tabelle 1.** Übersicht der Teilfunktionen einer Domäne, beteiligter Steuergeräte und zugehörigen Testfällen

Entsprechend der Teilfunktionen ist die Testabdeckung anhand der Anzahl der Testfälle zu erkennen und der Aufwand der Prüfinfrastruktur anhand der beteiligten Steuergeräte abzuschätzen. Die Teilfunktionen werden in der Tabelle nach Anzahl der gewichteten Testfälle sortiert. Die Gewichtung der Testfälle erfolgt nach Zeitaufwand in der Testdurchführung, welcher anhand des Erfahrungswissens vorliegt. Nach der Sortierung der gewichteten Testfallanzahl werden die Verbünde der Teilfunktionen miteinander verglichen, um den Kompromiss aus hoher Testabdeckung und geringsten Verbund zu bestimmen. Tabelle 1 zeigt dies exemplarisch für 4 Teilfunktionen auf. Teilfunktion 1 erfordert 3 Steuergeräte (Steuergerät 1, 2 und 3), um 500 gewichtete Testfälle durchführen zu können. Mit 500 gewichteten Testfällen trägt diese Teilfunktion am stärksten zur Testabdeckung bei, d.h. der dafür notwendige Verbund aus 3 Steuergeräten sollte verbaut werden. Mit diesem Verbund lassen sich noch 300 gewichtete Testfälle der Teilfunktion 2 und 200 gewichtete Testfälle der Teilfunktion 3 abdecken. Insgesamt werden mit diesem Verbund aus nur 3 Steuergeräten 1000 von 1250 gewichteten Testfällen ermöglicht, was einer Testabdeckung von 80 % entspricht.

Unter Hinzunahme von Steuergerät 4 können insgesamt weitere 150 gewichtete Testfälle aus Teilfunktion 1 und Teilfunktion 2 durchgeführt werden, was die Testabdeckung auf 92 % erhöht. Zusätzlich mit Steuergerät 5 werden noch weitere 8 % Testabdeckung erreicht.

Der Kompromiss aus Testabdeckung und Komplexität der Prüfinfrastruktur wurde so gewählt, dass mit einem Verbund aus 4 Steuergeräten ein hochautomatisierter Betrieb umgesetzt wird. Auf das 5. Steuergerät wird am System-HiL verzichtet. Die 8 % Testabdeckungsbeitrag werden an einem anderen Integrations-Prüfplatz erlangt, der den Steuergerätegesamtverbund enthält.

Entsprechend der verbauten Steuergeräte sind die Schnittstellen zur Systemumgebung definiert, die durch Simulationsmodelle bedient werden müssen. Mit dem sukzessiven Vorgehen in der Integration reifen auch die Simulationsmodelle stetig. Grundlage der Modelle sind Restbussimulationen aus den Fachabteilungen, die systematisch erweitert werden. Damit wird eine Durchgängigkeit bis hin zur vollständigen Gesamtfahrzeugintegration erzielt.

Die einzelnen Varianten des Systems finden Berücksichtigung, indem ausgehend von der Variante der technisch höchsten Komplexität die Darstellung entsprechend auf die geringer aufwändigen Varianten sowohl von der Prüflings- als auch von der Prüfinfrastrukturseite reduziert wird.

## 4 Ergebnisse

Bereits in frühen Phasen wird mit den domänenspezifischen System-HiLs ein wesentlicher Beitrag zu einer hohen Testabdeckung erlangt. Neben dem Ziel einer möglichst hohen Testabdeckung bei einem kleinen Steuergeräteverbund trägt maßgeblich der parallele Testbetrieb an den unterschiedlichen System-HiLs zum zeiteffizienten Vorgehen bei. Diese Parallelisierbarkeit im Testbetrieb erleichtert die Absicherung der Fahrzeugintegration wesentlich. Damit wird allen am Testbetrieb indirekt Beteiligten, wie Testfallspezifizierer, Testautomatisierer, Modellierern, genügend Zeit eingeräumt, um die Prüfinfrastruktur und Testautomatisierung in hoher Qualität entsprechend dem Projektfortschritt umzusetzen. Diese gelebte Durchgängigkeit trägt zur Effizienz bei, da in jedem weiteren Integrationsschritt von SiL mit virtuellen Steuergeräten bis hin zu HiL mit kompletten Steuergeräteverbund auf Vorhandenem aufgebaut wird.

Weiterhin ermöglicht die Vielzahl an domänenspezifischen System-HiLs eine systematische Abdeckung der Technikvarianten des Systems. Damit wird grundlegend eine deutlich höhere Testabdeckung in früheren Projektphasen erzielt. Zuerst wird mit maximal komplexer Technikvariante eine zuverlässige Testfallautomatisierung erstellt, die dann auf Technikvarianten geringerer Komplexität angepasst wird. Im Kontext mit Modul-Baukästen, die in zahlreichen Fahrzeugprojekten in einem Vielmarkenkonzern eingesetzt werden, sichert dieses Vorgehen eine hohe Testabdeckung der einzelnen Systeme in frühen Projektphasen.

Die Betriebsweise der System-HiLs sieht während der Arbeitszeit das Einrichten der Verbundkonfiguration und die Erprobung der Testfallautomatisierung neuer Testfälle vor. Weiterhin werden Auffälligkeiten geklärt und eventuell nachgestellt. Ebenso erfolgt auch die Entwicklung und Inbetriebnahme der Simulationsmodelle. Außerhalb der Arbeitszeit erfolgt der produktive Testautomatisierungsbetrieb. Mit diesem Vorgehen wird eine stetige Weiterentwicklung der Aufgaben aller Beteiligten neben einem hocheffizien-

ten, produktiven Betrieb sichergestellt. Die Auslastung der System-HiLs erfolgt durch eine systematische Planung des Testbetriebs ausgewählter Technikvarianten, um die einzelnen Verbundkonfigurationen gezielt abzusichern und eine vollständige Testabdeckung zu erlangen.

Das sukzessive Vorgehen in der Durchgängigkeit wird ergänzt um einen modularen Aufbau der System-HiLs. Damit lassen sich auf gleicher Integrationsebene Module unter den Prüfständen austauschen. Langfristig wird zum Zwecke der Durchgängigkeit eine höchstmögliche Wiederverwendbarkeit dieser Module auf Prüfplätzen höherer Integrationsebenen angestrebt. Diese durchgängige Wiederverwendbarkeit erfordert ein hohes Maß an Standardisierung der Schnittstellen, was aufgrund der Vielzahl an spezialisierten Herstellern und dem Abstimmungsaufwand in einem Vielmarkenkonzern nicht zu unterschätzen ist. In Teilen funktioniert diese hocheffiziente Durchgängigkeit bereits erfolgreich.

Weitere Synergien mit System-HiLs werden erlangt, indem z.T. unterschiedliche Domänen gemeinsam die Testinfrastruktur nutzen. Dies ist möglich, insofern der vorliegende Prüflingsverbund für die einzelnen Domänen hinreichend ist.

## 5 Ausblick

Mit der von frühen Phasen an gelebten Durchgängigkeit, die bereits bei geringen, aber erweiterbaren, Systemverbünden variable Schnittstellen zur Simulation erfordert, werden sukzessiv virtuelle Steuergeräte umgesetzt. Diese bilden das Verhalten realer Steuergeräte ab und ersetzen in späteren Integrationsphasen z.T. sehr teure reale Steuergeräte bzw. ermöglichen überhaupt eine Erprobungsfähigkeit vernetzter Funktionen in sehr frühen Phasen. Sukzessiv entstehen auf diese Weise nachgebildete Steuergeräte, die in den Entwurfs- und Absicherungsphasen Anwendung finden. Dieses Vorgehen stellt einen weiteren wesentlichen Beitrag zur Durchgängigkeit und damit Prozesseffizienz dar.

Mit den gewonnenen Zeit- und Qualitätsvorteilen wird, auf Basis des Erfolgs bestehender System-HiLs, dieses Konzept auf weitere Domänen ausgeweitet.

## 6 Referenzen

- [1] X. Liu-Henke, J. Lückel, K.-P. Jäker. *Ganzheitlicher mechatronischer Entwurf eines aktiven Feder-/Neigemoduls*. VDI-Tagung: Mechatronik – Mechanisch/Elektrische Antriebstechnik, 2000.
- [2] R. Buchta. *Mechatronische Entwicklung eines Forschungselektrofahrzeugs zur Erprobung von Fahrdynamikregelungen und Fahrerassistenzsystemen*. Dissertation, Uni Magdeburg, Deutschland, 2015.
- [3] J. Lückel, T. Koch, J. Schmitz. *Mechatronik als integrative Basis für innovative Produkte*. VDI-Berichte 1533, VDI-Verlag, Düsseldorf, 2000.

# Verlustanalyse im elektrischen Antrieb von Brennstoffzellenfahrzeugen bei variierender Zwischenkreisspannung

Dipl.-Ing. Anna-Lena Menn, Prof. Dr.-Ing. Markus Henke

TU Braunschweig, Institut für Elektrische Maschinen, Antriebe und Bahnen

Hans-Sommer-Straße 66, 38106 Braunschweig

a-l.menn@tu-bs.de

Dieser Aufsatz befasst sich mit der Analyse des Verlustverhaltens einer Asynchronmaschine als Antrieb eines Brennstoffzellenfahrzeugs bei variierender Zwischenkreisspannung. Zur Findung der optimalen Größe eines Brennstoffzellenstacks und somit der anliegenden Zwischenkreisspannung ist es wichtig den Zusammenhang zwischen Höhe der Zwischenkreisspannung und den Verlusten in elektrischer Maschine und Wechselrichter zu kennen. Für zwei Spannungsebenen wird dieser Zusammenhang untersucht, gleichzeitig wird auch das Regelkonzept verändert, um realisierbare Wirkungsgradverbesserung darzustellen.

## 1 Einleitung

Als Energiespeicher eines elektrisch angetriebenen Fahrzeugs dient aktuell überwiegend die Lithium-Ionen-Batterie. Diese weist allerdings eine geringe Energiedichte auf, was sich schlussendlich negativ in der Reichweite der jeweiligen Fahrzeuge niederschlägt. Weiterhin sind auch die Ladedauer und Lademöglichkeiten dieses Energiespeichers immer noch aktueller Diskussionsgegenstand bei der Marktdurchdringung von Batterie betriebenen Fahrzeugen.

Dem gegenüber steht die Möglichkeit Wasserstoff als Energiespeicher zu verwenden und mittels einer Brennstoffzelle die chemisch gespeicherte Energie in elektrische Energie umzuwandeln.

Brennstoffzellenfahrzeuge weisen zwar eine größere Reichweite als batteriebetriebene Fahrzeuge auf, sie sind allerdings ebenfalls wesentlich teurer als herkömmliche Fahrzeuge mit Verbrennungsmotoren. Eine Kostenreduktion ist hier über die Optimierung des Brennstoffzellenstacks möglich, welche auf eine möglichst kleine Dimensionierung des Stacks hinausläuft. Diese Optimierung hat eine reduzierte Zwischenkreisspannung zur Folge. Ausser Acht gelassen werden darf allerdings nicht der geforderte Leistungsbedarf des Antriebsstrangs.

Aus diesem Grund werden in den folgenden Untersuchungen die Auswirkungen der Änderung der Zwischenkreisspannung auf den Wirkungsgrad des Antriebsstrangs, im Besonderen von elektrischer Maschine und Leistungselektronik, betrachtet. Der Antriebsstrang wird als Simulinkmodell mit seinen dazugehörigen Komponenten für die durchzuführenden Untersuchungen abgebildet.

Zusätzlich wird der Einfluss unterschiedlicher Regelverfahren betrachtet.

## 2 Modulare Simulationsstruktur

Als Grundlage für die Untersuchungen dient eine quasi-stationäre Vorwärtssimulation eines Elektrofahrzeugs. Abbildung 1. zeigt die verwendete Fahrzeugtopologie und die dazugehörigen Komponenten. Das Simulationsmodell ist modular aufgebaut, so dass die einzelnen Komponenten bei Bedarf ausgetauscht werden können.

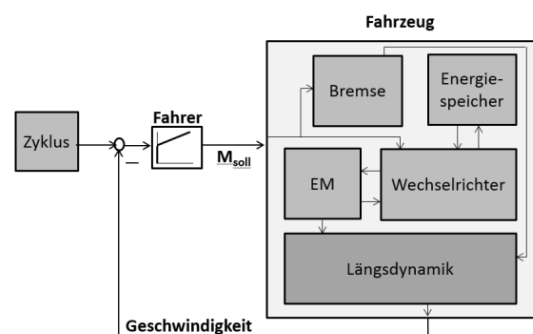


Abbildung 2. verwendete Antriebstopologie und Aufbau des Fahrzeugmodells

### 2.1 Fahrzeugtopologie und Modellkomponenten

Bei dem Fahrzeugkonzept handelt es sich um ein Elektrofahrzeug, welches von einer Asynchronmaschine an der Vorderachse angetrieben wird. Die Anordnung enthält ausschließlich ein Differentialgetriebe zur Drehmoment- und Drehzahlwandlung. Die verwendeten Fahrzeugdaten sind in Tabelle 1. zu sehen. Der Fahrer hat das Ziel einer vorgegebenen Geschwindigkeit zu folgen, daher ist das Fahrermodell als PI -

Regler ausgeführt und gibt die Drehmomentanforderung an den Antriebsstrang.

Im Block Längsdynamik werden die Fahrwiderstände berechnet, die ein Fahrzeug zur Fortbewegung überwinden muss. Ebenso wird die Radleistung der Asynchronmaschine berechnet, die zur Überwindung der Fahrwiderstände zur Verfügung steht. Für die Schubkraft gilt:

$$F_{Antrieb} = M_{ASM} * \frac{i_{Achse}}{r_{Rad}} \quad (1)$$

Über

$$\Delta F = F_{Antrieb} - F_{Widerstand} = m * a \quad (2)$$

ergibt sich dann eine realisierte Beschleunigung. Daraus resultiert dann über Integration die Ist-Geschwindigkeit, die an das Fahrermodul zur Ermittlung der Stellgröße (Drehmoment) gegeben wird.

Das Modell des Energiespeichers wird so konfiguriert, dass eine Bereitstellung von unterschiedlichen Spannungsniveaus im Zwischenkreis möglich ist. Auf eine genauere Darstellung des Energiespeichers wird an dieser Stelle verzichtet, da die Reaktion der Maschinenverluste auf die Spannungsänderung im Fokus der Untersuchungen steht.

Die Berechnung des Bremsdrehmoments erfolgt unter Berücksichtigung einer seriellen Strategie, welche abhängig von der angeforderten Verzögerung des Fahrers ist. Bei geringen Verzögerungen erfolgt zunächst nur die elektrische Nutzbremmung, bei höheren Verzögerungen erfolgt zusätzlich die mechanische Reibbremsung. Das Bremsdrehmoment wird an das Modul der Längsdynamik weitergegeben.

Das Modell der antreibenden elektrischen Maschine besteht aus einem dynamischen feldorientierten d,q-Modell, das sicherstellt, dass der Übergang zwischen zwei Betriebspunkten exakt dargestellt wird [1]. Das Maschinenmodell generiert unter Einfluss der vom Wechselrichter kommenden Steuerspannungen ein vom Fahrer gefordertes Solldrehmoment für das Modell der Längsdynamik, sodass nach Formel (2) eine geforderte Beschleunigung bzw. Geschwindigkeit eingehalten werden kann. Von dem Maschinenmodell werden darüberhinaus der Rotorfluss  $\Psi_R$  und die (unabhängigen) Stromkomponenten  $i_{sd}$ ,  $i_{sq}$  an das Modell des Wechselrichters für die Maschinenregelung übergeben. Für die Wirkungsgradberechnung werden in diesem Modell die Maschinenverluste bestehend aus Stator-, Rotor- und Eisenverlusten berechnet. Als Traktionsmaschine wird eine Asynchronmaschine mit den Parametern aus Tabelle 2. verwendet.

In dem Modell des Wechselrichters sind die Maschinenregelung und die Verlustleistungsberechnung des

Umrichters implementiert. Bei dem Umrichter handelt es sich um eine voll-aktive B6 Brücke mit den Kenndaten aus Tabelle 1. Die Maschinenregelung (siehe Kapitel 3) berechnet die notwendigen Steuerspannungen für die Maschine, die vom Umrichter durch Pulsweitenmodulation umgesetzt werden, damit ein vom Fahrer gefordertes Solldrehmoment eingehalten werden kann.

<b>Asynchronmaschine</b>	
$P_{Nenn}$	39 kW
$P_{max}$	90 kW
$n_{nenn}$	5.000 1/min
$n_{max}$	15.000 1/min
$p$	3
<b>Wechselrichter</b>	
Schaltfrequenz	10 kHz
Leistungsfaktor	1
Modulationsgrad	0,9
Nennspannung	600V
<b>Fahrzeugdaten</b>	
Fahrzeugmasse	1447 kg
Radradius	0,317 m
Fahrzeughöhe	1,79 m
Differentialübersetzung	9,5
Summe Bremsmoment	1200 Nm

Tabelle 1. Fahrzeug- und Komponentendaten

### 3 Maschinenmodellierung und Regelverfahren

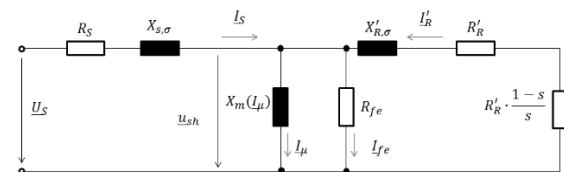


Abbildung 3. Verwendetes Ersatzschaltbild der ASM

In diesem Kapitel werden zwei verschieden agierende Regelverfahren zur Effizienzsteigerung der Asynchronmaschine vorgestellt. Grundlage für beide Verfahren ist das in Abbildung 4. gezeigte einphasige T-Ersatzschaltbild und den dazugehörigen bekannten Spannungs- und Stromgleichungen [2].

Bei beiden Verfahren wird ein optimaler Flusssollwert berechnet, der zu einer Wirkungsgradsteigerung in bestimmten Bereichen des Kennfeldes der Traktionsmaschine führt.

Diese Regelverfahren werden wie in Abbildung 5. dargestellt mit dem Block  $\eta_{opt}$  in die feldorientierte Regelung eingebunden, in diesem findet die Berechnung

des optimalen Flusses statt. Um auch im Feldschwächbereich zu gewährleisten, dass ein wirkungsgradoptimaler Betrieb vorliegt, wird hier die herkömmliche Struktur der Feldschwächung, in Form einer Spannungsbegrenzung, zusätzlich zum  $\eta_{opt}$ -Block verwendet [3]. Damit wird verhindert, dass ein zu hoher Fluss im Feldschwächbereich eingestellt wird. Übergeben wird demnach als Sollwert an den Flussregler:

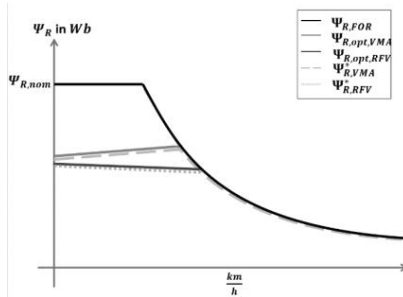


Abbildung 4. Flussverlauf über die Fahrzeuggeschwindigkeit für alle drei Regelungen

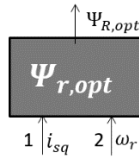


Abbildung 6. Einbindung des VMA in FOR

### 3.1 Verlustminimierungsalgorithmus

Bei dem verwendeten Verlustminimierungsalgorithmus (im Folgenden VMA genannt) werden die Verluste anhand des Ersatzschaltbildes der Asynchronmaschine aus Abbildung 3 berechnet und durch Einstellung eines optimalen Flusswertes minimiert.

Die Entwicklung des VMA ist ausführlich in [5] dargestellt. An dieser Stelle wird die Wirkungsweise kurz vorgestellt. Es werden nur folgende ausschlaggebenden Verluste berücksichtigt, die Eisenverluste werden dabei durch den ohmschen Eisenverlustwiderstand  $R_{fe}$  berücksichtigt:

$$P_{Verluste} = P_{Stator} + P_{Eisen} + P_{Rotor} \quad (4)$$

Dieser Algorithmus minimiert die oben genannten Verluste gegenüber der feldorientierten Regelung in Echtzeit. Dies geschieht über die Berechnung des op-

$$\Psi_{R, soll} = \min(\Psi_{R, opt}; \Psi_{R, FSB}) \quad (3)$$

Die Effektivität der Verlustminimierung hängt dabei von der richtigen Berechnung der Maschinenparameter, bspw. der Widerstandsgrößen und der Induktivität ab [4].

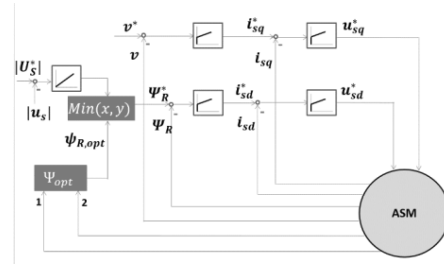


Abbildung 5. Regelungsschema der Asynchronmaschine basierend auf der FOR

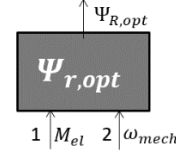


Abbildung 7. Einbindung der RFV in FOR

timalen Magnetisierungsstroms, bei dem die gesamten Verluste minimal werden. Aus diesem kann dann mit dem Wert der Hauptfeldinduktivität der optimale Fluss berechnet werden:

$$\Psi_{R, opt} = i_{\mu, opt} * L_m \quad (5)$$

Der Algorithmus wird durch die Abbildung 6 in die feldorientierte Regelung aus Abbildung 5 eingebunden. Die für den Algorithmus notwendigen Komponenten  $i_{sq}$  und  $\omega_r$  ergeben sich aus dem Modell der Asynchronmaschine.

In Abbildung 4 sind die qualitativen Flussverläufe der feldorientierten Regelung (dunkel fett) und des VMA (grau) dargestellt. Es ist zu erkennen, dass der Flusswert des VMA, auf Grund der direkten Abhängigkeit zum drehmomentbildenden Strom  $i_{sq}$ , ab einer bestimmten Geschwindigkeit/Belastung den Flusswert der feldorientierten überschreitet. Dies ist dann der Fall, wenn für eine höhere Geschwindigkeit eine stärkere Feldschwächung notwendig wird, als der VMA berechnet. Für einen optimalen Betrieb der Asynchronmaschine im Feldschwächbereich wird die lang gestrichelte Linie gewählt.

### 3.2 Rotorfrequenz - Optimierungsverfahren

Bei dem hier vorstellten Verfahren (im Folgenden ROV genannt) erfolgt die Berechnung des optimalen Flusses über die Bestimmung der dazugehörigen optimalen Rotorfrequenz. Eine ausführliche Darstellung der Vorgehensweise findet sich in [6]. Die wichtigsten Zusammenhänge sind nachfolgend kurz erläutert.

Die Vorgehensweise der Optimierung bezieht sich in den Berechnungsschritten auf das Ersatzschaltbild in Abbildung 6. Für einen beliebigen Punkt im Drehmoment-Drehzahl-Kennfeld der Asynchronmaschine wird zunächst unter Verwendung des Rotornennflusses die Rotorfrequenz  $\omega_r$  und der Schlupf  $s$  gemäß folgender Gleichungen bestimmt.

$$\omega_r = \frac{M \cdot (L_r \sigma + L_{sh})}{m_s \cdot p \cdot T_r \cdot \psi_f^2} \quad (6)$$

$$s = \frac{\omega_r}{2\pi p \cdot \frac{n}{60} + \omega_r} \quad (7)$$

Anschließend erfolgen die Berechnungen der Ströme und Spannungen gemäß des Ersatzschaltbildes für den Grundstell- und Feldschwächbereich. Bei Berechnungen im Feldschwächbereich ist dabei auf den Spannungsbedarf zu achten.

Danach wird der Wirkungsgrad  $\eta$  bestimmt.

Im Folgenden wird dieses Vorgehen für den gleichen Betriebspunkt mit einer erhöhten Rotorfrequenz wiederholt, bis der optimale Wirkungsgrad gefunden wurde. Abschließend wird die Rotorfrequenz  $\omega_r$ , für die sich der maximale Wirkungsgrad ergibt, in einer Look-Up Tabelle abgespeichert. Dieser Algorithmus wird auf alle vorhandenen Kennfeldpunkten angewendet.

Diese Look-Up Tabelle wird in die feldorientierte Regelung durch Abbildung 7. eingebunden. Innerhalb des  $\eta_{opt}$  -Blocks wird darüber hinaus die optimale Rotorfrequenz gemäß folgender Gleichung

$$\psi_r = \sqrt{\frac{M \cdot (L_r \sigma + L_{sh})}{m_s \cdot p \cdot T_r \cdot \omega_r}} \quad (8)$$

in den Rotorfluss umgewandelt, sodass ein Sollwert

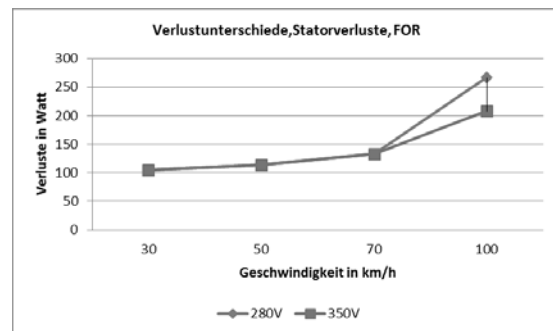
$\Psi_{R,opt}$  für den Flussregler generiert wird.

## 4 Verlustverhalten bei variierender Zwischenkreisspannung

Wie bereits in Kapitel 2 erwähnt, wurde der Energiespeicher so modelliert, dass er unterschiedliche Leerlaufspannungen ausgeben kann, im vorliegenden Fall 280V und 350V. Die folgenden Untersuchungen wur-

den jeweils für diese beiden Spannungslagen durchgeführt, bei gleichbleibender Wicklungskonfiguration der Maschine. Weiterhin wird der Flusssollwert für beide Spannungsebenen gleich angenommen, dieser Flusswert wurde für eine optimale elektromagnetische Auslegung ermittelt. Die Verluste werden bei typischen Geschwindigkeiten im deutschen Straßenverkehr gemessen.

Unabhängig von der Regelung sind die Kupfer – und Eisenverluste sowie die Durchlassverluste im Grundstellbereich nicht von der Batteriespannung abhängig, Dies resultiert aus der Unabhängigkeit der Statorstromkomponenten von der Batteriespannung und der ausschließlichen Berücksichtigung der Grundwelle der Asynchronmaschine. Dadurch wird der von der Zwischenkreisspannung abhängige Stromripple nicht berücksichtigt, der für eine Spannungsabhängigkeit der Eisenverluste sorgen würde. Somit sind diese Verluste für beide Spannungsebenen gleich. Abbildung 8 verdeutlicht dies am Beispiel der Kupferverluste im Stator wenn feldorientierte Regelung vorliegt.



**Abbildung 7.** Verlustunterschiede der Statorwicklung bei variabler Zwischenkreisspannung

Lediglich die Schaltverluste des Wechselrichters sind von dem Spannungsniveau der Batterie abhängig. Grund hierfür ist der Zusammenhang zwischen den Schaltverlusten der IGBT und Dioden mit der Batteriespannung.

Befindet sich die Maschine im Feldschwächbereich treten bei den drei Regelungen unterschiedliche Verlusteverläufe auf, die in den beiden weiteren Unterkapiteln näher betrachtet werden.

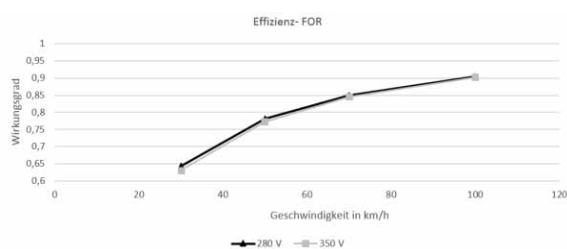
### 4.1 Verlustverhalten bei der FOR

Bei der feldorientierten Regelung befindet sich die Maschine bei 280 V bis 70 km/h im Grundstellbereich, bei 350V ausschließlich im Grundstellbereich. Das bedeutet einen späteren Übergang vom Grundstell- in den Feldschwächbereich bei 350V.

Aus diesem Grund steigen die Kupfer- und Durchlassverluste in Richtung niedriger Zwischenkreisspannung, da sich die Maschine bei 280V und 100 km/h bereits im Feldschwächbereich befindet und somit

durch die Flussverminderung ein erhöhter Statorstrombedarf vorliegt. Dieser resultiert aus der Forderung, dass die Leistung konstant gehalten werden soll. Die Eisenverluste sinken bei Abnahme der Zwischenkreisspannung, da bei 280V und 100km/h bereits Feldschwächung vorliegt.

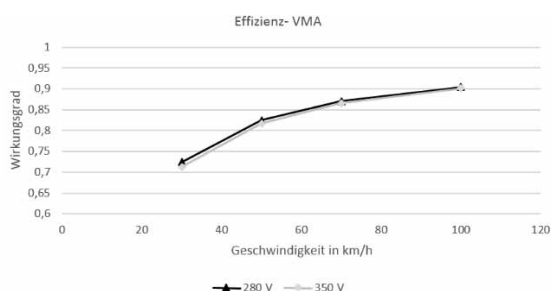
Bei 100 km/h werden die Eisenverluste um 37.7% reduziert, die Erhöhung der übrigen Verluste fallen geringfügiger aus. Aus diesem Grund sind die Gesamtverluste bei 350V Zwischenkreisspannung höher als bei 280V.



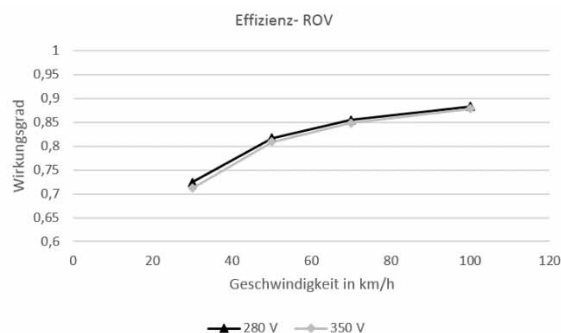
**Abbildung 7.** Wirkungsgrad von EM und LE mit FOR Regelung bei unterschiedlichen Geschwindigkeiten

#### 4.2 Verlustverhalten bei ROV und VMA

Wird die Maschine mit den beiden wirkungsgradoptimalen Regelverfahren betrieben, befindet sie sich über den gesamten Geschwindigkeitsbereich im Grundstellbereich. Somit bleiben die Statorstromkomponenten stets unabhängig von der Batteriespannung. Das bedeutet, dass die Eisen- und Kupferverluste für beide Spannungsebenen identisch sind. Gleiches gilt für die Durchlassverluste der Leistungselektronik. Lediglich die Schaltverluste variieren auf Grund ihrer Abhängigkeit von der Batteriespannung, diese steigen bei steigender Zwischenkreisspannung. Somit ist der Wirkungsgrad bei 280V Zwischenkreisspannung auch hier höher als bei 350V.



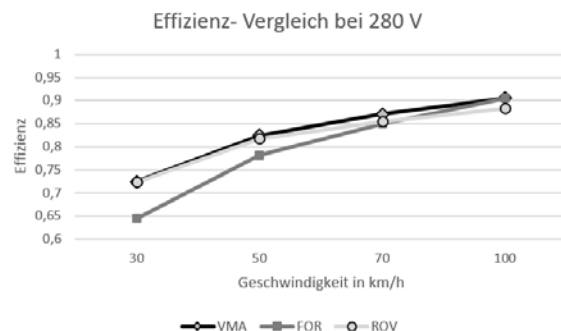
**Abbildung 8.** Wirkungsgrad von EM und LE mit VMA Regelung bei unterschiedlichen Geschwindigkeiten



**Abbildung 9.** Wirkungsgrad mit EM und LE mit VMA Regelung bei unterschiedlichen Geschwindigkeiten

Bei allen drei Regelungen ist zu sehen, dass die Wirkungsgrade bei 280V besser sind als bei 350V.

Im direkten Vergleich der drei Regelungen für eine feste Zwischenkreisspannung ist zu sehen, dass die VMA besser als die beiden anderen Regelungen agiert.



**Abbildung 10.** Direkter Vergleich der Regelkonzepte bei konstanter Zwischenkreisspannung

Die Ergebnisse zeigen, dass eine niedrige Zwischenkreisspannung für einen wirkungsgradoptimalen Betrieb der Antriebsmaschine vorteilhaft ist. Dies unterstützt den Wunsch nach einer Kostenreduzierung da es eine Verkleinerung des Brennstoffzellenstacks ermöglicht. Es ist zu prüfen, ob diese reduzierte Bauform des Stacks signifikante Auswirkungen auf die zu realisierende Beschleunigung hat. Hierfür ist eine fahrzyklusbasierte Simulation notwendig. Sollte die Beschleunigung nicht realisierbar sein, kann über den Einsatz eines DC/DC Wandlers diskutiert werden. Zu beachten ist dann aber wieder die Kostensteigerung bedingt durch eine zusätzliche Komponente im Antriebsstrang.

## 5 Zusammenfassung

Die Variation der Zwischenkreisspannung führt dazu, dass bei allen drei Regelungen die Wirkungsgrade mit sinkender Zwischenkreisspannung ansteigen. Der VMA ist sowohl bei 280V als auch bei 350V die Regelung mit den besten Wirkungsgraden über dem gesamten Geschwindigkeitsbereich.

In den folgenden Untersuchungen wird das Energiespeichermodell um eine Lithium-Ionen Batterie ergänzt. Dann sind Untersuchungen zur Bestimmung des optimalen Energiemanagements und dem akuraten Zusammenspiel der beiden Energiespeicher möglich.

## 6 Referenzen

- [1] Ali M. Bazzi, Philip T. Krein, Comparative Evaluation of Machines for Electric and Hybrid Vehicles Based on Dynamic Operation and Loss Minimization, IEEE Energy Conversion Congress and Exposition, ISBN 978-1-4244-5287-3, 2010 ,3345-3351
- [2] Eckhardt, Hanskarl, Grundzüge der elektrischen Maschinen , Teubner Taschenbücher, 1982
- [3] Jiajia Wu, Dawei Gao, Xin Zhao, Qingchun Lu, An Efficiency Optimization Strategy of Induction Motors for Electric Vehicles, IEEE Vehicle Power und Propulsion Conference (VPPC), ISBN 978-1-424-1849-7,2008
- [4] Ali M. Bazzi, Philip T.Krein, Review of Methods for Real-time Loss Minimization in Induction Machines, IEEE Transactions on Industry Applications, ISSN 0093-9994, Vol.46, No.6, 2010, 2319-2328
- [5] Nasir Uddin and Sang Woo Nam, New Online Loss-Minimization-Based Control of an Induction Motor Drive, IEEE Transactions on Power Electronics, ISSN 0885-8993, Vol.23, No.2, 2008, 926-933E.
- [6] Anna-Lena Menn, Wolf-Rüdiger Canders, Markus Henke, Holistic Analytical Design of Induction Motors for Automotive Application, Ninth International Conference on Ecological Vehicles and Renewable Energies (EVER), ISBN 978-1-4799-3787-5, 2014

# Schutzorgane und -konzepte in Niederspannungs-Gleichstromnetzen – Modellierung von Komponenten, Fehlersituationen und Gesamtsystem

Christian Strobl<sup>1</sup>, Maximilian Schäfer<sup>2</sup>, Rudolf Rabenstein<sup>2</sup>, Leopold Ott<sup>3</sup>, Julian Kaiser<sup>3</sup>

<sup>1</sup>E-T-A Elektrotechnische Apparate GmbH, Altdorf, Deutschland

<sup>2</sup>Lehrstuhl für Multimediakommunikation und Signalverarbeitung,  
Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen, Deutschland

<sup>3</sup>Fraunhofer Institut für Integrierte Systeme und Bauelementetechnologie IISB,  
Erlangen, Deutschland

*Christian.Strobl@e-t-a.de*

Moderne Niederspannungs-Gleichstromnetze erfordern im Vergleich zu herkömmlichen AC-Netzen überarbeitete und verfeinerte Schutzkonzepte und -komponenten. Für die Entwicklung adaptiver und selbstlernender Methoden zur Fehlererkennung ist die modulare Modellierung von leistungselektronischen Komponenten, von Kabelstrecken und möglichen Fehlersituationen Grundvoraussetzung.

## 1 Einleitung

Bedingt durch die Entwicklung moderner Leistungshalbleiter erfuhren Niederspannungs-Gleichstrom-(LVDC-)Netze in den letzten Jahren immer mehr Verbreitung. Neben Bordnetzen von Fahrzeugen, Schiffen und Flugzeugen sowie im Bereich der photovoltaischen Energieerzeugung werden auch erste Verteilnetze in Rechenzentren und Anwendungen im Bereich der Automatisierungstechnik und Robotik in DC-Topologie bei Spannungsebenen von mehreren 100 V betrieben.

Im Vergleich zu herkömmlichen Wechselspannungsnetzen zeichnen sich DC-Netztopologien insbesondere bei der Verwendung von Batteriespeichern und modularen regenerativen Erzeugungseinheiten aufgrund einer kleineren Anzahl nötiger Wandlungsstufen durch höhere Energieeffizienz bei geringerem Materialeinsatz aus.

In AC etablierte Schutzorgane können allerdings im Allgemeinen nicht ohne Weiteres in DC-Netze übernommen werden. Neuartige an die DC-Netztopologie mit teils bidirektionalem Stromfluss angepasste Regelungsmethoden sowie die durch Wandler bedingte interne Dauerstrombegrenzung erfordern eine grundlegende Überarbeitung der Schutzkonzepte und

-elemente. Besonders zu berücksichtigen ist dabei die Tatsache, dass die in mechanischen Schaltern und in Fehlersituationen auftretenden Lichtbögen aufgrund fehlender Stromnulldurchgänge nicht selbsttätig verlöschen, sondern u. U. langanhaltend brennen können, was zu schwerwiegenden Folgeschäden führen kann.

Die Detektion ausschließlich „harter“ Fehler (Kurzschlüsse und massive Überspannungsereignisse) ist somit in DC-Netzen häufig nicht ausreichend, eine verlässliche Erkennung und Abschaltung auch schleichender Prozesse und „weicher“ Fehlersituationen (hochohmige parallele und niederohmige serielle Fehler) sind notwendig.

Dies erfordert die Entwicklung intelligenter Schutzorgane, die mit verfeinerten adaptiv an den jeweiligen Netzzustand angepassten Auslösebedingungen bereits Fehler im Anfangsstadium erkennen können, sowie die Einbindung dieser sensorischen und aktorischen Komponenten in ein Gesamtschutzkonzept unter Einbeziehung der leistungselektronischen Wandler und ihrer Regelungsmethodik.

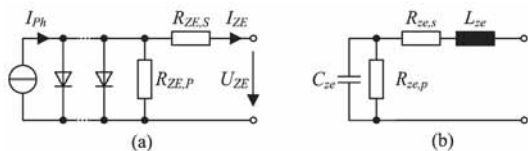
## 2 Modellierung von Komponenten und Fehlersituationen

Für Einflussgrößenanalysen im Zeit- und Frequenzbereich und der daraus abgeleiteten Entwicklung echtzeitfähiger Fehlerdetektionsverfahren – je nach Anwendung auch mit Verwendung maschineller Lernmethoden – ist dabei eine möglichst genaue Modellierung der leistungselektronischen Einheiten, der Verkabelung und der potentiellen Fehlersituationen unerlässlich.

Mit einem Ansatz basierend auf teilanalytischer Gray-Box-Modellierung können dabei physikalische – teils durch Impedanzspektroskopie ermittelte – Beschreibungen von Komponenten und Subnetzen, sowie von möglichen Fehlerfällen mit unterschiedlichem Genauigkeitsgrad in Submodellen erfasst werden. Diese werden dann in ein Systemmodell integriert, um mit dessen Hilfe möglichst präzise zwischen Quellen- und Lastveränderungen einerseits und Fehlerereignissen andererseits unterscheiden zu können. Liegt der Fokus auf der Detektion von anfänglich weichen Fehlern mit geringem Leistungsumsatz, ist dabei meist eine Trennung in Großsignalcharakteristika und Kleinsignalmodelle (linearisiert und mit Einbeziehung von Rausch- und AC-Ersatzquellen für Lichtbögen und Wandler) hilfreich [1].

### 2.1 Modellierung von Wandlern, Quellen und Lasten

Für die Beschreibung nichtlinearer leistungselektronischer Komponenten ist dieser zweiteilige Model-

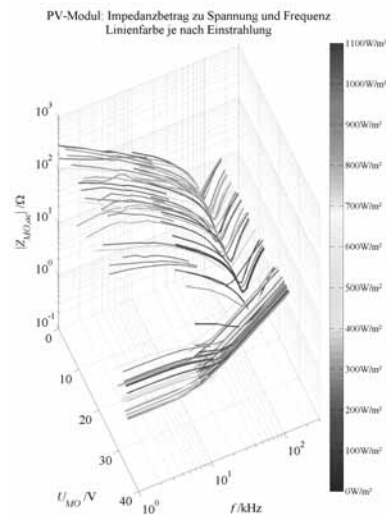


**Abbildung 1:** Groß- und Kleinsignalmodellierung am Beispiel einer Solarzelle, siehe [3, 4, 5]:

(a) Großsignal-Ersatzschaltbild zur Beschreibung des impliziten Zusammenhangs zwischen Zellenstrom  $I_{ZE}$  und Zellenspannung  $U_{ZE}$ : einstrahlungsabhängige Stromquelle  $I_{ph}$ , Dioden mit unterschiedlichen temperaturabhängigen Kennlinien, konstante Widerstände  $R_{ZE,S}$  und  $R_{ZE,P}$ ;

(b) korrespondierendes Kleinsignal-Ersatzschaltbild:  $R_{Ze,S} = R_{ZE,S}$ ,  $L_{Ze}$  konstant,  $C_{Ze}$  und  $R_{Ze,P}$  abhängig von  $U_{ZE}$ , Einstrahlung und Temperatur

lierungsansatz insbesondere dann sinnvoll, wenn zu erwartende höherfrequente Signalanteile relativ klein gegenüber DC- und niederfrequenten Anteilen sind. Als Grundlage kennlinien- und verhaltenbasierter Großsignalbeschreibungen dienen dabei eigene Messungen oder – speziell für quasistationäre Analysen – vom Hersteller angegebene Bauteil- oder Komponenten Kennlinien (vgl. Abb. 1(a)). Großsignalveränderungen geringer Dynamik werden dabei insbesondere bei getakteten Komponenten auf Basis üblicher sog. "averaged models" (Mittelung über kurze Zeitspannen) erfasst [2].



**Abbildung 2:** Mittels Impedanzspektroskopie gewonnene Kleinsignalimpedanz eines Solarmoduls (Parallelschaltung dreier Serien von Solarzellen): großer Wertebereich der abhängigen Kleinsignalersatzelemente nach Abb. 1(b)

Mittels impedanzspektroskopischer Messungen an unterschiedlichen Großsignal-Arbeitspunkten werden dann Kleinsignalmodelle basierend auf passiven linearen Ersatzelementen, deren Werte von den jeweiligen Großsignal-Parametern abhängen, erstellt (vgl. Abb. 1(b) und Abb. 2). Alternativ genügt in vielen Anwendungen eine im ersten Schritt resistive Ersatzschaltung, die aus der quasistationären Großsignalbeschreibung mittels Taylor-Entwicklung gewonnen werden kann, und dann modular mit ausreichend genau bekannten induktiven und kapazitiven Elementen erweitert wird. Dies gilt z. B. für hochdynamisch geregelte Wandler mit näherungsweise linearen und passiven Ein- bzw. Ausgangsfilterstufen, wenn arbeitspunktunabhängige Submodelle letzterer vorliegen bzw. mittels Messung am deaktivierten Wandler bestimmt werden können.

Unter Verwendung dieser somit gewonnenen passiven Kleinsignalersatzschaltung lässt sich das Klemmenverhalten der Komponente bei einer Einspeisung höherfrequenter Wechsel- oder Rauschsignale niedriger Amplituden abschätzen. Innerhalb der Komponente durch getaktete Halbleiterschalter und Umladeprozesse hervorgerufene AC-Signalanteile – in ihrer Gesamtheit als Schaltrippel bezeichnet – können dann im Modellansatz inkludiert werden, wenn dessen passive Struktur um AC-Ersatzquellen erweitert wird.

## 2.2 Modellierung von Leitungen

Zur Modellierung von Installationsleitungen zwischen Quellen, Verteilungsknoten und Verbrauchern sind die Leitungen in ausreichend homogene Segmente zu untergliedern.

### 2.2.1 Telegraphengleichung

Ein homogenes Leitungssegment (siehe Abb. 3) kann eindeutig durch seine Länge  $\ell$  und die seriellen Widerstands- und Induktivitätsbeläge  $r'$  [ $\Omega/\text{m}$ ] bzw.  $l'$  [ $\text{H}/\text{m}$ ] und die parallelen Leitwert- bzw. Kapazitätsbeläge  $g'$  [ $\text{S}/\text{m}$ ] bzw.  $c'$  [ $\text{F}/\text{m}$ ] mittels der Telegraphengleichung beschrieben werden. Im Laplace-Bereich stehen dabei die orts- und frequenzabhängigen Strom- und Spannungsverteilungen in folgendem Zusammenhang:

$$\begin{aligned} \frac{\partial}{\partial x} U(x, s) + l' s I(x, s) + r' I(x, s) - l' i(x, +0) &= 0, \\ \frac{\partial}{\partial x} I(x, s) + c' s U(x, s) + g' U(x, s) - c' u(x, +0) &= 0. \end{aligned} \quad (1)$$

Dabei sind  $i(x, +0)$  und  $u(x, +0)$  die Anfangsverteilungen zum Zeitpunkt  $t = 0$ .

In vielen Anwendungsfällen spielen die Querbeläge  $g'$  und  $c'$  für die Signalanalyse bei Abstraten bzw. Sensorbandbegrenzungen im Bereich von wenigen 100 kHz eine untergeordnete Rolle. Sie können insbesondere dann vernachlässigt werden, wenn der Abstand zwischen Hin- und Rückleiter relativ groß ist (z. B. durch Verlegung separater einadriger Kabel). Ist dies der Fall, lassen sich ohmsche und induktive Längsbeläge näherungsweise in konzentrierten Elementen zusammenfassen. Andernfalls ist dennoch eine Vereinfachung möglich, wenn vor einem Umschalteereignis oder dem Eintreten einer Fehlersituation eingeschwungene Verhältnisse vorliegen. Dann

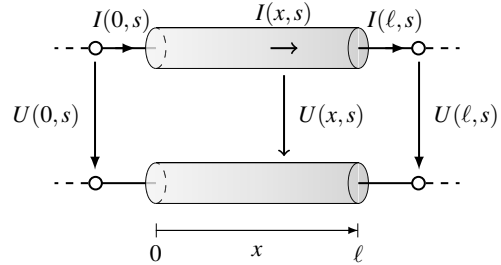


Abbildung 3: Vierpolmodell eines homogenen Leitungselements (Die Variable  $x$  ist die Ortsvariable auf der Leitung.)

kann die zum Ereigniszeitpunkt in den verteilten induktiven und kapazitiven Elementen gespeicherte Energie in konzentrierten Strom- und Spannungsersatzquellen an einem Ende des Leitungssegments ausgegliedert werden [6]. Dies erleichtert die numerische Berechnung von Strom- und Spannungsverläufen nach Ereigniseintritt auf Basis einer modularen Vierpolstruktur des Gesamtnetzwerks.

### 2.2.2 Modales Leitungsmodell

Leitungsbeläge können in der Realität erhebliche Frequenzabhängigkeiten aufweisen, die insbesondere auf den Skin-Effekt zurückzuführen sind (v. a. bei Kabeln mit Seelen aus kompakten Drähten oder Litzen aus wenigen Teildrähten). Ein Modell mit konstanten Belägen entspricht der Realität also nur in einem beschränkten Frequenzbereich.

Im Folgenden wird ein Leitungsmodell basierend auf modaler Zerlegung vorgestellt, das bereits in [7] und [8] verwendet wurde. Das Modell kann für frequenzabhängige Parameter erweitert werden, indem jede einzelne Mode mit frequenzunabhängigen Parametern modelliert wird, allerdings unterscheiden sich diese von Mode zu Mode.

Als Basis dient dafür eine impedanzspektroskopische Messung der Eingangsimpedanz bei kurzgeschlossenem Ausgang. In [12] wird beschrieben, wie das für den sekundärseitigen Kurzschluss erhaltene modale Modell mittels Koppelementen für einen allgemeinen Leitungsabschluss erweitert werden kann.

**Impedanzmodell:** Das Modell der Übertragungsleitung aus Gl. (1) kann bei sekundärseitigem Kurzschluss in eine mehrdimensionale Übertragungsfunktion überführt werden. Die theoretischen Grundlagen und der Ablauf dieser Transformation wurden ausführlich in [7], [10] und [9] unter dem

Namen Funktionaltransformations-Methode (FTM) beschrieben.

Für die Transformation der örtlichen Ableitungen wird eine Sturm-Liouville-Transformation verwendet. Deren Rücktransformation als Reihenentwicklung bildet die Basis für die Synthese. Die Eigenschaften der Transformation sind in [10] ausführlich beschrieben und basieren auf der Theorie aus [11].

Damit kann eine Synthesegleichung für die Eingangsimpedanz der sekundärseitig kurzgeschlossenen Leitung hergeleitet werden:

$$Z_{KS}(s) = \sum_{\mu} Z_{\mu}(s), \quad Z_{\mu}(s) = \frac{1}{N_{\mu}} \frac{1}{s - s_{\mu}}. \quad (2)$$

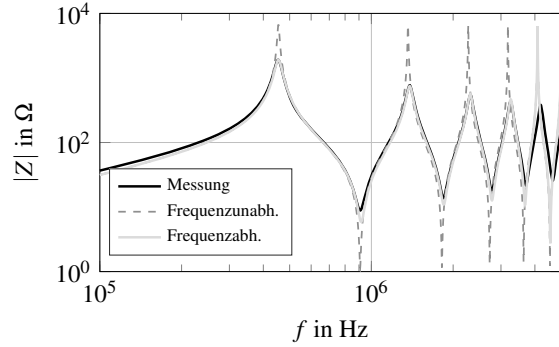
Die Impedanz ist eine Superposition der Teilimpedanzen  $Z_{\mu}(s)$  für alle Eigenfrequenzen  $s_{\mu}$  der Leitung. Die von den Leitungsbelägen abhängigen Eigenfrequenzen können durch Auswertung der Randbedingung (ausgangsseitiger Kurzschluss) bestimmt werden [7, 9].

**Anpassung der Moden:** Das Impedanzmodell aus Gl. (2) bietet die Möglichkeit, Frequenzabhängigkeiten der Leitungsbeläge einzubeziehen. Die Idee ist, dass die Frequenzabhängigkeit nicht für jeden Frequenzpunkt  $s = j\omega$  berücksichtigt wird, sondern lediglich für die Teilimpedanzen  $Z_{\mu}(s)$ . Jede einzelne wird spezifisch für konstante Beläge berechnet, allerdings können die Beläge von Teilimpedanz zu Teilimpedanz variieren.

Jedes komplex konjugierte Paar von Teilimpedanzen repräsentiert eine Mode der Übertragungsleitung (Resonanzspitzen in Abb. 4). Die Anpassung dieser Moden an Messungen führt zu einem approximativen Modell für frequenzabhängige Leitungen.

Ausgehend von geschätzten Anfangswerten werden dabei die spezifischen Leitungsbeläge der Teilimpedanzen  $Z_{\mu}(s)$  solange variiert, bis die zugehörige Mode das gewünschte Verhalten annimmt:  $r'_{\mu}$  und  $g'_{\mu}$  beeinflussen die Amplitude dieser Mode; das Produkt von  $l'_{\mu}$  und  $c'_{\mu}$  korrespondiert mit ihrer Resonanzfrequenz.

**Experimentelle Ergebnisse:** Eine Installationsleitung 1306 KEMA KEUR XMvK  $2 \times 1,5$  der Länge  $\ell = 100$  m mit Drahtseele wurde mit Impedanz-Spektroskopie vermessen (Bode 100 von OMICRON Lab). Im Frequenzbereich bis ca. 300 kHz stimmt ein Synthesemodell mit frequenzunabhängigen Belagswerten von  $r' = 0.023 \Omega/\text{m}$ ,  $g' = 1 \text{ fS/m}$ ,  $l' = 0.48 \mu\text{H/m}$  und  $c' = 63 \text{ pF/m}$  gut mit der Messung



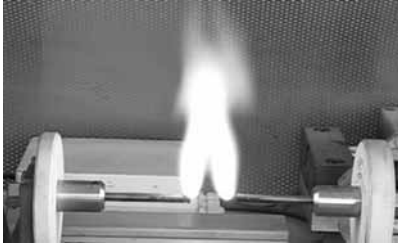
**Abbildung 4:** Vergleich der Messung (schwarz) mit dem Impedanzmodell aus Gl. (2). Frequenzunabhängiges Impedanzmodell (rot) und angepasstes Modell (grün).

überein. Bei höheren Frequenzen weichen Messung (schwarze Kurve in Abb. 4) und Modell (rot gestrichelte Kurve) aufgrund des nicht berücksichtigten Skin-Effekts jedoch stark voneinander ab – der Dämpfungseffekt des mit der Frequenz ansteigenden Widerstandsbelages und die Verringerung des induktiven Belages werden vom Modell nicht wiedergegeben.

Beim modalen Entwurf hingegen (hier mit 200 Moden) wurden die Parameter der ersten vier Moden an die Messungen angepasst – es kann eine sehr gute Übereinstimmung zwischen Messung und entsprechendem modalen Modell (grüne Kurve) bis ca. 3.5 MHz festgestellt werden. Die Anpassung der Moden kann auf einen bestimmten Frequenzbereich begrenzt werden, die offensichtlich abweichenden höheren Moden in der Abbildung könnten ebenso angeglichen werden.

### 2.3 Modellierung von Fehlersituationen

Bei seriellen oder parallelen Fehlerstellen treten üblicherweise Glühprozesse auf, bei Spannungen größer 20 V und Strömen größer 0.5 A auch Lichtbögen. Das Großsignalverhalten von Glühprozessen und Lichtbögen lässt sich dabei über stationäre physikalische Modelle abbilden (z. B. Ayerton'sches Lichtbogenmodell, vgl. [1]). Diese beinhalten neben Strom und Spannung auch Parameterinformation zur Leiterquerschnittsverengung bzw. zur Lichtbogenlänge. Die Werte letzterer sind dabei für potentielle Fehlersituationen nur grob abschätzbar, bei der modellbasierten Analyse muss somit mit einem großen Wertebereich der physikalischen Größen gerechnet werden. Dabei addiert sich ein materialabhängiger Min-



**Abbildung 5:** Lichtbogen in Testanlage mit thermodynamisch bedingter Aufwölbung

destspannungsabfall von 15-20 V – bedingt durch physikalische Prozesse an Anode und Kathode – zu einer bogenlängenabhängigen Komponente. Bei kurzen Lichtbogenlängen, also im Anfangsstadium serieller Fehler, ist somit die Großsignalmodellierung noch relativ genau. Bei größeren Bogenlängen hingegen erzeugen thermodynamische Prozesse schwieriger abbildbare und situationsabhängige niederfrequente Längenfluktuationen, vgl. Abb. 5.

Sowohl die Übergänge zwischen den verschiedenen Phasen Feststoff-Schmelze-Metaldampf-Plasma, die Bildung von Oxidschichten und ihre Absprengung sowie die dadurch bedingten Lichtbogenfußpunktwanderungen sorgen dabei für nieder- bis hochfrequente Wechsel der Leitfähigkeit. Dies kann näherungsweise im Kleinsignalverhalten durch stochastische Prozesse, d.h. durch Rausch-Ersatzquellen mit sog. rosa Rauschen auf Spannungsebene abgebildet werden. Bei unklarer Kontaktsituation, wie dies bei einem Fehlerereignis quasi immer der Fall ist, sind diese Rauschmodelle allerdings relativ ungenau.

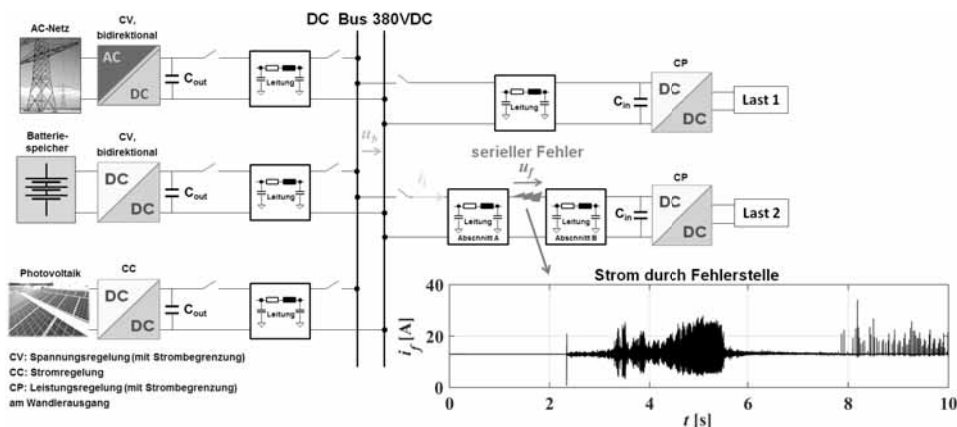
Besonders zu berücksichtigen sind Lichtbogenfehler

in DC-Verteilnetzen mit niederimpedanten Resonanzen. Diese definieren sich durch die Leitungsimpedanzen, durch die netzseitigen Filter- und Speicherkapazitäten der Baugruppen sowie durch die negativen differentiellen Eingangswiderstände der leistungsgeregelten Lastwandler (vgl. Abb. 6) – bei bestimmten Kontaktabständen können dann resonante Effekte in der Lichtbogensäule mit hohen Stromamplituden auftreten. Dynamische Modellansätze für den Lichtbogen sind erforderlich, je nach Situation können die Grundcharakteristika in einem nach [13, 14] verfeinerten linearen Kleinsignalmodell dargestellt werden oder erfordern dynamische Großsignalmodellierung in Anlehnung an Mayr'sche Modelle [15].

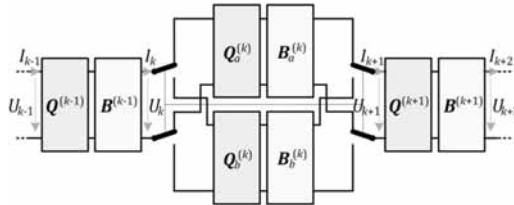
### 3 Modellierung des Gesamtsystems

Für eine Analyse der Kleinsignalanteile werden die durch geeignete Segmentierung erhaltenen Submodelle dann sinnvollerweise in einer Struktur aus verketteten Vierpolen mit möglichst linearisierten Elementen und ausgliederbaren (Ersatz-)Quellen sowie mit zusätzlichen Verzweigungselementen und Schaltern übertragen, siehe Abb. 7.

Jeder Großsignalarbeitsbereich wird dabei durch eine geeignete Kleinsignal-Vierpolstruktur repräsentiert. Bei einem Großsignalwechsel, z.B. bedingt durch Lastwechsel oder Fehlerereignisse, wird dann auf den jeweils passenden Vierpol umgeschaltet, wobei die Anfangsbedingungen der dann gültigen Struktur über ausgegliederte Ersatzquellen abgebildet werden.



**Abbildung 6:** Versuche am 380VDC-Netz des Fraunhofer IISB zu seriellen Fehlerlichtbögen: durch niederimpedante Netzresonanzen bedingtes resonantes Lichtbogenverhalten



**Abbildung 7:** Kleinsignalmodell verketteter und je nach Großsignalarbeitsbereich umschaltbarer Vierpole mit passiven und aktiven Komponenten  $B$  und  $Q$

Diese umschaltbare Vierpolstruktur ermöglicht dann die Analyse der Einflüsse von Systemkomponenten und Fehlerparametern auf sensierte Strom- und Spannungsgrößen im Frequenzbereich und – nach hierfür geeigneten Methoden der inversen Laplace-Transformation – ebenso im Zeitbereich. Dies dient dann als Basis für die Erstellung echtzeitfähiger Algorithmen zur Fehlerdetektion auf einer möglichst schlanken Zielhardware eines Schutzorgans.

## 4 Zusammenfassung und Ausblick

Modulare Modellierungskonzepte für die Komponenten von DC-Netzen sowie für Fehlersituationen mittels Groß- und Kleinsignalanalysen wurden aufgezeigt. Entsprechend für das Gesamtsystem erstellte Vierpolstrukturen können zur Sensitivitätsanalyse verwendet werden. Sie liefern die Grundlage für die Anwendung echtzeitfähiger maschineller Lernverfahren zur Fehlerdetektion auf Basis von Merkmalen im Zeit- und Frequenzbereich. Aktuell werden hierzu verschiedene rekursive Methoden, insbesondere aus der Gruppe der Kalman-Filter, anhand von Anwendungsbeispielen auf ihre Eignung untersucht.

*Die Forschungstätigkeit zu diesem Themenbereich wurde gefördert durch die Bayerische Forschungstiftung im Projekt "Vorausschauende Schutzgeräte für Gleichspannungsnetze", Nr. 1108-13.*

## Literatur

- [1] C. Strobl. *Arc fault detection - a model-based approach*. Proc. 27th International Conference on Electrical Contacts (ICEC), Dresden, 2014, S. 367-372.
- [2] S. Bacha, I. Munteanu, A. I. Bratcu. *Power Electronic Converters Modeling and Control*. London, Springer, 2014.
- [3] V. Quaschnig. *Regenerative Energiesysteme*. 7. Aufl., München, Hanser, 2011.
- [4] D. Schwander. *Dynamic Solar Cell Measurement Techniques: New Small Signal Measurement Techniques*. Space Power, Proc. of the Sixth European Conference, Porto, 2002, S. 603-608.
- [5] C. Strobl. *Störlichtbögen in Photovoltaikanlagen – modellbasierte Szenarien- und Detektierbarkeitsanalysen*. 22. Albert-Keil-Kontaktseminar, Karlsruhe, 2013, S. 64-73.
- [6] C. Strobl, M. Schäfer, R. Rabenstein *An analytical fault model for direct current lines.*, Proc. IEEE International Symposium on Circuits and Systems, Montreal, 2016, S. 2142-2145.
- [7] R. Rabenstein, L. Trautmann. *Partial differential equation models for continuous multidimensional systems*. in IEEE Int. Symp. on Circuits and Systems (ISCAS), vol. 1, Geneva, Switzerland, 2000, pp. 407-410.
- [8] G. Antonini. *Spectral models of lossy nonuniform multiconductor transmission lines*. in IEEE Transactions on Electromagnetic Compatibility, vol. 54, no. 2, pp. 474-481, April 2012.
- [9] M. Schäfer, P. Frenštátský, R. Rabenstein. *A physical string model with adjustable boundary conditions*. in 19th International Conference on Digital Audio Effects (DAFx-16), Brno, Czech Republic, September 2016, pp. 159 – 166.
- [10] S. Petrusch, R. Rabenstein. *A simplified design of multidimensional transfer function models*. in International Workshop on Spectral Methods and Multirate Signal Processing (SMMSP2004), Vienna, Austria, September 2004, pp. 35-40.
- [11] R. V. Churchill. *Operational Mathematics*. Boston, Massachusetts: Mc Graw Hill, 1972.
- [12] R. Rabenstein, M. Schäfer, C. Strobl. *Transfer Function Models for Multidimensional Systems with Impedance Boundary Conditions*. International Journal of Control, eingereicht 2016.
- [13] M. Streck, F. Nothnagel, and F. Berger. *Parameters' values of small signal equivalent circuit of electric arc gaps in dc networks*. in 50th International Universities Power Engineering Conference (UPEC), Stoke-on-Trent, UK, 2015.
- [14] M. Streck, F. Nothnagel, and F. Berger. *Impact of the contact material on the extinction process of low voltage dc arcs*. Proc. 28th International Conference on Electric Contacts (ICEC), Edinburgh, 2016, S. 99-104.
- [15] O. Mayr. *Beiträge zur Theorie des statischen und des dynamischen Lichtbogens*. in Archiv f. Elektrotechnik, Vol. 37, Nr. 12, S. 588-608, Dez. 1943.

# Gebäudesimulationen für ein Heimenenergiemanagement im Kontext zu verschiedenen Gebäudetypologien mit Solarenergienutzung

Prof. Dr.-Ing. Gerhard Mengedoht, Prof. Dr. Walter Commerell,

M.Eng. Sebastian Schmidt, M.Eng. Mitat Tuzcu

Hochschule Ulm, Institut für Energie- und Antriebstechnik (IEA)

*mengedoht@hs-ulm.de*

Gebäude- und Systemsimulationen für ein Heimenenergiemanagement im Kontext zu verschiedenen Gebäudetypologien unter Anwendung der Solarenergienutzung, Solarthermie, Photovoltaik, Wärme- und Stromspeicher, hocheffiziente Brennstoffzelle, Simulator-Kopplung, TRNSYS, Matlab-Simulink

## 1 Einleitung

Der Fahrplan der Europäischen Union zur Umsetzung einer CO<sub>2</sub>-armen Wirtschaft bis zum Jahr 2050 sieht vor, daß die EU ihre Treibhausgasemissionen um 80 % gegenüber dem Stand von 1990 senken sollte. Bei diesem Fahrplan sollen in den emissionsintensiven Bereichen Stromerzeugung, Industrie, Verkehr, Gebäude, Bauwesen und Landwirtschaft die folgenden verschiedenen Reduktionspotentiale realisiert werden [1]:

- Stromerzeugung und –verteilung 100%
- Verkehr > 60%
- Gebäude > 90%
- Industrie > 80%
- Landwirtschaft (Prognose CO<sub>2</sub>-Zunahme um 30%)

In diesem Kontext muss konstatiert werden, dass der Gebäudebereich für ca. 40% des Energiebedarfes verantwortlich ist. 50% der Weltbevölkerung leben in Städten, während in Deutschland 75% der Bevölkerung in Städten und deren Speckgürteln leben [2].

## 2 Gebäude- und Systemsimulation

Unter Gebäude-Simulation versteht man die energetische und thermische Beschreibung eines Gebäudes mit Hilfe eines Computerprogrammes. Das Simulationsprogramm berücksichtigt alle entscheidenden Einflußgrößen, die auf das energetische Verhalten (z. B. Heizwärmebedarf, Kühlkältebedarf, etc.) oder das thermische Verhalten (mittlere Raumluft-, Empfindungs- und Oberflächentemperaturen) eines Gebäudes einwirken. Ein großer Teil der Einflußgrößen ist bauphysikalischer Natur, wie z. B. Einspeicherungs- und Entspeicherungsvorgänge, Einfluß transparenter Bauteile und

Einfluß der Dämmung. Weitere Einflußgrößen sind das Wetter (Außenlufttemperatur, Außenluftfeuchte, Solarstrahlung, Windgeschwindigkeit), Verschattungseffekte, Luftaustauschvorgänge (Infiltration, Ventilation), Energie- und Stoffströme von haustechnischen Anlagen sowie Nutzerprofile (innere Lasten wie Personen, Beleuchtung, Computer etc.). Hierbei werden insbesondere sich kontinuierlich ändernde Randbedingungen (z. B. Außenlufttemperatur, direkte und diffuse solare Einstrahlung) sowie komplexe Be- und Entspeicherungsvorgänge, die man auch als Übergangs- oder transiente Vorgänge bezeichnet, berücksichtigt. Diese transienten Vorgänge üben einen maßgeblichen Einfluß z. B. auf die Heiz- und / oder Kühl-last eines Gebäudes sowie auf die Behaglichkeit für die Gebäude-Nutzer aus. Übliche statische Berechnungsverfahren (wie z. B. EnEV-Programme, Heiz-/Kühl-lastprogramme etc.) versagen hier. Gebäude- und Anlagensimulationen werden überwiegend für ein TRY Klima-Jahr (Testreferenz-Jahr) bei einer Simulations-Zeitschrittweite von einer halben Stunde durchgeführt. Jahressimulationen mit einer speziellen Anlagentechnik im Gebäude können auch mit einem Zeitschritt von bis zu 1 Minuten erfolgen.

Die Leistungsfähigkeit des dynamischen Gebäude-Simulationsprogrammes TRNSYS soll exemplarisch an einem südorientierten Büroraum (Standort Ulm) demonstriert werden.

Für den ersten Architekten-Entwurf ergab die Simulation am 2. September eine maximale Raumluft-Temperatur von 42,6 °C (Bild 1, roter Kurvenverlauf, vor Optimierung).

Da diese Temperaturen nicht akzeptabel sind, wurden dem Architekten folgenden Maßnahmen vorgeschlagen, um die maximalen Raumluft-Temperaturen abzusenken:

- der Sonnenschutz wird von innen nach außen verlegt
- der Glasflächenanteil von der Außenfassade wird von 90% auf 40% reduziert
- die abgehängte Decke wird durch eine massive Betondecke ersetzt

Die Simulationsergebnisse in Bild 1 (grüner durchgezogener Kurvenverlauf nach Optimierung) zeigen das Resultat: Das Maximum der Empfindungs-Temperatur liegt am 2. September nur noch bei 28,8 °C, d. h. die empfundene Temperatur konnte durch die drei unterschiedlichen o. a. Maßnahmen um 13,8 K von 42,6 °C aus sehr deutlich abgesenkt werden (gelber Bereich).

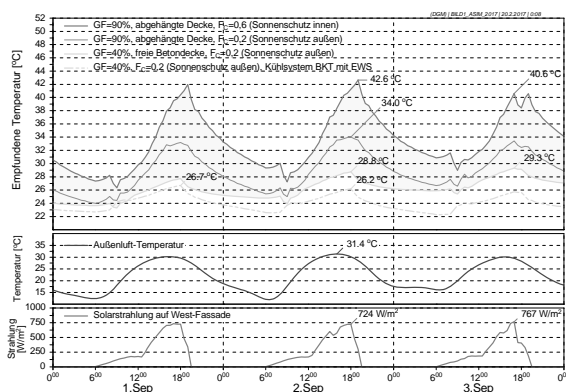


Bild 1: Sommerliche empfundene Temperaturen in einem Süd-orientierten Büroraum

Somit konnte unter diesen speziellen Randbedingungen die Kälteleistung für ein benötigtes Kühlsystem stark reduziert werden. Behagliche Temperaturen (26,2 °C) werden dann durch eine Bauteiltemperierung (BKT) gewährleistet, die umweltfreundlich z. B. durch Erdwärmesonden (EWS) bereitgestellt wird.

### 3 Gebäude- und Systemsimulationen für ein Heimenergiemanagement

In Zeiten eines Energieversorgungsnetzes, das zunehmend erneuerbare Energien integriert, ist es von großer Bedeutung, diese Energiequellen in ihrer Gesamtheit zu betrachten und passende Lösungen für eine zuverlässige und nachhaltige Energieversorgung zu entwickeln.

Das vom Bundesministerium für Wirtschaft und Energie (BMWi) geförderte Projekt iHEM (Intelligentes Heimenergiemanagement) setzt hierbei konkret auf Photovoltaik und Solarthermie im Zusammenspiel mit einer Mikro-KWK und Speichern für thermische sowie elektrische Energie im Privathaushalt [3].

Hierbei wird zusätzlich berücksichtigt, dass die Solarthermie (Solarwärme) gegenüber der Photovoltaik (Solarstrom) aus physikalischen Gründen über eine deutliche höhere energetische Flächeneffizienz (Ertrag Jahresenergie pro m<sup>2</sup> Solarapertur-Fläche) verfügt.

Um fossile Energieträger so weit wie möglich zu umgehen, liegt es nahe, Strom und Wärme aus erneuerbaren Energien in einem System zu kombinieren. Des Weiteren stellt dieser Ansatz zusammen mit elektrischen und thermischen Speichern eine effiziente Lösung für den Umgang mit Einspeisebegrenzung und Überschussstromverwertung dar. Zur Gewährleistung der Versorgungssicherheit im Gebäude vereint das Projekt iHEM ein Heimenergiemanagementsystem mit einer Brennstoffzelle. Diese wird hocheffizient mit Erdgas betrieben.

Voraussetzung zur rationellen Gebäudeenergieversorgung ist eine optimale Dimensionierung der Systemkomponenten, die nur unter genauer Kenntnis des Energiebedarfs möglich ist. Der Energiebedarf eines Gebäudes ist von diversen unterschiedlichen Faktoren abhängig, wodurch eine entsprechende Vielfalt an Freiheitsgraden resultiert. Zur Ermittlung des Wärmebedarfs für Wohngebäude entwickelt die Hochschule Ulm deshalb komplexe Simulationsmodelle mit dem Fokus auf Neubauten und Sanierungsobjekte. Die modellierten Gebäude repräsentieren mit ihren Abmessungen in Deutschland gängige Gebäudetypen (Einfamilien- und Mehrfamilienhäuser) und sind beispielsweise in Bezug auf Standort, Ausrichtung, Bauphysik und Nutzerverhalten flexibel anpassbar.

Durch die Verknüpfung eines Gebäudesimulationsmodells mit weiteren Modellen, wie z.B. Solarthermie- und PV-Anlage, Wärme- und Stromspeicher, Brennstoffzelle, etc. zu einer Gesamtsystemsimulation (siehe Abbildung 2), ist unter Angabe eines definierten Szenarios und dem zugehörigen Betriebsmodus das zur jeweiligen Gebäudetypologie passende Energieversorgungssystem dimensionierbar.

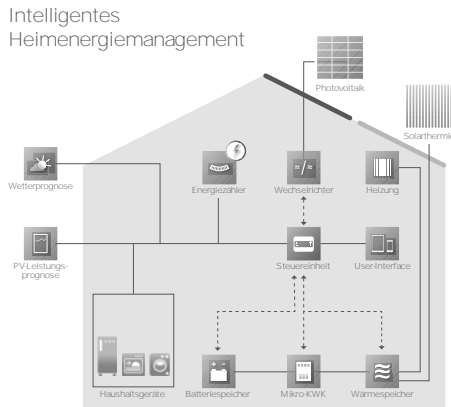


Bild 2 Komponenten-Übersicht im Projekt Intelligentes Heimenenergiemanagement (iHEM) [3]

Des Weiteren können durch die Simulation typische Kennzahlen wie z.B. Autarkie- und Eigenverbrauchsquote ermittelt werden, die den Vergleich sowie Benchmarking verschiedenster Energiesystem-Konstellationen ermöglichen.

Als erste Anwendung der Gesamtsystems simulation wurde ein iHEM-Demonstratorsystem entwickelt, welches derzeit aufgebaut wird. Der Demonstrator ist die reale Umsetzung der im Projekt iHEM entwickelten Verfahren und Komponenten, welcher neben dem Nachweis der Funktionalität des Gesamtsystems eine Validierungsmöglichkeit von Simulationen bietet.

Die Hochschule Ulm ergänzt den Demonstrator in einem Bestandsgebäude in Ehingen (Baden-Württemberg) um ein Monitoring-System zur Messdatenerfassung.

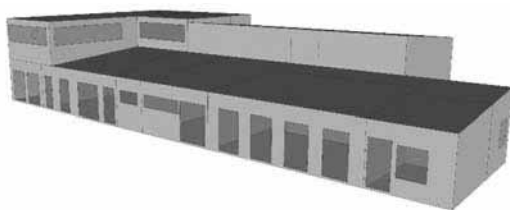


Bild 3 Sketchup-TRNSYS-Gebäudesimulationsmodell des Gebäudes für den iHEM-Demonstrator [4]



Bild 4 Zonierung des Gebäudesimulationsmodells für den iHEM-Demonstrator [5]

## Danksagung

Teile der o. a. Untersuchungen wurde im Rahmen des EnOB/EnBop Verbundvorhaben „Intelligentes Heimenenergiemanagementsystem (iHEM) mit Nutzung des Synergiepotenziale der thermischen und elektrischen Objektversorgung mit PV, Batterie- und Wärmespeicher, Solarthermie und KWK“ als Teilvorhaben „Gebäudetypen, Evaluation Pufferspeicher, Anbindung Wärmequellen“, durchgeführt.

Die Förderung durch den Projektträger Jülich unter dem Aktenzeichen Förderkennzeichen 03ET1205F erfolgt mit Mitteln des Bundesministerium für Wirtschaft und Energie (BMWi) aufgrund eines Beschlusses des deutschen Bundestages. Die Autoren danken für diese Unterstützung.

## 4 References

- [1] Auer, Thomas, Startseite des Lehrstuhls für Lehrstuhl für Gebäudetechnologie und klimagerechtes Bauen an der TU-München, <https://www.klima.ar.tum.de/startseite/> 2017
- [2] Europäische Kommission zur Klimapolitik, [https://ec.europa.eu/clima/policies/strategies/2050\\_de](https://ec.europa.eu/clima/policies/strategies/2050_de), Brüssel, 2017
- [3] Mengedoht, Gerhard; Stern, Katharina, Poster über den Projektanteil der Hochschule Ulm am iHEM Forschungsprojekt anlässlich vom 1. Kongress „ENERGIEWENDEBAUEN - Forschung zu effizienten Technologien und intelligenten Konzepten für die Strom- und Wärme-wende“ in Berlin vom 30. - 31.01.2017
- [4] Tuzcu, Mitat, Grafik vom TRNSYS Modell zum iHEM-Demonstrator, Hochschule Ulm, Ulm, 2017
- [5] Schmidt, Sebastian, Grafik zum iHEM-Demonstrator, Hochschule Ulm, Ulm 2017

# Sektorkopplung als Baustein in der Energiewende – Ergebnisse einer bidirektionalen Simulation von Energienetzen

David E. Stakic<sup>1</sup>, Falko Ebe<sup>1</sup>

<sup>1</sup>Hochschule Ulm, Institut für Energie- und Antriebstechnik, SmartGrids Forschungsgruppe  
*stakic@hs-ulm.de*

Die Kostenplanung für den Ausbau der Verteilstromnetze ist für die Netzbetreiber in einem zunehmend auf dezentrale Stromerzeugung setzenden Energiesystem eine große Herausforderung. Bisher wurden in urbanen Gebieten die drei Netze zur Verteilung von Energie (Strom, Wärme und Gas) gesondert betrachtet und potenzielle Synergieeffekte nicht untersucht. Um den aufkommenden Herausforderungen zu begegnen sowie für die Sektorkopplung eine Planungsgrundlage zu entwickeln wurde im Europäischen Forschungsprojekt OrPHEuS als Methode ein Co-Simulationsmodell entwickelt. Dabei wurde auf reale Daten der Infrastruktur, Dachpotenziale, Anlagen und Meteorologie zurückgegriffen. Diese Daten wurden sowohl für die Entwicklung der Simulationsmodelle der Netze und Anlage verwendet, als auch für die Entwicklung realitätsnaher Szenarien und Use Cases sowie Kontrollstrategien der Anlagen. Es wurden drei Hybrid-Szenarien entwickelt, die die unterschiedlichen Energienetze einbeziehen. Außerdem wurden innerhalb der Hybridszenarien Kontrollstrategien für die involvierten Anlagen entwickelt. Der Ansatz zur Aufbereitung der Rohdaten für das Netzmodell bedarf in erster Linie einer umfangreichen Bearbeitung der Topologie und Sachdaten. Zur Koordinierung der Datenflüsse wurde eine standardisierte Schnittstelle basierend auf dem „Functional Mock-up Interface Framework“ entwickelt. Dieses stellt sicher, dass die Netzmodelle in PowerFactory (Stromnetz) und Dymola (Wärmenetz) sowie die Funktionalität von Controllern zusammen simuliert werden können. Die Ergebnisse lassen eine detaillierte dynamische Bewertung von Netzzuständen zu, die weit über die heutige Informationsdichte im Netzbetreiber hinausgeht.

## 1 Einleitung

Der Ausbau der erneuerbaren und dezentralen Stromerzeugung erfolgte und erfolgt in Deutschland weiterhin überwiegend in den unteren Stromnetzebenen. Dabei entstehen neue Herausforderungen in den Bereichen Spannungsband, Kabel- und Transformatorbelastung. Konventionell wird diesen Herausforderungen durch die Verstärkung des Netzes mit neuen oder zusätzlichen Leitungen oder Transformatoren begegnet. Dies ist jedoch mit hohen Kosten sowie erheblichen Eingriffen durch Baumaßnahmen in Siedlungsgebieten verbunden. Neben dem Stromnetz können die anderen Energienetze (Gas und Wärme) zur Unterstützung herangezogen werden. In diesem Fall spricht man von Sektorkopplung. Kopplungstechnologien sind z.B. Blockheizkraftwerke (BHKW), Elektrolyse und Methanisierung sowie elektrische Heizstäbe und Durchlauferhitzer. Diese können die nötigen Leistungen und Leistungsgradienten für Regelleistung bereitstellen [10]. Das Wärme- als auch das Gasnetz bringen weitere Eigenschaften, die einen sicheren Stromnetzbetrieb ergänzen. Dazu zählen:

- Zusätzliche Verbraucher, die Überschüsse im Stromnetz direkt aufnehmen können

- Erhöhung der Verteilungskapazitäten sowie
- Speichereigenschaften im Bereich von Stunden und Tagen, ebenso wie Monaten im Fall von Methan (Erdgas).

Neben den einzelnen Komponenten der Netze sowie deren Kopplungsanlagen spielen Einsatz- und Kontrollstrategien eine entscheidende Rolle. Dieses sind die Komponenten, welche aus einem messtechnisch erschlossenen Netz ein Smart Grid machen. Sogenannte Multi-Agenten-Systeme erfordern eine komplexe Logik, die nicht ohne weiteres in gängigen Softwaretools abgebildet werden kann.

## 2 Untersuchung von Optimierungsmöglichkeiten von Hybriden Netzen

### 2.1 Ansatz

Für eine effiziente Auslegung und Planung von Energienetzen müssen Wechselwirkungen zwischen den Energienetzen evaluiert werden. Die Simulation liefert dabei ein umfangreiches Werkzeug, um sowohl statische Ansätze für die Planung als auch dynamische Ansätze für den Betrieb von Energienetzen detailliert bewerten zu können. Netzmodelle werden in spezifischen Softwaretools (z.B. PowerFactory für das

Stromnetz, STANET für das Gasnetz, Dymola für das Wärmenetz) erstellt. Dazu ist eine umfangreiche Datenbasis sowie eine automatisierte Lösung zum aggregieren und transformieren der Daten notwendig. Die Datenbasis für die Netzinfrastruktur liefern Geoinformationssysteme (GIS) der Netzbetreiber. Dabei müssen die Daten an die spezifischen Anforderungen der Software zur Netzberechnung angepasst oder ergänzt werden.

Netzmodelle können bisher noch nicht als kommerzielle „Fertiglösung“ bidirektional simuliert werden. Hierfür sind z.B. Open Source Software-Tools wie mosaik<sup>1</sup> nötig. Für große Netzgebiete mit tausenden von Knoten ist eine Kopplung der Netzmodelle mit erheblichem Aufwand verbunden. Im EU-Projekt OrPHEuS wurde eine solche Co-Simulation im Smart Grid Testgebiet (TG) Einsingen durchgeführt. Die Co-Simulation ermöglicht hier, netzübergreifend detaillierte Ergebnisse zu erzeugen, die für die Entwicklung neuer Ansätze entscheidend sind.

Um belastbare Ergebnisse erhalten zu können, wurde in OrPHEuS der Entwicklung der Datenbasis sowie der simulierten Szenarien ein hoher Stellenwert eingeräumt. Dies verdeutlicht auch die Relevanz einer Kooperation interdisziplinärer Fachbereiche für gute Simulationsergebnisse. Um einen Einblick in dieses Feld in OrPHEuS zu bekommen und die Ergebnisse besser einordnen zu können, werden im folgenden Kapitel die simulierten Szenarien beschrieben.

## 2.2 Szenarien

Konventioneller Netzausbau beinhaltet, neue Kabel und Transformatoren aufzubauen, um die Aufnahmefähigkeit von PV-Leistung zu erhöhen, dies ist jedoch eine der teuersten Lösungen [2]. Alle Szenarien, die für die Co-Simulation entwickelt wurden, werden mit dem Referenzszenario, das als „Baseline- Szenario“ bezeichnet wird, verglichen. Als Rahmenbedingungen wurden im „Baseline Szenario“ Grenzen des Spannungsbands und der Netzbetriebsmittelbelastung berücksichtigt. Außerdem wurden Daten basierend auf einer Solardachpotenzialanalyse, PV-Modul Nennleistung der Bestandsanlagen und Verbrauchswerte der Stromzähler verwendet um Photovoltaik (PV)- Ausbauszenarien zu entwickeln. [1]

<sup>1</sup> Mosaik ist ein flexibler Smart Grid Co-Simulation Framework. <https://mosaik.offis.de/>

Durch die aktuelle Funktion des EEG in Deutschland, zunehmende Strompreise für Privatkunden und abnehmenden Einspeisetarifen für PV, ist der Eigenverbrauch des Stroms aus PV die Hauptmotivation für Geschäftskonzepte bei PV Anlagen. Daher wurde für die PV-Ausbauszenarien der Stromverbrauch als Haupteinfluss auf die Leistung der PV definiert.

Der konventionelle Weg von Verteilnetzbetreibern (VNB), den Herausforderungen der dezentralen Stromerzeugung in Verteilnetzen zu begegnen, stellt den technischen und ökonomischen Rahmen für das Baseline Szenario aus Netzbetreibersicht. Dabei wurden drei Hauptszenarien („Hybrid-Szenarios“) identifiziert.

Eine Übersicht der maximalen PV-Überschüsse im TG Einsingen für verschiedene PV-Zubauszenarien oberhalb eines Lastfaktors<sup>2</sup> von 0,8 p.u. am Transformator im simulierten Jahr ist in **Tabelle 1** aufgelistet. Ein Lastfaktor von 0,8 p.u. für den Transformator wurde vom VNB auf Basis von Erfahrungswerten festgelegt und entspricht der üblichen maximalen Belastung. Ein zusätzliches Ziel war es, den Eigenverbrauch des TG Einsingen darüber hinaus zu bewerten. Es wurde eine Rückspeisung in die überlagerten Netzebenen in der Höhe von 0,8 p.u. des Transformators als akzeptabel definiert. Diese elektrische Leistung steht im Mittelspannungsnetz einer sehr viel größeren Anzahl von Verbrauchern in der Region gegenüber und kann so zeitlich direkt genutzt werden. Der Bezug aus den höheren Netzebenen wird so vermieden. Überregionale konventionelle bzw. kerntechnische Kraftwerke stellen den Strom dort in Verbindung mit hohen CO2 Emissionen bereit. Solange dies der Fall ist, ist es sinnvoll, Netzkapazitäten auszunutzen und dem Eigenverbrauch in Wärme oder Gas vorzuziehen.

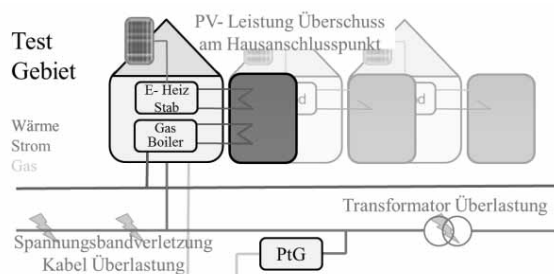
Szenario	MWh/a	MWh/d (max/a)	MW (max/a)
50%-PV Überschuss	297	3,52	0,63
75%-PV Überschuss	669	6,50	1,05
100%-PV Überschuss	961	8,64	1,34

**Tabelle 1.** PV Überschüsse im Testgebiet Einsingen

<sup>2</sup> Der Lastfaktor stellt das Verhältnis von Nennleistung zur aktuellen Belastung dar

In **Abbildung 1** sind die drei „Hybrid-Szenarien“ zur Orientierung auf den folgenden Beschreibungen schematisch dargestellt. Bewertungskriterien in allen drei „Hybrid-Szenarien“ sind:

- Anzahl und Dauer der Spannungsband-Verletzung
- Vermeidung von Belastung für Netzbetriebsmittel
- Der Einfluss auf das Gas- und Fernwärme-Netz
- Reduktion von Netzausbaukosten
- Reduktion von Treibhausgas-Emissionen



**Abbildung 1:** Schematische Darstellung der „Hybrid-Szenarios“ im TG Einsingen [5]

Das „Hybrid-Szenario“ 1 beschreibt einen lokalen auf Prosumenten<sup>3</sup> und Endkunden abzielenden Power to Heat<sup>4</sup> (PtH) Ansatz. Für alle PV-Ausbauszenarien wurden vor der Simulation Vergleiche untersucht, die unterschiedliche Kombinationen von Heizstäben für Warmwasser und Heizenergie sowie Wärmespeicher vergleichen. Dabei wurde für das TG Einsingen eine einheitliche Kombination von Leistung (Heizstab) und Speicherkapazität (Wärmespeicher) pro Haushalt ermittelt. Die Speicher wurden nach Planungsregeln für die Auslegung von Wärmespeichern rechnerisch dimensioniert [3].

Das „Hybrid- Szenario“ 2 ist nahe an den Annahmen des ersten Szenarios angelehnt, zusätzlich wird ein Fernwärmenetz mit berücksichtigt. Durch das zusätzliche Netz und die zusätzliche Verteilkapazität werden neue Verbraucher erschlossen, die direkt mit den PV Überschüssen, besonders in den Sommermonaten, versorgt werden können. Für das „Hybrid-Szenario“ 2 kann es sinnvoll sein, Wärmepumpen zu berücksichtigen. Die höhere Effizienz der Wärmepumpen kann in

diesem Szenario voll ausgeschöpft werden. Für die Co-Simulation wurde ein Szenario entwickelt, das mit den Erfahrungswerten des Netzbetreibers am vielversprechendsten erscheint. Dabei wurde aufgrund der Verfügbarkeit, Zuverlässigkeit, Steuerbarkeit sowie Investitionskosten der Heizstab der Wärmepumpe vorgezogen. [5]

Im „Hybrid- Szenario“ 3 wird der Ansatz des „Hybrid-Szenario“ 1 erweitert. Dabei werden Warmwasser und Heizungswärme berücksichtigt sowie das Gasnetz mit einbezogen. Dadurch können mehrere Vorteile der PtH- und PtG-Technologien kombiniert werden. Aufgrund des höheren Wirkungsgrades sowie der Möglichkeit, dezentral im Niederspannungsnetz zur Spannungsregulierung und Kabelbelastung einzugreifen, soll primär PtH zum Einsatz kommen. Dabei kommen thermische Kurzzeitspeicher (Stunden) zum Einsatz. PtG soll zusätzliche PV-Überschüsse am Transformator<sup>5</sup> nutzen und gleichzeitig durch die Speicherwirkung des Gasnetzes weitere Wärmelasten als Puffer (Tage) und saisonaler Speicher (Wochen und Monate) adressieren. [5]

In urbanen Gebieten mit Fernwärmenetz ist empfehlen, trotz der Speichermöglichkeiten des Gasnetzes und wegen des höheren Umwandlungswirkungsgrades von PtH zu, die technisch möglichen Verteilungspotenziale des Fernwärmenetzes auszuschöpfen, bevor PtG zu Einsatz kommt. [5]

### 3 Daten der Infrastruktur

#### 3.1 Daten aus realen Testgebieten

Neben der Verwendung von synthetischen Netzen aus Forschungsansätzen wie z.B. [9], sind für die konkrete Ermittlung der Wirkung in realen Netzen, diese auch abzubilden. Hierfür ist es erforderlich, umfangreiche Datensätze aus Bestandsystemen von VNB, inklusive der damit zusammenhängenden Ungenauigkeiten und Fehler, zu importieren. Der Prozess der Aufbereitung und Transformation von Daten kann mitunter ein Vielfaches der Untersuchungszeit in Anspruch nehmen. Jedoch führt dies ggf. zu weniger abweichenden Ergebnissen als bei abstrahierten Netzen, weil Besonderheiten mit berücksichtigt werden können [8].

<sup>3</sup> Prosumenten sind Kunden die sowohl Energie bereitstellen als auch beziehen.

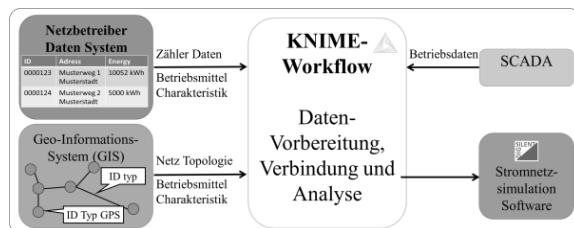
<sup>4</sup> Power to Heat meint das Nutzen von elektrischer Leistung zur Bereitstellung von Wärme.

<sup>5</sup> Transformator als Netzverknüpfungspunkt zur nächst Höheren Spannungsebene, hier Mittelspannung.

### 3.2 Aufbereitung

Die im vorliegenden Untersuchungsfall beschriebenen Netzmodelle wurde aus den Rohdaten des beteiligten Projektpartners<sup>6</sup> erstellt. Wichtige Schritte (vgl. **Abbildung 2**) sind hierbei:

- Aufbereitung/ Reparatur der Topologie
- Anreicherung mit Sachdaten
- Transformation in Datenmodell der Simulationsumgebung
- Erstellung von geeigneten Grafiken für die Arbeit mit den Netzmodellen



**Abbildung 2** Übersicht der Systemarchitektur, hier sind die drei wichtigsten Datenquellen zusammen mit der Netzberechnungssoftware (Power System Analysis Software) PowerFactory visualisiert. Der KNIME7-Workflow bildet die Verbindung der Softwarekomponenten.

Der erste Schritt ist die Aufbereitung der Rohdaten und die Erstellung einer entsprechenden Topologie. Ggf. muss ein auf Annahmen basierender Korrekturalgorithmus implementiert werden. Abhängig vom Netztyp unterscheidet sich die Topologie signifikant. Im Besonderen sind hier die Unterschiede zwischen Strom und Medienetzen zu nennen. Medienetze (Gas & Fernwärme) werden mit einem hohen Vermaschungsgrad gebaut, entsprechend einfach ist der Korrekturalgorithmus. Alle im Umkreis befindlichen, eines an die Druckebene angepassten Radius, werden zusammengefasst, und die entsprechenden Kantelemente (Rohrleitungen) werden auf diesen Punkt verbunden. Neben der Topologie und den Leitungselementen zugehörigen Sachdaten werden ebenso Modelle für Verbrauchs- bzw. Erzeugungsanlagen benötigt. Diese können im realen Umfeld aus den Systemen der Kategorie Enterprise-Ressource-Planning (ERP) sein.

Der nachfolgende Schritt ist die Umformatierung der Daten in ein Transferformat, welches von der Simula-

tionssoftware unterstützt wird. Im Fall von PowerFactory ist dies durch das proprietäre Datenformat DGS gegeben, welches die relationale Datenbank des Systems fast identisch abbildet. Für den STANET Import ist dies ebenso möglich. Hierbei ist jedoch eine größere Freiheit bezüglich der Nomenklatur gegeben.

### 3.3 Netzmodelle in der Netzberechnungssoftware

Kernpunkt eines jeden Netzmodells ist die Topologie des Netzes in Form eines Knoten-Kanten-Modells, welches die einzelnen Verbraucher, Erzeuger oder Einspeiser aus höherer Netzebene an den Knoten über Leitungen miteinander verbindet. Für jedes Energienetz werden individuell die relevanten Parameter und physikalischen Vorgänge dargestellt.

## 4 Der Co-Simulations Ansatz

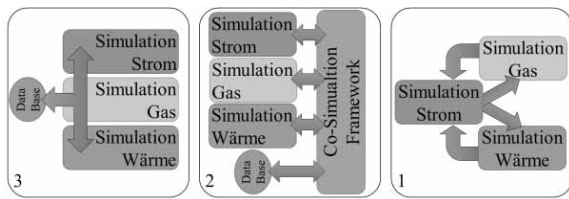
### 4.1 Kategorien der Verflechtung von Simulationssystemen

Abhängig vom Grad der Verflechtung der Simulationsumgebungen, können mehrere Kategorien erstellt/eingeordnet werden. Diese Unterscheidung ist in **Abbildung 3** in drei Stufen dargestellt. Als Endausbau können Totally Integrated Co-Simulation Systems (TICSS) gesehen werden (Stufe 3). Diese verknüpfen die unterschiedlichen Domänen in einer einheitlichen Systemoberfläche. Die Verknüpfung reicht aber weiter als nur bis zur Oberfläche. Modelle, Datenstrukturen, Programmier- und Skriptschnittstellen sind vereinheitlicht. Dies ist nach aktuellem Stand für die in Europa/Deutschland relevanten Mehrspartennetzberechnungsprogramme<sup>8</sup> der Fall. Eine Vorstufe zeigt sich insbesondere im Forschungsumfeld. Hierbei kommt eine Simulationsumgebung zum Einsatz, welche auf das jeweilige Simulationssoftwaresystem zugreift, die Simulation steuert sowie die Daten austauscht. Diese Stufe findet auch im Forschungsprojekt OrPHEuS Anwendung (Stufe 2). Die am wenigsten fortgeschrittene Kopplung ist die iterative Simulation mit Ergebnissdaten der jeweils anderen Simulationsumgebung (Stufe 1). Hierbei kann man im eigentlichen Sinne nicht von einer Schnittstelle sprechen. Diese Lösung ist nicht für den Einsatz mit mehreren Kopplungspunkten mit hohem Aufwand verbunden.

<sup>6</sup> Stadtwerke Ulm/Neu-Ulm Netze GmbH, <https://www.ulm-netze.de/>

<sup>7</sup> KNIME ist eine freie Software für die interaktive Datenanalyse.

<sup>8</sup> Neplan: [www.neplan.ch](http://www.neplan.ch); Siemens: [www.siemens.de/pss-sincal](http://www.siemens.de/pss-sincal); Ingenieurbüro Fischer-Uhrig: [www.stafu.de](http://www.stafu.de)



**Abbildung 3** Stufen von Co-Simulationsansätzen

## 4.2 Softwarelösung

Eine entscheidende Funktion jeder Co-Simulationsumgebung ist die Koordinierung und Ausführung der Datenflüsse der individuellen Komponenten. In OrPHEuS wurde eine Functional Mock-up<sup>9</sup> Interface (FMI) verwendet, um dieser Aufgabe zu begegnen. Das entwickelte Framework baut auf Ptolemy II<sup>10</sup> sowie der FMI++ Library<sup>11</sup> auf. [3]. Das Co-Simulations-Framework wurde entwickelt für Simulationssoftware, welche FMI-Schnittstellen unterstützt.

Für das thermische Netzmodell wurde die Software Dymola genutzt. Dymola bietet eine integrierte Funktion, die es erlaubt Simulationsmodelle der FMI Spezifizierung entsprechend zu exportieren. [6]

Für das elektrische Netzmodell liefert die genutzte Software PowerFactory keine FMI-Anpassungsschnittstelle. Mithilfe der FMI++ Anwendung wurde eine FMI-Schnittstelle entwickelt. Diese ermöglicht mit Modellen zu arbeiten, die folgendes erfüllen: [6]

Für eine Interaktion der verschiedenen Modelle innerhalb der Co-Simulation wird eine geeignete Kontrollstrategie benötigt. Dieser Controller muss eine Schnittstelle für das FMI aufweisen. Dies bedingt eine Reihe von Anforderungen: [6]

- Es wird ein Werkzeug benötigt, das eine Exportfunktion nach den FMI-Anforderungen unterstützt bzw. muss eine geeignete FMI-Schnittstelle entwickelt werden, z.B. unter Zuhilfenahme der FMI++ Library. [6]
- Es muss eine gründliche Definition der Ein- und Ausgangswerte durchgeführt werden, die den Ansprüchen der unterschiedlichen Netzmodelle gerecht werden. [6]
- Der Controller sollte diskrete Eingangswerte bearbeiten und ausgeben. [6]

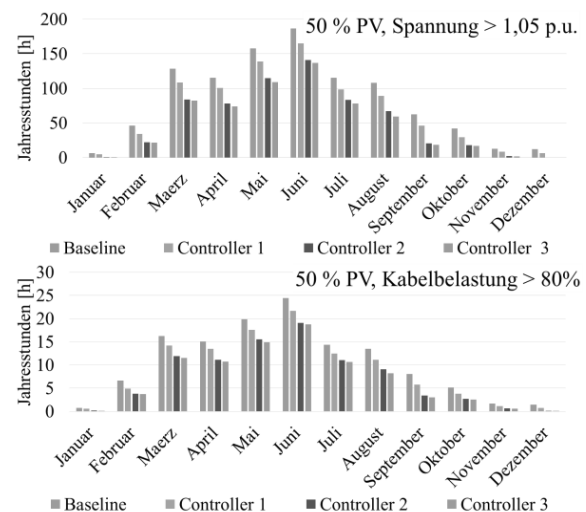
<sup>9</sup> Zum Koppeln verschiedener Simulationssoftware wird eine standardisierte Schnittstelle definiert, die als Functional Mock-up Interface oder FMI bezeichnet wird.

<sup>10</sup> Ptolemy II ist ein generisches Framework zur Modellierung und Simulation von wechselseitigen und simultan laufenden Prozessen. Er wurde entwickelt am Zentrum für

Für das Gasnetzmodell wurde die Simulations- und Planungssoftware STANET genutzt. Diese bietet keine kompatible Exportfunktion nach FMI-Schnittstellen. Aufgrund der Anforderungen im Projekt OrPHEuS wurde darauf verzichtet, eine Schnittstelle mithilfe der FMI++ Library zu entwickeln. Dabei wurde zunächst das PtG-Modell mit den Ergebnisdaten aus der Co-Simulation (Stromnetz) in KNIME modelliert und simuliert. Die so berechneten Daten (Gas) wurden im Gasnetzmodell in STANET weiter verarbeitet.

## 5 Fazit und Ergebnisse

Die Kombination von realen Netzdaten, praktikablen Szenarien und Simulationen hat dem Verteilnetzbetreiber eine Entscheidungsgrundlage für weitere Maßnahmen zu einem nachhaltigen, dezentralen und sektorgekoppelten Energiesystem geliefert. Die Ergebnisse liefern eine solide und quantifizierbare Grundlage, die Herausforderungen neuer Technologien berücksichtigt sowie deren Potenziale sichtbar machen. **Abbildung 4** und Fehler! Verweisquelle konnte nicht gefunden werden. beschreiben zwei wichtige Parameter, die der Verteilnetzbetreiber heute nur mit hohem Aufwand und Einsatz von Ressourcen bestimmen kann. Diese sind jedoch der Hauptgrund für die meisten Netzausbaumaßnahmen im Verteilnetz.



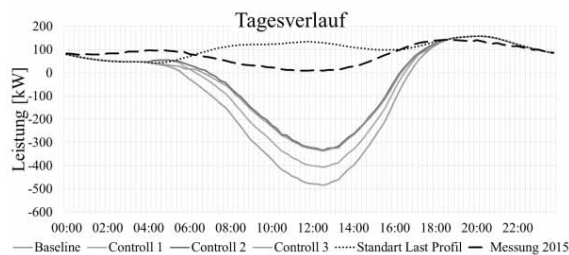
**Abbildung 4** Stunden mit Kabelbelastung größer 80 % und Spannungsbandüberschreitungen größer 1,05 p.u

„Hybrid and Embedded Software Systems“ (CHES) an der University of California, Berkeley. [4]

<sup>11</sup> The FMI++ library wird genutzt, um die FMI-Simulationskomponenten zu verbinden. [3]

für das Hybrid Szenario 1 und das 50 %- PV Szenario, unter Berücksichtigung des Warmwasserbedarfs.

In **Abbildung 5** wurde eine weitere für den Netzbetrieb kritische Größe visualisiert. Es zeigt sich, dass der Lastverlauf am Transformator durch die PV-Einspeisung eine signifikante Veränderung erfährt und, dass die Einspeiseleistung der PV-Anlagen die bestimmende Größe für die Auslegung des Transformators ist. Durch die Maßnahmen, wie sie im „Hybrid Szenario“ 1 beschrieben werden, konnte je nach Kontrollstrategie die maximale Belastung in **Abbildung 5** um 16-30 % reduziert werden. Dies wäre nicht nur für die Netzplanung eine entscheidende Verbesserung. Im Netzbetrieb kann durch die Leistungsreduktion während hoher Außentemperaturen die Alterung des Transformators sehr stark reduziert werden. In der Norm DIN 60076-7 wird ein exponentiell steigender Lebensdauerverbrauch in Abhängigkeit der Öltemperatur beschrieben [7].



**Abbildung 5** Mittlerer Tagesverlauf eines Jahres der Leistung am Transformator für das Hybrid Szenario 1 das 50 %- PV Szenario, unter Berücksichtigung des Warmwasserbedarfs.

## 6 Disclaimer

Diese Ausarbeitung basiert auf den Inhalten und wurde unterstützt durch die Forschungsprojekte Nathan- PV (BMBF Förderkennzeichen 03FH03013) sowie dem Forschungsprojekt OrPHEuS (Europäische Kommission, 7th Framework Program, Grant agreement no: 608930)

Die alleinige Verantwortung für die Inhalte dieser Publikation liegt bei den Autoren. Sie reflektieren nicht zwingendermaßen die Meinung der europäischen Kommission. Die Europäische Kommission ist nicht verantwortlich für eine Nutzung, die aufgrund der beinhaltenen Informationen durchgeführt werden könnten.

## 7 References

- [1] H. Ruf et al, "Active Grid Planning Based on Solar Power Roof Potential Analysis," in 27th European Photovoltaic Solar Energy Conference: Proceedings of the International Conference held in Frankfurt a.

M, 24-28 September 2012, S. Nowak, A. Jäger-Waldau, and P. Helm, Eds, München: WIP, 2012, pp. 3782–3787.

- [2] M. Schwarz and A. Kollmann, "Wirtschaftliche Betrachtung von Netzausbau und Smart Grid-Lösungen in der Niederspannungsebene," *Elektrotech. Inftech*, vol. 131, no. 3, pp. 99–104, Jahr 2014.
- [3] D. Basciotti, S. Henein, E. Widl, C. Marguerite, H. Brunner, R. R. Schmidt; Austrian Institute of Technology AIT; Deliverable 4.1, WP4 System Modelling and Simulation for the Evaluation of the OrPHEuS Control Strategies; Project: OrPHEuS, Optimising Hybrid Energy grids for smart cities; Jahr 2015
- [4] A. Schülke, T. Jacobs, S. Nicolas, A. Papageorgiou, H. Auer, D. Schwabeneder, D. Basciotti, S. Henein, E. Widl, M. Schroedter-homscheidt, G. Heilscher, D. Stakic, K. Dietz, I. Weiss; WP7 Project Results Evaluation and Conclusions, Deliverable 7.1.2, Impact of Cooperative Control Strategies for cities' Hybrid Energy Networks on European scale; Project: OrPHEuS, Optimising Hybrid Energy grids for smart cities; Jahr 2017
- [5] D. Stakic et al, Simulation of local Energy Surplus Usage in Hybrid Grids with a High PV Penetration Rate; EU PVSEC, Munich; Year 2016; Poster + Paper
- [6] H. Brunner, D. Basciotti, S. Henein, E. Widl; WP4 System Modelling and Simulation for the Evaluation of the OrPHEuS Control Strategies, Deliverable 4.2, Technical Requirements Definition for Cooperative Control Strategies; Project: OrPHEuS, Optimising Hybrid Energy grids for smart cities; Jahr 2015
- [7] David E. Stakic, Holger Ruf, Falko Ebe, Gerd Heilscher, Florian Meier; Simulation of the Lifetime Consumption from Oil-immersed Transformers with Consideration of the Further Development of Decentralized PV Feed-in; ETG Congress; Germany, Bonn; Year 2015; Paper + Poster
- [8] Ebe, Falko; Morris, Jeremie; Ruf, Holger; Heilscher, Gerd; Meier, Florian; Feeder Analyses of a Medium Voltage Grid with the Utilization of a Solar-Roof-Potential-Analyses; In: 6th Solar Integration Workshop - Vienna - Proceedings, Year 2016, S. 456 – 461; ISBN: 978-39816549-3-6
- [9] Kerber, Georg; Aufnahmefähigkeit von Niederspannungsverteilnetzen für die Einspeisung aus Photovoltaikkleinanlagen; Dissertation; Technische Universität München; Jahr 2011
- [10] regelleistung.net; gesichtet 17.02.2017  
<https://www.regelleistung.net/ext/static/prequalification-bzw.-transmission-code-2007>

# Leistungsverteilung in Elektrofahrzeugen mit Range Extender

Sören Scherler<sup>1</sup>, Xiaobo Liu-Henke<sup>2</sup>

Ostfalia Hochschule für angewandte Wissenschaften, Fakultät Maschinenbau,

Institut für Mechatronik, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel

<sup>1</sup>*so.scherler@ostfalia.de*, <sup>2</sup>*x.liu-henke@ostfalia.de*

Der vorliegende Beitrag beschreibt die modellbasierte Entwicklung einer Betriebsstrategie zur energieeffizienten Leistungsverteilung in einem Elektrofahrzeug mit Brennstoffzelle (BZ) als Range Extender (RE). Grundlage hierfür stellt ein Fahrzeugmodell basierend auf Differentialgleichungen dar. Die Betriebsstrategie basiert auf einem prädizierten zeit- und energieoptimalen Geschwindigkeitsprofil. Mithilfe dieses Profils und eines inversen Antriebsmodells kann die benötigte elektrische Leistung berechnet werden, welche wiederum die Grundlage für eine Ladestandsprädiktion darstellt. Diese Verläufe werden verwendet, um in einem iterativen Verfahren die vom RE aufzubringende Leistung zur Erzielung eines gewünschten Zielladestandes zu ermitteln.

## 1 Einleitung

Der Wandel der verbrennungsmotorischen Mobilität hin zur Elektromobilität wird von vielen Automobilherstellern seit der Abgasaffäre um Volkswagen deutlich stärker forciert. Die breite Marktdurchdringung elektrischer Fahrzeuge wird neben hohen Anschaffungskosten und fehlender Infrastruktur durch die mit derzeitigen Energiespeichern erzielbaren Reichweiten, welche wesentlich geringer verglichen zu konventionellen Fahrzeugen ausfallen, verhindert. Um dieses Reichweitenproblem zu lösen bieten sich Elektrofahrzeuge mit einer BZ als RE an, da der Wirkungsgrad höher als bei Verbrennungsmotoren ausfällt und eine, je nach Erzeugung des Wasserstoffs, zumindest lokal emissionsfreie Fahrt ermöglicht wird.

In diesem Beitrag wird modellbasiert eine Betriebsstrategie zur Leistungsverteilung in einem Range Extended Vehicle (REV) entworfen, um einen möglichst energieeffizienten Fahrbetrieb zu erreichen.

## 2 Stand des Wissens

Generell lässt sich zwischen heuristischen Betriebsstrategien für Fahrzeuge mit einem „konventionellen“ Fahrer und Betriebsstrategien für autonome Fahrzeuge unterscheiden. In diesem Beitrag wird ein Fahrzeug mit mindestens automatisierter Längsführung fokussiert, um Vorteile prädiktiver Algorithmen zur energieoptimalen Fahrt nutzen und einregeln zu können.

In [1] werden in einer Datenbank bereits gefahrene Strecken (u.a. GPS-Koordinaten und Geschwindigkeiten) gespeichert. Bei einer neuen Fahrt werden die

GPS-Daten des Navigationsgerätes genutzt, um die Strecke mit Daten aus der Datenbank zu beschreiben. Mithilfe dieser Daten werden ein Geschwindigkeits- und Leistungsverlauf prädiziert und einer globalen Optimierung hinsichtlich Fahrkomfort und Energieverbrauch unterzogen und ein energieoptimiertes Leistungsprofil für den als RE eingesetzten Verbrennungsmotor berechnet. Diese Berechnung wird zu Fahrtbeginn durchgeführt und alle 100 ms aktualisiert.

Gausemeier [2] verfolgt einen auf erweiterten Kartendaten basierenden Ansatz zur Prädiktion eines energie- und zeitoptimalen Geschwindigkeitsprofils für Elektrofahrzeuge. Ein Vorausschau-Assistent bildet aus Navigationsdaten und unter Restriktionen wie zulässiger Höchstgeschwindigkeit einen Lösungsraum, in dem mittels Mehrziel-Optimierung und dyn. Programmierung das Geschwindigkeitsprofil gebildet wird.

Einen vergleichbaren Ansatz verfolgen Biermann et al. [3] an einem Parallelhybrid mit Verbrennungsmotor und Synchronmaschine. Die vom Fahrer durch Fahr- und Bremspedalstellung vorgegebenen Wunschmomente werden mithilfe eines Vorausschaumoduls basierend auf prädizierten Kartendaten und Abstandsinformationen der Fahrzeugsensorik hinsichtlich Antriebsstrangwirkungsgrad angepasst und von einer Advanced Adaptive Cruise Control eingeregelt.

## 3 Methodik

Für die Entwicklung mechatronischer Systeme hat sich die durchgängig modellbasierte und verifikationsorientierte Funktionsauslegung und -absicherung (vgl. **Abbildung 1**) als zeit- und kosteneffizient erwiesen.

Ausgehend von Anforderungen und Lastenheft erfolgt die Model-in-the-Loop-Simulation (MiL), in welcher Regelalgorithmen ohne Anspruch auf Echtzeitfähigkeit an einem Fahrdynamikmodell erprobt werden. Aus den simulativ erprobten Regelalgorithmen wird im Rahmen der Software-in-the-Loop-Simulation (SiL) mittels automatischer Code-Generierung ein ausführbarer Target-Code erzeugt, welcher wiederum mit einem Fahrdynamikmodell auf Funktion und Fehler getestet wird. Daraufhin erfolgt die Hardware-in-the-Loop-Simulation (HiL), bei welcher ein um physische Teilkomponenten ergänztes echtzeitfähiges Fahrdynamikmodell zur Funktionsabsicherung und -optimierung verwendet wird. Während aller Schritte ist eine enge Verflechtung mit Messungen am realen Fahrzeug notwendig, um die Modelle und Funktionen frühzeitig verifizieren und absichern zu können. Mithilfe dieser Methodik wird eine durchgängige Entwicklung basierend auf virtuellen Prototypen bis zum fertigen Produkt sichergestellt. Im Rahmen dieser Arbeit wird eine MiL-Simulation durchgeführt.



Abbildung 1. Durchgängig modellbasierte, verifikationsorientierte Entwicklungsmethodik nach [4].

## 4 Fahrzeugkonzept

Der konzeptionelle Aufbau des Fahrzeugs ist in **Abbildung 2** dargestellt. Für eine performante Traktion sind vier Radnabenmotoren mit einem Dauermoment von je 200 Nm und einem Spitzenmoment von je 500 Nm vorgesehen [5]. Zum Bremsen des Fahrzeugs werden die elektrischen Antriebe zur Rekuperation und bei Überschreiten der maximal rekuperierbaren Leistung eine hydraulische Reibbremse verwendet. Für die Versorgung leistungsstarker Bordnetzverbraucher stehen eine Lithium-Ionen-Traktionsbatterie (HV-Bordnetz) und für die Versorgung leistungsschwacher Verbraucher ein konventionelles 12-V-Bordnetz (NV-Bordnetz) zur Verfügung. Das HV-Bordnetz versorgt das NV-Bordnetz und kann an einer Ladesäule oder durch die als RE verwendete PEM-BZ, für die ein zusätzlicher Wasserstofftank verbaut ist, geladen werden.

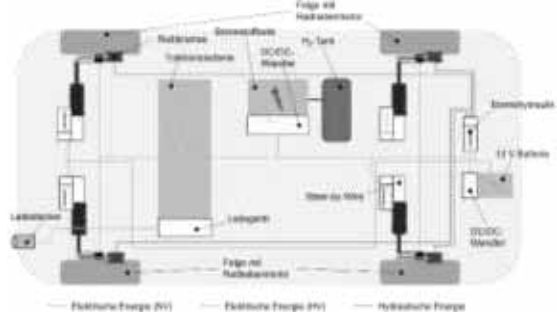


Abbildung 2. Fahrzeugkonzept.

## 5 Modellbildung

### 5.1 Fahrdynamik

Es wird lediglich die Längsdynamik des Fahrzeugs betrachtet, da diese wesentliche Rückschlüsse auf den Energieverbrauch und somit die Reichweite des Fahrzeugs zulässt. Unter Vernachlässigung des Radschlupfes kann nach dem Newton'schen Ansatz

$$\sum F = ma \quad (1)$$

die Beschleunigung  $a$  der Fahrzeugmasse  $m$  aus der Summe aller Antriebs- und Fahrwiderstandskräfte  $F$  bestimmt werden. In  $m$  sind alle relevanten rotatorischen Massenträgheiten enthalten. Die Summe aller Kräfte bzw. die Beschleunigungskraft

$$\sum F = F_a = F_{Mot} - F_{Br} - F_{St} - F_R - F_L \quad (2)$$

setzt sich aus der in Längsrichtung wirkenden Kraft der vier Radnabenmotoren  $F_{Mot}$ , der Bremskraft  $F_{Br}$ , der Steigungswiderstandskraft  $F_{St}$ , der Rollwiderstandskraft  $F_R$  und der Luftwiderstandskraft  $F_L$  zusammen. Die Motorkraft

$$F_{Mot} = \frac{4}{r} M_{Mot} \quad (3)$$

ergibt sich aus den gleichmäßig verteilten Drehmomenten  $M_{Mot}$  der vier Motoren und dem Reifenradius  $r$ , die Steigungswiderstandskraft

$$F_{St} = mg \sin \alpha \quad (4)$$

aus der Gewichtskraft des Fahrzeugs  $mg$  und dem Steigungswinkel  $\alpha$ , die Rollwiderstandskraft

$$F_R = mgf \quad (5)$$

ebenfalls aus der Gewichtskraft sowie dem Rollwiderstandsbeiwert  $f$  und die Luftwiderstandskraft

$$F_L = \frac{1}{2} \rho_L c_w A v^2 \quad (6)$$

aus der Luftdichte  $\rho_L$ , dem Widerstandsbeiwert  $c_W$ , der Stirnfläche  $A$  und der Fahrzeuggeschwindigkeit  $v$ . Die zurückgelegte Strecke

$$s = \int v \, dt = \iint a \, dt \quad (7)$$

wird durch Integration von  $v$  und diese wiederum durch Integration von  $a$  bestimmt.

## 5.2 Antrieb und Bremse

Die elektrischen Radnabenmotoren werden zur Bewegung des Fahrzeugs benötigt und stellen die leistungstärksten Verbraucher im Elektrofahrzeug dar. Unter Vernachlässigung des Schlupfes zwischen Rad und Straße kann die Winkelgeschwindigkeit des Motors

$$\omega_{Mot} = \omega_{Rad} = \frac{v}{r} \quad (8)$$

aus der Fahrzeuggeschwindigkeit  $v$  und dem Reifenradius  $r$  bestimmt werden, da sie der Winkelgeschwindigkeit  $\omega_{Rad}$  entspricht. Das Motormoment entspricht dem Wunschemoment des Fahrers in den Grenzen der maximalen Momente für Rekuperation und Antrieb. Das Moment wird durch die thermische Belastbarkeit, den Betrieb in Feldschwächung sowie übergeordnete Regelalgorithmen begrenzt. In [5] liegt eine mathematische Beschreibung des Motorwirkungsgrades

$$\eta(M, \omega) = \frac{1,9687 \cdot M\omega - 0,03269 \cdot \omega^2 - 4,0055 \cdot \omega}{0,075424 \cdot M^2 + 1,9687 \cdot M\omega} \quad (9)$$

vor. Die elektrischen Größen des Motors liegen nicht vor und müssen aus den vorhandenen mechanischen Größen berechnet werden, da die Motorphysik zur Erzielung akzeptabler Simulationszeiten nicht durch ein elektrisches Ersatzschaltbild (ESB) modelliert wird. Mithilfe der Motorkonstante  $c_{Mot}$  wird aus dem Drehmoment der durch den Motor fließende Strom

$$i_{Mot} = \frac{M_{Mot}}{c_{Mot}} \quad (10)$$

berechnet. Der die HV-Batterie belastende Strom

$$i_{HV} = \frac{u_{Mot}}{u_{HV}} \cdot i_{Mot} \quad (11)$$

ein anderer, da  $i_{Mot}$  aus der Leistungselektronik mit der pulsweitenmodulierten Spannung  $u_{Mot}$  hervorgeht, die kleiner oder gleich der Batteriespannung  $u_{HV}$  ist. Die Motorspannung

$$u_{Mot} = \frac{P_{el}}{i_{mot}} = \frac{P_{mech}}{i_{Mot} \cdot \eta(M, \omega)} = \frac{\omega \cdot M_{Mot}}{i_{Mot} \cdot \eta(M, \omega)} \quad (12)$$

wird aus der elektrischen Motorleistung  $P_{el}$ , welche sich aus  $P_{mech}$  und  $\omega$  ergibt, berechnet.

Falls das Rekuperationsvermögen der Antriebe nicht ausreicht, wird eine hydraulische Bremsanlage als Rückfallebene verwendet. Diese wird nicht im Detail modelliert, sondern mit einem maximalen Bremsmoment von 500 Nm pro Rad festgelegt.

## 5.3 Fahrermodell

Zur Erprobung werden verschiedene Fahrzyklen wie der CADC verwendet. Ein Regler für die Längsdynamik ersetzt den Fahrer und gibt die Antriebs- und Bremsmomente vor, sodass die Referenzgeschwindigkeit  $v_{Ref}$  vom Fahrzeug eingeregelt wird.

## 5.4 Bordnetz

Das Bordnetz besitzt zwei Spannungsebenen und die Anbindung an einen RE (vgl. **Abbildung 3**). Im HV-Bordnetz werden leistungsstarke Verbraucher wie die elektrischen Antriebe mit Leistung versorgt. Die HV-Batterie besteht aus Lithium-Ionen-Zellen und weist eine Nennspannung von 330 V sowie eine Maximalspannung von 400 V auf. Für das Laden der Batterie an einer Ladesäule ist ein Lader integriert. Durch einen unidirektionalen DCDC-Wandler wird das NV-Bordnetz mit Leistung aus dem HV-Bordnetz versorgt. Eine Bleibatterie liefert die NV-Spannung zwischen 12 bis 14 V, mit welcher leistungsschwache Verbraucher versorgt werden. Durch einen weiteren unidirektionalen DCDC-Wandler kann die BZ das HV-Bordnetz mit elektrischer Leistung versorgen, welche entweder zum Laden der Batterie oder zur Versorgung elektrischer Verbraucher genutzt werden kann.

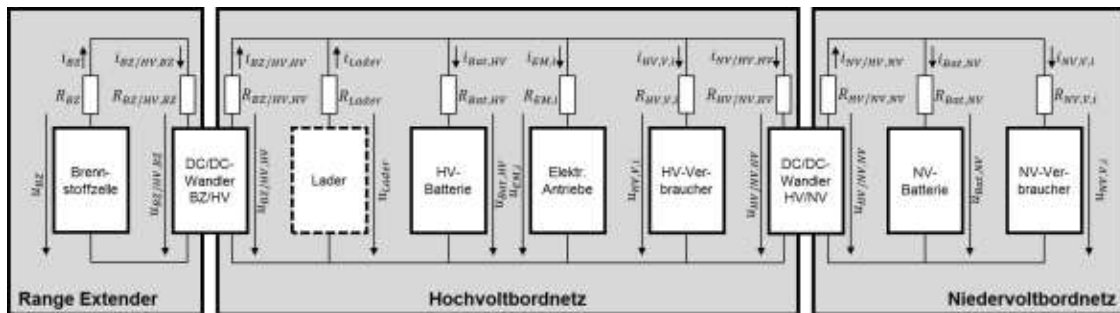
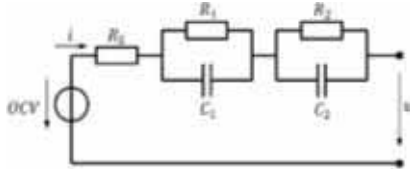


Abbildung 3. Aufbau des Bordnetzes.

### 1) HV- und NV-Batterien

Als HV-Speicher werden LiFeYPO<sub>4</sub>-Zellen genutzt. Der Spannungsbereich liegt zwischen 2,8 und 4,0 V bei einer Nennspannung von 3,3 V. Die Nennkapazität beträgt 64,8 Ah. Die Batterie wird durch ein Elektrisches Ersatzschaltbild (ESB, vgl. **Abbildung 4**) beschrieben, in dem eine Spannungsquelle die nichtlineare Ruhespannung  $OCV$  (Open Circuit Voltage) darstellt. Der Innenwiderstand wird durch  $R_S$  und die Dynamik durch zwei RC-Glieder approximiert.



**Abbildung 4.** ESB der HV-Batterie.

Die Parameter sind nichtlinear abhängig u.a. von der Temperatur und dem  $SOC$ . Durch das 2. Kirchhoff'schen Gesetzes kann die Klemmenspannung

$$u = OCV - Ri - \frac{1}{c_1} \int \left( i - \frac{u_{c1}}{R_1} \right) dt - \frac{1}{c_2} \int \left( i - \frac{u_{c2}}{R_2} \right) dt \quad (13)$$

abhängig von der  $OCV$ , der am Innenwiderstand abfallenden Spannung und den dynamischen Spannungen an den RC-Gliedern bestimmt werden. Der Ladestand

$$SOC = SOC_0 + \frac{1}{c_n} \int \eta i dt \quad (14)$$

ergibt sich durch Integration des mit dem Wirkungsgrad  $\eta$  behafteten Batteriestroms, Normierung auf die Zellkapazität  $C_n$  unter Berücksichtigung eines Startwerts  $SOC_0$ . Parametrierung und Modellvalidierung können [6] entnommen werden. Das Modell der NV-Batterie unterscheidet sich nur durch eine zusätzliche Induktivität vom Modell der HV-Batterie. Es wurde für eine Varta E44 mit 77 Ah parametrisiert.

### 2) Brennstoffzelle

Die BZ soll der des Toyota Mirai mit einer Leistung von 114 kW entsprechen, um auch einen leistungsintensiven Betrieb der Antriebsmaschinen ohne Batterie zu ermöglichen. Als Grundlage wird das Simulationsmodell aus [7] verwendet und entsprechend umparametrisiert, da keine Möglichkeit zur Vermessung einer BZ bestand. Es handelt sich um ein experimentell validiertes, empirisches Modell. Die Spannung der BZ

$$u_{BZ} = A - B \ln \left( 1 + \frac{i_{BZ}}{C} \right) - D e^{\frac{i_{BZ}}{E}} \quad (15)$$

ergibt sich aus empirischen Parametern  $A, B, C, D$  und  $E$  sowie dem Strom  $i_{BZ}$ . Der Strom

$$i_{BZ} = i_{Last} + i_{dyn} \quad (16)$$

setzt sich aus dem Laststrom  $i_{Last}$  und dem dynamischen Anteil

$$i_{dyn} = \frac{u_{BZ}}{Z} \quad (17)$$

zusammen. Durch den dynamischen Strom  $i_{dyn}$  und die Impedanz  $Z$  wird das dynamische Verhalten approximiert. Nach [8] wird der Wirkungsgrad

$$\eta_{BZ} = \frac{E_Z}{E_H^0} \quad (18)$$

als Verhältnis der Zellspannung  $E_Z$  zur thermoneutralen Spannung  $E_H^0$  berechnet. Die Zellspannung

$$E_Z = \frac{u_{BZ}}{n_{zellen}} \quad (19)$$

wird mithilfe der Anzahl im Stack befindlichen Brennstoffzellen  $n_{zellen}$  und der Stackspannung  $u_{BZ}$  bestimmt. Die thermoneutrale Spannung

$$E_H^0 = \frac{\Delta RH^0}{zF} \approx 1,147 \text{ V} \quad (20)$$

wird aus der Reaktionsenthalpie von Wasserstoff  $\Delta RH^0$ , der Ladungszahl  $z$  als Maß der an der Reaktion beteiligten Elektronen und der Faraday-Konstante  $F$  ermittelt. Der Verbrauch der Arbeitsgase Wasserstoff wird nach [9] als Massestrom abhängig von der Anzahl der Zellen  $n_{zellen}$  sowie der Stromstärke  $i_{BZ}$  bestimmt. Es wird ein Wasserstoffspeicher mit einer Füllmenge von 5 kg Wasserstoff vorgesehen.

### 3) Klimatisierung und Innenraum

Es wird ein Innenraummodell verwendet, welches das thermische Verhalten mittels ESB beschreibt, sodass ein Modell der geregelten Klimatisierung mit entsprechender Leistung das Bordnetz belastet. Eine detaillierte Beschreibung ist [10] zu entnehmen.

### 4) Verbraucher und DCDC

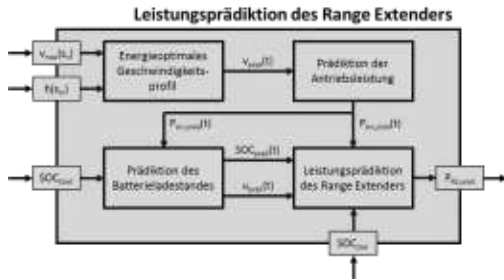
Alle Verbraucher bis auf elektrische Antriebe und Klimatisierung wurden mit einer konstanten Leistung berücksichtigt und den binären Zuständen ein- oder ausgeschaltet abgebildet. Für die DCDC-Wandler wurden konstante Wirkungsgrade von 85% angenommen.

## 6 Prädiktive Betriebsstrategie

Die prädiktive Betriebsstrategie berechnet ein energieeffizientes Leistungsprofil für den RE.

## 6.1 Struktur

Die Leistungsprädiktion des RE besteht aus vier Teilfunktionen (vgl. **Abbildung 5**). Aus erweiterten Kartendaten des Navigationssystems liegen die zulässige Höchstgeschwindigkeit  $v_{max}$  über den Wegpunkten  $s_n$  und das Höhenprofil  $h$  über den Wegpunkten  $s_m$  vor. Die Funktion *Energieoptimales Geschwindigkeitsprofil* prädiziert aus diesen Daten die Fahrzeuggeschwindigkeit  $v_{präd}$  für die vorausliegende Fahrtzeit  $t$ .



**Abbildung 5.** Struktur der Leistungsprädiktion des RE.

Basierend auf diesem Verlauf erfolgt die *Prädiktion der Antriebsleistung*  $P_{An,präd}(t)$ . Mithilfe dieses Leistungsprofils und dem aktuellen Ladestand  $SOC_{Start}$  werden Verläufe für Ladestand und Batteriespannung prädiziert, bevor diese Informationen letztlich gemeinsam mit einem gewünschten Ladestand  $SOC_{Ziel}$  am Ende der Fahrt zur Berechnung eines Leistungsverlauf für den RE genutzt werden.

Im Folgenden werden die Funktionen zur Prädiktion von Antriebsleistung, Ladestand und RE beschrieben. Eine ausführliche Betrachtung des zeit- und energieoptimalen Geschwindigkeitsprofils kann [11] und [12] entnommen werden.

## 6.2 Prädiktion der Antriebsleistung

Die von den Motoren aufzubringende mechanische Antriebsleistung des Fahrzeugs

$$P_{An,mech}(t) = F_{Mot}(t) \cdot v_{präd}(t) \quad (21)$$

ergibt sich aus der durch alle Motoren aufzubringenden Kraft  $F_{Mot}(t)$  in Längsrichtung und der prädizierten Fahrzeuggeschwindigkeit. Durch Einsetzen von  $F_{Mot}(t)$  (vgl. Gl. 2) ergibt sich die Antriebsleistung

$$P_{An,mech}(t) = \left( m a_{präd}(t) + m g \sin \alpha + m g f + \frac{1}{2} \rho_L c_W A v_{präd}^2(t) \right) v_{präd}(t) \quad (22)$$

in Abhängigkeit der prädizierten Geschwindigkeit und der daraus bestimmaren Beschleunigung sowie diversen Konstanten. Die elektrische Leistung der Motoren

$$P_{An,el}(t) = P_{präd}(t) = \frac{P_{An,mech}(t)}{\eta(M(t), \omega(t))} \quad (23)$$

kann mithilfe des Motorwirkungsgrades (hier wurden alle Motoren zu einem Motor zusammengefasst) berechnet werden. Die Motorgrößen

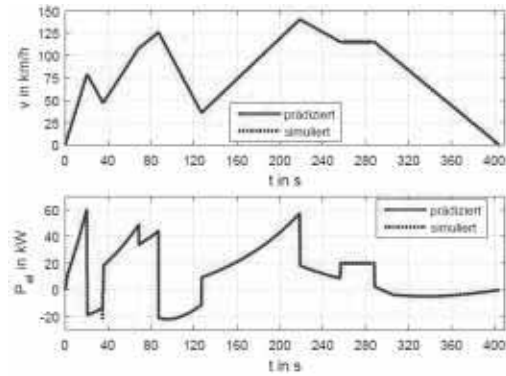
$$\omega(t) = \frac{v_{präd}(t)}{r} \quad (24)$$

sowie

$$M(t) = \frac{P_{An,mech}(t)}{\omega(t)} = \frac{P_{An,mech}(t) \cdot r}{v_{präd}(t)} \quad (25)$$

können aus den prädizierten Daten berechnet werden, sodass letztendlich  $P_{präd}(t)$  in Abhängigkeit bekannter, zeitvarianter Größen bestimmt werden kann.

In **Abbildung 6** ist ein exemplarisches Validierungsszenario dargestellt. Einer Simulation des Fahrzeugmodells und dem Prädiktionsalgorithmus werden identische Geschwindigkeitsverläufe (oben) vorgegeben. Die simulierte elektrische Leistungsaufnahme stimmt, bis auf eine geringe Abweichung bei 35 s aufgrund numerischer Abweichungen, mit dem prädizierten Verlauf überein, sodass der Algorithmus als validiert betrachtet wird.



**Abbildung 6.** Antriebsleistungsprädiktion.

## 6.3 Prädiktion des Batterieladestandes

Der Batterieladestand wird mithilfe eines invertierten Batteriemodells prädiziert. Das Leistungsprofil  $P_{präd}(t)$  entspricht der Batterieleistung

$$P_{bat}(t) = P_{präd}(t), \quad (26)$$

welche aus prädizierter Klemmenspannung  $u_{präd}$  und prädiziertem Strom  $i_{präd}$  berechnet werden kann:

$$P_{bat}(t) = u_{präd}(t, SOC_{präd}) \cdot i_{präd}(t) \quad (27)$$

Aus Gründen der Übersichtlichkeit wird in den folgenden Formeln der Index *präd* für  $SOC$ ,  $u$  und  $i$  nicht weiter verwendet. Die Klemmenspannung

$$\begin{aligned}
 u(t, SOC) &= OCV(SOC) + R(SOC) \cdot i \\
 &+ \frac{1}{C_1} \int \left( i - \frac{u_{RC1}(SOC)}{R_1(SOC)} \right) dt \\
 &+ \frac{1}{C_2} \int \left( i - \frac{u_{RC2}(SOC)}{R_2(SOC)} \right) dt
 \end{aligned} \quad (28)$$

ist abhängig von Zeit und SOC, sodass diese Abhängigkeiten bei der Leistungsberechnung berücksichtigt werden müssen. Da das Leistungsprofil  $P_{\text{präd}}(t)$  nicht kontinuierlich, sondern zeitdiskret mit konstantem  $\Delta t$  zwischen zwei Zeitpunkten vorliegt, wird auch die vorliegende Gleichung diskret betrachtet und gelöst. Mit  $k = 2, 3, \dots, n$  kann die Klemmenspannung

$$\begin{aligned}
 u_k &= OCV(SOC_{k-1}) + R(SOC_{k-1}) \cdot i_{k-1} \\
 &+ u_{RC1,k-1} + \frac{i_{k-1} - \frac{u_{RC1,k-1}}{R_1(SOC_{k-1})}}{C_1(SOC_{k-1})} \Delta t \\
 &+ u_{RC2,k-1} + \frac{i_{k-1} - \frac{u_{RC2,k-1}}{R_2(SOC_{k-1})}}{C_2(SOC_{k-1})} \Delta t
 \end{aligned} \quad (29)$$

berechnet werden. Die Initialisierung für  $k = 1$  erfolgt mit den Batteriegrößen im Equilibrium. Der Strom

$$i_k = \frac{P_{\text{bat},k}}{u_k} \quad (30)$$

ergibt sich aus Batterieleistung und Klemmenspannung. Mithilfe des Stromes kann abhängig von Batteriekapazität  $C$ , dem vorherigen Ladestand  $SOC_{k-1}$  und dem Wirkungsgrad  $\eta_c$  der aktuelle Ladestand

$$SOC_k = SOC_{k-1} + \frac{i_k \cdot \eta_c}{C} \cdot \Delta t \quad (31)$$

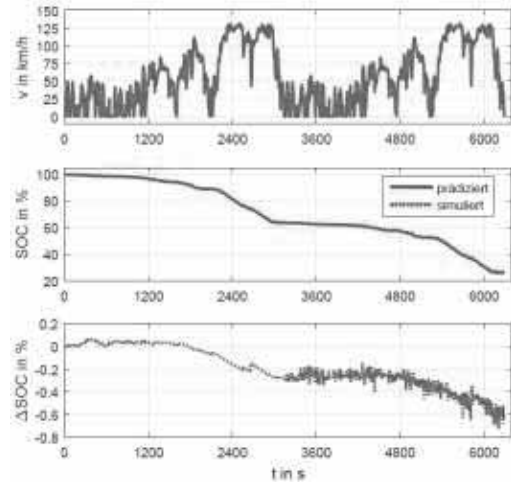
prädiziert werden. Durch den Zusammenhang

$$t = (k - 1) \Delta t \quad (32)$$

kann  $SOC_k$  mit  $k = 1, 2, \dots, n$  über der Zeit als  $SOC_{\text{präd}}(t)$  dargestellt werden. Zusätzlich liegt durch die Prädiktion des Ladestandes die prädizierte Batteriespannung  $u_{\text{präd}}(t)$  vor.

**Abbildung 7** zeigt die Ergebnisse einer Validierung am Fahrzeugmodell unter Vernachlässigung aller Leistungen außer der Antriebsleistung. Oben ist das sowohl für Prädiktion als auch Simulation verwendete Geschwindigkeitsprofil dargestellt. In der Mitte sind die Verläufe des SOC dargestellt und es zeigt sich eine gute Übereinstimmung zwischen Simulation und Prädiktion. Unten ist die SOC-Differenz dargestellt. Bei einer Entladung von 100 auf 25% ergibt sich zum Ende eine Abweichung von -0,6%, welche auf das Fahrermodell und numerische Abweichungen der Prädiktion zurückzuführen ist. Bezogen auf den Zeitraum von 6400 s ist

die Abweichung sehr gering, sodass der Algorithmus als hinreichend genau und validiert betrachtet wird.



**Abbildung 7.** Validierung der Ladestandsprädiktion.

#### 6.4 Leistungsprädiktion des Range Extenders

Wirkungsgradbedingt ist es hinsichtlich des Wasserstoffverbrauchs am sinnvollsten, den RE über einen möglichst langen Zeitraum mit einer möglichst niedrigen Leistung zu betreiben, um einen gewünschten SOC am Fahrtende zu erreichen. Mithilfe des Ladestandes am Ende des Prädiktionshorizonts  $T$  kann die vom RE aufzubringende Ladestandsdifferenz

$$\Delta SOC = SOC_{\text{präd}}(T) - SOC_{\text{Ziel}}(T) \quad (33)$$

berechnet werden. Die Ladestandsdifferenz als normierte Größe der in der Batterie gespeicherten Ladung kann mithilfe der Batteriekapazität  $C$  und des Prädiktionshorizontes  $T$  in einen konstanten, vom RE aufzubringenden, Strom

$$i_{RE} = \frac{\Delta SOC \cdot C}{100\% \cdot T} \quad (34)$$

umgerechnet werden. Mithilfe der gemittelten Spannung  $\bar{u}_{\text{präd}}$  wird eine konstante Leistung

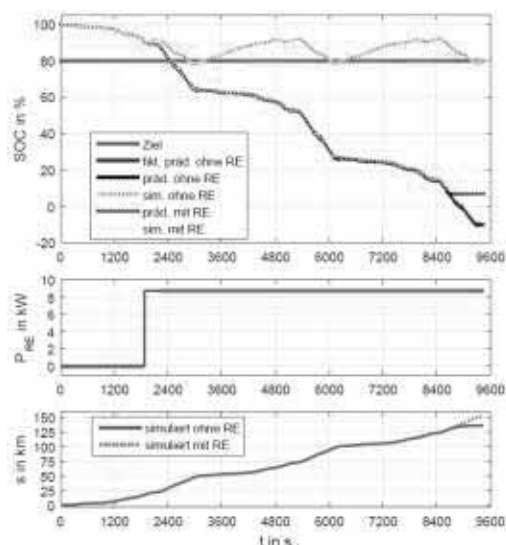
$$P_{RE} = \bar{u}_{\text{präd}} \cdot i_{RE} \quad (35)$$

bestimmt. Mit RE wird erneut ein SOC zum Fahrtende berechnet und die Abweichung zum Zielladestand bestimmt. Durch Einsatz des RE ergibt sich eine neue mittlere Spannung  $\bar{u}_{\text{präd}}$ , sodass auch der Strom  $i_{RE}$  ein anderer ist. Die RE-Leistung wird iterativ angepasst, bis die Abweichung weniger als 0,1% beträgt.

Falls die Batterie zu Beginn der Fahrt einen SOC über 90 % aufweist, wird der RE nicht zugeschaltet um genügend Reserven für Rekuperation zu haben. Reicht

die Ladung der Batterie nicht zum Erreichen des Ziels aus, also würde das Auto liegenbleiben, wird ein fiktiver *SOC* bis zum Erreichen des Fahrtziels berechnet.

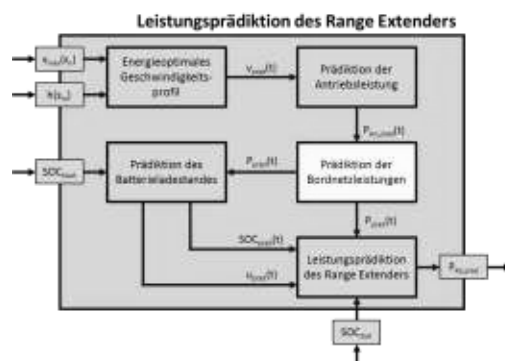
Diese ganzen Aspekte sind im Testszenario in **Abbildung 8** erkennbar. Der Startladestand beträgt 100% und lässt keine Ladung der Batterie zu. Weiterhin reicht der Ladestand nicht aus, um drei aufeinanderfolgende CADC-Zyklen durchzuführen (oben, grüner und schwarzer Verlauf). Der RE wird bei Unterschreiten eines *SOC* von 90% nach 1800 s zugeschaltet. Zur Bestimmung der Leistung wird ein fiktiver *SOC* präzisiert (oben, dunkelblau), sodass eine Leistung für den RE (mittig) bestimmt werden kann. Durch den RE wird zum einen der gewünschte *SOC* von 80% eingeregelt (oben, gelb und rot) und zum anderen die vorgegebene Strecke von ca. 150 km zurückgelegt. Das repräsentative Testszenario zeigt, dass die Algorithmen zur Prädiktion konstanter RE-Leistungsverläufe funktionieren und als validiert betrachtet werden können.



**Abbildung 8.** Ergebnis der Betriebsstrategie.

## 6.5 Optimierungsoptentiale

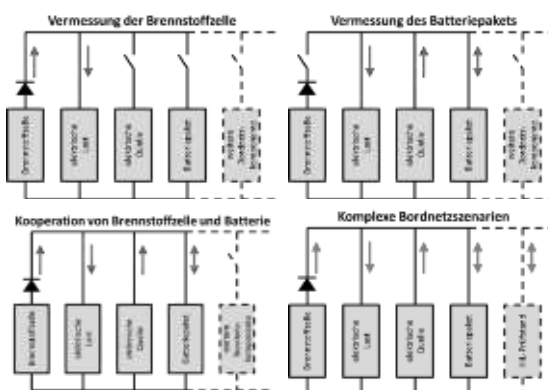
Aktuell wird die Prädiktion zu Beginn der Fahrt ausgeführt und nicht aktualisiert. Durch zyklische Aktualisierungen können Abweichungen durch Parameterunterschiede oder numerische Verfahren minimiert werden. Weiterhin findet die Leistung der Bordnetzverbraucher aktuell keine Berücksichtigung. Diese können z.B. modellbasiert (Klimatisierung), kartenbasiert (z.B. Fahrtrichtungsanzeiger) oder durch Konstanten, die während der zyklischen Aktualisierung angepasst werden, berücksichtigt werden. In **Abbildung 9** ist die Erweiterung der Betriebsstrategie dargestellt.



**Abbildung 9.** Struktur der Leistungsprädiktion des RE ergänzt um eine Funktion zur Bordnetzprädiktion.

## 7 HiL-Prüfstand

Zur Parameteridentifikation und Validierung verschiedener Regelalgorithmen wird aktuell ein modular konfigurierbarer HiL-Prüfstand eines Elektrofahrzeugs mit RE aufgebaut. Der Prüfstand beinhaltet eine PEM-BZ, eine elektrische Last zur Belastung, ein Lithium-Ionen-Batteriepaket, eine elektrische Quelle, die alle mithilfe entsprechender Schütze zugeschaltet werden können. Darüber hinaus können einfach weitere Bordnetzkomponenten hinzugefügt werden. In **Abbildung 10** sind vier Konfigurationsmöglichkeiten dargestellt.



**Abbildung 10.** Mögliche Testfälle des HiL-Prüfstands.

1. **Vermessung der BZ:** Die BZ kann mit der elektrischen Last über ihren gesamten Betriebsbereich vermessen werden. An den verschiedenen Lastpunkten sind Elektrochemische Impedanzspektroskopien vorgesehen, um das Verhalten der BZ in einem breiten Frequenzspektrum zu untersuchen.
2. **Vermessung des Batteriepakets:** Das Batteriepaket kann mithilfe einer Last und einer Quelle gezielt geladen und entladen werden um sie zu vermessen.

3. **Kooperation von BZ und Batteriepaket:** Es können Untersuchungen zur Wirkungsgradoptimierung im kombinierten Betrieb von BZ und Batterien durchgeführt werden.
4. **Komplexe Bordnetzzenarien:** Durch Anbindung weiterer Prüfstände, z.B. eines elektrischen Antriebsstrangs, können beliebige Bordnetzzenarien durchgeführt werden.

## 8 Zusammenfassung und Ausblick

In diesem Beitrag wurde modellbasiert eine Betriebsstrategie zur Leistungsverteilung in Elektrofahrzeugen mit RE basierend auf prädiktiven Daten entworfen. Durch die Betriebsstrategie kann der RE in einem hohen Wirkungsgrad betrieben werden und ein gewünschter Endladestand eingeregelt werden.

Weitere Arbeiten befassen sich mit der Optimierung und Erweiterung des Algorithmus sowie dem Aufbau eines HiL-Prüfstands zur Untersuchung der Komponenten und Algorithmen in der Praxis.

## 9 Danksagung

Dieser Beitrag wurde im Rahmen des Projekts *Zukünftige Fahrzeugtechnologien im Open Region Lab* durch das MWK Niedersachsen und die VW-Stiftung unter dem Förderkennzeichen VWZN3236 gefördert. Die Verantwortung für den Inhalt liegt bei den Autoren.



## 10 Literaturverzeichnis

- [1] Styler, A.; Nourbakhsh, I.; Sauer, A.; Rottengruber, H.: *Learned Optimal Control of a Range Extender in a Series Hybrid Vehicle*, IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, September 15 - 18, 2015.
- [2] Gausemeier, S. F.: *Ein Fahrerassistenzsystem zur prädiktiven Planung energie- und zeitoptimaler Geschwindigkeitsprofile mittels Mehrzieloptimierung*, Dissertation, Universität Paderborn, 2013.
- [3] Biermann, J.-W.; Barkow, A.; Töpler, F.: *Vorrausschauende Betriebsstrategie für Hybridfahrzeuge der Subkompaktklasse*, ATZeлектроник 4.6, S. 10–19, 2009.
- [4] Liu-Henke, X.; Duym, S.: *Modellgestützte Funktionsabsicherung des vernetzten mechatronischen Kraftfahrzeugs*. VDI-Tagung Mechatronik 2005, Wiesloch, Germany, July 01 - 02, 2005.
- [5] Borchardt, N.: *Modellierung, elektromechanische Auslegung und Validierung eines Radnabenmotors mit nutzenloser Luftspaltwicklung und hoher gravimetrischer Leistungsdichte*. Dissertation, Otto-von-Guericke-Universität Magdeburg, 2014.
- [6] Quantmeyer, F.; Liu-Henke, X.: *Modeling and Identification of Lithium-Ion Batteries for Electric Vehicles*, IEEE International Symposium on Electrodynamics and Mechatronic Systems, Zawiercie, Poland, 15-18 May 2013.
- [7] Boscaino, V.; Capponi, G.; Miceli, R.; Rizzo Galluzzo, G.; Rizzo, R.: *Comparison of models of fuel cells based on experimental data for the design of power electronic systems*. IET Renewable Power Generation 9.9, S. 660–668, 2015.
- [8] Eichlseder, H.: *Wasserstoff in der Fahrzeugtechnik: Erzeugung, Speicherung, Anwendung*. 3., überarb. Aufl.. Wiesbaden: Vieweg+Teubner Verlag, 2012.
- [9] Amphlett, J. C.; Mann, R. F.; Peppley, B. A.; Roberge, P. R.; Rodrigues, A.: *A practical PEM fuel cell model for simulating vehicle power sources*. Tenth Annual Battery Conference on Applications and Advances (BCAA 1995), Long Beach, CA, USA, Jan 10 - 13, 1995.
- [10] Scherler, S.; Quantmeyer, F.; Liu-Henke, X.: *Modellbasierte Entwicklung des elektrischen Energiemanagements für Elektrofahrzeuge mit einem Zweispannungsbordnetz*. ASIM/GI STS/GMMS Workshop 2015, Stralsund, Germany, June 18 - 19, 2015.
- [11] Fritsch, M.; Scherler, S.; Liu-Henke, X.: *Eine intelligente Fahrerassistenzfunktion zum energieoptimalen Fahrbetrieb für Elektrofahrzeuge*. ASIM/GI STS/GMMS Workshop 2016, Lippstadt, Germany, March 10 - 11, 2016.
- [12] Fritsch, M.; Scherler, S.; Liu-Henke, X.: *Prädiktives Energiemanagement für Elektrofahrzeuge*, ASIM/GI STS/GMMS Workshop 2017, Ulm, Germany, March 09 - 10, 2017.

# Simulationsgestützter Entwurf eines 48 V/ 12,5 kVA Wechselrichters für Mild-Hybrid Fahrzeuge

Christoph Faraji-Tajrishi<sup>1</sup>, Konstantin Siebke<sup>1</sup>, Markus Henke<sup>1</sup>

<sup>1</sup>TU-Braunschweig

Institut für elektrische Maschinen, Antriebe und Bahnen

Hans-Sommer-Str. 66

38106 Braunschweig

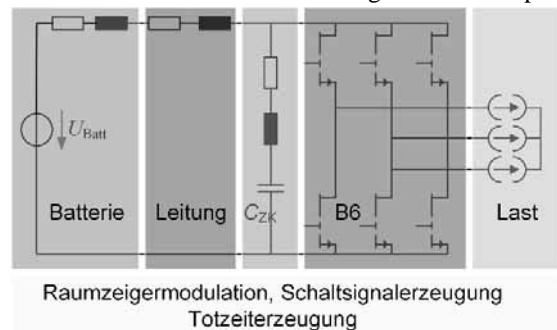
*k.siebke@tu-bs.de*

In dieser Arbeit soll ein 48 V/12,5 kVA Wechselrichter simulationsgestützt entworfen und aufgebaut werden. Zum Entwurf des Wechselrichters werden elektrische und thermische Simulationen durchgeführt. Grundlage für die elektrischen Simulationen ist ein Simulationsmodell des Wechselrichters in ANSYS Simplorer. Es dient der Verlustleistungsberechnung, Dimensionierung und Auswahl der Leistungshalbleiter. Dabei werden modernste Leistungshalbleiter mit Silizium (Si) oder Galliumnitrid (GaN) miteinander verglichen. Weiterhin soll anhand des Simulationsmodells der Zwischenkreiskondensator ausgelegt werden. Dazu werden die Spannungsverläufe am Zwischenkreis simuliert. Mittels dieser Simulation werden der auftretende Spannungsripple und die Strombelastung abgeschätzt, mit dessen Hilfe der Zwischenkreiskondensator gewählt werden kann. Zusätzlich wird ein LC-Filter des Wechselrichters simulationsgestützt bewertet. Auf Grundlage der ermittelten Verlustleistungen wird das thermische Verhalten simuliert. Zur optimalen Entwärmung der Leistungshalbleiter werden unterschiedliche Konzepte zur Kühlung durch thermische Simulationen bewertet.

## 1 Einleitung

Hybridfahrzeuge erlangen aufgrund der CO<sub>2</sub>-Grenzwerte der Europäischen Kommission immer höhere Aufmerksamkeit in der Automobilbranche. Aufgrund der hohen Preise für Hybridfahrzeuge finden diese jedoch noch nicht sehr hohen Anklang auf dem Markt. Das liegt u. A. an der Verwendung von Bordnetzen im Hochvoltbereich. Diese weisen eine Betriebsspannung von über 60 V auf, die einen aufwändigen Berührungsschutz erforderlich macht. Daraus folgt eine sehr kostenintensive Produktion. [1] 48 V Bordnetze stellen dazu eine kostengünstige Alternative dar, weil durch den geringeren Aufwand für den Berührungsschutz Hybridfahrzeuge kostengünstiger herzustellen sind. Das Bordnetz muss aufgrund von Rekuperations- und Boostfunktionen hohe Leistungen (z. B. 10 kW) liefern, welche zu großen Strömen (ca. 200 A) führen. Die Beherrschung dieser großen Ströme stellt die Leistungselektronik vor neue Herausforderungen. Gleichzeitig bietet die 48 V Spannungsebene die Möglichkeit neuartige Leistungshalbleiter mit Gallium-Nitrid (GaN) oder Silizium (Si) einzusetzen. In diesem Beitrag wird ein 48 V Wechselrichter für Mild-Hybrid Fahrzeuge mit einer Nennscheinleistung von 12,5 kVA/ 10 kW ausgelegt. Dabei soll ein maximaler

Wirkungsgrad erzielt werden. Zusätzlich wird eine möglichst hohe Leistungsdichte bzw. ein optimaler Bauraum verfolgt. Um diesen Wechselrichter optimal auszulegen werden elektrische und thermische Simulationen des Wechselrichters durchgeführt. In diesem Beitrag werden die Simulationsmodelle ausführlich beschrieben. Auf elektrischer Ebene werden die entstehenden Verluste im Wechselrichter bestimmt und unterschiedliche Leistungshalbleiter mit Si und GaN untersucht. Ebenso wird der Zwischenkreiskondensator und ein LC-Filter durch die elektrische Simulation ausgelegt. Die thermische Simulation des Wechselrichters dient der Dimensionierung des Kühlkörpers.



**Abbildung 1.** Elektrische Simulation des Wechselrichters

Als Software wird ANSYS Simplorer und Workbench verwendet. Dabei dient Simplorer der elektrischen sowie Workbench der thermischen Simulation.

## 2 Elektrische Simulation des Wechselrichters

Zunächst wird die elektrische Simulation betrachtet. Sie dient der Ermittlung der entstehenden Verluste der verschiedenen Leistungshalbleiter. Ferner soll der Zwischenkreiskondensator in dieser Simulation zur Zwischenkreisanbindung ausgelegt werden. Darüber hinaus wird ein LC-Sinusfilter simulationsgestützt bewertet.

Abbildung 1 zeigt das Simulationsmodell des Wechselrichters, welches in Simplorer übernommen und zur Simulation herangezogen wird.

Es bildet zunächst die Batterie ab. Sie wird aus einer Spannungsquelle mit einem ohmsch-induktivem Anteil modelliert. Analog dazu werden die Leitung zur Zwischenkreisanbindung sowie der Zwischenkreiskondensator mit ohmsch-induktivem Anteil modelliert. Schließlich folgt die eigentliche B6-Brücke. In Abbildung 1 sind die Leistungshalbleiter durch GaN-HEMTs (High Electron Mobility Transistor) dargestellt, wodurch die erforderlichen Dioden nicht explizit dargestellt sind. Die Last wird in Form von Stromquellen dargestellt. Zusätzlich wird dem Simulationsmodell eine State-Machine, ein Pulsweitenmodulator, ein Raumzeigermodulator sowie ein Totzeitglied zur Ansteuerung der Leistungshalbleiter hinzugefügt. Besonders die Totzeit spielt eine wichtige Rolle, da sie entscheidend zu den Verlusten beiträgt.

### 2.1 Simulation der Verlustleistung

Ziel der Untersuchungen ist es moderne GaN- und Si-Leistungshalbleiter in der Simulation miteinander zu vergleichen. Dazu werden Parallelschaltungen dieser Leistungshalbleiter untersucht. Die Gesamtverlustleistung in den Leistungshalbleitern setzt sich aus vier Anteilen zusammen: Schaltverluste der Transistoren, Schaltverluste der Dioden, Durchlassverluste der Transistoren und Durchlassverluste der Dioden.[2]

Als Simulationsmodell des Leistungshalbleiters werden statische MOSFET-Modelle verwendet. Der Kanal des MOSFETs besteht darin aus einem Schalter und einem Durchlasswiderstand  $R_{DS(on)}$ . Die Kennlinie der Diode wird auf eine Forwardspannung  $V_F$  und einen Forwardwiderstand  $R_F$  reduziert (Abbildung 2). Diese

Parameter lassen sich aus dem Datenblatt des Leistungshalbleiters ermitteln. Da diese Parameter temperaturabhängig sind, werden sie für eine Worst-Case-Abschätzung für eine Junction-Temperatur von  $150^\circ\text{C}$  gewählt. Damit lassen sich in der Simulation nur Durchlassverluste darstellen. Die Schaltverluste lassen sich in separaten Simulationsmodellen parallel mitsimulieren oder im Anschluss an die Simulation analytisch berechnen.

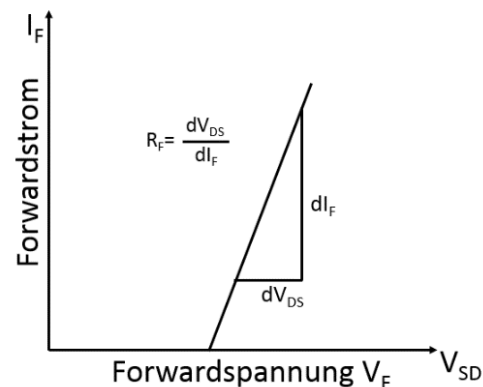
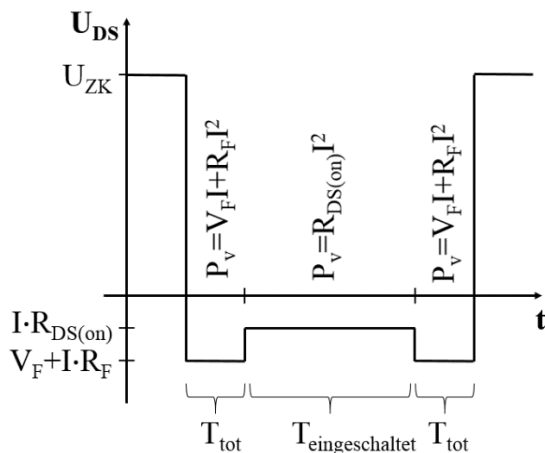


Abbildung 2. Vereinfachte Diodenkennlinie

Aufgrund der, im Vergleich zum Kanal des MOSFETs, schlechten Durchlasseigenschaften der Diode, werden die Leistungshalbleiter einer Halbbrücke komplementär und mit einer Totzeit geschaltet. Lediglich während der Totzeit muss der Laststrom auf die Dioden kommutieren. Abbildung 3 zeigt den dabei entstehenden Spannungsverlauf über dem MOSFET. Leitet der komplementäre Leistungshalbleiter, liegt an dem Leistungshalbleiter die Zwischenkreisspannung an. Während der Totzeit fließt der Laststrom über die Diode und die Spannung beträgt  $-(V_F + R_F I)$ . Wird der MOSFET nun eingeschaltet wird angenommen, dass der Laststrom vollständig auf den Kanal des MOSFETs kommutiert und die Spannung  $R_{DS(on)} I$  beträgt. Daraus ergeben sich zu den einzelnen Zeitabschnitten verschiedene Durchlassverluste [3].



**Abbildung 3.** Durchlassverluste während der Rückwärtsleitung des MOSFETs [3]

Die Durchlassverluste der Diode lassen sich daher analytisch nur schwer abschätzen, sodass die Simulation an dieser Stelle sinnvoll ist.

**Tabelle 1** zeigt die Simulationsergebnisse für die untersuchten Leistungshalbleiter. Für eine ausreichende Stromtragfähigkeit des Leistungshalbleiters müssen die Leistungshalbleiter parallel geschaltet werden. Durch die geringere Stromtragfähigkeit des GaN-Leistungshalbleiters gegenüber den Si-Leistungshalbleitern müssen für den GaN-Wechselrichter mehr Leistungshalbleiter parallel geschaltet werden. Der mit zwei parallelen Si-Leistungshalbleitern bestückte Wechselrichter erreicht eine Verlustleistung von 32 W/Parallelschaltung und weist damit die geringste Verlustleistung auf.

Si MOSFET 1	48,13 W
Si MOSFET 2	61,45 W
Si MOSFET 2 - 2 parallel	31,99 W
GaN-HEMT	404,41 W
GaN-HEMT - 2 parallel	206,89 W
GaN-HEMT - 3 parallel	139,38 W
GaN-HEMT - 4 parallel	105,2 W
GaN-HEMT - 5 parallel	84,57 W

**Tabelle 1.** Gesamtverlustleistung des Wechselrichters für unterschiedliche Topologien

Die Simulation wurde exemplarisch für eine Schaltfrequenz von 8 kHz angewendet. Aufgrund der großen

Durchlasswiderstände der GaN-HEMTs sind die Verluste in den GaN-Varianten deutlich größer als in den Si-Varianten. Der Vorteil der geringen Schaltverluste der GaN-Leistungshalbleiter kann in dieser Anwendung nicht ausgenutzt werden.

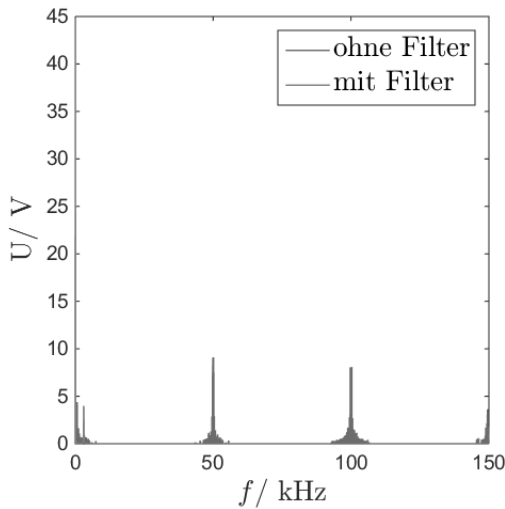
## 2.2 Simulationsgestützte Bestimmung der Zwischenkreiskapazität

Die benötigte Zwischenkreiskapazität wird ebenfalls durch eine Simulation abgeschätzt. Für eine korrekte Abschätzung ist die Modellbildung der Zuleitung zum Wechselrichter, der Batterie und der parasitären Eigenschaften des Zwischenkreiskondensators wichtig, da diese Eigenschaften wesentlichen Einfluss auf die Spannungswelligkeit am Zwischenkreiskondensator haben. Durch die Simulation in Simplorer wird der Betriebspunkt mit maximaler Spannungswelligkeit bestimmt. Die Spannungswelligkeit wird anschließend durch Skalierung der Zwischenkreiskapazität auf einen vorgegebenen Wert von 1 % (0,5 V) der Zwischenkreisspannung begrenzt.

Für den vorliegenden Wechselrichter und auf Grundlage der Modellbildung ergibt sich die Zwischenkreiskapazität zu 600  $\mu$ F.

## 2.3 Simulationsgestützte Bewertung eines LC-Sinusfilters

Die gewählten Leistungshalbleiter lassen hohe Schaltfrequenzen (50 kHz) zu. Um die hohen Schaltfrequenzen ausnutzen zu können, wird ein mögliches Sinusfilter, welches in den Wechselrichter integriert werden kann, untersucht. Der Bauraum der passiven Komponenten des Sinusfilters wird mit steigender Schaltfrequenz kleiner. Als Filtertopologie wird in diesem Beitrag ein einfaches LC-Sinusfilter betrachtet. Mittels Simulation in Simplorer kann eine Aussage über die Güte des Filters getroffen werden. Bewertungsgrundlage ist das Frequenzspektrum mit und ohne Filter. Dieses kann über die FFT-Funktion in Simplorer dargestellt werden. Das Ergebnis der FFT mit und ohne LC-Filter ist in Abbildung 4 gezeigt. Der Verlauf ohne Filter zeigt deutlich, dass eine Vielzahl von Frequenzen oberhalb der Grundfrequenz auftreten, insbesondere die Schaltfrequenz ist zu erkennen. Das LC-Sinusfilter wird abhängig von der Schaltfrequenz ausgelegt. Exemplarisch wird dieser für eine Frequenz von 50 kHz ausgelegt.



**Abbildung 4.** Ausgangsseitiges Frequenzspektrum des Wechselrichters mit LC-Sinusfilter

Es ist deutlich zu erkennen, dass annähernd alle Frequenzen über der Grundfrequenz durch das Filter eliminiert werden. Die Werte der ausgelegten Bauelemente des LC-Sinusfilters entsprechen für eine Schaltfrequenz von 50 kHz

$$L = 24 \mu H \text{ und } C = 42 \mu F$$

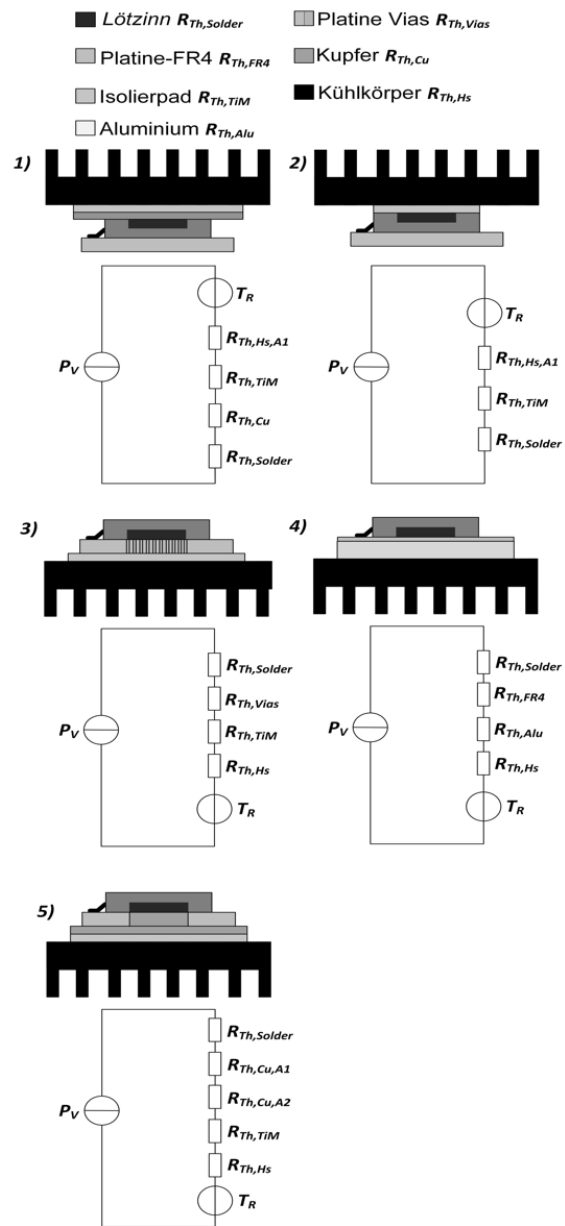
### 3 Thermische Simulation des Wechselrichters

Aufgrund der hohen Ströme in den Schaltern, darf das thermische Verhalten nicht unberücksichtigt bleiben. Bei modernen Leistungshalbleitern mit nur kleinen Kühlflächen, begrenzt die geringe Fähigkeit, die Wärme abzuführen, die Leistung der Applikation [4]. Grundlage der Analyse ist die Verlustleistung (Tabelle 1). Um eine optimale Wärmeabführung von den Transistoren zu gewährleisten, werden unterschiedliche Konfigurationen zur Kühlkörperanbindung untersucht.

#### 3.1 Entwärmungsoptionen

Abbildung 5 zeigt die untersuchten Konfigurationen mit den entsprechenden thermischen Ersatzschaltbildern. Ein Unterschied der Si- und GaN-basierten Leistungshalbleiter ist die Platzierung der Kühlflächen (cooling-area). Der HEMT (1-2) wird oberseitig, der MOSFET (3-5) hingegen unterseitig gekühlt. Die einfachste Lösung der Kühlung des HEMT zeigt Konfiguration 2. Dabei wird die Wärme vom Gehäuse des Leistungshalbleiters direkt über das Isolierpad zum Kühlkörper abgeführt. Um eine bessere Wärmeabfuhr

des GaN-Leistungshalbleiters zu ermöglichen, wird ein Heatspreader eingefügt. Der Heatspreader ist ein Kupferblech, welches für eine Vergrößerung der Kühlfläche sorgt. Dadurch verringert sich der thermische Widerstand der Isolationsschicht (Konfiguration 1). Da der MOSFET auf der Unterseite gekühlt wird, muss die Wärme durch das Platinenmaterial (FR4) abgeführt werden. Da die thermische Leitfähigkeit von FR4 sehr gering ist (1,6 W/mK) werden hier drei mögliche Lösungsansätze zur Optimierung der Kühlanbindung vorgestellt.



**Abbildung 5.** Konfigurationsmöglichkeiten zur Kühlkörperanbindung

Die Konfiguration 3 sieht mit Kupfer gefüllte Durchkontaktierungen (thermische Vias) in der Platine vor. Konfiguration 4 besteht aus einer Aluminiumplatine mit einer sehr dünnen Schicht FR4 zur Isolation und als Träger für die Leiterbahnen. Konfiguration 5 sieht ein Fräsloch unter dem Leistungshalbleiter vor, das mit Kupfer aufgefüllt wird. Zusätzlich wird dazu bei Konfiguration 5 ein Heatspreader eingesetzt.

### 3.2 Bewertung der Entwärmungsoptionen

Zu den vorgeschlagenen Konfigurationen werden eindimensionale Temperaturberechnungen durch elektrische Ersatzschaltbilder vorgenommen. Da die laterale Ausbreitung des Wärmestroms durch die eindimensionale Berechnung schlecht erfasst werden kann, werden zusätzlich mit Hilfe von ANSYS Workbench thermisch, stationäre Simulationen durchgeführt. Ziel ist es die Temperaturdifferenz zwischen Kühlkörperoberfläche und Gehäuse des Leistungshalbleiters zu bestimmen und so die Leistungsfähigkeit der Konfigurationen bewerten zu können. Die Temperatur der Kühlkörperoberfläche wird zu 20 °C gewählt. Die im Leistungshalbleiter anfallenden Verluste werden als Wärmestrom mit 60 W modelliert.

Für die eindimensionalen Berechnungen werden alle Komponenten der Konfigurationen durch ihre thermischen Widerstände dargestellt. Der thermische Widerstand  $R_{th}$  berechnet sich zu

$$R_{th} = \frac{d}{A \lambda} \quad (1)$$

$A$ : Vom Wärmestrom durchsetzte Fläche

$d$ : Stärke des Materials

$\lambda$ : thermische Leitfähigkeit

Übergangswiderstände zwischen den Komponenten sowie der Leistungshalbleiter selbst werden hier nicht berücksichtigt. Da in diesem Beitrag nur die Temperaturdifferenz zwischen Kühlkörperoberfläche und Gehäuse des Leistungshalbleiters berechnet wird, muss der Wärmestrom noch mit dem thermischen Widerstand zwischen Junction und Gehäuse des Leistungshalbleiters ( $R_{th,JC}$ ) multipliziert und zur Gehäusetemperatur addiert werden um die Junction-Temperatur zu erhalten.

Konfiguration	Berechnung	Simulation
1 GaN mit Heatspreader	40,2 °C	54,1 °C
2 GaN ohne Heatspreader	73,8 °C	84 °C

3 Si mit Vias	24,5 °C	53,9 °C
4 Si mit Aluplatine	35,2 °C	50,4 °C
5 Si mit Kupferfüllung im PCB	24,8 °C	26 °C

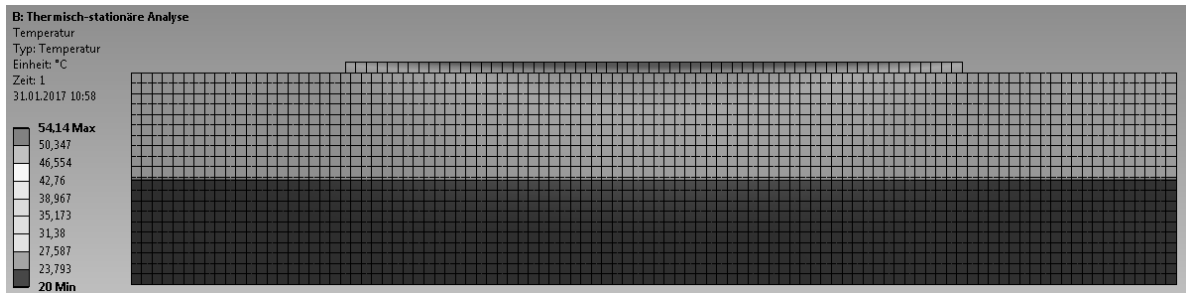
**Table 2.** Gehäusetemperatur des Leistungshalbleiters

Es fällt auf, dass die berechneten Temperaturen bei den Konfigurationen 1 bis 4 deutlich geringer sind als die simulierten Temperaturen. Bei Konfiguration 5 hingegen beträgt die Abweichung zwischen berechneten und simulierten Temperaturen lediglich 1,2 °C. Damit stimmen die Ergebnisse bei Konfiguration 5 gut überein. Bei den Berechnungen der Temperaturen und der thermischen Widerstände wird davon ausgegangen, dass sich der mittig in die Konfiguration eingeleitete Wärmestrom ideal gleichmäßig lateral in der Struktur ausbreitet und die gesamte Fläche  $A$  zu dem thermischen Widerstand beiträgt. In der Praxis wird es zu einer effektiven Querschnittsreduzierung durch nicht vollständige laterale Ausbreitung des Wärmestroms kommen, wodurch sich die thermischen Widerstände der einzelnen Komponenten der Konfigurationen erhöhen. In der Simulation kann dieser Effekt berücksichtigt werden, wodurch die simulierten Temperaturen größer sind als die berechneten.

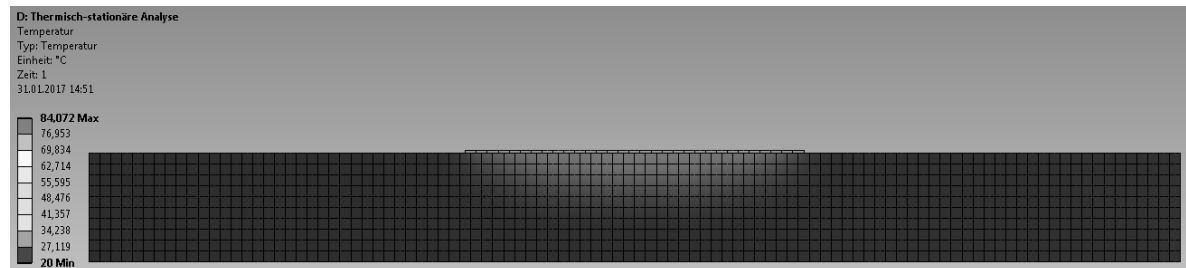
In Konfiguration 5 befindet sich unter dem Leistungshalbleiter in der Platine eine Kupferfüllung in der Größe des Leistungshalbleiters. Der thermische Widerstand der Platine um die Kupferfüllung kann aufgrund der geringen thermischen Leitfähigkeit des Platinenmaterials (FR4) vernachlässigt werden. Die Vernachlässigung der nicht gleichmäßigen lateralen Verteilung des Wärmestroms fällt bei der Konfiguration 5 nicht ins Gewicht, womit berechnete und simulierte Temperaturen gut übereinstimmen.

Die Abbildung 6 zeigt die Konfigurationen im Querschnitt mit den Temperaturen aus den durchgeführten Simulationen und macht die Wirkungsweisen der Konfigurationen deutlich. Der Wärmestrom von 60 W wird jeweils von oben in die Konfiguration eingeleitet, die Temperatur an der Unterseite (Kühlkörper) ist auf 20 °C fixiert.

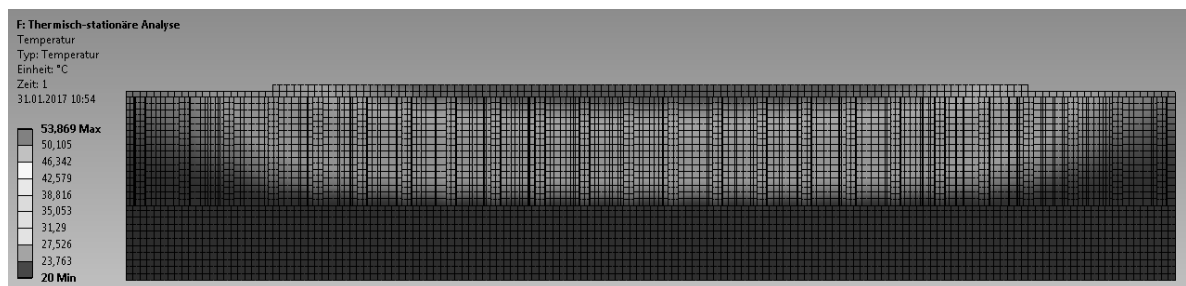
Für den GaN-Leistungshalbleiter mit Kühlfläche auf der Oberseite eignet sich die Konfiguration 1 (Heatspreader). Für den Si-Leistungshalbleiter wäre die Konfiguration 5 (Kupferfüllung) optimal. Der hohe



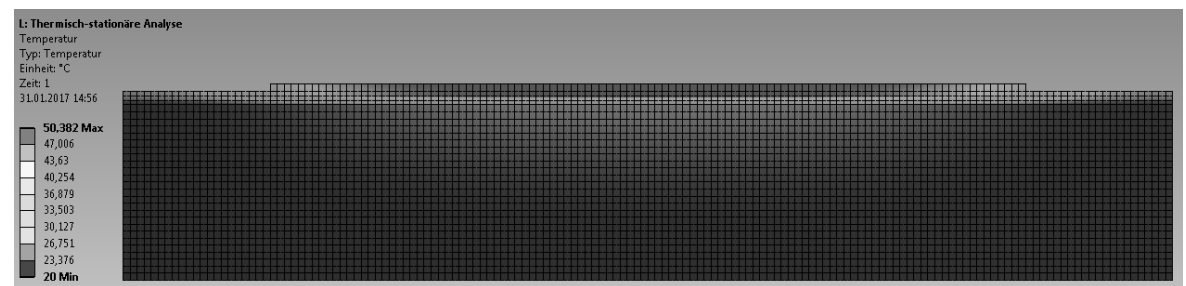
**Abbildung 6a.** Konfiguration 1, GaN Leistungshalbleiter mit Heatspreader



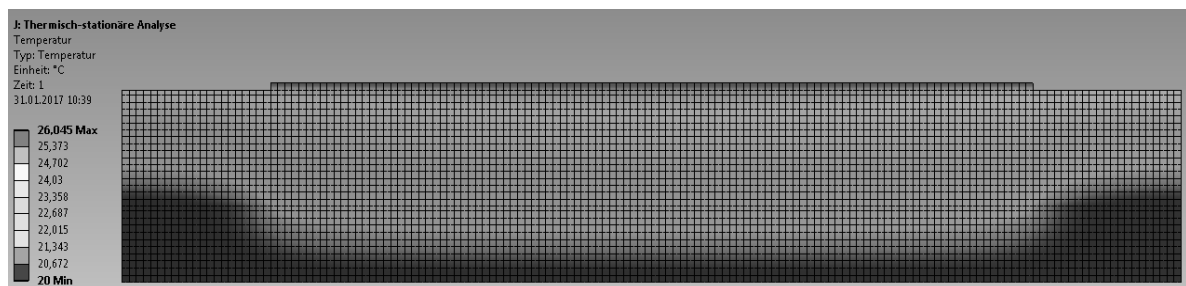
**Abbildung 6b.** Konfiguration 2, GaN Leistungshalbleiter ohne Heatspreader



**Abbildung 6c.** Konfiguration 3, Si Leistungshalbleiter mit thermischen Vias



**Abbildung 6d.** Konfiguration 4, Si Leistungshalbleiter mit Aluminiumplatte



**Abbildung 6e.** Konfiguration 5, Si Leistungshalbleiter mit Kupferfüllung im PCB

Aufwand für den mechanischen Aufbau und das Layout kann allerdings den Einsatz der Konfiguration 3 oder 4 rechtfertigen.

## 4 Zusammenfassung

In dieser Arbeit wird ein 12,5 kVA Wechselrichter für 48 V-Bordnetze mit Hilfe von elektrischen und thermischen Simulationen ausgelegt. Auf elektrischer Ebene spielt die Simulation eine entscheidende Rolle bei der Wahl der Leistungshalbleiter (Si vs. GaN). Die optimale Wahl ist auf Grundlage dieser Simulation eine Parallelschaltung zweier Si-MOSFETs. Durch Modellierung der Zwischenkreisanbindung wird die benötigte Zwischenkreiskapazität durch die Simulation abgeschätzt. Dazu wird der Betriebspunkt mit maximaler Zwischenkreisbelastung durch die Simulation ermittelt. Es ergibt sich eine Zwischenkreiskapazität von 600  $\mu\text{F}$ . Auch die Bewertung eines LC-Sinusfilters wird mittels elektrischer Simulation vorgenommen. Dazu wird das Frequenzspektrum mit und ohne Filter gegenübergestellt. Das LC-Sinusfilter wird für eine Schaltfrequenz von 50 kHz ausgelegt, um den positiven Einfluss der hohen Schaltfrequenz deutlich zu machen. Die passiven Bauelemente fallen dabei besonders klein aus.

Für die thermische Simulation werden 5 verschiedene Kühlkonfigurationen betrachtet und bewertet. Da die laterale Ausbreitung des Wärmestroms in eindimensionalen Berechnungen nur schwer abgeschätzt werden kann, werden thermische Simulationen herangezogen. Für den GaN-Leistungshalbleiter ist die Konfiguration mit Heatspreader sehr geeignet. Für den Si-Leistungshalbleiter liefert die Konfiguration mit Kupferfüllung in der Platine die geringste Gehäusetemperatur. Aufgrund der schweren Umsetzung dieser, wird für den physischen Aufbau des Wechselrichters die Konfiguration mit Vias in der Platine angewendet, die zu einer ähnlichen Gehäusetemperatur führt.

## 5 Literatur

- [1] J. Bilo, H.-G. Burghoff, *48-V-Bordnetz – Schlüsseltechnologie auf dem Weg der Elektromobilität*. Zentralverband Elektrotechnik- und Elektronikindustrie, Frankfurt a. M., Dez 2015.
- [2] G. Tareilius, *Der Auxiliary Resonant Commutated Pole Inverter im Umfeld schaltverlustreduzierter IGBT-Pulswechselrichter* Dissertation, TU-Braunschweig, 2002.
- [3] Christoph Faraji-Tajrishi, *Entwurf und Aufbau eines 10 kW Wechselrichters mit 48V Bordnetz für Hybridfahrzeuge*, Masterarbeit, TU-Braunschweig, 2017
- [4] K. Siebke, T. Schobre, N. Langmaack, R. Mallwitz, *GaN based DC-DC Power Conversion-Experiences from Earth to Space*, ESA/ECSAT, 8th Wide Bandgap Semiconductor and Components Workshop, Harwell, UK, 2016

# Dynamische Fehlerinjektion für die Systemsimulation mittels SystemC – effizient, flexibel, automatisierbar

Thomas Markwirth, Stephan Gerth

Fraunhofer IIS/ EAS, Zeunerstraße 38, 01069 Dresden

*Thomas.Markwirth@eas.iis.fraunhofer.de, Stephan.Gerth@eas.iis.fraunhofer.de*

Vorgestellt wird ein Ansatz zur dynamischen Fehlerinjektion in SystemC/ SystemC AMS -Systemmodelle als eine Möglichkeit zur Erfüllung der im Standard ISO 26262 definierten Vorgaben. Hierzu sind weder fehlerhaftes Verhalten noch spezielle Fehlerstrukturen in die Modelle des zu untersuchende Designs einzufügen. Vielmehr wird eine dynamische Injektion direkt zur Simulationslaufzeit durch Auftrennen von Verbindungen und Einbringen von Fehlermodellen realisiert. Daraus ergibt sich eine einfache Möglichkeit, Fehlerinjektionen auch in bereits existente Systemmodelle nachträglich realisieren zu können. Der Ansatz ist außerdem weitgehend automatisierbar und anwendbar in einer Hardware-In-the-Loop-Verifikation.

## 1 Einleitung

Die steigende Komplexität elektronischer Systeme, besonders der Automobilindustrie, ist gekennzeichnet durch Wechselwirkungen digitaler und analoger Hardware mit eingebundener Software, nichtelektrischen Systemkomponenten, sowie der Umgebung und deren physikalischen Eigenschaften. Der Nachweis der Funktionssicherheit solcher Systeme ist dabei zunehmend schwierig und es genügt nicht mehr, nur die Funktion für den Gutfall (nominal case) nachzuweisen. Ebenso ist das Systemverhalten für den Fall fehlerhafter Komponenten und Funktionen zu untersuchen. Damit wird nicht nur die Erfüllung der funktionalen Anforderungen, sondern auch die funktionale Sicherheit eines Systems sichergestellt. Die verwendeten Design- und Verifikationswerkzeuge müssen somit auch die Möglichkeiten zum Entwurf und zum Nachweis funktional richtiger, robuster und sicherer Systeme beinhalten. Besonders für die Automobilindustrie gibt es hierzu bereits Vorgaben, welche durch den ISO 26262 Standard definiert sind. Darin ist u.a. der Nachweis der funktionalen Sicherheit unter dem Einfluss von Fehlern zu untersuchen und zu klassifizieren. Der nachfolgend beschriebene Ansatz kann für Systeme angewendet werden, um diese Vorgaben zu erfüllen.

Simulationen basierend auf der Anwendung der Sprache SystemC/ SystemC AMS sind geeignet, das Nominalverhalten auch sehr komplexer Systeme effizient zu validieren. Neben einer hohen Performance bei ausreichend hoher Genauigkeit bietet diese Sprache die Anwendung von Softwareentwurfsaspekten im Entwurfs- und Verifikationsprozess. Zusätzlich bietet

der Ansatz die Möglichkeit einer Anwendung in Hardware-in-the-Loop (HiL) –Simulationen und damit zur Validierung in Verbindung mit realer Hardware.

Dieses Paper beschreibt ein Verfahren zur Injektion von Fehlern in Nominal-Verhaltensbeschreibungen, ohne die Modelle oder Netzlisten ändern zu müssen. Es besteht für eine breite Klasse an Fehlern keine Notwendigkeit, nominale SystemC oder SystemC AMS-Beschreibungen zu modifizieren, um Fehler einzubringen. Dies ist einer der Hauptvorteile dieses Ansatzes. Fehlerhaftes Verhalten wird hingegen auf der Testebene in Form von Fehlerszenarien beschrieben.

Bisherige Ansätze zur Injektion von Fehlern in Systembeschreibungen realisieren i.d.R. Fehlerstrukturen direkt in den jeweiligen Modellen, welche durch bestimmte Ereignisse aktiviert werden. Das hat den entscheidenden Nachteil keiner klaren Trennung zwischen Beschreibung des nominalen und fehlerhaften Verhaltens. Dadurch ergibt sich das Risiko von Inkonsistenzen zwischen beiden Beschreibungen.

Vorgestellt wird im Folgenden ein Bibliotheksansatz für eine dynamische Fehlerinjektion in SystemC/ SystemC AMS Beschreibungen. Ein generischer Ansatz bietet dabei die Möglichkeit einer breiten Anwendbarkeit auf verschiedenste Modelle und Systeme. Es werden verschiedene Modelldomänen und sog. MoC (models of computation) unterstützt. Dadurch, dass das zu testende Design (DUT) nur das Nominalverhalten abbilden muss, ergibt sich eine klare Trennung zwischen funktionalem Modell und Testbeschreibung, zu der auch die Injektion von Fehlern zählt. Diese Separation erfolgt dabei sowohl hinsichtlich des Daten-

managements (Fehlerinjektion wird in Form von Fehleriszenen im Rahmen der Testumgebung beschrieben), als auch bei der Testausführung (Fehlerinjektion wird dynamisch zur Laufzeit eines Testfalls realisiert). Damit ist eine komplett separate Beschreibung von funktionalem Modell (DUT), der Testumgebung und der zu berücksichtigenden Fehler realisierbar. Ein anderer wichtiger Aspekt ist die einfache Anwendbarkeit des Ansatzes für Untersuchungen.

## 2 Realisierung

Die hier vorgestellte Fehlerinjektionsbibliothek nutzt einen hierarchischen Ansatz. Damit ist die Beschreibungsmöglichkeit von Fehlern auf unterschiedlichen Abstraktionsebenen gemeint. Beispielsweise existieren Injektionsstrukturen auf der untersten Abstraktionsebene, um Verbindungen im DUT aufzutrennen und Fehlermodelle einfügen zu können. Diese „Umverdrahtung“ erfolgt dabei dynamisch unmittelbar bevor eine Testfall-Simulation startet, ohne die DUT-Modellbeschreibung hierzu ändern zu müssen.

### 2.1 Reconnecting und Low-Level-Strukturen

Um eine dynamische Injektion von Fehlermodellen/ -strukturen realisieren zu können, muss eine Möglichkeit bestehen, Verbindungen zwischen einzelnen Modulen zu lösen. Hierbei kann ein Vorteil von SystemC genutzt werden, der es erlaubt, Einfluß auf den Ablauf und die einzelnen Phasen einer Simulation über sogenannten Callback-Funktionen zu nehmen. Diese werden vom Simulatorkern jeweils beim Durchlauf der entsprechenden Phase aufgerufen. Beispielsweise wird die Callback-Funktion `before_end_of_elaboration` aufgerufen, wenn die Modulstruktur bereits aufgebaut, das „Verdrahten“ (Port-Binding) jedoch noch nicht abgeschlossen ist. Zu diesem Zeitpunkt ist es noch möglich, Verbindungen wieder zu lösen und zusätzliche Strukturen einzufügen.

Im Prinzip ist dieser Ansatz für alle Berechnungsmodelle (MoCs) von SystemC / SystemC AMS anwendbar, im Detail werden je nach MoC unterschiedliche Low-Level-Strukturen eingefügt. Eine Low-Level-Struktur bildet dabei die unterste Abstraktionsebene von Fehlerinjektionsmodellen in der Fehlerinjektionsbibliothek. Im einfachsten Fall, z.B. für nichtkonservative Signale/ Ports kann dies ein einfacher Multiplexer sein. Dieser bietet später die Möglichkeit, während der anschließenden Simulationsphase zwischen dem originalen und einem alternativen Signalreiber, z.B. einer neuen Quelle umzuschalten. Damit sind bereits die Vo-

raussetzungen gegeben, um etwa Haftfehler nachzubilden. Für lineare Netzwerke, als ein anderes Berechnungsmodell in SystemC AMS, werden schaltbare Widerstände und optionale Spannungsquellen als Low-Level-Struktur verwendet.

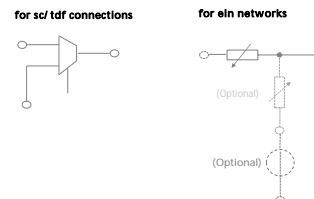


Abbildung 1: Low-Level-Strukturen zur Fehlerinjektion

Wichtig ist in jedem Falle, dass sowohl MoC als auch Signaltyp des Zielports mit den der originalen und der alternativen Quellen übereinstimmen. Abbildung 1 zeigt Beispiele für Low-Level-Strukturen für unterschiedliche MoCs.

### 2.2 Fehlermodelle

Die Fehlerinjektionsbibliothek enthält eine Reihe von klassischen Fehlermodellen. Diese nutzen die im vorhergehenden Abschnitt vorgestellten Low-Level-Strukturen. Diese Fehlermodelle stellen dabei die nächst höhere Abstraktionsebene in der Bibliothek dar. Vordefinierte Fehlermodelle existieren für Haftfehler (stuck-at value or signal), Übersprechen (Crosstalk), Unterbrechung/ Kurzschluß, Verzögerung, Glitch und Überbrückung. Diese Auswahl wird permanent erweitert. Abbildung 2 zeigt Beispiele solcher Fehlermodelle. Die vordefinierten Fehlermodelle sind als sogenannte Template-Klassen realisiert, wodurch sie für unterschiedliche Signaltypen anwendbar sind. Dieser Typ ist in Form eines Templatearguments bei der Instanziierung zu übergeben. Das zu Grunde liegende Berechnungsmodell (MoC) des jeweiligen Signals wird hingegen automatisch erkannt. Dadurch wird die zugehörige Low-Level-Struktur ebenfalls automatisch ausgewählt – ein Eingreifen des Nutzers ist nicht notwendig. Um ein vordefiniertes Fehlermodell anzuwenden, muss dieses lediglich instanziiert werden, wobei Argumente zu übergeben sind. Im einfachsten Fall ist dies ein String, welcher den Zielport innerhalb der Modul-Hierarchie repräsentiert, also den Ort wo das Fehlermodell eingefügt werden soll. Ein zweiter String ist notwendig bei Fehlermodellen, die einen alternativen Quellport benötigen (z.B. bei Crosstalk). Die Strings der Ziel- bzw. Quellports werden verwendet, um automatisch die Modulkategorie nach den entsprechenden

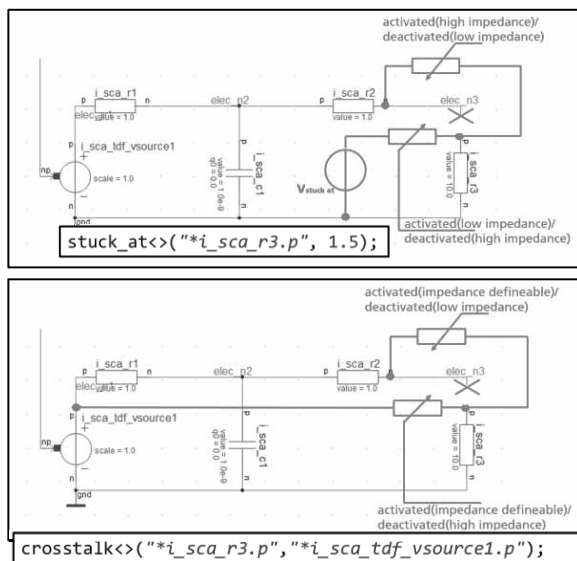


Abbildung 3: Fehlerinjektionsbeispiele für Haftfehler und Übersprechen in einem elektrischen Netzwerk

Objekten zu durchsuchen und bei Erfolg an diesen Stellen die vorhandenen Verbindungen aufzutrennen und das entsprechende Fehlermodell einzufügen. Je nach Fehlermodell können optional weitere Argumente übergeben werden. Beispielsweise erlaubt das Fehlermodell Crosstalk für digitale bzw. nichtkonservative Signale die Definition der Art der logischen Verknüpfung von Original- und Alternativtreiber. Aktuell verfügbar sind die Kombinationen WAND, WOR, ADD, SUB, REPLACE und USER. Letztere schließt die Einbindung einer externen Anwenderfunktion ein.

### 2.3 Szenarien

Szenarien bilden die höchste Ebene der Fehlerinjektionsabstraktion. Ein solches Szenario instanziiert dazu ein oder mehrere Fehlermodelle und ermöglicht deren Konfiguration. Die Bibliothek beinhaltet ein Szenario-Template, welches die Anwendung der vordefinierten Fehlermodelle ermöglicht. Bei der Instanziierung eines solchen Templates sind verschiedene Argumente zu übergeben, um eine grundsätzliche Initialisierung zu erreichen. Zwingend müssen Ziel- bzw. Quellports und der Typ des zu verwendenden Fehlermodells definiert werden. Erstere erfolgt wieder über Strings der Modulhierarchie. Hierbei werden Vektoren erwartet, was den Vorteil hat, beliebig viele gleiche Fehlermodelle mit einem Aufruf injizieren zu können.

Welches Fehlermodell injiziert wird, wird über ein weiteres Argument definiert. Zusätzliche Konstruktor-Argumente können notwendig sein, beispielsweise die Definition der logischen Verknüpfung in Crosstalk-

Fehlermodellen. Ein Fehlerszenario beinhaltet Funktionen zur dynamischen Aktivierung/ Deaktivierung des/ der entsprechenden Fehlermodelle. Alternativ ist es aber auch möglich, eine periodische Aktivierung/ Deaktivierung zu konfigurieren. Des Weiteren können Konfigurationen spezifisch zum gewählten Fehlermodell realisiert werden, beispielsweise Parameter- oder Signalsweeps. Abbildung 3 zeigt z.B. das Ergebnis eines Parametersweeps für verschiedene Verstärkungsfehler in einer PLL-Regelschleife.

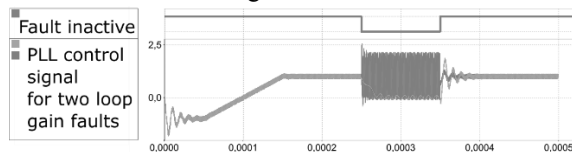


Abbildung 2: Ergebnisse für einen Fehlerparametersweep

### 2.4 Anwendung der Bibliothek

Die in den vorherigen Abschnitten beschriebenen Abstraktionsebenen der Fehlerinjektion befinden sich allesamt in einer neu erstellten SystemC/ SystemC AMS-Fehlerinjektionsbibliothek. Damit stehen dem Anwender der grundsätzliche Ansatz, verschiedene vordefinierte Fehlermodelle, sowie weitreichende Konfigurationsmöglichkeiten zur Verfügung. Damit können, komplexe Fehlerinjektionsszenarien auf sehr einfache und effiziente Weise in eigene Untersuchungen integriert werden. Die Aufgabe des Anwenders beschränkt sich darauf festzulegen, an welcher Stelle im Design welches Fehlermodell einzufügen ist und wie dieses zu aktivieren oder deaktivieren ist. Eventuell sind noch einfache Konfigurationen, wie Fehlerdauer oder periodisches Auftreten vorzunehmen. Das folgende Bild zeigt die Erstellung eines solchen Fehlerszenarios.

```
stimuli_fault_injection_tutorial_3(
    sc_core::sc_module_name nm, params pa = params() ) : p(pa){
    // Definition der Ziel- und optional Quellports zur Fehlerinjektion
    port.push_back("**i_delay_tdf1.tdf_i*");
    port_b.push_back("**");

    // Erstelle 1. Szenario: Stuck-At: Haftwert gesetzt auf 0.0 Volt
    stuck_at_fault1 = new fault_scenario_template<double> (
        std::make_pair(port, port_b),
        fault_injection_base::FAULT_STUCK_AT);
    stuck_at_fault1->fehler_mod[0]->set_voltage(0.0);
    std::vector<std::string> port, port_b;
    fault_scenario_template<double> * stuck_at_fault1;
```

Für die Anwendung der Bibliothekselemente in Testfällen (test cases) hat sich die Erstellung eines eigenen SystemC Moduls bewährt. Dieses beinhaltet lediglich die Instanziierung und Konfiguration der Fehlerszenarien und stellt eine Aktivierungsfunktion zur Verfügung. Optional ist das Tracing interner Fehlersignale. Ein solches als Stimuli-Fehlerinjektionsmodul be-

gezeichnetes Modul kann auch automatisch erstellt werden, worauf im Ausblick noch eingegangen wird.

## 2.5 Fehleraktivierung während eines Testfalls

Eine Fehlerinjektion erfolgt in der Regel bezogen auf einen speziellen Testfall. Um ein mittels o.g. Ansatz erstelltes Fehlerinjektionsmodul in einem solchen Testfall anzuwenden, sind lediglich zwei Schritte durchzuführen: 1) Instanziierung des Fehlerinjektionsmoduls 2) Aktivierung über Aufruf der Aktivierungsfunktion. Diese kann beispielsweise zeit- oder ereignisabhängig erfolgen. Letztlich genügen zwei Zeilen Code, um die Fehlerinjektion in einem Testfall anzuwenden.

```

////////////////////////////////////
// instantiation of a fault injection stimulus module
////////////////////////////////////
fault_stim = new stimuli_fault_injection_ikeba("fault_stim");
::
fault_stim->activate_fault_injection(true); //activation

```

## 2.6 Zielportauswahl durch Definition von Suchkriterien

Neben der genauen Vordefinition der Zielorte, an denen Fehler eingefügt werden sollen, können alternativ auch Suchkriterien definiert werden. Anschließend wird die Modulhierarchie nach potentiellen Zielen durchsucht, die diese Kriterien erfüllen. Solche Kriterien können beispielsweise MoCs, Porttypen oder Instanznamensfragmente sein. Der folgende Quellcodeausschnitt zeigt exemplarisch das Durchsuchen der Modulhierarchie nach Objekten, die das Namensfragment „mosi\_i“ beinhalten. Handelt es sich des Weiteren dabei um sc-Eingänge vom Typ bool, dann werden diese zur Injektion eines Fehlermodells verwendet.

```

stimuli_fault_injection_fehler_regr(sc_core::sc_module_name nm, params
pa = params() ) : p(pa) {
////////////////////////////////////
Suche alle Objekte mit mosi_i String und gleichzeitig vom Typ
sc_in<bool>
////////////////////////////////////
vector<sc_object*> sc_obj = get_matching_objects("mosi_i");
cout << "...NUMBER of matching objects: " << sc_obj.size() << endl;

for (unsigned int h = 0; h < sc_obj.size(); ++h)
{
    sc_core::sc_in<bool>* sc_inst = dynamic_cast<sc_core::sc_in<bool>*>
(sc_obj.at(h));
    if (sc_inst != NULL)
    {
        port.push_back(sc_obj.at(h)->name());
        cout << "...picking matching object, which is a SC input and"
        << " additionally of type boolean. Matching object: "
        << h << " " << sc_obj.at(h)->name() << " "
        << port.size() << endl;
    }
}
cout << "...sum up: System includes " << port.size() << " potential
failure injection target objects: " << endl;

```

Die zugehörige Kommandozeilenausgabe während der Simulation:

```

Searching for all objects in the DUT system, which include 'mosi' in
their name/ path....
...NUMBER of matching objects: 12...
...picking matching object, which is a SC input and additionally of
type boolean. Matching object: 2 dut.i_spi_channel1.i_spi_slavel.mosi 1
...picking matching object, which is a SC input and additionally of
type boolean. Matching object: 5 dut.i_spi_channel2.i_spi_slavel.mosi 2
...picking matching object, which is a SC input and additionally of
type boolean. Matching object: 8 dut.i_spi_channel3.i_spi_slavel.mosi 3

```

```

...picking matching object, which is a SC input and additionally of
type boolean. Matching object: 11 dut.i_spi_channel4.i_spi_slavel.mosi 4
...sum up: System includes 4 potential fault injection target objects:

```

## 2.7 Anwendung als Stimuligenerator

Bisher wurde die dynamische Injektion von Strukturen in ein Systemmodell nur unter dem Aspekt der Fehlerinjektion beleuchtet. Der beschriebene Ansatz ist jedoch universeller einsetzbar. Die Bibliothek kann beispielsweise auch als „In-Depth“-Stimuligenerator angewendet werden, wofür Abbildung 4 ein Beispiel liefert. Damit können Testfälle initiiert werden, bei denen interne Systembestandteile direkt in ihrer vollständiger Systemumgebung stimuliert werden. Wohlgermerkt besteht die Möglichkeit der Anwendung als Stimuli-

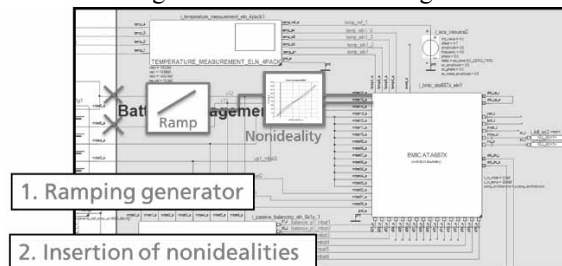


Abbildung 4: Anwendungsbeispiel Stimuligenerator generator, ebenso wie die als Fehlerinjektionswerkzeug auch für bereits existierende Systemmodelle.

## 3 Anwendung des Ansatzes am Beispiel eines Batteriemanagementsystems

Ein wichtiger Aspekt beim Entwurf von Systemkomponenten der Automobilindustrie ist der Nachweis deren funktionaler Sicherheit. Der hier vorgestellte Bibliotheksansatz ist geeignet, diesen Nachweis insbesondere bei Auftreten von Fehlern zu erbringen und hat sich bereits in verschiedenen Projekten bewährt. U.a. konnten die verschiedensten Aspekte des Ansatzes im öffentlich geförderten Projekt IKEBA [1] auf ein komplexes heterogenes Systemmodell eines Batteriemanagementsystems für die Automobilindustrie angewendet werden. Dieses Modell, dessen Toplevel in Abbildung 5 dargestellt ist, enthält neben analogen und digitalen Hardwarekomponenten auch die Software-Applikationen des Zielsystem und bezieht nichtelektrische Effekte mit ein. Das Batteriemodell wurde unter Anwendung von Charakterisierungsergebnissen realer Batteriezellen erstellt, welche durch das KIT in Karlsruhe bereitgestellt wurden. Ein wesentliches Ziel des Projektes war es, die Fähigkeit der Batteriemanagement-Software zu untersuchen, schnell genug Fehler zu erkennen und, wenn notwendig, darauf geeignet zu reagieren. Ein solches Fehlerszenario ist die Beschädigung einzelner Zellen oder deren Überhitzung.

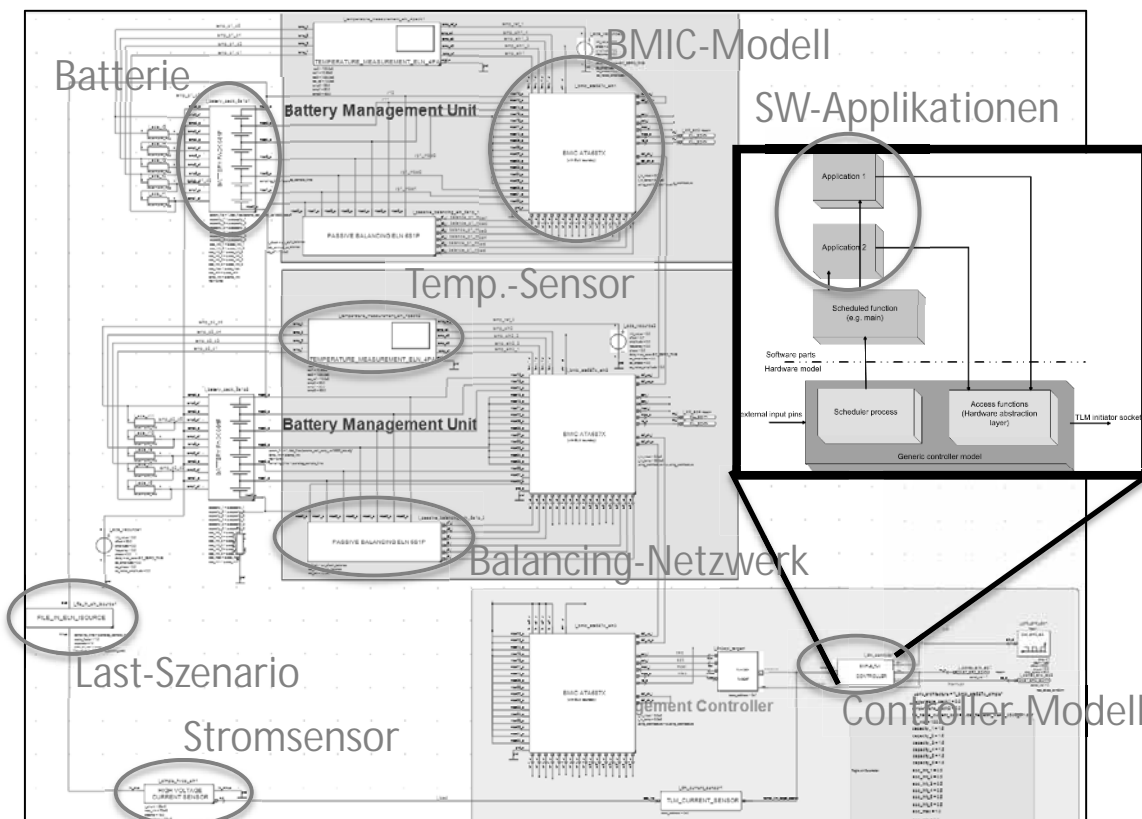


Abbildung 5: heterogenes IKEBA-Toplevelmodell

Der Ansatz zur dynamischen Fehlerinjektion wurde im Rahmen des Projektes exemplarisch für die folgenden Aufgaben erfolgreich u.a. eingesetzt:

- Untersuchung des Systemverhaltens beim Verlassen der spezifizierten Spannungs- und Temperaturbereiche einzelner Zellen
- Untersuchung von SPI-Kommunikationsfehlern
- Provozierung des dynamischen Load-Balancings zwischen den Zellen
- Erzeugen spezieller Stimuli, z.B. für die ADCs im Monitoring-IC

Das Besondere bei der Anwendung des dynamischen Fehlerinjektionsansatzes: Bei der Ausführung des Testfalls wird durch den Simulator das Systemmodell, welches nur das fehlerfreie Verhalten beinhaltet, geladen und damit die zu simulierende Modulhierarchie erstellt. Unmittelbar vor Beginn der Simulation werden Verbindungen innerhalb dieser Modulhierarchie durch den dynamischen Fehlerinjektionsansatz aufgetrennt und das gewünschte Fehlermodell eingefügt. Beispielhaft zeigt die folgende Darstellung die Anwendung der Bibliothek zur Injektion von Fehlern in Modelle von Batteriepaketen. Hierbei werden im

Fehlerfall einzelne Zellen durch alternative Spannungsquellen überbrückt. Realisiert wird dies durch die Instanziierung von „Bridge“-Fehlermodellen in der Testfallumgebung. Das Auftrennen der Modulhierarchie und das Einfügen der Fehlerstruktur erfolgt dabei im Hintergrund, also ohne dass der Anwender darauf Einfluss nehmen muss.

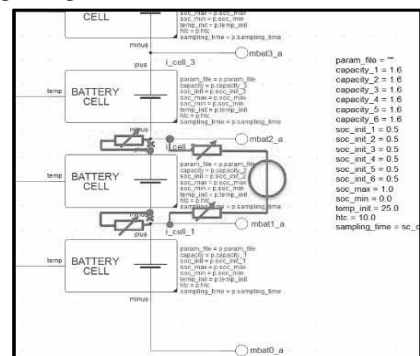


Abbildung 6: Fehlerinjektion in Batterie

Im Detail realisiert wird dies durch das Einfügen von Widerständen in Reihe zu den Anschlüssen der entsprechenden Zellen. Damit ist es möglich, zwischen einer nieder- und hochohmigen Verbindung umzuschalten. Zusätzlich wird parallel zur Zelle eine Spannungsquelle geschaltet, deren Verbindung ebenso umschaltbar ist. Anschließend kann während der Simulationslaufzeit zwischen fehlerfreiem und fehlerbehafteten Verhalten

beliebig umgeschaltet werden. Ebenso kann durch die Testfallbeschreibung der Wert der Fehlerspannung

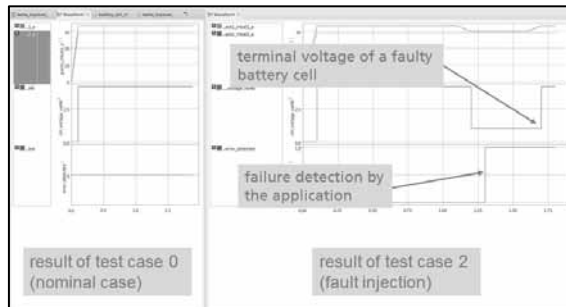


Abbildung 7: Vergleich zwischen Nominal- und Fehlerfall definiert werden. Abbildung 7 zeigt einen Ausschnitt eines Vergleichs der Simulationsergebnisse für fehlerfreien (nominal) und fehlerbehafteten Testfall.

### 3.1 Hardware – In – The – Loop – Anwendung (HiL)

Die Effizienz der Systembeschreibungssprache SystemC AMS konnte bereits in vielen Projekten nachgewiesen werden und bietet auch die Möglichkeit der Anwendung auf Echtzeitsysteme, wie dSpace [2]. Im Rahmen des IKEBA-Projektes konnte auch die Anwendung des Fehlerinjektionsansatzes in einem HiL-System validiert werden. Dynamische Fehlerinjektionen sind dabei für die virtuellen Komponenten eines solchen HiL-Systems möglich. Aktivierungen dynamisch eingefügter Fehlermodelle können jedoch auch in Abhängigkeit realer Hardware-signale erfolgen. Die Vorteile, die sich in diesem Zusammenhang ergeben sind vielfältig, wie beispielsweise:

- Untersuchungen in Grenzbereichen und darüber hinaus ohne reale Hardwareschäden zu riskieren
- Detektion und Reaktion auf Fehler untersuchen
- Einbeziehung bereits existierender Hardwarekomponenten
- Auslagerung sehr rechenintensiver Systembaugruppen in reale Hardware, usw.

Mit dem hier gezeigten Ansatz können auf sehr effiziente Weise die sich aus dem ISO 26262 Standard ergebenden Anforderungen für die Verifikation von Systemen speziell der Automobilindustrie erfüllt bzw. unterstützt werden.

### 3.2 Einbeziehung statistischer Aspekte

Ein weiterer interessanter Aspekt ist die Kombination von Fehlerinjektion und statistischer Analysen. Das

Auftreten von Fehlern in realen Systemen kann von verschiedensten Bedingungen abhängen und ist oft nicht vorhersagbar. Stand der Technik zur Beschreibung von zufälligen Bedingungen und Verhalten ist die Anwendung statistischer Methoden [3]. Bei der Entwicklung der Fehlerinjektionsbibliothek wurde dies berücksichtigt. So ist es möglich, Verteilungsfunktionen für ein realitätsnahes Fehlerverhalten einzubinden.

Der folgende Codeausschnitt zeigt beispielhaft das statistische Auswürfeln von Fehlerort und Fehlerzeitpunkt für ein periodisches Szenario mit mehreren Zielorten, an denen Fehler auftreten können. Die zu verwendenden Verteilungsfunktionen können durch den Anwender festgelegt werden.

```

////////////////////////////////////
/// occurrence time and active failure location are statistical
/// varied and activated/ deactivated in a sequence loop
////////////////////////////////////
scenario1 = new fault_scenario_template<bool> >
    (std::make_pair(port, port_b), fault_injection_base::FAULT_CROSSTALK,
     fault_injection_base::REPLACE);
scenario1->stat_failure_time_function = exponential;
scenario1->location_function         = fault_location;
scenario1->set_mean_fault_occurrence(sc_core::sc_time(30.0, SC_US));
scenario1->set_fault_duration(sc_core::sc_time(5.0, SC_US));

stat_mode_val = STAT_STATISTICAL_SUB;

```

## 4 Ausblick

Eine automatisierte Erstellung von Fehlerinjektionstestfällen bietet eine weitere Erhöhung der Anwenderfreundlichkeit und stellt gleichzeitig ein effizientes Mittel zur schnellen Nutzung des Ansatzes dar. Ausgangspunkt können dabei Konzeptspezifikationsdokumenten oder sog. „Requirement Management Tools“ wie DOORS sein. Darüber hinaus haben bereits Arbeiten zur automatisierten Auswertung und Klassifizierung, bezogen auf die Vorgaben des ISO-Standards begonnen.

## 5 Referenzen

- [1] Integrierte Komponenten und integrierter Entwurf energieeffizienter Batteriesysteme (Vorhaben IKEBA). URL: <http://www.iam.kit.edu/awp/ikeba/>
- [2] Einwich K.; Arndt, T. (2012): SystemC-AMS basierte Modellierung, Simulation und HiL Echtzeitsimulation für Anwendungen der Automobilelektronik. In: ASIM-Konferenz STS/GMMS 2012. Vienna: ARGESIM-Verl, S. 203–208.
- [3] T.Markwirth, J.Haase, K.Einwich; Statistical Modeling with SystemC-AMS for Automotive Systems; FDL'08; 2008

Die diesem Beitrag zugrundeliegenden Arbeiten wurden mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen FKZ 16N12440 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren

# Vergleichende Untersuchungen an DC/DC-Wandler Modellen für die Bordnetzsimulation

Thomas Lang<sup>1</sup>

<sup>1</sup>Robert Bosch GmbH, Tübinger Str. 123, 72762 Reutlingen

*thomas.lang3@de.bosch.com*

Gleichspannungswandler spielen eine große Rolle für den Energietransfer in Teilbordnetzen im Fahrzeug, beispielsweise in Zweispannungsbordnetzen mit 48V und 12V Ebene. Der vorliegende Beitrag untersucht die Eignung unterschiedlicher Wandlermodelle für die Systemsimulation. Dazu wurden unterschiedliche Modellierungsansätze für einen Tiefsetzsteller (Buck-Wandler) untersucht. Die Ansätze waren sowohl generisch, als auch basierend auf Kennwerten eines real existierenden Geräts. Aus der Modellierung sich ergebende Unterschiede in den Strom- und Spannungsverläufen wurden analysiert und mit Messungen an der Hardware verglichen. Ein besonderer Aspekt beleuchtet dabei die Simulation eines Beispielbordnetzes mit Modellierung der 48V Batterie.

## 1 Einleitung

Gleichspannungswandler sind in modernen Fahrzeugbordnetzen nicht mehr wegzudenken. Überall dort wo mehrere Spannungsebenen im Fahrzeug miteinander verbunden, und Spannungsstabilität gewährleistet werden muss, ist deren Einsatzbereich. In der Bordnetzsimulation ist es daher wichtig, hinreichend genaue Modelle zu haben um Bordnetze bezüglich Energiebereitstellung und -verteilung, unter Berücksichtigung der Spannungsstabilität auslegen und bewerten zu können. Es stellt sich die Frage welches Modellierungskonzept das Verhalten eines DC/DC-Wandlers hinreichend genau für verschiedene Lastszenarien im Fahrzeug beschreibt. Für die Systemsimulation wurde das Tool OpenModelica verwendet.

Grundmodelle für das Bordnetz wurden hauptsächlich aus der sich in der Vorbereitung befindlichen Modellbibliothek „OnBordSystem“ des VDA FAT Arbeitskreises AK30 [1] verwendet.

Die Modelle wurden um ein kennlinienbasiertes Modell eines 48V Batteriemoduls bestehend aus Einzelzellen erweitert.

### 1.1 Wandlertyp

Aus einer Auswahl von möglichen Wandlerausführungen wurde ein Wandler mit folgenden Eigenschaften ausgewählt

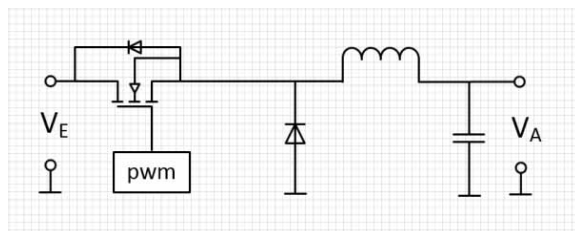
- Tiefsetzsteller, galv. getrennt
- Nom. Eingangsspannung: 48V
- Nom. Ausgangsspannung 12V

- Nom. Leistung 1000W

Für diesen Wandler lag ein elektrisch voll funktionsfähiges Demonstratorgerät vor. Dieses Gerät wurde zur Validierung der im Beitrag untersuchten Modelle herangezogen.

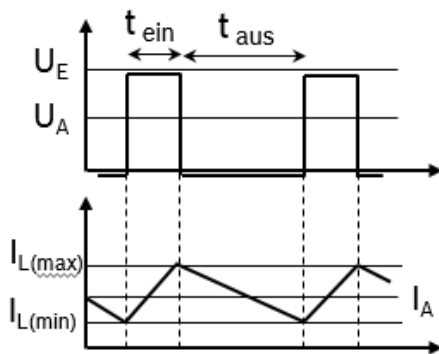
### 1.2 Aufbau und Funktion eines Tiefsetzstellers

Der Tiefsetzsteller besteht vereinfacht aus einem Leistungsschalter mit pwm-gesteuertem Gate-Treiber, einer Diode, einer Induktivität sowie eine Kondensator (Abbildung 1)



**Abbildung 1.** Ersatzschaltbild Tiefsetzsteller

Durch getaktetes Ein- und Ausschalten des Schalter ergibt sich der in Abbildung 2 exemplarisch gezeigte Spannungs- und Stromverlauf [2].



**Abbildung 2.** Spannungs- und Stromverlauf (buck-converter)  $U_E$ : Eingangsspannung,  $U_A$ : Ausgangsspannung,  $I_L$ : Laststrom

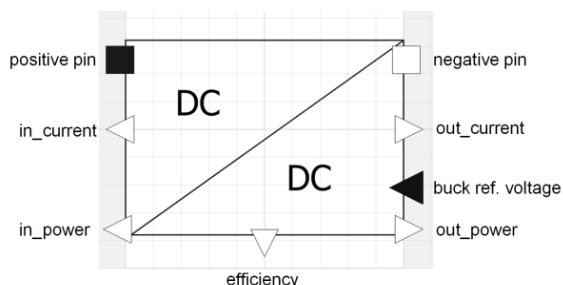
### 1.3 Modellierung des Wandlers

In der Modellierung der DC/DC-Wandler wurden drei Ansätze verwendet

1. Ein Verhaltensmodell basierend auf den elektrischen Kenngrößen des Demonstratorgeräts incl. der Nachbildung der Regelstrecke
2. Ein generisches Verhaltensmodell welches für grundsätzliche Untersuchungen in einem Fahrzeugmodell herangezogen werden soll.

#### Verhaltensmodell

Das Verhaltensmodell wurde basierend auf Datenblattangaben für Ein- und Ausgangsspannungen und -ströme, Wirkungsgrad erstellt. Für die Regelung auf Sollspannung auf Sekundärseite wurde ein PID-Regler erstellt. Die Referenzspannung wurde auf 13,2V festgelegt. Abbildung 3 Zeigt den Wandler mit den entsprechend Anschlüsse. Ströme, Leistungen und Wirkungsgrad lassen sich über Signalpins auslesen.



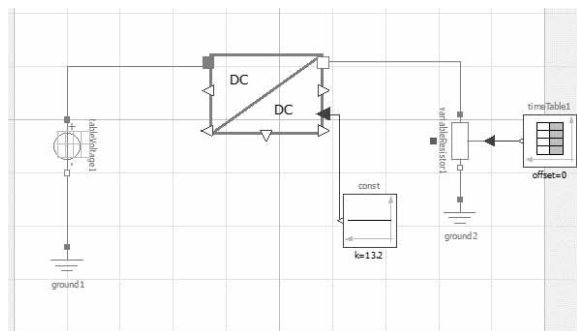
**Abbildung 3.** Icon des Wandlerrmodells (Modell 1)

#### Generisches Modell

Im generischen Modell wurde ein elektrischer Schalter modelliert, dessen Ein- und Ausschalttaktung über einen Pulsbreitenmodulator (pwm) und einen Regler mit Spannungsreferenz erzeugt wurde. Der Modulator besteht aus einem Sägezahngenerator und einen Komparator. Ein LC-Tiefpassfilter wurde nachgeschaltet.

#### Testbench

Die Testbench des Wandlers besteht aus einer Spannungsquelle auf 48V-Seite und einem Lastwiderstand auf der 12V -Seite, welcher über eine timeTable eingelesen wird. Dieser widerstand ist über die Zeit variabel im Bereich von 0.5 Ohm bis 0.8 Ohm



**Abbildung 3.** Testbench des 48V/12V Wandlers

#### Messungen

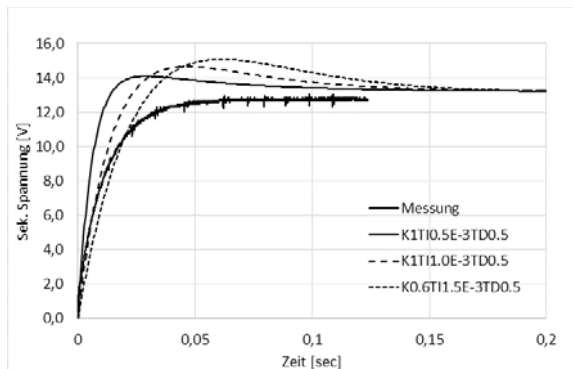
Für Messungen am Gerät wurde ein äquivalenter Aufbau gewählt. Die Last bestand auf 12V-Seite aus einem Widerstand von 1.0 Ohm. Auf Seiten des Eingangs wurde eine Konstant-Spannungsquelle verwendet. Die Spannungsmessung erfolgte auf beiden Seiten des Wandlers mittels eines Oszilloskops, das auf die steigende Flanke des Eingangskanals getriggert wurde

## 2 Simulation

### 2.1 Simulation Testbench

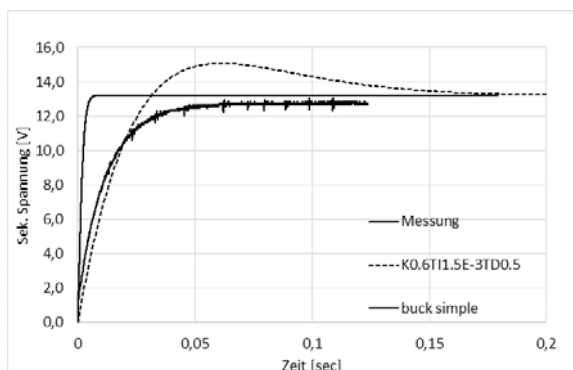
Zunächst wurde das Verhaltensmodell mittels der Testbench simuliert. Dabei wurden Unterschiedliche Regelparameter im PID-Modell verwendet. Abbildung 4 stellt die Spannungsverläufe gegenüber der Messung dar. Der Regler im Gerät arbeitet sehr schnell und erreicht bei der Belastung mit ca. 26A nach etwas mehr als 50 msec die Sollspannung und neigt auch nicht zum Überschwingen. Im Gegensatz dazu werden im Modell davon abweichende Anstiegszeiten gefunden. Je

nach Parameter übersteigt die Ist-Spannung die Sollspannung von 13,2V kurzzeitig um bis zu ca. 2 Volt. Diese Spannungsüberhöhung wäre in einem Fahrzeugbordnetz aber noch nicht kritisch zu sehen [3]. Die besten Ergebnisse der Simulation gegenüber den Messungen wurde mit folgenden Parametern (Proportionalitätsfaktor, Integralanteil, Differentialanteil) erreicht, K-Faktor: 0.6, TI: 1.5E-3, TD:0.5. Eine weitere Anpassung der Reglereigenschaften kann noch zu einer besseren Übereinstimmung führen. Dazu können aber auch im Modell vorhandene LC-Filter angepasst werden.



**Abbildung 4.** Vergl. Messung mit Simulation mit unterschiedlichen Regelparametern

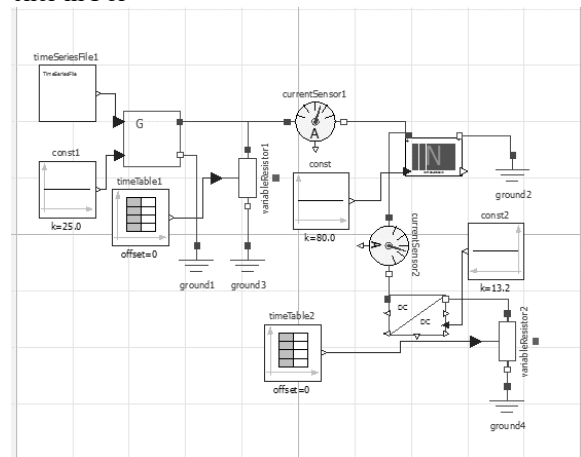
Für den Vergleich wurde die Testbench nun auch mit dem generischen Modell („buck simple“) bestückt. Abbildung 5 zeigt die Gegenüberstellung der Ergebnisse. Das generische Modell zeigt ideales Regelverhalten auf Sollspannung ohne Überschwinger. Trotz der Taktung des Schalters mit hoher Frequenz (20 kHz) und einer internen Diode (vgl. Abbildung 1) beträgt die Simulationszeit nur wenige Sekunden



**Abbildung 5.** Vergl. Messung mit Simulation mit Verhaltensmodell (K0.6...) und generischem Modell („buck simple“) mit unterschiedlichen Regelparametern

## 2.2 Simulation Bordnetzmodell

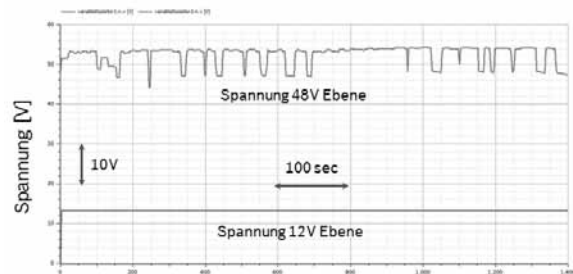
Das Verhaltensmodell des Wandlers wurde in ein Modell eines Zweispannungsbordnetzes eingebaut. Das Bordnetz besteht aus einer 48V Maschine, einem veränderlichen 48V Verbraucher in Form eines



**Abbildung 6.** Nachbildung eines Zweispannungsbordnetzes

variablen Widerstands sowie einer 48V Batterie. Ein variabler Lastwiderstand auf 12V-Seite stellt die 12V Bordnetzlast dar.

In das Generatormodell wird ein Drehzahlprofil eingelesen, welches einem Fahrscenario entspricht. Die Ergebnisse sind in Abb. 7 gezeigt. Spannungsschwankungen auf 48V Seite bedingt durch sich dynamisch ändernde Lastströme, Batterieladezustand und Generatordrehzahlen werden durch den Wandler „gepuffert“, so dass sich diese nicht negativ auf die 12V-seite auswirken. Die 12V-Seite wird durch den Wandler in der Spannung stabilisiert.



**Abbildung 7.** Spannungsverlauf am Eingang und Ausgang des DC/DC-Wandlers

### 3 Zusammenfassung

Die beiden beschriebenen Modellierungsansätze für Tiefsetzsteller eignen sich in gleichem Maße für die Bordnetzsimulation. Beide Modelle sind bezüglich ihrer Komplexität und den Simulationszeiten handhabbar. Für die Untersuchung von langsamen sich ändernden elektrischen Vorgängen im Fahrzeug (z.B. Batterieladeströme) sind beide Modelle gut geeignet. Das generische Modell stellt einen sehr idealen Ansatz dar. Für die Untersuchung der Spannungsstabilität im Bordnetz insbesondere in Hybrid- und Elektrofahrzeugen [4] sollten die Modelle noch weiter ausgefeilt werden. Nicht zuletzt sind auch weitere Messungen, insbesondere der Stromverläufe, am realen Wandler dazu notwendig.

### 4 Referenzen

- [1] VDA FAT Arbeitskreis AK30 "Entwurfsmethodik und Simulation elektrischer und mechatronischer Systeme im Kfz". Verfügbar über <https://www.vda.de/de/services/Publikationen/Publikation.~897~.html> bzw. beim Autor
- [2] U.Tietze, Ch.Schenk. *Halbleiter Schaltungstechnik*. 13. Auflage, Kap.15.5, Springer Verlag 2010
- [3] ZVEI: Spannungsklassen in der Elektromobilität, Dezember 2013. Verfügbar über <http://www.zvei.org/Publikationen/Spannungsklassen-Elektromobilitaet.pdf>
- [4] EEHE - Elektrik/Elektronik in Hybrid- und Elektrofahrzeugen und elektrisches Energiemanagement. *Internationale Tagung*, 8./9.06.2016, WIES-LOCH

# Konzeptplanung von Montagesystemen mit graphenbasierten Entwurfssprachen

Theresa Breckle<sup>1</sup>, Jens Kiefer<sup>1</sup>, Markus Kiesel<sup>2</sup>, Martin Manns<sup>3</sup>

<sup>1</sup>Hochschule Ulm

<sup>2</sup>Hochschule Albstadt-Sigmaringen

<sup>3</sup>Universität Siegen

*breckle@hs-ulm.de*

Produzierende Unternehmen unterliegen einem kontinuierlichen und zum Teil fundamentalen Wandel. Bereits heute stehen diese Unternehmen unter einem immer stärker zunehmenden Zeitdruck, einem anhaltenden Kostendruck, und gleichzeitig steigt die Komplexität der Produkte und Prozesse. Für Unternehmen, deren Kernkompetenz Montageprozesse sind, ist daher ein Ansatz, mit dem schneller bessere und nachhaltigere Ergebnisse bei der Planung von Montagesystemen erzielen zu können, erforderlich. In diesem Beitrag wird die Idee eines innovativen Ansatzes für einen automatisierten Entwurfsprozess, der die Konzeptphase von Montagesystemen umfasst, dargestellt. Die Berücksichtigung von unterschiedlichen Eingangsgrößen (z. B. die Produktstruktur) und Anforderungen (z. B. wirtschaftliche oder technische) an die Auslegung eines Montagesystems, sind wesentliche Elemente, um ein ganzheitliche Gestaltung unter Berücksichtigung aller relevanten Eingangsgrößen zu erzielen. Häufig sind diese Eingangsgrößen mit Unsicherheiten behaftet, die es ebenfalls in der Modellierung zu berücksichtigen gilt.

Als ein möglicher Ansatz für eine effiziente und schnelle Planung mit Hilfe einer wissensbasierten Entwurfsmethode ist der Einsatz von graphenbasierten Entwurfssprachen. Durch die wissensbasierte Entwurfsmethode kann ein digitales und regelbasiert ausführbares Abbild des Entwurfsprozesses dargestellt werden. Dabei wird das Entwurfswissen in Form von Regeln und Vokabeln abgelegt. In einem Engineering-Framework wird dieses ausgeführt und ein zentrales Modell erzeugt. Durch den automatisierten Entwurfsprozess können dann vollautomatisch weitere Modelle (z. B. in domänenspezifischen Anwendungen) erzeugt werden, die anschließend validiert, bewertet oder im Entwurfsprozess weiterverarbeitet werden können.

## 1 Einleitung

Megatrends wie beispielsweise die Globalisierung, Digitalisierung, Individualisierung oder auch der Nachhaltigkeitsgedanke beeinflussen und verändern produzierenden Unternehmen. Um wettbewerbsfähig zu bleiben, müssen sich produzierende Unternehmen diesen grundlegenden Veränderungen stellen [1]. Der Begriff „Industrie 4.0“ fasst einen Teil dieser Megatrends zusammen [2]. Bereits heute ist eine Verkürzung des Produktentstehungsprozess (PEP) mit dem Ziel, eine schnellere Markteinführung zu realisieren und einen früheren Start der Produktion (SOP) zu erreichen, eine bestehende Herausforderung [3]. Diese Megatrends, verkürzte Produktlebenszyklen, turbulente Märkte sowie sich ändernde Kundenanforderungen führen zu komplexeren Produkten, Prozessen und Technologien [4]. Dazu kommt der Druck auf die Unternehmen, wirtschaftlich erfolgreich zu sein. Diese Herausforderungen erfordern Methoden und Strategien, um effizient, schnell und ressourcenschonend planen und fle-

xibel reagieren zu können. Gerade die frühe Planungsphase hat einen großen Einfluss auf die Kostenfestlegung [5]. In diesem Beitrag wird ein neuer Ansatz eines weitgehend automatisierten Entwurfsprozesses für die Konzeptionierung eines Montagesystems vorgestellt. Dabei werden Anforderungen an das System ebenso berücksichtigt wie dessen Komplexität, Unsicherheiten der Eingangsgrößen sowie die Idee der Integration in den gesamten Produktlebenszyklus. Umgesetzt wird dieser Ansatz mit sogenannten graphenbasierten Entwurfssprachen [6–8]. Voraussetzung für den Entwurf des Montagekonzepts ist ein weitgehend fertiggestelltes Produktkonzept.

## 2 Grundlagen

### 2.1 Planung von Montagesystemen

Eine Montage hat zur Aufgabe aus einer Summe an Einzelteilen und zum Teil vormontierten Baugruppen ein komplettes Produkt herzustellen. Dabei ist das

Montagesystem ein Teilbereich des Produktionssystems [9]. In der Literatur finden sich zahlreiche klassische Prozessmodelle für die Planung von Produktionssystemen, wie z. B. von Kettner [10], Grundig [11], Wiehndahl und Nyhuis [12] oder Schenk [13]. Darauf basierend gibt es neuere Ansätze wie z. B. von Suh [14], das Gegenstromverfahren [15] oder bezogen auf die Montage von Konold und Reger [16]. Diese Prozessmodelle unterscheiden sich in der Anzahl der Phasen, Start- und Endpunkt und ihrem Detaillierungsgrad. Gemeinsam sind ihnen der grundsätzliche Inhalt und die Art des analytischen Verfahrens. Sie können in die in die Phasen Vorbereitung, Strukturplanung, Detailplanung, Ausführungsplanung und Ausführung eingeteilt werden [17].

## 2.2 Entwurfsmethodik

Der Entwurfsprozess eines konkreten Objekts stellt die Synthese eines finalen Entwurfs dar, welcher anfangs beschriebene Definitionen des zu entwerfenden Objektes berücksichtigt. Anforderungen (Requirements) und sich im Entwurfsprozess ergebende Zwangs- und Randbedingungen (Constraints) sind zu erfüllen [18]. Es gibt keine generell akzeptierte und vereinheitlichte Entwurfstheorie des ingenieurwissenschaftlichen Entwurfsprozesses [6]. Eine oft zitierte Quelle der Entwurfsmethodik stellt das Werk von Pahl und Beitz [19] dar, sehr ähnlich ist die entsprechende Richtlinie des Vereins Deutscher Ingenieure (VDI) [20]. Darauf basierend lässt sich das Vorgehen im Entwurfsprozess in folgende Schritte aufteilen: Planen und Klären der Aufgabe, Konzipieren, Entwerfen und Ausarbeiten. Während der Konzeptphase wird die prinzipielle Lösung eines Produkts oder eines Systems erarbeitet, dafür werden die wesentlichen Problemstellungen in Teilaufgaben (z. B. Teilfunktionen) zerlegt und nach passenden Lösungen gesucht, es wird auf vorhandenes Lösungswissen zurückgegriffen. Die Kombination von Teillösungen führt zu unterschiedlichen Lösungsvarianten, die anschließend nach bestimmten Kriterien (z. B. wirtschaftliche oder technische) bewertet und so die geeignetste Lösungsvariante ausgewählt werden. Komplexe Planungsaufgaben erfordern Iterations Schleifen [19], [21]. Nach Feldmann [22] kann diese Entwurfsmethodik auch auf den Entwurf von Montagesystemen übertragen werden.

Eine zunehmende Komplexität der Entwürfe sowie eine gewünschte Steigerung der Effizienz erfordern computergestützte Methoden zur Optimierung und Automatisierung [9] im Entwurfsprozess.

## 2.3 Modellierung von Montagesystemen

Zur Modellierung von Montagesystemen existiert in Literatur und Industrie eine Vielzahl von verschiedenen Ansätzen, Methoden und Softwarelösungen (siehe z.B. [23,24]). Bei der Modellierung von Montagesystemen bilden Modelle nicht nur ein Objekt (z. B. Anlage) ab, sondern sind auch Arbeitsmittel im Planungsprozess [13]. In Anlehnung an [25] können bei der Modellgenerierung von ereignisdiskreten Systemen (zu denen Montagesysteme zählen) drei Grundansätze unterschieden werden: Parametrische, strukturbasierte und hybrid-wissensbasierte. In [25] erfolgt ein Überblick über existierende Ansätze zur automatisierten Modellgenerierung. Als ein generisches Modellierungskonzept ist die Unified Modelling Language (UML) zu nennen, die in einem objektorientierten Ansatz z. B. in [17] verwendet wird. Der objektorientierten Ansatz mit UML bildet die Repräsentation des Entwurfswissens (im UML-Klassendiagramm) und die Methode (Umsetzung des Planungswissens im zeitlichen Verlauf durch das UML-Aktivitätsdiagramm) ab. So können diese Modelle wiederverwendet werden [26].

## 2.4 Graphenbasierte Entwurfssprachen

Als eine wissensbasierte Entwurfsmethode kann durch die Entwurfssprachen der Entwurfsprozess regelbasiert ausgeführt und digital abgebildet werden. Dadurch ist er wiederausführbar. Dies erlaubt eine Automatisierung des Entwurfs und zählt dadurch zu den formalisierten Entwurfssystemen. Das Konzept der graphenbasierten Entwurfssprachen [27,6,7] hat sich in den vergangenen 15 Jahren zu einem leistungsfähigen Engineering-Framework entwickelt. Dieses Konzept leitet sich von den von Menschen gesprochenen Sprachen ab. Diese können komplexe und verknüpfte Zusammenhänge mit Hilfe von Vokabeln und Regeln, die eine Grammatik bilden, repräsentieren. Ein gültiger Satz entspricht einer zulässigen Kombination aus Vokabeln und beschreibt so syntaktisch gültige Entwürfe [6]. Für die Entwurfssprache entstehen die Vokabeln durch die Dekomposition des Entwurfsproblems. Die Regeln leiten sich aus den formalen Regeln (z. B. aus grundlegenden Design Patterns) und implizitem menschlichen Expertenwissens ab. Die logische Abfolge der Regeln bildet das so genannte Produktionssystem [8]. Als Modellierungssprache wird die UML eingesetzt (mehr Informationen siehe [28]). Vokabeln werden dabei durch Klassen dargestellt. Das vernetzte Wissen wird im Klassendiagramm repräsentiert und

Regeln durch graphische UML-Regeln oder Java-Regeln abgebildet. Dabei kann das Klassendiagramm als Ontologie bezeichnet werden. Durch den so genannten Entwurfscompiler werden die Regeln im Produktionssystem verarbeitet, währenddessen wird der Entwurfsgraph inkrementell aufgebaut. Die Wissensverarbeitung innerhalb des vollständigen Entwurfsprozesses ist in Abbildung 1 dargestellt.

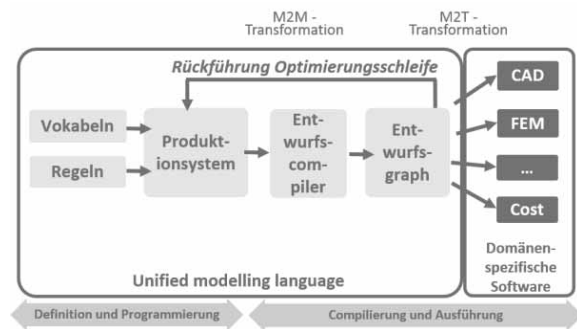


Abbildung 1. Wissensverarbeitung Entwurfsprozess

Der Design Compiler 43 wurde von der IILS GmbH in Zusammenarbeit mit dem Institut für Statik und Dynamik der Luft- und Raumfahrtindustrie (ISD) an der Universität Stuttgart entwickelt [29]. Die Entwurfsrepräsentation wird durch den Design Compiler erzeugt, wie auch in anderen modernen Programmiersprachen. Dabei können den Klassen auch Constraints angehängt werden. Der Entwurfscompiler führt dabei die Regeln für einen automatisierten Aufbau des Entwurfsgraphs aus, der den Entwurf durch eine Model-to-Model Transformation beschreibt. Zum anderen können durch eine Model-to-Text-Transformation (Analyse-)Ergebnisse über Plug-Ins an domänen-spezifische Softwaresysteme exportiert und verarbeitet werden, sowie die Ergebnisse wieder zurückgeführt werden. Dabei ist die Definition und die Erstellung der Entwurfssprache eine Ingenieursaufgabe, die Ausführung des Modells erfolgt durch den Design Compiler [30,8].

## 2.5 Problembeschreibung

Durch eine Integration der realen und der virtuellen Welt sind Planungsmodelle die Grundlage, um komplexe System erstellen zu können [2]. Im Kontext der Digitalen Fabrik hat die Konzeptplanung das Ziel, eine ganzheitliche Planung unter Berücksichtigung aller Anforderungen und Unsicherheiten in Verbindung mit dem Produktkonzept durchzuführen und so die relevanten Strukturen, Prozesse und Ressourcen zu planen [23]. Dabei entsteht eine Vielzahl an Modellen, was zu Konsistenzproblemen führen kann. Nach Reichwein [26] bestehen zwei Ansätze zum Austausch von Daten

zwischen Modellen: Jedes Modell tritt mit jedem Modell in den Austausch (führt zu einer  $(n*(n-1))$  Anzahl an Schnittstellen) oder ein zentrales Modell wird eingesetzt, das alle Informationen enthält. Dadurch tritt das Konsistenzproblem nicht auf.

Wie oben beschrieben, erfordern komplexe Planungsaufgaben Iterationsschleifen, um auch beispielsweise Unsicherheiten im Planungsprozess berücksichtigen können. Diese sind in den bestehenden Ansätzen häufig nicht automatisierbar, was zu Akzeptanzproblemen und aufgrund von manuellen Tätigkeiten führen kann. Modelländerungen und Ergebnisse von Simulationen können bei bestehenden Ansätzen häufig nicht in die ursprünglichen Datenhaltungssysteme zurückgespielt werden [25]. Modellierungsansätze, die einen lebenszyklusübergreifenden Ansatz verfolgen, fehlen laut [25] und [2]. Dabei ist auch die Frage nach einer tool-übergreifender Modellbildung bzw. einem allgemeinen werkzeugunterstützten Meta-Modell relevant. Die Wiederverwendung von Expertenwissen für die Modellierung ist gewünscht [25].

Daher ist für die Konzeptplanung von Montagesystemen ein formalisierter, den ganzheitlichen Planungsansatz verfolgender, und auf einem zentralen Datenmodell basierender Ansatz, der eine (Teil-) Automatisierung der Planung ermöglicht, erforderlich.

## 3 Konzeptplanung mit graphenbasierten Entwurfssprachen

Die Fähigkeit eines wiederausführbaren und durchgängigen Entwurfsprozesses verbessert die Güte der Ergebnisse der frühen Phase und verkürzt im gleichen Zuge die Entwurfszeit, wie z. B. in [31] für den Produktentwurf beschrieben. Durch die Übertragung der Idee der graphenbasierten Entwurfssprachen auf die Konzeptphase der Montagesystemplanung entsteht ein neuer Ansatz für eine computergestützte frühe Planung von Montagesystemen. Durch diese Art der abstrakten Wissensrepräsentation kann die Idee der ganzheitlichen, schnelleren und effektiveren Planung durch eine formale Ontologie verwirklicht werden. Ganzheitlich ist hier im Sinne der Definition der Digitalen Fabrik definiert [23], die darüber hinaus eine Wirtschaftlichkeitsbetrachtung, als auch ein umfassendes Anforderungsmanagement beinhaltet. Abbildung 2 zeigt die Wissensverarbeitung der Konzeptplanung von Montagesystemen mit den Design Compiler. Im (abstrakten) Entwurfswissen sind Entwurfshandlungen, Domänenobjekte und Domänenwissen enthalten. Als Output

können möglich Ergebnisse z. B. ein Layout, ein Lastenheft für eine Anlage (Spezifikation) oder eine Wirtschaftlichkeitsberechnung sein, die in einer domänen-spezifischen Software modelliert werden können.

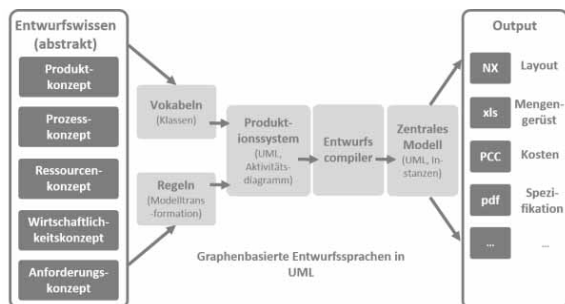


Abbildung 2. Wissensverarbeitung Konzeptplanung

Für die Erstellung einer formalisierten graphenbasierten Entwurfssprache wird das Montagesystem in seine Entitäten zerlegt (Dekomposition). Dabei entsprechen die Entitäten den Vokabeln, wie in Abbildung 3 auf höchster Ebene abgebildet. Für die Dekomposition ist das Wissen über den Inhalt, die Struktur und die Abhängigkeiten des Montagesystems erforderlich. Die dargestellte Ebene wird weiter heruntergebrochen und zur Komplexitätsreduktion in Sub-Design-Sprachen dargestellt, um das Entwurfswissen vollständig abzubilden. Das Produktionssystem (UML-Aktivitätsdiagramm) bildet sich aus Vokabeln und Regeln.

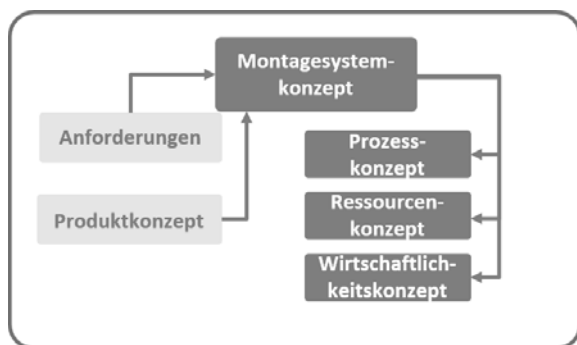


Abbildung 3. Dekomposition Montagesystemkonzept

Nach der Methodik des systematischen Entwerfens von Pahl und Beitz [19] wird eine Modellierung in verschiedene Kategorien aufgeteilt, wie in Abbildung 4 gezeigt. Anforderungen werden in abstrakte Funktionen übertragen und in Lösungsprinzipien umgesetzt. Entsprechend der Lösungsprinzipien werden die Komponenten ausgelegt. Aus der Phase „Klären und Beschreiben der Aufgabe“ sind die Anforderungen an das

Montagesystem, im Beispiel hier an den Montageprozess des Schraubens, bekannt. Mögliche Lösungsprinzipien für die Gestaltung dieses Fügeprozesses sind ebenfalls bekannt. Dieser Übertragungsprozess von Anforderungen zu Komponente erfolgt in der Entwurfssprache über Regeln.

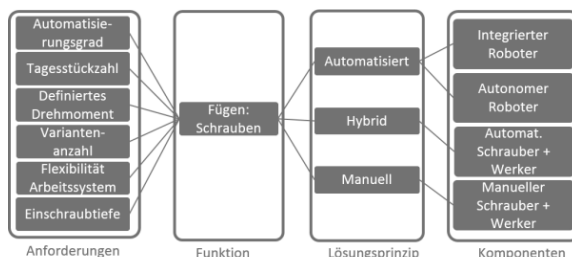


Abbildung 4. Modellierungskategorien Fügeprozess

Das Verständnis der Anforderungen und Abhängigkeiten innerhalb des Modells führen zu den Regeln und deren Reihenfolge um das Produktionssystem zu erstellen. Dieses besteht aus logischen Regeln um den Designgraph zu bauen, welcher den Entwurf des Montagesystems repräsentiert. Einschränkungen können als Constraints modelliert werden.

Der vollständige Entwurfsprozess ist in Abbildung 5 dargestellt. Wird ein realer Planungsfall modelliert, so können nach der Modellierung noch weitere Auswertungstools angeschlossen werden (siehe Abbildung 2). So können bei Bedarf Optimierungen vorgenommen werden und Ergebnisse der Bearbeitung in domänen-spezifischen Modellierungstools (z. B. CAD) auch wieder zurückfließen. Häufig sind in der frühen Phase der Montagesystemplanung Eingangsgrößen noch unsicher. Durch die formalisierte und automatisierte Modellierung können automatisiert Alternativen modelliert werden. Dadurch können mögliche Alternativen frühzeitig analysiert und bewertet werden.

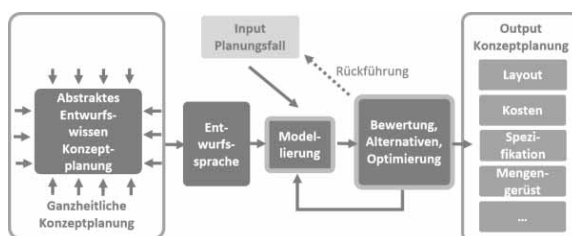


Abbildung 5. Methode graphenbasierte Entwurfssprache für Konzeptplanung Montage

## 4 Diskussion

Ansätze und Methoden für eine computerunterstützte Montageplanung und Modellierung existieren. Umsetzungen mit UML sind dabei Stand der Technik. Ein Ansatz jedoch, der durch seine wissensbasierte und Entwurfsmethode ein digitales und regelbasiert ausführbares Abbild des Entwurfsprozesses darstellt und durch (Teil-)Automatisierung und Wiederverwendbarkeit der Entwurfssprachen eine schnelle und effektive Modellierung der frühen Montagesystemplanung ermöglicht, fehlt nach bestem Wissen der Autoren. Der hier vorgestellte Ansatz der Konzeptplanung von Montagesystemen mit graphenbasierten Entwurfssprachen bietet ein Lösungsansatz für die oben beschriebenen Problemstellungen. Mit einem zentralen Datenmodell, das über Schnittstellen zu domänenspezifischen Programmen verfügt, kann das Konsistenzproblem der Modellierung gelöst werden. Darüber hinaus steigert die Berücksichtigung aller relevanten Eingangsgrößen und Anforderungen in den Planungsprozess die Güte der Planungsergebnisse. Eine lebenszyklusübergreifende Modellierung eröffnet die Perspektive für die Durchgängigkeit des Engineerings über den gesamten Lebenszyklus. Das zentrale Datenmodell und der Aufbau der Methode der graphenbasierten Entwurfssprachen ermöglicht eine Automatisierung bei Iterationen und bei der Rückkopplung von Daten.

Die Berücksichtigung von Unsicherheiten als ein Planungshindernis in der frühen Planungsphase wird in [32] mit einem Fokus auf eine wissensbasierte Planung und Bewertung in der Flugzeugindustrie beschrieben. Die Möglichkeit der schnellen und effizienten Modellierung schafft bei der Berücksichtigung von Unsicherheiten durch die Modellierung und anschließende Bewertung eine nachhaltigere und nachvollziehbare Entscheidungsgrundlage im Rahmen der frühen Planungsphase. Der Ansatz kann einen Beitrag für eine höhere Güte der Planungsergebnisse bei einer kürzeren Planungsphase leisten. Der originäre Aufwand für die Entwurfssprachen ist dabei relativ hoch und ein Einsatz ist wirtschaftlich sinnvoll, wenn die Entwurfssprachen öfter eingesetzt werden [31].

## 5 Ausblick

Der Einsatz von graphenbasierten Entwurfssprachen erfordert ein tiefes und umfassendes Wissen der Montagesysteme und ihrer Planung. Durch eine Ausweitung der bestehenden, vorgestellten Entwurfssprache für Montagekonzepte wird der Entwurfsraum für die

Konfiguration eines Montagesystems weiter aufgespannt. Der hier vorgestellte Ansatz wird weiter ausgearbeitet und an weiteren Beispielen validiert. Durch die vollständige Integration in den gesamten Produktlebenszyklus entsteht ein durchgängiges Modell. Dieses soll in der Lage sein, geänderte Bedingungen innerhalb des gesamten Produktentstehungsprozesses verarbeiten zu können. Durch den Ansatz des zentralen Datenmodells soll es zu einem möglichst optimalen Gesamtergebnis führen und die so entstandenen Entwürfe selbstständig bewertet und optimiert werden können.

## 6 Danksagung

Der in diesem Beitrag vorgestellte Ansatz ist Teil des Forschungsprojektes „Digitaler Produktlebenszyklus“ (ZaFH), welches vom Europäischen Fond für regionale Entwicklung und dem Ministerium für Wissenschaft, Forschung und Kunst des Landes Baden-Württemberg unterstützt wird. Bei diesem Projekt kooperieren verschiedene Hochschulen (Weingarten, Ulm, Albstadt-Sigmaringen, Reutlingen) und die Universität Stuttgart. Mehr Informationen unter (<http://dip.reutlingen-university.de> und [www.rwb-efre.baden-wuerttemberg.de](http://www.rwb-efre.baden-wuerttemberg.de)).

## 7 References

- [1] Westkämper, E., Spath, D., Constantinescu, C., Lentz, J. *Digitale Produktion*. Springer Berlin, 2013.
- [2] BITKOM, VDMA, ZVEI. *Umsetzungsstrategie Industrie 4.0: Ergebnisbericht der Plattform Industrie 4.0*, 2016.
- [3] Wang, L., Shen, W., Xie, H., Neelamkavil, J., Pradasani, A. *Collaborative conceptual design - state of the art and future trends*. Computer-Aided Design 34 34, 981–996, 2002.
- [4] ElMaraghy, W., ElMaraghy, H., Tomiyama, T., Monostori, L. *Complexity in engineering design and manufacturing*. CIRP Annals - Manufacturing Technology 61 (2), 793–814, 2012.
- [5] Ehrlenspiel, K., Kiewert, A., Lindemann, U., Mörtl, M. *Kostengünstig Entwickeln und Konstruieren*. 7. Aufl. ed. Springer Vieweg, Berlin, 2014.
- [6] Rudolph, S. *Übertragung von Ähnlichkeitsbegriffen*. Habilitationsschrift, Universität Stuttgart, Deutschland, 2002.
- [7] Rudolph, S. *Aufbau und Einsatz von Entwurfssprachen für den Ingenieurentwurf*. 3. Forum Knowledge Based Engineering, CAT-PRO, Stuttgart, Germany, Oct 9th, 2003.

- [8] Rudolph, S. *Vorlesungsunterlagen "Digital Engineering"*. Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart, 2016.
- [9] Feldhusen, J., Grote, K.-H. (Eds.). *Pahl/Beitz Konstruktionslehre*. 8., vollst. überarb. Aufl. ed. Springer Vieweg, Berlin, 2013.
- [10] Kettner, H., Schmidt, J., Greim, H.-R. *Leitfaden der systematischen Fabrikplanung*. Unveränd. Nachdr. der Ausg. 1984 ed. Hanser, München, 2010.
- [11] Grundig, C.-G. *Fabrikplanung*. 5., aktualisierte Aufl. ed. Hanser, München, 2015.
- [12] Wiendahl, H.-P., Reichardt, J., Nyhuis, P. *Handbuch Fabrikplanung*. 1. Aufl. ed. Carl Hanser Fachbuchverlag, 2010.
- [13] Schenk, M., Wirth, S., Müller, E. *Fabrikplanung und Fabrikbetrieb*. 2., vollst. überarb. und erw. Aufl. ed. Springer Vieweg, Berlin, 2014.
- [14] Suh, N.P., Cochran, D.S., Lima, P.C. *Manufacturing System Design*. CIRP Annals - Manufacturing Technology 47 (2), 627–639, 1998.
- [15] Schuh, G., Gottschalk, S., Lösch, F., Wesch, C. *Fabrikplanung im Gegenstromverfahren*. wt online 97 (4), 195–199, 2007.
- [16] Konold, P., Reger, H. *Praxis der Montagetechnik*. 2., überarb. und erw. Aufl., korrigierter Nachdr ed. Vieweg, Wiesbaden, 2009.
- [17] Bergholz, M.A., *Objektorientierte Fabrikplanung*. Dissertation, RWTH Aachen, Deutschland, 2005.
- [18] Lawson, B. *How designers think: The design process demystified*, 4th ed. ed. Elsevier/Architectural, Oxford, Burlington, MA, xii, 321, 2006.
- [19] Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H. *Konstruktionslehre*. 6. Aufl., Springer, Berlin, 2005.
- [20] Verein Deutscher Ingenieure (VDI). *Konstruktionsmethodik*. Beuth Verlag, Berlin, 1997.
- [21] Lindemann U., Ponn, J. *Konzeptentwicklung und Gestaltung technischer Produkte*. 2. Aufl., Springer, Heidelberg, 2011.
- [22] Feldmann, C. *Eine Methode für die integrierte rechnergestützte Montageplanung*. Dissertation, TU München, Deutschland, 1997.
- [23] Bracht, U., Geckler, D., Wenzel, S. *Digitale Fabrik: Methoden und Praxisbeispiele*. Springer, Berlin, 2011.
- [24] Brandis, R. *Systematik für die integrative Konzipierung der Montage auf Basis der Prinzipiöpfung mechatronischer Systeme*. Disseration, Universität Paderborn, Germany, 2014.
- [25] Zülch, G., Stock, P. (Eds.) *Integrationsaspekte der Simulation*. KIT Scientific Publ, Karlsruhe, 37-44, 2010.
- [26] Reichwein, A. *Application-specific UML Profiles for Multidisciplinary Product Data Integration*. Dissertation. Universität Stuttgart, Deutschland, 2011.
- [27] Kröplin, B., Rudolph, S. *Entwurfsgrammatiken – Ein Paradigmenwechsel?* Der Prüflingenieur 26, 34–43, 2005.
- [28] Object Management Group OMG. *OMG Unified Modeling Language: Version 2.5*, 2015.
- [29] IILS GmbH. *Design Compiler 43*. www.iils.de. Letzter Zugriff: 9.2.2017.
- [30] Rudolph, S. *On the problem of multi-disciplinary system design -and a solution approach using graph-based design languages*. 1st ACCM Workshop on Mechantronic Design, Linz, Austria, Nov 30<sup>th</sup>, 2012.
- [31] Vogel, S. *Über Ordnungsmechanismen im wissensbasierten Entwurf von SCR-Systemen*. Dissertation, Universität Stuttgart, Deutschland 2016.
- [32] Weidner, R. *Wissensbasierte Planung und Beurteilung von Montagesystemen in der Luftfahrtindustrie*. Dissertation, Helmut-Schmidt-Universität, Deutschland, 2014.

# Generierung und Anreicherung von virtuellen Inbetriebnahmemodelle durch graphenbasierte Entwurfssprachen

Markus Kiesel<sup>1</sup>, Nicolai Beisheim<sup>1</sup>, Theresa Breckle<sup>2</sup>

<sup>1</sup>Hochschule Albstadt-Sigmaringen, Jakobstraße 1, 72458 Albstadt-Ebingen

<sup>2</sup>Hochschule Ulm, Prittwitzstraße 10, 89075 Ulm

*kiesel@hs-albsig.de*

Industrie 4.0 basierte Produktionssysteme werden die Produktionsprozesse von morgen maßgeblich beeinflussen. Die Bedeutung der Informationstechnologie auf die Produktion wird wachsen und bietet somit die Möglichkeit Produktionssysteme flexibler zu gestalten. Jedoch steigt dadurch auch die Komplexität der Systeme. Um trotz allem solide Systeme entwickeln zu können wird die Simulation einzelner Aspekte bis hin zur Gesamtsimulation zwingend notwendig. Simulationen welche zur Absicherung der Software dienen werden wegen des gestiegenen Einflusses folglich vermehrt erforderlich, hier ist vor allem die virtuelle Inbetriebnahme zu nennen.

Die Hürde um die virtuelle Inbetriebnahme einzusetzen ist derzeit jedoch sehr hoch, da meist eine große Datenbank an Standardkomponenten erforderlich ist. Darüber hinaus ist der Aufbau der Modelle zusätzlich mit einem hohen Anteil an händischer Arbeit verbunden, daher wird in den meisten Fällen höchstens ein Modell aufgebaut. Falls sich anschließend die Änderungen am Systems ergeben, werden diese in der Regel nicht mehr übertragen.

Graphenbasierte Entwurfssprachen können durch die Automatisierung der Modellerstellung dabei helfen diese Hürde zu überwinden. Dabei wird ein abstraktes Modell des zu entwerfenden Produktionssystems angefertigt, welches anschließend auf die domänenspezifischen Anwendungen abgebildet werden kann. Falls Änderungen des Systems auftreten, lässt sich das domänenspezifische Modell automatisiert regenerieren.

## 1 Motivation

Die virtuelle Inbetriebnahme ist eine bewährte Methode um die Robustheit und Qualität eines Produktionssystems dauerhaft zu erhöhen. Dabei erhält der Anwender die Möglichkeit Prozesse bereits anhand digitaler Prototypen zu testen, evaluieren und zu verbessern (siehe u.a. [1], [2]).

Technologien wie Industrie 4.0 und Internet der Dinge halten zunehmend Einzug in die Produktion. Dadurch ergeben sich vielfältige neue Möglichkeiten für die Entwicklung von Produktionssystemen (vgl. [3], [4]). Jedoch steigt damit auch die Gesamtkomplexität, insbesondere Konzepte wie Dezentralisierung und verteilte Intelligenz tragen zu diesem rasanten Zuwachs bei. Folglich werden Simulationstechniken wie die virtuelle Inbetriebnahme zwingend notwendig

um den gestiegenen Softwareanteil und die dadurch entstehende Fehleranfälligkeit besser abfangen zu können. Dabei gestaltet sich die Erstellung der Modelle für die virtuelle Inbetriebnahme als überaus zeitintensiv, da dieser Prozess einen hohen Anteil händischer Arbeit erfordert. Die Konsequenz sollte daher eine möglichst automatisierter Prozess sein. Durch den Einsatz von graphenbasierten Entwurfssprachen kann eine solche Automatisierung erreicht werden.

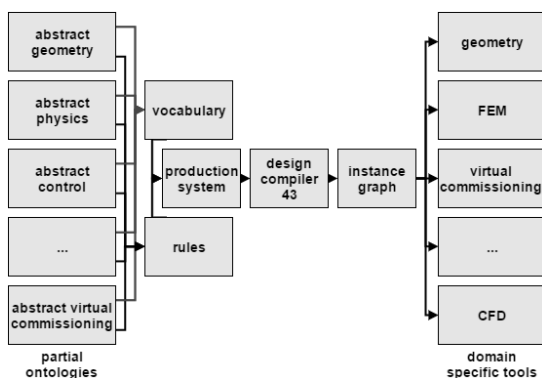
## 2 Grundlagen

Die Methodik in diesem Beitrag nutzt zwei Konzepte um die Erzeugung und Anreicherung von virtuellen Inbetriebnahmemodelle zu erreichen. Dieses Kapitel liefert einen Überblick über diese Konzepte.

## 2.1 Graphenbasierte Entwurfssprachen

Graphenbasierte Entwurfssprachen werden zur Automatisierung und Optimierung von Entwurfprozessen eingesetzt (siehe u.a. [5], [6]). Entwurfssprachen bestehen dabei aus drei Grundkomponenten:

- **Vokabular**  
Das Vokabular stellt einen Teil des abstrakten Wissens dar. Es beschreibt die verfügbaren Komponenten innerhalb einer Entwurfssprache und deren Beziehungen.
- **Regeln**  
Die Regeln sind der zweite Teil des abstrakten Wissen und dienen als Bauplan für den Entwurfprozess. Die Regeln werden dabei von einer Vielzahl an Parametern beeinflusst, was zu stark heterogenen Ergebnissen führen kann.
- **Compiler**  
Ein Compiler instantiiert den Entwurf welcher von Vokabular und Regeln beschrieben wurde. Dieses Modell kann anschließend als Meta-Modell für weitere Anwendungen verwendet werden.



**Abbildung 1:** Prozessübersicht der graphenbasierten Entwurfssprachen

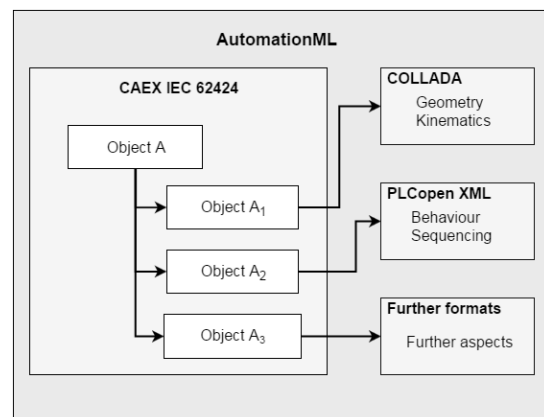
Graphenbasierte Entwurfssprachen können nahezu jeden Aspekt des zu entwerfenden Systems abdecken. Hierzu gehören beispielsweise die Geometrie eines Produktes oder der Programm-Code für das Produktionssystem. Dazu muss das abstrakte Wissen der einzelnen Teilgebiete in die Entwurfssprache eingebracht werden (linke Seite in Abbildung 1). Diese Teilgebiete können dann im Vokabular und

den Regeln verwendet werden um die entsprechenden Teilentwürfe automatisiert zu erzeugen.

In diesem Forschungsprojekt stellt der Design Compiler 43 (IILS mbH [7]) die Funktionalität des Erzeugens und Ausführens von Entwurfssprachen zur Verfügung.

## 2.2 AutomationML

Durch die steigende Komplexität von Industrie 4.0 basierten Produktionssystemen ist es erforderlich, dass Mitarbeiter unterschiedlicher Abteilungen Daten effizient austauschen können. Ein Format welches einen Austausch dieser meist heterogenen Daten ermöglicht, ist das XML-basierte Datenformat AutomationML (siehe u.a. [8], [9]). Um die Integration von AutomationML zu vereinfachen, wurden diverse offene Standards unter diesem Hüllformat vereint.



**Abbildung 2:** Überblick AutomationML

Die offenen Standards welche von AutomationML verwendet werden sind in Abbildung 2 dargestellt. Die AutomationML Datei selbst basiert auf dem CAEX Format (IEC 62424), welches nur um wenige Elemente erweitert wurde. Auf Grund der XML-Struktur ist das AutomationML Format sehr gut erweiterbar. Alle vorhanden Komponenten, die hierarchische Struktur sowie die Verbindungen zwischen den Komponenten werden in dieser CAEX Struktur abgebildet.

Der COLLADA Standard stellt die Funktionalität der Geometrierepresentation zur Verfügung. Dabei können Geometriedaten als Begrenzungsflächen- sowie Gittermodell gespeichert werden. Auch weitere Informationen wie die Kinematik oder Animationen kön-

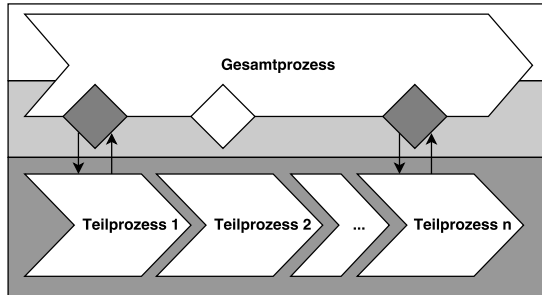


Abbildung 3: Prozessschichtenmodell

nen aufgenommen werden.

Das PLCopen XML Format wurden ebenfalls in den AutomationML Standard integriert und bietet vor allem im Bereich der virtuellen Inbetriebnahme diverse Vorteile. Da es auf der Europäischen-Norm EN 61131 basiert, können alle gängigen Sprachen für die Ablaufprogrammierstellung von speicherprogrammierbaren Steuerungen gespeichert werden. Diese Daten werden während der virtuellen Inbetriebnahme gewöhnlich mit Hardware-in-the-Loop oder Software-in-the-Loop-Systemen evaluiert.

### 3 Methodik

Die in diesem Beitrag aufgezeigte Methode zielt darauf ab, die Erstellung und Anreicherung von virtuellen Inbetriebnahmemodellen zu vereinfachen. Hierbei wird vorausgesetzt, dass bereits eine Basisstruktur in AutomationML vorliegt. Diese könnte aus einer vorangegangenen Simulation zur digitalen Fabrik stammen (z.B. Siemens Process Simulate [10]) oder bereits aus einer graphenbasierten Entwurfssprache.

#### 3.1 Grundlegender Ablauf

Um die Komplexität der einzelnen Entwurfssprachen zu reduzieren, werden diese funktional zerlegt und im späteren Gesamtprozess über das gemeinsame Datenformat integriert. Der abstrahierte Ablauf ist in Abbildung 3 dargestellt. Daraus wird ersichtlich, dass die Ansteuerung von einem übergeordneten Prozess erfolgt. Es lassen sich in der Abbildung drei Schichten erkennen:

1. Gesamtprozessschicht
2. Entscheidungsschicht

#### 3. Teilprozessschicht

Die Entscheidungsschicht kann in den Gesamtprozess integriert oder losgelöst implementiert werden. In diesem Beitrag ist die Entscheidungsschicht in den Gesamtprozess integriert, somit entscheidet dieser ob ein Teilprozess angesteuert wird.

#### 3.2 Beispiel

Um ein besseres Verständnis der Methodik zu schaffen wird das folgende Beispiel definiert.

Betrachtet wird eine Roboterzelle bestehend aus Vorrichtung, Greifer und Roboter. In der zugehörigen AutomationML Datei sind bereits die Geometriedaten, die Positionen der Komponenten, sowie Punktinformationen für die Bahnkurven des Roboters enthalten. Wie eingangs erwähnt könnten diese Informationen aus einer vorangegangenen Simulationen zur digitalen Fabrik bzw. einer weiteren Entwurfssprache stammen.

Zur Demonstration der Methodik wird exemplarisch das Teilziel der Generierung eines lauffähigen Roboterprogrammes herausgegriffen.

#### 3.3 Einlesen und Zuordnen der Daten

Um die vorliegende AutomationML Datei in einer graphenbasierten Entwurfssprache verwenden zu können, muss diese in die passende Struktur gebracht werden. Hierzu wird ein Einleseprozess gestartet, welcher die notwendigen Informationen aus der Datei liest und die entsprechenden Objekte der AutomationML-Struktur instantiiert. Die Hierarchieinformationen werden dabei ebenfalls extrahiert und in diese Struktur übertragen.

Der so entstandene Graph dient als Ausgangspunkt für die Weiterverarbeitung innerhalb der Entwurfssprache, welche die Generierung der Bahnkurven und Programme für den Roboter vornimmt.

#### 3.4 Erzeugen des Roboterprogrammes

Die Erzeugung der Bahnkurven basiert auf den zur Verfügung stehenden Punkten und 3-D-Modellen. Um eine Abschätzung der benötigten Bewegungsart und -geschwindigkeit vornehmen zu können werden folgende Rahmenbedingungen festgelegt.

- Annäherungspunkte und Umkehrpunkte werden immer mit linearen Bewegungen und reduzierter Geschwindigkeit ausgeführt.
- Punkte durch den freien Raum werden mit Punkt-zu-Punkt Bewegungen und grundsätzlich mit der maximalen Geschwindigkeit realisiert.

Die unterschiedlichen Punkte sind in Abbildung 4 dargestellt. In diesem Fall ist der Punkt dessen Umkreis das Störvolumen schneidet, sowohl ein Annäherungs- als auch, auf Grund des spitzen Winkels, ein Umkehrpunkt. Auf eine Betrachtung der resultierenden Traglast, welche aus der Kombination aus Roboter, Greifer und Werkstück hervorgeht, wird in diesem einfachen Beispiel verzichtet.

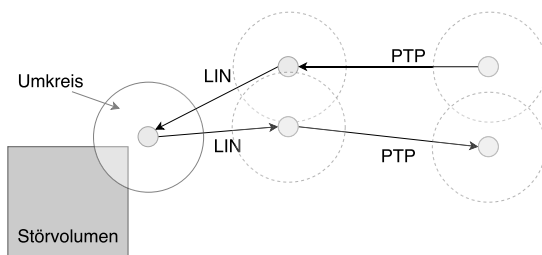


Abbildung 4: Darstellung spezifischer Punkte

Der Algorithmus welcher die Eingruppierung vornimmt bildet dabei Kugeln um die einzelnen Punkte und prüft deren Abstand zum nächsten geometrischen Objekt. Somit lässt sich eine Vorauswahl all jener Punkte bestimmen welche keine besonderen Anforderungen an das Bahnprofil stellen. Gemäß den Rahmenbedingungen werden diese durch Punkt-zu-Punkt Bewegungen realisiert.

Darüber hinaus können durch eine Analyse der Bahnkurve weitere Rückschlüsse gezogen werden. Bei Handlingsaufgaben kann beispielsweise das zusammenfallen eines Umkehr- und Annäherungspunkt auf eine Übergabeposition hindeuten.

Um die spätere Inbetriebnahme zu vereinfachen ist es sinnvoll die Programme so auszulegen, dass eine händische Korrektur möglichst einfach vorgenommen werden kann. Ungenauigkeiten welche bei der Produktion und Montage des realen Produktionssystem auftreten, können so einfacher abgefangen werden. Hierfür kann beispielsweise bei Übergabepositionen ein Sicherheitsabstand mitberechnet werden. Ein weiteres Beispiel wäre die Zuweisung aller Punkte eines kontinuierlichen Prozesses (z.B. Klebeprozess) zu einem gemeinsamen Koordinatensystem.

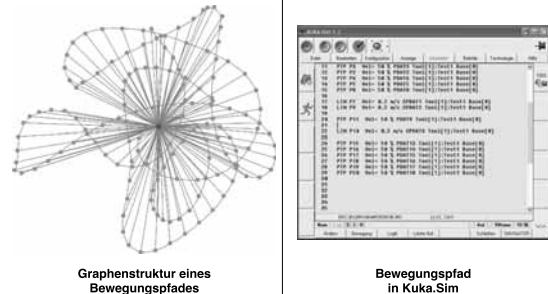


Abbildung 5: Bahnstruktur innerhalb und außerhalb der Entwurfssprache.

Die so entstehende Struktur ist bisher von keinem Robotertyp bzw. dessen Steuerung und Sprache abhängig. Um jedoch ein lauffähiges Programm zu erzeugen, muss das abstrakte Roboterprogramm anschliessend in ein konkretes und herstellerspezifisches Programm gewandelt werden. Hierzu werden die abstrakten Bahninformationen in tatsächliche Befehle übersetzt. Wie diese Bahnstruktur vor und nach einem solchen Übersetzungsvorgang aussieht kann Abbildung 5 entnommen werden.

Ein Ablaufprogramm für einen Roboter beinhaltet neben den reinen Bahninformationen zumeist noch weitere Informationen welche beispielsweise zur Interaktion mit externer Sensorik und Aktorik dienen. Ob diese Informationen ebenfalls automatisiert integriert werden können hängt stark von den zur Verfügung gestellten Daten ab, ist jedoch generell möglich.

### 3.5 Anreicherung der AutomationML Datei

Um die erzeugten Informationen nachfolgenden Prozessen zur Verfügung stellen zu können sollten diese wieder in die Ausgangsdatei zurückgeführt werden. Dabei ist darauf zu achten, nicht wahllos alle Daten zurückzuschreiben, da die Dateigröße ansonsten unnötig anwächst. Grundsätzlich sollten jedoch alle Daten die in einem nachfolgenden Prozess verwendet werden könnten auch gespeichert werden, Ergebnisse von Zwischenberechnungen oder andere temporäre Daten können jedoch vernichtet werden.

## 4 Fazit

Durch den Einsatz von graphenbasierten Entwurfssprachen kann auch klein- und mittelständischen Unternehmen die Verwendung der virtuelle Inbetriebnahme für alle Revisionsstufen ihrer Produktionssysteme ermöglicht werden. Dies wird vor allem durch den geringeren Aufwand für die Erstellung der Modelle und einer Parallelisierung der Tätigkeiten erreicht. Der einmalige Anfangsaufwand für die Erstellung der Entwurfssprachen ist zwar relativ hoch, kann jedoch über mehrere Projekte verteilt werden.

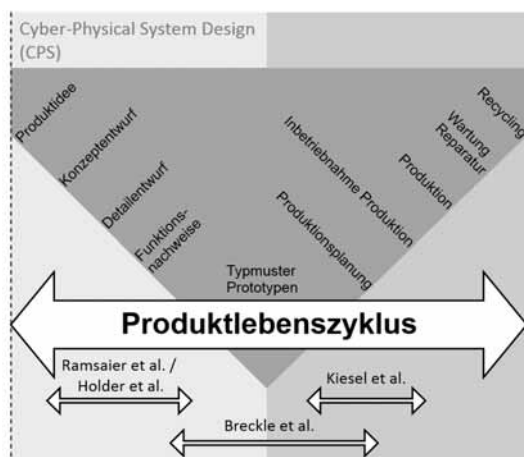


Abbildung 6: Produktlebenszyklus

Der Abdeckung des in Abbildung 6 dargestellte Produktlebenszyklus durch graphenbasierte Entwurfssprachen hat das aktuelle Forschungsprojekt „Digitaler Produktlebenszyklus (ZaFH)“ zum Ziel. Dabei soll ein durchgängiges Modell entstehen, dass eigenständig während des Entwurfs auf diverse Randbedingungen reagieren kann und somit zu einem möglichst optimalen Gesamtergebnis führt. Das System soll die dabei entstanden Entwürfe selbständig bewerten und optimieren. Dabei kann das System den Entwurfsraum eigenständig explorieren um einen möglichst optimalen Entwurf zu finden. Die Führungsgröße wird dem System dabei vorher von einem Benutzer vorgegeben.

## Danksagung

Das Projekt „Digitaler Produktlebenszyklus (ZaFH)“ (mehr Informationen unter: <http://dip.reutlingen->



university.de) wird unterstützt vom Europäischen Fond für regionale Entwicklung und dem Ministerium für Wissenschaft, Forschung und Kunst des Landes Baden-Württemberg (Mehr Informationen unter: [www.rwb-efre.baden-wuerttemberg.de](http://www.rwb-efre.baden-wuerttemberg.de)).

## References

- [1] R. Neugebauer, P. Klimant und M. Witt, *Realistic Machine Simulation with Virtual Reality*, Procedia CIRP, 2012.
- [2] J. Kiefer, T. Baer und H. Bley, *Mechatronic-oriented Engineering of Manufacturing Systems*, Proceedings of 13th CRIP International Conference on Lifecycle Engineering, 2006.
- [3] H. Kagermann, W. Wahlster und J. Helbig, *Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry*, final report of the Industrie 4.0 Working Group, Forschungsunion, 2013.
- [4] N. Beisheim und M. Kiesel, *Industrie 4.0: Neue Anwendungsfelder für die Simulation*, Forschungsreport für den Maschinenbau in Baden-Württemberg, 2014.
- [5] S. Rudolph, *Aufbau und Einsatz von Entwurfssprachen für den Ingenieurentwurf*, Forum Knowledge Based Engin., CAT-PRO, 2003.
- [6] S. Rudolph und B. Kröplin, *Entwurfsgrammatiken – Ein Paradigmenwechsel?*, Der Prüfingenieur, 26, 34-43, 2005.
- [7] DESIGN COMPILER 43, *IILS Ingenieurgesellschaft für Intelligente Lösungen und Systeme mbH*, [Online: 30.01.2017]. <http://www.iils.de/>.
- [8] D. Rainer, A. Lüder, J. Peschke und L. Hundt, *AutomationML - the glue for seamless automation engineering*, IEEE International Conference on Emerging Technologies and Factory Automation, 2008.

- [9] A. Lüder, L. Hundt und A. Keibel, *Description of manufacturing processes using AutomationML*, IEEE Conference on Emerging Technologies and Factory Automation (ETFA), 2010.
- [10] Process Simulate, *Siemens Industry Software GmbH*, [Online: 01.02.2017]. [https://www.plm.automation.siemens.com/de\\_de/products/tecnomatix/manufacturing-simulation/robotics/process-simulate.shtml](https://www.plm.automation.siemens.com/de_de/products/tecnomatix/manufacturing-simulation/robotics/process-simulate.shtml)

# Ansatz zur rechnergestützten Synthese und Analyse von Entwurfsvarianten für Formula Student Getriebe mittels graphenbasierter Entwurfssprachen

Kevin Holder<sup>1</sup>, Andreas Zech<sup>2</sup>, Ralf Stetter<sup>2</sup>, Markus Till<sup>2</sup>

<sup>1</sup> ZF Friedrichshafen AG

<sup>2</sup> Hochschule Ravensburg-Weingarten

*kevin.holder@zf.com*

Das Konzept graphenbasierter Entwurfssprachen hat sich in den letzten zehn Jahren in ein generisches Engineering Framework für die Definition und automatisierte Ausführung von computergestützten Entwurfsprozessen weiterentwickelt. In diesem Engineering Framework stellen die Ingenieurobjekte (Produktmodelle, etc.) den Wortschatz dar; die erforderlichen Modelltransformationen stellen die Regeln, d.h. die Grammatik der Designsprache, dar. Vokabeln und Regeln werden in einem sogenannten "Produktionssystem" verknüpft. Durch einen speziell entwickelten Compiler (Design Compiler 43™) werden die Regeln ausgeführt, um die Klassen mit den Vokabeln zu instanzieren. Dieser Kompilierungsprozess baut das zentrale Datenmodell auf. Von diesem zentralen Datenmodell aus generieren verschiedene Schnittstellen domänenspezifische, generische Modelle wie Geometrie-Modelle oder Simulationsmodelle. Eine zentrale Herausforderung bei der Entwicklung von Fahrzeuggetrieben ist die durchgängige Abbildung des Getriebeentwurfs ausgehend von den Fahrzeugparametern bis zur detaillierten Dimensionierung der Maschinenelemente. In diesem Artikel wird beschrieben, wie auf der Basis der wichtigsten Fahrzeugparameter ein Getriebeentwurf mittels graphenbasierter Entwurfssprachen abgebildet werden kann. Die Entwurfssprache setzt dabei die Teilschritte der Getriebesynthese mit Analyse und Bewertung um und ermöglicht das schnelle Untersuchen von Getriebevarianten. Zentrales Element stellt dabei die in UML modellierte Entwurfssprache dar.

## 1 Einleitung

Im Rahmen des Forschungsprojektes ZAFH:Digitaler Produktlebenszyklus wird untersucht, wie der Produktlebenszyklus ganzheitlich mit den Methoden der graphenbasierten Entwurfssprachen abgebildet werden kann. Dabei wurde ein Ansatz ermittelt, mit dem eine rechnergestützte Getriebeentwicklung mit Synthese und Analyse von Entwurfsvarianten deaxialer Zweiwellengetrieben durchgeführt werden kann. Diese Getriebe mit zwei parallelen Wellen finden z.B. Einsatz in Formula Student Rennfahrzeugen. In dem mit einer Entwurfssprache realisierten Ansatz werden zunächst über eine Eingabemaske die Syntheseparameter in den Design Compiler 43™ [1] (entwickelt von der IILS mbH (<http://www.iils.de>) zusammen mit der Universität Stuttgart) übergeben. Auf dieser Basis werden zentrale Produktparameter (Übersetzung, Zähnezahlen, Auslegungszahnbreiten, ...) bestimmt. Im Design Compiler (DC43) werden diese Produktparameter den entsprechenden Produktinstanzen zugewiesen. Zur Berechnung der Bewertungskriterien (Verzahnungssicherheiten) wird die FVA-Workbench 4.0 [2]

automatisiert angesprochen. Ergebnis der Entwurfsanalyse sind auf Basis von gängigen Normen berechnete Sicherheitswerte für die Verzahnung des Getriebes, die anhand definierter Grenzwerte bewertet werden. Durch die Entwurfssprache können sehr schnell neue Getriebeentwürfe detailliert ausgeprägt und bewertet werden. Für eine Einordnung des behandelten Themas in den Projektkontext ist in Abbildung 1 der Produktlebenszyklus dargestellt, sowie die für diese Konferenz eingereichten Beiträge markiert.

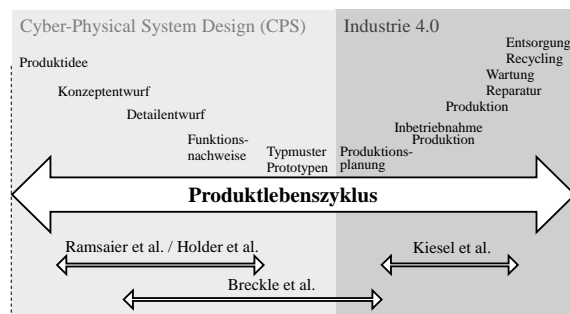
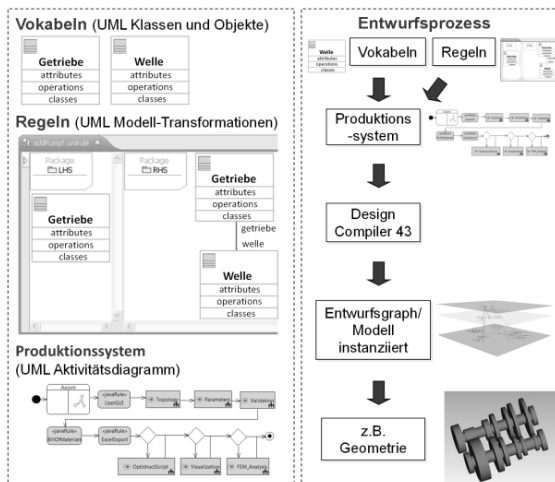


Abbildung 1. Produktlebenszyklus und Einordnung

## 2 Graphenbasierte Entwurfssprachen

Ziel des übergeordneten Forschungsvorhabens ist die ganzheitliche durchgängige digitale Beschreibung des Produktlebenszyklus. Für diese digitale Beschreibung werden graphenbasierte Entwurfssprachen eingesetzt [3], [4]. Graphenbasierte Entwurfssprachen stellen eine regelbasierte Erweiterung baukastenorientierter Konfigurationswerkzeuge dar. Im Allgemeinen ist ein Graph eine Darstellung eines Satzes von Objekten, wobei einige Objekte untereinander durch Links miteinander verbunden sein können. Die Objekte werden dabei durch mathematische Abstraktionen repräsentiert, welche "Knoten" genannt werden; die Verbindungen werden von Abstraktionen repräsentiert, welche "Kanten" genannt werden. Graphenbasiert bedeutet also, dass Knoten eines Graphen als abstrakter Platzhalter für reale Objekte dienen und stellvertretend für das Entwurfsobjekt eine Graphenrepräsentation maschinell verarbeitet wird. Dabei stellen die Ingenieurobjekte (Produktmodelle, etc.) den Wortschatz (die Vokabeln) dar; die erforderlichen Modelltransformationen stellen die Regeln, d.h. die Grammatik der Designsprache dar [5]. Abbildung 2 (rechte Seite) und Abbildung 3 bieten einen Überblick über den generischen Entwurfsprozess, welcher aus Vokabeln und Regeln ein konsistentes Modell in der Modellierungssprache UML erzeugt und damit unter anderem Simulationen und Bewertung in verschiedenen domänenspezifischen Werkzeugen erlaubt.

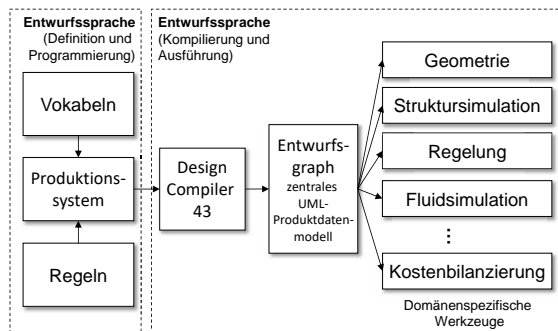


**Abbildung 2.** Repräsentation des Entwurfswissens in UML (links); Wissensverarbeitung in einer Entwurfssprache zur Getriebeentwurfsgenerierung (rechts)

Dabei sind Vokabeln UML Basis-Komponenten, z.B. die Klasse *Getriebe* selbst oder *Wellen*. Regeln sind UML Modell-Transformationen – diese zeigen jeweils

den Eingangszustand und den Ausgangszustand (siehe mittig linke Seite der Abbildung 2, links nur *Getriebe*; rechts *Getriebe* mit *Welle*). Vokabeln und Regeln werden in einem sogenannten "Produktionssystem" verknüpft. Durch den speziell entwickelten Compiler (DC43) werden die Regeln ausgeführt, um die Vokabeln zu instanziiieren. Beim Durchlauf des Programms entsteht der Entwurfsgraph, dieser entspricht dem zentralen Datenmodell. Von diesem Datenmodell aus generieren verschiedene Schnittstellen domänenspezifische, generische Modelle wie Geometrie-Modelle oder Simulationsmodelle (vgl. Abbildung 3).

Als Modellierungssprache kommt die international standardisierte Modellierungssprache Unified Modeling Language (UML) [6] zum Einsatz, da sie objektorientiert und leicht erweiterbar ist und sehr flexible grafische Notationen und Sichten erlaubt. Zudem können UML-Modelle kompiliert werden, wodurch ausführbarer Code erzeugt werden kann.



**Abbildung 3.** Wissensverarbeitung in einer Entwurfssprache

## 3 Entwurfssprache zur automatisierten Getriebeentwurfsgenerierung

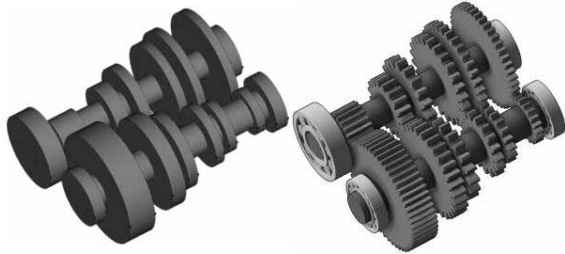
Der hier gezeigte Ansatz dient einer automatisch-rechnergestützten Getriebesynthese mit Getriebeanalyse und damit zur programmgesteuerten Konzeptfindung von Getriebevarianten. Innerhalb des Ansatzes werden folgende Prozessschritte durchlaufen:

1. Synthese; 2. Analyse; 3. Bewertung

Die betrachteten Getriebevarianten stellen deaxiale Zweiwellegentriebe dar, wie sie häufig bei Formula Student Rennfahrzeugen zum Einsatz kommen. Es wird die Getriebestruktur in Form eines Radsatzes betrachtet. Alle weiteren Getriebekomponenten wie z.B. Gehäuse und Schaltbetätigung werden zum bisherigen Entwicklungsstand der Entwurfssprache vernachlässigt. Innerhalb des ersten Prozessschritts erfolgen die Instanziierung der Getriebekomponenten, sowie eine



bes. Sowie die Visualisierung des Berechnungsmodells der FVA Workbench 4.0, welches automatisch ausgeprägt wurde (Abbildung 7, rechts).



**Abbildung 7.** Geometrische Ausprägung des Entwurfsgraphen in DC43 (links); Visualisierung des Berechnungsmodells der FVA-Workbench 4.0 (rechts)

### 3.2 Synthese von Getriebeentwürfen durch Rechenmethodik

Für die automatisierte Synthese von Getriebekonzepten müssen vom Benutzer mehrere Randbedingungen (Syntheseparameter) angegeben werden. Die Syntheseparameter liegen als Anforderungsprofil im DC43 vor und bilden grundlegende Fahrzeugeigenschaften und Kenngrößen für die Getriebeauslegung ab. Zu den Fahrzeugeigenschaften gehören unter anderem die Parameter zur Fahrwiderstandsberechnung sowie die geplante minimale und maximale Fahrzeuggeschwindigkeit. Zu den Getriebeparametern gehören z.B. der Drehmomentverlauf des Antriebsmotors, die gewünschte Anzahl der Gänge oder der Progressionsfaktor für die Übersetzungsabstufung. Die zugrunde liegende Rechenmethodik ist in einem Matlab-Programm [8] hinterlegt, welches über DC43 angesteuert wird und dem Modell die Syntheseparameter übergibt. Auf dieser Basis generiert das Matlab-Programm Produktparameter wie Übersetzung, Zähnezahlen, Auslegungszahnbreiten, usw. Die implementierte Vorgehensweise ist an [9] angelehnt und kann wie folgt zusammengefasst werden:

#### *Berechnung der Übersetzungen mit progressiver Abstufung*

Durch die DC43-Schnittstelle erfolgt der Input der Eingangsparameter, über die die Berechnung von Fahrwiderständen, Leistungsbedarf und Zugkrafthyberbel erfolgt. Darauf aufbauend werden die größte und kleinste Übersetzungsstufe berechnet, gefolgt von der Bestimmung der Zwischengänge. Dafür wird gemäß [9] die Gesamtspreizung bestimmt. Aus der berechneten Spreizung und dem gewählten Progressionsfaktor

kann der Grundstufensprung und daraus die verbleibenden Übersetzungen berechnet werden.

#### *Auslegung der Zahnräder*

Innerhalb des Matlab-Programmaufrufs werden Zähnezahlskombinationen ermittelt, die die Übersetzungsanforderungen erfüllen. Mit Hilfe derer erfolgt die Berechnung der präzisen Verzahnungsgeometrie einer jeden Gangstufe. Designkriterien bilden beispielsweise eine vordefinierte Profilüberdeckung ( $\varepsilon_\alpha > 1$ ). Abgeschlossen wird die Auslegung der Verzahnungsgeometrie durch Berechnung überschlägiger Zahnbreiten auf Basis des zu übertragenden Drehmoments und den daraus resultierenden Zahnkräften.

#### *Berechnung und Auswahl der Lager*

Nach der Definition der Verzahnungsgeometrie werden die Lagerpositionen festgelegt. Über die bekannten Verzahnungskräfte ist die Ermittlung der maximal auftretenden Lagerkräfte möglich. Dadurch kann eine iterative Lebensdauerberechnung stattfinden, die eine vorgeschriebene Ziellebensdauer für jeden Gang als Randbedingung berücksichtigt. Auf Basis der Ergebnisse wird ein Lagerauswahlprozess für jede Lagerstelle gestartet.

#### *Berechnung der Wellen*

Die zugrunde liegende Berechnung liefert Wellengeometrien (z.B. Mindestwellendurchmesser) unter Berücksichtigung diverser Festigkeitskennwerte (z.B. maximale Biege- und Torsionsmomente, Kerbfaktoren). Dabei nehmen die Wellengeometrien ebenfalls Einfluss auf den Lagerauswahlprozess.

### 3.3 Getriebeentwurfsanalyse durch programmgesteuerte Festigkeitsberechnung der Verzahnungsgeometrie

Der erste Schritt führte zu einer Synthese eines Getriebeentwurfs. Nun werden Aspekte untersucht, die in den Bereich Getriebeanalyse einzuordnen sind.

#### *Gewicht- und Kostenanalyse*

Weitere essentielle Faktoren für die frühe Phase des Produktentstehungsprozesses werden ermittelt. Beispielsweise werden auf Basis des Getriebeentwurfs die Gesamtmasse sowie die daraus resultierenden Materialkosten (konkret über das Bauteilvolumen und die Kosten der Normteile) bestimmt.

#### *Plausibilisierung*

Anhand festgelegter Kriterien wird überprüft, ob die berechneten Geometrien zueinander kompatibel sind.

Hier kann unter anderem die Untersuchung der Nenn-durchmesser von Lagerinnenring und Wellendurchmesser genannt werden.

Über die DC43-Schnittstelle erfolgt eine Rückkopplung der berechneten Daten in das UML-Modell, welche zu einer Anreicherung der Informationsdichte im Entwurfsgraph führen. Eine geometrische Ausprägung des Entwurfsgraphen liefert ein Modell, welches Abbildung 7 (links) entspricht.

#### *Validierung der Verzahnung über einen Tragfähigkeitsnachweis*

Die Getriebeanalyse erfordert einen weiteren Prozessschritt in Form einer Validierung der konkretisierten geometrischen Größen des Getriebe-Grobentwurfs. Dabei soll im Rahmen einer programmgesteuerten Festigkeitsberechnung ein Tragfähigkeitsnachweis der Verzahnung erfolgen. Dazu muss diese einen Nachweis auf Zahnfußtragfähigkeit sowie einen Nachweis der Zahnflankentragfähigkeit liefern, d.h. die Sicherheiten auf Zahnfuß- und Flankentragfähigkeit müssen oberhalb eines vordefinierten Grenzwerts (meist in Form von Sicherheitskennwerten angegeben) liegen. Der zugrunde liegende Tragfähigkeitsnachweis wird über die Berechnungsplattform FVA-Workbench 4.0 erbracht, welches über eine scriptgetriebene DC43-Schnittstelle aufgerufen wird. Dabei kann der Festigkeitsnachweis nach gängigen Richtlinien wie zum Beispiel nach ISO 6336 (2006) [10] durchgeführt werden. Alle notwendigen Inputparameter für den Festigkeitsnachweis liegen im Produktgraphen in DC43 vor und werden schnittstellenübergreifend dem Modell in der FVA-Workbench zur Verfügung gestellt. Beispielsweise sind die Stufenübersetzungen, Zähnezahlen oder Auslegungszahnbreiten genannt.

### **3.4 Bewertung der Resultate der Getriebeentwurfsanalyse**

Durch Nachrechnung und Vergleich der Verzahnungsfestigkeiten mit vordefinierten Grenzwerten kann jeder Getriebeentwurf überprüft und validiert werden. Hierfür erfolgt eine Rückkopplung der Resultate in das UML-Modell des Getriebeentwurfs über die bereits beschriebene Schnittstelle. Dadurch kann bei Bedarf eine Anpassung der Syntheseparameter erfolgen und eine erneute Iteration durchgeführt werden.

### **3.5 Diskussion**

Mit dem hier gezeigten Ansatz einer vollautomatisiert-programmgesteuerten Getriebeauslegung können

schnell die Auswirkungen von Anforderungsänderungen auf den Produktentwurf gezeigt werden. Dabei beschränken sich die Bewertungskriterien momentan auf einen Nachweis der Verzahnungssicherheit. Das Spektrum an Bewertungskriterien bzw. Untersuchungsszenarien kann erweitert werden um die Ermittlung von Domänen übergreifender Zusammenhänge. Z.B. zwischen Anzahl der Gänge und dem Gewicht des Entwurfs oder die Auswirkungen von veränderten Drehmoment- und Leistungsdaten des Antriebsmotors auf die Kosten. Bei einer programmgesteuerten Getriebe-synthese und Getriebeanalyse ist eine Vorgehensweise wie beim konventionellen Konstruktionsprozess die Basis. D.h. analog zu Konstruktionsprozessen sollten folgende Phasen bzw. Prozessschritte nach [11] im Entwurf durchlaufen werden:

1. Synthese; 2. Analyse; 3. detailliertere Optimierung;
4. Bewertung; 5. Selektion und Verwerfen; 6. Mutation und Rekombination

Der gezeigte Ansatz berücksichtigt die hier geforderten Prozessschritte nur teilweise. So bildet der Ansatz zwar die Phasen *Synthese* und *Analyse* ab, und stellt anhand definierter Bewertungskriterien (in Form eines Festigkeitsnachweises der Verzahnung) eine *Bewertung* sicher. Eine Auslegung und Optimierung hinsichtlich der bereits genannten Untersuchungsszenarien oder erweiterter Bewertungs- und Designkriterien wie z.B. Bauraumbeschränkungen ist derzeit noch nicht umgesetzt. Eine Realisierung eines Optimierungszyklus zur Entwurfsoptimierung gefolgt von *Selektions-* und/oder *Rekombinationsphasen* eines Produktentwurfs ist daher Gegenstand zukünftiger Arbeiten und soll den hier gezeigten Ansatz erweitern.

### **3.6 Ausblick**

In einer Projektarbeit [7] im Rahmen des Forschungsprojektes ZAFH:Digitaler Produktlebenszyklus wurde ein Ansatz zur rechnergestützten Synthese und Analyse von Getriebekonzepten für Formula Student Rennfahrzeuge mittels graphenbasierter Entwurfssprachen vorgeschlagen und programmiert. Das bisher gezeigte Ergebnis stellt einen Demonstrator dar und ist spezialisiert auf die Erstellung von Entwurfsvarianten deaxialer Zweiwellegengetriebe. Eine Realisierung einer automatischen, iterativen Entwurfsoptimierung anhand Bewertungs- und Designkriterien wie beispielsweise Gewicht, Bauraum und Kosten wird zukünftig untersucht. Darüber hinaus wird daran gearbeitet, wie weitere Teilaspekte der Getriebeentwicklung wie z.B.

ein Getriebegehäuseentwurf in den Ansatz implementiert werden können. Grundsätzlich bedingt die Integration weiterer Teilaspekte in die Grobauslegung eine Abbildung von Entwicklungsschleifen bzw. Mehrfachdurchführungen bestimmter Entwicklungsschritte im System. Neben der Notwendigkeit einer Einbindung ist zudem die Interaktion der Teilsysteme Gegenstand zukünftiger Forschung. In der derzeitigen Implementierung ist die Entwurfsabfolge (Phasen bzw. Prozessschritte) starr festgelegt. Die Festlegung auf die Reihenfolge bei der Auslegung (Verzahnung, Wellen und Lager) entspricht dabei der Vorgehensweise, wie sie bei der manuellen Getriebeauslegung üblich ist. Wird dabei nach der Analyse eines Teilsystems der Gesamtentwurf für nicht ausreichend befunden, so kommt es bei der manuellen Getriebeauslegung zu Iterationen in Form von Veränderungen eines vorhergehenden Teilsystems (Entwicklungsschleifen). Als Beispiel kann genannt werden, dass eine Lagerauslegung für eine vorgegebene Position und Welle keine zufriedenstellenden Lagerlebensdauern aufweist. Grund hierfür kann ein zu kleiner Einbauraum oder zu hohe Lagerkräfte sein. Eine neue Festlegung der Lagerposition fordert demzufolge eine neue Wellenberechnung. Diese Tatsache der sich gegenseitigen Beeinflussung von Entwurfsteilsystemen ist bisher in der gezeigten Entwurfssprache nicht berücksichtigt. In [12] ist ein Ansatz beschrieben, wie auf Basis der Dimensionalität der Teilsysteme eine geeignete sequenzielle Entwurfsabfolge ermittelt werden kann.

## 4 Danksagung

Die beschriebenen Arbeiten im Rahmen des ZAFH „Digitaler Produktlebenszyklus“ (Information unter <https://dip.reutlingen-university.de/>) werden gefördert mit Mitteln des Europäischen Fonds für regionale Entwicklung (EFRE) und des Landes Baden-Württemberg (Information unter: [www.rwb-efre.baden-wuerttemberg.de](http://www.rwb-efre.baden-wuerttemberg.de)). Dank gebührt auch Frau Jana Klein, Herr Ralph Eichheimer, Herr Johannes Kugler und Herr Dominik Lutz für die Durchführung der Projektarbeit in diesem Gebiet.

## 5 References

- [1] IILS. Ingenieurgesellschaft für intelligente Lösungen und Systeme mbh, Albstraße 6, 72818 Steinhilben.
- [2] FVA GmbH, Lyoner Straße 18, 60528 Frankfurt am Main
- [3] Rudolph, S.: *On the problem of multi-disciplinary system design - and a solution approach using graph-based design languages*. 1st ACCM Workshop on Mechanronic Design, Linz, November 30, 2012.
- [4] Till, M.; Stetter, R.; Rudolph, S.: *Multi-disziplinäre digitale Repräsentation des Produktlebenszyklus auf der Basis graphenbasierter Entwurfssprachen*. In: Forschungsreport für den Maschinenbau in Baden-Württemberg 2016. Bingen: Public, 2016. S. 3 – 6. ISSN 2196-8659.
- [5] Groß, J.; Rudolph, S.: *Generating simulation models from UML – a FireSat example*. In: Proceedings of the 2012 Symposium on Theory of Modeling and Simulation – DEVS Integrative M&S Symposium. San Diego: Society for Computer Simulation International, 2012.
- [6] Object Management Group: UML Superstructure Specification. 2.5. 2015.
- [7] Kugler, J.; Klein, J.; Eichheimer, R.; Lutz, D.: *Getriebeentwurf mittels graphenbasierter Entwurfssprache am Beispiel eines Stirnradgetriebes*. Projektarbeit, Hochschule Ravensburg-Weingarten, 2016
- [8] MathWorks, 3 Apple Hill Drive, Natick, Massachusetts 01760 USA
- [9] Naunheimer, H.: *Fahrzeuggetriebe - Grundlagen, Auswahl, Auslegung und Konstruktion*. 2. überarbeitete Auflage, Berlin Heidelberg: Springer-Verlag, 2007
- [10] ISO 6336:2006 Calculation of load capacity of spur and helical gears - Part 1; Part 2; Part 3
- [11] Stangl, M.: *Methodik zur kinematischen und kinetischen Berechnung mehrwelliger Planeten-Koppelgetriebe*. Institut für Maschinen und Fahrzeugtechnik – Lehrstuhl für Maschinenelemente, Technische Universität München, Deutschland, 2006.
- [12] Vogel, S.: *Über Ordnungsmechanismen im wissenschaftlichen Entwurf von SCR-Systemen*. Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart, Deutschland, 2016, ISBN 978-3-942807-03-6.

# Modellierung und Simulation eines Formula Student Rahmens mittels graphenbasierter Entwurfssprachen

Manuel Ramsaier, Ralf Stetter, Markus Till

Hochschule Ravensburg-Weingarten

manuel.ramsaier@hs-weingarten.de

Graphenbasierte Entwurfssprachen sind eine neue Sichtweise auf die ganzheitliche Beschreibung von Produkten, die sich am Aufbau natürlicher Sprachen orientiert, in welcher Vokabeln und Regeln eine Grammatik bilden. In Verbindung mit graphenbasierten Entwurfssprachen steht mittlerweile ein leistungsfähiges Engineering-Framework zur Verfügung, das auf maschinelle Abarbeitung und Wiederverwendung von Entwurfs- und Fertigungswissen abzielt und den Produktentwickler durch automatische Modellgenerierung und Berechnungsautomation von Routinearbeiten entlastet. Der Beitrag beschreibt den Einsatz einer solchen Entwurfssprache für die Auslegung eines Rahmens für ein Formula Student Rennfahrzeug. Im Reglement der Formula Student werden gewisse Lastfälle vorgegeben. Durch die Abbildung der Rahmengengeometrie in einer Entwurfssprache ist es möglich, automatisiert einen digitalen Funktionsnachweis zu führen. Im vorliegenden Projekt wurde ein bestehender Rahmen in einer Entwurfssprache abgebildet und automatisiert mittels Finiter Elemente Methode (FEM) und Mehrkörpersimulation (MKS) hinsichtlich der Lastfälle im Reglement untersucht. Ein großer Vorteil der Implementierung eines Modells in der Entwurfssprache besteht in der Flexibilität: Die Knotenpunkte des Rahmens können annähernd beliebig im Raum verschoben werden, die Rohrtopologie kann geändert werden und generelle Kennwerte wie Torsions- und Biegesteifigkeit können automatisiert neu berechnet werden. Somit ist eine automatisierte Exploration des Entwurfsraums möglich.

## 1 Einleitung

Im Rahmen des Forschungsprojektes ZAFH:Digitaler Produktlebenszyklus wird untersucht, wie der Produktlebenszyklus ganzheitlich mit den Methoden der graphenbasierten Entwurfssprachen abgebildet werden kann. Der Rahmen des Formula Student Rennwagens dient als Grundlage für diese Arbeit, welche die frühen Phasen des Produktlebenszyklus abdeckt, in denen ein Konzept erstellt und durch Funktionsnachweise bewertet wird. Für eine Einordnung des behandelten Themas in den Projektkontext, ist in Abbildung 1 der Produktlebenszyklus dargestellt, sowie die für diese Konferenz eingereichten Beiträge markiert.

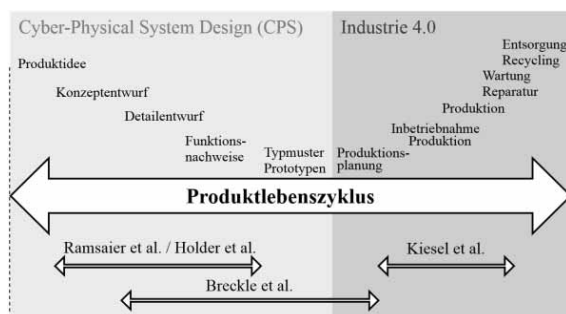


Abbildung 1. Produktlebenszyklus und Einordnung

Kurze Iterationszyklen sind in der Formula Student und allgemein im Motorsport die Regel. Bei Formula Student wird mit jedem Rennwagen ein neuer Rahmen entworfen und bezüglich seiner Leistungsfähigkeit verbessert. Dabei werden sowohl Heuristiken als auch numerische Methoden für den Funktionsnachweis des Rahmens eingesetzt. Sollte der Rahmen den im Reglement geforderten Kriterien nicht genügen, kann dies zur Disqualifikation führen. Diese Arbeit zeigt, wie mit Hilfe graphenbasierter Entwurfssprachen unter Verwendung des Engineering Frameworks DC43 [1] die Entwurfszeit für die Modellierung und Simulation des Rahmens des Rennwagens „Stinger 16“ reduziert werden kann. Dabei liegt der Schwerpunkt auf der automatisierten Abbildung der im Reglement der Formula Student geforderten Lastfälle. Der Rest dieser Arbeit gliedert sich wie folgt: In Abschnitt 2 wird das Konzept der graphenbasierten Entwurfssprachen näher erläutert. Abschnitt 3 stellt das Formula Student Team Weingarten kurz vor. Abschnitt 4 beschreibt den Aufbau der Entwurfssprache, Abschnitt 5 das Simulationsmodell, woraufhin die Potentiale in Abschnitt 6 diskutiert werden und ein Ausblick auf zukünftige Arbeitsfelder gegeben wird.

## 2 Graphenbasierte Entwurfssprachen

Das Konzept graphenbasierter Entwurfssprachen hat sich in den letzten zehn Jahren in eines der führenden generischen Engineering Frameworks für die Definition und automatisierte Ausführung von computergetstützten Entwurfsprozessen im Bereich Entwurf komplexer Systeme (Cyber-Physical Systems) weiterentwickelt [1], [3]. In diesem Engineering Framework stellen die Ingenieurobjekte (Produktmodelle, etc.) den Wortschatz (die Vokabeln) dar; die erforderlichen Modelltransformationen stellen die Regeln, d. h. die Grammatik der Entwurfssprache dar [4]. Abbildung 2 bietet einen Überblick über den generischen Entwurfsprozess, welcher aus Vokabeln und Regeln ein konsistentes Modell in der Modellierungssprache Unified Modeling Language (UML) erzeugt und damit unter anderem Simulationen und Bewertungen in verschiedenen domänenspezifischen Werkzeugen erlaubt.

Die Informatik ist bekannt für ihre sehr schnellen Entwicklungszyklen. Diese erfordern für den Softwareentwicklungsprozess sehr leistungsfähige abstrakte Beschreibungsformen (z. B. moderne graphische Modellierungssprachen wie die UML) und stark automatisierbare Entwicklungswerkzeuge, um diese schnell getakteten Innovationszyklen zu erreichen. Die UML ist eine international standardisierte Modellierungssprache, die objektorientiert und leicht erweiterbar ist sowie sehr flexible grafische Notationen und Sichten erlaubt [5]. Der Hauptverdienst der UML ist die Integration und Standardisierung eines bestimmten Bereichs des Wissens und die Auswahl und Dokumentation der verschiedenen Modellierungskonzepte [6]. Ein entscheidender Vorteil ist die Möglichkeit, UML-Modelle zu kompilieren und damit ausführbaren Code zu erzeugen (Stichwort: „Executable UML“). Diese Eigenschaft ist ein weiterer Grund für die Entscheidung, UML als Basis für graphenbasierte Entwurfssprachen zu verwenden.

Dabei sind Vokabeln UML-Basis-Komponenten, z. B. das Fahrzeug (*Car*) selbst oder sein Rahmen (*Frame*). Regeln sind UML Modell-Transformationen – diese zeigen jeweils auf der linken Seite den Eingangszustand (nur *Car*) und auf der rechten Seite den Ausgangszustand (*Car* mit verbundenem *Frame* – siehe Bild 1). Durch den Ingenieur entsteht das UML-Modell, welches das Programm in einen Entwurfsgraph kompiliert, aus dem beispielsweise die Geometrie des Produktes abgeleitet werden kann.

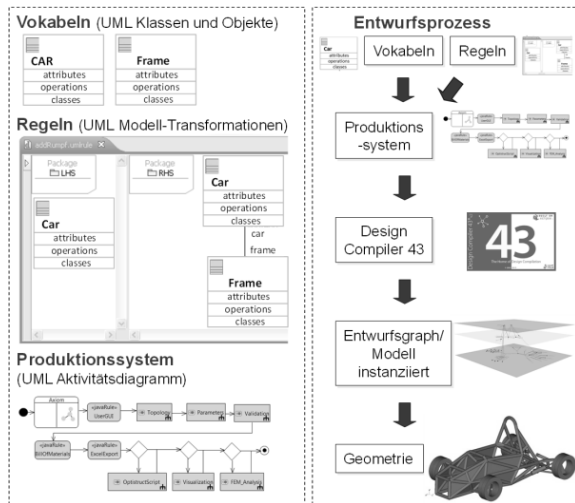


Abbildung 2. Elemente und Entwurfsprozess

Vokabeln und Regeln werden in einem sogenannten „Produktionssystem“ verknüpft. Durch einen speziell entwickelten Compiler (Design Compiler 43 [1]) werden die Regeln ausgeführt, um die Klassen/Vokabeln zu instanziiieren. Dieser Kompilierungsprozess baut das zentrale Datenmodell auf. Von diesem zentralen Datenmodell aus generieren verschiedene Schnittstellen domänenspezifische, generische Modelle wie Geometrie-Modelle oder Simulationsmodelle, dargestellt in Abbildung 3.

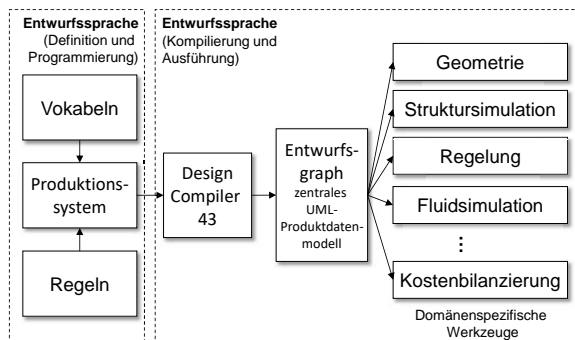


Abbildung 3. Einsatz der Entwurfssprache

## 3 Formula Student Team Weingarten

Das Formula Student Team Weingarten besteht seit 2009 und ist in den letzten Jahren immer erfolgreicher geworden. Auf der Weltrangliste der Verbrennungsteams in der Formula Student liegt das Team nach der Saison 2016 auf Platz 25 von 551 gewerteten Teams weltweit und auf Platz sieben der 70 Deutschen Teams. Das Fahrzeug Stinger16 war das leichteste Fahrzeug mit Vierzylindermotor im gesamten Feld. Für die Sai-

son 2017 ist das Team an weiteren Gewichtseinsparungen im Bereich des Rahmens interessiert, ohne dabei verringerte Torsions- und Biegesteifigkeiten in Kauf zu nehmen. Für diese Zielsetzung werden graphenbasierte Entwurfssprachen eingesetzt.

## 4 Entwurfssprache Rahmen

Wie in Abschnitt 2 beschrieben, besteht eine Entwurfssprache aus Vokabeln und Regeln, welche in der UML den Klassen und Aktivitäten entsprechen. Die Klassen werden im Klassendiagramm miteinander in Verbindung gesetzt, organisiert und in Regeln verwendet. Des Weiteren ist es möglich, Klassen aus anderen Klassendiagrammen zu verwenden und so die Funktionalität der eigenen Entwurfssprache zu erweitern. Im Folgenden wird das Klassendiagramm und der Aufbau der Entwurfssprache Rahmen vorgestellt

### 4.1 Klassendiagramm

Man unterscheidet in der UML mehrere Arten von Verbindungen zwischen Klassen. Exemplarisch dafür werden hier Vererbungen („ist ein“-Beziehungen) und Assoziationen („hat ein“-Beziehungen) diskutiert.

Wie in Abbildung 4 dargestellt, „ist“ ein *Rohr* ein *Component* aus dem Geometrie-Klassendiagramm, dessen Klassen durch ein Verknüpfungssymbol in der rechten oberen Ecke gekennzeichnet sind. Ein *Rohr* „hat“ des Weiteren zwei *Punkte*, einen „rohrEndPoint“ und einen „rohrStartPunkt“. Die *Punkte* wiederum erben von der Klasse *Point* und haben die Eigenschaften *x*, *y* und *z* mit einem Standardwert von „0“. Zum Schluss besitzt die Klasse *Rohr* noch eine *RohrEigenschaft*, welche die Information über Innen- und Aussendurchmesser beinhaltet.

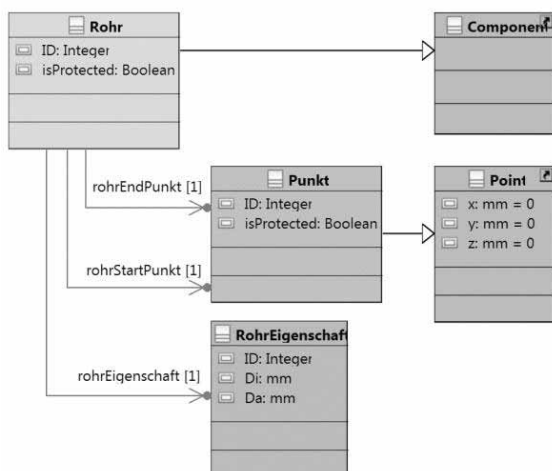


Abbildung 4. Klassendiagramm Rahmen (Auszug)

### 4.2 Produktionssystem und Regeln

Während das Klassendiagramm alle möglichen Kombinationsarten zwischen den Klassen zeigt, werden in den Regeln Kombinationen aus Instanzen der Klassen spezifiziert.

Wenn beispielsweise im Klassendiagramm festgelegt ist, dass ein *Fahrwerk* *Räder* hat, so wird in den Regeln spezifiziert, welche und wie viele. Die Regeln können dabei sowohl grafisch definiert werden, als auch über sog. „JavaRules“, in denen die UML-Transformationen durch Java Code erzeugt werden. Ein Beispiel für eine grafische Regel zeigt Abbildung 5.

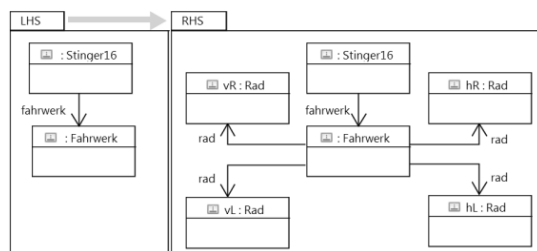


Abbildung 5. Grafische Regel (exemplarisch)

Im Folgenden wird der Prozess im Produktionssystem der Entwurfssprache, sowie die darin verwendeten Regeln beschrieben. Der Prozess kann grob in zwei Schritte aufgeteilt werden. Zum einen wird die Geometrie des Rahmens abstrakt als Graph aufgebaut, zum anderen muss die Geometrie automatisiert in ein Simulationsmodell transferiert werden. Der erste Schritt wird bis zur Regel createStepFiles, dargestellt in Abbildung 6, ausgeführt. Dabei wird zunächst in der Regel Axiom eine Instanz des Formula Student Rennwagens erstellt. In der nachfolgenden JavaRule werden die Koordinaten der Verbindungspunkte aus einer .csv Datei eingelesen. Diese dienen in den nachfolgenden vier Regeln als Grundlage für die Erstellung von Rohren, dem Überrollbügel und den Versteifungsplatten. Schlussendlich wird die erzeugte Geometrie im .step Format exportiert, um sie später für den Funktionsnachweis im Preprocessor verwenden zu können.

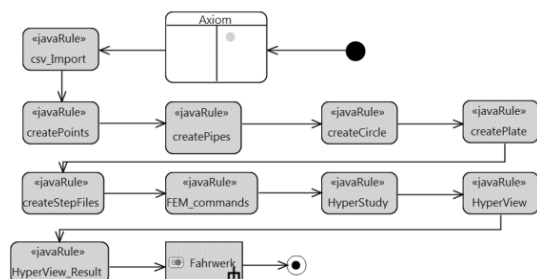
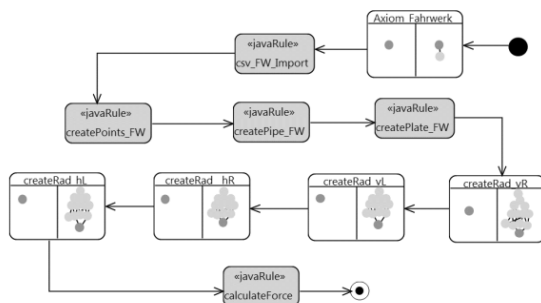


Abbildung 6. Produktionssystem der Entwurfssprache

Als Ausgangsbasis für dieses Vorhaben wird die vorhandene Geometrie des Rahmens verwendet, speziell der des „Stinger 16“ der Hochschule Ravensburg-Weingarten. Mithilfe der Geometrieinformation, nämlich den Koordinaten der *Punkte* des Rahmens sowie der Rohrdurchmesser und der Definition von Versteifungsblechen, kann der Rahmen beschrieben werden. Innerhalb der Entwurfssprache werden Instanzen für Rohre und Versteifungsbleche erstellt und die Eigenschaften der ausgelesenen Daten zugewiesen. Vom so entstandenen Graph kann die Geometrie abgeleitet werden. Für die Krafteinleitung in den Rahmen ist das Fahrwerk zentraler Baustein. Dieses wird in einem SubProgramm modelliert, welches in Abbildung 7 dargestellt ist.



**Abbildung 7.** SubProgramm Fahrwerk

Zunächst wird an die bestehende Instanz des Formula Rennwagens *Stinger16* das *Fahrwerk* im Axiom hinzugefügt. In der Regel *csv\_FW\_Import* werden die Koordinaten der Fahrwerkspunkte ausgelesen und in der nachfolgenden Regel *createPoints\_FW* als Instanzen der Klasse *Punkt* erstellt. Aus den *Punkten* werden in der Regel *createPipe\_FW* *Rohre* instanziiert. Gleichermaßen werden *Platten* in der Regel *createPlate\_FW* erstellt. Es folgen vier grafische Regeln, welche die *Räder* und deren geometrische Repräsentation erstellen. Um die Kräfte des Fahrwerks zu berechnen, wird in der letzten Regel ein Excel-Berechnungsblatt angesteuert und die Werte des erstellten Rennwagens eingetragen. Ein vereinfachtes MKS-Modell wurde in einer separaten Entwurfssprache aufgebaut und getestet.

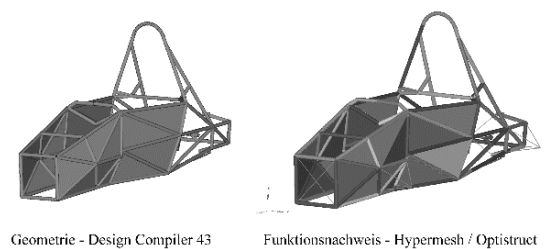
Um einen Funktionsnachweis für den Entwurf eines Rahmens zu erhalten, benötigt man neben der Geometrie die Definition der verwendeten Materialien, der Element-Diskretisierung und der Lastfälle. Diese Informationen werden zunächst ebenfalls wie die Geometrie abstrakt modelliert und sind Teil des Entwurfsgraphen. Für die Modellierung des Funktions-

nachweises wird der Preprocessor Hypermesh (Fa. Altair, Böblingen) verwendet, welcher über eine TCL/TK Schnittstelle ansprechbar ist. Um das TCL-Skript zu erzeugen, wird der Entwurfsgraph durchlaufen und relevante Informationen für die Modellierung ausgelesen. Dabei kann auf eine bestehende Arbeit für die Auslegung eines Multicopters zurückgegriffen werden. [8]

## 5 Simulationsmodell

In diesem Abschnitt wird das Simulationsmodell für den Funktionsnachweis beschrieben. Als Basis für die Lastfälle diente das Reglement der FSAE [9], welches im Kapitel AF (Alternative Frame Rules) statische Lastfälle vorgibt, welche für die Bewertung des Rahmens herangezogen werden. Zudem wurden weitere Lastfälle mittels heuristischen Methoden implementiert, welche das Team in Weingarten zur internen Bewertung einer Konstruktion heranzieht.

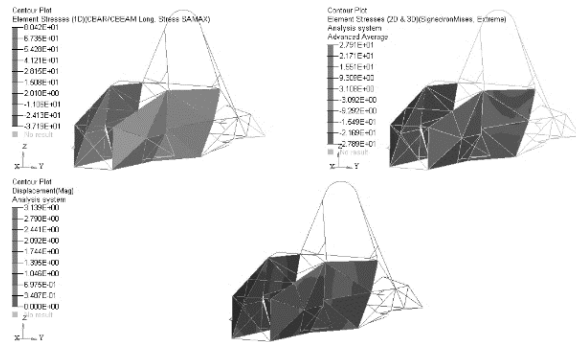
Für das Simulationsmodell sind die Rohrelemente durch 1D-Elemente diskretisiert. Aufgrund der fehlenden Notwendigkeit, beliebige (asymmetrische und über die Länge veränderliche) Querschnitte abbilden zu müssen, wurden pbar-Elemente verwendet. Diese garantieren eine schnelle Rechenzeit bei hoher Genauigkeit bezüglich den Verschiebungen und Spannungen. Eine Modellierung mit zweidimensionalen Elementen auf Mittelflächen würde zu keiner Verbesserung der Genauigkeit der Ergebnisse führen. Die Platten sind mit 2D-Elementen diskretisiert und mittels unendlich steifen Federn an die Gesamtstruktur angebunden. Die Krafteinleitung erfolgt mit RBE2-Elementen, welche das Fahrwerk abbilden.



**Abbildung 8.** Geometrie-, und Hypermesh Modell

Die Auswertung der durchgeführten Simulation erfolgt automatisiert in Hyperview. Ausgewertet werden Verschiebungen, sowie die vonMises Spannungen in den Platten und die Spannungen in den Randfasern der Stäbe. Beispielhaft ist eine Auswertung in Abbildung 8 dargestellt.

Für das Fahrwerk ist zukünftig die Ansteuerung eines MKS-Systems wie beispielsweise Adams (Fa. MSCSoftware, München) geplant, welches die Kraftberechnungen automatisiert ausführt und damit ein weiteres domänenspezifisches Modell aus dem Entwurfsgraph ableitet.



**Abbildung 9.** Auswertung eines Lastfalls in Hyperview

Die in Hyperview ermittelten Ergebnisse werden beim Aufruf des Postprocessing wiederum zurück in den Entwurfsgraph geschrieben und als .txt Datei gespeichert. So kann der Entwurf direkt in einem nachfolgenden Schritt innerhalb der Entwurfssprache bewertet werden. Die Bewertung stellt den letzten Schritt innerhalb der Produktentwicklung bis zur Erstellung von Prototypen dar. Aus der Bewertung können Kriterien zur Veränderung der einzelnen Entwurfsparameter ermittelt werden, um den Rahmen weiter zu optimieren.

## 6 Diskussion

Gezeigt wurden die Ergebnisse einer Projektarbeit im Rahmen des Forschungsprojektes ZAFH: Digitaler Produktlebenszyklus.

Dabei wird ein ganzheitliches UML-Modell des Rahmens und des Fahrwerks eines Formula Student Rennwagens erstellt und mit einem digitalen Funktionsnachweis bewertet. Die Modellierung erfolgt mit graphenbasierten Entwurfssprachen, welche einen neuartigen Ansatz der Sichtweise auf die Produktentwicklung darstellen und zentrales Thema des vorgestellten Forschungsprojektes sind. Durch diese neue Art der Produktmodellierung ist es möglich, digitale Funktionsnachweise automatisiert zu generieren und zu bewerten. Die Lastfälle des Reglements und weitere durch heuristische Methoden bestimmte Lastfälle sind in der Entwurfssprache implementiert.

Der Prozess bis hin zur Auswertung der relevanten Größen ist von der Definition der Verbindungspunkte,

der Rohre und Versteifungsbleche in einer .csv Datei vollautomatisiert. Durch die automatische Ausführung mittels des Entwurfscompilers verkürzt sich die benötigte Dauer für einen vollständigen Entwurf inklusive des strukturellen Funktionsnachweises auf die Laufzeit der einzelnen Programme. Dadurch sind mehr Entwürfe in gleicher Zeit möglich, was zu einer besseren Kenntnis des eigenen Entwurfsraums führt. Durch die bessere Kenntnis des Entwurfsraums besteht die höhere Wahrscheinlichkeit näher am globalen Optimum zu sein und damit einen Rundenzeitvorteil zu haben, was das Ziel des Formula Student Wettbewerbs darstellt.

Das Ergebnis dieser Arbeiten stellt somit eine ideale Basis für kommende Optimierungsaufgaben dar. Es können sowohl die Durchmesser der Rohre verändert werden, als auch die Position der Verbindungsknoten. Der entscheidende Vorteil der vorgestellten Methode mit graphenbasierten Entwurfssprachen gegenüber einer konventionellen Parameteroptimierung mittels eines impliziten Simulationsmodells liegt dabei in der Möglichkeit, auch große topologische Änderungen, wie das Wegnehmen oder Hinzufügen von Verstrebungen, verarbeiten zu können. Zudem ist durch die implementierten Regeln gewährleistet, dass nur jene Rohre verändert werden, welche nach dem Reglement auch veränderbar sind. Auch aus dem Reglement entstammende Grenzen für Rohrwandstärken können in den Entwurf als Randbedingung aufgenommen werden. Damit ist ein reglementkonformer Entwurf gewährleistet.

Die Entwurfssprache ist in mehrerer Hinsicht ausbaufähig. So könnten Fertigungsverfahren sowie Kosten implementiert werden, um den geforderten Cost Report für den Rahmen und schlussendlich für das ganze Formula Student Rennfahrzeug automatisiert generieren zu können bzw. die Ergebnisse des Cost Reports als (weiteres) Zielkriterium in die Optimierung mit einfließen zu lassen. Des Weiteren wäre es denkbar, die Modellierung des Fahrwerks dahingehend zu erweitern, dass Rundenzeiten auf Basis der Konfiguration ausgerechnet und mit realen Testläufen verglichen werden können.

Die Entwurfssprache ist auf einen Rahmen mit Rohrgeometrie beschränkt. Ein Ausbau hinsichtlich einer Monocoque-Bauweise kann verfolgt werden, ist für das Formula Student Team Weingarten von geringerer praktischer Relevanz, da auf absehbare Zeit kein Monocoque für den Rennwagen „Stinger“ gefertigt werden wird.

Die in dieser Arbeit gezeigte Abbildung des Rahmens ist ein wichtiger Teilschritt hin zur gesamthaften Abbildung des Formula Student Rennwagens mittels graphenbasierter Entwurfssprachen. Letzteres ermöglicht eine ganzheitliche, interdisziplinäre Optimierung.

## 7 Danksagung



Die beschriebenen Arbeiten im Rahmen des ZAFH „Digitaler Produktlebenszyklus“ (Information unter <https://dip.reutlingen-university.de/>) werden gefördert mit Mitteln des Europäischen Fonds für regionale Entwicklung (EFRE) und des Landes Baden-Württemberg (Information unter: [www.rwb-efre.baden-wuerttemberg.de](http://www.rwb-efre.baden-wuerttemberg.de)). Dank gebührt auch Frau Christin und den Herren Birk, Dittberner, Pollok, Rehm und Weing für die Durchführung einer Projektarbeit auf diesem Gebiet.

## 8 References

- [1] IILS. Ingenieurgesellschaft für intelligente Lösungen und Systeme mbh, Albstraße 6, 72818 Steinhilben.
- [2] Rudolph, S.: *On the problem of multi-disciplinary system design - and a solution approach using graph-based design languages*. 1st ACCM Workshop on Mechantron Design, Linz, November 30, 2012.
- [3] Till, M.; Stetter, R. und Rudolph, S.: *Multi-disziplinäre digitale Repräsentation des Produktlebenszyklus auf der Basis graphenbasierter Entwurfssprachen*. In: Forschungsreport für den Maschinenbau in Baden-Württemberg 2016. Bingen: Public, 2016. S. 3 – 6. ISSN 2196-8659.
- [4] Groß, J. und Rudolph, S.: *Generating simulation models from UML – a FireSat example*. In: Proceedings of the 2012 Symposium on Theory of Modeling and Simulation – DEVS Integrative M&S Symposium. San Diego: Society for Computer Simulation International, 2012.
- [5] Object Management Group: UML Superstructure Specification. 2.5. 2015.
- [6] Störrle, Harald; UML 2 für Studenten. München: Pearson 2005.
- [7] Birk et al., *Abstrakte Modellierung eines Rennwagen-Rahmens mit graphenbasierten Entwurfssprachen*. Projektarbeit. Hochschule Ravensburg-Weingarten, 2016
- [8] Ramsaier, M.; Spindler, C.; Stetter, R.; Rudolph, R. und Till, M.: *Digital representation in multicopter design along the product life-cycle*. Eingereicht für die Konferenz CIRP ICME 16, Ischia, 20-22 Juli, 2016
- [9] 2017-18 Formula SAE Rules, [www.fsaeonline.com](http://www.fsaeonline.com), 23.01.2017
- [10] Kiesel, M.; Beisheim, N. und Breckle, T.: *Automatisierte Generierung von virtuellen Inbetriebnahmmodellen auf Basis graphenbasierter Entwurfssprachen*. Eingereicht für die Konferenz ASIM/GI, Ulm, März, 2017
- [11] Breckle, T.; Kiefer, J.; Kiesel, J. und Manns, M.: *Konzeptplanung von Montagesystemen mit graphenbasierten Entwurfssprachen*. Eingereicht für die Konferenz ASIM/GI, Ulm, März, 2017
- [12] Holder, K.; Zech, A., Stetter, R. und Till, M.: *Ansatz zur rechnergestützten Synthese und Analyse von Entwurfsvarianten für Formula Student Getriebe mittels graphenbasierter Entwurfssprachen*. Eingereicht für die Konferenz ASIM/GI, Ulm, März, 2017

# Aussteller

## **LTX Simulation GmbH**

Wohlfahrtstraße 21b  
80939 München  
[www.ltx.de](http://www.ltx.de)

## **ANSYS Germany GmbH**

Staudenfeldweg 20  
83624 Otterfing  
[www.ansys.com/](http://www.ansys.com/)

## **Siemens Industry Software GmbH**

Otto-Hahn-Ring 6  
81739 München  
[www.siemens.com](http://www.siemens.com)

## **CENIT AG**

Industriestraße 52-54  
70565 Stuttgart  
[www.cenit.com](http://www.cenit.com)

## **Maplesoft Europe GmbH**

Auf der Hülz 198  
52068 Aachen  
[www.maplesoft.com](http://www.maplesoft.com)

