

# ASIM-Treffen STS/GMMS 2013

Treffen der ASIM/GI-Fachgruppen STS und GMMS

28. Februar - 1. März 2013, Düsseldorf

Tagungsband

Roland Reichardt (Hrsg.)



Arbeitsgemeinschaft Simulation ASIM  
in der Gesellschaft für Informatik GI

Fachgruppe Simulation technischer Systeme STS

Fachgruppe Grundlagen und Methoden in  
Modellbildung und Simulation GMMS



Fachhochschule Düsseldorf  
Fachbereich Maschinenbau und Verfahrenstechnik

ASIM MITTEILUNG AM145

\*

ARGESIM REPORT AM 41

ISBN ebook 978-3-901608-80-3 \* DOI 10.11128/arep.41 \* ISBN print 978-3-901608-41-4





**ARGESIM**

## **Kontaktadresse**

Fachhochschule Düsseldorf  
Fachbereich Maschinenbau und Verfahrenstechnik  
Josef-Gockeln-Straße 9  
40474 Düsseldorf  
  
Tel: +49 211 4351-400  
Fax: +49 211 4351-403  
  
[www.mv.fh-duesseldorf.de](http://www.mv.fh-duesseldorf.de)

Die Autoren sind für den Inhalt der Beiträge dieses Tagungsbandes verantwortlich.

Alle Rechte vorbehalten

Dieses Werk ist einschließlich aller seiner Teile urheberrechtlich geschützt. Jede Verwertung, die über die engen Grenzen des Urheberrechtsgesetzes hinausgeht, ist ohne schriftliche Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Speicherung in elektronischen Systemen.

Die Wiedergabe von Warenbezeichnungen und Handelsnamen in diesem Buch berechtigt nicht zu der Annahme, dass solche Bezeichnungen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und deshalb von jedermann benutzt werden dürften.

Soweit in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z.B. DIN, VDI) Bezug genommen oder aus ihnen zitiert worden ist, kann der Verlag keine Gewähr für Richtigkeit, Vollständigkeit oder Aktualität übernehmen.

**ISBN ebook 978-3-901608-80-3, 2021**

**DOI 10.11128/arep.41**

**ISBN print 978-3-901608-41-4, 2013**

**ARGESIM Report 41**

**ASIM Mitteilungsnummer AM 145**



## **Tagungsleitung**

Heinz-Theo Mammen, Hella KGaA Hueck & Co. Lippstadt, ASIM/GI-Fachgruppe STS

Roland Reichardt, FH Düsseldorf, Fachbereich Maschinenbau und Verfahrenstechnik

## **Programmorganisation**

Walter Commerell: [commerell@hs-ulm.de](mailto:commerell@hs-ulm.de)

Ingrid Bausch-Gall: [ingrid.bausch-Gall@bausch-gall.de](mailto:ingrid.bausch-Gall@bausch-gall.de)

Joachim Haase: [joachim.haase@eas.iis.fraunhofer.de](mailto:joachim.haase@eas.iis.fraunhofer.de)

Daniel Lückerrath: [lueckerath@informatik.uni-koeln.de](mailto:lueckerath@informatik.uni-koeln.de)

Heinz-Theo Mammen: [heinz-theo.mammen@hella.com](mailto:heinz-theo.mammen@hella.com)

Klaus Panreck: [klaus.panreck@fh-bielefeld.de](mailto:klaus.panreck@fh-bielefeld.de)

Thorsten Pawletta: [thorsten.pawletta@hs-wismar.de](mailto:thorsten.pawletta@hs-wismar.de)

Roland Reichardt: [roland.reichardt@fh-duesseldorf.de](mailto:roland.reichardt@fh-duesseldorf.de)

Michael Striebel: [michael.striebeI@zf-lenksysteme.com](mailto:michael.striebeI@zf-lenksysteme.com)

## **Tagungsorganisation**

Fachhochschule Düsseldorf

Fachbereich Maschinenbau und Verfahrenstechnik

Josef-Gockeln-Straße 9

40474 Düsseldorf

Tel: +49 211 4351-400

Fax: +49 211 4351-403

[www.mv.fh-duesseldorf.de](http://www.mv.fh-duesseldorf.de)

## **Tagungsort**

Fachhochschule Düsseldorf

Fachbereich Maschinenbau und Verfahrenstechnik

Josef-Gockeln-Straße 9

40474 Düsseldorf



## Inhaltsverzeichnis

### Plenarvortrag

<i>Influence of module geometry and contact resistance on the performance of thermoelectric generators analyzed by multiphysics simulation</i> Dirk Ebling – FH Düsseldorf .....	1
---	---

### Plenarvortrag

<i>Modellbildung eines Schwerlastfahrzeugs für den Containertransport</i> Dr. Mohammad Ahmadian – TEREX GOTTWALD (nicht im Tagungsband enthalten)	
--	--

### Tutorium 1

<i>Introduction to Autosar</i> Jens Zimmermann – Altran GmbH .....	7
---	---

### Tutorium 2

<i>Einführung in Modelica zur Modellierung von Energiesystemen</i> L. Gall – Bausch-Gall GmbH (nicht im Tagungsband enthalten)	
---	--

### Modellierungssprachen und Standards

<i>Neues zu Modelica Libraries</i> I. Bausch-Gall – Bausch-Gall GmbH (nicht im Tagungsband enthalten)	
<i>Memristor-Modellen für Modelica</i> Ch. Clauss, K. Majetta – Fraunhofer IIS/EAS .....	21

<i>Physical Modeling and Simulation with MapleSim 6</i> J. Friebe, Th. Richard – Maplesoft (nicht im Tagungsband enthalten)	
--	--

<i>ModelOpt – ein Werkzeug zur Verbesserung von Simulationsmodellen</i> O. Engel, A. Renz, F. Gottelt – XRG Simulation GmbH .....	31
--	----

### Verkehrsmodellierung und -simulation

<i>Effizientere Verkehrsplanung mittels Automatisierungstools</i> M. Boden, H. Weger, K. Kabitzsch – Technische Universität Dresden .....	35
--	----

<i>Applicability of rescheduling strategies in tram networks</i> D. Lückerrath, O. Ullrich, E. Speckenmeyer – Uni Köln .....	43
---	----

<i>Semantic Road Network Models for Rapid 3D Traffic Scenario Generation</i> T. Haubrich – Hochschule Bonn-Rhein-Sieg .....	51
--	----

### Variantenmodellierung / Ereignisdiskrete Systeme

<i>Ontologische Modellierung und Modellgenerierung in der MATLAB/Simulink Umgebung: Die "Tiny SES Toolbox"</i> F. Schwatinski, Th. Pawletta – Hochschule Wismar .....	57
--	----

<i>Ontologische Modellierung und ereignisdiskrete Simulation mit Matlab/SimEvents am Beispiel der ressourceneffizienten Prozesskettenplanung</i> A. Schmidt, Th. Pawletta – Hochschule Wismar .....	65
--	----

---

<i>Modellbasierter Entwurf ereignisdiskreter Steuerungen</i>	
D. Weidemann, R. Diekmann – FH-Bielefeld .....	75

<i>Anmerkungen zur Modellierung hybrider dynamischer Systeme</i>	
J. Haase, E. Hessel – FhG IIS-EAS, Hella KGaA Hueck & Co. ....	83

## Systemsimulation im Automobilbereich

<i>Hochflexibles HiL-System zur modellbasierten Entwicklung vernetzter Regelsysteme für Elektrofahrzeuge</i>	
W. Diehl, F. Quantmeyer, X. Liu-Henke – Hochschule Ostfalia .....	89

<i>Ein Verhaltensmodell und ein Laborprüfstand zur SiL und HiL-Simulation eines zentralen Steuergeräts für ein intelligent-effizientes Elektrofahrzeug</i>	
C. Masjosthusmann, N. Decius, U. Köhler, U. Büker – HELLA KGaA Hueck & Co .....	97

<i>Modellbildung und Identifikation der Energiespeicher für Elektrofahrzeuge</i>	
F. Quantmeyer, J. Kießling, X. Liu-Henke – Hochschule Ostfalia .....	105

<i>Variantenmanagement in der modellbasierten Produktlinienentwicklung von Fahrzeugsystemen</i>	
H. Holdschick, W. Commerell – Daimler AG; Hochschule Ulm .....	111

## Thermische Systeme

<i>Thermische Prozesssimulation, auch für „Nicht-Simulationsexperten“, auf der Basis von Computeralgebra</i>	
St. Braun – SmartCAE (nicht im Tagungsband enthalten)	
<i>Thermische Modellierung eines doppelstöckigen Wagenkastens für Next Generation Train</i>	
I. Windemut – DLR .....	119

## Hydraulische Systeme

<i>Nichtlineare Modelle hydraulischer Komponenten zur energieflussbasierten Modellierung</i>	
I. Alkov, D. Weidemann – FH Bielefeld .....	125
<i>Echtzeitsimulation hydraulischer Antriebe - vom Analogrechner zum FPGA</i>	
P. Beater – Fachhochschule Südwestfalen .....	135

## Physikalische Modellierung Antriebsstrang

<i>Multiphysikalische 1D-Simulation hybrider Antriebssysteme für Schienenfahrzeuge</i>	
S. Schmid – Voith Turbo GmbH & Co. KG.....	147
<i>Auslegung eines Rangeextender-Antriebsstrangs mit Hochtemperatur-Brennstoffzelle</i>	
M. Nasri – DLR .....	155

## Modellbasierte Bewertungsverfahren

<i>Prozess kalibriertes Materialmodell für die Spritzgießsimulation von hochgefüllten, elektrisch und thermisch leitfähigen Thermoplasten</i>	
J. Dörner, A. Swoboda, J. Wortberg – Universität Duisburg-Essen .....	161

---

<i>Cost- Benefit Analysis of Model-Based Software Development</i> Jens Zimmermann – Altran GmbH.....	171
---	-----

## **Mathematische Verfahren in Modellbildung und Simulation**

<i>Mehrgitterbasierte Lösungsmethoden zur optimalen Kontrolle von instationären Strömungen</i> M. Köster – Technische Universität Dortmund .....	183
---	-----

<i>Mathematical Modelling and Numerical Algorithms for Simulation of Oil Pollution</i> Qu. A Dang, M. Ehrhardt, Gia Lich Tran, Duc Le – Bergische Universität Wuppertal .....	191
--	-----

<i>Methods for modelling and numerical simulation of fluid flow problems in open channels and closed pipes</i> T. Jax, G. Steinebach – Hochschule Bonn-Rhein-Sieg.....	193
---	-----

<i>Simulationsbasierte Optimierung von ESD-Schutzkonzepten für Integrierte Schaltungen</i> St. Scheier, A. Razavi, St. Frei – TU Dortmund .....	199
--	-----

## **Komponentenmodellierung im Automobilbau**

<i>Systemkonzept einer aktiven Fahrzeugfederung für Elektrofahrzeuge</i> X. Liu-Henke, R. Buchta, M. Scheele – Hochschule Ostfalia .....	211
---	-----

<i>Modeling Worm Gear Mechanism in Three Architectures, Parameter Identification and Comparison</i> Xin Zhao, S. Petkun – Brose Fahrzeugteile GmbH & Co. KG.....	219
---	-----

<i>Practical aspects of modeling of mechatronics system at the example of BroSAnT system</i> S. Petkun – Brose Fahrzeugteile GmbH & Co. KG .....	227
---	-----



# Influence of module geometry and contact resistance on the performance of thermoelectric generators analyzed by multiphysics simulation

D. Ebling<sup>1</sup>, K. Bartholomé<sup>2</sup>, M. Bartel<sup>2</sup>, M. Jägler<sup>2</sup>

<sup>1</sup> University of Applied Science Düsseldorf

<sup>2</sup> Fraunhofer Institut IPM

*dirk.ebling@fh-duesseldorf.de*

The performance of thermoelectric devices is not only determined by the properties of the thermoelectric material but also by the geometrical design and thermal matching of the materials. Length of the legs and the quality of the contacts strongly influence the efficiency of thermoelectric generators. Experimental results of the contact properties are compared with latest measurements of the performance of modules manufactured from Bi<sub>2</sub>Te<sub>3</sub> compounds. The performance of modules is related to the obtained contact resistance and thermoelectric material properties. The different influences are studied by examples for thermoelectric multiphysics finite element modeling, where in addition to the thermoelectric field equations further effects like convection and radiation as well as the temperature dependency of the material properties are taken into account. An extensive modeling of the thermoelectric device describes the experimental findings with respect to contact properties and geometry.

Keywords:

Thermoelectric generators, modeling, contact resistance, module geometry

## 1 Introduction

Good thermoelectric material properties are inevitable requirements for a thermoelectric module exhibiting high efficiency. Rowe et al.<sup>1</sup> describe the impact of the module's contact resistance on the performance of thermoelectric generators. Even with very good thermoelectric material, the device performance can be rather poor, if the contact resistances of the module are too large<sup>2</sup>.

The figure of merit  $ZT$  of a thermoelectric generator is a measure of the performance and is closely related to the efficiency of a module<sup>3</sup>. It is strongly affected by the modules resistance and is given by:

$$ZT_{Module} = \frac{\alpha^2 T}{K_{Module} R_{Module}} \quad (1)$$

where  $\alpha$  denotes the Seebeck coefficient of the thermoelectric material,  $T$  the averaged temperature of the module, and  $K_{Module}$  and  $R_{Module}$  are the heat conductance value and total resistance of the module, respectively.

They are given by:

$$K_{Module} = \frac{\lambda_n A_n}{l} + \frac{\lambda_p A_p}{l} \quad (2)$$

$$R_{Module} = R_{legs} + R_c = \frac{\rho_n l}{A_n} + \frac{\rho_p l}{A_p} + R_c \quad (3)$$

Here,  $A$  and  $l$  denote the cross-section area and the lengths of the legs, and  $\lambda$  and  $\rho$  are the thermal conductivity and specific electric resistance, respectively. The resistance of the legs is given by  $R_{legs}$ , and the contact resistance is denoted by  $R_c$ . The subscripts n and p denote the n-type and p-type doping of the legs.

Now, the modules  $ZT$ -value can be rewritten to:

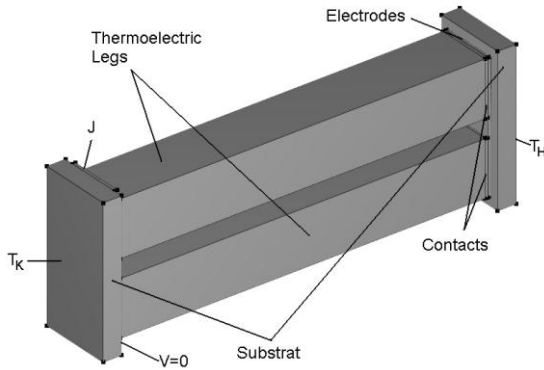
$$ZT_{Module} = ZT_{mat} \frac{1}{1 + \frac{R_c}{R_{legs}}} \quad (4)$$

where  $ZT_{mat}$  denotes the  $ZT$ -value of the material without contact resistance.

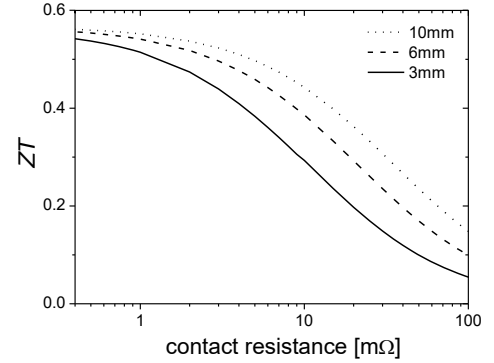
For non-vanishing contact resistances of the module its  $ZT$ -value is always smaller than what would be reachable given the material's  $ZT$ -value. However, the influence of the contact resistance  $R_c$  vanishes for increasing leg's resistance  $R_{\text{legs}}$ . This suggests the possibility to minimize the effect of contact resistances by increasing the size  $l$  of the thermoelectric legs, since the module's resistance  $R$  scales with  $l$ .

## 2 Simulations

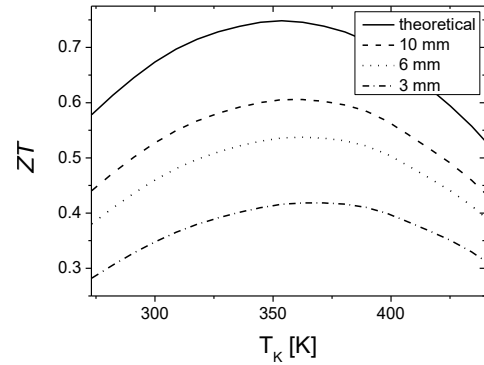
These theoretical predictions were analysed using finite element analysis (FEA) simulations for a model displayed in Figure 1, following the approach of Jäggle et al.<sup>4,5</sup>. Figure 2 shows the impact of the contact resistance on the module's  $ZT$ -value for different lengths of the thermoelectric legs. It shows, that the decrease of the  $ZT$ -values for increasing contact resistance is reduced for longer thermoelectric legs. Figure 3 also supports these findings in dependence of the cold side temperature  $T_K$  of the device. The maximal module performance at  $T_K \sim 360\text{K}$  results from the temperature-dependent material properties of the thermoelectric legs used in these simulations<sup>2</sup>. For these simulations, the contact resistance was assumed to be  $R_c = 10\text{m}\Omega$ . For the entire temperature range, the modules with 10mm leg size performed clearly better than the modules with leg sizes 6mm and 3mm.



**Figure 1.** Model of a pair of thermoelectric legs used for the FEA-Simulations. The simulated thermocouple consists of two legs with a basis of size  $1.5\text{mm} \times 1.5\text{mm}$ . The legs are contacted with copper on an alumina substrate. The contact resistances were simulated with an additional subdomain between the electrodes and the thermoelectric legs.



**Figure 2.** Simulation results: Impact of the contact resistance on the  $ZT$ -value of the module for modules with different leg sizes, calculated at room temperature. For vanishing contact resistance, the  $ZT$ -values of the modules approach the maximum value of  $\sim 0.57$  of the material applied. Modules with larger legs have a higher  $ZT$ -value than modules with smaller legs.



**Figure 3.** Simulation results: The modules  $ZT$ -values versus the cold-side temperatures  $T_K$  for different leg sizes. The simulations predict, that modules with longer legs show a better performance.

## 3 Module Fabrication

In order to verify the simulation results experimentally, modules with different leg sizes were fabricated. The manufacturing steps from the wafers produced by spark plasma sintering (SPS)<sup>6,7</sup> to thermoelectric modules with different leg sizes is illustrated in Figure 4. In the first processing step, both sides of the SPS-wafers were polished to a thickness of  $1.5\text{mm}$ . Legs with a base area of  $1.5 \times 1.5\text{mm}^2$  and length  $l$  of 3mm, 6mm and 10mm were cut from the polished n- and p-doped SPS-wafers using a wafer-saw. After the determination of the thermoelectric properties of these legs, they were soldered to an  $\text{Al}_2\text{O}_3$  substrate with contact pads by connecting the legs electrically in series and thermally in parallel, building modules with two and eight thermoelectric leg pairs, respectively. In order to minimize the contact resistance,

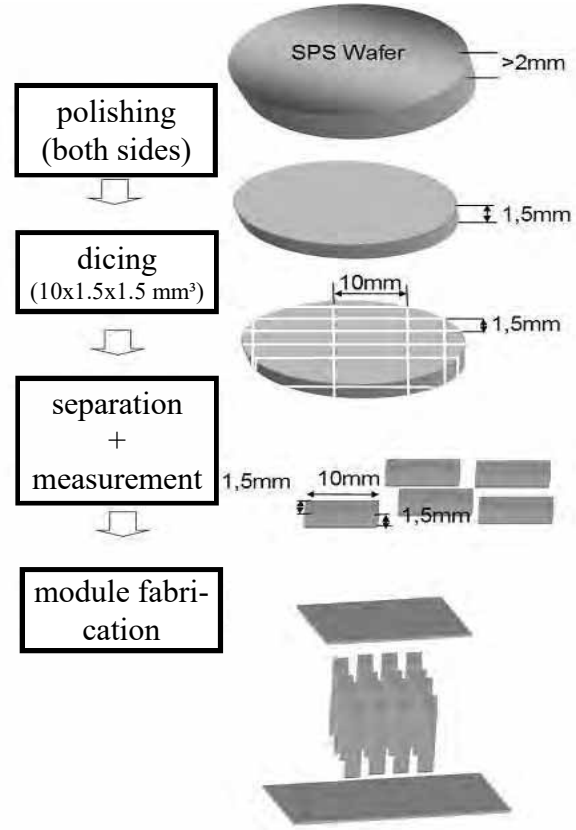


different commercial solders were tested and compared, yielding very different contact resistances and hence  $ZT$ -values of the modules, see Table 1. In Figure 5, the simulation results for the temperature-dependent  $ZT$ -values are displayed for different solderings according to Equation 1. For each soldering, a range of  $ZT$ -values is given, corresponding to the range of the experimentally determined contact resistances. The  $ZT$ -value for modules with SnPbBi soldering is up to five times larger than the  $ZT$ -value for modules fabricated with a SnAgCu soldering. The best soldering material results in  $ZT$ -values for modules getting very close to the theoretical limit determined by the material properties (black line in Figure 5).

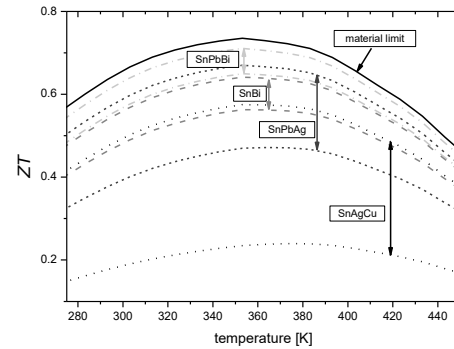
Experimentally, the  $ZT$ -values, electric resistances and maximum  $\Delta T$  were measured for each module using the LTD Z-Meter Model DX4065 from RMT Ltd. This commercial device determines the thermoelectric parameters by applying the Harman method<sup>8</sup>. A short current pulse is applied to the module in order to produce a temperature gradient along the thermoelectric legs. The voltage drop along the module resulting from this temperature gradient is determined, enabling the calculation of the desired thermoelectric parameters<sup>9</sup>. The experimental setup of the commercial equipment was modified to perform measurements not only at room temperature but also up to at least 300°C.

SOLDER	$ZT_{MAX}$	CONTACT RESISTANCE [MΩ]
SnPbBi	0.41	0.5-1.9
SnBi	0.31	1.4-8.1
SnPbAg	0.25	2.1-4.4
SnAgCu	0.23	4.0-30.6

**Table 1.** The impact of different solderings on the contact resistance. The size of the contact resistances can vary up to two orders of magnitudes depending on the soldering.



**Figure 4.** Production of a thermoelectric module from SPS-material.



**Figure 5.** Simulation results of the  $ZT$ -value for different solderings. The minimum and maximum values of contact resistances given in Table 1 were used to determine the range of  $ZT$ -values in dependence on the temperature for each soldering. The black curve shows the theoretical limit determined by the material properties.

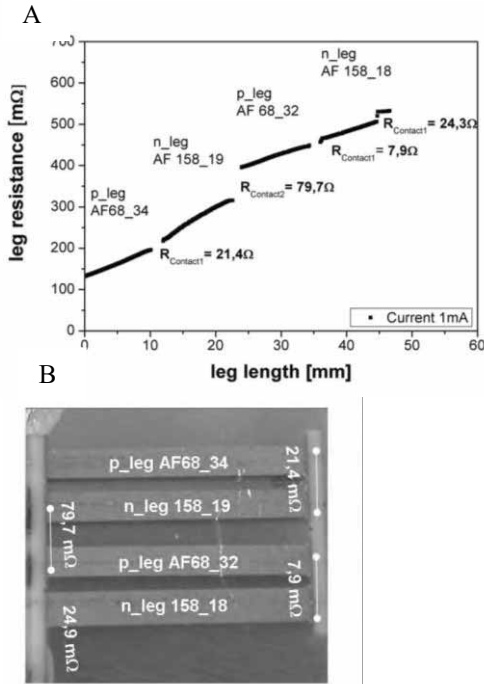
#### 4 Measuring the contact resistance

In a first approach, the contact resistance was calculated from difference between the measured total resistance of the module and the sum of the resistances of each thermoelectric leg. Therefore, the electric

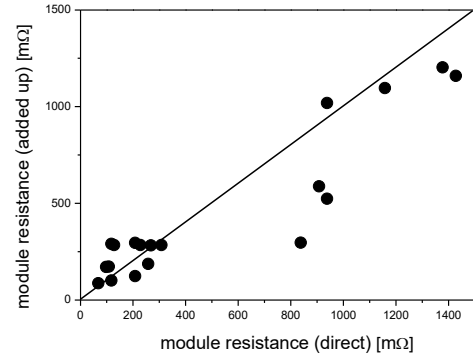
conductivity of each single leg was determined in advance.

For validation, an alternative method to determine the contact resistance was applied by measuring the distribution of the resistance directly across the module. Therefore, the voltage drop generated by a pulsed current is measured using a contact tip at a certain position on the thermoelectric leg. By scanning the contact tip across the module, the resistance can be measured in dependence of the leg length point by point. Figure 6 shows a course of these measurements for a module with large contact resistance assembled of two pairs of legs. The linear increase of the resistance is caused by the electric conductivity of the thermoelectric material. The discontinuities result from the contact resistances of the solderings. Thus, the actual contact resistance of each solder joint can be determined by measuring the difference of the resistance before and after the soldering part. The sum of the individual contact resistances then gives the total contact resistance of the module.

Figure 7 compares directly the measured total module resistance to the sum of individually obtained point by point resistances of a module. The deviation from the bisecting line is caused by the uncertainties of the measuring methods.



**Figure 6.** Exemplary results of the measurement of the contact resistance of a module. The discontinuities represent the contact resistances, the linear slopes the resistance of the thermoelectric legs.



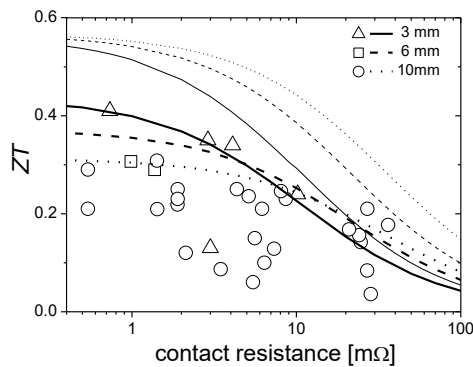
**Figure 7.** Comparison of the experimental module resistances: Displayed on the x-axis are the directly measured module resistances, and on the y-axis the module resistances resulting from adding up the individually measured resistances from its constituents. The deviations from the bisecting line represent the uncertainty of the measuring method.

## 5 Experimental Results

In order to compare the theoretical predictions, several modules with leg sizes of 3mm, 6mm and 10mm were built according to the procedure described in Figure 1. The  $ZT$ -values and contact resistances were determined for each module using the methods described above. The results are depicted in Figure 8. In agreement with Equation (1) and the FEM-simulation results, the  $ZT$ -values of the modules get larger, the smaller the contact resistances become. However, in contrast to the theoretical predictions (see Figure 2), the modules built with legs of 3mm length show a better performance than modules with the longer leg sizes.

But the theoretical prediction of Figure 2 does not account for the impact of heat loss along the thermoelectric legs due to convection and radiation. The inclusion of these effects (see thick lines, Figure 8) yields a good agreement of the theoretical predictions with the experimental data (details of the simulations are given in Bartholomé et al<sup>10</sup>). The simulation takes into account the heat losses due to the measurement of the emissivity of the thermoelectric material. For convection an assumption was done by defining an average heat transfer coefficient, which is in agreement with the experimental results. But in detail this can be only a rough estimate due to the unknown local temperatures of the transporting air and the type of convective flow. Inclusion of heat loss could also explain the inverse dependence of the performance on the length of the legs. (fig.8). From the simulation it is shown, that the most evident influence is due to

convection. A detailed experimental analysis is due to future investigations, which will be carried out in vacuum to suppress the convective part. Also the potential influence of radiation shields has to be tested for a better understanding. These uncertainties may also cause the broad scattering of the measured values for  $ZT$  with the length of the legs. A detailed analysis is still difficult at this state without further experimental results.



**Figure 8.** The impact of contact resistance on the module  $ZT$ -value for different module lengths, comparison of experimental results (triangle, squares and circle) with the simulation results with (thicker lines) and without (thinner lines) the inclusion of radiative and convective heat loss.

## 6 Conclusion

In this manuscript, the impact of contact resistance on the module's performance is investigated. Following theoretical investigations and simulation results, a number of modules with different leg size were produced in order to analyze the influence of leg length and the negative effect of contact resistance on the  $ZT$ -values of thermoelectric generators. Different soldering materials were used to reduce the value of contact resistance. The prediction of better performance with smaller contact resistance could be verified in case of our manufactured modules. However, the prediction that the module performance increases with the length of their legs, could not be supported by the experiments. Further analysis showed that heat loss along the thermoelectric legs by convection and radiation is mainly responsible for this deviations.

## 7 References

[1] D.M. Rowe, G. Min, Science, Measurement and Technology, IEE Proceedings, **143**, 351, (1996)

[2] D. Ebling, M. Jaegle, M. Bartel, A. Jacquot, H. Böttner, J. Electronic Materials, **38**, 1456, (2009)

[3] J. Yang, T. Caillat, MRS Bulletin, **31**, 224 (2006)

[4] M. Jaegle, M. Bartel, D. Ebling, A. Jacquot H. Böttner, Proceedings of the 6th European Conference on Thermoelectrics, Paris, France, 2–4 July 2008

[5] M. Jaegle, Proceedings of the 5th European Conference on Thermoelectrics, Odessa, Ukraine, 10–12 September, 2007

[6] H. Böttner, D Ebling, A Jacquot, J König, L Kirste, and J Schmidt, Phys. Stat. Sol. (RRL) **1**, 235 (2007)

[7] D Ebling, A Jacquot, M Jäggle, H Böttner, U. Kühn, and L Kirste, Phys. Stat. Sol. (RRL) **1**, 238 (2007)

[8] T C Harman, "Special Techniques for Measurement of Thermoelectric Properties", J. Appl. Phys. **29**, 1373 (1958)

[9] A. Jacquot, M. Jäggle, J. König, D.G. Ebling, H. Böttner, Proceedings of the 5th European Conference on Thermoelectrics, Odessa, Ukraine, 10–12 September, 2007

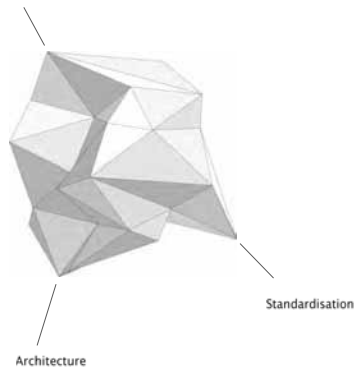
[10] K. Bartholome, M. Bartel, M. Jäggle, D. Ebling, to be published in this journal



# Short Introduction into AUTOSAR

Tutorial for the ASIM Workshop  
28.02.2013 – 01.03.2013 FH Düsseldorf  
Dr. Jens Zimmermann, Altran

Embedded Software



F 084 D R06 2012-07

altran

## Agenda

- 1 – The AUTOSAR Partnership**
- 2 – AUTOSAR Architecture
  - 2.1 – Software Components
  - 2.2 – Runtime Environment
  - 2.3 – Basic Software
- 3 – Development Process
- 4 – Demonstration

F 084 D R06 2012-07

altran



**AUTomotive Open System ARchitecture**

3

F 084 D R06 2012-07

altran

## Challenges of the automotive software development

### Flexibility increase

- OEM specific basic software
- Project specific functional interfaces
- Strong dependency between function, basic software and control unit
- Time-consuming adjustments of software in case of hardware changes

### Efficiency increase

- Complicated, but not perceivable hardware integration
- Time-consuming function reuse
- Growing complexity of the different automobile functions
- Growing time pressure in the development
- Different tools have no compatibility

4

F 084 D R06 2012-07

altran

## Official goals of the consortium



- Consideration of availability and safety requirements
- Scalability to different vehicle and platform variants
- Standardization (and Implementation) of basic system functions as an OEM-wide "Standard Core" solution
- Transferability of functions throughout network
- Integration of functional modules from multiple suppliers
- Maintainability throughout the whole "Product Life Cycle"
- Increased use of "Commercial off the shelf hardware"
- Software updates and upgrades over vehicle lifetime
- Redundancy activation

5

F 084 D R06 2012-07

altran

## AUTOSAR development partnership

### 9 Core Partners

BMW, Bosch, Continental, Daimler,  
Ford, GM, PSA, Toyota, Volkswagen

BMW Group



BOSCH

Continental

DAIMLER



PSA PEUGEOT CITROËN

TOYOTA

VOLKSWAGEN AG

### 50\* Premium Partners and 16\* Development Partners

Participation in the standard creation,  
Access to the current specifications

### 76\* Associate Partners

\*data end of 2012

Access to all the Releases

Only partners have the right to use the standard for commercial purposes

6

F 084 D R06 2012-07

altran

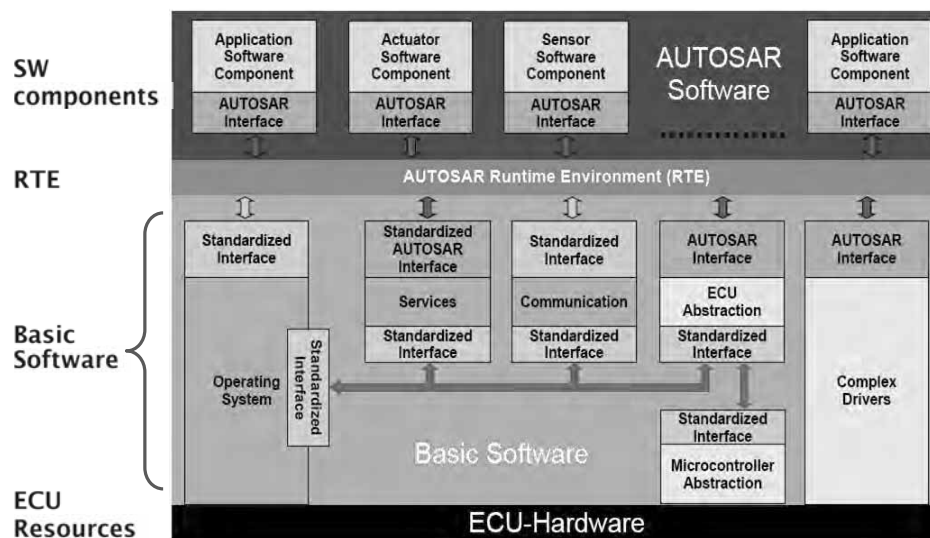
# Agenda

- 1 - The AUTOSAR Partnership
- 2 - AUTOSAR Architecture
  - 2.1 - Software Components
  - 2.2 - Runtime Environment
  - 2.3 - Basic Software
- 3 - Development Process
- 4 - Demonstration

F 084 D R06 2012-07

altran

## AUTOSAR Architecture



8

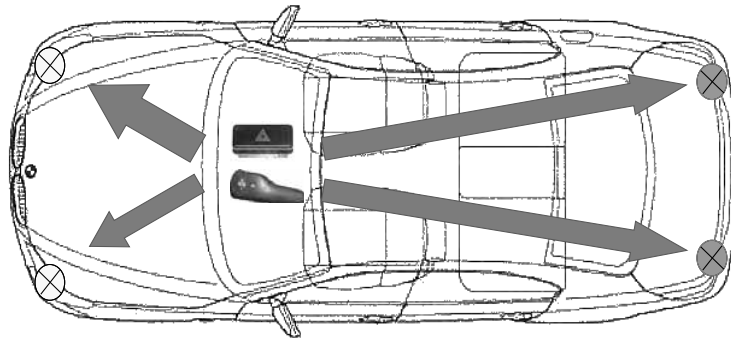
F 084 D R06 2012-07

AUTOSAR Architecture - Introduction

altran



## Project example: indicator light control



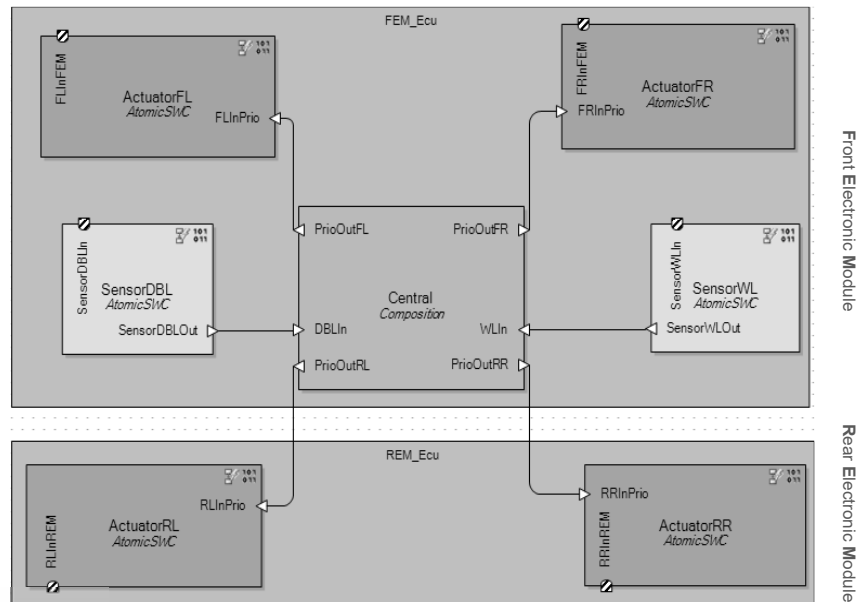
9

F 084 D R06 2012-07

AUTOSAR Architecture – Software Components

altran

## Project example: function network



10

F 084 D R06 2012-07

AUTOSAR Architecture – Software Components

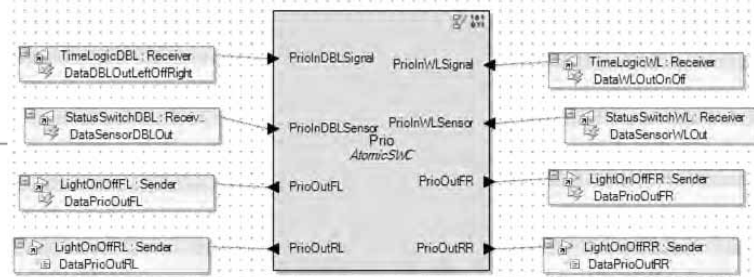
altran

## Standardization of the component description

### Software-Component-Description

- Standardized XML description of an AUTOSAR Software Component (SW-C)
- Description structure: „Software Component Template“ (generated from the meta-model)
- Content:
  - Provided or required operations and data elements of a SW-C
  - Internal structure
  - No implementation!

Visualization of a Software-Component-Description



11

F 084 D R06 2012-07

AUTOSAR Architecture – Software Components

ALTRAN

## Standardization of the component description

```
<ATOMIC-SOFTWARE-COMPONENT-TYPE>
<SHORT-NAME>Prio</SHORT-NAME>
<PORTS>
  <R-PORT-PROTOTYPE>
    <SHORT-NAME>PrioInWL</SHORT-NAME>
    <REQUIRED-COM-SPECS>
      <UNQUEUED-RECEIVER-COM-SPEC>
        <DATA-ELEMENT-IREF>
          <R-PORT-PROTOTYPE-REF DEST="R-PORT-PROTOTYPE">/koffer/Prio/PrioInWL</R-PORT-PROTOTYPE-REF>
          <DATA-ELEMENT-PROTOTYPE-REF DEST="DATA-ELEMENT-PROTOTYPE">/koffer/TimeLogicWL/DataWLOutTime</DATA-ELEMENT-PROTOTYPE-REF>
        </DATA-ELEMENT-IREF>
      </UNQUEUED-RECEIVER-COM-SPEC>
      <QUEUED-RECEIVER-COM-SPEC>
        <DATA-ELEMENT-IREF>
          <R-PORT-PROTOTYPE-REF DEST="R-PORT-PROTOTYPE">/koffer/Prio/PrioInWL</R-PORT-PROTOTYPE-REF>
          <DATA-ELEMENT-PROTOTYPE-REF DEST="DATA-ELEMENT-PROTOTYPE">/koffer/TimeLogicWL/DataWLOutOnOff</DATA-ELEMENT-PROTOTYPE-REF>
        </DATA-ELEMENT-IREF>
        <QUEUE-LENGTH>1</QUEUE-LENGTH>
      </QUEUED-RECEIVER-COM-SPEC>
    </REQUIRED-COM-SPECS>
    <REQUIRED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/koffer/TimeLogicWL</REQUIRED-INTERFACE-TREF>
  </R-PORT-PROTOTYPE>
```

Extract from the ARXML Software-Component-Description

12

F 084 D R06 2012-07

AUTOSAR Architecture – Software Components

ALTRAN

## Implementation of Software Components

### Separation of architecture design and implementation

- Fully defined interfaces
- Only interfaces to the RTE

### Implementation task „simply“

- Fill the skeleton (c-functions are declared)
- Using the RTE interfaces

### Choice of programming environment

- Basis: C
- Model based software development
- Object-oriented: C++

## Types of Software Components

### Application SW-Component

- Pure algorithm
- Data exchange only with other SW components  
(via Virtual Functional Bus in System View)

### Sensor / Actuator SW-Component

- Connection to the defined target ECU
- Access to the basic software I/O-components through the RTE

### Service-Component (BSW module offering a service interface)

- Connection to the defined target ECU
- Access to defined parts of the basic software
- Example: Diagnostic Event Manager

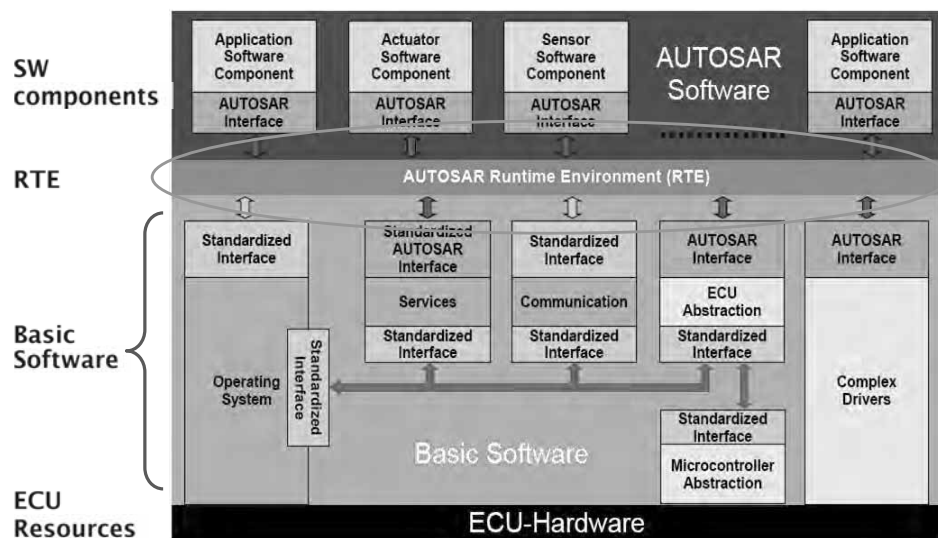
# Agenda

- 1 - The AUTOSAR Partnership
- 2 - AUTOSAR Architecture
  - 2.1 - Software Components
  - 2.2 - Runtime Environment
  - 2.3 - Basic Software
- 3 - Development Process
- 4 - Demonstration

F 084 D R06 2012-07

ALTRAN

## AUTOSAR Architecture



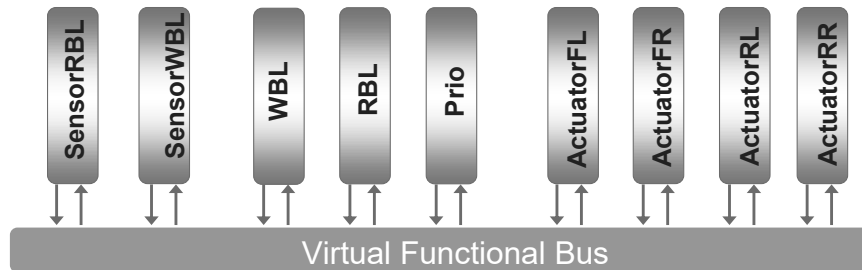
16

F 084 D R06 2012-07

AUTOSAR Architecture - Runtime Environment

ALTRAN

## Virtual Functional Bus



- Data exchange independent of communication path
- No deployment definition
- Hardware independence

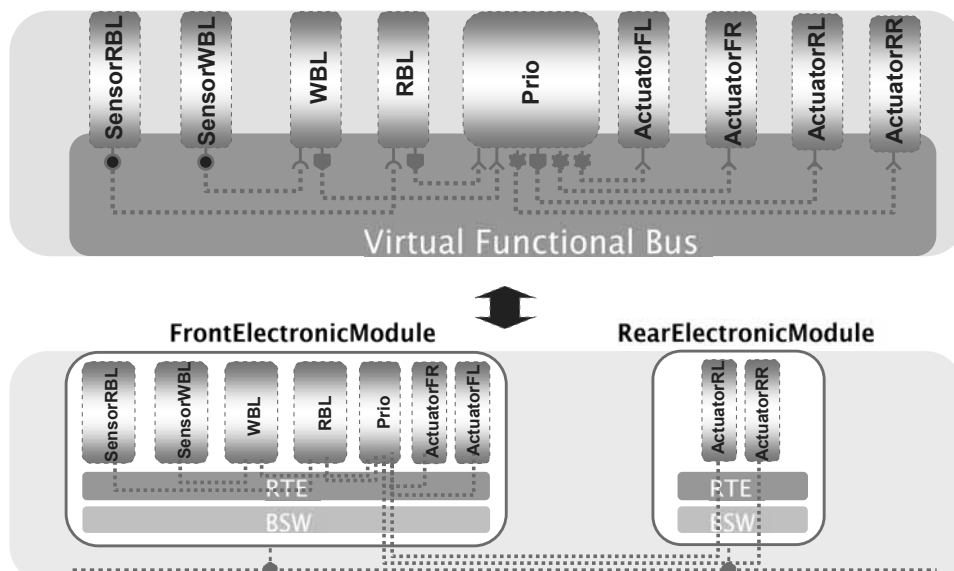
17

F 084 D R06 2012-07

AUTOSAR Architecture – Runtime Environment

altran

## Virtual Functional Bus and Runtime Environment



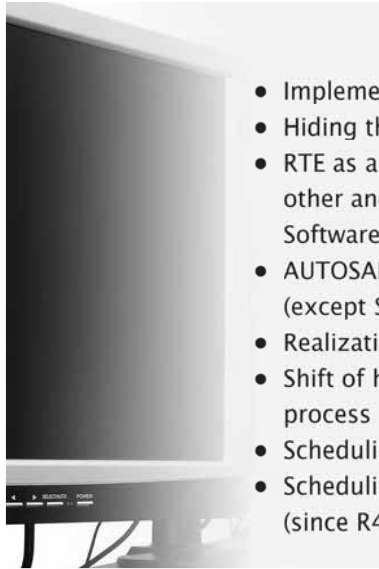
18

F 084 D R06 2012-07

AUTOSAR Architecture – Runtime Environment

altran

## Functionality of the RTE

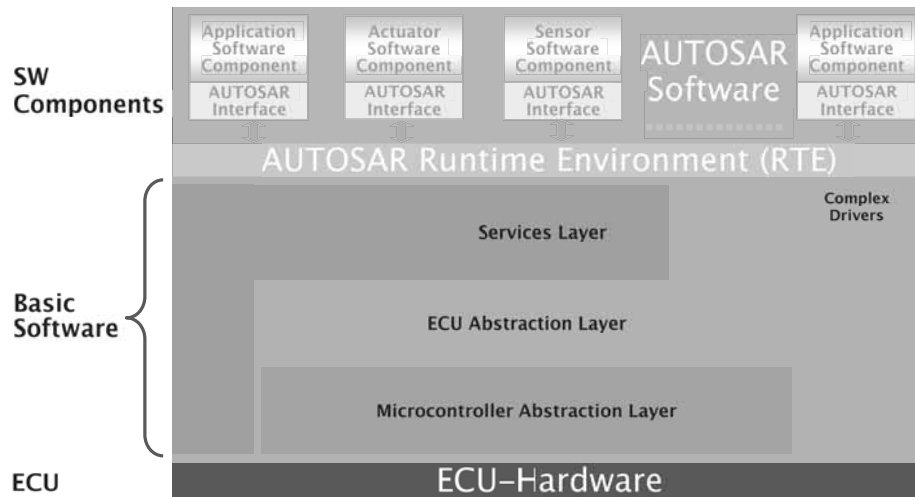


- Implementation of Virtual Function Bus
- Hiding the lower SW layers
- RTE as a communication way of the SW-C between each other and at the same time SW-C connection to the Basic Software
- AUTOSAR SW-Cs independence from the specific ECU (except Sensor/Actuator SW-C)
- Realization of a certain deployment in a network of ECU
- Shift of hardware-dependency in the SW-development-process
- Scheduling of SW-C functions
- Scheduling of main functions of BSW modules (since R4.0)

## Agenda

- 1 – The AUTOSAR Partnership
- 2 – AUTOSAR Architecture
  - 2.1 – Software Components
  - 2.2 – Runtime Environment
  - 2.3 – Basic Software
- 3 – Development Process
- 4 – Demonstration

## Abstraction layers



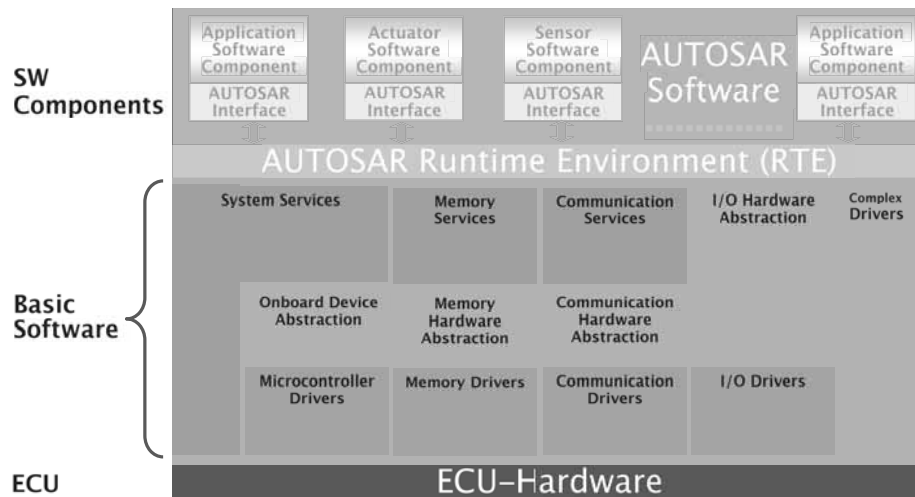
21

F 084 D R06 2012-07

AUTOSAR Architecture – Basic Software

ALTRAN

## Functional stacks



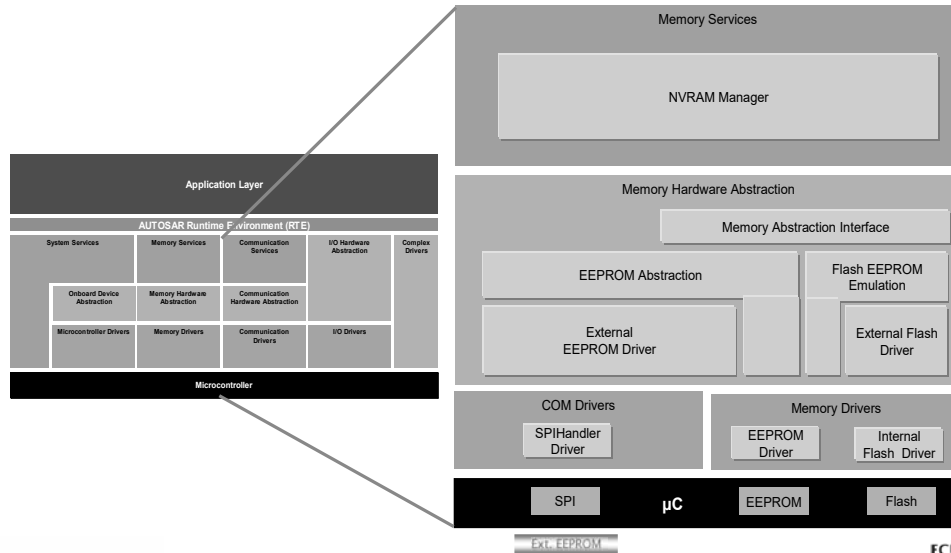
22

F 084 D R06 2012-07

AUTOSAR Architecture – Basic Software

ALTRAN

## Example: Layered abstraction in the memory stack



23

F 084 D R06 2012-07

AUTOSAR Architecture – Basic Software

ECU  
altran

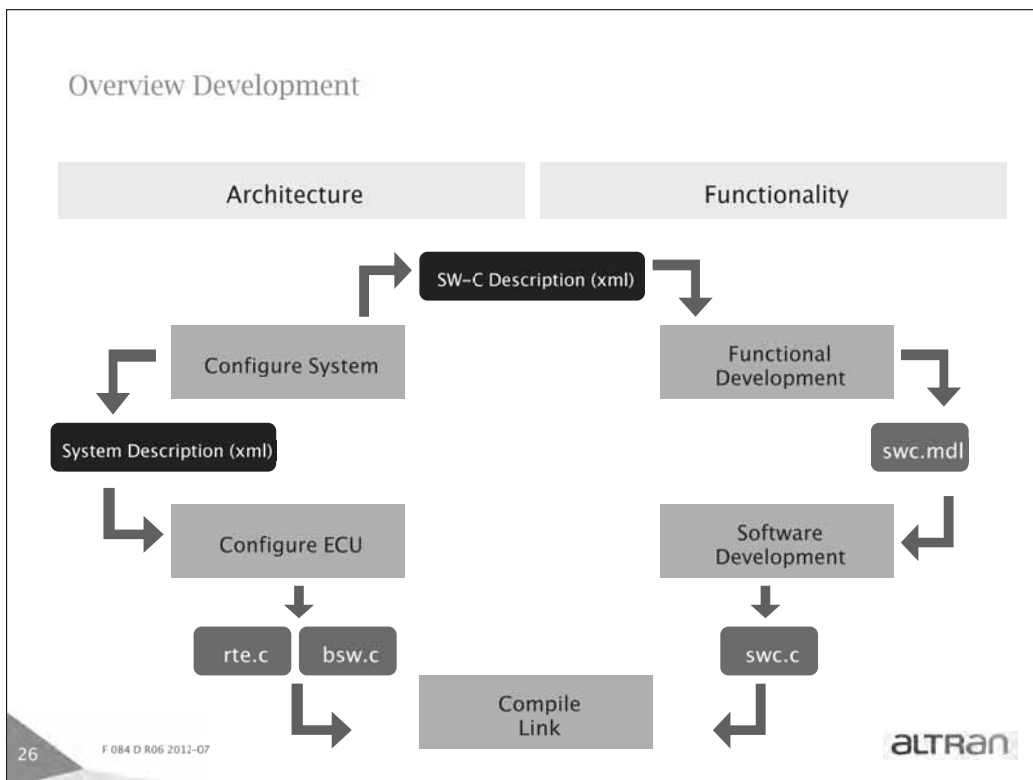
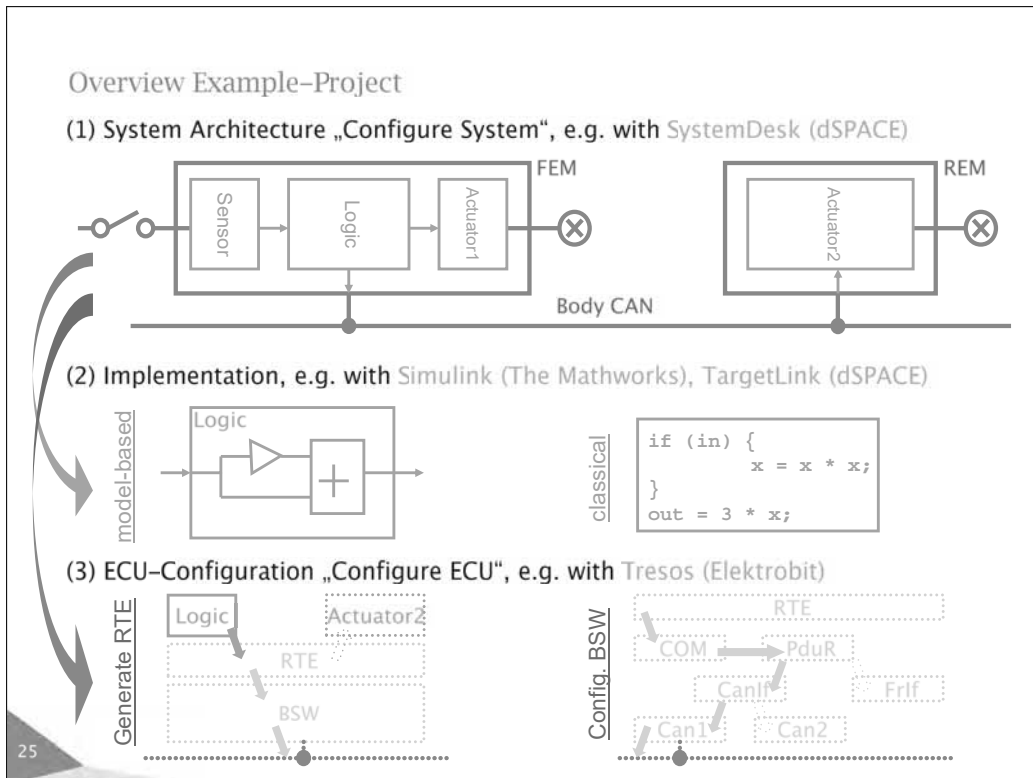
## Agenda

- 1 – The AUTOSAR Partnership
- 2 – AUTOSAR Architecture
  - 2.1 – Software Components
  - 2.2 – Runtime Environment
  - 2.3 – Basic Software
- 3 – Development Process
- 4 – Demonstration

F 084 D R06 2012-07

altran





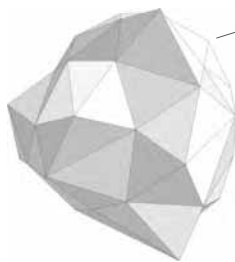
# Agenda

- 1 – The AUTOSAR Partnership
- 2 – AUTOSAR Architecture
  - 2.1 – Software Components
  - 2.2 – Runtime Environment
  - 2.3 – Basic Software
- 3 – Development Process
- 4 – **Demonstration**

F 084 D R06 2012-07

altran

INNOVATION MAKERS



F 084 D R06 2012-07

altran

# Simulationsmodelle für Memristoren

Kristin Majetta<sup>1</sup>, Christoph Clauß<sup>1</sup>, Torsten Schmidt<sup>2</sup>, Ronald Tetzlaff<sup>3</sup>

<sup>1</sup>Fraunhofer-Institut für Integrierte Schaltungen IIS, Institutsteil Entwurfsautomatisierung EAS  
Zeunerstraße 38, 01069 Dresden

<sup>2</sup>Hochschule Ansbach, Residenzstraße 8, 91522 Ansbach

<sup>3</sup>Technische Universität Dresden, Fakultät Elektrotechnik und Informationstechnik  
Helmholtzstraße 18, 01063 Dresden

*kristin.majetta@eas.iis.fraunhofer.de*

Es werden drei veröffentlichte Memristormodelle vorgestellt und deren Implementierung in Modelica anhand eines der Modelle beschrieben. Bisher veröffentlichte Simulationsergebnisse konnten nachvollzogen werden. Die Anwendbarkeit in analogen Schaltungen wird anhand einer ODER- und einer Integratorschaltung demonstriert. Die Modelica-Modelle wurden in einer Memristorbibliothek zusammengefasst. Am Beispiel eines Memristormodells wird die Implementierung in VHDL-AMS gezeigt.

## 1 Einleitung

Seit einigen Jahren ist es möglich, Memristoren zu realisieren. Memristoren sind veränderliche Widerstände, deren Widerstandswert von weiteren Zustandsvariablen abhängt; im ursprünglichen Fall von der dem Strom zugeordneten Ladung. Vor allem bei der Realisierung neuer Speichertechnologien spielen sie eine bedeutende Rolle. Auch als Verbindungsglieder zwischen den Zellen Neuronaler Netzwerke, die zur Bildverarbeitung oder zur Modellierung nichtlinearer, dynamischer Systeme genutzt werden, finden Memristoren Anwendung. Vor der technischen Anwendung von Memristoren steht immer deren Simulation. Dazu benötigte Memristor-Modelle sind in der Literatur ([1], [2], [3], [4], [5], [6], [7], [8]) in unterschiedlichen Abstraktionsgraden zu finden. In diesem Beitrag wurden diese Modelle in der Modellierungssprache Modelica implementiert, getestet und in einer Bibliothek samt einigen Anwendungsbeispielen zusammengefasst. Weiterhin wurde ein Memristor-Modell auch in VHDL-AMS beschrieben.

Das Papier ist wie folgt strukturiert: Im zweiten Abschnitt werden wesentliche grundlegende Zusammenhänge des Memristors beschrieben. Im dritten Abschnitt erfolgt die Vorstellung von Modellansätzen, deren Implementierung in Modelica im vierten Abschnitt beschrieben ist. Der fünfte Abschnitt zeigt beispielhaft anhand von zwei analogen Schaltungen, wie Memristormodelle verwendet werden könnten. Der sechste Abschnitt widmet sich der Beschreibung

eines Memristormodells in VHDL-AMS. Abschnitt sieben vermittelt eine Zusammenfassung und einen Ausblick auf weitere Arbeiten.

## 2 Grundlagen

Ein Memristor ist ein passives Bauelement, welches gemäß der ursprünglichen Definition von Chua [9] die elektrische Ladung  $q(t)$  und den magnetischen Fluss  $\varphi(t)$  miteinander verknüpft.

$$f(\varphi(t), q(t)) = 0 \quad (1)$$

Eine mögliche Realisierung von  $f$  ist zum Beispiel folgender einfache Polynomansatz:

$$f(\varphi(t), q(t)) = (m_0 + m_1 q(t))q(t) - \varphi(t) = 0 \quad (2)$$

Da der magnetischen Fluss die integrale Größe der elektrischen Spannung

$$\varphi(t) = \int_0^t u(\tau) d\tau, \quad \varphi(t_0) = \varphi_0 = 0 \quad (3)$$

und die elektrische Ladung die integrale Größe des Stromes ist

$$q(t) = \int_0^t i(\tau) d\tau, \quad q(t_0) = q_0 = 0 \quad (4)$$

führt das Differenzieren von (2) auf eine Beschreibung des Zusammenhangs zwischen Strom und Spannung, den üblichen Klemmengrößen an elektrischen Schaltelementen:

$$\underbrace{(m_0 + 2m_1 q(t))}_{M(q(t))} i(t) = u(t) \quad (5)$$

In der Memristorgleichung nach (5) bezeichnet man  $M(q(t))$  als Memristanz. Sie ist im Gegensatz zum linearen Widerstand nicht konstant, sondern hängt von der Ladung ab, die durch den Memristor fließt. Dadurch ist die Strom-Spannungs-Kennlinie des Memristors bei angelegter sinusförmiger Spannung  $u(t)$  keine Gerade, sondern eine Hysterese. Sie wird als „pinched“ bezeichnet [9], was ausdrückt, dass sie durch den Koordinatenursprung verläuft. Wählt man in (5)  $m_0 = 1$ ,  $m_1 = 1$ ,  $u(t) = u_0 \sin(\omega t)$  mit  $u_0 = 1,3 \text{ V}$  und  $\omega = 2\pi$ , ergibt sich die in Abbildung 1 gezeigte Hysterese.

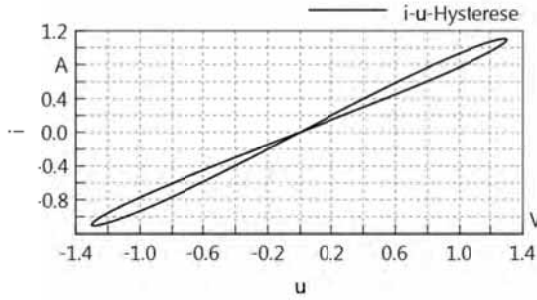


Abbildung 1. Hysterese bei sinusförmiger Spannung

Nach [9] besitzt ein Memristor neben seiner typischen Hysterese-Kennlinie zwei weitere wesentliche Merkmale. Zum einen wird die Fläche, die durch die Hysterese eingeschlossen ist, mit steigender Frequenz  $f$  immer kleiner, und zum anderen geht der Memristor für  $f \rightarrow \infty$  in einen linearen Widerstand über. Nachdem 2008 das erste Mal ein reales Bauelement mit der Theorie von Memristoren in Verbindung gebracht wurde, konnten anhand seines technischen Aufbaus und basierend auf Messungen Modelle erstellt werden, die das reale Verhalten nachbilden, das vom idealen Verhalten nach (5) in der Regel abweicht. Dennoch existieren auch bei diesen nach physikalischem Vorbild gestalteten Modellen unterschiedliche Abstraktionsgrade.

### 3 Modellansätze

Das erste in der Literatur veröffentlichte Modell (genannt HP-Modell) orientiert sich stark an dem 2008 bei Hewlett-Packard (HP) hergestellten Memristor. Es basiert auf einer beweglichen Grenzschicht, die sich zwischen einem dotierten und somit leitenden und

einem wenig leitenden Titandioxidgebiet ausbildet. Je nach Polarisation der angelegten Spannung verschiebt sich die Grenzschicht in Richtung des leitenden bzw. wenig leitenden Gebietes und die Memristanz wird größer bzw. kleiner. Weitere Beschreibungen zu Funktionsweise, Referenzergebnissen und Erweiterungen dieses Modells sind u. a. [1], [2], [3], [13] zu entnehmen.

Das derzeit detaillierteste Modell, es soll im Folgenden als Simmons-Modell bezeichnet werden, hat den gleichen physikalischen Aufbau wie das HP-Modell zum Vorbild - eine Titandioxidschicht, die zwischen zwei Platin-Elektroden liegt, und in die ein dotierter und somit leitender Kanal eingebracht ist. Zwischen dem Kanal und der gegenüberliegenden Elektrode bildet sich aufgrund des Driftens der Dotierungen ein Barrieretunnel der Breite  $x$  aus. Abbildung 2 zeigt den Aufbau in schematischer Form.

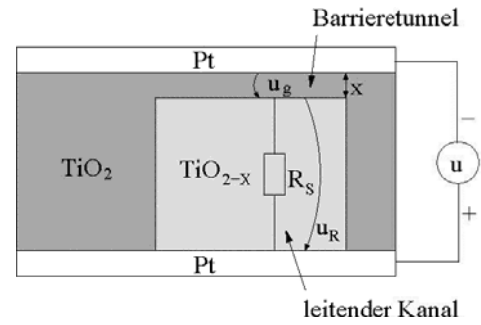


Abbildung 2. Schematischer Aufbau des Simmons-Modells

Aufgrund des Driftens der Dotierungen in Richtung der Kontaktstelle zwischen Titandioxid und Platin (obere Elektrode in Abbildung 2) breitet sich der leitende Kanal aus, die Tunnelbarriere wird zunehmend überwunden und es fließt schließlich ein erhöhter Strom. Für den Tunneleffekt, der am Übergang zwischen Titandioxid und Platin eine Rolle spielt, gab Simmons [10] folgende mathematische Beschreibung an:

$$i = \frac{j_0 A}{\Delta \omega^2} \left\{ \Phi_1 e^{-B\sqrt{\Phi_1}} - (\Phi_1 + eu_g) e^{-B\sqrt{\Phi_1 + eu_g}} \right\} \quad (6)$$

$$u_g = u - u_R = u - iR_S \quad (7)$$

In (6) und (7) bezeichnet  $A$  die Fläche des leitenden Kanals,  $e$  die Elementarladung,  $R_S$  den Widerstand des leitenden Kanals, der als konstant angenommen wird, und  $u_g$  die Spannung, die über dem Barrieretunnel anliegt (vgl. Abbildung 2). Die Größen  $j_0$ ,  $\Phi_1$

und  $B$  sind Variablen, deren Berechnung nach [4] gemäß:

$$j_0 = \frac{e}{2\pi h}, \quad (8)$$

$$\Phi_1 = \Phi_0 - eu_g \left( \frac{w_1 + w_2}{2x} \right) - \left( \frac{1.15\lambda x}{\Delta w} \right) \ln \left[ \frac{w_2(x - w_1)}{w_1(x - w_2)} \right], \quad (9)$$

$$\Delta w = w_1 - w_2, \quad (10)$$

$$w_1 = \frac{1.2\lambda x}{\Phi_0}, \quad (11)$$

$$w_2 = w_1 + x \left[ 1 - \frac{9.2\lambda}{3\Phi_0 + 4\lambda - 2eu_g} \right], \quad (12)$$

$$\lambda = \frac{e^2 \ln 2}{8\pi\kappa\epsilon_0 w} \quad \text{und} \quad (13)$$

$$B = \frac{4\pi\Delta w \sqrt{2m}}{h} \quad (14)$$

vorgenommen werden kann.

Die Breite  $x$  der Tunnelbarriere wird als Zustandsgröße des Memristors bezeichnet und ändert sich gemäß der Differentialgleichung

$$\frac{dx}{dt} = \begin{cases} f_{off} \sinh\left(\frac{i}{i_{off}}\right) \exp\left[-\exp\left(\frac{x - a_{off}}{w_c} - \frac{|i|}{b}\right) - \frac{x}{w_c}\right] & i > 0 \\ f_{on} \sinh\left(\frac{i}{i_{on}}\right) \exp\left[-\exp\left(-\frac{x - a_{on}}{w_c} - \frac{|i|}{b}\right) - \frac{x}{w_c}\right] & i \leq 0 \end{cases} \quad (15)$$

Das Simmons-Modell ist streng nichtlinear und führt bei der Simulation auf numerische Probleme, die auch darin begründet liegen, dass mit sehr kleinen Zahlenwerten umgegangen werden muss [10]. Um diesen Schwierigkeiten entgegenzuwirken, werden in der Literatur vereinfachte Modelle angegeben, die das Verhalten des Memristors mit noch vertretbarer Genauigkeit beschreiben, dafür aber „stabiler“ simulierbar sind. Zwei dieser Modelle, hier „TEAM-Modell“ (Threshold Adaptive Memristor Model, [12]) und das nach Lehtonen und Laiho benannte Lehtonen-Laiho-Modell ([6]) werden nachfolgend dargestellt.

Vergleichssimulationen in [12] haben gezeigt, dass das TEAM-Modell bereichsweise eine gute Näherung des Simmons-Modells darstellt. Dabei wird die Strom-Spannungs-Beziehung am Memristor nicht mehr durch (6) und (7) nach Simmons, sondern vereinfachend durch

$$u(t) = R_{ON} e^{\frac{\lambda}{x - x_{on}}(x_{off} - x_{on})} \cdot i(t) \quad (16)$$

mit

$$e^\lambda = \frac{R_{OFF}}{R_{ON}} \quad (17)$$

beschrieben. Die Änderung der Zustandsgröße  $x$  wird beschrieben durch:

$$\frac{dx}{dt} = \begin{cases} k_{off} \left(\frac{i}{i_{off}} - 1\right)^{\alpha_{off}} \cdot f_{off} & 0 < i_{off} < i \\ 0 & i_{on} < i < i_{off} \\ k_{on} \left(\frac{i}{i_{on}} - 1\right)^{\alpha_{off}} \cdot f_{on} & i < i_{on} < 0 \end{cases} \quad (18)$$

Dabei werden  $f_{off}$  und  $f_{on}$  als Fensterfunktionen bezeichnet. Sie stellen im Modell sicher, dass die Zustandsgröße  $x$  im zulässigen Bereich verbleibt. Es gilt:

$$f_{off}(x) = \exp\left[-\exp\left(\frac{x - a_{off}}{w_c}\right)\right] \quad (19)$$

$$f_{on}(x) = \exp\left[-\exp\left(-\frac{x - a_{on}}{w_c}\right)\right] \quad (20)$$

Das Lehtonen-Laiho-Modell [6] stellt ebenfalls eine Vereinfachung des Simmons-Modells dar. Anders als das TEAM-Modell wurden seine Gleichungen jedoch nicht direkt aus dem Simmons-Modell abgeleitet, sondern durch Messungen ermittelt. Der dabei entdeckte Zusammenhang zwischen Strom und Spannung lautet:

$$i(t) = x(t)^n \beta \sinh(\alpha u(t)) + \chi [\exp(\gamma u(t)) - 1] \quad (21)$$

Wie im Simmons- und im TEAM-Modell ist  $x$  die Zustandsgröße, welche Werte zwischen Null und Eins annehmen kann. Der erste Term, der für Werte von  $x$  nahe Eins in Gleichung (21) dominiert, repräsentiert das Verhalten der Ladungsträger beim Überwinden der Tunnelbarriere, das heißt, im leitenden Zustand des Memristors. Für kleine Werte von  $x$  gewinnt der zweite Summand in (21) an Einfluss. Er beschreibt den Stromfluss durch den Memristor im gesperrten Zustand. Die Zustandsgleichung für die Änderungsgeschwindigkeit  $\frac{dx}{dt}$  wird beschrieben durch

$$\frac{dx}{dt} = a \cdot f(x) \cdot u(t)^m, \quad (22)$$

wobei  $a$  und  $m$  Parameter sind und  $m$  ungerade sein muss. Die Fensterfunktion  $f(x)$  lautet nach [2]

$$f(x) = 1 - (2x - 1)^{2p}. \quad (23)$$

## 4 Implementierung, Modellkonvergenz und Stabilität

Die beschriebenen und weitere Modelle nach [13] wurden in der gleichungsbasierten Modellierungssprache Modelica implementiert und liegen, zusammengefasst in einer Bibliothek, anwendungsbereit vor. Die Modelle sind kompatibel zu den Elektronik-Bibliotheken der Modelica-Standard-Bibliothek und können somit leicht in größeren Schaltungen verwendet und untersucht werden.

Die Implementierung wird exemplarisch anhand des **Lehtonen-Laiho-Modells** gezeigt (s. Code 1). Andere Modelle wurden in ähnlicher Weise implementiert.

```

model Memristor_Lehtonen_Laiho
    "Memristormodell according to [LeL-10]"

    extends
        Modelica.Electrical.Analog.Interfaces.OnePort;

    parameter SI.InversePotential alpha(start=2);
    parameter SI.Current beta(start=0.9e-6);
    parameter SI.InversePotential gamma(start=4);
    parameter Real ne(start=2);
    parameter Real m(start=5);
    parameter SI.Current chi(start=1e-9);
    parameter Real a(start=1);
    parameter Real P(start=1);
    Real x(start=0.9, fixed=true);
    Real f_Jog "windowfunction according to
        Joglekar";

    equation

        i = x^ne * beta * sinh(alpha*v) +
            chi * (exp(gamma*v) - 1);
        der(x) = a * f_Jog * v^m;
        f_Jog = 1 - (2*x - 1)^(2*P);

end Memristor_Lehtonen_Laiho;

```

**Code 1.** Modelica-Code des Lehtonen-Laiho-Modells

Im ersten Teil des Modells werden mittels des Schlüsselwortes `extends` elektrische Anschlüsse in das Modell geerbt. Damit sind die Memristormodelle kompatibel zu den Elektronik-Bibliotheken der Modelica-Standard-Bibliothek und können mit Elementen dieser zu größeren Schaltungen verknüpft werden. Weiterhin werden die Parameter vereinbart. Dabei folgt dem Schlüsselwort `parameter`, die Typangabe (meist in Verbindung mit einer SI-Einheit), der Name des Parameters und ein Standard-Parameterwert. Nach der Angabe der Parameter werden die Variablen vereinbart. Dazu wird der Datentyp angegeben (Real) gefolgt vom Namen der Variable und wenn erforderlich die Angabe eines Startwertes (andernfalls wird Null angenommen). Im zweiten Teil des Modells werden nach dem Schlüsselwort `equation` die Modellgleichungen angegeben. Dabei ist die Reihenfolge unerheblich, weil Modelica gleichungsbasiert arbeitet. Dies bedeutet, dass Zusammenhänge als mathematische Gleichungen behandelt werden und nicht, wie in Programmiersprachen üblich, als Zuweisungen.

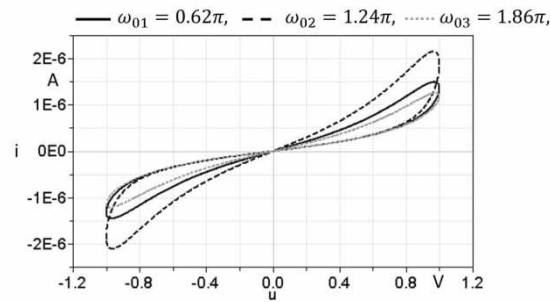
Erste Vergleiche von Simulationsergebnissen der Modelica-Implementierung des Lehtonen-Laiho-Modells mit Ergebnissen aus [12] zeigen Übereinstimmung. Für diese Vergleiche wurde das Memris-

tormodell mit einer sinusförmigen Spannung  $u = u_0 \sin(\omega_0 t)$  angeregt. Folgende Parameterwerte wurden gewählt:

Parameter	$\alpha$	$\beta$	$\gamma$	$n$	$m$	$\chi$	$a$	$p$	$x(t_0)$
Wert	2	0.9e-6	4	2	5	1e-9	1	1	0.5

**Tabelle 1.** Memristorparameter des Lehtonen-Laiho-Modells

Es wurden drei Simulationen durchgeführt mit jeweils  $u_0 = 1\text{ V}$  und den Frequenzen  $\omega_{01} = 0.62\pi$ ,  $\omega_{02} = 1.24\pi$  und  $\omega_{03} = 1.86\pi$ . Abbildung 3 zeigt die Ergebnisse.



**Abbildung 3.** Simulationsergebnis des Lehtonen-Laiho-Modells bei sinusförmiger Spannung

Auch die Modelica-Implementierung des **TEAM-Modells** zeigt bei sinusförmiger Eingangsspannung  $u = u_0 \sin(\omega_0 t)$  die für den Memristor typische Hysterese im  $i$ - $u$ -Diagramm (s. Abbildung 1). Die für die Simulation verwendeten Parameter sind Tabelle 2 zu entnehmen. Aufgrund unvollständiger Parameterangaben in [12] konnten die dort angegebenen Ergebnisse nicht nachvollzogen werden.

Parameter	$R_{ON}$	$R_{OFF}$	$x_{on}$	$x_{off}$	$i_{on}$
Wert	50	1000	1.2e-9	1.8e-9	-8.9e-6

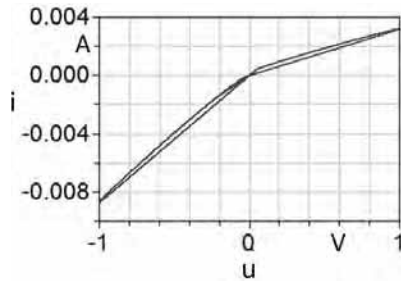
Parameter	$i_{off}$	$\alpha_{on}$	$\alpha_{off}$	$k_{on}$	$k_{off}$
Wert	115e-6	9.5	9.5	-4.68e-13	1.46e-9

Parameter	$a_{on}$	$a_{off}$	$w_c$	$x(t_0)$
Wert	1.8e-9	1.2e-9	107e-12	1.415e-9

**Tabelle 2.** Parameter des TEAM-Modells

Die Simulation wurde mit  $u_0 = 1\text{ V}$  und  $\omega_0 = 100\pi$  durchgeführt und führte zum in Abbildung 4 dargestellten Ergebnis.



**Abbildung 4.** Simulationsergebnis des TEAM-Modells bei sinusförmiger Eingangsspannung

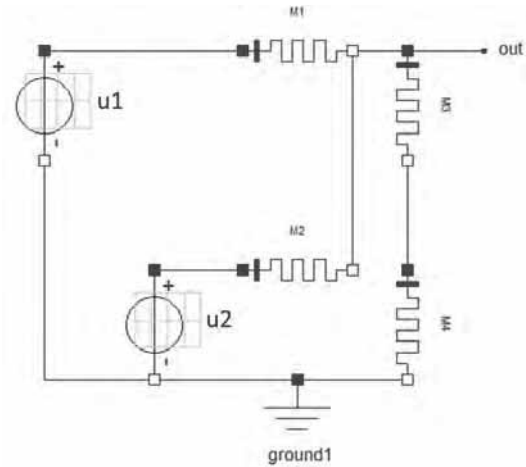
Die Simulation des **Simmons-Modells** ist problematisch. Fehlende Parameterangaben und numerische Probleme durch das Auftreten sehr kleiner Zahlen verhindern eine stabile Simulation über einen großen Bereich von Eingangsspannungen hinweg. Durch Umnormierung (nach einem Vorschlag von [4]) und Anpassen der Parameter, konnte eine deutliche Verbesserung erzielt werden. Die Anwendbarkeit beschränkt sich jedoch nur auf einen eingeschränkten Spannungsbereich und ist somit zum allgemeinen Einsatz in elektrischen Schaltungen nicht geeignet.

## 5 Memristoren in Schaltungen – Zwei Beispiele

Werden Memristoren in analogen Schaltungen anstelle von Schaltelementen wie Widerständen oder Dioden eingesetzt, kann gegenüber konventionellen Netzwerken ein neues Verhalten beobachtet werden. In [14] werden z.B. Memristoren anstelle von Dioden in einer Memristorbrückenschaltung verwendet. Damit wird das Gleichrichterverhalten der Schaltung frequenzabhängig. Im Folgenden sollen eine ODER-Schaltung und eine Operationsverstärkerschaltung mit Memristoren untersucht werden.

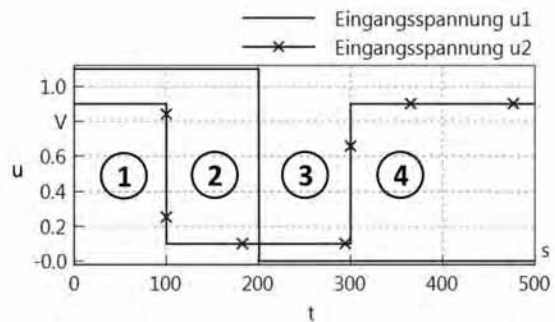
### 5.1 ODER-Schaltung

Abbildung 5 zeigt eine ODER-Schaltung, deren Verhalten für verschiedene Memristormodelle untersucht wurde.



**Abbildung 5.** ODER-Schaltung

Die verwendeten Eingangsspannungen  $u_1$  und  $u_2$  sind in Abbildung 6 dargestellt.

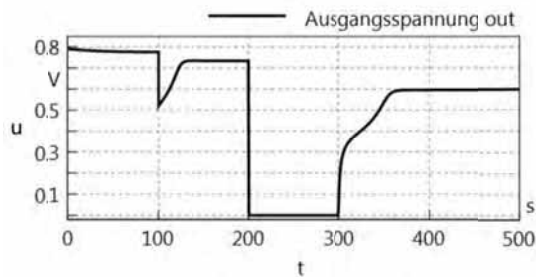


**Abbildung 6.** Eingangsspannungen  $u_1$  und  $u_2$  der ODER-Schaltung

- ① Beide Eingangsspannungen  $u_1$  und  $u_2$  liegen auf hohem Potential ( $u_1, u_2 > 0,5\text{ V}$ ). Somit sind die Memristoren M1 und M2 in Flussrichtung geschaltet und leiten. Dadurch liegt auch an M3 und M4 hohes Potential an und sie werden ebenfalls leitend. Die Gesamtmemristanz der Reihenschaltung aus M3 und M4 ist größer als die Memristanzen von M1 und M2. Daher wird der Ausgang out nicht auf Masse gezogen, sondern bleibt hochohmig.
- ② Sinkt das Potential von  $u_2$  auf low, wird der Memristor M2 in Sperrrichtung betrieben. M1 bleibt leitend und es behält auch der Ausgang out der Schaltung hohes Potential, besitzt jedoch einen etwas niedrigeren Spannungswert als im Fall ①.
- ③ Beide Eingänge liegen auf niedrigem Potential ( $u_1, u_2 < 0,2\text{ V}$ ). Der Ausgang out wird durch die zwei leitenden Memristoren M3 und M4 auf niedriges Potential gezogen.

- ④ Die Eingangsspannung  $u_2$  steigt auf hohes Potential, M1 sperrt und M2 ist leitend. Der Ausgang nimmt steigt auf hohes Potential und nimmt dem Spannungswert wie in Fall ② an.

Abbildung 7 zeigt die Ausgangsspannung  $u_{out}$  der ODER-Schaltung bei Verwendung des Memristors nach Lehtonen und Laiho.



**Abbildung 7.** Ausgangsspannung  $u_{out}$  der ODER-Schaltung

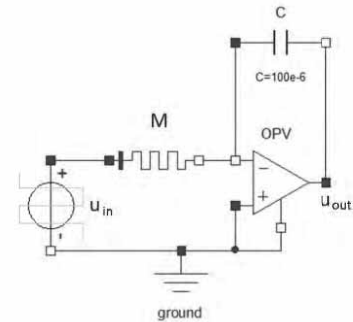
Nicht alle Memristormodelle eignen sich zu Verwendung in der betrachteten ODER-Schaltung. Sowohl bei Anwendung des Simmons-Memristors, als auch bei der des TEAM-Memristors bricht die Simulation ab. Ein Grund dafür ist das Fehlen gültiger Parametersätze für einen großen Bereich von Eingangsspannungen der Memristoren. Die Verwendung anderer Modelle dagegen, wie z.B. die des HP-Memristors oder des Yakopcic-Memristors (beschrieben in [13]) führen auf das erwartete Ergebnis. Tabelle 3 fasst die Untersuchungsergebnisse zusammen.

Schaltung	Eignung
Simmons-Memristor	Simulation divergent
Lehtonen-Laiho-Memristor	✓
TEAM-Memristor	Simulation divergent
HP-Memristor, Fensterfunktion Biolek	✓
Yakopcic-Memristor	✓

**Tabelle 3.** Eignung verschiedene Memristor-Modelle in der ODER-Schaltung

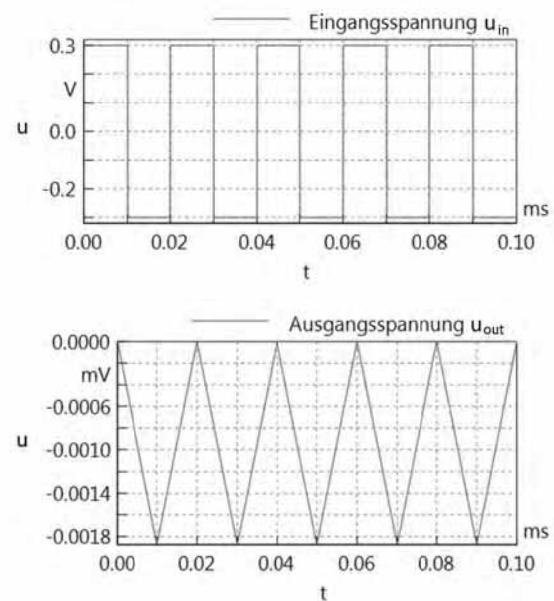
## 5.2 Integrator-Schaltung

Das zweite untersuchte Beispiel ist die in Abbildung 8 dargestellte Integrator-Schaltung, bei der der üblicherweise verwendete Widerstand durch einen Integrator ersetzt wurde.



**Abbildung 8.** Integrator-Schaltung

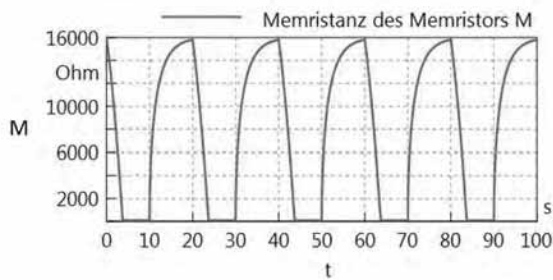
Für hohe Frequenzen der Eingangsspannung verhält sich die Schaltung wie eine übliche Integrator-Schaltung, bei der anstelle des Memristors ein linearer Widerstand verwendet wird, weil für hohe Frequenzen der Memristor zum linearen Widerstand wird. Das beschriebene Verhalten ist in Abbildung 9 unter Verwendung des in [13] vorgestellten HP-Modells mit Fensterfunktion nach Biolek gezeigt.



**Abbildung 9.** Ein- und Ausgangsspannung der Integratorschaltung bei hohen Frequenzen

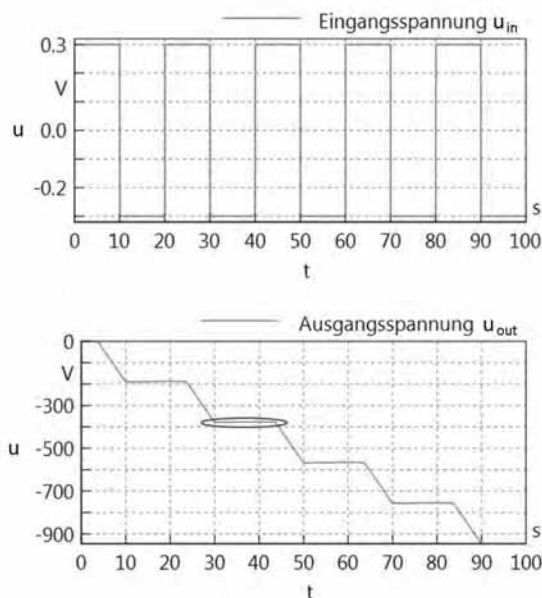
Für niedrige Frequenzen der Eingangsspannung wechselt der Memristor **M** in Abbildung 8 zwischen leitendem (niedrige Memristanz) und gesperrtem (hohe Memristanz) Zustand hin und her (siehe Abbildung 10).





**Abbildung 10.** Memristanz bei niedriger Eingangsfrequenz

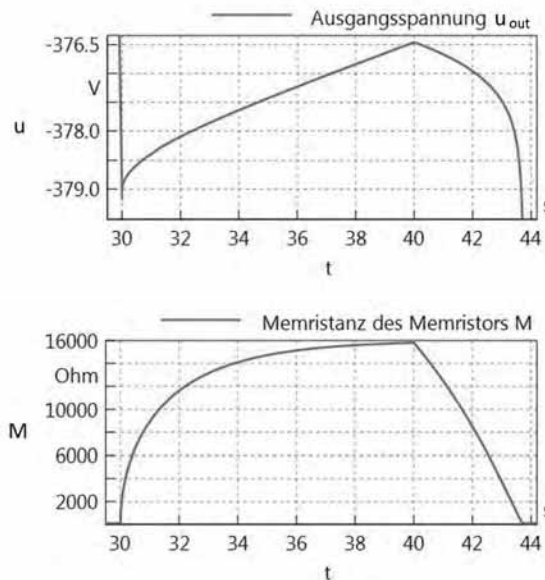
Die Eingangsspannung wird nur dann integriert, wenn sich der Memristor im leitenden Zustand befindet, also bei positiver Eingangsspannung, wobei jedoch die Übergangsintervalle für den Memristanzwechsel zu beachten sind (Abbildung 11).



**Abbildung 11.** Ein- und Ausgangsspannung der Integrator-Schaltung bei niedrigen Frequenzen

In Abbildung 12 ist der in Abbildung 11 gekennzeichnete Ausschnitt der Ausgangsspannung vergrößert gezeigt. Zum Zeitpunkt  $t = 30$  s wird die Eingangsspannung negativ und der Memristor ist in Sperrrichtung geschaltet. Die Memristanz nimmt sehr schnell und die Ausgangsspannung steigt jedoch sehr schwach an, weil der Memristor nicht vollständig sperrt. Zum Zeitpunkt  $t = 40$  s liegt am Memristor wieder eine positive Spannung an und er wird leitend. In der Ausgangsspannung zeigt sich zunächst ein nichtlinearer Verlauf, der, sobald die Memristanz ihr

Minimum (etwa bei  $t = 43,5$  s) erreicht hat, in eine Gerade übergeht.



**Abbildung 12.** Vergrößert dargestellter Ausschnitt der Ausgangsspannung und der Memristanz

Auch diese Schaltung wurde mit unterschiedlichen Memristormodelle untersucht. Tabelle 4 fasst die Ergebnisse zusammen.

Schaltung	hohe Frequenz	niedrige Frequenz
Simmons-Memristor	✓	x
Lehtonen-Laiho-Memristor	✓	bedingt
TEAM-Memristor	✓	✓
HP-Memristor, Fensterfunktion Biolek	✓	✓
Yakopcic-Memristor	✓	✓

**Tabelle 4.** Eignung verschiedene Memristormodelle in der Integrator-Schaltung

Die beschriebenen Memristormodelle und Schaltungen sind in einer Modelica-Bibliothek zusammengefasst und liegen, mit einigen Einschränkungen des zulässigen Spannungsbereichs beim Simmons-Modell, die in den in Abschnitt 3 beschriebenen numerischen Schwierigkeiten begründet liegen, anwendungsbereit vor.

## 6 Implementierung der Modelle in VHDL-AMS

Um die Modelle vielen Nutzern zugänglich zu machen, ist es wünschenswert, sie auch in VHDL-AMS zur Verfügung zu haben. Daher wird im Folgenden am Beispiel des **HP-Modells** nach [13] die Implementierung der Modelle in VHDL-AMS beschrieben.

Zunächst sollte eine ENTITY angelegt werden, in der mittels des Schlüsselwortes GENERIC die Parameter und mittels PORT die elektrischen Klemmen des Modells zur Verfügung gestellt werden (Code 2).

```
ENTITY Mem IS
    -- memristor
    GENERIC ( RINIT:REAL:=11000.0;
              RON:REAL:=100.0;
              ROFF:REAL:=16000.0;
              D:REAL:=10.0e-9;
              muev:REAL:=1.0e-14);
    PORT ( TERMINAL p1, p2:ELECTRICAL);
END ENTITY Mem;
```

**Code 2.** ENTITY des HP-Modells in VHDL-AMS

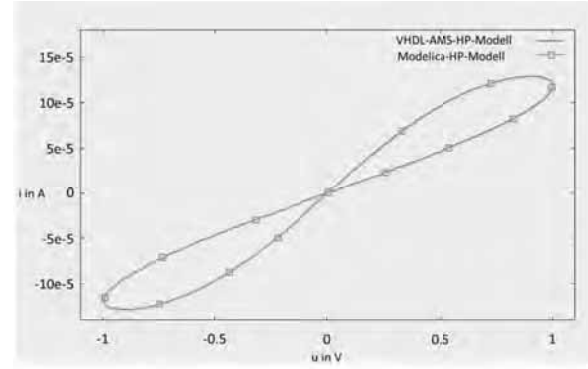
Die Definition der Variablen (mittels des Schlüsselwortes Quantity) und die Angabe der Modellgleichungen erfolgt in der ARCHITECTURE-Umgebung (Code 3).

```
ARCHITECTURE v1 OF Mem IS
    QUANTITY v ACROSS i THROUGH p1 TO p2;
    QUANTITY M : REAL;
    QUANTITY x : REAL;
BEGIN
    IF (DOMAIN = QUIESCENT_DOMAIN) USE
        x == (ROFF-RINIT)/(ROFF-RON);
    else
        x'DOT == (muev*RON*i)/(D*D);
    END USE;
    M*i == v;
    M == RON*x+ROFF*(1.0-x);
END ARCHITECTURE v1;
```

**Code 3.** ARCHITECTURE des HP-Modells in VHDL-AMS

Zum Test der prinzipiellen Funktionsweise und zum Überprüfen der Korrektheit des VHDL-AMS-Modells wurde es mit der Modelica-Implementierung des HP-Modells verglichen. Dazu wurden beide Modelle mit einer sinusförmigen Spannungsquelle nach der Form  $u = u_0 \sin(\omega_0 t)$  mit  $u_0 = 1 \text{ V}$  und  $\omega_0 = 2\pi$  angeregt. Die Memristorparameter wurden nach [3] gewählt und sind Code 2 zu entnehmen.

Abbildung 13 (u-i-Diagramm des VHDL-AMS-Modells und des Modelica-Modells) zeigt, dass für die gewählten Parameter die Simulationsergebnisse übereinstimmen.



**Abbildung 13.** Simulationsergebnisse des VHDL-AMS-HP-Modells und des Modelica-HP-Modells

## 7 Zusammenfassung und Ausblick

In diesem Beitrag wurde die Implementierung aktueller Memristormodelle in Modelica beschrieben sowie die Richtigkeit der Implementierung anhand von Vergleichsrechnungen gezeigt. Des Weiteren wurde die Eignung dieser und weiterer Modelle in zwei analogen Testschaltungen untersucht. Ferner wurde am Beispiel des HP-Memristormodells ein Weg zur Übertragung der Modelica-Modelle nach VHDL-AMS gezeigt.

In Zukunft wird es erforderlich sein, den Gültigkeitsbereich der Modelle genauer festzulegen. Dazu gehört die Ermittlung gültiger Parametersätze und Betriebsbereiche. Des Weiteren ist zu erwarten, dass weitere Memristormodelle veröffentlicht werden, die es im Sinne einer aktuellen Bibliothek in diese aufzunehmen gilt. Ziel ist es, ein Modell bereit zu stellen, was auf der einen Seite das Verhalten realer Memristoren möglichst genau abbildet, auf der anderen Seite aber robust genug ist, so dass es für viele Anwendungen eingesetzt werden kann.

Dank gebührt Dr. Joachim Haase für die Unterstützung bei der Modellierung und Simulation des HP-Modells in VHDL-AMS.

## 8 References

- [1] D. Strukov, D. Snider, D. Stewart, S. Williams: *The missing memristor found*. Nature, vol 453, S. 80-83, 2012
- [2] Y. Joglekar, S. Wolf: *The elusive memristor: properties of basic electrical circuits*. Eur. J. Phy., vol. 30, S. 661-675, 2009
- [3] Z. Biolek, D. Biolek, V. Biolková: *Spice Model of Memristor With Nonlinear Dopant Drift*. Radioengineering, vol. 18, no. 2, S. 210-214, 2009
- [4] H. Abdalla, M. D. Pickett: *SPICE modeling of memristors*. Proceedings ISCAS, S. 1832-1835, 2011
- [5] M. D. Pickett, D. B. Strukov, J. L. Borghetti, J. J. Yang, S. Snider, G. S. Steward, R. S. Williams: *Switching dynamics in titanium dioxide memristive devices*. Journal of Applied Physics, vol. 106, no. 7, 2009
- [6] E. Lehtonen, M. Laiho: *CNN using memristors for neighborhood connections*. Cellular Nanoscale Networks and Their Applications (CNNA), 2010.
- [7] X. Wang, Y. Chen, H. Xi, H. Li, D. Dimitrov: *Spintronic memristor through spin-torque-induced magnetization motion*. IEEE Electron Device Letters, vol. 30, no. 3, S. 294-297, 2009
- [8] C. Yakopcic, T. M. Taha, G. Subremanyam, R. E. Pino, S. Rogers: *A Memristor Device Model*. IEEE Electron Device Letters, vol 31, S. 1436-1438, 2011
- [9] L. O. Chua: *The Fourth Element*. Proceedings of the IEEE, vol. 100, no. 6, S. 1920-1926, 2012
- [10] G. S. Simmons: *Generalized Formula for the Electric Tunnel Effect between Similar Electrodes Separated by a Thin Insulation Film*. Journal of Applied Physics, vol. 34, no. 6, 1963
- [11] F. Garcia-Redondo, M. López-Vallejo, P. Ituerto, C. L. Barrio: *A CAD Framework for the Characterization and Use of Memristor Models*. 2012 International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuits Design (SMACD.), S. 25-28, 2012
- [12] S. Kvatinsky, E. G. Friedman, A. Kolodny, U. C. Weiser: *TEAM: ThrEshold Adaptive Memristor Model*. IEEE Transactions on Circuits and Systems, 2012.
- [13] K. Majetta, C. Clauß, T. Schmidt: *Towards a Memristor Library in Modelica*. 9<sup>th</sup> International Modelica Conference, S. 507-512, 2012
- [14] O. Papst: *Entwurf einfacher Memristiver Schaltungen für die technische Anwendung*. Studienarbeit, Fakultät Elektrotechnik und Informationstechnik, TU Dresden, 2011



# ModelOpt – ein Werkzeug zur Verbesserung von Simulationsmodellen

Ole Engel<sup>1</sup>, Ala Renz<sup>1</sup>, Friedrich Gottelt<sup>1</sup>

<sup>1</sup>XRG Simulation GmbH

*ole.engel@xrg-simulation.de*

Mit geeigneten Parametern angepasste Simulationsmodelle zeigen gute Übereinstimmung mit der Wirklichkeit. Hier wird an zwei praktischen Beispielen gezeigt, wie so eine Parameter-Optimierung mit dem Werkzeug ModelOpt durchgeführt werden kann.

## 1 Parameter-Optimierung mit ModelOpt

### 1.1 Einleitung

Simulationsmodelle bilden verschiedene Aspekte realer Komponenten und System in vereinfachter Form ab. Mit geeigneten Parametern angepasste Modelle zeigen gute Übereinstimmung mit der Wirklichkeit. Im Folgenden wird eine Methode beschrieben, mit der solche Parameter gefunden oder verbessert werden können.

### 1.2 XRG Simulation

XRG Simulation führt im Automotivebereich, in der Luft- und Schifffahrt, für Kraftwerke, Gebäude- und Verfahrenstechnik Systemsimulationen mit numerischen Methoden durch. XRG ist auf Energietechnik spezialisiert und unterstützt Industrie- und Gewerbeunternehmen sowie Forschungsinstitutionen bei der Erforschung, Entwicklung und Verbesserung von Projekten und Produkten. Darüber hinaus entwickelt XRG Simulation eigene Produkte.

### 1.3 ModelOpt

Die kommerzielle Software ModelOpt [1] ist aus einem Forschungsprojekt hervorgegangen und wurde zunächst firmenintern zur Parameter-Optimierung

von Dymola-Modellen eingesetzt. Dazu wird ein dynamisches Blax-Box-Modell mit unterschiedlichen Parametern simuliert, siehe Abbildung 2. Im Modell muss eine Größe, die Zielfunktion, berechnet werden, die mit Hilfe des Optimierungsprogramms ModelOpt ausgewertet und minimiert werden kann.

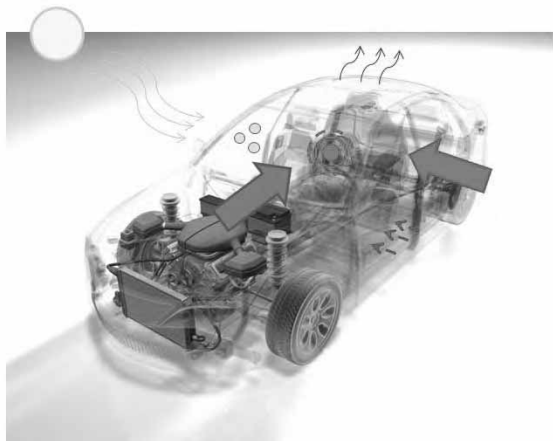


Abbildung 1: Beispiel Energieströme in einer Fahrzeugkabine

### 1.4 Beispiel Fahrzeugkabine

Ein typisches Beispiel für die Nutzung des Optimierungswerkzeugs ModelOpt ist die Anpassung eines Modells an Messwerte. Die Simulationsmodelle basieren auf physikalischen Daten, aber auch auf vereinfachenden Annahmen. Um trotz der Vereinfachungen auf plausible Ergebnisse zu kommen, werden häufig Korrekturparameter eingeführt. Für die Simulation einer PKW-Fahrzeugkabine, siehe Abbildung 1, wird im Beispiel die Luft zu einem Knoten mit mittlerer Temperatur zusammengefasst [2]. Der Strahlungsaustausch wird durch ein Modell mit nur 2 Knoten abgebildet, Abbildung 3. In dem Beispiel wird für eine dynamische Simulation die Temperatur der abgesaugten Luft für den Umluftbetrieb einer Fahrzeug-Klimaanlage mit den entsprechenden Messwerten verglichen. Die Abweichung wird als Betrag über die Dauer eines vorgegebenen Fahrzykluses aufintegriert und dieses Integral als Zielgröße benutzt. Stellparameter sind in diesem Fall die mittleren Wärmeübergangskoeffizienten und die mittlere Zahl der Reflexionen, die im Modell berücksichtigt werden.

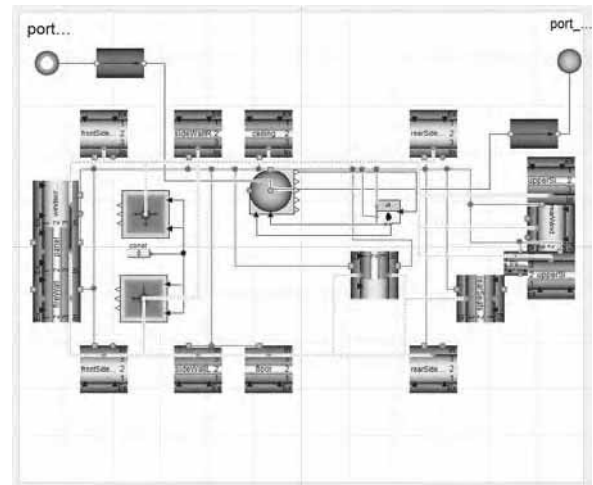
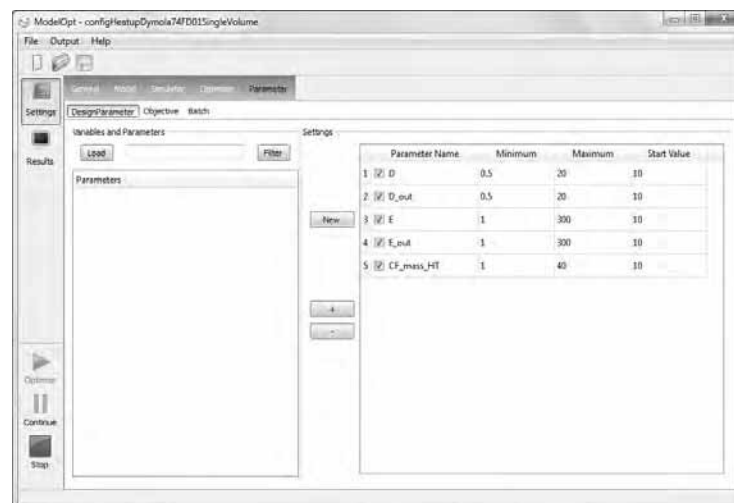


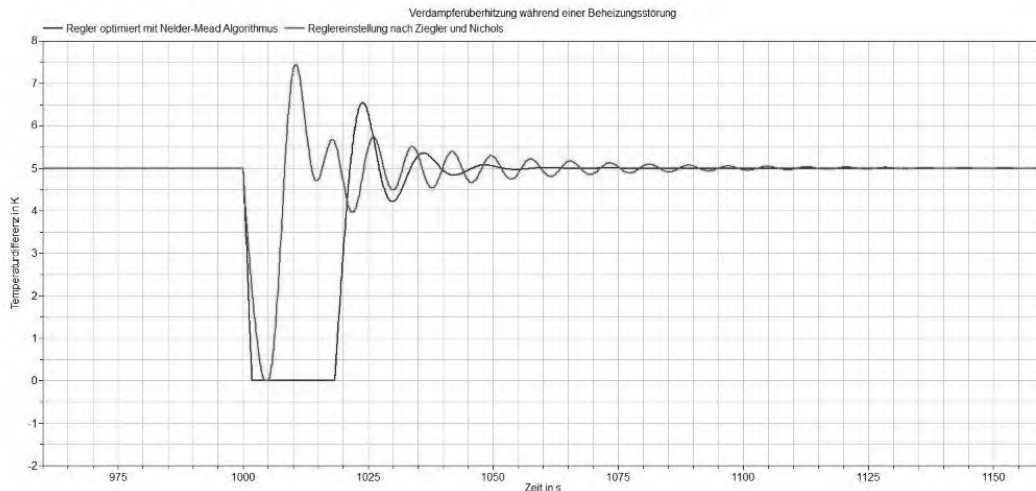
Abbildung 3: Modelica-Modell der Fahrzeugkabine mit 1 Luftknoten und 2 Strahlungsknoten

### 1.5 Beispiel Regelung eines Verdampfers

Technische Anlagen können ohne die Regelungstechnik nicht effizient und sicher betrieben werden. Die Ermittlung guter Reglerparameter für die in den Anlagen eingesetzten Regler ist oft schwierig oder mit vereinfachten Annahmen nur sehr ungenau realisierbar. ModelOpt kann auch in diesem Zusammenhang sehr hilfreich sein, was im folgenden Beispiel gezeigt werden soll.

Die Überhitzungsregelung eines Verdampfers eines Kältekreis mit dem Kältemittel R134a ist als PID-Regelung ausgeführt. In der Abbildung 4 ist das Modell zur Verdampfung des Kältemittels R134a in einem Verdampferrohr dargestellt, wobei der Kältemittelmassenstrom durch die Ventilöffnung geregelt werden kann und der zugeführte Wärmestrom als zeitlich veränderliche Störgröße dient. Das Szenario soll einen plötzlichen Abfall des Verdampferwärme-





stroms und den damit verbundenen Abfall der Überhitzertemperatur simulieren, was z.B. bei der Abschaltung einer Kabinenlüftung der Fall wäre. Die Aufgabe des Reglers ist es, durch Anpassung des Kältemittelmassenstroms die Überhitzertemperatur bei 5 Kelvin zu halten.

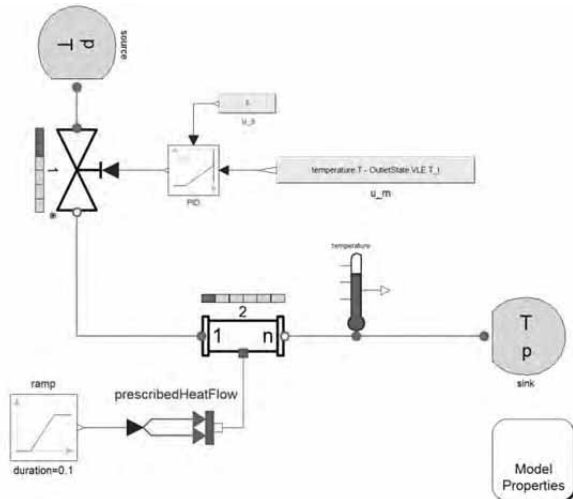


Abbildung 4: Verdampfer-Modell mit Regelung

Für dieses Beispiel wird die Bestimmung der Regler-Parameter mit dem Standardverfahren nach Ziegler und Nichols [3] mit der Optimierungsmethode unter Verwendung von ModelOpt verglichen.

Das Standardverfahren funktioniert auf die Weise, dass eine Störung in Form eines vordefinierten Wärmestromsprungs auf das System gegeben wird. Aus der Antwort des Systems und den Einstellregeln von Ziegler und Nichols können die Verstärkung, die Vorhaltezeit sowie die Nachstellzeit des PID-Reglers bestimmt werden.

Um die Reglerparameter mit ModelOpt zu bestimmen, muss man mindestens eine zu optimierende Zielfunktion und die Stellgrößen angeben. In diesem Beispiel soll die Zielfunktion, welche das Integral der Abweichung des Messsignals vom Sollwert darstellt, mithilfe der drei oben erwähnten Reglerparameter minimiert werden.

In Modelica wird die Zielfunktion folgendermaßen implementiert:

$$\text{der}(Q) = \text{abs}(\text{PID}.u_s - \text{PID}.u_m);$$

Das Ergebnis der Simulation ist in Abbildung 5 dargestellt.

Der blaue Verlauf stellt die Temperaturdifferenz zwischen Austrittstemperatur und Sättigungstemperatur oder anders ausgedrückt die Überhitzungstemperatur mit den Einstellregeln von Ziegler und Nichols und der rote Verlauf die Überhitzungstemperatur den von ModelOpt optimierten Reglerparametern dar. Es fällt auf, dass der blaue Verlauf sehr schnell abfällt und in das Nassdampfgebiet kommt und erst nach etwa 23 Sekunden wieder den Sollwert erreicht. Dagegen fällt der rote Verlauf langsamer und erreicht den Sollwert nach etwa 8 Sekunden.

Der rote schwingt länger und schwingt stärker über.

Die Dauer der Optimierung mit ModelOpt betrug 9 Minuten

Optional können weitere Zielfunktionen hinzugefügt werden, um z.B. ein zu starkes Überschwingen zu verhindern.

## 1.6 Zusammenfassung

Unterschiedliche Modelle können mit dem Werkzeug ModelOpt ohne großen Aufwand verbessert werden. An die Modelle werden keine besonderen Ansprüche gestellt, da ModelOpt mit Black-Box-Modellen arbeitet. Je nach Problemstellung, reicht es aus, die Einstell-Parameter auszuwählen, obere und untere Grenzen zu setzen und die Zielgröße festzulegen. Nur wenn keine natürliche Zielgröße vorhanden ist, muss eine Funktion im Modelica-Code hinzugefügt werden, die minimiert werden kann.

## 2 Referenzen

- [1] ModelOpt: <http://www.xrg-simulation.de/de/produkte/applications/modelopt>
- [2] St. Wischhusen. *Modelling and Calibration of a Thermal Model for an Automotive Cabin using HumanComfort Library*. Proceedings Modelica Conference, München, 2012.
- [3] H. Lutz und W. Wendt. *Taschenbuch der Regelungstechnik: mit MATLAB und Simulink*. Verlag Harri Deutsch, Deutschland, 2012.



# Effizientere Verkehrsplanung mittels Automatisierungstools

Maik Boden<sup>1</sup>, Heiko Weger<sup>1</sup>, Klaus Kabitzsch<sup>1</sup>

<sup>1</sup>Technische Universität Dresden, Fakultät Informatik, Institut für Angewandte Informatik

*Maik.Boden@tu-dresden.de*

Die Einsatzgebiete der Verkehrsplanung sind sehr vielfältig. Sie dient der Standortplanung, der Analyse von Infrastrukturprojekten oder der Prognose von Verkehrsnachfrage und deren Verteilung auf Verkehrsnetze und Verkehrssysteme. Dazu werden komplexe Verkehrsmodelle von Städten oder Regionen modelliert, welche nicht nur die Verkehrsinfrastruktur, sondern auch die Verkehrsnachfrage integrieren.

Ziel soll es sein, Verkehrsnetze möglichst effizient ändern zu können. Dazu wurden zwei Softwaretools von den Autoren entwickelt, welche sich mit der automatisierten Veränderung der Verkehrsinfrastruktur sowie der Ermittlung von Evakuierungsströmen und deren Optimierung zur Katastrophenprävention beschäftigen. Diese Tools greifen auf Verkehrsnetze zu, welche in einer führenden Verkehrsplanungssoftware modelliert und simuliert werden.

Der Beitrag stellt diese neuen Anwendungen und ihre Einsatzmöglichkeiten vor. Ausgehend vom aktuellen Stand der Forschung und Modelltheorie im Bereich der (makroskopischen) Verkehrsplanung sollen schemenhaft die Vor- und Nachteile bei der taktischen Planung oder der Wirkungsanalyse von Netzänderungen dargelegt werden. Dabei sollen auch die Schnittstellen zu mikroskopischen Modellen, welche die Analyse des Verkehrsflusses zum Kern haben, identifiziert werden.

Ausgehend von der analysierten Modelltheorie, der in der Praxis verwendeten Softwarewerkzeuge, Netzmodelle und Datenmaterial wurde ein Anforderungskatalog erstellt, welcher Grundlage für die entwickelten beiden Softwarelösungen war. Deren Anwendungsbreite und Verwendung wird als Kern dieses Beitrages anschaulich vorgestellt.

Schlagwörter: Automatisierung, Verkehrsplanung, Routenmodelle, Evakuierungsforschung

## 1 Einleitung

Die Anforderungen an die Verkehrsplanung werden zunehmend komplexer. Sie dient der Prognose der Verkehrsnachfrage auf strategischer bzw. taktischer Ebene sowie deren Auswirkung auf die zur Verfügung stehende Verkehrsinfrastruktur. Sie untersucht also den Verkehrsablauf. Dazu zählen beispielsweise die Anzahl und Art der zurückgelegten Wege einschließlich der Wahl des Ziels, des benutzten Verkehrsmittels sowie die eigentliche Routenwahl. Zur Anwendung gelangen dabei Verkehrsmodelle, welche auf makroskopischen Ansätzen beruhen.

### 1.1 Makroskopische Verkehrsmodelle der Verkehrsplanung

Diese Modellklasse widmet sich der Analyse der Verkehrsbelastungen eines Streckennetzes oder ganzer Gebiete. Mit ihr ist es damit möglich aus raumbezeichnenden Kenngrößen einer Region (Einwohner, Arbeitsplätze usw.) die resultierenden Verkehrsströme zwischen Quelle und Ziel von Ortsveränderungen zu ermitteln und zu prognostizieren. Unter Verwendung verschiedener Umlegungsverfahren wird die Nachfrage derart auf das Streckennetz bzw. die möglichen Routen verteilt, dass die gefundenen Wege den minimalen Widerstand zwischen Quelle und Ziel wiedergeben. Ergebnis sind resultierende Kantenbelastungen im Netzwerk [1].

## 1.2 Mikroskopische Ansätze (Verkehrsdynamik)

Mikroskopische Modelle hingegen widmen sich der Analyse des Fahrverhaltens der Verkehrsteilnehmer. Sie bilden in der Regel nur einen eher kleinen Zeitraum ab (maximal eine Stunde), da sie nach einer hohen Detailtreue verlangen (sowohl von der Netzstruktur mit Abbiegespuren und Lichtsignalsteuerungen als auch von Seiten der sich im System bewegenden Elemente mit Brems- und Beschleunigungsverhalten, Emissionen usw.). Sie dienen der dynamischen bzw. zeitschrittorientierten Modellierung des Verkehrs (Verkehrsflussmodellierung bzw. Verkehrsdynamik). Im Gegensatz zur Verkehrsplanung werden hier die Nachfrage zwischen Quelle und Ziel sowie die vorhandene Infrastruktur als gegeben betrachtet und sich auf die Untersuchung des Fahrverhaltens und der daraus ergebenden Effekte konzentriert (z. B. Untersuchung von Verkehrsströmen an Kreuzungen, Staueffekte, zeitliche Überlastung von Kantenelementen usw.) [2].

Die in diesem Beitrag vorgeschlagene Lösung soll ausgehend von makroskopischen Verkehrsnetzen einen Weg aufzeigen, wie eine automatisierte und effiziente Anpassung von Verkehrsmodellen die Verkehrsplaner unterstützen kann. Entscheidungsrelevant sind hier u. a. die verwendeten Modelle, Programme und Datengrundlagen.

## 2 Entwicklung der Programme

Ursächlich für die Entwicklung der in diesen Beitrag vorgestellten Tools war die Jahrhundertflut von 2002. Insbesondere die Auswirkungen auf die Verkehrsinfrastruktur und die Analyse und Prognose ähnlicher Extremereignisse im Bereich der Katastrophenprävention standen im Fokus. Es zeigte sich allerdings, dass die Netzadaption auch für kleine Verkehrsstörungen nach einer Automatisierung verlangt. Dazu sollte ein Ansatz dienen, welcher sich einerseits zur universellen Anpassung von Verkehrsnetzmodellen als auch zur Modellierung und Analyse von Extremereignissen in solchen Modellen eignet. Die notwendigen Anforderungen an die zu entwickelnden Programme sollen im Folgenden überblicksartig dargestellt werden.

### 2.1 Anforderungen

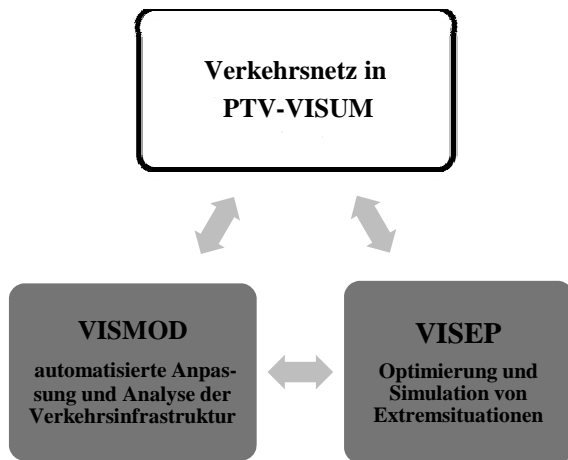
Ausgehend von der analysierten Modelltheorie, der in der Praxis verwendeten Softwarewerkzeuge, Netzmodelle und Datenmaterial wurde ein Anforderungskatalog erstellt, welcher Grundlage für die entwickel-

ten Softwarelösungen war und u. a. folgende Punkte enthält:

- Umfangreiche Anwendbarkeit auf integrierte makroskopische Verkehrsmodelle von Städten und Regionen
- Implementierung von Verkehrsstörungen durch Zuordnung von Flächen auf denen sie wirken (bspw. in Form von Shapefiles aus Geo-Informationssystemen) und Elementen im Verkehrsnetz (z. B. Kreuzungen)
- Getrennte Bearbeitung der Netztopologie (entspricht dem Verkehrsangebot) und der Nachfragedaten
- Ermittlung und Optimierung von Evakuierungsströmen bzw. Verkehrsbelastungen bei Extremereignissen
- Hohe Bedienungsfreundlichkeit und geringer Einarbeitungsaufwand

Um diesen Anforderungen möglichst gerecht zu werden, wurden die Softwarekomponenten für unterschiedliche Szenarien der Netzadaption entwickelt, welche sich mit Ansätzen der Verkehrsplanung analysieren lassen. Solche Szenarien können aus den Bereichen der Verkehrsnachfrage- und -infrastrukturanalyse (z. B. Streckensperrungen durch Baustellen) als auch der Evakuierungsforschung stammen (z. B. Identifikation nutzbarer Streckenabschnitte, optimale Evakuierungswege).

Die Umsetzung erfolgte dabei in zwei größeren Add-ons bzw. Tools. Das erste implementiert hierbei die spezielle Modellierung von Extremsituationen, das zweite stellt ein generisches Automatisierungswerkzeug für den Verkehrsplaner bei der Analyse und Veränderung von Streckenattributen dar. Beide behalten dabei kein eigenes Verkehrsmodell, sondern bauen auf PTV-VISUM zur rechnergestützten Verkehrsplanung auf (Abbildung 1) [3].



**Abbildung 1:** Interaktion der entwickelten Programme VISMOD und VISEP bei der Veränderung und Optimierung eines Verkehrsnetzes

## 2.2 Das Tool VISMOD

Das Programm VISMOD („Verkehr in Städten und Regionen **Modifizierung**“) stellt eine Softwarelösung dar, mit welcher kurzfristig eintretende Veränderungen im Verkehrsgeschehen realistisch und automatisiert abgebildet werden können. Es ermöglicht eine effiziente Adaption von Verkehrsnetzen hinsichtlich der Verkehrsangebotsseite als auch der Veränderung der Nachfragedaten nach verschiedenen Kriterien. So sind u. a. eine streckentypfeine Veränderung der zugelassenen Verkehrssysteme, ein Auffinden der durch Sperrungen betroffenen Linienrouten des Öffentlichen Verkehrs, eine automatisierte Generierung neuer Anbindungen oder eine individuelle Nachfragebearbeitung möglich. Das Programm steuert dabei über eine COM-Schnittstelle ein in PTV-VISUM generiertes Verkehrsmodell an.

Das Programm VISMOD ist universell auf verschiedene Verkehrsnetze anwendbar. So wird gewährleistet, Verkehrsnetze effizient und mit geringem Aufwand anpassen zu können. Änderungen im Verkehrsgeschehen können bspw. durch Sperrungen aufgrund von Großveranstaltungen, Katastrophen, Bombenfunden oder auch nur durch Baustellen ausgelöst werden.

Im einfachsten Fall kann die Sperrung ein kreisförmiges Gebiet betreffen, ebenso ist aber auch die Sperrung z. B. eines Häuserblocks oder größerer Gebiete möglich. Den unterschiedlichen Szenarien wird eine Flächenart zugeordnet, wobei der Anwender zwischen Kreisflächen um Netzknoten (bspw. um eine Kreuzung als Quelle der Verkehrsstörung), Flächen von Verkehrsbezirken sowie PTV-VISUM-Gebieten

zur Darstellung komplexerer Bereiche wählen kann (z. B. Überflutungsgebiete oder Häuserblocks). Für diese definierte Fläche werden die betroffenen Strecken gefunden und deren zulässige Verkehrssysteme in Abhängigkeit zum jeweiligen Streckentyp verändert.

Es lassen sich durch das Programm die Verkehrssysteme sowohl streckentypfein sperren als auch freigeben. Die durch VISMOD gefundenen Strecken werden nach Abschluss der Berechnungen in einem Protokoll mit den Veränderungen der zulässigen Verkehrssysteme ausgegeben und können gespeichert oder ausgedruckt werden.

Abbildung 2 stellt die Programmoberfläche des entwickelten Programms bei der Angebotsanpassung eines Verkehrsnetzes dar. Nachdem der Anwender die zu verändernde Fläche definiert hat, werden die zugehörigen Netzelemente aus PTV-VISUM gelesen und vom Anwender ausgewählt. Danach werden die Streckentypen definiert, in denen die Verkehrssysteme gesperrt oder zugelassen werden sollen.



**Abbildung 2:** Anpassung der Verkehrsinfrastruktur (Angebotsanpassung) eines Verkehrsnetzes mit Hilfe des entwickelten Programms VISMOD

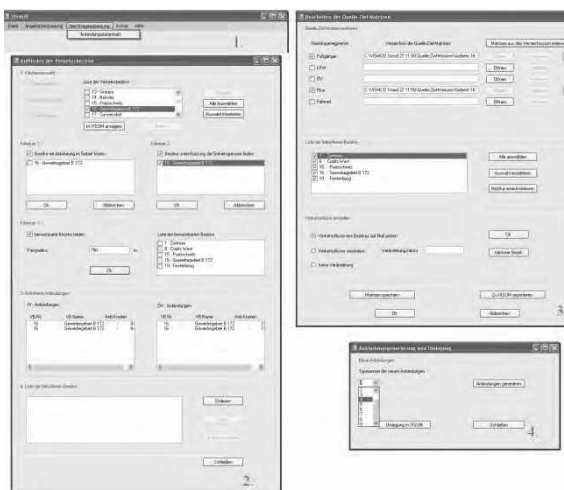
Außer der Veränderung des Verkehrsangebots wurde auch eine Anpassung der Verkehrsnachfrage realisiert. Verkehrsbezirke als Quelle und Ziel von Ortsveränderungen werden nach verschiedenen Kriterien gefunden und die hinterlegten Nachfragedaten verändert. Dafür können alle Quell- und Zielverkehrsströme eines erkannten Verkehrsbezirks verändert oder einzelne Quelle-Ziel-Relationen angepasst werden.

Weiterhin besteht die Möglichkeit, sich fehlerhafte Anbindungen innerhalb des betroffenen Gebiets getrennt nach Anbindungen des Öffentlichen bzw. Indi-

vidualverkehrs aufzeigen zu lassen und neue Anbindungen zu generieren.

Netzänderungen können sich aber auch auf die Nachbarbezirke in der Nähe des Ereignisses auswirken. Diese sind zwar nicht direkt betroffen (alle Wege zu den Nachbarbezirken stehen bspw. noch zur Verfügung), aber es kann sich auch für diese Bezirke die Nachfrage ändern. Das Programm findet hierzu unter Vorgabe eines Radius die Nachbarbezirke heraus und ermöglicht die Bearbeitung dieser Bezirke.

Die Nachfrageanpassung soll die folgende Darstellung (Abbildung 3) verdeutlichen. Nach Definition der Flächenauswahl (nur erforderlich, wenn vorher keine Angebotsanpassung durch den Anwender vorgegeben wurde, anderenfalls wird die Auswahl aus der Angebotsanpassung übernommen) erfolgt die Definition nach welchen Kriterien die Bezirke gefunden und ob Nachbarbezirke aufgezeigt werden sollen. Danach werden die Nachfragedaten gelesen und können für jedes Nachfragesegment und jeden betroffenen Bezirk auf verschiedene Weisen angepasst werden.

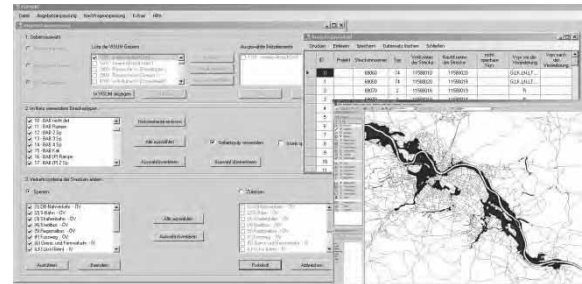


**Abbildung 3:** Veränderung der Nachfragedaten in Form von Verkehrsbezirken an einem Netzbeispiel

Nachdem die Matrizen durch das Programm gespeichert und in das Verkehrsnetz exportiert wurden, können neue Anbindungen generiert werden. Diese werden erzeugt, um Probleme bei der Umlegung aufgrund fehlerhafter Anbindungen vermeiden zu können.

Je nach Verwendungszweck von VISMOD kann entweder eine Anpassung der Verkehrsinfrastrukturnutzung (Angebotsanpassung) oder nur die Änderung der Nachfragedaten erfolgen. Dazu wurden die beiden

Schritte Angebots- und Nachfrageanpassung als eigenständige Programmteile realisiert [4], [5].



**Abbildung 4:** Anwendungsbeispiel: Anpassung der Verkehrsinfrastruktur des Verkehrsmodells der Stadt Dresden an den Pegelhöchststand der Elbe vom 17.08.2002

### 2.3 Das Tool VISEP

Mit dem Programm VISEP („Verkehr in Städten und Regionen Evakuierungsplan“) kann der Anwender reale oder fiktive Evakuierungsszenarien abbilden. So können Evakuierungsnachfragen berechnet und das Verkehrsnetz der betroffenen Region durch verschiedene Maßnahmen beeinflusst werden. Das Programm wurde wie VISMOD in Visual Basic .Net erstellt und verfügt über eine eigene Programmoberfläche.

VISEP dient vor allem der Modellierung großräumiger Evakuierungsszenarien. Im Vordergrund stehen die Berechnung der Evakuierungsnachfrage, die Abbildung der berechneten Evakuierungsströme im Verkehrsnetz sowie die automatisierte Beeinflussung der vorhandenen Infrastruktur anhand verschiedener Maßnahmen zur Evakuierungsbeschleunigung.

Die Berechnung der Verkehrsnachfrage des jeweiligen Evakuierungsszenarios erfolgt auf der Grundlage von Verkehrsstrommatrizen (Personenfahrten eines Werktages) oder Fahrtenmatrizen, die mit der Software PTV-VISEVA oder PTV-VISUM berechnet wurden. Unabhängig von diesen Inputdaten kann VISEP die Evakuierungsnachfrage jedoch auch mittels vorgegebener Bevölkerungsmerkmale des Untersuchungsraumes in Form einer Raumstrukturdatei ermitteln.

Die Anpassung des Verkehrsangebotes und der Verkehrsnachfrage des Nichteвакуierungsverkehrs erfolgt unter Verwendung von VISMOD.

VISEP ist in drei Hauptbereiche gegliedert:

1. **Die Szenariodefinition.** Hier werden der Startpunkt und das Ziel der Evakuierung definiert. Dazu stehen dem Anwender unter-

schiedliche Möglichkeiten die Evakuierungsfläche zu definieren zur Verfügung. Über einen Reduzierungsfaktor kann die Anzahl der zu evakuierenden Personen gesenkt und unter Verwendung von Kenngrößen des Verkehrsnetzes (z. B. Reisezeit zwischen Verkehrsbezirken) eine optimale Verteilung auf die Ziele ermittelt werden.

## 2. Die Evakuierungsnachfrageberechnung.

Strecken innerhalb eines zu evakuierenden Gebietes sollten nur für Verkehrssysteme freigegeben sein, die für die Evakuierung verwendet werden. Der Normalverkehr sollte hingegen das Gebiet umfahren. Dazu werden in VISEP eigene Verkehrssysteme und Nachfragesegmente zum Evakuierungsablauf angelegt. Der Anwender kann auswählen, auf welchen Routen diese Verkehrssysteme erlaubt sind. Für diese neuen Nachfragesegmente werden die Nachfragedaten durch VISEP ermittelt und als Fahrtenmatrizen ausgegeben. Da die Datenlage sehr unterschiedlich sein kann, besteht die Wahlmöglichkeit zwischen einer Raumstrukturdatei, Verkehrsstrommatrizen oder Fahrtenmatrizen. Wobei erstere Informationen über die Soziodemographie des Untersuchungsgebietes enthalten (z. B. Einwohner, Arbeitsplätze einer Region bzw. eines Verkehrsbezirkes), die Matrizen hingegen Informationen über die eigentlichen Quelle-Ziel-Relationen bereitstellen.

3. **Die Optimierung.** Um eine minimale Reisezeit für den Evakuierungsverkehr zu erhalten, ohne dass der Normalverkehr behindert wird, stehen zwei Möglichkeiten der Infrastrukturbeeinflussung zur Verfügung: Zum einen wird auf den Evakuierungsrouten geprüft, ob die Gegenfahrbahnen für den Evakuierungsverkehr freigegeben werden können, um somit diese Kapazität für die Evakuierung nutzbar zu machen. Zum anderen besteht die Möglichkeit, die Abbiegevorgänge des Normalverkehrs auf die Evakuierungsrouten zu unterbinden. Als Optimierungskriterium dienen die Längen aller Evakuierungsstrecken und die Gesamtreisezeit, jeweils gewichtet mit der Gesamtzahl der Evakuierungsfahrzeuge. Als Optimierungsverfahren stehen der Greedy-Algorithmus

und das Simulated-Annealing-Verfahren zur Verfügung. Der Greedy-Algorithmus verwirft sofort schlechtere Alternativen, wohingegen das Simulated-Annealing schlechtere Alternativen mit einer bestimmten Wahrscheinlichkeit akzeptiert. Beispielsweise kann eine einzelne Streckensperrung die Evakuierungszeit erhöhen, wird aber ein weiterer Bereich gesperrt, kann es zur Reduzierung der zur Evakuierung notwendigen Zeit kommen. Dem Nutzer werden die Resultate schließlich in Form eines Routenplans aufbereitet. Dieser enthält Vorschläge zur Infrastrukturanpassung, die Anzahl der zu evakuierenden Personen und die Verteilung auf die optimalen Evakuierungsrouten.

Die Programmoberfläche von VISEP bei der Ermittlung der Evakuierungsnachfrage zeigt Abbildung 5.

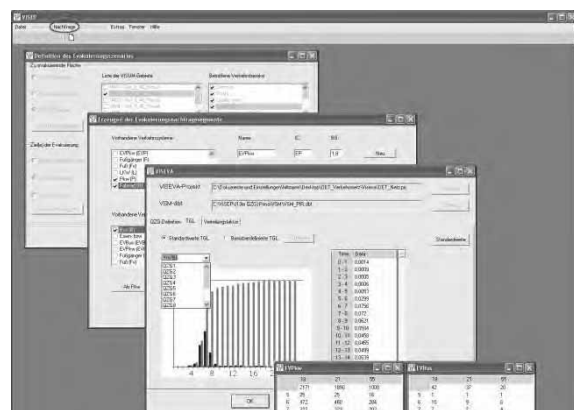


Abbildung 5: Die Programmoberfläche von VISEP bei der Ermittlung der Evakuierungsnachfrage

Die Optimierung der Evakuierungsrouten wird in Abbildung 6 veranschaulicht. Dargestellt sind beispielhaft die Definition der Quellen und Senken der Evakuierung, das Erzeugen und Implementieren der zur Evakuierung zu verwendenden Verkehrssysteme, die Zuordnung und Berechnung der Nachfragedaten der Evakuierung sowie die Wahl der Veränderung der Verkehrsinfrastruktur und der Optimierungsstrecken [6].



Abbildung 6: Die Programmoberfläche von VISEP bei der Optimierung der Evakuierungsverkehre

### 3 Diskussion

Die dargestellten Tools zur automatisierten Adaption von makroskopischen Verkehrsnetzen haben ihre Eignung an verschiedenen Verkehrsmodellen und Szenarien unter Beweis stellen können.

Innerhalb der entwickelten Tools wird auf die in den jeweiligen Netzmodellen integrierten und kalibrierten Umlegungsverfahren zurückgegriffen, welche mittels COM angesteuert werden. Somit ist es möglich, auch neue Umlegungsalgorithmen, die in PTV-VISUM implementiert wurden, zu berücksichtigen. Diese Flexibilität hilft dem Verkehrsplaner „seine“ Umlegungsansätze in PTV-VISUM zu kalibrieren und auszuführen. Die entwickelten Add-ons wiederum nutzen sie als exogene Variable, müssen daher keine neuen Algorithmen integrieren und können gleichwohl neue, in PTV-VISUM implementierte Ansätze für die Optimierung nutzen.

Die in diesen Verkehrsnetzen ermittelte Streckenbelastung beruht allerdings i. d. R. auf statischen Umlegungsalgorithmen wie dem Gleichgewichtsverfahren. Dabei werden die Routen im Netz derart verteilt, dass ein Wechsel der Route zu einer Reisezeiterhöhung für den Verkehrsteilnehmer führen würde. Dabei kommt das 1. Wardrop'sche Prinzip (Nutzergleichgewicht) zum Tragen, welches postuliert, dass alle benutzten Routen eine minimale Reisezeit ausweisen, alle unbenutzten eine höhere [7]. Diese Modelle gehen neben der Kenntnis der Verkehrsteilnehmer über Alternativrouten auch von einer konstanten Verteilung der Inputdaten über die Zeit aus. Auch eine Berücksichtigung von Rückstauwirkungen unterbleibt, da hier davon ausgegangen wird, dass alle Fahrten abgeschlossen sind, das Netz somit „geräumt“ ist. Dies erscheint im Sinne von plötzlich auftretenden Überlastungen als problematisch. Sind die Randbedingungen des Netz-

werkes durch Schwankungen in der Nachfrage gegeben (bspw. bei einer Evakuierung) und möchte man die Robustheit des Transportsystems untersuchen, so sind dynamische Modelle zwingend notwendig.

Der dynamische Umlegungsansatz betrachtet dabei nicht das „Verhalten“ der einzelnen Verkehrsteilnehmer untereinander (z. B. Brems- oder Beschleunigungsverhalten), sondern erweitert die statischen Ansätze der makroskopischen Verkehrsplanung um eine zeitliche Komponente. Beispiele sind das dynamische Gleichgewichtsverfahren und die dynamische stochastische Umlegung [8], [9].

Die Integration dieser Algorithmen stellt die künftige Herausforderung bei der Weiterentwicklung der Programme durch die Autoren dar.

### 4 Zusammenfassung

Die in diesem Paper dargestellten Programme können einen entscheidenden Baustein bei der Automatisierung der angewandten Verkehrsplanung sein. Sie eignen sich neben der Streckenmanipulation auch zur strategisch-taktischen Planung möglicher Großschadenslagen im Rahmen der Katastrophenprävention. Neue Umlegungsverfahren oder Ansätze der Netzmanipulation werden die Anwendungsfelder der Programme zukünftig noch erweitern bzw. ausbauen.

### 5 Danksagung

Die Autoren möchten sich für die Unterstützung bei der PTV AG Karlsruhe für die Bereitstellung der Verkehrsplanungssoftware bedanken.

### 6 Literatur

- [1] W. Schnabel und D. Lohse. *Grundlagen der Straßenverkehrstechnik und der Verkehrsplanung*, Band 2, Verlag für Bauwesen, Berlin, 1997.
- [2] M. Treiber und A. Kesting. *Traffic Flow Dynamics, Data, Models and Simulation*. Springer, 2012.
- [3] PTV Group. *Wer behält im Verkehr den Überblick?* Broschüre PTV VISUM, Karlsruhe 2012.
- [4] M. Boden und H. Weger. *Entwicklung von Tools für eine effizientere Verkehrsmodellierung in makroskopischen Verkehrsnetzen*, 22.

- Verkehrswissenschaftliche Tage, Technische Universität Dresden, 2009.
- [5] M. Boden, L. Buzna and H. Weger. *Simulation of Evacuation Scenarios in Urban Regions – Developing Tools for Modeling and Optimization of Evacuation Traffic Flows*, Transcom 2007, University of Žilina (SK), 2007.
- [6] M. Boden und H. Weger. *Simulation von Verkehrsstörungen und Optimierung von Evakuierungsströmen am Beispiel der Jahrhundertflut 2002*, 4. KatNet-Tagung - Netzwerk zur Katastrophenforschung, Lutherstadt-Wittenberg, 2008.
- [7] J. G. Wardrop and J. I. Whitehead. *Correspondence. Some Theoretical Aspects of Road Traffic Research*. ICE Proceedings: Engineering Divisions1 (5): 767, 1952
- [8] G. Bellei, G. Gentile and N. Papola. *A within-day dynamic traffic assignment model for urban road networks*. Transportation Research Part B 39, 1–29, 2005.
- [9] G. Gentile, L. Meschini and N. Papola. *Spill-back congestion in dynamic traffic assignment: A macroscopic flow model with time-varying bottlenecks* Transportation Research Part B 41, 1114–1138, 2007.





# Applicability of rescheduling strategies in tram networks

Daniel Lückcrath<sup>1,2</sup>, Oliver Ullrich<sup>1</sup>, Ewald Speckenmeyer<sup>1</sup>

<sup>1</sup>Institut für Informatik, Universität zu Köln

<sup>2</sup>Institut für Nachrichtentechnik, Fachhochschule Köln

*lueckerath@informatik.uni-koeln.de*

Highly utilized tram networks, where multiple lines share tracks and stations, are inevitably affected by disturbances during daily operation. While consequences of small, local perturbations may be counteracted by schedule characteristics, e.g. robustness, long lasting disturbances have to be addressed by dispatchers via schedule adjustments.

Several methods for the identification and assessment of different rescheduling actions have been proposed. However, most of these methods have only been applied in railway networks. Therefore, in this paper we compare different rescheduling strategies and assess their applicability in tram networks.

This paper begins with a description of possible rescheduling actions and the requirements and limitations to rescheduling strategies in tram networks. Different strategies for railway networks are then described and compared in regard to their applicability in tram networks.

## 1 Introduction

Tram networks are frequently affected by disturbances, many of which are inevitable. Especially highly utilized networks, where several lines share tracks and stations, are prone to perturbations.

While the consequences of small, local disturbances can be counteracted by considering schedule characteristics like robustness during the planning stage, long lasting perturbations call for direct intervention by dispatchers during daily operation, e.g. via schedule adjustments.

The method of adjusting a schedule during daily operation is called rescheduling. It aims for the generation of transitional schedules in order to cope with the impacts of major perturbations. To generate those temporary schedules dispatchers may change dwell times of vehicles at stations, travel times between stations or even whole routes of vehicles through the network. Because the schedule adjustments are carried out during daily operation, rescheduling is time-critical, i.e. dispatchers have only a limited time frame to devise and evaluate possible schedule adjustments without causing further disturbances. At the same time the resulting schedule should also be robust and allow for reinstatement of the original time table after the perturbation subsided.

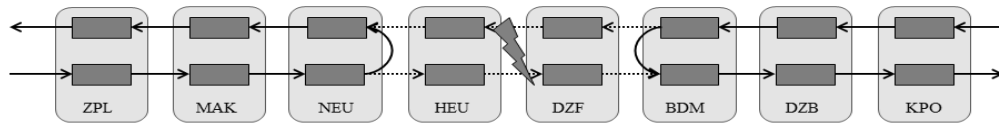
Thus, there is a need for methods and tools that help network operators decide which rescheduling actions to take and how these actions will affect tram opera-

tion. To address this issue we want to broaden the scope of our project *Computer Aided Traffic Scheduling (CATS)*. Up until now we developed simulation and optimization methods to generate and evaluate robust time tables, which adhere to transport planning requirements (see [7, 12]). We now want to employ our simulation software during daily operation, utilizing the underlying parallelization framework (see [11]) to provide dispatchers with a tool set for quick evaluation of different rescheduling actions.

Several rescheduling strategies have been proposed (see [1, 2, 3, 4, 5, 6, 8, 9, 10, 13]). However, most of them have only been applied to railway networks. Because tram and railway networks differ in multiple aspects, e.g. schedule density or network size, rescheduling strategies that work in railway networks may not be applicable in tram networks.

Thus, in this paper we conduct our initial comparison of different rescheduling strategies for railway networks and assess their applicability in tram networks.

The paper continues with a description of the requirements and limitations to rescheduling strategies for tram networks (section 2). We then present and evaluate some rescheduling strategies for railway networks (section 3), which are afterwards compared for their applicability in tram networks (section 4). The paper closes with a short summary of lessons learned and some remarks on further research (section 5).



**Figure 1.** Example for route separation. Small gray/red rectangles depict platforms, while big blue rectangles with rounded edges show how platforms are joined into stations. Dotted arrows indicate the regular route

## 2 Requirements and limitations to re-scheduling strategies for tram networks

To assess the applicability of existing railway re-scheduling strategies we identify requirements and limitations to tram rescheduling strategies, which result from differences in network and schedule design between both systems.

Besides the mere number of stations and the length of tracks, differences in network design relate to the size of stations, track redundancy and the number of track switches that join different parts of the network. While stations in railway networks often consist of multiple platforms which may be approached by vehicles from different directions, most stations in tram networks consist only of two platforms (one for each direction). Therefore, re-routing of vehicles within stations, i.e. dynamic platform assignment, is not applicable for most tram networks. A similar observation can be made for tracks: In highly utilized areas of railway networks often multiple tracks for the same direction exist, e.g. one track for commuter trains and another for freight trains or long distance trains which do not stop at every station. In tram networks, on the other hand, there exists at most one track for each direction, which makes it impossible for vehicles to overtake a slower, damaged vehicle. Lastly, in most tram networks the number of locations where trams can switch from one route to another is small, limiting the degree to which redirection (e.g. due to a blocked route) is practicable.

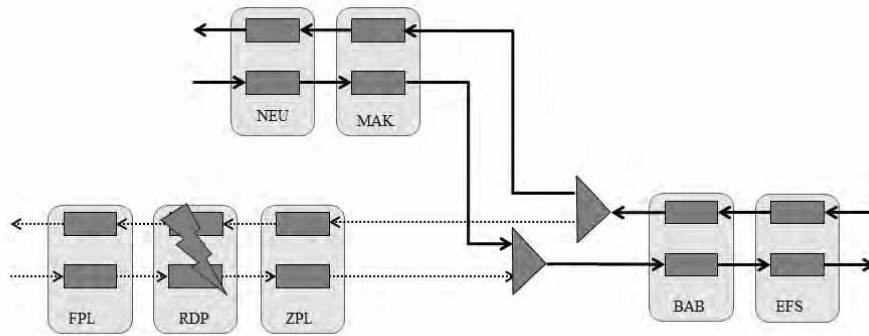
In regard to commuter trams/trains both systems employ periodic time tables, but schedules for tram networks usually have a smaller tact interval than schedules for railway networks (e.g. ten minutes versus 60 minutes). Thus, the safety distance between successive vehicles is tighter in tram systems, resulting in shorter dwell times at stations and smaller time frames for dispatchers to devise schedule adjustments. Should the employed rescheduling action take

longer than the safety distance, the follow-up vehicle will also be affected by the disruption and further rescheduling actions have to be undertaken.

As a result of these limitations the following rescheduling actions seem most applicable in tram systems:

- I. Separation of a line route into two partial routes in order to avoid blocked tracks or stations. For example the route of Cologne's line 9 could be separated by turning vehicles around at station Deutz/Messe (BDM) for the west-bound variant and at station Neumarkt (NEU) for the east-bound variant, thus omitting hypothetically blocked stations Deutzer Freiheit (DZF) and Heumarkt (HEU) (see figure 1).
- II. Shortening of routes at stations where vehicles can turn around. This is a special case of route separation, where only one partial route is serviced.
- III. Redirection of vehicles at track switches where different lines from varying directions meet, e.g. Cologne's line 12 could be redirected at station Zülpicher Platz (ZPL) and travel along the route of line 9 to station Neumarkt (see figure 2).
- IV. Adjusting arrival and departure times at relevant locations in the network (e.g. platforms or track switches) to adjust the tram order at the (next) joining track switch.

In addition to providing these actions, a feasible approach must also be able to handle realistic problem instances in acceptable time. As a reference point for this we choose Cologne's tram network of 2001, which consists of 528 platforms and 58 track switches connected via 584 tracks. 15 lines with 182 line routes are served by 178 vehicles which execute 2,814 trips per operational day. At most inner city platforms this results in a safety distance of two minutes, limiting the available computational time.



**Figure 2.** Example for vehicle redirection. Triangles depict track switches, small gray/red rectangles platforms and big blue rectangles with rounded edges show stations. Dotted arrows indicate the regular route

### 3 Rescheduling in railway systems

There exist several different approaches to rescheduling (see [1, 2, 3, 4, 5, 6, 8, 9, 10, 13]), some even considering bimodal traffic systems (see e.g. [5, 6, 13]). Because an exhaustive review of all approaches would go beyond the scope of this paper, we will only review those which seem most promising for applicability within tram networks and consider only one traffic system.

#### 3.1 D'Ariano and Pranzo

D'Ariano and Pranzo in [4] describe an extension to the real-time dispatching system *ROMA (Railway traffic Optimization by Means of Alternative Graphs)* for short-term prediction of railway traffic under strong disturbances. Their objective is to evaluate the effects of rescheduling actions for a given time horizon. Because of the complexity of the problem they decompose the time period under examination into smaller, tractable time intervals which are solved in cascade. The output of each subproblem (i.e. position and speed of the vehicles at the end of the corresponding time interval) is used as input constraint for the subsequent time interval.

To model the railway network the authors use stations, signals and block sections (a track segment between two signals which may host at most one vehicle at a time). The movement of all vehicles during a given time horizon is defined by the schedule, which specifies, for each train, planned arrival/passing times at relevant locations along its route (e.g. stations or track switches).

Within each tractable time interval D'Ariano and Pranzo solve the following three problems:

1. Finding a feasible route for each vehicle without using already occupied tracks.
2. Scheduling train orders and exact arrival/departure times at stations as well as at relevant locations in the network (e.g. track switches).
3. Ensuring a minimum safety distance between vehicles while maintaining acceptable speed profiles.

To solve problem 1 the applied software module checks if there are different possible routes for a vehicle to use and whether those are already occupied or not. If no feasible route can be assigned external support by the dispatcher is requested.

Problem 2 is formulated as a job shop scheduling problem using the alternative graph formulation (see [3]). This formulation requires that a route for each vehicle is given (i.e. problem 1 is solved) and traversing times for tracks are known in advance. If conflicts between trains arise, a passing order must be defined. This is done by either using a *Branch and Bound* algorithm or a simple *First Come First Served* dispatching rule.

To solve problem 3 ROMA checks the compatibility of the schedule with the current vehicle dynamics and signal states and if necessary adjusts the vehicle speed profiles. These two steps are performed until a feasible schedule with acceptable speed profiles is obtained.

By solving those three problems the resulting application can be used to perform all rescheduling actions from section 2. The solution to problem 1 corresponds to performing rescheduling action I, II or III, while defining the passing order and scheduling the

exact arrival and departure times at stations (i.e. solving problem 2) corresponds to rescheduling action IV. Finally, changing the speed profiles while solving problem 3, indirectly corresponds to performing rescheduling action IV.

D'Ariano and Pranzo apply the resulting software to the route Utrecht – Den Bosch in the Dutch railway network. This area consists of 191 block sections and 21 platforms. The employed time table has a tact interval of 60 minutes and up to 40 vehicles are scheduled. Depending on the length of the time horizon for which traffic predictions are done, ROMA takes between 14 seconds (one hour time horizon) and 787 seconds (nine hours) to solve the problem. It has to be noted, that the runtime does not increase linear with the time horizon. For example with a time horizon of two hours the application needs 503 seconds to solve the problem, which is too long for dense tram schedules, where the safety distance between vehicles is only two minutes. Thus, the method is either only applicable for small networks with few vehicles or its runtime has to be decreased, e.g. through solving the problem in parallel for separated parts of the network.

### 3.2 Corman et al.

Corman et al. in [2] present a solution strategy for the *Bi-objective conflict detection and resolution (BCDR)* problem, which deals with finding a set of non-dominated schedules that minimize train delay as well as the number of missed connections.

The authors formulate the BCDR problem as an alternative graph and determine the *Pareto front* (i.e. the maximal set of non-dominated solutions) by iteratively solving the *conflict detection and resolution (CDR)* problem for different sets of enforced train connections. To solve the CDR problem the authors employ the Branch and Bound algorithm described in [3], which minimizes train delay.

As D'Ariano and Pranzo in [4] the authors assume that the traveling time of vehicles can be determined in advance, i.e. trains travel at their scheduled speed whenever possible, recovering small delays by using buffer times inserted in the time table. In addition they use the same approach as D'Ariano and Pranzo to model the railway network and the schedule.

To approximate the Pareto front Corman et al. describe two algorithms, named *Add* and *Remove*. *Add* starts with the schedule solution that maintains none

of the possible train connections and generates new solutions by adding enforced connections. *Remove* on the other hand starts with the schedule solution that maintains all train connections and generates new solutions by removing connections.

Because it solves the CDR problem the resulting application is at least able to perform rescheduling action IV from section 2. Given the information contained in [2] and [3] the software seems not to be able to adjust the routes of vehicles (i.e. performing rescheduling actions I – III).

While preserving connections may be crucial in railway networks with long tact intervals, this objective is less crucial in tram networks with dense schedules where passengers do not have to wait long for the next vehicle. However, as soon as connections between the tram network and other traffic systems (e.g. long distance train) have to be considered this once again becomes crucial.

Like D'Ariano and Pranzo, Corman et al. conduct computational experiments based on the Dutch railway network area around Utrecht, employing a time table with a tact interval of 60 minutes, which schedules up to 80 trains. Depending on the number of enforced connections and the severity of inserted perturbations, algorithm *Add* solves the problem in 166 to 309 seconds, while algorithm *Remove* takes between 283 and 705 seconds. Like the approach of D'Ariano and Pranzo this seems to be too long for dense tram schedules, but could very well be accelerated by parallelizing the implementation.

### 3.3 Törnquist Krasemann

Törnquist Krasemann in [9] proposes a greedy algorithm for rescheduling during daily operation which generates good-enough feasible schedules, independent of the underlying disturbance scenario, within 30 seconds. The algorithm is a complement to a previous approach (see [8, 10]) which formulates the rescheduling problem as a mixed integer linear program, but is not able to find good solutions for some disturbance scenarios and a time horizon longer than 60 minutes within acceptable time. Furthermore, resulting from a recent analysis of the infrastructure under examination the new algorithm includes the possibility to consider routing of vehicles within stations.

To model the railway network Törnquist Krasemann divides it into line sections and station sections. Each section has up to  $n$  parallel tracks and line sections

Rescheduling action	Approach		
	D'Ariano/Pranzo	Corman et al.	Törnquist Krasemann
<i>I</i>	✓	✗	✗
<i>II</i>	✓	✗	✗
<i>III</i>	✓	✗	?
<i>IV</i>	✓	✓	✓
<b>Runtime</b>	14 s – 787 s	166 s – 705 s	30 s

**Table 1.** Summary of possible rescheduling actions and runtimes for the compared approaches

are sequences of one or several consecutive blocks. Unlike D'Ariano/Pranzo and Corman et al. blocks are not modeled explicitly but rather through the adherence to safety distance constraints between vehicles traveling in the same direction. A schedule is defined as sequences of consecutive events, which define points in time at which a specific vehicle is planned to occupy a certain section.

To solve the rescheduling problem the greedy algorithm uses depth-first search to build up a tree with events as nodes. With each new level of the tree a successor to the previous event is chosen. In addition each node holds an estimation of the disturbance consequences of the partial solution. After the first branch of the tree is completed a feasible schedule is obtained and the remaining computational time, up to a predefined time limit, is used to improve the solution, by backtracking to nodes which provide more promising disturbance estimations.

By modeling sections with up to  $n$  parallel tracks the resulting application may be able to at least partially perform rescheduling action III from section 2. However, given the information from [9] it is not clear if the developed method is able to handle a change of the destination station, which is the only viable rescheduling action in tram networks regarding changes in routes.

Finally, because the developed algorithm schedules the occupation of each section, it is able to perform rescheduling action IV.

For computational experiments Törnquist Krasemann uses data from the Norrköping traffic district in Sweden. The sub network is composed of 28 stations, 15 double-tracked sections and 17 single tracked sec-

tions. All tracks are bi-directional and all but one station have between two and 14 tracks. The author conducts experiments using 20 different scenarios, a maximum tolerated computational time of 30 seconds and a time horizon of 90 minutes, during which 46 to 51 vehicles are scheduled.

To evaluate the solution quality of the algorithm, the scenarios are also solved with CPLEX using a modified version of the formulation proposed in [8, 10]. While the greedy algorithm finds a first feasible solution in less than one second and very often finds other solutions within the first few seconds, in many of the examined scenarios the solution quality of the greedy algorithm cannot be evaluated because CPLEX does not find a solution within 24 hours. A reduction of the time horizon from 90 to 60 minutes (and subsequently reducing the number of scheduled trains) resolves this issue for most scenarios.

While in eight out of 20 scenarios the greedy algorithm finds a solution with an objective function value equal to the optimal solution obtained by CPLEX, in another eight scenarios the objective function values differ significantly. However, the difference between objective function values gives little information about the applicability of the time table, because the schedules may differ due to the way they are constructed. To evaluate the applicability of the schedules during daily operation they should be simulated.

The main appeal of the application lies in its short runtime even with a time horizon larger than 60 minutes, which could make it very promising for use in tram networks. This is especially true if the implementation is parallelized to explore several branches

of the tree simultaneously, as suggested by Törnquist Krasemann.

#### 4 Comparison of rescheduling strategies

Substantial evaluation of the rescheduling approaches of D'Ariano/Pranzo, Corman et al. and Törnquist Krasemann allows for the conclusion that they are all applicable to tram networks to a certain degree. (For a summary see table 1)

While the approach of D'Ariano and Pranzo can be used to perform all rescheduling actions from section 2, its runtime increases too fast for applicability in tram networks with dense schedules. Solving the problem in parallel for separated parts of the network or for exploration of different solution alternatives simultaneously might resolve this issue.

The approach of Corman et al. seems to be able to perform rescheduling action IV only, and it is the only one we have examined that considers more than one optimization goal. However, minimizing the number of missed connections is not as crucial in case of tram networks as in case of railway networks. Nonetheless their approach can be adapted to incorporate other optimization goals, e.g. satisfying given sets of transport planning requirements (see [12]). As with the approach of D'Ariano and Pranzo, the runtime may become problematic when applying the approach to tram networks. A parallel version of the Branch and Bound algorithm may remedy this problem.

Although it seems to be able to perform rescheduling action IV only, the approach by Törnquist Krasemann is very appealing because of its very short runtime of 30 seconds regardless of the underlying disturbance scenario. Furthermore, it can be easily parallelized by completing different branches of the tree simultaneously. However, the applicability of the obtained solutions remains open. Firstly, because for time horizons longer than 60 minutes only a few reference solutions were obtained using CPLEX. Secondly, because no dynamic system influences were considered, i.e. the schedules were not simulated or tested in the real system.

#### 5 Conclusions and future work

In this paper we described, evaluated and compared different rescheduling strategies for railway networks in regard to their applicability to tram networks. While our first theoretical examination of the de-

scribed strategies indicates that all of them seem to be applicable to tram networks to a certain degree, it also shows a couple of problems that have to be addressed: Based on the information available to us it is not clear if the approaches of Corman et al. and Törnquist Krasemann can be used to perform rescheduling actions I, II or III. In addition, the runtime for the approaches of D'Ariano/Pranzo and Corman et al. seems to be too long for applicability in tram networks with dense schedules, making it necessary to look into the possibility of parallelizing the implementations.

To assess the applicability of different rescheduling strategies more thoroughly, we want to implement a tool which allows us to manually apply schedule adjustments and gives instant visual assessment of the expected consequences. Based on the data obtained by using this tool we plan to implement an optimization module that incorporates promising rescheduling strategies and is tightly connected to our existing simulation module (see [7, 12]). We hope this will allow for developing strategies specialized on tram networks.

#### 6 Acknowledgements

This work was partially supported by *Rhein Energie Stiftung Jugend Beruf Wissenschaft* under grant number W-10-2-002. We thank the other members of our work group Patrick Kuckertz and Bert Randerath.

#### 7 References

- [1] Adenso-Díaz, B.; Oliva González, M.; González-Torre, P.: *On-line timetable re-scheduling in regional train services*. Transportation Research Part B: Methodological, Vol. 33, 1999, pp. 387-398.
- [2] Corman, F.; D'Ariano, A.; Pacciarelli, D.; Pranzo, M.: *Bi-objective conflict detection and resolution in railway traffic management*. Transportation Research Part C: Emerging Technologies, Vol. 20, 2012, pp. 79-94.
- [3] D'Ariano, A.; Pacciarelli, D.; Pranzo, M.: *A branch and bound algorithm for scheduling trains in a railway network*. European Journal of Operational Research, Vol. 183, 2007, 643-657.
- [4] D'Ariano, A.; Pranzo, M.: *An Advanced Real-Time Train Dispatching System for Minimizing the Propagation of Delays in a Dispatching*

*Area Under Severe Disturbances*. Networks and Spatial Economics, Vol. 9, 2009, pp. 63-84.

Research Part E: Logistics and Transportation Review, Vol. 48, 2012, pp. 578-590.

- [5] Darmanin, T.; Lim, C.; Gan, H.: *Public Railway Disruption Recovery Planning: A New Recovery Strategy for Metro Train Melbourne*. Proceedings of the 11<sup>th</sup> Asia Pacific Industrial Engineering and Management Systems Conference, 2010.
- [6] Ginkel, A.; Schöbel, A.: *To wait or not to wait? The bicriteria delay management problem in public transportation*. Transportation Science, Vol. 41, 2007, pp. 527-538.
- [7] Lückcrath, D.; Ullrich, O.; Speckenmeyer, E.: *Modeling time table based tram traffic*. Simulation Notes Europe (SNE), ARGESIM/ASIM Pub., TU Vienna, Vol. 22, No. 2, 2012, pp. 61-68.
- [8] Törnquist, J.: *Railway traffic disturbance management: an experimental analysis of disturbance complexity, managements objectives and limitations in planning horizon*. Transportation Research Part A: Policy and Practice, Vol. 41, 2007, pp. 249-266.
- [9] Törnquist Krasemann, J.: *Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances*. Transportation Research Part C: Emerging Technologies, Vol. 20, 2012, pp.62-78.
- [10] Törnquist, J.; Persson, J.: *N-tracked railway traffic re-scheduling during disturbances*. Transportation Research Part B: Methodological, Vol. 41, 2007, pp. 342-362
- [11] Ullrich, O.: *Modellbasierte Parallelisierung von Anwendungen zur Verkehrssimulation: Ein dynamischer und adaptiver Ansatz*. Dissertation, University of Cologne, Germany, to appear.
- [12] Ullrich, O.; Lückcrath, D.; Franz, S.; Speckenmeyer, E.: *Simulation and optimization of Cologne's tram schedule*. Simulation Notes Europe (SNE), ARGESIM/ASIM Pub., TU Vienna, Vol. 22, No. 2, 2012, pp. 69-76.
- [13] Zeng, A. Z.; Durach, C. F.; Fang, Y.: *Collaboration decisions on disruption recovery service in urban public tram systems*. Transportation





# Semantic Road Network Models for Rapid 3D Traffic Scenario Generation

Tobias Haubrich<sup>1</sup>, Sven Seele<sup>1</sup>, Rainer Herpers<sup>1,2,3</sup>, Martin E. Müller<sup>1</sup>, Peter Becker<sup>1</sup>

<sup>1</sup>Bonn-Rhein-Sieg University of Applied Sciences, Institute of Visual Computing

<sup>2</sup>University of New Brunswick, Canada

<sup>3</sup>York University, Canada

tobias.haubrich@inf.h-brs.de

## 1 Introduction

Traffic simulations are generally used to forecast traffic behavior or to simulate non-player characters in computer games and virtual environments. These systems are usually modeled in such a way that traffic rules are strictly followed. However, rule violations are a common part of real-life traffic and thus should be integrated into such models.

One objective of the AVeSi Project (“Agenten-basierte Verkehrssimulation mit psychologischen Persönlichkeitsprofilen” – Engl. “Agent-based traffic simulation with psychological personality profiles”) is the development of a traffic simulation in which simulated traffic participants (agents) may act against better knowledge and traffic rules and cause potentially dangerous situations [1]. The idea is to use this simulation within the FIVIS environment for road safety training of school children [2][3].

To provide a large variety of learning scenarios, the generation of new 3D scenarios needs to be as cost efficient and simple as possible. Ideally, data describing real city roads should be used to create more realistic conditions. Two components have to be generated from the data: a 3D model and a road description file. The 3D model, consisting of a 3D mesh and material properties, is needed by the simulation engine for visualization and physics simulation. The road description file is used to generate a semantic road network model. The model and its generation process are examined in this contribution.

In AVeSi two traffic simulation levels are executed in parallel to simulate traffic agents with different levels of detail (cf. [4]). The first simulation level (*queue-based mesosimulation*) applies a queue-based representation of the road network [5]. Agents are passively moved through the network – without influence of physics – by being transferred from one queue to a selected succeeding queue. The second simulation

level (*cognitive microsimulation*) consists of a detailed waypoint representation of the roads. In this model the agents are actively traveling along the waypoints using a car following model [6]. Currently the Intelligent Driver Model (IDM) [7] is applied.

Agents need to be constantly transferred between both simulation levels to ensure proper functionality of the system. To avoid mismatches during these conversions, it is necessary that the semantic models used by each simulation level are generated from the same data. Additionally, by generating the individual parts of both models from the same underlying description, a mapping between elements of both models is automatically achieved.

## 2 Semantic Road Network Models

To achieve a simple to use and cost efficient scenario generation, we introduce a structured workflow which is shown in Figure 1. The basis of this workflow are two semantic road network models. Each model needs to include all data required for the simu-

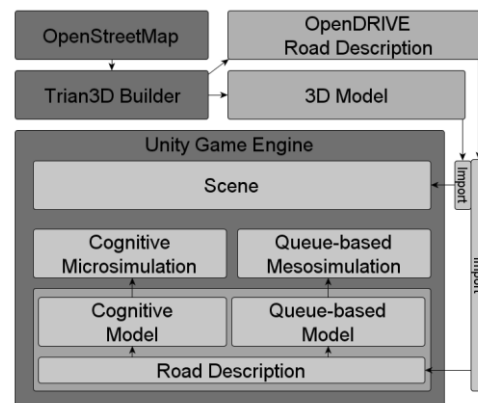


Figure 1. Illustration of a workflow realizing rapid scenario generation. OpenStreetMap data is used by the Trian3D Builder ([www.triangraphics.com](http://www.triangraphics.com)) to generate a road description file. This file is imported into the simulation engine (Unity) and used to generate the road network models for the simulation.

lation of traffic participants, such as speed limits, priorities, road geometry, and more. However, before introducing the models, the underlying standard (OpenDRIVE<sup>®</sup>) is briefly described.

### 2.1 OpenDRIVE<sup>®</sup>

As road network description, data in OpenDRIVE<sup>®</sup> file format is used. OpenDRIVE<sup>®</sup> is an open standard for the description of track-based road networks. The standard provides the option to describe the geometry of roads as well as features along the roads which can influence the driving process [8]. A road description consists of a number of information such as planar geometry, elevation, lane sections, lane features and objects surrounding the road. The general course of a road is described by a single reference line. Junctions are described by a list of links which connect incoming roads to paths (1:n). Each path leads to exactly one outgoing road (1:1). Paths are described as roads leading through a junction.

### 2.2 Model for Queue-based Mesosimulations

The queue-based model [5], which is motivated by the FastLane model [9], is based on a graph representation consisting of a number of vertices and edges. Each road is represented by two edges, each representing lanes of the same direction. Each vertex represents a junction and is generated from the corresponding junction element in the description. Information about the priority of incoming roads can be derived from different sources within the standard, e.g. from priorities stored within the junction element. Since the mesosimulation does not require information about the exact course of a road or its lanes, only two edges are generated for every road in the standard. Each one represents one driving direction. Information that is required, like the length of a section, the number of lanes, the average speed limit, and the start- and end-vertices, are provided by the OpenDRIVE<sup>®</sup> standard. After the generation of the elements, the adjacency information between edges and vertices is added from the successor and predecessor information described in the standard.

### 2.3 Model for Cognitive Microsimulations

The model required for the microsimulation [4] needs to be more detailed than the mesoscopic one, since the cognitive agents need more information to make plausible decisions. The current model is based on segments which are either lanes or connectors. Each segment consists of a sequence of waypoints, which

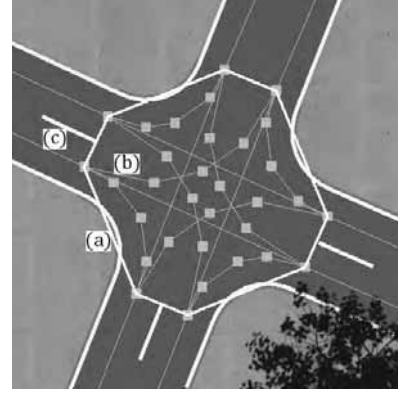


Figure 2. Visualization of a sample of the road network model at a crossroads area. The frame (a) represents the junction element. Inside the frame is the junction area which contains the assigned connector elements (b). The gray lines outside of the frame are the lane elements (c).

is used by the agents for navigation. Every lane is connected to  $n$  preceeding connectors and to  $m$  succeeding connectors (1:n/1:m). Each connector is linked to one predecessor lane and one successor lane (1:1). Neighboring lanes are subsumed in a road element and neighboring connectors are subsumed in a path element. When agents navigate through the network they look at a number of following waypoints to retrieve information. For this purpose, the OpenDRIVE<sup>®</sup> standard provides useful information like: speed limits, road marks, lane types and lane neighborhood.

In the following this road network model is described formally. At first the main elements of the road network model and their relations are defined. Figure 2 shows an example of a crossroads area including all of the main elements: lanes, connectors, waypoints and a junction element.

$L$  is a set of lanes. Each lane describes a road segment between two junctions.  $C$  is a set of connectors. A connector starts where a lane ends by running into a junction, then leads through this junction and connects to another lane leading away from the junction. Thus a connector describes one possible way through a junction.

$$L = \{l_1, l_2, \dots, l_n\}, n \in \mathbb{N},$$

$$C = \{c_1, c_2, \dots, c_m\}, m \in \mathbb{N}$$

$R$  is a set of roads. A road is an element subsuming a set of lanes. Therefore, the function *road* is defined to assign a road to a lane and the function *lanes* to assign a unique subset of all lanes to a road.

$$R = \{r_1, r_2, \dots, r_o\}, o \in \mathbb{N},$$

$$road : L \rightarrow R,$$

$$lanes : R \rightarrow \wp(L),$$

$$lanes(r) = \{l \in L \mid road(l) = r\},$$

$$l \in lanes(r_i) \wedge l \in lanes(r_j) \Rightarrow r_i = r_j, i, j \in \mathbb{N}$$

$P$  is a set of paths. A path is an element subsuming a set of connectors. Therefore, the function *path* is defined to assign a path to a connector and the function *connectors* to assign a unique subset of all connectors to a path.

$$P = \{p_1, p_2, \dots, p_q\}, q \in \mathbb{N},$$

$$path : C \rightarrow P,$$

$$connectors : P \rightarrow \wp(C),$$

$$connectors(p) = \{c \in C \mid path(c) = p\},$$

$$c \in connectors(p_i) \wedge c \in connectors(p_j) \Rightarrow p_i = p_j, i, j \in \mathbb{N}$$

To describe the predecessor and successor information of lanes and connectors, the following functions are defined. Each connector is assigned to exactly one preceding and exactly one succeeding lane ( $l:I$ ). Each lane is assigned to a set of preceding connectors and a set of succeeding connectors ( $l:n/l:m$ ).

The function *prelane* is defined to assign a preceding lane to a connector and the function *sucplane* to assign a succeeding lane to a connector.

$$prelane : C \rightarrow L,$$

$$sucplane : C \rightarrow L$$

The function *preconns* is defined to assign a set of preceding connectors to a lane and the function *succonns* to assign a set of succeeding connectors to a lane.

$$preconns : L \rightarrow \wp(C),$$

$$preconns(l) = \{c \in C \mid suc(c) = l\},$$

$$succonns : L \rightarrow \wp(C),$$

$$succonns(l) = \{c \in C \mid pre(c) = l\}$$

The basic network model is built from the above mentioned components and their relations. For the missing geometric description of the road network,

waypoint sequences are defined and assigned to each segment (connector or lane). Additionally, each waypoint is assigned to a set of information which can be used by the agents during the simulation process.

$W$  is a set of waypoints and  $S$  is a set of possible sequences of these waypoints. The function *waypoints* is defined to return a sequence of waypoints assigned to a lane or a connector.

$$W = \{w_1, w_2, \dots, w_s\}, s \in \mathbb{N},$$

$$S = \{(w_u, \dots, w_v) \mid w_u, \dots, w_v \in W, u, v \in \mathbb{N}\},$$

$$waypoints : C \cup L \rightarrow S$$

Furthermore, *pos* is a function which is defined to assign two coordinates  $(x, y)$  to a given waypoint to specify its spatial position.

$$pos : W \rightarrow \mathbb{R}^2,$$

$$pos(w) = (x, y)$$

$I$  is a set of information. The function *wpinf* is defined to assign a subset of  $I$  to a given waypoint.

$$I = \{i_1, i_2, \dots, i_t\}, t \in \mathbb{N},$$

$$wpinf : W \rightarrow \wp(I)$$

In addition, the right of way priorities for the segments need to be described. Thus, a set of junction elements is introduced. Each junction element is assigned to a set of information enabling the representation of the priorities as well as additional information. Because connectors represent possible ways through the defined junctions each connector is explicitly assigned to one of these junction elements. Now agents, which are traveling on a lane, are able to look at the connectors ahead and their assigned junction to retrieve the priority information.

$J$  is a set of junctions. The function *junction* is defined to assign a junction to a connector and the function *jconns* to assign a unique subset of all connectors to a junction.

$$J = \{j_1, j_2, \dots, j_u\}, u \in \mathbb{N},$$

$$junction : C \rightarrow J,$$

$$jconns : J \rightarrow \wp(C),$$

$$jconns(j) = \{c \in C \mid junction(c) = j\},$$

$$c \in jconns(j_k) \wedge c \in jconns(j_l) \Rightarrow j_k = j_l, k, l \in \mathbb{N}$$

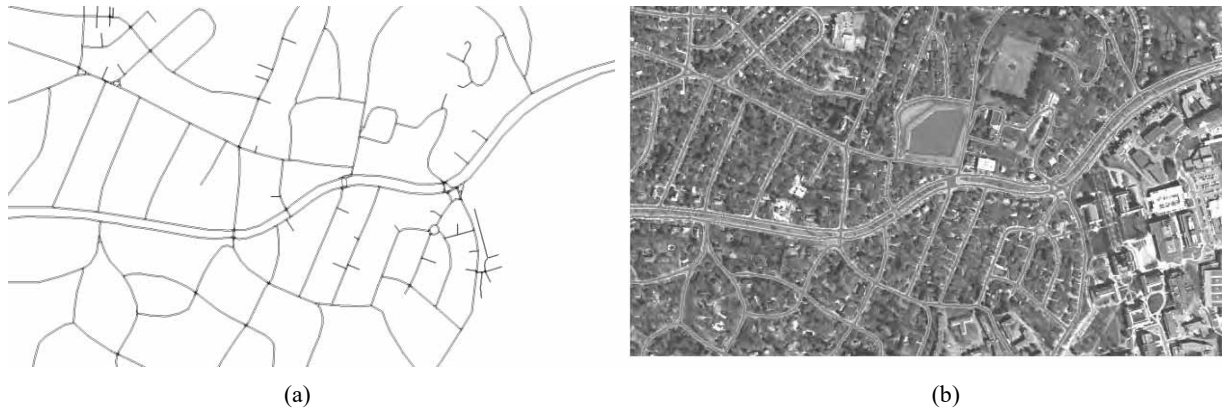


Figure 3. (a) Visualization of a road network generated from OpenDRIVE® data using the semantic road network model for cognitive microsimulations. Data was generated using the Trian3D Builder software. (b) Associated 3D model also generated by the Trian3D Builder.

*juncinf* is a function which is defined to assign a subset of  $I$  to a given junction.

$$juncinf: J \rightarrow \wp(I)$$

With the definition of the above mentioned elements and their relations, a graph-like representation of the road network model and its characteristics is provided. By assigning waypoint sequences to each lane and connector, the network is spatially represented. Information can be added to every waypoint and thus to arbitrary positions within the network. Additionally, information can be added to the junction elements which provides the possibility to add, for example, right of way priorities.

The generation process of the cognitive microsimulation model is described in the following. It is divided into four steps. In the first step the specific road network elements (roads, lanes, paths, connectors and junctions) are generated. Afterwards the links between the lane and connector elements are set. By setting the two explicit connector links it is guaranteed that every connection (lane to connector and vice versa) is considered, because of the one to one relation. Additionally, every connector is assigned to its related junction. After the links for every segment (lane/connector) are set, the exact course of every lane and every path is known. Waypoints are then generated to provide agents with a basic network for navigation. In the last step, additional information is added to the network model. Information can be added to the junction elements or existing waypoints. If information is needed at positions where no element exists, waypoints with special information are generated at these significant positions. This allows the integration of various road traffic related aspects.

In Figure 3 a visualization of a road network model generated by this approach and the associated 3D model are shown. Waypoints are generated at suitable distances. To calculate their position, the point on the reference line of the road must be calculated for the wanted position. An offset to the reference line is added depending on the actual lane widths.

The distance between waypoints depends on the road's curvature; strongly curved sections need more waypoints and less curved sections need less waypoints. To calculate characteristic waypoint positions the Douglas-Peucker-Algorithm is applied [10]. This algorithm minimizes the number of necessary waypoints while preserving the shape of the sequence within a given tolerance.

### 3 Conclusions

One of the main objectives of developing an automated generation of traffic scenarios is to achieve efficiency in terms of time. For example, consider the sample scenario shown in Figure 3, which contains 380 lanes and 495 connectors with a total of ca. 3600 waypoints. Manually creating one lane and connecting it to its successor and predecessor lanes via a set of connectors usually took between 5 and 10 minutes for a sample scenario. Thus, assuming an average time of 7.5 minutes for creating one lane and setting its connections, manually creating the sample scenario shown in Figure 3 would take approximately 2850 minutes (47.5h) without considering any mistakes or revisions. An automated generation is able to provide the same results in only a few minutes. Additionally, there are several other advantages in comparison to a manual creation. For instance, the real road network layout is reproduced much more precisely by auto-

matically placing the elements than by manually positioning them. An additional advantage of an automated generation is that information assigned to elements is not influenced by human error. This could be noise from unprecise positioning or worse, faulty information. Generating real and precise data enhances the realism, which increases the benefit of the constructed scenario. One additional advantage for our specific application is that elements from the mesoscopic and microscopic models are automatically mapped avoiding problems during the conversion between both simulation levels.

## 4 Acknowledgments

The FIVIS project was funded by the BMBF-FH<sup>3</sup> program “Angewandte Forschung an Hochschulen im Verbund mit der Wirtschaft”, funding for the FIVIS 2 project was provided by the DGUV (FP 307), and the AVeSi project is being funded by the FHprofUnt program of the BMBF (17028X11).

## 5 References

- [1] M. Kutz, R. Herpers. *Urban Traffic Simulation for Games*. In: Proceedings of the 2008 Conference on Future Play: Research, Play, Share, ACM, 2008.
- [2] R. Herpers, et al. *Multimedia Sensory Cue Processing in the FIVIS Simulation Environment*. In: Multiple Sensorial Media Advances and Applications: New Developments in MulSeMedia, IGI Global, 2011.
- [3] R. Herpers, et al. *FIVIS – A Bicycle Simulation System*. World Congress on Medical Physics and Biomedical Engineering (WC 2009). IFMBE Proceedings Vol. 25/4, Springer, 2009.
- [4] S. Seele, T. Dettmar, R. Herpers, C. Bauckhage and P. Becker. *Cognitive Aspects of Traffic Simulations in Virtual Environments*. In: Simulation Notes Europe 22 (2), Special Issue „Traffic Systems – Technical Systems“, AR-GESIM / ASIM, Vienna, 2012.
- [5] T. Dettmar, S. Seele, R. Herpers, P. Becker and C. Bauckhage. *Efficient Mesoscopic Simulations for Persistent Agents in 3D-Applications and Games*. Submitted to the 8<sup>th</sup> Int. Conf. on the Foundations of Digital Games, FDG 2013.
- [6] M. Treiber, M. and A. Kesting. *Verkehrsdynamik und –simulation, 1st ed.* Springer-Verlag, Berlin, Heidelberg, 2010.
- [7] M. Treiber, A. Hennecke and D. Helbing. *Congested traffic states in empirical observations and microscopic simulations*. *Phys. Rev. E*, 62. S. 1805-1824, 2000.
- [8] M. Dupuis, et al. *OpenDRIVE® Format Specification, Rev 1.3*. VIRES GmbH, 2010.
- [9] C. Gawron. *Simulation-based traffic assignment*. PhD thesis, University of Cologne, Germany, 1998.
- [10] D. Douglas and T. Peucker. *Algorithms for the reduction of the number of points required to represent a digitized line or its caricature*. In: Cartographica: Int. J. for Geographic Information and Geovisualization 10.2, S. 112-122, 1973.



# Ontologische Modellierung und Modellgenerierung in der MATLAB/Simulink Umgebung: Die „Tiny SES Toolbox“

Tobias Schwatinski, Thorsten Pawletta

Hochschule Wismar, University of Applied Sciences

*tobias.schwatinski@hs-wismar.de*

Der „System Entity Structure and Model Base“ (SES/MB) Formalismus ist eine Ontologie, die vornehmlich im Bereich des Data-Engineering und der Simulationstechnik verwendet wird. Es wird anhand der „Tiny SES Toolbox“ praktisch gezeigt, wie der SES/MB Formalismus für die Meta-Modellierung von Systemen und für eine automatische Modellsynthese verwendet werden kann. Dazu werden verschiedene Systeme in einer SES spezifiziert und daraus, mittels der Tiny SES Toolbox, ausführbare Matlab/Simulink Modelle generiert.

## 1 Problemstellung

Komplexe Systemstrukturen lassen sich oft durch Graphen darstellen. Die Systemkomponenten beziehungsweise ganze Subsysteme werden als Knoten und deren Beziehungen untereinander durch verbindende benannte Kanten abgebildet. Ein Spezialfall der allgemeinen Strukturierung stellt die Hierarchisierung dar [1]. Im Allgemeinen sind in einem hierarchischen System alle Elemente nach einer bestimmten Rangordnung angeordnet, wobei die Elemente gleicher Rangordnung eine gemeinsame Schicht innerhalb der Hierarchie bilden. Auf der einen Seite können dadurch chaotische Strukturen verhindert werden, auf der anderen Seite werden aber alle möglichen Systemstrukturen auf definierte gerichtete Strukturen beschränkt [1]. Weiterhin werden alle Elemente innerhalb einer Hierarchie, welche nicht weiter zerlegt werden können, als Systemelemente bezeichnet.

Neben der Hierarchisierung spielt die Modularisierung eine bedeutende Rolle innerhalb sämtlicher Ingenieurwissenschaften. Dies gilt im Besonderen für die Softwaretechnik als auch für die Modellbildung und Simulation (M&S). In [1] wird ein Modul als eine weitgehend abgeschlossene Bau- oder Funktionsgruppe definiert. Derartige Module können, innerhalb der M&S, einzelnen Systemelementen einer Hierarchie zugeordnet werden. Somit ist, systemtheoretisch betrachtet, ein Modul die Abstraktion des Ein-/Ausgangsverhaltens eines Systemelements durch eine „Black Box“. Das dynamische Verhalten der „Black Box“ ist entweder mittels Zustandsvariablen, Ein-/Ausgangsfunktionen

beziehungsweise Zustandsüberföhrungsfunktionen oder durch ein Submodell zu konkretisieren.

Die beiden genannten Prinzipien werden, in Form von Signalflussgraphen (CBD: Causal Block Diagrams), in vielen grafischen Modellierungswerkzeugen verwendet, wie beispielsweise in Scilab/Xcos, Matlab/Simulink oder AnyLogic. Ein CBD ist ein Graph, der aus miteinander verbundenen Operationsblöcken gebildet wird. Dabei stellt jede Verbindung eine Signalleitung dar, während jeder Operationsblock eine entsprechende Funktionseinheit definiert. Die Operationsblöcke sind in Bibliotheken katalogisiert und können sowohl zeitbasierte Funktionalitäten wie Integrieren oder Differenzieren, als auch einfache algebraische Funktionalitäten wie Summieren oder Multiplizieren umsetzen.

Der „System Entity Structure and Model Base“ (SES/MB) Formalismus [2] ist eine Ontologie, deren Entwicklung in den 80'er Jahren durch Zeigler begonnen [3] und heute neben der Simulationstechnik insbesondere im Bereich des Data-Engineering verwendet wird [4]. Dabei stellen, ähnlich den CBD, die Hierarchisierung und die Modularisierung zentrale Prinzipien (Axiome) des SES/MB Formalismus dar. Die SES ist in erster Näherung ein Baum mit Knoten, Kanten und Attributen. Darüber hinaus werden auch Variablen, Kopplungen, Zwangsbedingungen und Auswahlregeln definiert. Die Entwicklung der SES ist eng mit der Entwicklung des Discrete Event System Specification (DEVS) Formalismus gekoppelt [3, 5]. Ähnlich den CBD können bei DEVS einzelne Operationsblöcke (atomic DEVS, coupled DEVS) modelliert und durch Signalleitungen (DEVS couplings) miteinander

verbunden werden. Darüber hinaus werden in [6, 7] unterschiedliche DEVS Modelle, in Form eines Metamodells, mit einer SES spezifiziert.

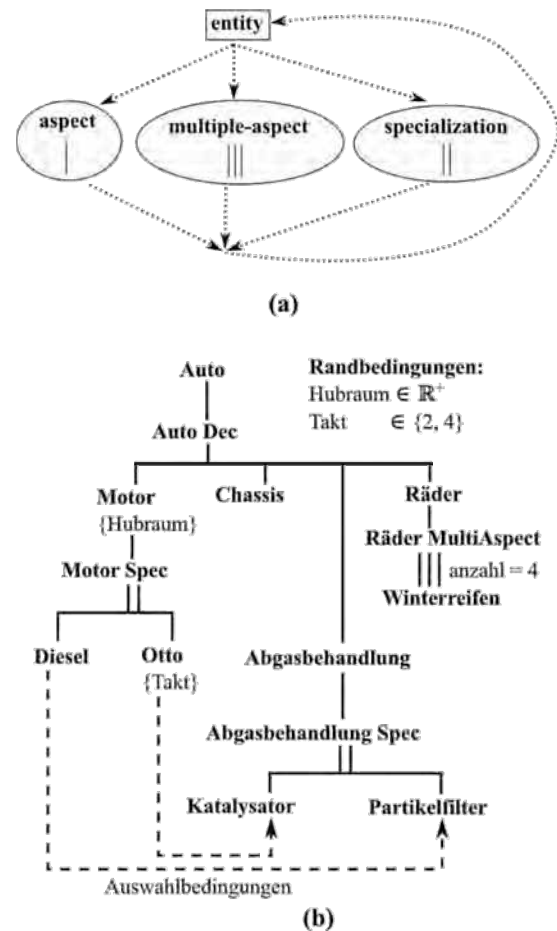
In neuerer Zeit wird durch verschiedene Arbeiten versucht, den formalen SES/MB Formalismus auch im direkten Zusammenhang mit CBD und allgemeinen Blockdiagrammen zu nutzen. In [8] werden beispielsweise fertigungstechnische Prozesskettenvarianten in einer SES spezifiziert und daraus automatisch Matlab/SimEvents Modelle generiert. Weiterhin wird in [9] untersucht, ob anstelle der formalen graphischen Beschreibungsmittel der SES auch die Verwendung adäquater Blockdiagramme möglich ist. Im Rahmen dieses Beitrags wird die praktische Anwendung des SES/MB Formalismus im Zusammenhang mit Matlab/Simulink Modellen thematisiert.

Dazu werden im nächsten Abschnitt die unterschiedlichen Arten von Beschreibungsmitteln einer SES aufgeführt und deren Verwendung für eine ontologische Systemmodellierung diskutiert. Danach wird am Beispiel des Standardregelkreises gezeigt, wie der SES/MB Formalismus für die Meta-Modellierung von dynamischen Systemen und für eine automatische Modellsynthese verwendet werden kann. Im vierten Abschnitt wird exemplarisch die SES von einem einschleifigen Regelkreis mit verschiedenen LTI-Systemen spezifiziert und daraus, mittels der „Tiny SES Toolbox“ [10], ausführbare Matlab/Simulink Modelle generiert.

## 2 Ontologische Systemmodellierung mit der SES

Die System Entity Structure (SES) ist eine Ontologie. Sie bildet einen Baum, dessen Knoten in vier unterschiedlichen Ausprägungen auftreten können [2]. Diese Ausprägungen heißen Entity-, Aspect-, Multiple-Aspect- und Specialization-Knoten. Die prinzipielle Abfolge der Knoten innerhalb einer SES ist in Abbildung 1 (a) dargestellt. Entity-Knoten repräsentieren die Elemente einer realen oder imaginären Welt. Aspect-Knoten dienen der Zerlegung eines Entity-Knotens in feiner aufgelöste Strukturen. Multiple-Aspect-Knoten definieren die Vielfachheit des nachfolgenden Entity-Knotens und Specialization-Knoten repräsentieren Kategorien oder Familien bestimmter Ausprägungen eines Entity-Knotens. Darüber hinaus können bestimmten Knoten oder Kanten Attribute zugewiesen werden.

Abbildung 1 (b) zeigt beispielhaft die SES zur Spezifikation verschiedener Automodelle. Jede Entity *Auto* besteht entsprechend dem Aspect-Knoten *Auto Dec* aus den Entities *Motor*, *Chassis*, *Abgasbehandlung* und *Räder*.



**Abbildung 1:** (a) Allgemeine Struktur einer SES [2],  
(b) Beispiel einer SES

Die Entity *Motor* wird dabei entweder spezialisiert zu der Entity *Diesel* oder der Entity *Otto*. Darüber hinaus wird der Entity *Otto* ein Attribut *Takt* zugewiesen, welches entsprechend den Randbedingungen den Wert 2 oder 4 annehmen kann. Die SES unterstützt auch das Konzept der Vererbung. So wird beispielsweise das Attribut *Hubraum* der Entity *Motor* an die Entities *Diesel* und *Otto* vererbt. Weiterhin sind die jeweiligen Spezialisierungen der Entities *Motor* und *Abgasbehandlung* über Auswahlbedingungen miteinander verknüpft. So führt beispielsweise die Auswahl des Dieselmotors auch zur Auswahl des Partikelfilters. Auf die Entity *Räder* folgt ein Multiple-Aspect-Knoten mit dem Parameter *anzahl = 4*. Deshalb wird dieser Knoten durch vier



Entities des Typs *Winterreifen* aufgelöst. Die SES in Abbildung 1 (b) beschreibt demnach zwei strukturell unterschiedliche Automodelle:

- Dieselmotor, Chassis, Partikelfilter, 4 Winterreifen
- Ottomotor, Chassis, Katalysator, 4 Winterreifen

Außerdem können die Attribute Takt und Hubraum, für beide Automodelle, mit beliebigen Werten belegt werden

Auf diese Weise lassen sich mit einer SES eine Vielzahl von verschiedenen Ausprägungen eines beliebigen Systems anschaulich darstellen. Ursprünglich entwickelt wurde die SES zur deklarativen Beschreibung von Modellstrukturen in der Simulationstechnik. Wenn die SES mit einer Modellbasis (MB) kombiniert wird, lassen sich aus ihr mit Hilfe eines Modellgenerators vollständige Simulationsmodelle ableiten. Diese erweiterte Form der SES, SES/MB genannt, weist jedem Blattknoten der Baumstruktur genau ein Softwaremodul der Modellbasis zu. Dazu ist es notwendig, dass an den Aspect-Knoten die entsprechenden Kopplungsbeziehungen der Entities spezifiziert werden. Darüber hinaus unterstützt die SES auch Auswahlregeln für Aspect- und Spezialisierung-Knoten.

### 3 Modellierung auf Basis des SES/MB Frameworks und das Konzept der Modellsynthese

In diesem Abschnitt wird am Beispiel des Standardregelkreises praktisch gezeigt, wie unterschiedliche Modellstrukturen mittels einer SES sukzessive spezifiziert werden können. Darüber hinaus werden, aus der in diesem Abschnitt entwickelten SES, unterschiedliche Modellstrukturen synthetisiert.

Der Standardregelkreis in Abbildung 2 (a) besteht aus den drei Operationsblöcken: *Führungsgröße*, *Vergleichsstelle* und *offene Kette*. Der Operationsblock *offene Kette* besitzt einen Eingang und zwei Ausgänge und entspricht somit einem SIMO-System. Die Eingangsgröße entspricht dabei der Regeldifferenz  $e$ . Die beiden Ausgangsgrößen entsprechen zum Einen der gemessenen Regelgröße  $y_m$  und zum Anderen dem Vektor  $(y, x)^T$ , welcher sich aus den Streckenzuständen  $x$  und der Regelgröße  $y$  zusammensetzt. In diesem Zusammenhang stellt

lediglich  $y_m$  die Regelgröße für die Rückführung dar, während  $(y, x)^T$  ausschließlich informativen Charakter hat. Zum Zwecke der graphischen Bewertung von Simulationsergebnissen, wie z.B. der Auswertung von Führungsgröße  $w$ , Regelgröße  $y$  und Streckenzuständen  $x$ , sind außerdem ein *Mux*- und ein *Scope*-Operationsblock in Abbildung 2 (a) vorhanden.

Abbildung 2 (b) zeigt die erste Schicht der SES des Standardregelkreises. Dabei sind die relevanten Operationsblöcke als Entity-Knoten und die Kopplungen zwischen den Operationsblöcken als Kantenattribute des Aspect-Knotens *Standardregelkreis Dec* spezifiziert. Das Kantenattribut "Führungsgröße.w, Vergleichsstelle.+" definiert beispielsweise, dass der Ausgang  $w$  des Blocks *Führungsgröße* mit dem Eingang  $+$  des Blocks *Vergleichsstelle* verbunden ist.

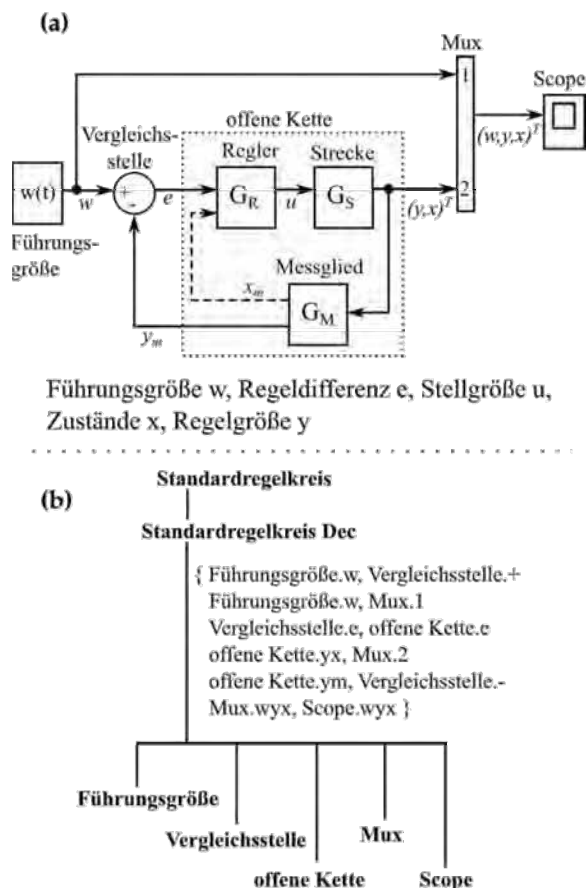


Abbildung 2: Standardregelkreis mit SES

Im Folgenden werden die Entity-Knoten *Führungsgröße* und *offene Kette* der SES aus Abbildung 2 (b) sukzessive erweitert. Zunächst wird in Abbildung 3 (a) die Entity *Führungsgröße* um einen Teilbaum erweitert. Dieser Teilbaum definiert

eine Taxonomie und ordnet verschiedene Signalgeneratorblöcke, in Form von spezialisierten Entities, der allgemeinen Entity *Führungsgröße* zu. Außerdem sind die Entities *Sinus* und *Sprung* durch Attribute parametrisiert. Weiterhin können Auswahlregeln, für die Auswahl einer konkreten Spezialisierung, als Kantenattribut des Specialization-Knotens *Führungsgröße Spec* definiert werden.

Abbildung 3 (b) zeigt, in Form eines weiteren Teilbaums, die Taxonomie der Entity *offene Kette*. Diese unterscheidet zwischen einem Zustandsraummodell *ZR-Modell* und einer allgemeinen Systembeschreibung *allgemeines System*. Die Entity *allgemeines System* setzt sich dabei, entsprechend dem Aspect-Knoten *allgemeines System Dec*, aus den Entities *Regler*, *Strecke* und *Messglied* zusammen. In diesem Zusammenhang können die Komponenten des allgemeinen Systems sowohl linear als auch nichtlinear beschrieben werden. Abbildung 2 (a) zeigt dazu exemplarisch die Komposition der offenen Kette aus einzelnen Übertragungsfunktionen. Weiterhin sind Auswahlregeln und Kopplungsbeziehungen als Kantenattribute angegeben.

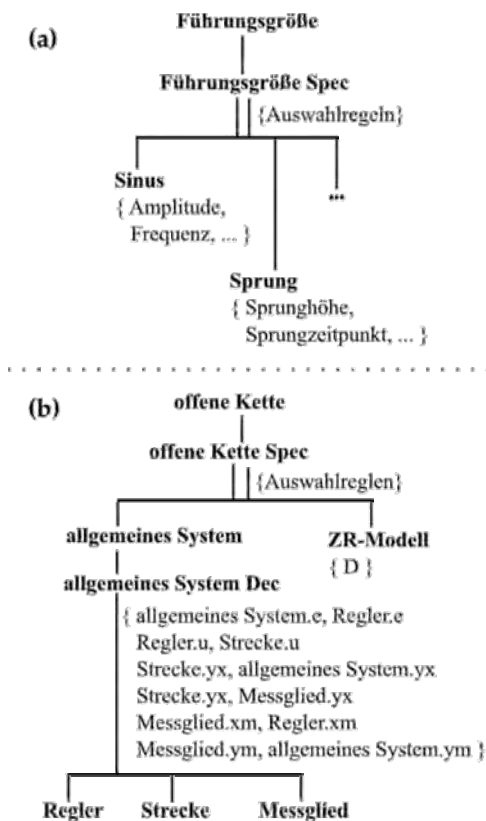


Abbildung 3: SES des Standardregelkreises, 2. Teil

In Abbildung 4 werden sämtliche Blattknoten der SES in Abbildung 3 (b) um Teilbäume erweitert. Abbildung 4 (a) zeigt die Taxonomie der Entity *ZR-Modell*. Diese umfasst verschiedene *SIMO*-Systeme, welche durch die Attribute Systemmatrix *A*, Eingangsmatrix *B*, Ausgangsmatrix *C* und der Durchgangsmatrix *D* hinreichend spezifiziert werden. Im vorliegenden Fall wird die Durchgangsmatrix *D* von der Entity *ZR-Modell* vererbt und ist somit für alle *SIMO*-Entities identisch.

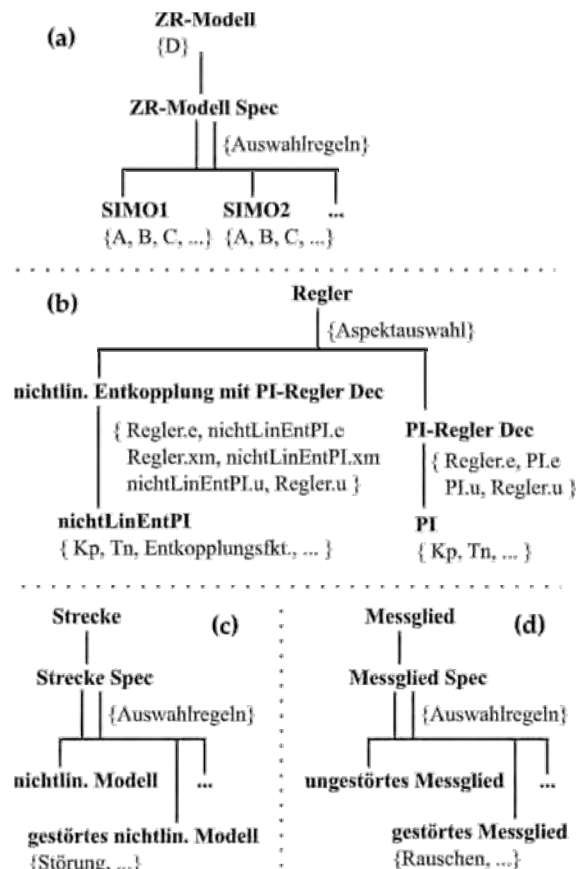


Abbildung 4: SES des Standardregelkreises, 3. Teil

Abbildung 4 (c) und (d) stellen Taxonomien der Entities *Strecke* und *Messglied* dar. In beiden Fällen umfassen sowohl die Regelstrecke als auch das Messglied verschiedene nichtlineare mathematische Modelle zzgl. deren Attribute. Die konkrete Struktur dieser Modelle ist nicht näher spezifiziert.

Abbildung 4 (b) spezifiziert für die konkrete Reglerstruktur der Entity *Regler* zwei unterschiedliche Aspekte zzgl. zugehöriger Auswahlregeln. Zum Einen kann der Regler in Form eines gewöhnlichen *PI*-Reglers, mit den beiden Attributen Proportionalbeiwert *Kp* und Nachstellzeit

$T_n$ , ausgeführt werden. Zum Anderen können die nichtlinearen Streckenmodelle *Strecke* durch die nichtlineare Entkopplung linearisiert und somit einfach geregelt werden, wenn zusätzlich die Messung der Streckenzustände  $x$  möglich ist. In diesem Zusammenhang soll die Schätzung von Streckenzuständen durch einen Beobachter nicht berücksichtigt werden. Beim vorliegenden Anwendungsbeispiel wird innerhalb der offenen Kette, entgegen dem üblichen Aufbau, der durch das Messglied bestimmte Streckenzustand  $x_m$  direkt an den Regler übermittelt. Einen derartigen nichtlinearen Regler stellt die Entity *nichtLinEntPI* dar, welche eine Kombination aus nichtlinearer Entkopplung und PI-Regler repräsentiert und neben den Attributen  $K_p$  und  $T_n$  auch eine *Entkopplungsfunktion* definiert.

Die einfache SES des Standardregelkreises, aufgespalten in die Abbildungen 2 bis 4, spezifiziert 20 unterschiedliche Modellstrukturen. Darüber hinaus werden 18 Attribute für die Parametrierung der einzelnen Modellstrukturen definiert. Durch die Beschneidung der SES, *Prunen* genannt, kann aus der Menge an möglichen Modellstrukturen einer SES genau eine Modellstruktur synthetisiert und parametrisiert werden. Dazu muss sämtlichen Attributen ein konkreter Wert zugewiesen und alle Entscheidungsknoten, wie Specialization-Knoten, Aspect-Knoten und Multiple-Aspect-Knoten aufgelöst werden. Während des Prunens werden unter anderem Entities umbenannt, Kopplungsbeziehungen aktualisiert und Teilbäume bzw. Parameter vererbt. Weitere Details dazu sind in [2, 4] zu finden. Das Ergebnis des Prunens ist die Pruned Entity Structure (PES), welche genau eine Systemstruktur der ursprünglichen SES darstellt. Die Abbildungen 5 und 6 zeigen beispielhaft die PES zweier möglicher Modellstrukturen, welche aus der zuvor betrachteten SES des Standardregelkreises abgeleitet wurden.

#### 4 „Tiny SES Toolbox“ und deren Verwendung in Matlab/Simulink

In diesem Abschnitt wird die praktische Verwendung der „Tiny SES Toolbox“ von der Spezifikation von Metamodellen bis hin zur automatischen Modellgenerierung von Matlab/Simulink Modellen thematisiert. Die theoretische Grundlage dafür bildet der SES/MB Formalismus, welcher jedem Blattknoten einer SES genau ein Softwaremodul aus einer Modellbasis (MB) zuordnet und darüber hinaus die Ableitung ausführbarer simulatorspezifischer Simulationsmodelle aus einer PES ermöglicht.

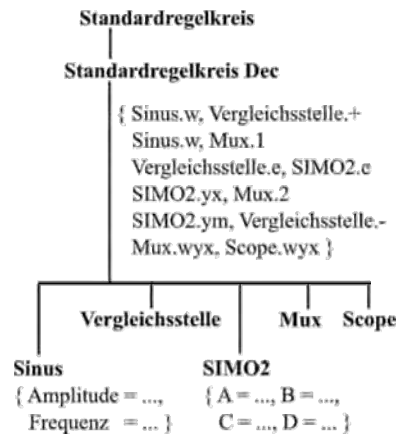


Abbildung 5: PES Beispiel 1

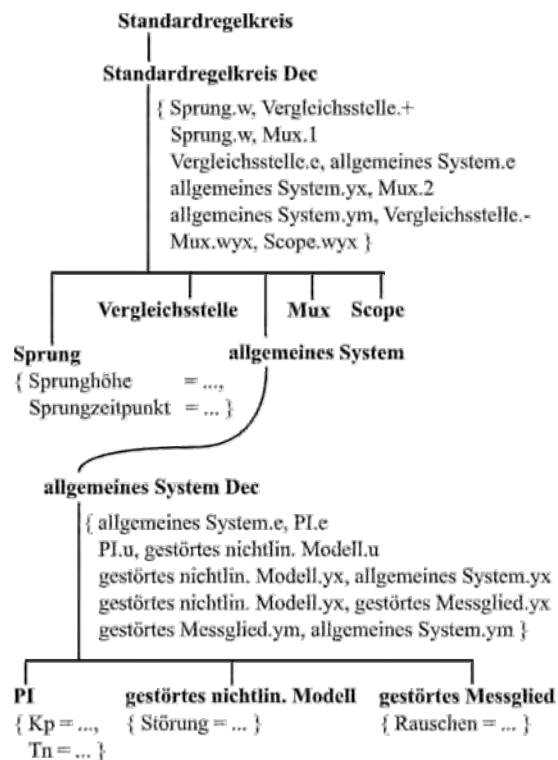


Abbildung 6: PES Beispiel 2

Das Konzept der Modellgenerierung, auf Grundlage des SES/MB Formalismus, zeigt Abbildung 7. Als Anschauungsbeispiel dient im Folgenden der einschleifige Regelkreis in Abbildung 8. In diesem Beispiel können die allgemeinen Operationsblöcke *Führungsgröße* und *offene Kette* durch andere konkrete Operationsblöcke ausgetauscht werden. Im vorliegenden Fall soll anstelle des Operationsblocks *Führungsgröße* entweder ein Sinusgenerator oder ein Sprunggenerator verwendet werden. Weiterhin soll der Operationsblock *offene Kette* durch zwei

unterschiedlich parametrisierte LTI-SISO-Systeme modelliert werden. Die Modellbasis des einschleifigen Regelkreises zeigt Abbildung 9. Die zugehörige SES zeigt Abbildung 10.

Die SES in Abbildung 10 spezifiziert 4 unterschiedliche Modellstrukturen und definiert 16 Attribute. Weiterhin sind an den beiden Specialization-Knoten in Abbildung 10 (b) und (c), in Form von Kantenattributen, verschiedene Auswahlregeln definiert. Darüber hinaus definiert die SES in Abbildung 10 (d) die drei SES-Variablen: *auswahlFührung*, *auswahlLTI* und *parameterFührung*. Diese Variablen sind ausschließlich für das Prunen von Bedeutung und werden z.B. für die logische Auswertung von Auswahlregeln oder für die Wertzuweisung an

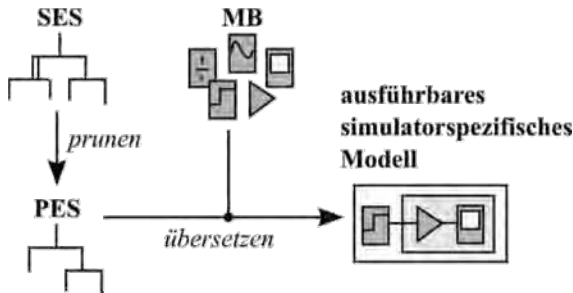


Abbildung 7: Modellgenerierung nach SES/MB Formalismus

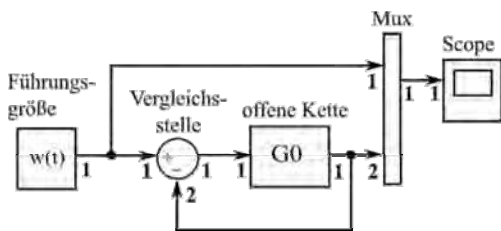


Abbildung 8: Einschleifiger Regelkreis

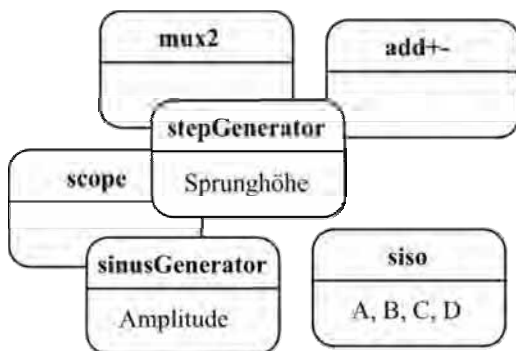


Abbildung 9: Modellbasis des einschleifigen Regelkreises

parametrierbare Attribute verwendet. Die semantischen Bedingungen in Abbildung 10 (d) definieren zusätzlich die zulässige Wertemenge für jede einzelne Variable. In Abbildung 10 (b) bestimmt die Variable *auswahlFührung* die konkrete Ausführung der Entity *Führungsgröße*. Außerdem dient die Variable *parameterFührung* zur

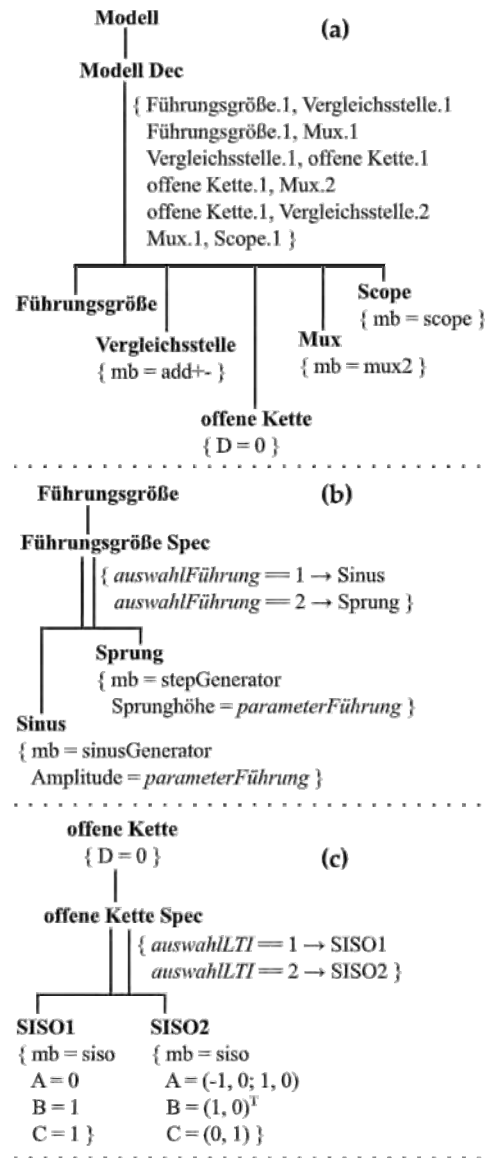


Abbildung 10: SES des einschleifigen Regelkreises

```

1  /*Definition von Knoten*/
2  node(modell, [ ]).
3  node(modellDec, [ ]).
4  /*Definition von Kanten*/
5  edge(modell, modellDec).
6  edge(modellDec, fuehrungsgroesse).
7  /*Definition von Kopplungen*/
8  decomp(modellDec, modell,
    [[fuehrungsgroesse,1,vergleichsstelle,1],
    [fuehrungsgroesse,1,mux,1],
    [vergleichsstelle,1,offeneKette,1],
    [offeneKette,1,mux,2],
    [offeneKette,1,vergleichsstelle,2],
    [mux,1,scope,1]
    ]).
9  /*Definition von Attributen*/
10 parameter(mux, [value(mb, mux2)]).
11 parameter(scope, [value(mb, scope)]).
12 /*Definition von Auswahlregeln*/
13 specrule(offeneKetteSpec, offeneKette,
    [[sisol, [value(auswahlLTI, 1)]],
    [siso2, [value(auswahlLTI, 2)]]
    ]).
14 /*Definition von semantischen
    Bedingungen*/
15 semanticconds(
    (value(parameterFuehrung,X), X>0 )
    ).
    
```

Abbildung 11: Auszug aus der prädikatenlogisch formulierten SES

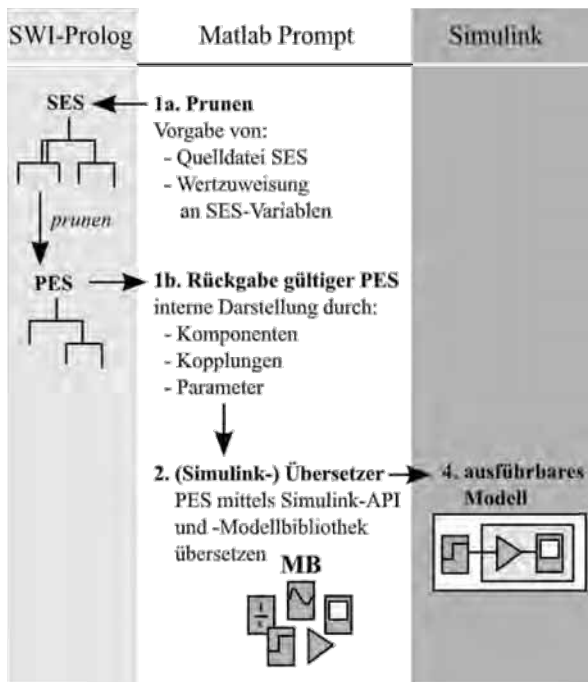


Abbildung 12: Verwendung der „Tiny SES Toolbox“

Parametrierung der beiden Attribute *Amplitude* und *Sprunghöhe*. Weiterhin bestimmt die Variable *auswahlLTI* die konkrete Ausführung der Entity *offene Kette* in Abbildung 10 (c). Das Attribut *mb* weist jedem Blattknoten der SES genau ein Softwaremodul aus der Modellbasis in Abbildung 9 zu.

Die vorliegenden SES muss für die Verwendung mit der „Tiny SES Toolbox“ in einer separaten Datei codiert werden. Dazu sind sämtliche Elemente, wie z.B. Knoten, Kanten und Attribute, prädikatenlogisch zu formulieren. Einige Beispiele zeigt Abbildung 11. Die genaue Syntax ist in [10] dokumentiert.

Darüber hinaus implementiert die „Tiny SES Toolbox“ einen Prune-Algorithmus in SWI-Prolog und einen (Simulink-)Übersetzer in Matlab. Die prinzipielle Verwendung der „Tiny SES Toolbox“ ist in Abbildung 12 dargestellt.

Im ersten Schritt wird die Prune-Funktion in Matlab gerufen und ihr die Quelldatei einer SES zzgl. deren SES-Variablen übergeben. Die Rückgabewerte der Prune-Funktion sind Matlab-Cell-Arrays, welche die Komponenten, Kopplungen, Attribute und Gültigkeit der PES beinhalten. Falls die PES gültig ist, wird im zweiten Schritt der (Simulink-)Übersetzer gerufen. Dieser kann die PES, mittels der Simulink-API und der Simulink-Modellbibliothek, in ein ausführbares Simulink-Modell übersetzen, welches anschließend ausgeführt werden kann. Abbildung 13 zeigt dazu beispielhaft ein Matlab-Skript. Das mittels des Skripts erzeugte Simulink-Modell ist in Abbildung 14 dargestellt.

## 5 Zusammenfassung und Ausblick

Die SES kann für die Spezifikation von unterschiedlichen CBD verwendet werden. Darüber hinaus können im Rahmen des SES/MB Formalismus (i) einzelne Modellstrukturen aus einer SES synthetisiert und (ii) diese unter Verwendung einer Modellbasis in ausführbare simulatorspezifische Modelle übersetzt werden. In diesem Beitrag wurde am Beispiel des Standardregelkreises gezeigt, dass dies auch für Matlab/Simulink-Modelle möglich ist.

Die in diesem Beitrag vorgestellte „Tiny SES Toolbox“ implementiert beide Aspekte des SES/MB Formalismus. Eine Besonderheit der Toolbox ist, dass die Programmierung und Verarbeitung der SES mittels SWI-Prolog realisiert wird. Dazu stellt die „Tiny SES Toolbox“ eine geeignete Matlab-SWI-Prolog Schnittstelle bereit. Daneben beinhaltet die

Toolbox den Prototyp eines Übersetzers, der einzelne Modellstrukturen einer SES in ausführbare Simulink-Modelle übersetzt. Somit kann die „Tiny SES Toolbox“ durchgängig von der Meta-Modellierung von Systemen bis hin zur automatisierten Modellgenerierung, innerhalb von Matlab/Simulink, verwendet werden.

Zukünftig wird die „Tiny SES Toolbox“ an verschiedenen Stellen erweitert werden. Ein wichtiges Ziel ist beispielsweise die Übersetzung von SES-Modellstrukturen in Matlab/SimEvents-Modelle.

```

1 %SES-Quelldatei und SES-Variablen
  definieren
2 inputSES = 'standardRegelkreisSES.pl';
3 v_in = { {'auswahlFuehrung', '1'},...
          {'auswahlLTI', '1'},...
          {'parameterFuehrung','0.5'} };
4 %SES prunen
5 [Components,Couplings,Parameter,Valid]= ...
  Matlab_Interface_SES(inputSES, v_in);
6 %(Simulink-)Übersetzer
7 if strcmp(Valid,'valid')
8   %Simulinkmodell initialisieren
9   nameSystem='pesTemp';
10  load_system('Simulink');
11  h=new_system(nameSystem);
12  open_system(h);
13  set_param(h, 'Solver', 'ode45',...
             'StopTime', '10');
14  %PES nach Simulink übersetzen
15  modelBuilder(nameSystem, Components,...
               Couplings, Parameter);
16  %Modell simulieren
17  sim(nameSystem);
18 else
19  disp(['PES not valid!']);
20 End

```

Abbildung 13: Modellgenerierung und Modellausführung in Matlab

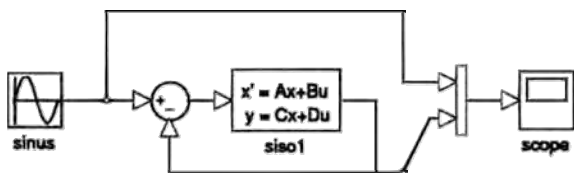


Abbildung 14: Einschleifiger Regelkreis in Simulink

## 6 Quellenangaben

- [1] Balzert: *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*, 3. Auflage, Spektrum Akademischer Verlag Heidelberg, 2009
- [2] Zeigler, Hammonds: *Modeling and Simulation-Based Data Engineering*, Elsevier Academic Press, 2007
- [3] Zeigler: *Multifaceted Modelling and Discrete Event Simulation*. Academic Press, 1984
- [4] Kim, Won, Sim, Kim: *System Integration Method Based on System Entity Structure*, United States Patent Application Publication, US 2008/0092207 A1
- [5] Zeigler, Sarjoughian: *Guide to Modeling and Simulation of Systems of Systems*. Springer Pub., 2012
- [6] Zeigler, Praehofer, Kim: *Theory of Modeling and Simulation*, Second Edition, Elsevier Academic Press, 2000
- [7] Hagendorf, Pawletta: *A Framework for Simulation Based Structure and Parameter Optimization of Discrete Event Systems*. In: *Discrete-Event Modeling and Simulation: Theory and Applications*, CRC Press Inc. of Taylor & Francis Group, USA, 2011
- [8] Larek, Brinksmeier, Pawletta, Hagendorf: *Model-Based Planning of Resource Efficient Process Chains Using System Entity Structures*. In: *Future Trends in Production Engineering - Proc. of the 1st Conf. of the German Academic Soc. for Prod. Eng. (WGP)*, Springer Pub., 2013
- [9] Shin, Jung, Lee, Chi, Han: *SES-based Structure Modeling Method Using the Block Diagram*, In: *Proc. of The 2012 Int. Conf. on Foundations of Comp. Science (FCS'12)*, Las Vegas/Nevada, USA, 2012
- [10] Schwatinski, Schmidt, Pawletta: *Tiny SES Toolbox*, Website 2012, [http://www.mb.hs-wismar.de/cea/SES\\_Tbx/sesToolboxMain.html](http://www.mb.hs-wismar.de/cea/SES_Tbx/sesToolboxMain.html)

# Ontologische Modellierung und ereignisdiskrete Simulation mit MATLAB/SimEvents am Beispiel der ressourceneffizienten Prozesskettenplanung.

Schmidt, Artur; Pawletta, Thorsten  
Hochschule Wismar, Forschungsgruppe CEA  
*artur.schmidt@hs-wismar.de*

Modellbasierte Planungsverfahren haben sich von der strategischen Planung bis zum Betrieb von Produktionsprozessen als Werkzeuge etabliert. Diese beinhalten bis heute kaum Ansätze zur Bewertung der Ressourceneffizienz von Fertigungsverfahren in Prozessketten. Anhand einer neuen Modellbibliothek zur gezielten Untersuchung der Ressourceneffizienz spanender Fertigungsverfahren werden Prinzipien der fertigungstechnischen Prozesskettenmodellierung/-simulation demonstriert und bestehende Probleme, wie die zu berücksichtigende Variantenvielfalt, identifiziert. Zur Lösung des Variantenproblems wird ein ontologischer Modellierungsansatz zur formalen Spezifikation von Prozesskettenstrukturen mit differenzierten Parametrierungen diskutiert. Anschließend wird gezeigt, wie aus dem ontologischen Modell in Kombination mit einer Modellbibliothek automatisiert konkrete Prozesskettenmodelle generiert und simuliert werden können.

## 1 Problemstellung

Steigende Rohstoff- und Energiepreise stellen produzierende Unternehmen vor neue Herausforderungen. Vor diesem Hintergrund werden zunehmend Methoden der Modellbildung und Simulation eingesetzt, um den Ressourcenverbrauch von fertigungstechnischen Prozessketten quantitativ zu untersuchen. Oft steht die Untersuchung von logistischen, produktionssteuerungstechnischen und gebäudetechnischen Aspekten [1,2] beziehungsweise die Untersuchung einzelner sehr energieintensiver Fertigungsverfahren [3] im Vordergrund.

Gemäß der Forderung in [4] auch die Ressourceneffizienz der Fertigungsverfahren in Prozessketten eingehend zu analysieren, wurden in [5,6,7] erste Modellbibliotheken für spezielle Fertigungsverfahren entwickelt. Diese basieren auf einem zeitdynamischen ereignisorientierten Modellierungsansatz und ermöglichen somit nicht nur die Berechnung von Ressourcenverbräuchen, sondern auch die zeitliche Bewertung von Verbrauchsspitzen. Die einzelnen Fertigungsverfahren sind als konfigurierbare dynamische Modelle mit kompatiblen Ein-/Ausgangsschnittstellen in einer Simulationsumgebung implementiert und in Modellbibliotheken organisiert. In der Simulationsumgebung können die atomaren Modelle parametrisiert, zu Prozessketten komponiert und ausgeführt werden.

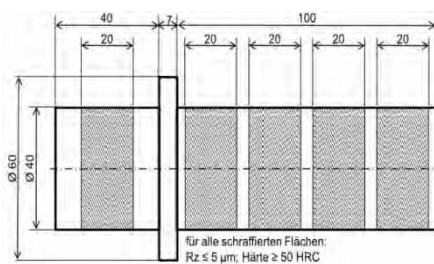
In [6,8,9] wird gezeigt, dass bei realen produktionstechnischen Problemstellungen aufgrund der Varianten- und Parametervielzahl, trotz der Verwendung von Modellbibliotheken, die manuelle Komposition und Parametrierung von Prozesskettenstrukturen schnell sehr aufwendig werden kann. Ein Lösungsansatz wird in Metamodellierungsansätzen in Kombination mit automatischen Modellgenerierungskonzepten und simulationsbasierten Optimierungstechniken gesehen. Unter einem Metamodell wird in diesem Zusammenhang eine formale Spezifikation von einer Menge unterschiedlicher Modellstrukturen und Modellparametrierungen verstanden.

Am Beispiel der Modellbibliothek in [6] werden die Prinzipien der fertigungstechnischen Prozesskettenmodellierung/-simulation aufgezeigt, bestehende Probleme identifiziert und einige Lösungen diskutiert. Anschließend erfolgt eine Konzentration auf das Problem der Varianten- und Parametervielzahl. Dazu wird das Konzept der Metamodellierung von Modellstrukturen und Modellparametrierungen auf Basis der System Entity Structure (SES) Ontologie nach [10] eingeführt und auf die Anwendung der Prozesskettenmodellierung vertieft. Danach wird der Ansatz zur automatisierten Modellgenerierung auf Basis einer SES und einer Modellbibliothek nach [11] erläutert. Weiterhin wird dessen Anwendung zur Generierung ausführbarer Prozesskettenmodelle in der MATLAB/SimEvents Umgebung gezeigt. Im Vergleich zu den Arbeiten in [6,8] werden neue SES Konstrukte

eingeführt, die die Modellierung der Parametervielfalt vereinfachen. Darüberhinaus basiert die Modellgenerierung auf einer allgemeinen Toolbox [12] und ist nicht mehr spezifisch auf die Komponenten der Modellbibliothek in [6] ausgelegt. Abschließend werden exemplarische Simulationsergebnisse demonstriert und es erfolgt ein Ausblick auf fortführende Arbeiten.

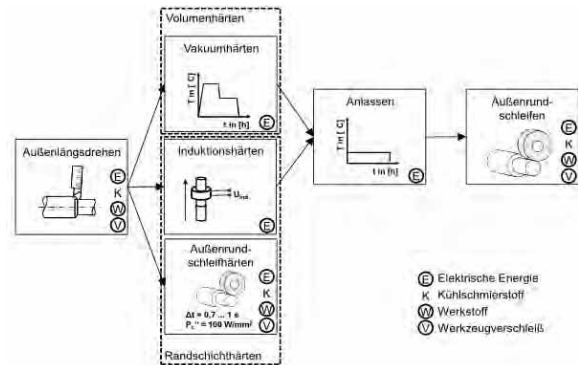
## 2 Klassische Prozesskettenmodellierung/-simulation, Probleme und Lösungsansätze

Die klassische Vorgehensweise der Modellbildung und Simulation unter Verwendung moderner Simulationsumgebungen, wie z.B. MATLAB/SimEvents, besteht aus drei wesentlichen Schritten: (i) Komponentenmodellierung, (ii) Modellgenerierung und (iii) Modellausführung. Im Rahmen der Komponentenmodellierung werden dynamische Modelle, zu betrachtender Systemkomponenten erstellt und in einer Modellbibliothek organisiert. Aus diesen werden komplexe modular-hierarchische Modellstrukturen manuell komponiert und parametriert. Diese Phase wird nachfolgend allgemein als Modellgenerierung bezeichnet. Die anschließende Modellausführung erfolgt manuell oder teilautomatisiert, z.B. skriptgesteuert, in der Simulationsumgebung. Die klassische Vorgehensweise soll anhand der in [6] vorgestellten Modellbibliothek gezeigt werden. Dazu werden exemplarisch drei alternative Prozessketten zum Herstellen von Werkstücken gemäß Abbildung 1 betrachtet.



**Abbildung 1.** Exemplarischer Werkstücktyp mit Randbedingungen nach [6].

In Abbildung 2 ist schematisch die Struktur der drei Prozessketten mit den zu betrachtenden Ressourcenarten dargestellt.

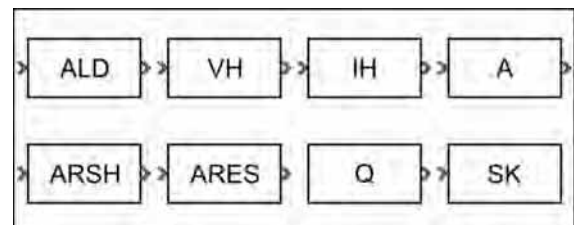


**Abbildung 2.** Prozesskettenstrukturen mit den zu betrachtenden Ressourcenarten je Fertigungsverfahren nach [6].

Die gewählten Prozessketten bestehen aus folgenden Fertigungsverfahren:

- Außenlängsdrehen (ALD), Vakuumhärten (VH), Anlassen (A), Außenrundscheißen (ARES)
- Außenlängsdrehen, Induktionshärten (IH), Anlassen, Außenrundscheißen
- Außenlängsdrehen, Außenrundscheißenhärten (ARSH)

Jedes Fertigungsverfahren ist als ein ereignisdiskretes Modell, Basismodell genannt, in MATLAB/SimEvents modelliert. Die Basismodelle sind, wie in Abbildung 3 dargestellt, in einer Modellbibliothek organisiert.



**Abbildung 3.** Schamtische Darstellung der Modellbibliothek nach [6].

Die Basismodelle können in drei Grundtypen klassifiziert werden:

- CNC-Satzbasierte Modelle
- Messdatenbasierte Modelle
- Hilfsmodelle

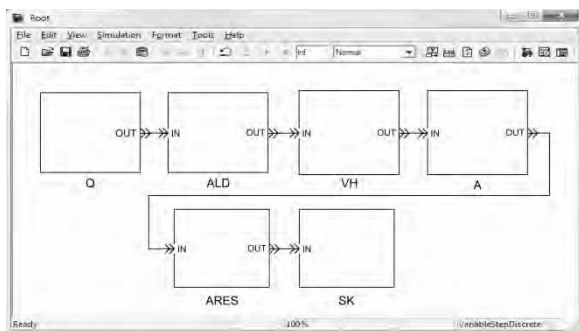
Bei den CNC-Satzbasierten Modellen werden die Betriebszustände, Zustandsübergänge und die Ressourcenverbräuche direkt auf Basis der CNC-Programme der Fertigungseinrichtungen berechnet. Den Modellen können Einstellgrößen, wie Schnitttiefe, Schnittgeschwindigkeit, Vorschub und radialer Vorschub übergeben werden. Zu den CNC-Satzbasierten Modellen zählen ALD, ARES und A-



RESH. Die Basismodelle VH, IH und A zählen zur Klasse der messdatenbasierten Modelle. Ihr dynamisches Verhalten und die Ressourcenverbräuche wurden unmittelbar aus Messdaten der realen Fertigungseinrichtungen identifiziert. Die Parametrierung der messdatenbasierten Modelle ist relativ begrenzt. Zu den Hilfsmodellen gehören Quelle (Q) und Senke (SK). Q dient der Generierung von Werkstücken, welche als Entitäten abgebildet werden. SK dient der Terminierung von Entitäten. Die werkstoffspezifischen Eigenschaften, d.h. der Werkstofftyp und die aktuelle Werkstückgeometrie, sind direkt mit jeder Werkstück-Entität verknüpft.

Jedes Basismodell ist als eine Black-Box zu sehen, in welche eine Werkstück-Entität ereignisorientiert eintritt und nach erfolgter Bearbeitung wieder austritt. Der Bearbeitungsvorgang verbraucht Ressourcen und verändert die Werkstückgeometrie. In Verbindung mit einer Quelle und einer Senke stellt jedes fertigungstechnische Basismodell bereits ein ausführbares Simulationsmodell dar. Somit kann jedes Verfahren separat bezüglich seiner spezifischen Ressourcenverbräuche untersucht werden. In Abbildung 5 sind exemplarische Simulationsergebnisse für zwei Basismodelle dargestellt. Um den jeweiligen Gesamtverbrauch einer Ressourcenart zu erhalten, sind die gezeigten Verläufe numerisch zu integrieren.

Die Basismodelle können manuell zu Prozessketten gemäß Abbildung 2 komponiert und parametrieren werden (Modellgenerierung). Abbildung 4 zeigt beispielhaft die Modellstruktur der ersten Prozesskette in der MATLAB/SimEvents Umgebung. Die Modellausführung erfolgt manuell über das Simulink GUI oder teilautomatisiert (skriptgesteuert) über die Simulink API.

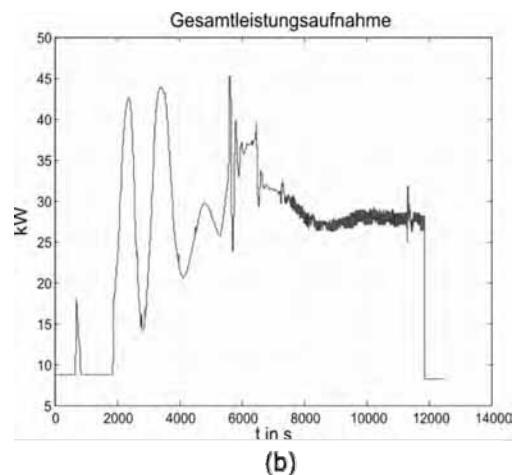
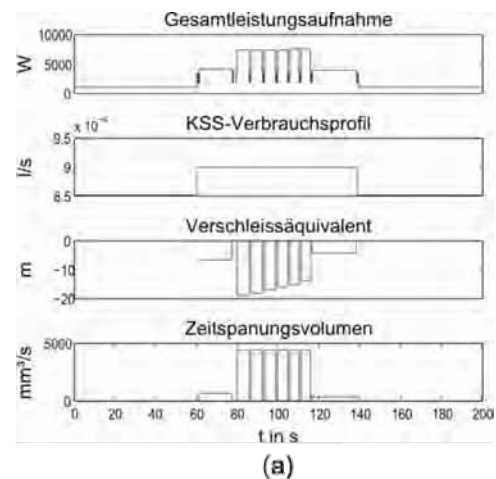


**Abbildung 4.** Ereignisdiskretes Simulationsmodell der ersten Prozesskette in MATLAB/SimEvents.

Die Untersuchungen in [6,8] zeigen, dass sich bei realen Problemstellungen schnell eine hohe Varian-

tenvielfalt an Prozessketten ergibt. Diese folgt aus alternativen Prozesskettenstrukturen, unterschiedlichen Werkstofftypen und umfangreichen Wertebereichen der Fertigungsparameter, so dass sich schnell Hunderte bis Tausende von Prozessvarianten ergeben (vgl. Abschnitt 3).

Experimente mit der Modellbibliothek zeigten, dass die Prozessketten schon bei einfachen Werkstücktypen Rechenzeiten von bis zu 200 Sekunden aufweisen [13]. Weiterhin unterstützt die Modellbibliothek in [6] keine mehrfache Instanziierung von Basismodellen eines Typs in einem Prozesskettenmodell und sie gestattet keine gleichzeitige Untersuchung verschiedener Werkstücktypen. Damit muss ein zu untersuchender Produktionsauftrag (Los) sortenrein sein.



**Abbildung 5.** Exemplarische Simulationsergebnisse (a) des ALD und (b) des VH Basismodells.

Zur Lösung der drei aufgezeigten Probleme erfolgten in [13] eine interne Neustrukturierung und Reimplementierung der Basismodelle. Die in [6,7] angestrebte einheitliche Grundstruktur der Basismodelle bezüg-

lich der Ereignisgenerierung wurde gemäß der fertigungstechnischen Klassifizierung durch eine Unterscheidung in CNC-Satzbasierte und messdatenbasierte Modelle, plus eine neue Modellart für differentialgleichungsbasierte Basismodelle, ersetzt. Die Schnittstelle zur Integration der fertigungstechnischen Aspekte wurde beibehalten, so dass die fertigungstechnische Modellbildung nicht modifiziert wurde. Das Ergebnis ist eine deutliche Reduzierung der Ereignismenge bei den messdatenbasierten Basismodellen, wodurch sich deren Rechenzeit um bis zu 90% verringert. Dadurch reduziert sich die Rechenzeit für die Prozessketten in Abbildung 2 insgesamt um jeweils circa 50% [13]. Darüberhinaus wurde eine strenge Datenkapselung umgesetzt und die Parametrierung der Basismodelle umgestaltet. Dadurch können Basismodelle des gleichen Typs beliebig mehrfach in einer Prozesskette verwendet werden. Weiterhin kann ein Produktionsauftrag aus verschiedenen Werkstücktypen bestehen.

Zur Beherrschung der Variantenvielfalt bei produktionstechnischen Prozessen werden in [6,8,9,14,15] Metamodellierungskonzepte zur formalen Spezifikation von Prozessvarianten (Strukturen + Parametrierungen) vorgeschlagen. Im nachfolgenden Abschnitt wird ein erweiterter Metamodellierungsansatz allgemein eingeführt und anschließend konkret auf die Spezifikation von Prozesskettenvarianten vertieft.

### 3 Ontologische Prozesskettenmodellierung mit der SES

Die System Entity Structure (SES) ist eine Ontologie, welche ursprünglich in [16] zur Repräsentation von Systemstrukturen in der Simulationstechnik eingeführt wurde. In [10] ist die SES Ontologie auf Data-Engineering Probleme erweitert worden. Sie definiert alle typischen Elemente einer anwendungsbezogenen Ontologie: (i) Konzeptionalisierung, (ii) Taxonomie, (iii) Vererbung und (iv) Instanziierung. Im Folgenden werden wesentliche Aspekte der SES kurz eingeführt, wobei eine Konzentration auf die anschließend zur Prozesskettenmodellierung benutzten Konstrukte erfolgt. Bezüglich umfassenderer Darstellungen sei auf die originäre Quelle [10] beziehungsweise auf [12,17] verwiesen.

#### 3.1 Einführung in die SES

Die SES wird durch einen gerichteten und beschrifteten Baum dargestellt. Die Knoten des Baumes gliedern sich in zwei Kategorien: (i) echte Knoten und

(ii) deskriptive Knoten. Echte Knoten repräsentieren Objekte der realen oder imaginären Welt, welche allgemein als Entitäten bezeichnet werden. Beispiele hierfür sind eine Fertigungseinrichtung, ein Fertigungsverfahren, ein Tier oder Wahrnehmungen wie Schmerz oder Freude. Deskriptive Knoten beschreiben in welcher Beziehung ein Vorgängerknoten mit einem Nachfolgeknoten steht. Bei den deskriptiven Knoten wird zwischen Aspekt-Knoten, Spezialisierungs-Knoten und Multipler-Aspekt-Knoten unterschieden. Ein Aspekt-Knoten beschreibt die Zerlegung einer Entität in Teilentitäten und wird durch die Endung Dec gekennzeichnet. Seine Kante wird durch einen einfachen senkrechten Strich dargestellt. Der Spezialisierungs-Knoten stellt eine Taxonomie einer Entität dar. Der Knoten besitzt die Endung Spec und einen doppelten senkrechten Strich als Kante. Der deskriptive Knoten Multipler-Aspekt ist ein Spezialfall des Aspekt-Knotens. Dieser Knoten beschreibt die Zerlegung einer Entität in typengleiche Teilentitäten und wird oft durch die Endung MultiAspect gekennzeichnet. Die Kante des Multiplen-Aspektes wird durch drei senkrechte Striche dargestellt. Bei der Modellierung mit einer SES muss die grundlegende Abfolge der Knoten beachtet werden. Es gilt, wie in Abbildung 6 gezeigt, einen alternierenden Modus von Entitäten-Knoten und deskriptiven Knoten einzuhalten. Dabei ist zu beachten, dass der Wurzelknoten und die Blattknoten immer Entitäten-Knoten darstellen.

An Entitäten-Knoten dürfen Attribute in Form von Konstanten und Variablen mit Wertebereichen definiert werden. Bei einer Taxonomiebeziehung, d.h. bei einem nachfolgenden Spezialisierungs-Knoten, werden diese an die nachfolgenden Entitäten-Knoten vererbt. An den Kanten der deskriptiven Knoten können verschiedene Kantenattribute definiert werden. Die Aspekt- und Multi-Aspekt-Kanten können Kopplungsbeziehungen beschreiben. Kopplungen spezifizieren Verknüpfungen zwischen den direkt nachfolgenden Entitäten. Die Multiple-Aspekt-Kante definiert zusätzlich den Wertebereich für die zulässige Anzahl der nachfolgenden typengleichen Teilentitäten. Eine Spezialisierungs-Kante kann Auswahlregeln definieren. Die Semantik von Auswahlregeln wird nachfolgend am Beispiel der Prozesskettenmodellierung gezeigt. Desweiteren wird eine spezielle Form von Zwangsbedingungen erläutert und es wird das Konzept der SES Variablen und Funktionen eingeführt.

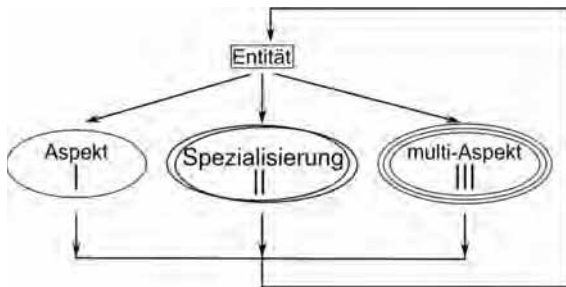


Abbildung 6. Grundlegende Knotenabfolge einer SES nach [10].

### 3.2 Exemplarische Prozesskettenmodellierung mit der SES

Für die Prozesskettenmodellierung soll von den in Abbildung 2 dargestellten Prozesskettenstrukturen ausgegangen werden. Weiterhin wird angenommen, dass das zu fertigende Werkstück aus zwei verschiedenen Materialtypen hergestellt werden kann. Werkstoffspezifisch sollen jeweils fünf verschiedene Einstellungen für die Fertigungsparameter der Verfahren ALD und ARES zulässig sein. Dabei handelt es sich um die ALD Parameter Schnitttiefe  $a_p$ , Schnittgeschwindigkeit  $v_{c,ALD}$  und Vorschub  $f$  sowie die ARES Parameter Schnittgeschwindigkeit  $v_{c,ARES}$  und die radialen Vorschubgeschwindigkeiten  $v_{fr}$ . Aus den drei Prozesskettenstrukturen und den zulässigen Parametereinstellungen folgen  $2 \cdot (5^3 \cdot 5^2) + 5^3 = 6.375$  Prozesskettenvarianten pro Werkstofftyp. Abbildung 7 zeigt die Spezifikation der Prozesskettenvarianten mit einer SES. Aus Platz- und Übersichtsgründen sind die Knotenattribute der Entitäten sowie die im unteren Teil spezifizierten SES Variablen und Funktionen nur teilweise dargestellt.

Der Wurzelknoten Root mit dem nachfolgenden Aspekt-Knoten RootDec beschreibt die Zerlegung jeder Prozesskette in die Basismodelle Q, ALD und SK sowie in eine spezielle Prozessvariante PV. Die Kopplungsbeziehungen der einzelnen Entitäten in einer Ebene sind an den Aspekt-Kanten definiert. Der Term (Q.out, ALD.in) an der RootDec Kante bedeutet, dass der Ausgangsport von Q mit dem Eingangsport von ALD verbunden ist. Alle Blattknoten der SES, außer PV1-PV3, entsprechen einem Basismodell aus der Modellbibliothek. Die Blattknoten PV1-PV3 repräsentieren spezielle Prozessvarianten, deren Struktur und Parametrierung in den Abbildungen 7b und 7c dargestellt ist. Die Blattknoten PV1-PV3 in Abbildung 7a können mit den namensgleichen Wurzelknoten der SES in Abbildung 7b und 7c substituiert werden.

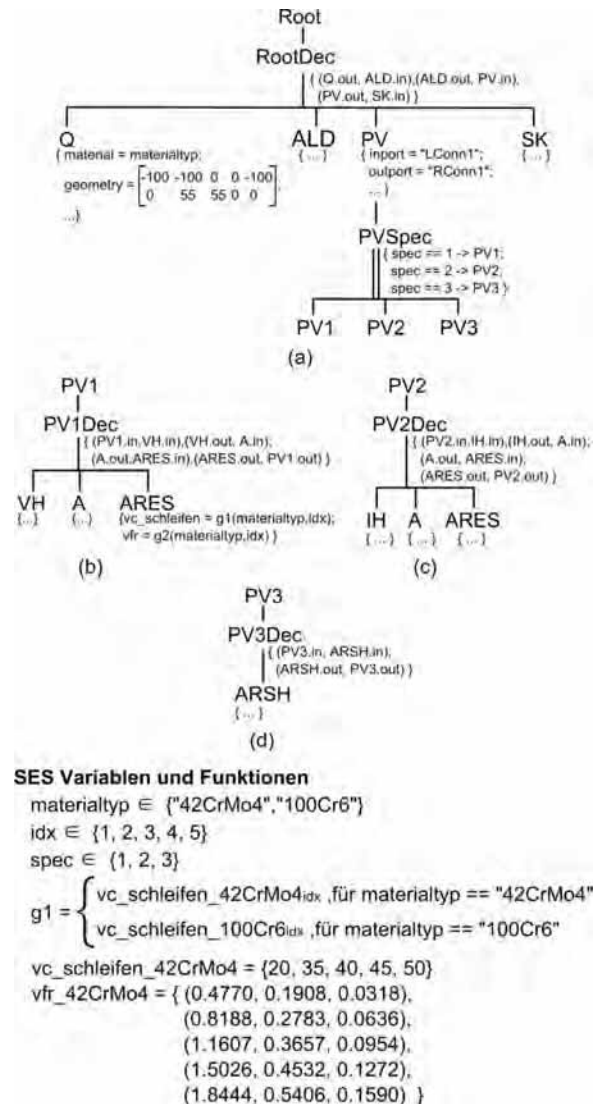


Abbildung 7. SES zur Spezifikation der Prozesskettenstrukturen aus Abbildung 2 einschließlich zulässiger Parametrierungen.

In der dargestellten SES definieren alle Entitäten Knotenattribute {...}. Sie spezifizieren die zulässigen Parametrierungen des entsprechenden Basismodells. Die Entität Q besitzt zum Beispiel die Knotenattribute material und geometry. Das Attribut geometry ist eine Konstante und definiert in Form einer z-x-Koordinatenmatrix eine gemeinsame Anfangsgeometrie für alle zu erzeugenden Werkstücke. Das Attribut material ist eine Variable, deren Wert sich aus der SES Variablen materialtyp ergibt. Die zulässige Wertemenge der SES Variablen materialtyp ist im Abschnitt SES Variablen und Funktionen definiert. Eine konkrete Wertzuweisung an die SES Variablen erfolgt erst während der Modellgenerierung (vgl. Abschnitt

4). Damit definieren die SES Variablen eine Art Eingangsschnittstelle.

Der Spezialisierungs-Knoten PVSpec beschreibt, dass die Entität PV durch eine der Nachfolger-Entitäten PV1-PV3 zu konkretisieren ist, und dass diese die Knotenattribute der Entität PV erben (Vererbungsrelation). Die Auswahlregeln an der PVSpec Kante definieren, wie die Konkretisierung bei der Modellgenerierung zu erfolgen hat. Im vorliegenden Beispiel erfolgt die Konkretisierung in Abhängigkeit von der SES Variablen spec. Die Entitäten der spezialisierten Prozessvarianten PV1-PV3 sind zusammengesetzte Entitäten, die im Kontext der Systemsimulation gekoppelte Systeme darstellen. Ihre Zerlegung wird durch Aspekt-Knoten (PV1Dec-PV3Dec) beschrieben. Wie im Fall der Prozessvariante PV3 kann eine zusammengesetzte Entität auch nur eine Teilentität enthalten.

Die Spezifikation der differenzierten fertigungstechnischen Parameter ist am Blattknoten ARES der Prozessvariante PV1 (Abb. 7b) anhand der Attribute vc\_schleifen und vfr dargestellt. Ihre Belegung ist abhängig vom zu bearbeitenden Werkstofftyp und der für den Werkstofftyp zulässigen Wertemenge. Diese beiden Sachverhalte sind mit SES Variablen und SES Funktionen spezifiziert. In Abbildung 7 sind beispielhaft die SES Variablen mit den zulässigen Wertemengen für den Werkstofftyp 42CrMo4 und die SES Funktion  $g1(\text{materialtyp}, \text{idx})$  definiert. Der Funktionsparameter  $\text{idx}$  ist ebenfalls eine SES Variable. Sie kodiert den Index zur Zuordnung eines Fertigungsparameters aus der geordneten Wertemenge. Die ebenfalls am ARES Knoten benutzte SES Funktion  $g2(\text{materialtyp}, \text{idx})$  ist analog zur Funktion  $g1$  aufgebaut.

Die in [12] eingeführte Definition von SES Variablen und Funktionen stellt eine Erweiterung zur originären SES Ontologie dar. Damit wird eine einfache Spezifikation von Zwangsbedingungen über Knotenattribute ermöglicht, was zu stark reduzierten Baumstrukturen führt. Weiterhin unterstützen die SES Variablen und Funktionen eine einfache Erweiterung von SES Spezifikationen. Soll beispielsweise die betrachtete SES um einen neuen Werkstofftyp erweitert werden, so ist lediglich eine Anpassung bei der Definition der SES Variablen und Funktionen durchzuführen.

Im Vergleich zu den SES basierten Prozesskettenmodellen mit geringerer Komplexität in [6,8] und den produktionsplanerischen SES Modellen in [14,15]

zeichnet sich die SES in Abbildung 7 durch eine wesentlich bessere Übersichtlichkeit und vereinfachte Erweiterbarkeit aus.

## 4 Vom ontologischen Modell zum ausführbaren Simulationsmodell

Die Entwicklung der SES Ontologie ist eng verbunden mit den Arbeiten zur Discrete Event System Specification (DEVS) [16]. In diesem Zusammenhang wurden auch Methoden zur automatisierten Modellgenerierung, in Kombination mit einer Modellbasis (MB), und zur Modellausführung entwickelt. Die Kombination dieser Methoden mit der SES Ontologie wird SES/MB Framework genannt [11]. Die originären SES/MB Entwicklungen in [16,10,11] fokussieren die Generierung von modular-hierarchischen DEVS Modellen und deren Ausführung durch einen DEVS Simulator. Mit der Tiny SES Toolbox [12] werden die Methoden des SES/MB Frameworks allgemein für die MATLAB/Simulink Umgebung [17] sowie die SimEvents Toolbox zur Verfügung gestellt. Im Folgenden werden der prinzipielle SES/MB Ansatz eingeführt und dessen Anwendung zur Generierung von Prozesskettenmodellen in der MATLAB/SimEvents Umgebung unter Verwendung der Tiny SES Toolbox aufgezeigt.

### 4.1 Prinzipieller SES/MB Ansatz

Beim klassischen Vorgehen der Modellbildung und Simulation werden die Schritte Modellierung, Modellgenerierung und Modellausführung manuell durchgeführt. Der SES/MB Ansatz ermöglicht eine automatisierte beziehungsweise teilautomatische Durchführung dieser Schritte. Darüberhinaus wurden in [15,14,9] Methoden zur Kombination mit simulationsbasierten Optimierungstechniken entwickelt. Der prinzipielle SES/MB Ansatz ist in Abbildung 8 dargestellt.

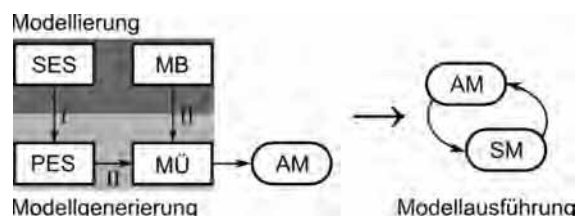


Abbildung 8. Prinzip des SES/MB Ansatzes.

Die Menge der Systemstrukturen und deren Parametrierungsvarianten werden gemäß Abschnitt 3 simulatorunabhängig in einer SES spezifiziert. Die dynamikbeschreibenden Modellkomponenten werden

simulatorabhängig als konfigurierbare Basismodelle mit kompatiblen Ein-/Ausgangsschnittstellen implementiert und in einer MB organisiert. Im Rahmen der Modellgenerierung wird im ersten Schritt (I) simulatorunabhängig aus der SES eine konkrete, parametrisierte Systemstruktur abgeleitet. Dazu werden den SES Variablen gemäß den Wertebereichsdefinitionen Werte zugewiesen. Dies kann interaktiv, skriptgesteuert oder in Analogie zu den Ansätzen in [15,14,9] durch ein Optimierungsverfahren erfolgen. Das Ergebnis der Modellgenerierung ist eine reduzierte Baumstruktur ohne entscheidungsbehaftete Ausdrücke. Die deskriptiven Knoten vom Typ Spezialisierung und Multipler-Aspekt sind aufgelöst und den Attributen an den Entitäten-Knoten sind eindeutige Werte zugewiesen (vgl. Abb. 9). In [11] wird der Vorgang als Prunen bezeichnet, weshalb die reduzierte Baumstruktur Pruned Entity Structure (PES) genannt wird. Im zweiten Schritt (II) generiert ein simulatorabhängiger Modellübersetzer (MÜ), auf Basis der Struktur- und Parameterinformationen der PES, aus den Modellkomponenten der MB ein ausführbares Modell (AM). Das AM kann anschließend manuell oder automatisiert durch den Simulator der Simulationsumgebung ausgeführt werden.

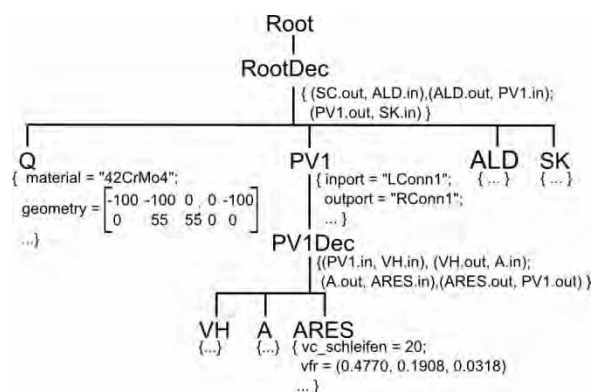
## 4.2 Generierung ausführbarer Prozesskettenmodelle in MATLAB/SimEvents

Die Generierung von Prozesskettenmodellen in der MATLAB/SimEvents Umgebung erfolgt unter Verwendung der in [13] überarbeiteten Modellbibliothek. Als praktischer Modellgenerator wird die Tiny SES Toolbox [12] verwendet. Diese stellt einen Pruning-Algorithmus (Schritt I) und einen Modellübersetzer (Schritt II) für Simulink/SimEvents zur Verfügung. Bevor auf einige Spezifika der Tiny SES Toolbox eingegangen wird, soll werkzeugunabhängig der erste Schritt der Modellgenerierung am Beispiel der SES in Abbildung 7 erläutert werden.

Für den Prune-Vorgang müssen die SES Variablen (materialtyp, idx und spec) mit einem Wert der zugehörigen Wertemenge belegt werden. Mit Hilfe dieser Variablen werden der Spezialisierungs-Knoten und alle Knotenattribute mit alternativen Wertzuweisungen aufgelöst. Für die SES in Abbildung 7 werden folgende Wertzuweisungen angenommen:

- spec = 1;
- materialtyp = '42CrMo4';
- idx = 1;

Durch die Variablenbelegung spec=1 wird am PVSpec Knoten die Entität PV1 mit dem zugehörigen Teilbaum (Abb. 7b) gewählt. Die Belegung der Variablen materialtyp ermöglicht eine konkrete Wertzuweisung beim Attribut material am Entitäten-Knoten Q. Die Funktionen g1 und g2 ermöglichen die Wertzuweisung der Knotenattribute vc\_schleifen und vfr am Entitäten-Knoten ARES. Das Resultat des Prune-Vorgangs, die abgeleitete PES, ist in Abbildung 9 dargestellt. Sie repräsentiert eine vollständig parametrisierte Prozesskettenvariante der Prozesskettenstruktur 1 nach Abbildung 2.



**Abbildung 9.** PES einer Prozesskette mit teilweise angegebener Attributierung.

Zur Verarbeitung mit der Tiny SES Toolbox muss eine SES prädikatenlogisch formuliert werden, wie zum Beispiel:

```

/*Definition von Knoten*/
node(root, []).

...

/*Definition von Kanten */
edge(root, rootDec).

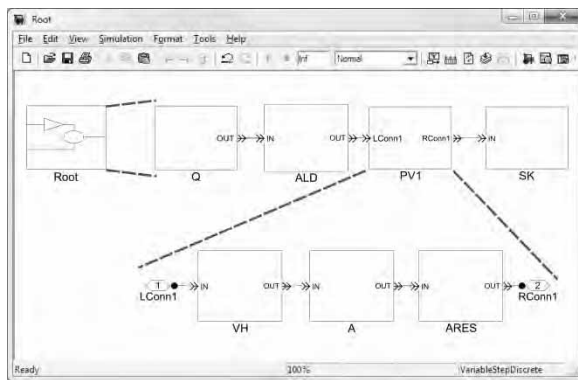
...
    
```

Die genaue Syntax ist in [12] dokumentiert. Bezüglich eines einführenden Beispiels sei auch auf [17] verwiesen. Die Verwendung der prädikatenlogischen Notation ermöglicht eine symbolische Verarbeitung der SES. Die Tiny SES Toolbox verwendet dazu das freie SWI-Prolog. Es sei aber hervorgehoben, dass der Anwender ausschließlich in der MATLAB/Simulink oder MATLAB/SimEvents Umgebung arbeitet, sich die Spezifikation auf wenige prädikatenlogische Ausdrücke beschränkt und in keinem Fall das Erlernen von Prolog erfordert. Im Vergleich zu XML-basierten SES Notationen (vgl. [10,6,8,9,14])

ist die prädikatenlogische Formulierung kompakt und übersichtlicher.

Der in SWI-Prolog implementierte Prune Algorithmus wird vom Nutzer über eine MATLAB Funktion aufgerufen. Als Parameter sind der Name der SES Datei und die Werte der SES Variablen zu übergeben. Die Rückgabe der abgeleiteten PES erfolgt in Form von Strings in MATLAB-Cell-Arrays. Hinsichtlich einer detaillierten Darstellung sei wiederum auf [12,17] verwiesen.

Falls eine gültige PES abgeleitet werden kann, folgt der zweite Schritt der Modellgenerierung, die Modellübersetzung. Dem MÜ werden Informationen bezüglich der Modellstruktur und der Modellparameter in Form der PES Datenstrukturen und die Modellbibliothek (MB) übergeben. Dieser muss die PES Datenstrukturen auswerten, parametrisierte Instanzen der Basismodelle erzeugen und gemäß den Koppelungsbeziehungen ein ausführbares Modell generieren. Die Tiny SES Toolbox stellt den MÜ in Form einer MATLAB Funktion zur Verfügung (vgl. [17]). Diese generiert unter Verwendung der Simulink API und der MB ein ausführbares MATLAB/SimEvents Modell (Abb. 10). Die gestrichelten Linien zeigen die Inhalte der Subsysteme konform zur PES in Abbildung 9.



**Abbildung 10.** Ausführbares MATLAB/SimEvents Modell der ersten Prozesskette.

Da die Methodik automatisierte Modelluntersuchungen zum Ziel hat, verzichtet die Tiny SES Toolbox auf die Generierung von Subsystemen und löst diese automatisch auf.

## 5 Exemplarische Prozesskettenuntersuchung

Zur Validierung der methodischen Ansätze und zur Verifikation der entwickelten Softwarewerkzeuge

wurde eine in [6] beschriebene Prozesskettenuntersuchung durchgeführt. Bei dem Beispiel handelt es sich um ein simulationsbasiertes Optimierungsproblem. Das Ziel der Optimierung kann wie folgt formuliert werden: Ausgehend von den Prozesskettenstrukturen in Abbildung 2 ist eine Prozesskettenvariante (Struktur + Parameter) gesucht, bei der der elektrische Energieverbrauch, der Kühlschmierstoffverbrauch und der Werkzeugverschleiß minimal sind. Die meisten simulationsbasierten Optimierungsansätze betrachten nur Parameteroptimierungsprobleme bezogen auf ein Modell mit einer statischen Modellstruktur [18]. Für die Durchführung der Optimierung mit den genannten Zielen wurde auf den in [14,15] beschriebenen Ansatz zur simulationsbasierten Struktur- und Parameteroptimierung (SSPO) auf Basis des SES/MB Frameworks in Kombination mit einem Genetischen Algorithmus (GA) aufgebaut.

Das Anwendungsbeispiel enthält einen strukturbezogenen Optimierungsparameter am Spezialisierungsknoten PVSPEC. Dieser ist mit der Variablen `spec` kodiert und besitzt die Wertemenge  $\{1,2,3\}$ . Im Gegensatz zu [6] folgen der Parameter und der Wertebereich direkt aus der SES. Dazu kommen weitere Optimierungsparameter in Form von Modellparametern der Basismodelle. Diese sind nur teilweise als Knotenattribute und SES Variablen in Abbildung 7 dargestellt: Schnittgeschwindigkeit  $v_{c,ALD}$ , Vorschub  $f_{ALD}$ , Schnitttiefe  $a_{p,ALD}$ , Schnittgeschwindigkeit  $v_{c,ARES}$  (entspricht `vc_schleifen` in Abb. 7), radiale Zustellung als  $3 \times 1$  Vektor bestehend aus den Zustellungen für das Schruppen, Schlichten und Feinschlichten  $v_{fr,ARES}$  (entspricht `vfr` in Abb. 7).

Der GA stellt für die Modellgenerierung und Modellausführung den übergeordneten Experimentrahmen dar. Der aktuell berechnete Parametervektor des GA initialisiert die SES Variablen für die nächste Modellgenerierung und anschließende Modellausführung. Die Simulationsergebnisse jeder Modellvariante werden durch eine Gütefunktion bewertet. Die Gütefunktion beinhaltet die normierten Verbräuche  $E_v$ ,  $KSS_v$  und  $WKZ_v$ , welche mit Gewichtungsfaktoren behaftet zu einem normierten Verbrauchswert verdichtet werden.

$$G = \frac{1}{3} E_v + \frac{1}{3} KSS_v + \frac{1}{3} WKZ_v \rightarrow \min \quad (1)$$

Abbildung 11 zeigt die Simulationsergebnisse der Prozesskettenuntersuchung. Dabei ist zu erwähnen, dass jede Experimentnummer einer Prozesskettenva-

riante entspricht. Als GA wurde identisch zu [6] die MATLAB Genetic Algorithm Toolbox verwendet.

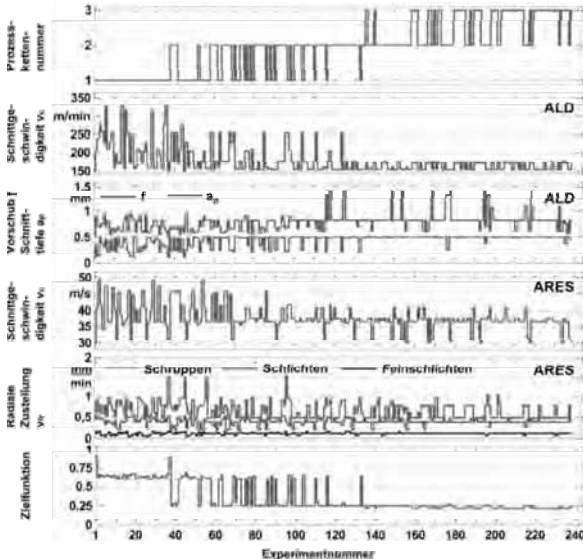


Abbildung 11. Ergebnisse der Prozesskettenuntersuchung.

Der GA untersucht etwas weniger als 240 Prozesskettenvarianten. Als optimale Prozesskettenvariante wird die dritte Prozesskettenstruktur, bestehend aus den Verfahren ALD und ARESH, mit den Fertigungsparametern  $v_{c,ALD} = 175 \text{ mm/min}$ ,  $f_{ALD} = 0,4 \text{ mm/U}$  und  $a_{p,ALD} = 0,75 \text{ mm}$  ermittelt. Diese Ergebnisse entsprechen den Untersuchungsergebnissen in [6].

## 6 Zusammenfassung und Ausblick

Ausgehend von der klassischen Vorgehensweise der Modellierung und Simulation fertigungstechnischer Prozessketten werden anhand einer konkreten Modellbibliothek bestehende Probleme analysiert und Lösungsansätze aufgezeigt. Als gravierendes Problem wird die Beherrschung der Variantenvielfalt bei Prozesskettenuntersuchungen identifiziert. Wie bereits in anderen Arbeiten wird als Lösungsansatz auf dem System Entity Structure/Model Base (SES/MB) Framework aufgebaut. Methodisch basiert dieses auf drei grundlegenden Konzepten: (i) Modellierung der Systemdynamik in Modellen mit kompatiblen Ein-/Ausgangsschnittstellen und deren Organisation in einer MB, (ii) Spezifikation aller zulässigen Systemstrukturen und Modellparametrierungen mittels einer SES, (iii) Bereitstellung von Methoden zur automatisierten Generierung ausführbarer Simulationsmodelle auf Basis der MB und der SES.

Anhand eines konkreten Beispiels wird die Anwendung des SES/MB Frameworks im Rahmen der Prozesskettenplanung aufgezeigt. Zur besseren Beherr-

schung der Parametervielheit wird das Konzept der SES Variablen und Funktionen eingeführt. Darüberhinaus wird am Beispiel der Prozesskettenmodellierung/-simulation die grundlegende Umsetzung des SES/MB Frameworks in der MATLAB/SimEvents Umgebung aufgezeigt. Anhand einer exemplarischen Prozesskettenuntersuchung wird die Kombination mit simulationsbasierten Optimierungsmethoden demonstriert, wobei neben klassischen Modellparametern auch unterschiedliche Strukturvarianten Bestandteil des Optimierungsproblems sind.

Ausblickend sollen nicht nur lineare Prozessketten, sondern gesamte Fertigungsstrukturen mit Verzweigungen und Nebenläufigkeiten betrachtet werden. Darüberhinaus sollen auch umformende Fertigungsverfahren mit einbezogen werden. Neben den bereits betrachteten Ressourcenarten sollen zukünftig auch logistische Größen, wie Pufferkapazitäten, Losgrößen, Maschinenauslastungen usw. mit einbezogen werden.

## 7 References

- [1] M. Junge: *Simulationsgestützte Entwicklung und Optimierung einer energieeffizienten Produktionssteuerung*. Diss., Univ. Kassel, 2007
- [2] A. Pechmann, I. Schöler: *Optimizing Energy Costs by Intelligent Production Scheduling*. In Globalized Solutions for Sustainability in Manufacturing, Proc. Of 18<sup>th</sup> CIRP Conf. on Life Cycle Eng., Springer-Verl., 2011
- [3] K. Mawick, B. Stranziger, W. Bender: *Qualitätsverbesserung und Energieeinsparung bei der Wärmebehandlung in Durchlauföfen durch betriebsorientierte Prozesssimulation*. BFI Berichte Band 5.45.086, Düsseldorf, 2006
- [4] R. Neugebauer: *Energieeffizienz in der Produktion, Untersuchung zum Handlungs- und Forschungsbedarf*. Fraunhofer Gesellschaft, 2008
- [5] N. Weinert: *Vorgehensweise für Planung und Betrieb energieeffizienter Produktionssysteme*. Diss., TU Berlin, 2010
- [6] R. Larek: *Ressourceneffiziente Auslegung von fertigungstechnischen Prozessketten durch Simulation und numerische Optimierung*, Universität Bremen, Dissertation 2012
- [7] R. Larek, E. Brinksmeier, D. Meyer, T. Pawletta, O. Hagendorf: *A discrete-event simulation approach to predict energy con-*

- sumption in machining processes*. In: Production Engineering Research and Development, Springer Pub., 2011/07/14, 575-579, DOI 10.1007/s11740-011-0333-y
- [8] R. Larek, E. Brinksmeier, T. Pawletta, O. Hagendorf: *Model-Based Planning of Resource Efficient Process Chains Using System Entity Structures*. In: Future Trends in Production Engineering - Proc. of the 1st Conf. of the German Academic Soc. for Prod. Eng. (WGP), Berlin, 8th-9th June 2011, G. Schuh, R. Neugebauer, E. Uhlmann (Eds.), Springer Pub., 2013, 361-372 p., ISBN 978-3-642-24490-2
- [9] O. Hagendorf, T. Pawletta, R. Larek: *Meta-Model-based Estimation of Optimal System Variants with MATLAB/Simulink*. SNE Simulation Notes Europe, 22(1), 2012, S. 7-14. (DOI: 10.11128/sne.22.tn.10105)
- [10] B.P. Zeigler, P. Hammonds: *Modeling and Simulation-Based Data Engineering*. Elsevier Academic Press, 2007
- [11] B.P. Zeigler, H. Praehofer, T.G. Kim: *Theory of Modeling and Simulation*. Second Edition, Elsevier Academic Press, 2000
- [12] T. Schwatinski, A. Schmidt, T. Pawletta: Tiny SES Toolbox., Website 2012, [http://www.mb.hs-wismar.de/cea/SES\\_Tbx/sesToolboxMain.html](http://www.mb.hs-wismar.de/cea/SES_Tbx/sesToolboxMain.html)
- [13] A. Schmidt: *Entwicklung eines Simulators zur energetischen Bewertung von Prozessketten der spanenden Bauteilfertigung mit MATLAB/SimEvents*. Master-Thesis, Hochschule Wismar, 2012
- [14] O. Hagendorf: *Simulation Based Parameter and Structure Optimisation of Discrete Event Systems.*, Liverpool John Moores University, Dissertation, 2009
- [15] O. Hagendorf, T. Pawletta: *A Framework for Simulation Based Structure and Parameter Optimization of Discrete Event Systems*. In: Discrete-Event Modeling and Simulation: Theory and Applications, Editors: G.A. Wainer and P.J. Mosterman, CRC Press Inc. of Taylor & Francis Group, USA, 2011, S. 199-222, ISBN-13: 9781420072334
- [16] B.P. Zeigler: *Multifaceted Modelling and Discrete Event Simulation.*, Academic Press, 1984
- [17] T. Schwatinski, T. Pawletta: *Ontologische Modellierung und Modellgenerierung in der MATLAB/Simulink Umgebung: Die „Tiny SES Toolbox“*. 2013 (in diesem Tagungsband)
- [18] L. März, W. Krug, O. Rose, G. Weigert: *Simulation und Optimierung in Produktion und Logistik*, Springer Verlag, 2011



# Modellbasierter Entwurf ereignisdiskreter Steuerungen

Dirk Weidemann<sup>1</sup>, Robin Diekmann<sup>1</sup>

<sup>1</sup>Fachhochschule Bielefeld

{*dirk.weidemann, robin.diekmann*}@fh-bielefeld.de

Es wird eine Methodik zum modellbasierten Entwurf ereignisdiskreter Steuerungen präsentiert. Diese beruht auf dem Formalismus der Supervisory Control Theory, beinhaltet aber eine Erweiterung, um Erzwingbarkeit von Ereignissen zu berücksichtigen und somit die Synthese von Ablaufsteuerungen möglich zu machen. Anhand eines Beispiels werden die sukzessiven Schritte der Steuerungssynthese erläutert. Zunächst wird auf Basis von Zustandsautomaten das unter Einhaltung einer Spezifikation größtmögliche zulässige Verhalten bestimmt. Daraus lässt sich mittels eines Optimierungsverfahrens eine kostenoptimale Steuerung ermitteln, welche in ablauffähigen SPS-Code übersetzt werden kann. Im Anschluss wird die Anwendung aller beschriebenen Schritte auf ein reales System gezeigt.

## 1 Einleitung

Während sich beim Entwurf kontinuierlicher Regler modellbasierte Verfahren etabliert haben, werden ereignisdiskrete Steuerungen auch heute noch häufig ad hoc entworfen. Bei dieser heuristischen Vorgehensweise entfällt ein großer Anteil des Arbeitsaufwandes auf das Testen und Nachbessern des Steuerungsalgorithmus, sei es mittels einer Simulation oder direkt an der realen Steuerstrecke. Die Korrektheit einer so entworfenen Steuerung kann nur für getestete Szenarien sichergestellt werden. Mit der in den 1980er Jahren entstandenen Supervisory Control Theory (SCT) hingegen existiert ein modellbasiertes Verfahren zum strukturierten Steuerungsentwurf für ereignisdiskrete Systeme [1]. Der Vorteil dieser Methode liegt in der mathematisch garantierten Korrektheit der synthetisierten Steuerung, sofern die Modellierung der ungesteuerten Strecke und die Formalisierung der Spezifikation fehlerfrei erfolgt sind. Somit findet gegenüber dem heuristischen Ansatz eine Arbeitsverlagerung vom Verifizieren der Steuerung hin zur Modellbildung statt.

Während des hier vorgestellten Entwurfsverfahrens (s. Abbildung 1) wird aus dem Modell der ungesteuerten Strecke und der formalen Spezifikation die sogenannte supremale steuerbare Untersprache bestimmt, welche die größtmögliche Teilmenge des zulässigen Verhaltens unter Einhaltung der Spezifikation repräsentiert. Diese dient als Ausgangspunkt für ein Optimierungsverfahren, um ein kostenoptimales Verhalten zu ermitteln. Der letzte Schritt besteht in der Übersetzung in SPS-Code, um die Ablaufsteuerung zu implementieren. Die Schritte der Synthese der supremalen steuerbaren Untersprache, der Opti-

mierung und der Implementierung lassen sich algorithmisch formulieren, sodass sie prinzipiell automatisiert erfolgen werden können.

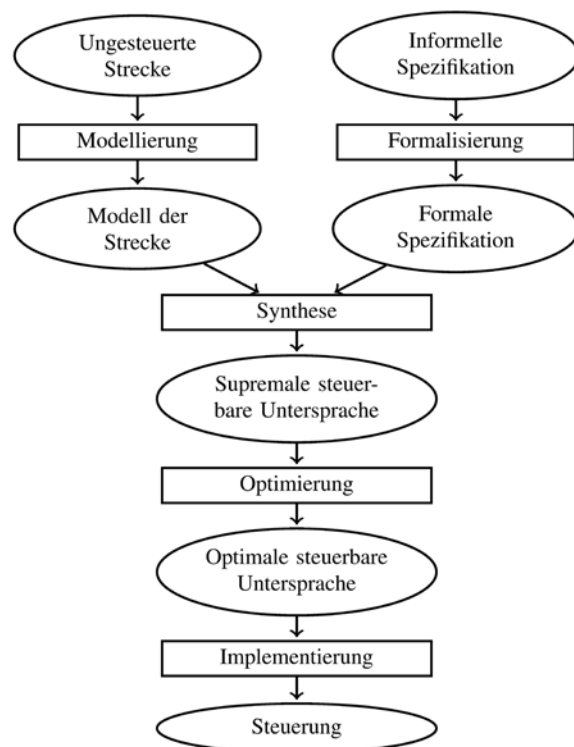


Abbildung 1. Modellbasierter Steuerungsentwurf

## 2 Supervisory Control Theory (SCT)

Nach der Supervisory Control Theory ist der geschlossene Steuerkreis unterteilt in die ungesteuerte Strecke  $G$  und die Steuerung  $S$ , genannt Supervisor. Das Ziel bei der Synthese des Supervisors besteht darin, die größte Teilmenge des zulässigen Verhaltens

zu finden, bei dem sichergestellt werden kann, dass ein zuvor formal spezifiziertes Verhalten nicht verlasen wird. Diese Teilmenge wird durch die supremale steuerbare Untersprache  $K^\uparrow$  ausgedrückt. Sowohl das ungesteuerte Prozessverhalten, wie auch die formale Spezifikation lassen sich durch deterministische Zustandsautomaten oder alternativ durch formale Sprachen repräsentieren. Der zugrundeliegende Formalismus wird ausführlich in [2] erläutert.

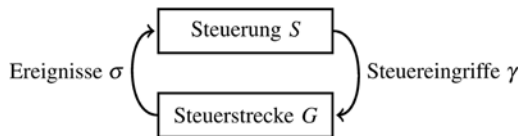


Abbildung 2. Geschlossener Steuerkreis

Die Steuerung  $S$  reagiert auf Ereignisse  $\sigma$  der Strecke  $G$ , indem sie über Steuereingriffe jeder Ereignisfolge eine Menge  $\gamma$  von als nächstes erlaubten Ereignissen zuordnet, wie in Abbildung 2 dargestellt. Nach der klassischen SCT ist die Ereignismenge  $\Sigma$  unterteilt in zwei disjunkte Mengen. Zum einen die Menge  $\Sigma_{uc}$  von nichtverhinderbaren Ereignissen, deren Auftreten durch die Steuerung nicht beeinflusst werden kann und folglich zu jeder Zeit zugelassen sein muss. Zum anderen die Menge  $\Sigma_c$  von verhinderbaren Ereignissen, deren Auftreten die Steuerung unterbinden kann. Dieses wird ausgedrückt durch das Steuerungsgesetz der klassischen SCT

$$\Gamma = \{\gamma \subseteq \Sigma \mid \Sigma_{uc} \subseteq \gamma\}, \quad (1)$$

wobei  $\Gamma$  die Menge aller möglichen Steuereingriffe  $\gamma$  darstellt. Die Steuerung hat keine Möglichkeit, aktiv Ereignisse zu erzwingen. Die klassische SCT eignet sich deshalb zur Synthese von Verriegelungssteuerungen, nicht aber von Ablaufsteuerungen.

### Beispiel: Klassische SCT

Abbildung 3 zeigt das Labormodell einer Produktionsanlage. Über ein Förderband werden Paletten transportiert, wobei über einen Sensor detektiert werden kann, ob sich eine Palette an einer bestimmten Stelle befindet.

Zunächst wird das mögliche Verhalten des Sensors ( $s_m$ ) als Zustandsautomat modelliert (s. Abbildung 4), welcher entweder ein- (Zustand  $y_0$ ) oder ausgeschaltet (Zustand  $y_1$ ) sein kann. Gleiches gilt für das Förderband ( $fb$ ) (s. Abbildung 5), wobei hier außerdem berücksichtigt wird, dass der Sensor seinen Signalwert nur ändern kann, wenn das Förderband eingeschaltet ist (Zustand  $f_1$ ). In den Automaten erfolgt die

Kennzeichnung initialer Zustände über einen kleinen Pfeil. Markierte, d.h. gewünschte Endzustände sind über einen Doppelkreis und nichtverhinderbarer Ereignisse über ein vorangestelltes Ausrufezeichen gekennzeichnet. Das Einschalten der entsprechenden Komponente wird durch eine nachgestellte  $1$ , das Ausschalten durch eine  $0$  ausgedrückt.

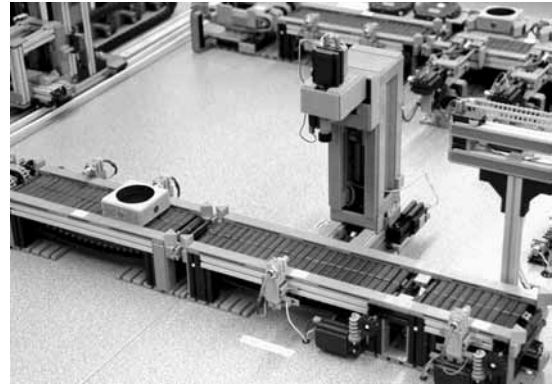


Abbildung 3. Labormodell

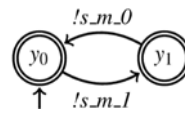


Abbildung 4. Modell des Sensors

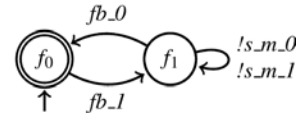


Abbildung 5. Modell des Förderbandes

Das kombinierte Verhalten beider Automaten lässt sich über die Operation der Parallelkomposition finden. Hierbei wird berücksichtigt, dass Automaten für private, d.h. ausschließlich in ihnen vorkommende Ereignisse unabhängig schalten dürfen. Beim Schalten nicht privater Ereignisse hingegen müssen alle das entsprechende Ereignis beinhaltende Automaten gemeinsam schalten. Das Ergebnis der Parallelkomposition ist in Abbildung 6 gezeigt und stellt für das angenommene Beispiel das Modell  $G$  der Steuerstrecke dar.

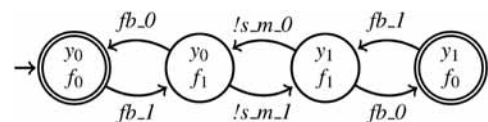


Abbildung 6. Modell  $G$  des ungesteuerten Verhaltens

Darüber hinaus muss das Sollverhalten als Automat modelliert werden. In diesem Fall ist die einzige Anforderung an das Verhalten des geschlossenen Steuerkreises, dass der Prozess erst abgeschlossen sein soll, nachdem eine Palette vom Sensor detektiert wurde und dass sie ihre Position vor dem Sensor nicht wie-

der verlassen darf. Formal wird diese Spezifikation durch den in Abbildung 7 dargestellten Automaten ausgedrückt.

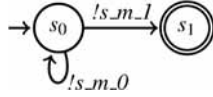


Abbildung 7. Formale Spezifikation

Um das zulässige Verhalten zu bestimmen, wird eine Parallelkomposition des Automaten der Spezifikation mit dem Modell der Strecke durchgeführt. Das Ergebnis zeigt Abbildung 8.

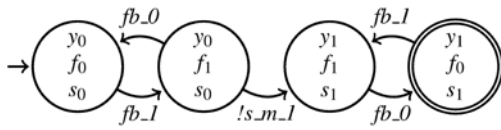


Abbildung 8. Automat  $H$  des zulässigen Verhaltens

Eine zulässige Folge von Ereignissen zum Erreichen des markierten Zustandes besteht beispielsweise aus dem Einschalten des Förderbandes ( $fb_1$ ), dem Einschalten des Sensors ( $!s_m_1$ ) und dem Ausschalten des Förderbandes ( $fb_0$ ).

Zur Bestimmung des Supervisors muss geprüft werden, ob die Einhaltung des zulässigen Verhaltens durch diesen sichergestellt werden kann. Ein formaler Algorithmus hierfür ist beispielsweise in [2] angegeben. Dazu wird der Automat  $H$  des zulässigen Verhaltens mit dem Automaten  $G$  des ungesteuerten Verhaltens verglichen. In  $G$  ist beginnend vom initialen Zustand die Ereignisfolge  $fb_1, !s_m_1, !s_m_0$  möglich. Diese ist aber in  $H$  nicht enthalten und damit unzulässig. Nach dem Auftreten der Ereignisfolge  $fb_1, !s_m_1$  müsste der Supervisor zur Sicherstellung des zulässigen Verhaltens bewirken, dass das Ereignis  $!s_m_0$  nicht auftritt. Allerdings handelt es sich hierbei um ein nichtverhinderbares Ereignis, welches der Supervisor nicht beeinflussen kann. Um trotzdem diese unzulässige Ereignisfolge ausschließen zu können, müsste bereits vorher durch den Supervisor eingegriffen und verhindert werden, dass die Ereignisfolge  $fb_1, !s_m_1$  auftritt. Da  $!s_m_1$  wiederum nichtverhinderbar ist, besteht die einzige Möglichkeit im Verhindern von  $fb_1$ . Das Förderband darf also zu Beginn des Prozesses gar nicht erst eingeschaltet werden, damit ein Verlassen des zulässigen Verhaltens nicht möglich ist. Andererseits soll während des Prozesses der markierte Zustand erreicht werden, was ohne Einschalten des Förderbandes nicht möglich ist. In diesem Fall kann also kein Supervisor

gefunden werden, d.h. die supremale steuerbare Untersprache  $K^\dagger$  entspricht der leeren Menge.

### 3 SCT mit erzwingbaren Ereignissen

Wie sich gezeigt hat, stößt die klassische SCT bei dem in Abschnitt 2 diskutierten einfachen Beispiel an ihre Grenzen. Dieses liegt an der Annahme, Ereignisse seien aus Sicht der Steuerung lediglich entweder verhinderbar oder nichtverhinderbar. Viele reale Steuerungen haben allerdings darüber hinaus die Möglichkeit, Ereignisse aktiv zu erzwingen. Um dieses in die SCT zu integrieren, existieren mehrere Ansätze (vgl. [3,4]), die eine kompliziertere Modellierungsstrategie zur Berücksichtigung von Erzwingbarkeit verfolgen. Hierbei besteht allerdings der Nachteil, dass der ohnehin komplexe Prozess der Modellbildung weiter erschwert wird.

Eine Alternative besteht in der Modifikation des Steuerungsgesetzes der klassischen SCT. Hierzu werden die beiden disjunkten Mengen der erzwingbaren Ereignisse  $\Sigma_f$  und der nichterzwingbaren Ereignisse  $\Sigma_{uf}$  eingeführt, in die sich  $\Sigma$  unterteilen lässt. Es wird keine allgemeine Annahme über den Zusammenhang zwischen Verhinderbarkeit und Erzwingbarkeit gemacht, da jede mögliche Kombination der Eigenschaften erzwingbar oder nichterzwingbar und verhinderbar oder nichtverhinderbar sinnvoll sein kann, wie Tabelle 1 zeigt. Erzwingbarkeit von Ereignissen wird durch ein nachgestelltes Sternchen gekennzeichnet.

Ereignis	Eigenschaft	Beispiel
$\sigma^* \in \Sigma_c \cap \Sigma_f$	Verhinderbar und erzwingbar	Stelleingriff durch eine Steuerung
$\sigma \in \Sigma_c \cap \Sigma_{uf}$	Verhinderbar und nichterzwingbar	Stelleingriff durch einen Bediener
$!\sigma^* \in \Sigma_{uc} \cap \Sigma_f$	Nichtverhinderbar und erzwingbar	Ansprechen einer Schutzeinrichtung
$!\sigma \in \Sigma_{uc} \cap \Sigma_{uf}$	Nichtverhinderbar und nichterzwingbar	Änderung eines Sensorsignals

Tabelle 1. Mögliche Klassen von Ereignissen

Durch die Einführung erzwingbarer Ereignisse erweitern sich die Möglichkeiten einer Steuerung derart, dass sie nun entweder durch das Erzwingen einiger Ereignisse andere Ereignisse übergehen kann oder, wie im Fall der klassischen SCT, das Auftreten einiger Ereignisse verhindern kann. Wie in [5] vorge-

schlagen, lässt sich dieses durch eine Erweiterung des Steuerungsgesetzes ausdrücken:

$$\Gamma = \{\gamma \subseteq \Sigma \mid (\Sigma_{uc} \subseteq \gamma) \vee (\gamma \subseteq \Sigma_f)\} \quad (2)$$

Auch im Fall des Steuerungsgesetzes nach (2) existiert eine supremale steuerbare Untersprache. Ein entsprechender Beweis und ein Algorithmus zur Berechnung sind in [6] angegeben.

Es sollte beachtet werden, dass ein unter Berücksichtigung des erweiterten Steuerungsgesetzes berechneter Supervisor nicht in jedem Fall einer implementierbaren Steuerung entspricht. Dieses liegt daran, dass nach (2) mehrere erzwingbare Ereignisse gleichzeitig zulässig sind. Zum Übergehen von Ereignissen muss aber genau ein Ereignis unverzüglich durch die Steuerung erzwungen werden. Die Bedeutung der supremalen steuerbaren Untersprache  $K^\dagger$  ist vielmehr die, dass sie alle möglichen, mit der Spezifikation zu vereinbarenden Ereignisfolgen enthält, die auf einen markierten Zustand führen. Hieraus kann im nächsten Schritt eine kostenoptimale Ablaufsteuerung extrahiert werden, wobei die Einhaltung der Spezifikation garantiert ist.

#### Beispiel: SCT mit erzwingbaren Ereignissen

Für das in Abschnitt 2 präsentierte Beispiel soll nun unter Berücksichtigung des erweiterten Steuerungsgesetzes die supremale steuerbare Untersprache bestimmt werden. Hierzu wird angenommen, dass das Starten und Stoppen des Förderbandes, d.h. die Ereignisse  $fb\_1^*$  und  $fb\_0^*$  erzwingbar sind. Wie oben erörtert, ist nach der Ereignisfolge  $fb\_1^*$ ,  $!s\_m\_1$  das Ereignis  $!s\_m\_0$  nicht zulässig. Nach  $fb\_1^*$ ,  $!s\_m\_1$  kann aber nun  $fb\_0^*$  erzwungen werden, wodurch  $!s\_m\_0$  übergangen und in  $H$  (s. Abbildung 8) der markierte Zustand erreicht wird. Auf diese Weise ist sichergestellt, dass das zulässige Verhalten nicht verlassen wird. Eine entsprechende Steuerung wird sofort nach dem Ansprechen des Sensors, d.h. Ereignis  $!s\_m\_1$  das Stoppen des Förderbandes, d.h. Ereignis  $fb\_0^*$  auslösen. Der in Abbildung 8 dargestellte Automat des zulässigen Verhaltens repräsentiert nun auch die supremale steuerbare Untersprache  $K^\dagger$ . Im Gegensatz zur klassischen SCT kann also eine Lösung gefunden werden, welche die Einhaltung der Spezifikation sicherstellt.

#### Softwaregestützte Berechnung der supremalen steuerbaren Untersprache

Während für die unter Abschnitt 2 und 3 diskutierten Beispiele die Berechnung der supremalen steuerbaren

Untersprache manuell durchgeführt werden kann, lässt sich diese für komplexere Systeme praktisch nur automatisiert bestimmen. Für die klassische SCT existieren hierzu einige Software-Werkzeuge, z.B. das in [7] vorgestellte Supremica.

Bei der Berücksichtigung erzwingbarer Ereignisse durch das erweiterte Steuerungsgesetz kann der in [6] aufgeführte Algorithmus genutzt werden, welcher allerdings noch nicht in einem Software-Werkzeug umgesetzt wurde. Eine Alternative, welche die Nutzung vorhandener Software gestattet, liegt in der Verwendung eines generischen Hilfsautomaten  $E$ , der in Abbildung 9 gezeigt ist. Diese Möglichkeit kann dann genutzt werden, wenn die Menge der erzwingbaren Ereignisse eine Teilmenge der verhinderbaren Ereignisse ist, was in vielen Fällen zutrifft. Durch eine Parallelkomposition wird der Automat  $E$  mit den anderen Komponenten der Strecke verknüpft. Das Hilfseignis  $\tau$  darf in keinem anderen Automaten vorkommen. Dann wird die supremale steuerbare Untersprache nach der Methode der klassischen SCT bestimmt und zuletzt  $\tau$  aus allen Ereignisfolgen entfernt.

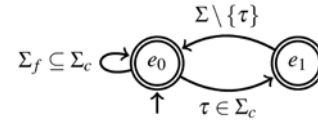


Abbildung 9. Generischer Hilfsautomat  $E$  zur Berücksichtigung erzwingbarer Ereignisse

## 4 Kostenoptimierung

Die supremale steuerbare Untersprache  $K^\dagger$  enthält alle Ereignisfolgen, die das Erreichen eines markierten Zustandes bei gleichzeitiger Einhaltung der formalen Spezifikation garantieren. Zur Extrahierung einer Ablaufsteuerung können somit beliebige sukzessive Steuereingriffe  $\gamma_1, \dots, \gamma_n$  gewählt werden, welche in  $K^\dagger$  enthaltenen und auf einen markierten Zustand führenden Ereignisfolgen entsprechen. Es muss allerdings sichergestellt sein, dass gewählte Ereignisfolgen nicht durch nichtverhinderbare Ereignisse verlassen werden können.

Zur Bestimmung der Ablaufsteuerung ist es nicht sinnvoll, willkürliche Ereignisfolgen aus  $K^\dagger$  auszuwählen, sondern diese über eine Kostenoptimierung festzulegen. Es sind verschiedene Arten von mit Ereignissen verbundenen Kosten möglich, wie beispielsweise umgesetzte Energie, zurückgelegte Strecke, personeller Aufwand, finanzielle Kosten, Verschleiß von Maschinenteilen, etc.

Werden allen Zustandsübergängen im Automaten der supremalen steuerbaren Untersprache positive Kosten  $C$  zugeordnet, so kann das kostenoptimale Verhalten algorithmisch bestimmt werden, welches sich durch die sogenannte optimale steuerbare Untersprache  $K_{opt}$  ausdrücken lässt. Ein entsprechender Algorithmus ist in [8] aufgeführt und ausführlich erörtert. Hiermit kann das kostenoptimale Verhalten azyklischer oder zyklischer Automaten unter Berücksichtigung des in (2) angegebenen Steuerungsgesetzes berechnet werden.

### Beispiel: Kostenoptimierung

Der Automat der supremalen steuerbaren Untersprache (s. Abbildung 8) dient als Ausgangspunkt für die Kostenoptimierung. Der Einfachheit halber wird angenommen, dass jede Transition Kosten von  $C = 1$  hervorruft. Das Ergebnis der Optimierung ist in Abbildung 10 dargestellt. Die Kosten  $C$  eines Zustandes entsprechen den kumulierten vom Anfangszustand aus auftretenden Kosten, die Kosten  $C^S$  entsprechen den bis zum Erreichen eines markierten Zustandes maximal möglichen Kosten unter dem Einfluss der optimalen Steuerung.

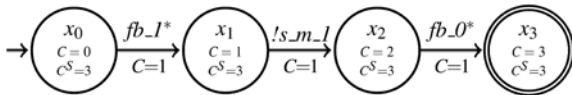


Abbildung 10. Ergebnis der Kostenoptimierung

Hier besteht die optimale steuerbare Untersprache  $K_{opt}$  mit  $fb\_I^*$ ,  $!s\_m\_I$ ,  $fb\_O^*$  aus genau einer Ereignisfolge, was bedingt durch das Vorhandensein nicht-verhinderbarer Ereignisse im Allgemeinen allerdings nicht der Fall ist (vgl. [8]). Die  $K_{opt}$  entsprechende Steuerung startet zu Beginn des Prozesses das Förderband, wartet auf die Aktivierung des Sensors und stoppt sofort nach dieser das Förderband.

## 5 Implementierung

Zur Implementierung der Steuerung kann der die optimale steuerbare Untersprache repräsentierende Automat direkt in SPS-Code übersetzt werden. Im Folgenden wird eine Übersetzung in die Siemens S7-spezifische Sprache AWL gezeigt, die einen Dialekt des nach IEC 61131-3 definierten Sprachstandards Instruction List (IL) darstellt und leicht daran angepasst werden kann. Der Vorteil bei der Übersetzung des Automaten in eine textbasierte Programmiersprache liegt in der einfacheren Automatisierbarkeit, prinzipiell ist aber auch die Übertragung in eine grafi-

sche Programmiersprache denkbar, wie beispielsweise Ladder Diagram (LD).

Der erzeugte Programmcode besteht aus zwei Teilen: Ein Teil spiegelt die Transitionsstruktur des zugrundeliegenden Automaten wider, während der andere Teil den Steuereingriffen entspricht. Dabei wird jeder Zustand durch genau eine boolesche Variable, genannt Merker, repräsentiert. Zu Beginn ist nur der zum initialen Zustand korrespondierende Merker auf „wahr“ gesetzt. Ereignisse werden durch die Änderung von Signalwerten berücksichtigt.

Eine durch das Ereignis  $\sigma_1$  ausgelöste Transition vom Zustand  $x_0$  in den Zustand  $x_1$  wird folgendermaßen umgesetzt: Wenn der zum Zustand  $x_0$  korrespondierende Merker  $M(x_0)$  auf „wahr“ gesetzt ist und die zum Ereignis  $\sigma_1$  korrespondierende Signalwertänderung  $V(\sigma_1)$  detektiert wird, dann wird  $M(x_0)$  auf „falsch“ gesetzt und  $M(x_1)$  auf „wahr“ gesetzt.

Die Implementierung eines Steuereingriffes hängt von den Eigenschaften des Ereignisses ab: Ereignisse, die gleichzeitig nichtverhinderbar und nichterzwingbar sind, müssen nicht berücksichtigt werden. Erzwingbare Ereignisse werden sofort ausgelöst, sobald der Merker des entsprechenden Zustandes „wahr“ wird. Zur Schnittmenge der verhinderbaren und nichterzwingbaren Ereignisse gehörende Ereignisse sind meist über die gemeinsame Änderung eines Sensorsignals und eines Aktorsignals mit zwei Signalwertänderungen  $V(\sigma)$  und  $V'(\sigma)$  verbunden. Dies können beispielsweise ein durch einen Bediener betätigter Schalter und ein damit ausgelöster Start einer Maschine sein. Durch die Steuerung wird dieses nur erlaubt, wenn der zum Zustand korrespondierende Merker auf „wahr“ gesetzt ist, für den das verhinderbare Ereignis zugelassen ist.

Abbildung 11 zeigt einen beispielhaften Automaten und den sich daraus ergebenden SPS-Code.

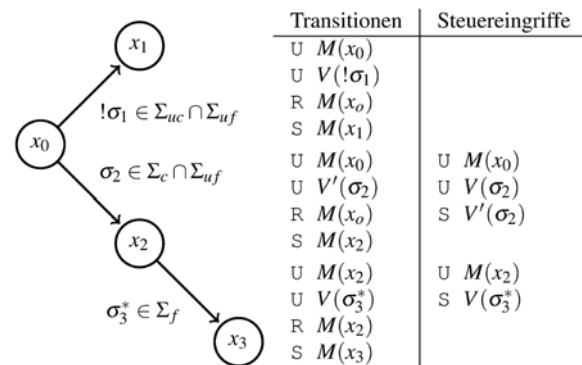


Abbildung 11. Beispielautomat und SPS-Code

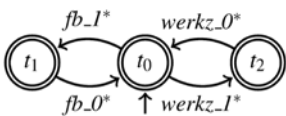
## 6 Anwendung

In den bis jetzt aufgeführten Beispielen wurde lediglich ein kleiner Teil der in Abbildung 3 dargestellten Produktionsstraße behandelt. Nun soll gezeigt werden, wie mittels der präsentierten Methodik eine Steuerung für einen umfangreicheren Teil des Systems erstellt werden kann. Der im Folgenden behandelte Part besteht aus dem Förderband, welches Paletten transportiert, sowie einer mit einem Werkzeug ausgestatteten Maschine. Die Maschine kann vor oder zurück und hoch oder runter gefahren werden, um das Werkzeug zu positionieren. Dieses ist entweder ein- oder ausgeschaltet. Sensoren können die Position einer Palette vor der Maschine und am Ende des Förderbandes detektieren. Über das Förderband soll eine Palette vor der Maschine positioniert werden, ihr Inhalt soll von dem Werkzeug bearbeitet werden und die Palette soll danach zum Ende des Förderbandes transportiert werden.

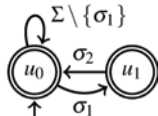
Zunächst werden alle Sensoren und Aktoren in der unter Abschnitt 2 beschriebenen Weise modelliert, wobei aber nun alle Zustände als markiert angenommen werden. Zur Erstellung des Modells  $G$  der ungesteuerten Strecke werden die Automaten mittels Parallelkomposition verknüpft.

Die erforderlichen Spezifikationsautomaten lassen sich in Sicherheits- und Ablaufspezifikationen unterteilen. Der Automat einer Sicherheitsspezifikation ist in Abbildung 12 gezeigt und fordert, dass das Förderband und das Werkzeug nicht gleichzeitig eingeschaltet sein dürfen.

In einigen Fällen müssen bestimmte Ereignisse erzwungen werden, um Fehler zu verhindern. Beispielsweise muss die Bewegung der Maschine gestoppt werden, sobald sie eine ihrer Endpositionen erreicht. Der Automat einer generischen Sicherheitsspezifikation, die das Erzwingen des Ereignisses  $\sigma_2$  direkt nach dem Auftreten von  $\sigma_1$  fordert, ist in Abbildung 13 gezeigt.



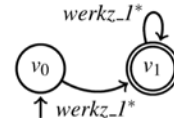
**Abbildung 12.**  
Beispielhafte Sicherheitspezifikation



**Abbildung 13.**  
Sicherheitsspezifikation für Erzwingen

Ablaufspezifikationen beschreiben gewünschtes Systemverhalten. Eine Möglichkeit der Umsetzung be-

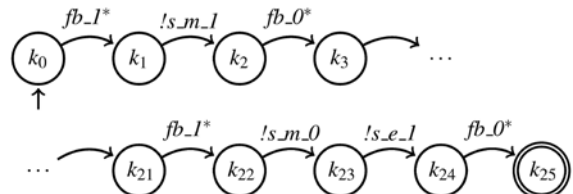
steht darin, nur in Automaten der Ablaufspezifikation nichtmarkierte Zustände zu verwenden und in sämtlichen anderen Automaten alle Zustände zu markieren. Durch die Definition der Parallelkomposition werden somit nur durch Ablaufspezifikationen gewünschte Endzustände des gesteuerten Prozesses vorgegeben. Der in Abbildung 14 dargestellte Automat einer Ablaufspezifikation legt fest, dass das Werkzeug während des Prozesses mindestens einmal eingeschaltet wird.



**Abbildung 14.** Beispielhafte Ablaufspezifikation

Der Automat  $H$  des zulässigen Verhaltens wird durch Parallelkomposition aller Spezifikationsautomaten mit Automat  $G$  und – zur Berücksichtigung von Erzwingbarkeit – mit Hilfsautomat  $E$  (s. Abbildung 9) ermittelt. Zur Bestimmung des Automaten der supremalen steuerbaren Untersprache kann Supremica genutzt werden. Über den Befehl *Synthesize* wird der Supervisor berechnet. Danach wird das Hilfsereignis  $\tau$  mittels *Hide events* und *Minimize* entfernt, um den Automaten der supremalen steuerbaren Untersprache  $K^\uparrow$  zu erhalten.

Es wird abermals angenommen, dass alle Transitionen Kosten von  $C = 1$  hervorrufen. Der Algorithmus zur Berechnung des kostenoptimalen Verhaltens  $K_{opt}$  steht noch nicht in einem Software-Werkzeug zur Verfügung, weshalb  $K_{opt}$  „von Hand“ bestimmt werden muss. Um die zu implementierende Steuerung abzuleiten, kann daraus eine auf den markierten Zustand führende Ereignisfolge gewählt werden. Für dieses Beispiel ist in jedem Fall sichergestellt, dass diese nicht durch nichtverhinderbare Ereignisse verlassen wird. Ein Teil des entsprechenden Automaten ist in Abbildung 15 dargestellt.



**Abbildung 15.** Automat der implementierten Steuerung (Ausschnitt)

Die Übersetzung in SPS-Code wurde ebenfalls noch nicht implementiert und muss manuell ausgeführt

werden. Einen Ausschnitt des ablauffähigen SPS-Codes, der dem oberen Teil des Automaten aus Abbildung 15 entspricht, zeigt Abbildung 16.

Transitionen	Steuereingriffe
U M 20.0	U M 20.0
U A 8.2	S A 8.2
R M 20.0	
S M 20.1	
U M 20.1	
U E 0.4	
R M 20.1	
S M 20.2	
U M 20.2	U M 20.2
UN A 8.2	R A 8.2
R M 20.2	
S M 20.3	
...	...

**Abbildung 16.** SPS-Code der implementierten Steuerung (Ausschnitt)

Tabelle 2 fasst die einzelnen Schritte des Steuerungsentwurfs für das präsentierte Anwendungsbeispiel zusammen.

Entwurfsschritt	Umfang
Komponenten der ungesteuerten Strecke	13 Automaten
Automat $G$ des Modells der Strecke	1620 Zustände, 9648 Transitionen
Sicherheitsspezifikationen	17 Automaten
Ablaufspezifikationen	5 Automaten
Automat $H$ des zulässigen Verhaltens	854 Zustände, 1847 Transitionen
Automat der supremalen steuerbaren Untersprache $K^\uparrow$	227 Zustände, 432 Transitionen
Automat der implementierten Steuerung	26 Zustände, 25 Transitionen
SPS-Code	152 Zeilen

**Tabelle 2.** Zusammenfassung: Anwendung der vorgestellten Methodik

## 7 Zusammenfassung

Die vorgestellte Erweiterung des Formalismus der klassischen Supervisory Control Theory ermöglicht die Berücksichtigung von erzwingbaren Ereignissen. Damit liegt eine Methodik vor, um Ablaufsteuerungen auf Basis von Zustandsautomaten zu synthetisieren. Ausgehend vom Modell der ungesteuerten Stre-

cke und der formalen Spezifikation wird zunächst die unter Einhaltung der Spezifikation größtmögliche Teilmenge des zulässigen Systemverhaltens bestimmt. Im nächsten Schritt wird hieraus mittels eines Optimierungsverfahrens ein die Ablaufsteuerung repräsentierender Automat extrahiert. Zur Implementierung kann dieser automatisiert in ablauffähigen SPS-Code übersetzt werden. Alle Schritte wurden erfolgreich auf ein reales Beispielsystem angewandt.

## 8 Quellen

- [1] P. J. Ramadge und W.M. Wonham. *The Control of Discrete Event Systems*. Proc. of the IEEE, 77(1), S. 81–98, 1989.
- [2] C. G. Cassandras und S. Lafontaine. *Introduction to Discrete Event Systems*. Springer, USA, 2008.
- [3] W. M. Wonham. *Supervisory Control of Discrete Event Systems*. Lecture notes, [www.control.utoronto.ca/~wonham](http://www.control.utoronto.ca/~wonham)
- [4] M. Cantarelli und J.-M. Roussel. *Reactive control system design using the Supervisory Control Theory: evaluation of possibilities and limits*. Proc. of 9th WODES, S. 200–205, 2008.
- [5] Y.-L. Chen und F. Lin. *An Optimal Effective Controller for Discrete Event Systems*. Asian Journal of Control, 10(4), S. 393–404, 2008.
- [6] D. Weidemann und R. Diekmann. *Optimal Control Design for DES Using Supervisory Control Theory with Enforceable Events*. Proc. of 17th MMAR, S. 649–654, 2012.
- [7] K. Åkesson, M. Fabian, H. Flordal und R. Malik. *Supremica – An integrated environment for verification, synthesis and simulation of discrete event systems*. Proc. of 8th WODES, S. 384–385, 2006.
- [8] D. Weidemann und R. Diekmann. *Model-based Synthesis of Optimal Controllers Using Supervisory Control Theory with Enforceable Events*. In: Methods and Applications in Automation, Herausgeber: A. Gräser und D. Ristić-Durrant, Shaker, Germany, S. 110–121, 2012.





# Anmerkungen zur Modellierung hybrider dynamischer Systeme

Joachim Haase<sup>1</sup>, Ewald Hessel<sup>2</sup>

<sup>1</sup> Fraunhofer-Institut für Integrierte Schaltungen/Institutsteil Entwurfsautomatisierung Dresden

<sup>2</sup> Hella KGaA Hueck & Co. Lippstadt

*joachim.haase@eas.iis.fraunhofer.de, ewald.hessel@hella.com*

Hybride dynamische Systeme kombinieren zeitkontinuierliches und zeitdiskretes Verhalten. Sie stellen ein adäquates Modellierungskonzept für gemischt analog-digitale Systeme dar, die mit Verhaltenbeschreibungssprachen wie VHDL-AMS, Verilog-AMS und auch Modelica beschrieben werden können. Im Beitrag werden unter anderem numerische Aspekte, die bei der Modellerstellung in diesem Zusammenhang zu beachten sind, diskutiert.

## 1 Einleitung und Problembeschreibung

Hybride dynamische Systeme sind in den 1990er Jahren als Modellierungskonzept zur Beschreibung von Systemen mit zeitkontinuierlichem und zeitdiskretem Verhalten ausführlich untersucht worden. Ausgangspunkt ist dabei ein Zustandsautomat. Durch die einzelnen Zustände (Moden) wird ein unterschiedliches zeitkontinuierliches Verhalten (flow) festgelegt. Außerdem wird beschrieben, unter welchen Bedingungen ein Übergang von einer Mode in eine andere erfolgt (guards) und wie Anfangswerte für die Fortsetzung des kontinuierlichen Verhaltens in der neuen Mode zu bestimmen sind (set action).

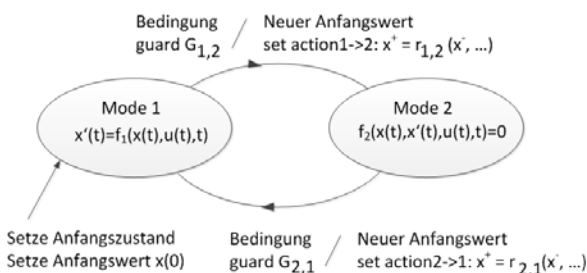


Abbildung 1. Hybrides dynamisches System

Abbildung 1 zeigt ein Beispiel für ein solches System. Eigenschaften spezieller derartiger Systeme sind unter dem Gesichtspunkt einer formalen Analyse ausführlich theoretisch untersucht worden. Im Zuge der Beschäftigung mit Untersuchungen zu cyber-physischen Systemen (cyber-physical systems) hat dieses Konzept erneut an Interesse gewonnen [1], stellt es doch einen systematischen Zugang zur Beschreibung gemischt analog-digitaler Systeme – auch für die Modellierung mit Verhaltensbeschreibungssprachen wie VHDL-AMS, Verilog-AMS und Modelica – dar.

## 2 Formale Beschreibung

Auf die Wiedergabe einer strengen mathematischen Definition hybrider dynamischer Systeme soll an dieser Stelle verzichtet werden. Die für die Modellbildung im Folgenden benötigten wesentlichen Aspekte sollen nur kurz referiert und diskutiert werden. Die Darlegungen stützen sich auf [2, 3].

Ein hybrides dynamisches System ist durch eine endliche Menge  $Q$  diskreter Moden (auch als diskrete Zustände bezeichnet) beschrieben. Jeder Mode  $i \in Q$  kann ein Definitionsbereich  $D_i \subseteq \mathbb{R}^{n_i}$  zugeordnet werden. Die ganzzahligen Werte  $n_i < \infty$  können für die einzelnen Moden prinzipiell unterschiedliche Werte annehmen. Das zeitkontinuierliche Verhalten in einer Mode wird allgemein durch eine differentielle Inklusion  $F$  (flow map) mit  $x'(t) \in F_i(x(t), t)$  mit  $x(t) \in D_i$  beschrieben. Dargestellt werden kann eine derartige Inklusion beispielsweise durch ein explizites oder implizites Differentialgleichungssystem. Die bei den  $F_i$  zu berücksichtigende Abhängigkeit von der Zeit  $t$  ergibt sich z.B. als Folge von zeitabhängigen Eingangsgrößen  $u: \mathbb{R}^+ \cup \{0\} \rightarrow U \subseteq \mathbb{R}^{n_u}$ . Der Ausgang  $y(t) \in Y \subseteq \mathbb{R}^{n_y}$  des hybriden dynamischen Systems wird durch  $u$  und den Verlauf  $x$  der aktuellen Mode festgelegt.

Mögliche Übergänge von einer Mode in eine andere können durch einen Graphen erfaßt werden, dessen Knoten die Menge  $Q$  der Moden bildet. Die Übergänge sind gerichtete Kanten in diesem Graphen. Für eine Kante (edge)  $e$  sei  $s(e)$  die Ausgangsmode und  $z(e)$  die Zielmode (target). Ein durch eine Kante  $e$  festgelegter Übergang wird aktiviert, wenn eine Bedingung (guard)  $G_{s(e),z(e)}(x, u, t)$  erfüllt ist. Ferner wird durch den Übergang zur Zeit  $t = t_a$  bestimmt

wie aus dem Wert  $x(t_a)$  in  $D_{s(e)}$  bei Eintreten einer Bedingung zur Kantenaktivierung ein Wert  $x^+(t_a)$  in  $D_{z(e)}$  abgeleitet wird (set action).

In der Initialisierungsphase zur Zeit  $t = 0$  und beim Übergang von einer Mode in eine andere zur Zeit  $t_a$  muss gesichert sein, dass es sich beim Startwert  $x(0)$  bzw. bei  $x^+(t_a)$  um konsistente Anfangswerte [4] für den weiteren Fluss handelt. Für explizite Differentialgleichungssysteme der Form  $x'(t) = f(x(t), u(t), t)$  ist die Bestimmung konsistenter Anfangswerte in der Regel evident. Schwieriger ist sie bei impliziten Differentialgleichungssystemen der Form  $f(x(t), x'(t), u(t), t) = 0$  oft dann, wenn die Komponenten von  $x$ , deren Ableitungen zur Beschreibung benötigt werden, keine Zustandsgrößen zur Charakterisierung des Flusses sind. Die Bestimmung konsistenter Anfangswerte ist selbst bei linearen Systemen der Form  $A \cdot x'(t) + B \cdot x(t) = C \cdot u(t)$  mit singulärer Koeffizientenmatrix  $A$  u.U. aufwendig [5]. So ist es nicht weiter verwunderlich, dass im Mittelpunkt theoretisch orientierte Publikationen zu hybriden dynamischen Systemen Flüsse stehen, die mit expliziten Differentialgleichungen beschrieben werden. Bei der Modellerstellung ist aber auch sonst sicherzustellen, dass beim Übergang von einer Mode in eine andere konsistente Anfangswerte für die weitere Simulation zur Verfügung gestellt werden [6].

Schwierig kann sich auch die Bestimmung von Anfangswerten für die weitere Simulation gestalten, wenn beim Übergang von einer Mode  $i$  zu einer Mode  $j$  mit den zugehörigen Definitionsbereichen  $D_i \subseteq R^{n_i}$  bzw.  $D_j \subseteq R^{n_j}$  die Dimensionen  $n_i$  und  $n_j$  verschieden sind. Simulationsprogramme, die mit Verhaltensbeschreibungssprachen wie VHDL-AMS und Verilog-AMS arbeiten, unterstützen diesen Fall nicht. Die Anzahl der Unbekannten des zu lösenden Differentialgleichungssystems in den unterschiedlichen Modi ändert sich nicht. Eine Änderung der Anzahl der Unbekannten und damit der Dimension des Differentialgleichungssystems kann beispielsweise sinnvoll sein, wenn eine Änderung der Modellgenauigkeit oder der Modellstruktur während einer Simulation erforderlich ist [7]. Die Vorgabe neuer Anfangswerte beim Übergang von einer Mode in eine andere ist auch hier ein durch den Modellersteller zu lösendes Problem.

Damit das Verhalten eines hybriden dynamischen Systems determiniert ist, dürfen die Bedingungen, die

zur Aktivierung eines Übergangs in eine andere Mode führen für verschiedene von der aktuellen Mode wegführende Kanten nicht gleichzeitig erfüllt sein [3].

Die Bedingungen zum Übergang zwischen den Moden dürfen unter Berücksichtigung neu zu setzender Anfangswerte in der neuen Mode nicht dazu führen, dass in dem Graph, dessen Knoten die Moden bilden, Schleifen existieren, in denen für bestimmte  $x$  Werte alle Bedingungen, die den Kanten einer solchen Schleife zugeordnet sind, erfüllt sind.

Ferner müssen Bedingungen angegeben werden, wie die Anfangswerte  $x(0)$  für die einzelnen Moden zur Anfangszeit  $t = 0$  zu bestimmen sind. Die Mode zur Anfangszeit kann – soweit möglich – vorgegeben werden oder ergibt sich auf Grund des Wertes  $u(0)$  des Eingangssignals.

### 3 Beispiele

Zahlreiche Modellbeschreibungen lassen sich unter Verwendung des Konzepts hybrider dynamischer Systeme erstellen. Beispiele dafür sind Ladungspumpen (charge pumps), Anschläge, Lose, Integratoren mit Begrenzung u.a. Einige Beispiele sollen kurz skizziert werden.

#### Modell für idealen Generator

Für Systemuntersuchungen elektrischer Bordnetze werden Modelle für geregelte Generatoren benötigt. Für diesen Zweck existieren eine Reihe ausgefeilter Modelle. Für erste Untersuchungen stehen aber häufig nur Kennlinien zur Verfügung, die den maximalen Generatorstrom in Abhängigkeit von Drehzahl und Temperatur beschreiben.

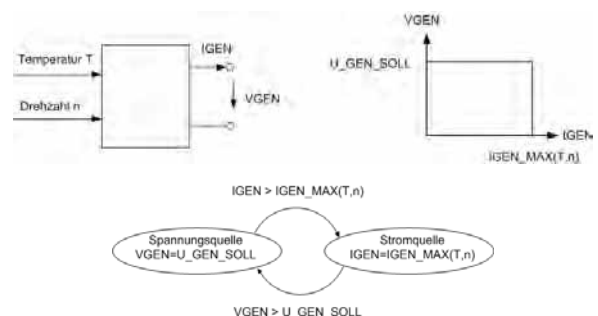


Abbildung 2. Modell für einen idealen Generator

In Abbildung 2 ist das Vorgehen skizziert. Das Bild zeigt das ideale Verhalten an den Klemmen des Generators und die Beschreibung durch einen hybriden (dynamischen) Automaten. Die Abhängigkeit des maximalen Generatorstroms von Temperatur und

Drehzahl  $\text{IGEN\_MAX}(T, n)$  ergibt sich dabei aus dem Generatorkennlinienfeld. Welche Mode sich zur Anfangszeit  $t=0$  einstellt, ergibt sich aus dem Netzwerk, in dem das Modell verwendet wird.

### Memristor

Dass das Konzept auch für neuere Entwicklungen wie beispielsweise bei der Modellerstellung für Memristoren auf Halbleiterbasis Vorteile bietet, soll im Folgenden kurz skizziert werden.

Ein Memristor ist das von L.O. Chua 1971 beschriebene fehlende Netzwerkelement, bei dem das Verhalten eines Zweipols durch eine Beziehung  $g(\varphi, q) = 0$  zwischen Magnetfluss  $\varphi$  und Ladung  $q$  beschrieben wird [8]. Für ein stromgesteuertes Memristor-Ein-Tor (mit einer Zustandsgröße  $x \in R$ ) wird das Verhalten von Klemmenspannung  $v$  und -strom  $i$  beschrieben durch [9]

$$\begin{aligned} x'(t) &= f(x(t), i(t), t) \\ v(t) &= R(x(t), i(t), t) \cdot i(t) \end{aligned}$$

$R$  ist dabei die sogenannte Memristanz. Das Interesse an diesem Konzept ist vor wenigen Jahren gewachsen als Memristoren mit einem Dünnschichtverbund realisiert werden konnten und sich damit ein Weg für neuartige integrierte Speicherstrukturen zu eröffnen schienen. Das mathematische Modell eines derartigen Memristors, das an dieser Stelle kurz diskutiert werden soll, führt auf eine Beschreibung der Form [10]

$$\begin{aligned} w'(t) &= \mu_V \cdot \frac{R_{ON}}{D} \cdot i(t) \\ v(t) &= \left( R_{ON} \cdot \frac{w(t)}{D} + R_{OFF} \cdot \left( 1 - \frac{w(t)}{D} \right) \right) \cdot i(t) \end{aligned}$$

Die Gesamtbreite der aktiv zu beeinflussenden Schicht ist  $D$ .  $w$  kann Werte zwischen 0 und  $D$  annehmen, aber keine Werte außerhalb des Intervalls  $[0, D]$ . Damit ergeben sich für die Memristanz in Abhängigkeit von  $w$  Werte im Intervall  $[R_{ON}, R_{OFF}]$ .  $\mu_V$  charakterisiert die Ladungsträgermobilität in der Schicht der Breite  $D$ . Die erste der beiden Gleichungen beschreibt eine lineare Drift der Ladungsträger.

Um sicherzustellen, dass die Werte von  $w$  das Intervall  $[0, D]$  nicht verlassen, werden verschiedenste Fensterfunktionen vorgeschlagen, mit denen die erste der beiden obigen Gleichungen multipliziert wird. Durch derartige Fensterfunktionen soll erreicht werden, dass  $w'(0) = w'(D) = 0$  bei  $i(t) < 0$  bzw.  $i(t) > 0$  gesichert wird und dass bei Strömen

$i(t) > 0$  und  $i(t) < 0$  die  $w$ -Werte 0 bzw.  $D$  wieder verlassen werden können. Diese Fensterfunktionen dienen darüberhinaus einer nichtlinearen Beschreibung des Driftverhaltens [11].

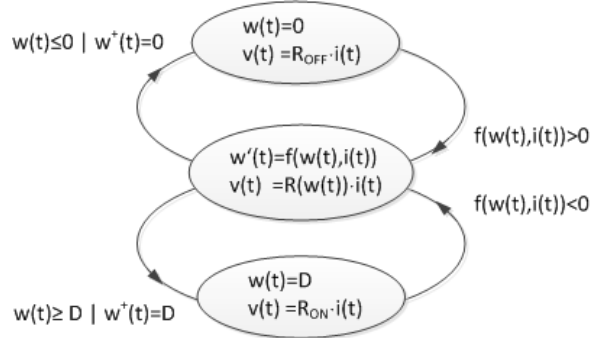


Abbildung 3. Memristormodell mit  $w(t) \in [0, D]$

Abbildung 3 zeigt für ein stromgesteuertes Memristor-Ein-Tor mit einer Dünnschichtanordnung ein hybrides dynamisches Modell, das die Einhaltung der Bedingung  $w \in [0, D]$  sichert, wobei keine besonderen Anforderungen an die Fensterfunktion gestellt werden. Aus der Vorgabe des Anfangswertes  $w(0)$  ergibt sich die Mode zur Zeit  $t = 0$ .

### Springender Ball

Das folgende Differentialgleichungssystem liefert eine (sehr) einfache Beschreibung eines springenden Balles:

$$\begin{aligned} s'(t) &= v(t) \\ v'(t) &= -g \end{aligned}$$

$s$  ist die Höhe des Balles über dem Boden,  $v$  die Fallgeschwindigkeit und  $g$  die Gravitationskonstante.

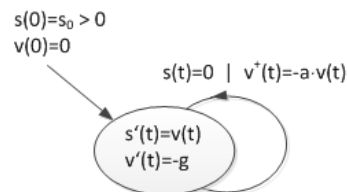


Abbildung 4. Springender Ball

Wenn der Ball fällt und auf den Boden zur Zeit  $t_b$  ( $s(t_b) = 0$ ) mit der Geschwindigkeit  $v(t_b)$  stößt (bump event), springt der Ball mit der Geschwindigkeit  $v^+(t_b) = -a \cdot v(t_b)$  mit  $0 < a < 1$  zurück. Das heißt, der Stoß ist unelastisch. Die Anfangshöhe des Balles sei  $s(0) = s_0 > 0$  und die Anfangsgeschwin-

digkeit  $v(0) = 0$ . Abbildung 4 zeigt das auf dieser Beschreibung basierende hybride dynamische System.

Der Anfangszustand ist bei diesem Beispiel eindeutig charakterisiert. Der springende Ball verdeutlicht ein mögliches Problem, das bei der Beschreibung mit hybriden dynamischen Systemen auftreten kann.

Es kann leicht gezeigt werden (siehe z.B. [12]), dass der Ball zum ersten Mal nach der Zeit  $t_1 = \sqrt{\frac{2 \cdot s_0}{g}}$  auf den Boden auftrifft. Die Zeit zwischen dem zweiten und ersten Auftreffen ist  $2a \sqrt{\frac{2 \cdot s_0}{g}}$  und zwischen dem dritten und zweiten ist  $2a^2 \sqrt{\frac{2 \cdot s_0}{g}}$ . Die Zeiten zwischen dem Auftreffen des Balles werden immer kleiner. Zeitpunkt des  $k$ -ten Auftreffens ist  $t_k = 2 \cdot \sqrt{\frac{2 \cdot s_0}{g}} \cdot \left(\frac{1}{2} + a \cdot \frac{1-a^{k-1}}{1-a}\right)$ . Bis zur Zeit  $t_\infty = \frac{1+a}{1-a} \cdot \sqrt{\frac{2 \cdot s_0}{g}}$  beschreibt das hybride dynamische System aus Abbildung 4 das Verhalten. Die Zeit  $t_\infty$  wird Zeno-Zeit genannt. Charakteristisch ist in diesem Fall, dass vor der Zeno-Zeit die Differenzen zwischen den Zustandswechseln immer kleiner werden, unendlich viele Zustandswechsel stattfinden und die Zeno-Zeit endlich ist. Damit ist das Verhalten des Systems nach dieser Zeit formal nicht beschrieben. Im Allgemeinen wird dieses Problem durch Einführung einer Post-Zeno-Mode gelöst, mit der das Verhalten nach Erreichen des Zeno-Punktes beschrieben wird (siehe [13, 14]). Das heißt, Abbildung 4 beschreibt das Verhalten für die Simulation nicht vollständig.

#### 4 Numerische Aspekte

Für die Simulation hybrider dynamischer Systeme existiert eine Reihe von Simulationswerkzeugen. Goebel u.a. [15] verweisen auf Matlab/Simulink, Modelica-basierte Simulatoren, Ptolemy u.a. Zu ergänzen sind in dieser Zusammensetzung Simulatoren für gemischt analog-digitale Systeme (AMS systems – analog mixed signal systems), die VHDL-AMS- und/oder Verilog-AMS [16, 17] unterstützen. Unterstützt werden muss auf Grund der in Abschnitt 2 beschriebenen Merkmale hybrider dynamischer Systeme

- Beschreibung des Verhaltens in den einzelnen Moden unter Verwendung von expliziten Differentialgleichungssystemen oder differential-algebraischen Gleichungssystemen
- Erkennen von Bedingungen (guards), die zum Übergang von einer Mode in eine andere führen
- Vorgabe neuer Anfangswerte nach dem Übergang von einer Mode in eine andere
- Zugriff auf die aktuelle Mode, in der sich ein Modell bei der Simulation befindet.

Unter praktischem Gesichtspunkt müssen die Modelle robust sein. Iterationsschleifen zwischen unterschiedlichen Moden in der Initialisierungsphase und bei der Simulation im Zeitbereich, immer kürzer werdende Umschaltzeiten zwischen den einzelnen Zuständen, z.B. Zeno-Effekt beim fallenden Ball, und das Verlassen des Definitionsbereiches für den Fluss in einer Mode müssen vermieden werden. Außerdem muss das Eintreten der Bedingungen für den Übergang in eine andere Mode (guard) sicher erkannt werden. Es ist sicherzustellen, dass die in Abschnitt 2 skizzierten Anforderungen für hybride dynamische Bedingungen erfüllt sind. Dabei ist die begrenzte numerische Genauigkeit bei der Ermittlung der Lösungen für differential-algebraische Gleichungssysteme in Folge der Diskretisierung zu berücksichtigen. Bedingungen, die auf die Gleichheit mit Null testen, sind nicht zu realisieren. Sie müssen durch die Abfrage, ob Werte größer oder kleiner Null (oder ein entsprechend kleiner  $\varepsilon$ -Wert) sind, in geeigneter Weise ersetzt werden.

Besonders Simulatoren, die gemischt analog-digitale Verhaltensbeschreibungen unterstützen, sind für die Simulation hybrider dynamischer Systeme geeignet [18]. Die Moden, in der sich Modelle befinden, werden unter Verwendung zeitdiskreter Signale gespeichert. Den Fluss in einer Mode beschreiben zeitkontinuierliche Verläufe. Zur Aktivierung der Bedingungen (guards) zum Übergang von einer Mode in eine andere werden zu diskreten Zeitpunkten Ereignisse generiert. Diese können aus diskreten Signalen oder dem Über- oder Unterschreiten von Schwellwerten für zeitkontinuierliche Verläufe abgeleitet werden. VHDL-AMS stellt dafür das ABOVE-Attribut zur Verfügung, Verilog-AMS die cross() und above() Funktionen. Zum Setzen von Anfangswerten in der Initialisierungsphase und nach Übergang von einer Mode in eine andere verfügen Verhaltensbeschrei-

bungssprachen für analog-digitale Systeme über entsprechende Sprachkonstrukte wie z.B. die break-Anweisung in VHDL-AMS.

Werden die Bedingungen nur zu diskreten Zeitpunkten ausgewertet, können die Wechsel zwischen zwei Moden nicht unendlich nahe beieinander liegen. Der Zeno-Effekt wird – zumindest theoretisch – vermieden. Kritisch erweist sich bei der Erstellung von Modellen die sichere Aktivierung der Bedingungen, wenn diese aus dem Über- oder Unterschreiten von Schwellwerten abgeleitet werden. Folgende Maßnahmen erweisen sich als hilfreich, wenn in diesem Zusammenhang mit einem erstellten Modell Schwierigkeiten auftreten

- Wird in Bedingungen (für unterschiedliche Kanten des unter Verwendung der Moden gebildeten Graphen) sowohl das Über- als auch das Unterschreiten eines Schwellwertes abgefragt, sollte diese Abfrage durch Kontrolle des Über- oder Unterschreitens des Schwellwertes + oder – einem kleinen Wert  $\varepsilon$  ersetzt werden.
- Abfrage der Bedingungen in vorgegebenen Zeitabständen. Das Modell wird dadurch in der Regel langsamer, aber auch weniger störanfällig.
- Einfügen eines Tests, ob der Definitionsbereich des Flusses für die aktuelle Mode verlassen wird, und Abfrage der Bedingungen an den Kanten, die vom Knoten mit der aktuellen Mode wegführen.
- Der Anfangswert, der beim Übergang von einer Mode in eine andere gesetzt wird, sollte im Innern und nicht auf dem Rand des neuen Definitionsbereiches liegen.

Im Übrigen sind die in Abschnitt 2 skizzierten Bedingungen einzuhalten.

## 5 Modellaustausch

In [19] wird eine allgemeine Modellschnittstelle - Functional Mock-up Interface for Model Exchange (FMI) - beschrieben. Diese Modellschnittstelle ist vordergründig für den Austausch von Modellen zwischen Modelica-basierten Simulatoren entwickelt worden. Sie eröffnet aber auch eine Möglichkeit unter Verwendung von Modelica entwickelte Modelle in andere Simulationsumgebungen zu integrieren. Durch die Schnittstelle werden C-Funktionen und deren

Funktionalität zur Beschreibung des Klemmenverhaltens eines Modells definiert. Das FMI for Model Exchange definiert Funktionen zur Beschreibung spezieller dynamischer Systeme. Die Flüsse in den einzelnen Moden werden unter Verwendung expliziter Differentialgleichungssysteme beschrieben (Funktion `fmiGetDerivatives`). Für die Auswertung der Bedingungen (guards) zum Übergang von einer Mode in eine andere gibt es die Funktion `fmiGetEventIndicators`. Diese Funktion wertet Nulldurchgänge spezieller zeitkontinuierlicher Verläufe (event indicators) und im Voraus ermittelte Ereigniszeitpunkte  $T_{next}$  aus. Nach Auftreten eines Ereignisses werden ggf. eine neue Mode und neue Anfangswerte für die Ermittlung des Flusses in der neuen Mode mit der Funktion `fmiEventUpdate` bestimmt. Für die Bestimmung von Anfangswerten in der Initialisierungsphase gibt es die Funktion `fmiInitializeModel`. Funktionen stehen zum Lesen der Eingangsgrößen  $u$  des Modells und zum Bestimmen der Ausgangsgrößen  $y$  zur Verfügung. Unter Verwendung dieser Grundfunktionen können hybride dynamische Systeme in anderen Beschreibungssprachen modelliert werden. VHDL-AMS-basierte Simulatoren müssen dazu die Verwendung reellwertiger quantity-Vektoren in „simultaneous statements“, das ABOVE-Attribut zum Erkennen der Nulldurchgänge der „event indicators“ und das Setzen von Anfangswerten in der Initialisierungsphase und nach dem Übergang von einer Mode in eine andere mit der break-Anweisung unterstützen. Nicht zuletzt muss eine (simulatorspezifische) C-Schnittstelle für den Funktionsaufruf unterstützt werden. Es wurden eine Reihe kleinerer Beispiele umgesetzt, mit denen gezeigt werden konnte, dass dieses Vorgehen möglich ist.

Da sich aber beispielsweise der VHDL-AMS-Simulationszyklus von der durch das FMI-Dokument beschriebenen Aufrufreihenfolge unterscheidet, ist in vielen Fällen, in denen nur Interesse an einer Zeitbereichssimulation besteht, einem anderen Weg der Vorzug zu geben. Bei einem FMI-basierten Modell ist gesichert, dass die Flussgrößen, die in den einzelnen Moden ermittelt werden, nicht in ein gemeinsames Gleichungssystem eines zu simulierenden Gesamtsystems einfließen. Das Modell wird mit einem Löser zur Auswertung der Flüsse in den Moden verknüpft. Eine übersetzte Version des Modells kann dann für unterschiedliche Rechnerplattformen zur Verfügung gestellt werden. Für den Simulator, in den das Modell eingebunden wird, muss ein C-Interface existieren.

## 6 Literatur

- [1] Lee, E.A.; Seshia, S.A. *Introduction to Embedded Systems*. LeeSeshia.org, 2011.
- [2] Goebel, R.; Sanfelice, R. G.; Teel, A. R.: *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, 2012.
- [3] Abate, A.; D’Innocenzo, A.; Di Benedetto, M.D.; Sastry, S.: Understanding Deadlock and Livelock Behaviors in Hybrid Control Systems. In: *Nonlinear Analysis: Hybrid Systems*, vol. 3, vssue 2, May 2009, pp. 150-162.
- [4] Leimkuhler, B.; Petzold, L.R.; Gear, C.W.: Approximation Methods for the Consistent Initialization of Differential-Algebraic Equations. In: *SIAM J. Numer. Anal.*, vol. 28, issue 1, pp. 205-226, 1991.
- [5] Reißig, G.; Boche, H.; Barton, P.I.: On Inconsistent Initial Condidions for Linear Time-Invariant Differential-Algebraic Equations. In: *IEEE Trans. Circuits Syst. I*, vol. 49, issue 11, pp. 1646-1648, November 2002.
- [6] Haase, J.: Rules for Analog and Mixed-Signal VHDL-AMS Modeling. In: Grimm, C. (Hrsg): *Languages for System Specification*. Springer, 2004.
- [7] Bastian, J.; Clauß, C.; Enge-Rosenblatt, O.; Schneider, P.: MOSILAB – a Modelica solver for multiphysics problems with structural variability. In: *Proc. 1<sup>st</sup> Conference on Multiphysics Simulation – Advanced Methods for Industrial Engineering*. Bonn, 12 S., June 22-23, 2010.
- [8] Chua, L.O.: Memristor – the Missing Circuit Element. In: *IEEE Trans. Circuit Theory*, vol. 18, issue 5, pp. 507-519, September 1971.
- [9] Chua, L.O.; Kang, S.M.: Memristive Devices and Systems. In: *Proc. of the IEEE*, vol. 64, issue 2, pp. 209-223, February 1976.
- [10] Strukov, D.V.; Snider, G.S.; Stewart, D.R.; Williams, R.S.: The Missing Memristor Found. In: *Nature*, vol. 453, pp. 80-83, May 2008.
- [11] Kvatinisky, S.; Friedman, E.G.; Kolodny, A.; Weiser, U.C.: TEAM: ThrEshold Adaptive Memristor Model. In: *IEEE Trans. Circuits Syst. I*, vol. 60, issue 1, pp. 211-221, January 2013.
- [12] Oltean, V.E.: On Simulation of Zeno Hybrid Systems. In: *Rev. Roum. Sci. Techn. – Électrotechn. et Énerg.*, vol. 52, issue 2, pp. 229-239, 2007. Online: <http://revue.elth.pub.ro/upload/285662art08.pdf>
- [13] Zhang, F.; Yeddanapudi, M.; Mosterman, P.J.: Zero-Crossing Location and Detection Algorithms For Hybrid System Simulation. In: *Proc. of the 17th IFAC World Congress*, 2008. DOI: 10.3182/20080706-5-KR-1001.01346
- [14] Zheng, H.: *Simulating Zeno Hybrid Systems Beyond Their Zeno Points*. Technical Report No. UCB/EECS-2006-114, September 8, 2006.
- [15] Goebel, R.; Sanfelice, R.G.; Teel, A.: Hybrid dynamical systems. In: *IEEE Trans. Control Systems*, vol. 29, issue 2, pp. 28-93, April 2009. DOI: 10.1109/MCS.2008.931718
- [16] *Behavioral languages-Part 6: VHDL Analog and Mixed Signal Extensions: IEEE Std 1076.1 IEC 61691-6 Edition 1.0 2009-12*. DOI: 10.1109/IEEESTD. 2009.5464492.
- [17] *Verilog-AMS Language Reference Manual: Version 2.3.1*. Accelera, June 2009.
- [18] Nutaro, J.; Kuruganti, P.T.; Protopopescu, V.; Shankar, M.: The Split System Approach to Managing Time in Simulations of Hybrid Systems having Continuous and Discrete Event Components. In: *Simulation*, vol. 88, issue 3, pp. 281-298, 2012.
- [19] *FMI for Model Exchange and Co-Simulation 2.0* URL: <https://www.fmi-standard.org/downloads>

# Hochflexibles HiL-System zur modellbasierten Entwicklung vernetzter Regelsysteme für Elektrofahrzeuge

Florian Quantmeyer, Waldemar Diehl, Xiaobo Liu-Henke

Ostfalia Hochschule für angewandte Wissenschaften

*Flo.Quantmeyer@ostfalia.de*

Dieser Beitrag stellt ein flexibles, integratives HiL-System für Elektrofahrzeug vor, das die Entwicklung eines integrierten Fahrzeugmanagements für ein Elektrofahrzeug mit dezentralen elektrischen Antrieben wesentlich unterstützt. Das hierarchisch angeordnete HiL-System besteht aus zwei unterlagerten HiL-Prüfständen: einem Prüfstand mit elektrischem Antriebsstrang und einem Prüfstand mit Traktionsbatterie des Elektrofahrzeugs. Die beiden unterlagerten HiL-Systeme können sowohl separat zum Einsatz kommen als auch zu einem HiL-Gesamtsystem kombiniert werden. Somit kann das gesamte HiL-System für unterschiedliche Untersuchungszwecke in verschiedenen Varianten hochflexibel konfiguriert und für die modellbasierte Entwicklung angewendet werden.

## 1 Einleitung

Motiviert durch Politik und Gesellschaft setzt die Automobilindustrie auf eine zunehmende Elektrifizierung des Antriebsstrangs von Kraftfahrzeugen. Gleichzeitig steigt die Anzahl der Softwarefunktionen besonders im Bereich Fahrdynamikregelsysteme und Fahrerassistenz stark an. Das Zusammenwirken dieser Technologien in zukünftigen Fahrzeugen führt zu hochgradig komplexen Systemen mit hohem Vernetzungsgrad. Im Rahmen des BMBF-Projekts „EVM - Elektronisches Fahrzeugmanagement für Elektrofahrzeuge mit dezentralen Direktantrieben“ forscht die Ostfalia Hochschule gemeinsam mit einem Konsortium aus Industrie und Wissenschaft an einem integrierten Fahrzeugmanagement.

Zur Beherrschung der Systemkomplexität werden im Entwicklungsprozess die modellbasierten Methoden von Model-in-the-Loop (MiL)- über Software-in-the-Loop (SiL)- bis hin zu Hardware-in-the-Loop (HiL)-Simulation (vgl. Abbildung 1) konsequent angewendet und weiterentwickelt. Durch ihre Leistungsfähigkeit und ihren verifikationsorientierten Charakter können sich diese Methoden zunehmend durchsetzen und sind bereits heute in den Absicherungsprozessen einiger OEMs fest verankert.

In der MiL-Ebene erfolgt die Entwicklung und Absicherung der Regelalgorithmen und Softwarefunktionen gegen ein Modell der Strecke. Dabei wird die Funktion angepasst bis die Anforderungen erfüllt sind. Die Modellbildung der Streckenkomponenten wird in dieser Ebene durch Messungen am realen System sowie Identifikation und Validierung unter-

stützt. Sind die Anforderungen erfüllt geht man in die SiL-Ebene über. Aus den Regelalgorithmen wird Softwarecode generiert. Dies kann entweder manuell oder durch die Anwendung s.g. Autocodegeneratoren geschehen. Der Softwarecode wird wiederum gegen ein Streckenmodell getestet und optimiert bis er die Anforderungen vollständig abdeckt. In der HiL-Ebene erfolgt die Übertragung des Targetcodes auf das reale Steuergerät oder sofern noch nicht verfügbar auf ein Steuergerät, das diesem sehr nahe kommt. Das Steuergerät wird an einen HiL-Simulator mit definierten Schnittstellen gekoppelt und unter Echtzeitbedingungen gegen das Streckenmodell getestet. Dies erfordert wiederum die Echtzeitfähigkeit des Modells, sodass häufig eine Reduzierung der Modelltiefe erforderlich ist.

In allen Ebenen erfolgt eine Unterstützung dieses Prozesses durch Messungen und weitergehende Validierung am realen System. Die Messungen verfolgen dabei unterschiedliche Ziele. Bereits bei der Modellentwicklung müssen reproduzierbare Messungen zur Analyse des Realsystems sowie für Identifikations- und Validierungszwecke erfolgen. Weiterhin sollte bereits in frühen Phasen der Entwicklung die Möglichkeit bestehen die Softwarefunktionen gegen die Reale Strecke zu testen. Dazu werden die Softwarefunktionen auf die Echtzeithardware geladen, sodass diese als Steuergerät fungiert.

Zur Unterstützung des modellbasierten Entwicklungsprozesses von Batteriemanagement, Antriebsregelsystemen und Fahrdynamikregelungen wird in diesem Beitrag ein flexibles HiL-System vorgestellt.

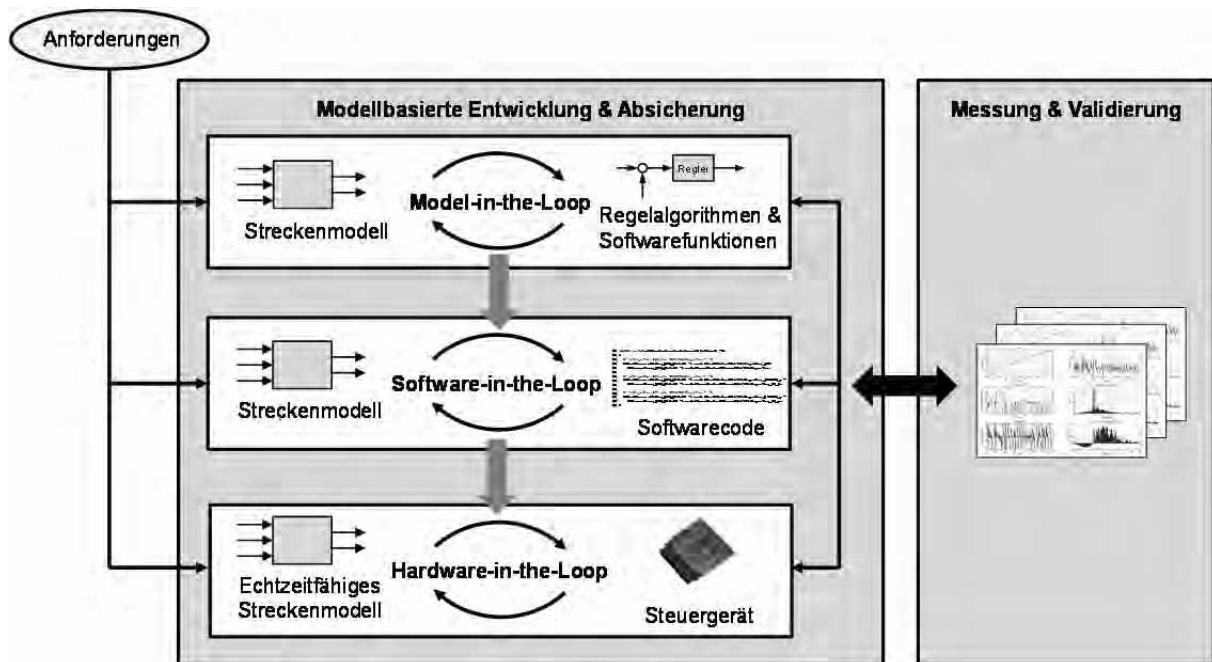


Abbildung 1. Modellbasierter Entwicklungsprozess

## 2 Konzept

Die Entwicklung von Regelsystemen für Elektrofahrzeuge erfordert die Lösung zahlreicher Problemstellungen. Diese beginnen bereits bei der Modellierung der Antriebsstrangkomponenten, wie Traktionsbatterie, Leistungselektronik und E-Maschine. Nach der Auslegung von Regelalgorithmen mittels MiL und SiL erfolgt eine weitergehende Validierung unter Echtzeitbedingungen mittels HiL. Dies kann sowohl bedeuten, dass das Steuergerät an die Echtzeithardware angekoppelt und gegen ein echtzeitfähiges Streckenmodell getestet wird, als auch die Implementierung der Regelalgorithmen auf der Echtzeithardware unter Anbindung von Realkomponenten der Regelstrecke (Rapid Control Prototyping).

Die Aufgabenstellung umfasst dabei sowohl die Auslegung von Batterie-, Antriebs- und Fahrdynamikregelung, als auch die Vernetzung dieser Systeme. Dementsprechend muss das geplante HiL-System in der Lage sein, die Entwicklung dieser Systeme in vollem Umfang zu unterstützen.

Aus diesen Anforderungen wurde hierarchisch angeordnetes HiL-System konzipiert, das unterlagerten eigenständigen HiL-Systemen besteht. Die unterlagerten HiL-Systeme verfügen über eine eigene Echt-

zeithardware sowie Benutzerschnittstelle und sind separat einsetzbar. Über definierte Schnittstellen kann die Kopplung zu einem Gesamtsystem erfolgen. Folglich kann das Prüfstandssystem in folgenden Konfigurationen eingesetzt werden:

- HiL-Prüfstand „Traktionsbatterie“ (Batterie und Batteriesystem)
- HiL-Prüfstand „Antrieb / Längsdynamik“ – (Leistungselektronik, E-Maschine, hydraulische Bremse, Rad, Reifen-Straße-Kontakt)
- HiL-System „E-Traktion“ (Energierückgewinnung)

Somit kann eine Unterstützung bei einer hohen Bandbreite von Entwicklungsaufgaben erfolgen. Die Realisierung dieses Konzepts wird in den folgenden Kapiteln näher erläutert.

## 3 Prüfstand „Traktionsbatterie“

Der Prüfstand lässt sich gemäß Abbildung 2 in folgende Teilsysteme unterscheiden:

- Strecke (Batterie oder einzelne Batteriezeilen)



- Anregungseinheit bestehend aus einem regelbaren Zwei-Quadranten-Modul
- Sensorik zur Erfassung der Messgrößen
- Hardware zur Informationsverarbeitung

Im Folgenden werden diese Teilsysteme näher beschrieben.

### 3.1 Strecke

Als Regelstrecke fungiert die Batteriezelle. Das Regelziel ist zum einen die Schätzung des Ladezustands und zum anderen dessen Regelung. Als Aktor dienen elektrische Bypässe, mithilfe derer Energie abgebaut oder umverteilt werden kann. Dies ermöglicht die Angleichung der Ladestände einzelner Zellen. Mithilfe verschiedener Sensoren werden die Messgrößen Temperatur, Strom und Spannung erfasst.

### 3.2 Anregungseinheit

Zur Anregung von Batterie oder Batteriezellen wird ein Zwei-Quadranten-Modul eingesetzt. Dies besteht aus einer geregelten Spannungsquelle und einer geregelten Stromsenke. Die Vorgabe der Sollwerte erfolgt durch die Informationsverarbeitung über Schnittstellen der Echtzeithardware. Die Betriebsarten Spannungs-, Strom- und Leistungsregelung lassen sich bei beiden Geräten während des Betriebs verstellen. Die Anregungseinheit deckt einen Spannungsbereich von 0-80 VDC ab und ermöglicht die Belastung des Prüflings mit bis zu 200 A.

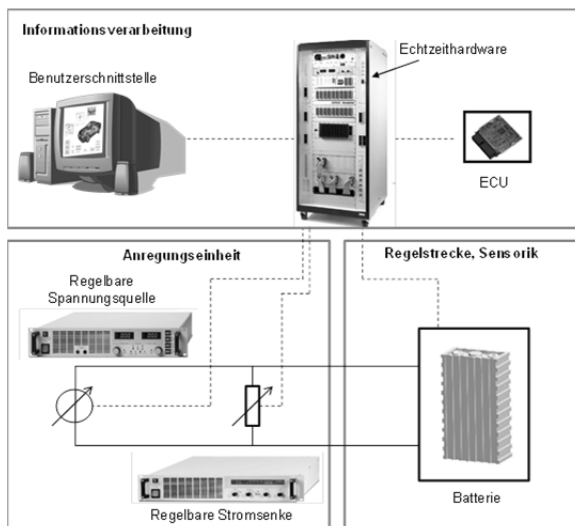


Abbildung 2. Aufbau HiL-Prüfstand Traktionsbatterie

### 3.3 Informationsverarbeitung

Für die Informationsverarbeitung ist ein leistungsfähiger Digital-Signal-Prozessor mit unterschiedlichen Peripherieboards zur Bereitstellung der erforderlichen Schnittstellen vorgesehen. Ein Softwaremodul übernimmt die digitale Signalaufbereitung und –skalierung sowie die Erzeugung von Stellsignalen für die Anregungseinheit. In diesem Modul sind weiterhin verschiedene Testprozeduren hinterlegt, die vom Benutzer abgerufen und automatisiert durchgeführt werden können. Es handelt sich dabei um Testzyklen für Charakterisierung und Identifikation von Zellen aber auch um skalierte Leistungszyklen, die aus Fahrprofilen abgeleitet wurden, wie beispielsweise der FUDS oder NEFZ-Zyklus.

Verschiedene Sicherheitsmechanismen überwachen den Zustand der Batteriezelle und verhindern das Verlassen des Betriebsbereichs.

### 3.4 Einsatzbereiche

Das Einsatzgebiet des Prüfstands „Traktionsbatterie“ umfasst die Analyse, Identifikation und Validierung einzelner Batteriezellen sowie verschalteter Batteriestacks. Im Rapid Control Prototyping Modus erlaubt der Prüfstand eine schnelle Erprobung von Regelungen und Softwarefunktionen durch Test gegen die reale Strecke. Dabei werden die Regelungen auf der Echtzeithardware des Prüfstands implementiert. Darüberhinaus verfügt der Prüfstand über Schnittstellen zur Anbindung des Batteriesteuergeräts. Gegen ein Simulationsmodell der Batterie können so unter Echtzeitbedingungen automatisierte Tests gefahren werden.

### 3.5 Anwendungsbeispiel

Am Beispiel der Entwicklung eines modellbasierten Ladezustandsschätzers wird im Folgenden der Einsatz des Batterieprüfstands näher erläutert. Als Batteriezelle wurde für diese Versuche eine Lithium-Eisen-Phosphat-Zelle mit einer Nennkapazität von 60 Ah verwendet.

Zunächst findet die automatisierte Vermessung der Batteriezelle statt. Parallel dazu wird ein Modell der Batteriezelle basierend auf einem elektrischen Ersatzschaltbild (vgl. Abbildung 3) erstellt und identifiziert (genauere Beschreibung der Vorgehensweise ist [1] und [2] zu entnehmen).

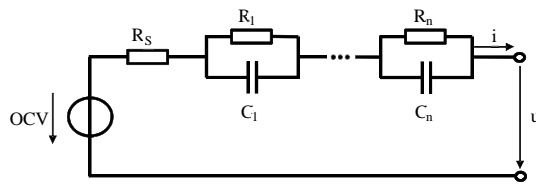


Abbildung 3. Batteriemodell

Basierend auf diesem Modell wird ein Extended Kalman Filter (EKF) zur Schätzung des Batterieladestands entwickelt. Anschließend wird das EKF mittels MiL- und SiL-Simulation verifiziert.

Das verifizierte EKF wird danach auf die Echtzeithardware des Prüfstands implementiert und gegen die reale Zelle getestet (vgl. Abbildung 4). Eine weitergehende Untersuchung anderer Einflüsse wie Messrauschen wird ermöglicht. Das Verfahren ermöglicht eine effiziente Durchführung und Realisierung von Optimierungen.

Als Testzyklus wurde der Federal Urban Driving Schedule (FUDS) gewählt, da dieser die Batterie mit schnell ändernden Strömen belastet. Damit wird vor allem die Genauigkeit des EKF bei dynamischer Belastung der Batterie ermittelt. Der Testzyklus startet bei voll geladener Batteriezelle und wird so oft wiederholt, bis die Zelle vollständig entladen ist.

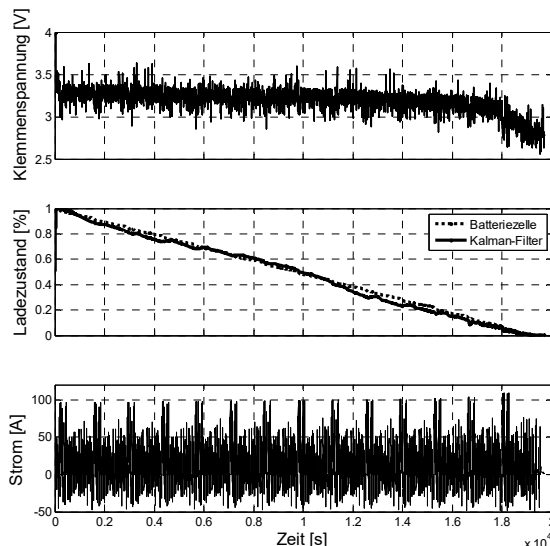


Abbildung 4. Validierung des Ladezustandsschätzers

Erfüllt das EKF die zuvor definierten Anforderungen wird es auf der Targethardware implementiert. Diese wird wiederum mit der Echtzeithardware gekoppelt um effiziente Tests durchzuführen. Dies ist besonders

hinsichtlich der Sicherheitsmechanismen relevant, da derartige Tests an der realen Zelle ein erhebliches Sicherheitsrisiko darstellen würden.

Am modellbasierten Entwicklungsprozess des Ladezustandsschätzers wird deutlich, dass der HiL-Prüfstand Traktionsbatterie jeden Entwicklungsschritt flexibel unterstützen kann.

#### 4 Prüfstand „Antrieb/Längsdynamik“

Für die Erprobung von Antriebs- und Schlupfregelungen wird der Traktionsprüfstand eingesetzt. Er bildet die Längsdynamik eines Viertelfahrzeugs mit elektrischem, dezentralem Direktantrieb realitätsnah ab.

Der Prüfstand lässt sich gemäß Abbildung 5 in folgende Teilsysteme unterscheiden:

- Regelstrecke inklusive Aktorik (Elektrischer Antriebsstrang, hydraulische Bremse und Rad)
- Anregungseinheit, bestehend aus einer angetriebenen Rolle mit definiertem Trägheitsmoment auf der das Rad abrollt
- Energieversorgungseinheit
- Sensorik zur Erfassung der Messgrößen
- Hardware zur Informationsverarbeitung

Im Folgenden werden diese Komponenten näher beschrieben.

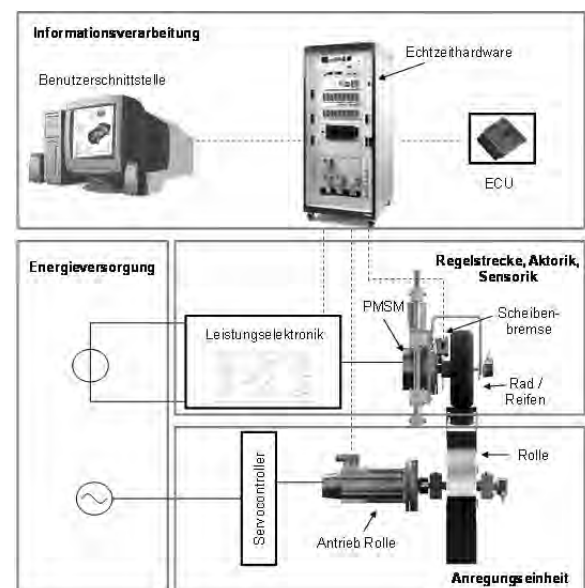


Abbildung 5. Traktionsprüfstand

#### 4.1 Elektrischer Antrieb

Die Permanentterregte Synchronmaschine (PMSM) wandelt die Energie in beide Richtungen. Beim Antreiben des Rades bzw. Fahrzeugs wird elektrische Energie in mechanische gewandelt. Beim Verzögern wird die überschüssige mechanische Energie in elektrische Energie gewandelt und zurückgespeist.

Das Bindeglied zwischen Regelungseinheit und E-Maschine und Energieversorgung stellt die Leistungselektronik dar. Sie erzeugt durch Umsetzung der Stellsignale des Reglers aus der verfügbaren Gleichspannung drei sinusförmige Strangspannungen für die E-Maschine.

Die am Prüfstand eingesetzte Leistungselektronik ist in Abbildung 6 dargestellt. Das Kernelement ist der IGBT-Wechselrichter der Fa. Infineon. Die darüberliegende Treiberschaltung erzeugt aus den Signalen des Controllers oder der Echtzeithardware die erforderlichen Spannungen zum Schalten der IGBTs und verfügt darüberhinaus über verschiedene Schutzfunktionen. Die Kühlung erfolgt mit einem flüssigen Medium.

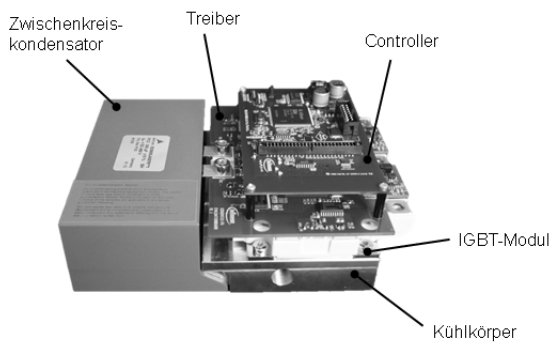


Abbildung 6. Leistungselektronik

Alternativ zur Regelung mittels Controllerboard kann eine Kopplung an die Echtzeithardware erfolgen. Dies ermöglicht eine effiziente Entwicklung der Regelalgorithmen durch direkte und reproduzierbare Erprobung am Realsystem.

#### 4.2 Rad und hydraulische Bremse

Die Kopplung von Rad und elektrischer Maschine erfolgt direkt. Im Gegensatz zur Lagerung im Fahrzeug ist hier neben dem rotatorischen Freiheitsgrad nur ein translatorischer Freiheitsgrad in Vertikalrichtung zugelassen. Durch Variieren der Federvorspannung kann eine manuelle Einstellung der Radaufstandskraft vorgenommen werden.

Neben der E-Maschine verfügt das Viertelfahrzeug über einen weiteren Aktor. Dabei handelt es sich um eine hydraulisch betätigte Scheibenbremse. Der Hydraulikkreislauf ist in Abbildung 7 dargestellt. Das Hydrauliksystem weist, wie bei herkömmlichen ABS-Bremssystemen üblich, eine Druckregelung auf. Das System wurde so ausgelegt, dass die Regelung eine 3dB-Bandbreite von 12 Hz aufweist. Der Soll-Druck wird testzyklenabhängig über die Echtzeithardware vorgegeben.

Im Bremsfall liegt ein überaktuiertes System vor. Eine Kooperation von Antriebsregelsystem und Bremsregelung ist erforderlich. Die modellbasierte Entwicklung entsprechender Funktionen wird mithilfe des Prüfstands massiv unterstützt.

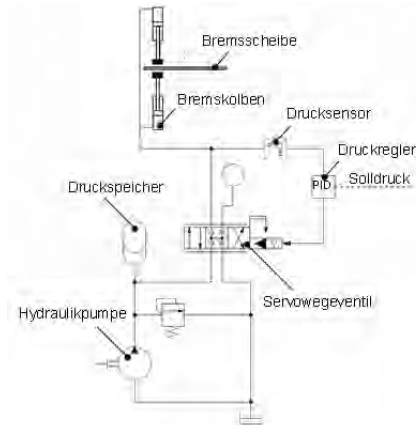


Abbildung 7. Hydraulisches Bremssystem

#### 4.3 Anregungseinheit

Um realitätsnahe Testbedingungen zu erreichen wird eine Anregungseinheit in den Prüfstand integriert, die die Nachbildung der Umgebungseinflüsse realisiert. Sie besteht aus einer Rolle deren Massenträgheit auf ein Viertel des Fahrzeugaufbaus abgestimmt ist. Ein zusätzliches Moment kann mittels Servoantrieb eingebracht werden.

Die Anregungseinheit ermöglicht somit die realistische Nachbildung aller Fahrwiderstände. Die Generierung des Drehmomentsollwerts für den Servoantrieb erfolgt modellgestützt. Dazu werden die Differentialgleichungen (1) und (2) herangezogen (vgl. Abbildung 8), die die Dynamik der Raddrehung und die Fahrzeuglängsbewegung beschreiben.

$$J \cdot \dot{\omega} = T_{EM} - T_B - (F_X + F_R) \cdot r_{dyn} \quad (1)$$

$$F_B = m \cdot \ddot{x} = F_X - F_{L,x} - F_{St} \quad (2)$$

Die Drehmomente der permanenten Synchronmaschine ( $T_{EM}$ ) und der hydraulischen Bremse ( $T_B$ ) rufen eine Änderung der Drehgeschwindigkeit des Rades mit der Massenträgheit  $J$  (inkl. alle rotierenden Elemente) hervor. Der Beschleunigung wirkt die Reifenkraft  $F_x$  entgegen. Sie ist maßgeblich aus den Eigenschaften des Reifens, der Radaufstandskraft und der Fahrbahnoberfläche bestimmt. Darüberhinaus entsteht durch die Reibung beim Abrollen des Reifens der Fahrwiderstand  $F_R$ , der der Drehrichtung grundsätzlich entgegengerichtet ist.

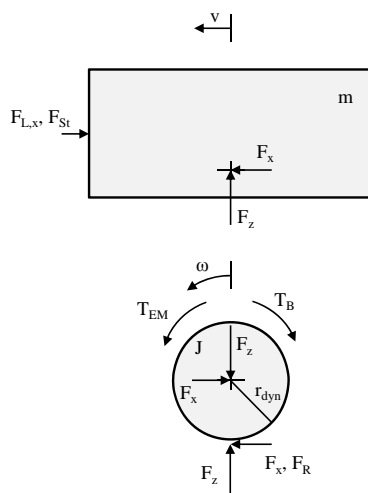


Abbildung 8. Physikalisches Ersatzmodell

Die Reifenlängskraft  $F_x$  wird durch die Radlagerung auf den Aufbau übertragen und führt zu einer Beschleunigung der Aufbaumasse  $m$ . Dieser Bewegung wirken Steigwiderstand  $F_{St}$  und Luftwiderstand  $F_{L,x}$  entgegen.

Das Massenträgheitsmoment der Rolle  $J$  wurde so ausgelegt, dass der Widerstand zur Beschleunigung der Rolle dem Beschleunigungswiderstand des Viertelfahrzeugs  $F_B$  entspricht. Die Fahrwiderstände  $F_{St}$  und  $F_{L,x}$  werden in einem Längsdynamikmodell berechnet und durch den Servoantrieb zusätzlich auf die Rolle aufgebracht.

#### 4.4 Sensorik

Neben der Messung aller Signale für Regelung, Versorgung und Anregungseinheit sind zusätzliche Sensoren für die Identifikation vorgesehen, sodass die Übertragungsfunktionen aller Teilsysteme bestimmt werden können.

#### 4.5 Hardware zur Informationsverarbeitung

Um alle Sensor- und Stellsignale in Echtzeit generieren und verarbeiten zu können, ist der Einsatz einer leistungsfähigen Hardware-Plattform nötig, die hinreichend harte Echtzeitbedingungen realisieren kann [3]. Zu der digitalen Datenverarbeitung gehören neben einem Signalprozessorboard auch Peripherieboards. Die eingesetzten Peripherieboards verfügen über ausreichende D/A-Kanäle zur Signalgenerierung, A/D-Kanäle und Encoderschnittstellen zur Aufnahme der Sensorsignale, CAN-Schnittstelle zur Kommunikation mit dem Servoantrieb sowie digitale Ein- und Ausgänge.

Die Sensor- und Stellsignale für die Antriebsregelung stellen besonders zeitkritische Anforderungen. Hierfür wird

Die Echtzeitanforderungen sind bei der Ansteuerung der Leistungselektronik besonders kritisch. Für diesen Fall ist ein FPGA-basiertes Peripherieboard inklusive entsprechender Schnittstellen vorhanden mit dem Abstraten von bis zu 40 kHz realisiert werden können.

### 5 HiL-System „E-Traktion“

Die vorgestellten Prüfstände für die Traktionsbatterie und die Viertelfahrzeugsdynamik lassen sich über definierte Schnittstellen in der jeweiligen Informationsverarbeitung zu einem hierarchisch übergeordneten HiL-System kombinieren (vgl. Abbildung 9). Dabei fungiert der Antriebsprüfstand als Master und steuert den Batterieprüfstand. Die Benutzerschnittstelle des Batterieprüfstands dient zur Visualisierung und Datenaquisition. Der Nutzer steuert die Testzyklen von der Benutzerschnittstelle 2 aus.

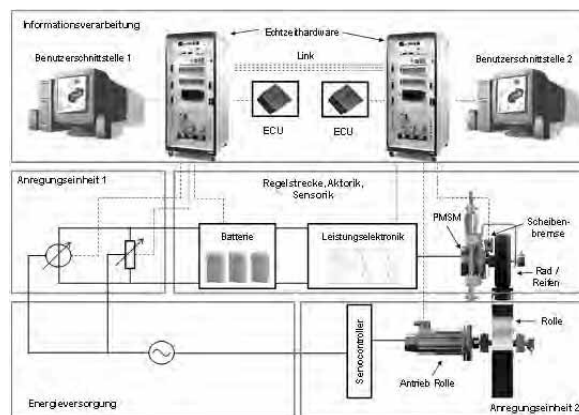


Abbildung 9. HiL-System „E-Traktion“

Auf diese Weise lässt sich der gesamte Elektrische Antriebsstrang von der Traktionsbatterie bis zum Reifen-Straße-Kontakt unter Laborbedingungen reproduzierbar untersuchen. Darüberhinaus wird bereits in frühen Phasen die modellbasierte Entwicklung und Absicherung der vernetzten Regelsysteme Batteriemanagement, Antriebs- und Bremsregelung unterstützt.

## 6 Literatur

- [1] F. Quantmeyer, X. Liu-Henke, W. Diehl und S. Bode. *Parametrierung von Batteriesimulationsmodellen mithilfe eines geeigneten Prüfstands*. Tagungsband zum ASIM/GI-Fachgruppentreffen STS/GMMS, Wolfenbüttel, S. 209-220, 2012.
- [2] F. Quantmeyer, J. Kießling, X. Liu-Henke. *Modellbildung und Identifikation der Energiespeicher für Elektrofahrzeuge*. Tagungsband zum ASIM/GI-Fachgruppentreffen STS/GMMS, Düsseldorf, 2013.
- [3] X. Liu-Henke. *Mechatronische Entwicklung der aktiven Feder- / Neigetechnik für das Schienenfahrzeug RailCab*. Dissertation, Uni Paderborn, 2005.

Die vorgestellten Ergebnisse sind im Rahmen des Projekts *EVM* entstanden. Dieses Projekt wird finanziell durch das Bundesministerium für Bildung und Forschung (BMBF) unterstützt. Förderkennzeichen: 17N0911.



# Ein Verhaltensmodell und ein Laborprüfstand zur SiL- und HiL-Simulation eines zentralen Steuergeräts für ein intelligent-effizientes Elektrofahrzeug

C. Masjosthusmann<sup>1</sup>, U. Köhler<sup>1</sup>, N. Decius<sup>1</sup>, U. Büker<sup>1</sup>

<sup>1</sup>HELLA KGaA Hueck & Co., Beckumer Straße 130, 59552 Lippstadt

*c.masjosthusmann@hella.com*

Das Ziel des von der EU geförderten Projektes eFuture besteht darin, ein sicheres und effizientes batterieelektrisches Fahrzeug zu entwickeln, um den Flaschenhals dieser Fahrzeuggattung, den immer noch die in Kapazität und Lebensdauer begrenzte und teure Batterie repräsentiert, zu weiten. Mit verschiedenen Maßnahmen soll dieses Ziel erreicht werden: Zwei Elektromotoren treiben die beiden Fronträder direkt an, wodurch ein Getriebe und das Differential überflüssig werden. Außerdem ersetzt eine aktive Antriebsdrehmomentverteilung auf die beiden Räder (Torque Vectoring) das ESP, das Fahrstabilität auf Kosten von Verlusten durch Bremsen erreicht. Kameras, Radar- und Navigationsdaten werden genutzt um den Fahrer prädiktiv zu unterstützen. Ein Batteriemanagementsystem und ein Fahrzeug-Energiemanagementsystem sorgen für einen sicheren und angemessenen Umgang mit der Batterie und koordinieren und optimieren den Energiefluss im Fahrzeug.

Diese Funktionen liegen auf einem zentralen, für das Projekt entwickelten, Steuergerät. Zum Testen der Basisfunktionen des Steuergeräts, bestehend aus Autosar-Basissoftware und der Basis-Fahrfunktion (ohne Fahrerassistenzsysteme), sowie des Fahrzeug-Energiemanagements wurde ein Fahrzeugverhaltensmodell entwickelt und ein Laborprüfstand mit Originalteilen des Elektrofahrzeugs aufgebaut. Das Steuergerät hat über den Prüfstand direkten Zugriff auf Aktuatoren und Sensoren aus dem Originalfahrzeug, während über eine CAN-Restbussimulation die Schnittstelle zum übrigen Fahrzeug simuliert wird: Antrieb, Batterie, Hochvolt-Bordnetz, Hochvoltverbraucher, DC/DC-Wandler, 12V-Bordnetz und im Fall der reinen SiL-Simulation der Fahrer.

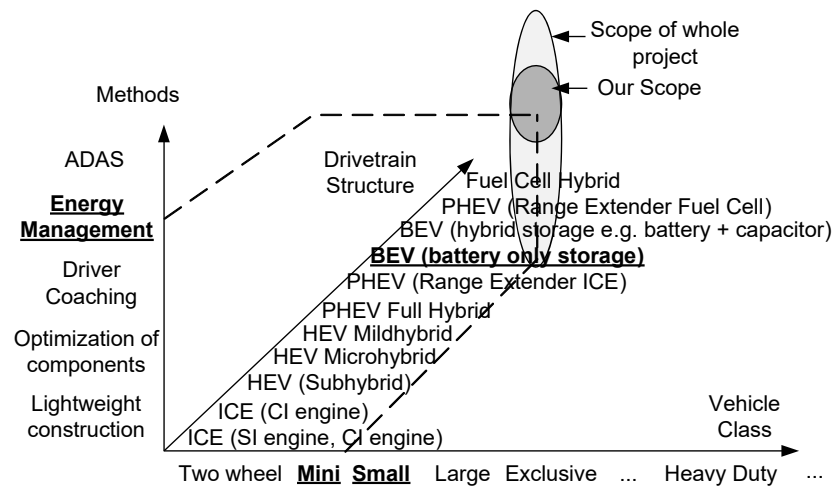
## 1 Einleitung

Begrenzte fossile Energiereserven beeinflussen die individuelle Mobilität von heute und morgen. Um die resultierenden Beschränkungen zu überwinden, befinden sich alternative Energieträger und Fahrzeugkonzepte derzeit in intensiver Untersuchung. Zum einen umfasst dies alternative Brennstoffe wie Wasserstoff oder Ethanol und zum anderen alternative Energiespeicher und -wandler wie Batterien, Supercaps, Schwungräder oder Brennstoffzellen. Dabei dienen hybride Fahrzeugkonzepte als Mischtypen zwischen den Technologien. Die Schlüsseltechnologie, um das Energieproblem im Transport zu lösen, scheint das batterieelektrische Fahrzeug (BEV) zu sein, mit Hybrid-Elektrischen Fahrzeugen (HEV) als wichtigen Zwischenschritt (**Abbildung 1**). Fast alle OEMs arbeiten an voll-elektrischen oder hybrid-

elektischen Fahrzeugen, einige haben sie bereits in der Serienproduktion.

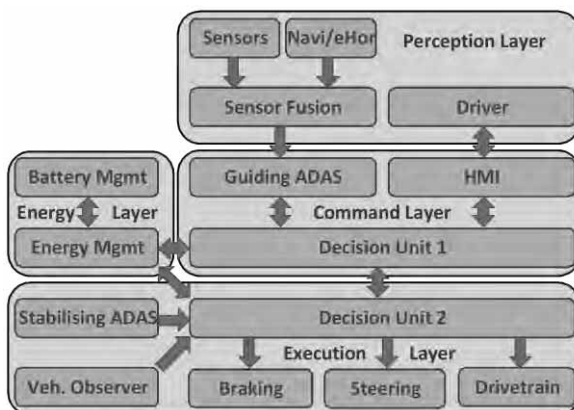
In HEVs spielt das komponentenübergreifende Energiemanagement eine wichtige Rolle beim Koordinieren der Energieflüsse und der Erhöhung der Energieeffizienz des Gesamtfahrzeugs [1].

Den Flaschenhals in BEVs stellt immer noch die teure Batterie dar, begrenzt die in Kapazität und Lebensdauer begrenzt ist. Große Forschungsaufwände konzentrieren sich auf Elektrofahrzeuge mit hybriden Speichern, wie Batterien plus Supercaps, was zwar Energiemanagementalgorithmen nötig macht, aber Gewichtseinsparungen verspricht. Die hier vorgestellte Arbeit ist Teil des laufenden öffentlich geförderten Projektes eFuture. Das Ziel des Projektes ist die Erhöhung der Sicherheit und Effizienz von Elektrofahrzeugen [2-5].



**Abbildung 1:** Verschiedene Klassifizierungen der Elektrifizierung des Fahrzeugs - Klassifizierungen nach Fahrzeugart, Antriebsstrang und Effizienzmaßnahmen - sowie der Fokus im Projekt

Im Rahmen des Projektes wurde dazu ein zentrales Steuergerät (sog. VHU, für Vehicle Head Unit) aufgebaut, das zum einen die Basisfunktionen steuert, die nötig sind um ein Elektrofahrzeug zu betreiben, und zum anderen neue Funktionen beheimatet oder zumindest Schnittstellen zu diesen stellt. Diese Software-Funktionen agieren zusammen innerhalb eines zentralen Ansatzes mit „Decision Units“ als zentrale Schaltstellen. Diese Funktionen sind die Mensch-Maschine-Schnittstelle (HMI, Human Machine Interface), „grüne“ und sichere ADAS Funktionen basierend auf Umgebungssensoren und Navigationsdaten, der Fahrzeugbeobachter, Fahrdynamikfunktionen wie Torque Vectoring sowie das Fahrzeug-Energiemanagement (**Abbildung 2**).



**Abbildung 2:** Software Architektur im Projekt eFuture

Zum Testen des Steuergeräts und für den Entwurf der Basisfahrfunktionalitäten sowie des Fahrzeugenergiemanagements wurde ein Simulationsmodell erstellt und ein Laborprüfstand aufgebaut.

Zusammen mit dem Laborprüfstand dient das Simulationsmodell der „Software in the Loop“ (SiL) als auch der „Hardware in the Loop“ (HiL) Simulation.

## 2 Steuergerät, Simulationsmodell und Laborprüfstand

Die Basisfunktionen zum Betrieb des Steuergeräts und das Fahrzeug-Energiemanagement befinden sich auf der zentralen VHU. Zum Testen dieses Steuergeräts und der Funktionen wurde ein Simulationsmodell entwickelt, das die Peripherie der VHU simuliert. Dies erfolgt anhand eines Beschreibungsmodells des Fahrzeugs inklusive Fahrdynamik, Bordnetz und Komponenten. Für die HiL-Simulation wurde des Weiteren ein Laborprüfstand aufgebaut, der auch Originalteile aus dem Fahrzeug beinhaltet.

### 2.1 Steuergerät

Die zentrale VHU zeigt **Abbildung 3**). Die VHU ist eine Hardware-Plattform, die zum einen die Basisfunktionalität zum Betrieb eines Elektroautos beinhaltet, und die zum anderen in der Lage ist fortschrittliche Software-Algorithmen auszuführen, die im Projekt eFuture entwickelt werden.



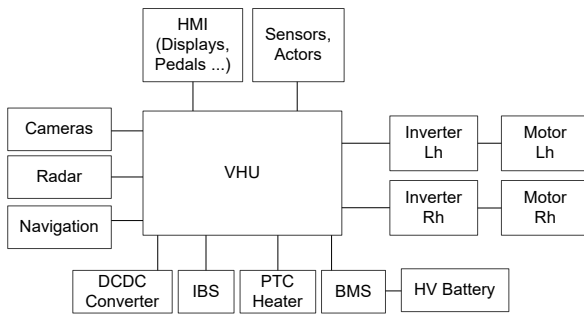
**Abbildung 3:** Entwickelte Hardware-Plattform: VHU



Es ist ein Multiprozessor-System mit vier 32 Bit-Controllern, die mit Autosar-Basissoftware betrieben werden. Das Steuergerät ist mit verschiedenen CAN-Bus-Systemen verbunden, um mit dem Antrieb, dem Batteriemanagementsystem, dem Navigationssystem und anderen ECUs zu kommunizieren.

Es besitzt verschiedene digitale und analoge I/Os, um zum Beispiel Relais zu treiben, Sensoren, wie das Fahrpedal zu lesen oder Lasten, wie die Unterdruckpumpe, zu treiben.

Außerdem bildet die VHU die Schnittstelle zu verschiedenen Sensoren (z. B. Kameras, Radar, Lenkradschalter, Beschleunigungs-, Gierraten- und Lenkwinkelsensoren), zur Mensch-Maschine-Schnittstelle und beheimatet die zentrale Antriebssteuerung, die den Antrieb in Form von sogenannten „Decision Units“ sicher ansteuert (**Abbildung 2** und **Abbildung 4**).



**Abbildung 4:** VHU mit Schnittstellen zu verschiedenen Komponenten

Das Steuergerät ist auch auf Sicherheitsanforderungen vorbereitet, z.B. in Form von redundanten Ein- und Ausgängen, Prozessoren und Spannungsversorgungen.

Insbesondere der Antrieb, mit seinen beiden Vorderrad-Direktantrieben, hat hohe Sicherheitsanforderungen wegen der Gefahr hoher Drehmomentunterschiede zwischen den beiden Rädern.

## 2.2 Simulationsmodell

Für die Entwicklung der Basisfahrfunktionalität und des Energiemanagements wurde ein geeignetes Simulationsmodell entwickelt (**Abbildung 5**). Es ist ein Beschreibungsmodell, das die longitudinale Dynamik des Fahrzeugs, den Antriebsstrang, sowie das Hochvolt-Bordnetz und Teile des 12 V-Bordnetzes simuliert. Dies beinhaltet die Hochvolt- und die 12 V-Batterie<sup>1</sup>, den DC/DC-Wandler, die PTC-Heizung für

den Fahrgastraum und einige weitere elektrische Verbraucher.

Für das Batteriemodell wurde ein Ersatzschaltkreismodell gewählt [1, 7], weil es nur einen relativ einfachen Parametersatz hat im Vergleich zu beispielsweise chemischen Batteriemodellen mit umfangreichen Parametersätzen, die aufwändig zu bestimmen wären.

Das Modell wurde in den Simulationstools „Simplorer“ und „Matlab/Simulink“ umgesetzt.

Für die HiL Simulation wird das Simulink Modell an die Restbussimulation mit dem Simulationstool CANoe gekoppelt.

Das Simulationsmodell lässt sich mit den folgenden Gleichungen charakterisieren.

Die longitudinale Dynamik beschreibt die Fahrzeuggeschwindigkeit  $v$ , während die beiden Antriebsmotoren ein Drehmoment  $\tau_{Lh}$  am linken Rad und ein Drehmoment  $\tau_{Rh}$  am rechten Rad aufbringen oder die Bremse ein Drehmoment  $\tau_{Br}$  erzeugt [1]:

$$v = \frac{1}{m_v \cdot e + m_L} \frac{\tau_{Lh} + \tau_{Rh} + \tau_{Br}}{r_w} - F_a - F_g - F_f \quad (1)$$

$$F_a = \frac{1}{2} \rho_{air} \cdot A_v \cdot c_w \cdot v^2 \cdot \text{sgn } v \quad (2)$$

$$F_g = m_v \cdot g \cdot \sin \alpha \quad (3)$$

$$F_f = m_v \cdot g \cdot c_r \cdot \cos \alpha \cdot \text{sgn } v \quad (4)$$

Hierbei ist  $m_v$  die Fahrzeugmasse,  $e$  der Massenfaktor,  $m_L$  die Masse der Fahrzeugzuladung,  $r_w$  der Radradius,  $F_a$  die Luftwiderstandskraft,  $F_g$  die Gravitationskraft,  $F_f$  die Rollwiderstandskraft,  $\rho_{air}$  die Luftdichte,  $A_v$  die effektive Stirnfläche,  $c_w$  der Strömungswiderstandskoeffizient,  $g$  die Erdanziehungskraft,  $\alpha$  die Steigung und  $c_r$  der Rollwiderstandskoeffizient.

Der Antrieb wird mit einem Wirkungsgrad  $\eta_{Dt} \tau_{Xh}, \omega_{Xh}$  beschrieben, der drehmoment- und geschwindigkeitsabhängig sein kann (Wirkungsgradkennfeld, vgl. [1]). Dieser beschreibt die Beziehung zwischen der mechanischen Leistung  $P_{mech, Xh}$  und der elektrischen Leistung  $P_{el, Xh}$  des linken und rechten ( $Xh := \{Lh, Rh\}$ ) Teils des Antriebsstrangs:

<sup>1</sup> In diesem Zusammenhang bezieht sich Hochvolt (HV) auf das Spannungsniveau des Antriebsstrangs. Der Ausdruck

bezieht sich nicht auf die IEC Definition für Hochspannung.

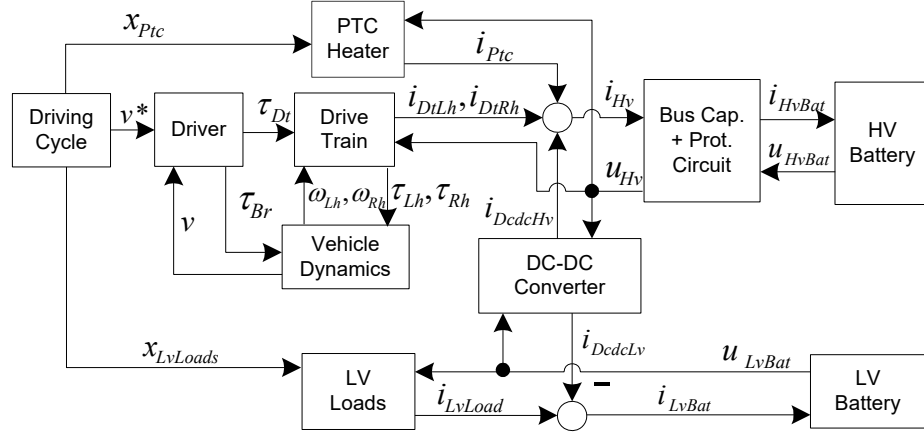


Abbildung 5: Übersicht zum Simulationsmodell

$$P_{el,Xh} = \frac{1}{\eta_{Dt}(\tau_{Xh}, \omega_{Xh})} P_{mech,Xh}, \quad P_{mech,Xh} \geq 0 \quad (5)$$

$$P_{mech,Xh} = \eta_{Dt}(\tau_{Xh}, \omega_{Xh}) P_{el,Xh}, \quad P_{mech,Xh} < 0$$

$$P_{mech,Xh} = \tau_{Xh} \cdot \omega_{Xh}, \quad \omega_{Xh} = \frac{v}{r_w} \quad (6)$$

$$\tau_{Xh}(s) = \frac{1}{2} \tau_{Dt}(s) \cdot G_{Dt}(s) \quad (7)$$

$$P_{el,Xh} = u_{Hv} \cdot i_{Dt,Xh} \quad (8)$$

Hierbei ist  $\tau_{Xh}$  das Drehmoment des linken/rechten Motors und  $\omega_{Xh}$  die entsprechende Winkelgeschwindigkeit,  $P_{el}$  ist die elektrische Leistung des gesamten Antriebstrangs,  $u_{Hv}$  ist die Spannung des Hochvolt-Bordnetzes an den Umrichtern,  $i_{Dt,Xh}$  ist der Strom, den der Antriebsstrang dem Bordnetz entnimmt,  $\tau_{Dt}$  das angeforderte Antriebsdrehmoment und  $G_{Dt}(s)$  spiegelt als Übertragungsfunktion die Dynamik zwischen dem angeforderten und tatsächlichen Drehmoment wider.

Für die Hochvolt-Batterie wurde ein elektrisches Ersatzschaltkreis-Modell gewählt, dessen Leerlaufspannung  $OCV$  vom Ladezustand  $SOC$  der Batterie und deren Temperatur  $T_{HvBat}$  abhängt [1,7]:

$$u_{HvBat} = N_s \cdot u_{Cell}, \quad i_{Cell} = \frac{i_{HvBat}}{N_p} \quad (9)$$

$$u_{Cell} = OCV(SOC, T_{HvBat}) - u_{drop}(i_{Cell}, T_{HvBat}) \quad (10)$$

$$u_{drop}(i_{Cell}, T_{HvBat}) = R_o(T_{HvBat}) \cdot i_{Cell} + u_{drop,TS} + u_{drop,TL} \quad (11)$$

$$u_{drop,TS} = \frac{i_{Cell} - u_{drop,TS}/R_{TS}(T_{HvBat})}{C_{TS}(T_{HvBat})} \quad (12)$$

$$u_{drop,TL} = \frac{i_{Cell} - u_{drop,TL}/R_{TL}(T_{HvBat})}{C_{TL}(T_{HvBat})} \quad (13)$$

$$SOC = \frac{i_{Cell}}{C_{Cell}} \quad (14)$$

mit den Größen Hochvolt-Batteriespannung  $u_{HvBat}$ , Batteriestrom  $i_{HvBat}$ , Anzahl der Zellen in Serien ( $N_s$ ) und Parallelschaltung ( $N_p$ ), Zellenspannung  $u_{Cell}$ , Zellenstrom  $i_{Cell}$ , ohmscher Widerstand  $R_o$ , Zellen-Spannungsabfall  $u_{drop}$ , der Anteil des Spannungsabfalls mit kleiner Zeitkonstante  $u_{drop,TS}$  und großer Zeitkonstante  $u_{drop,TL}$ , die sich aus den Parametern  $R_{TS}$ ,  $C_{TS}$ ,  $R_{TL}$  und  $C_{TL}$  zusammensetzen und der nominellen Zellenspannung  $C_{Cell}$ .

Die 12 V Batterie wurde auf ähnliche Weise modelliert.

Der Zwischenkreiskondensator, der Umrichter mit der Kapazität  $C_{Bus}$ , sowie die Vorlade- und Schutz-Schaltung sind folgendermaßen angenähert:

$$i_{HvBat} = \begin{cases} 0, & \text{Batterie Relais geöffnet} \\ \frac{u_{HvBat} - u_{Hv}}{R_{Precharge}}, & \text{Vorladen aktiv und} \\ \frac{u_{HvBat} - u_{Hv}}{R_{Relay}}, & \text{Batterie Relais geschlossen} \end{cases} \quad (15)$$

$$u_{Hv} = \frac{1}{C_{Bus}} \cdot i_{HvBat} - i_{Hv} \quad (16)$$

Hierbei ist  $R_{Precharge}$  der strombegrenzende Vorladewiderstand,  $R_{Relay}$  der Widerstand bei geschlossenem Schutzrelais und abgeschlossenem Vorladevorgang und  $i_{Hv}$  der Laststrom des Hochvolt-Bordnetzes.

Um das 12 V-Bordnetz mit Energie zu versorgen, ist ein DC/DC-Wandler notwendig, der die Hochvolt-Spannung  $u_{Hv}$  in eine 12 V-Spannung  $u_{LvBat}$  wandelt:

$$e_{Dcdc} = u_{LvBat}^* - u_{LvBat} \quad (17)$$

$$i_{DcdcLv} \quad (18)$$

$$g_{Dcdc}(e), \quad 0 < g_{Dcdc} e < I_{Dcdc,max} \text{ und Umrichter an} \\ = I_{Dcdc,max}, \quad g_{Dcdc} e_{Dcdc} > I_{Dcdc,max} \text{ und Umrichter an} \\ 0, \quad g_{Dcdc} e_{Dcdc} < 0 \text{ oder Umrichter aus} \quad (19)$$

$$P_{DcdcHv} = \frac{1}{\eta_{Dcdc}} \cdot P_{DcdcLv} \quad (20)$$

$$P_{DcdcHv} = u_{Hv} \cdot i_{DcdcHv} \quad (20)$$

$$P_{DcdcLv} = u_{LvBat} \cdot i_{DcdcLv}, \quad (21)$$

Wobei  $e_{Dcdc}$  der Regelfehler des DC/DC-Wandler-Modells ist,  $u_{LvBat}^*$  die 12 V Sollspannung,  $i_{DcdcLv}$  der 12 V Wandlerstrom,  $I_{Dcdc,max}$  der maximale 12 V-Wandlerstrom,  $i_{DcdcHv}$  der Hochvolt-Wandlerstrom,  $g_{Dcdc}$  die modellierte Umrichter-Übertragungsfunktion des offenen Regelkreises und  $\eta_{Dcdc}$  der Wirkungsgrad des Umrichters.

Die PTC-Heizung mit dem Strom  $i_{Ptc}$  und eine 12 V-Last mit dem Strom  $i_{LvLoad}$  werden charakterisiert durch

$$i_{Ptc} = \frac{D_{Ptc} \cdot u_{Hv}}{R_{Ptc}}, \quad \text{Heizung an} \quad (22)$$

$$0, \quad \text{Heizung aus} \\ i_{LvLoad} = \frac{u_{LvBat}}{R_{LvLoad}}, \quad 12 \text{ V Last an} \quad (23) \\ 0, \quad 12 \text{ V Last aus}$$

$D_{Ptc}$  ist das Tastverhältnis der PWM-gesteuerten PTC-Heizung,  $R_{Ptc}$  ist der Widerstand der PTC-Heizung und  $R_{LvLoad}$  ist der Widerstand der 12 V-Beispiellast.

Der Fahrer, der einem definierten Geschwindigkeits-Fahrzyklus  $v^*(t)$  folgt und dazu ein Drehmoment  $\tau^*$  auf das Fahrzeug ausüben möchte, wird mit einem Regler  $G_{Driver}$  approximiert [vgl. 6]:

$$\tau^* s = v^* s - v s \cdot G_{Driver}(s) \quad (24)$$

$$\tau_{Dt} \quad (25)$$

$$= \begin{cases} \tau^*, & \tau_{Dt,min} v < \tau^* < \tau_{Dt,max}(v) \\ \tau_{Dt,max}(v), & \tau^* > \tau_{Dt,max}(v) \\ \min(0, \tau_{Dt,min}(v)), & \tau^* < \tau_{Dt,min}(v) \end{cases} \quad (26)$$

$$\tau_{Br} = \begin{cases} \tau_{Dt} - \tau^*, & 0 < \tau_{Dt} - \tau^* < \tau_{Br,max} \\ \tau_{Br,max}, & \tau_{Dt} - \tau^* > \tau_{Br,max} \\ 0, & \tau_{Dt} - \tau^* < 0 \end{cases}$$

Hierbei ist  $\tau_{Dt}$  das Antriebsdrehmoment,  $\tau_{Br}$  das Bremsmoment,  $\tau_{Dt,min}$  das minimale und  $\tau_{Dt,max}$  das maximale Antriebsdrehmoment, das aus dem geschwindigkeitsabhängigen Pedalposition zu Drehmoment Kennfeld abgeleitet wurde, und  $\tau_{Br,max}$  das maximale Bremsmoment.

Die Temperatur  $T$  relevanter Komponenten wird mit einem einfachen thermischen Modell beschrieben (vgl. 4):

$$\Delta T = \frac{(P_{th} - P_{rad} - P_{conv} - P_{cond})}{C_{th}} \quad (27)$$

$$T = T_{Amb} + \Delta T \quad (28)$$

Die Heizleistung  $P_{th}$  des Antriebs ist

$$P_{th,DtXh} = |P_{el,Xh} - P_{mech,Xh}|. \quad (29)$$

Die Heizleistung einer Hochvolt-Batteriezele ist

$$P_{th,Cell} = R_o T_{Bat} \cdot i_{cell}^2 + \frac{u_{drop,TS}^2}{R_{TS} T_{Bat}} + \frac{u_{drop,TL}^2}{R_{TL} T_{Bat}} + c_{mr} \cdot i_{cell}. \quad (30)$$

Und somit ist die Heizleistung der Gesamtbatterie

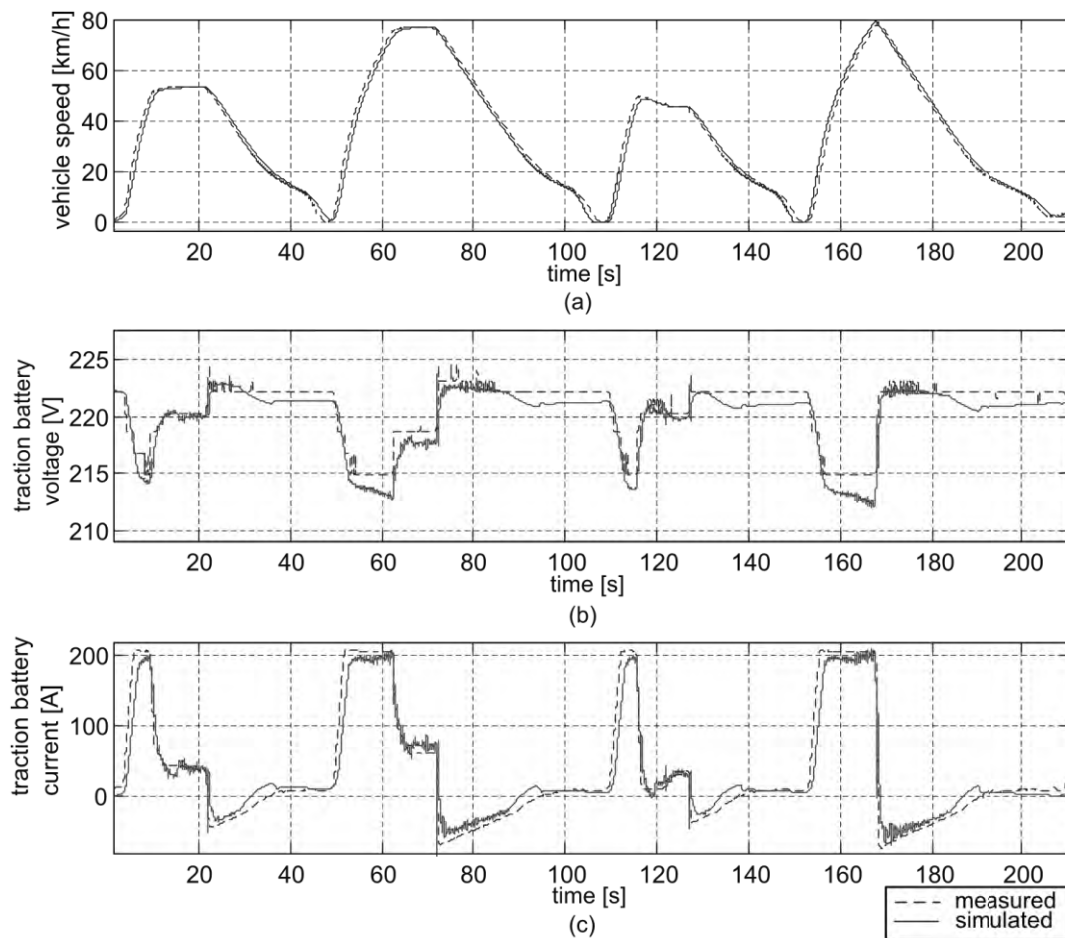
$$P_{th,HvBat} = N_s \cdot N_p \cdot P_{th,Cell}. \quad (31)$$

In einem ersten Ansatz wird die Wärmeabfuhr mit einem thermischen Widerstand  $R_{th}$  linear approximiert:

$$P_{rad} + P_{conv} + P_{cond} = \frac{\Delta T}{R_{th}} \quad (32)$$

$T_{Amb}$  ist die Umgebungstemperatur,  $P_{rad}$  ist die Wärmeabfuhrleistung durch Strahlung,  $P_{conv}$  ist die Wärmeabfuhrleistung durch Konvektion,  $P_{cond}$  ist die Wärmeabfuhrleistung durch Wärmeleitung,  $C_{th}$  ist die thermische Kapazität der Komponente und  $c_{mr}$  ist der thermische Koeffizient, der die Wärmeauf- und abnahme der Hauptreaktion beschreibt.

**Abbildung 6** zeigt, dass das Modell geeignet ist, um die zu entwickelten Softwarekomponenten in erster Instanz zu evaluieren.



**Abbildung 6:** Gegenüberstellung von Messergebnissen und Simulationsergebnissen: (a) Fahrzeuggeschwindigkeit, (b) Batteriespannung und (c) Batteriestrom eines ähnlichen Fahrzeugs der Vorgängergeneration.

### 2.3 Laborprüfstand

Für die Hardware in the Loop-Simulation (siehe **Abbildung 7** und **Abbildung 8**) wurde ein Laborprüfstand aufgebaut, unter Mitverwendung von Originalteilen des Fahrzeugs. Die VHU wird über deren Molex-Stecker mit dem Laborprüfstand verbunden.

Mit dem Mess- und Kalibrierungs-Tool CANape ist es möglich Variablen aus der Steuergerätesoftware über den Vehicle CAN auszulesen und zu schreiben. Dazu dient das XCP Protokoll. Des Weiteren ist es möglich, CANape mit dem Restbus-Simulationstool CANoe zu steuern und so auch während der Simulation auf die Mess- und Stellgrößen im Steuergerät zuzugreifen.

Die AUTOSAR Applikationssoftwarekomponenten werden mithilfe von Matlab/Simulink und TargetLink modellbasiert entwickelt. Der automatisch generierte C-Code wird zusammen mit der restlichen AUTOSAR-Software kompiliert, gelinkt und anschließend

mit dem Debugger über JTAG auf die Mikrocontroller geflasht.

Die Restbussimulation simuliert die übrigen Steuergeräte im Fahrzeug, wie zum Beispiel die Umrichter und das Batteriemanagementsystem, während die Signale auf dem simulierten Restbus vom Simulink Modell gelesen und geschrieben werden. Die Toolkopplung erfolgt über das CANoe Matlab Interface

Mit diesem Aufbau kann die entwickelte Software bereits im Labor getestet werden, bevor das Fahrzeug verfügbar ist und die Software in der Realität erprobt werden kann.

Für die Softwareintegration im Labor wird sinnvollerweise ein Laptop verwendet, denn derselbe Rechner lässt sich auch für Tests im Fahrzeug einsetzen.

Im letzten Jahr wurde die VHU mit der auf diese Weise entwickelten Software mit einfacher Fahrfunktion bereits erfolgreich in das Fahrzeug integriert.

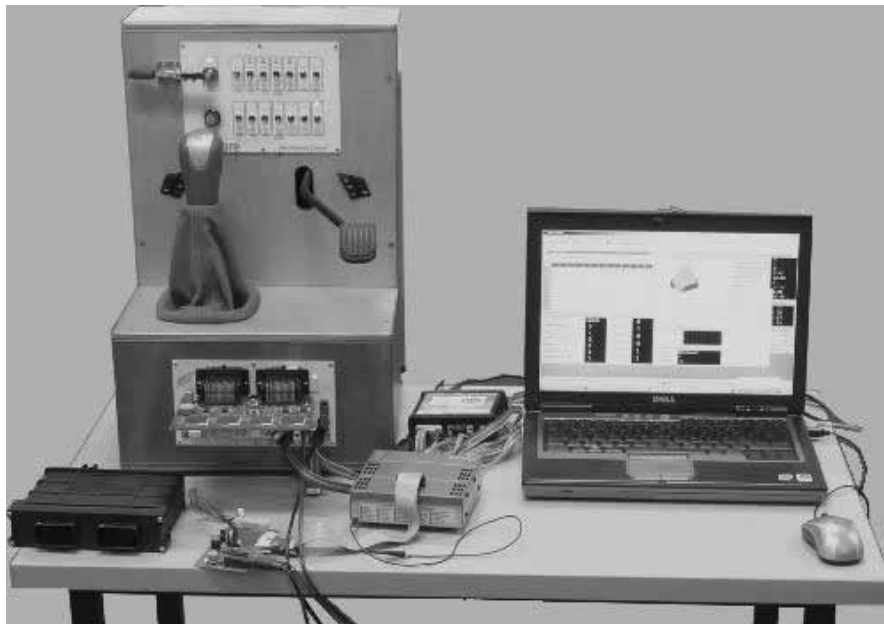


Abbildung 7: VHU, verbunden mit dem Laborprüfstand

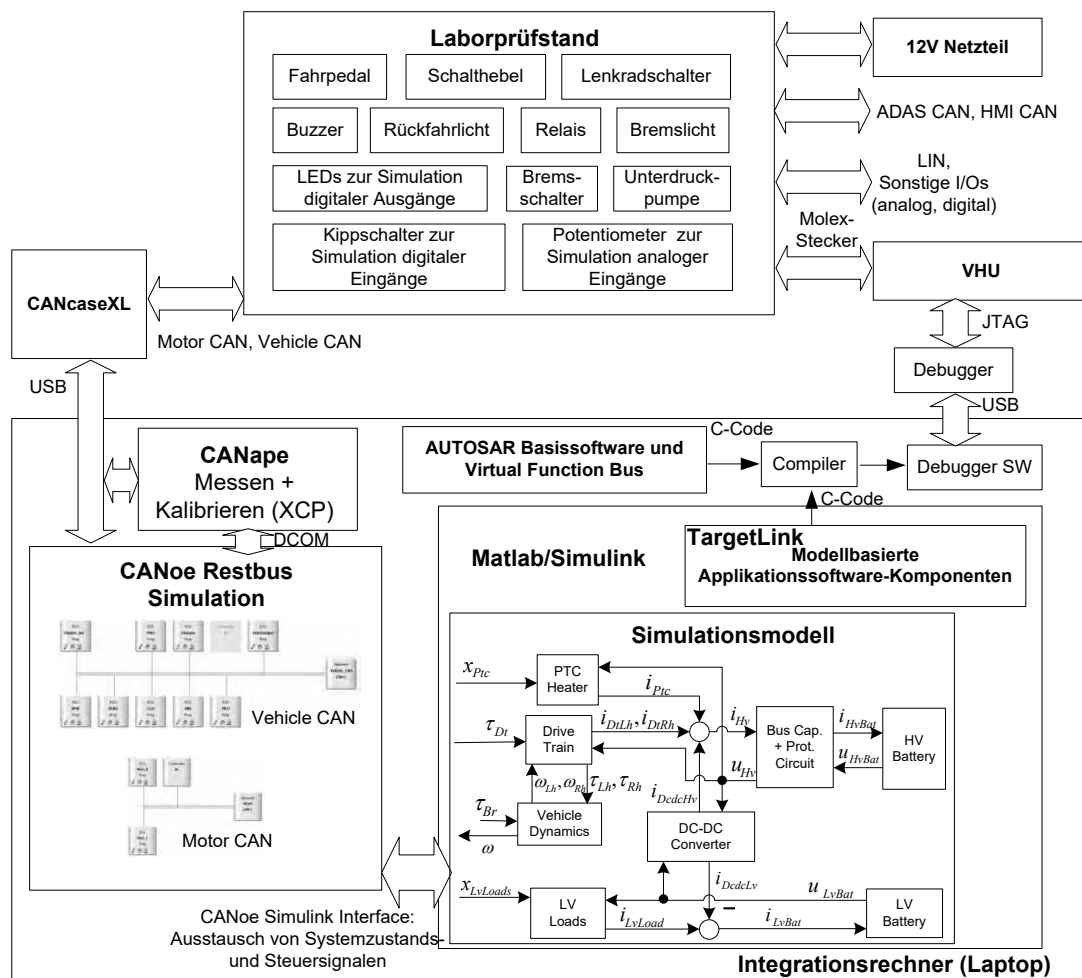


Abbildung 8: Blockdiagramm zum Aufbau der Hardware in the Loop Simulation

### 3 Zusammenfassung

Elektrofahrzeuge kombiniert mit regenerativen Energiequellen stellen eine Möglichkeit dar, die individuelle Mobilität unter der Bedingung schwindender fossiler Ressourcen zu erhalten. Im Rahmen des öffentlich geförderten Projektes eFuture wurden ein Fahrzeugsimulationsmodell eines Elektrofahrzeugs und ein Laborprüfstand entwickelt. Die vorgestellte Simulationsumgebung dient der Entwicklung und dem Test von Steuergerätesoftware. Die Basisfunktionalitäten des ebenfalls entwickelten Steuergeräts wurden auf diese Weise bereits erfolgreich implementiert: das Fahrzeug ist fahrbereit.

Als nächste Schritte folgen die weitere Implementierung und Integration des Energiemanagementmoduls und die Integration der von den Projektpartnern gelieferten Softwarekomponenten in das Steuergerät. Mit darauf folgenden Messfahrten wird das entwickelte Fahrzeugmodell und die entwickelte Steuergerätesoftware evaluiert und mit den gewonnen Erkenntnissen sukzessive verbessert.

Außerdem wird, aufbauend auf dem entwickelten Fahrzeugmodell und den gewonnen Erkenntnissen, derzeit ein generisches Fahrzeugmodell entwickelt, inklusive Konzept zur bedarfsweisen Einbindung weiterer Simulationstools wie Dymola, Simpler oder Carmaker.

### 4 References

- [1] L. Guzzella, and A. Sciarretta, *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*, Springer, Deutschland, 2005.
- [2] eFuture Webseite, <http://www.efuture-eu.org>, letzter Abruf im Januar 2013.
- [3] V. Scheuch., E/E Architecture for Battery-electric Vehicles, ATZelextronik worldwide Edition, Deutschland, 2011.
- [4] V. Scheuch, Simplified architecture by the use of decision units, 16th International Forum on Advanced Microsystems for Automotive Applications, Deutschland, 2012.
- [5] C. Masjosthusmann, U. Köhler, N. Decius and U. Büker, A Vehicle Energy Management System for a Battery Electric Vehicle, 2012 IEEE Vehicle Power and Propulsion Conference (VPPC), Südkorea, 2012.
- [6] E. Hessel, H. Klein, J. Haase, Analyse der Energiebilanz von Elektrofahrzeugen, ASIM-Konferenz STS/GMMS 2011, Deutschland, 2011
- [7] M. Knauff, J. McLaughlin, C. Dafis, D. Niebur, P. Singh, H. Kwatny, and C. Nwankpa, Simulink Model of a Lithium-Ion Battery for the Hybrid Power System Testbed, ASNE Intelligent Ships Symposium, USA, 2007.

# Modellbildung und Identifikation der Energiespeicher für Elektrofahrzeuge

Florian Quantmeyer, Jan Kießling, Xiaobo Liu-Henke

Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel

*Flo.Quantmeyer@Ostfalia.de*

In diesem Beitrag wird zuerst der Energiespeicher – eine Batterie, bestehend aus mehreren Lithium-Ionen-Zellen – mit verschiedener Modellierungstiefe modelliert. Durch Identifikation mittels Frequenzmessung in verschiedenen Arbeitspunkten und anschließende Verifikation im Zeitbereich werden sowohl die Modellstruktur als auch die Systemparameter identifiziert. Zur Durchführung der Frequenzmessung wird dabei die Elektrochemische Impedanzspektroskopie (EIS) herangezogen. Beispielhaft wird der iterative Charakter der gewählten Vorgehensweise verdeutlicht. Anhand dynamischer Testzyklen wird die Genauigkeit des Batteriemodells auch bei hohen Frequenzen nachgewiesen.

## 1 Einleitung

Derzeitig stellen Batteriesysteme ein wesentliches Hindernis für die flächendeckende Verbreitung von Elektrofahrzeugen dar. Die im Vergleich zu fossilen Energieträgern etwa um Faktor 40 geringere Energiedichte führt trotz hoher Wirkungsgrade bei der Energiewandlung zu einer erheblichen Reduzierung der Reichweite von Elektrofahrzeugen. Eine weitere Bremse sind die hohen Investitionskosten in Verbindung mit der begrenzten Lebensdauer. Auch die Sicherheit einiger Batterietypen stellt derzeit noch ein Risiko dar.

Hinsichtlich dieser Nachteile, die sich aus der Verwendung elektrischer Energiespeicher in Fahrzeugen ergeben, ist es umso wichtiger die Traktionsbatterie zur vollständigen Ausnutzung des verfügbaren Potentials bestmöglich zu betreiben. Dies gewährleistet das Batteriemanagementsystem, das gemeinsam mit der Batterie das Batteriesystem bildet. Es übernimmt unter anderem die Gewährleistung der Sicherheit, die Ladezustandsschätzung, den Ausgleich der Ladestände auf Zellbasis sowie die Temperaturregelung. Bei der Auslegung ist ein Kompromiss zwischen Maximierung der Lebensdauer und vollständiger Ausnutzung der Kapazität zu treffen. Die Eigenschaften des Batteriesystems werden somit maßgeblich durch das Batteriemanagement beeinflusst.

Für die Entwicklung des Batteriemanagements sind Batteriemodelle erforderlich. Einerseits ermöglichen sie den Einsatz der modellbasierten, verifikationsorientierten Entwicklungsmethodik von Model-in-the-Loop über Software-in-the-Loop bis hin zu Hard-

ware-in-the-Loop, andererseits dienen sie als Bestandteil des Batteriemanagements z.B. im modellbasierten Ladezustandsschätzer. Dabei hängt die Güte der Schätzung insbesondere von der Modellgüte ab.

## 2 Batteriemodelle

Grundsätzlich existiert eine hohe Anzahl von Batteriemodellen und Ansätzen zur Beschreibung des Verhaltens von Batteriezellen. Abbildung 1 stellt eine Übersicht verschiedener Modelle dar, wobei kein Anspruch auf Vollständigkeit erhoben wird.

Die Batteriemodelle werden hier in drei Kategorien eingeteilt. Die physikalisch-chemischen Modelle bilden die Prozesse in der Zelle auf mikroskopischer Ebene ab. Die Kenntnis der Modellparameter wie Geometrien, Diffusionskonstanten, Leitfähigkeiten, etc. ist für die Modellbildung vorausgesetzt [1]. Im Gegensatz zu den anderen Modellkategorien bilden diese Modelle die realen Prozesse innerhalb der Zelle ab. Damit sind sie die einzigen Modelle, die eine physikalische Modellbildung erlauben.

Zur vollständigen Abbildung des Gesamtzellverhaltens ist die Kombination mehrerer Teilmodelle, wie beispielsweise dem Diffusionsansatz und Elektrodenmodellen erforderlich.

Ersatzschaltbildmodelle bilden das Gesamtverhalten der Batteriezelle ab. Rückschlüsse auf die physikalisch-chemischen Vorgänge lassen sich häufig nicht ziehen. Mithilfe von Schaltbildern aus elektrischen Komponenten wie Widerständen, Kondensatoren und Induktivitäten wird sowohl das statische als auch das dynamische Verhalten nachgebildet. Dies kann so-

wohl das elektrische als auch das thermische Verhalten betreffen.

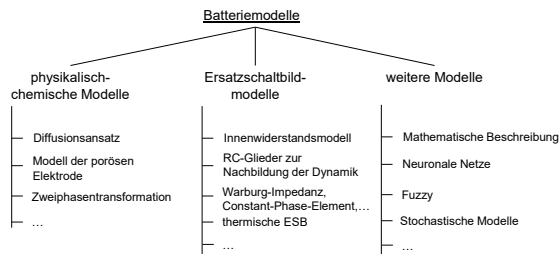


Abbildung 1. Batteriemodelle nach [1], [2], [3]

Unter der Kategorie weitere Modelle sind verschiedene Ansätze zur Modellierung zusammengefasst. Es handelt sich zumeist um Black-Box-Modelle wie mathematische Beschreibungen, Neuronale Netze und Stochastische Modelle.

### 3 Modellbildung

Im ingenieurwissenschaftlichen Bereich haben sich vor allem elektrische Ersatzschaltbildmodelle etabliert. Diese bieten den Vorteil, das Verhalten der Gesamtzelle mit geringem Rechenaufwand, echtzeitfähig und mit ausreichend hoher Genauigkeit simulieren zu können. Die Detailtiefe kann dabei durch einfache Modifikation des elektrischen Ersatzschaltbilds auf den Einsatzzweck zugeschnitten werden. Zudem ist verglichen mit chemischen Modellen nur ein geringer Aufwand für die Parameteridentifikation aufzubringen. Weiterhin ist eine Kopplung von Einzelmodellen einfach realisierbar.

Das elektrische Verhalten einer Zelle lässt sich durch eine Verschaltung elektrischer Grundelemente wie Induktivität ( $L$ ), Widerstand ( $R$ ) und Kapazität ( $C$ ) sowie Strom- bzw. Spannungsquelle abbilden. Mittels dieser Elemente kann ein elektrisches Ersatzschaltbild entwickelt werden, welches durch einen Strom angeregt, das Spannungsverhalten der Zelle abbildet. Durch Widerstände werden rein ohmsche Anteile der Batteriespannung und durch Induktivitäten bzw. Kapazitäten frequenzabhängige Spannungsabfälle nachgebildet.

Abbildung 2 zeigt das elektrische Ersatzschaltbild einer Batteriezelle mit einer idealen Spannungsquelle und einem seriellen Widerstand sowie einer beliebigen Anzahl von RC-Gliedern.

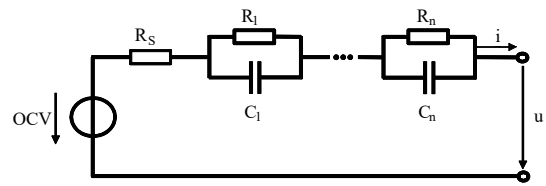


Abbildung 2. Ersatzschaltbildmodell

Die Spannungsquelle bildet die Leerlaufspannung (auch OCV: open circuit voltage) der Zelle ab. Der serielle Widerstand  $R_s$  beschreibt den ohmschen Innenwiderstand der Zelle und durch die RC-Glieder wird das dynamische Spannungsverhalten der Zelle charakterisiert.

Die Parameter OCV,  $R_s$  und  $R_{1...n}$  sowie  $C_{1...n}$  hängen nichtlinear von mehreren Einflussgrößen ab. Die Identifikation aller Einflüsse ist nur mit Hilfe umfangreicher Messungen realisierbar. Tabelle 1 zeigt die wichtigsten Einflüsse.

Tabelle 1

Parameter	Abhängigkeit
OCV	SoC, SoH, I, T, t
$R_s$	SoC, SoH, I, T
$R_{1...n}$ ; $C_{1...n}$	Soc, T, SoH

Da alle Parameter eine Abhängigkeit vom Ladezustand (SoC) der Zelle aufzeigen, muss der Ladezustand zu jedem Zeitpunkt  $t$  bekannt sein. Durch Integration des Stroms und Vergleich mit der Zellnennkapazität  $C_n$  kann der aktuelle SoC nach (1) ermittelt werden.

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

Zur Implementierung des Modells in die CAE-Umgebung wird die mathematische Beschreibung des Systems aus dem Ersatzschaltbild abgeleitet. Es wird zunächst lediglich die Abhängigkeit der Modellparameter vom Ladezustand berücksichtigt. Die Klemmenspannung  $u$  der Zelle infolge einer Stromanregung  $i$  ergibt sich wie folgt nach (2).

$$u(SoC, t) = u_{OC}(SoC) + u_{R_s}(SoC, t) + u_{RC1...n}(SoC, t) \quad (2)$$

Für das dynamische Verhalten leitet sich der Frequenzgang in jedem Betriebspunkt nach (3) ab.



$$G(j\omega) = \frac{u(j\omega)}{i(j\omega)} = R_s + \frac{R_1}{R_1 C_1 j\omega + 1} + \dots + \frac{R_n}{R_n C_n j\omega + 1} \quad (3)$$

## 4 Identifikation

Die wesentliche Problemstellung bei der Entwicklung eines realitätsnahen Batteriemodells liegt weniger in der Modellbildung selbst als in der Identifikation. Erfordert die Aufgabenstellung eine realitätsnahe Abbildung aller zeitvarianten und nichtlinearen Effekte kann sich der Aufwand vervielfachen. Aus diesem Grund sollten die Anforderungen an das Modell klar definiert sein.

Im Folgenden wird speziell der Identifikationsprozess für das elektrische Ersatzschaltbild durchgeführt. Auf die Kurve Leerlaufspannung über Ladezustand wird dabei nicht eingegangen (s. hierfür z.B. [4]).

### 4.1 Parameterermittlung im Zeitbereich

In [4] wird eine Ermittlung der Parameter für ein Zellmodell bestehend aus einem seriellen Widerstand und einem RC-Glied durchgeführt. Die Zelle wird am Prüfstand mit einem Stromsprung belastet und die Spannungsantwort gemessen. Aus den Charakteristika der Spannungsantwort wird auf die Modellparameter geschlossen. Der Erfolg dieser Methode ist dabei wesentlich vom Signal-Rausch-Verhältnis abhängig. Bei geringen Anforderungen an die Modellgenauigkeit ist diese Methode durchaus praktikabel. Wird jedoch insbesondere bei dynamischen Belastungen eine höhere Genauigkeit gefordert, zeigt das resultierende Modell Schwächen. Ein weiterer Nachteil dieser Methodik besteht in den schlechten Möglichkeiten zur Bestimmung der Modellstruktur. Bei Identifikationsmethoden im Frequenzbereich lässt sich die Modellstruktur hingegen häufig direkt aus dem Frequenzgang ablesen.

### 4.2 Identifikation im Frequenzbereich

Die Vorgehensweise dieser Identifikationsmethode ist in Abbildung 3 dargestellt. Aus der vorliegenden Aufgabenstellung und den Anforderungen wird die Planung der Messung durchgeführt. Die vorhandenen A-priori-Kenntnisse über die Li-Ionen-Zelle fließen sowohl in diese Planung als auch in die Entwicklung eines theoretischen Modells ein.

Es folgt eine Messung des Frequenzgangs der Lithium-Ionen-Zelle. Dieses Messergebnis wird mit dem Frequenzgang des theoretischen, parametrischen Modells verglichen. Durch Optimierung werden die Parameter des theoretischen Modells variiert, sodass

der Fehler zwischen realem System und theoretischem Modell minimiert wird.

Ist die geforderte Modellgüte erreicht, ist der Prozess nach der Verifikation beendet und das endgültige Modell liegt vor. Erfüllt das Modell die Anforderungen nicht, so ist eine Korrektur des theoretischen Modells vorzunehmen. Über die Größenordnung des Modellfehlers kann dabei häufig ein Rückschluss auf die einzuleitende Maßnahme, wie beispielsweise Steigerung der Modellordnung, gezogen werden. Durch den iterativen Charakter dieses Prozesses erfolgt neben der Identifikation der Parameter gleichzeitig eine Identifikation der Modellstruktur.

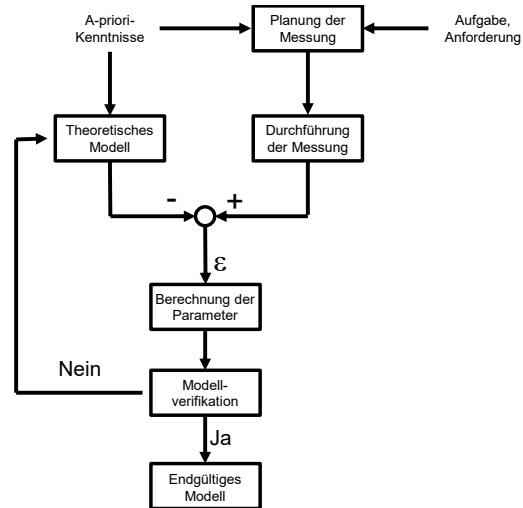


Abbildung 3. Identifikationsprozess nach [5]

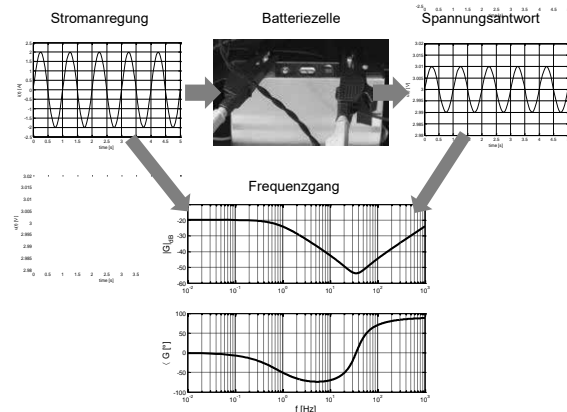
### 4.3 Frequenzgangmessung

Für die Durchführung der Frequenzgangmessungen wird ein elektrochemisches Testsystem der Fa. Solarton herangezogen. Es verfügt über einen Signalgenerator inklusive Verstärker sowie einen Frequenzanalyzer.

Das Gerät verfügt über zwei Modi zur Ermittlung der Impedanz (4).

$$\underline{Z}(j\omega) = \frac{\hat{u} \cdot e^{j\omega t}}{\hat{i} \cdot e^{j(\omega t + \varphi)}} \quad (4)$$

Im potentiostatischen Modus wird die Li-Ionen-Zelle über eine Spannung angeregt und die Stromantwort gemessen. Beim in dieser Arbeit verwendeten galvanostatischen Modus erfolgt die Anregung über den Strom und die Spannungsantwort wird gemessen (vgl. Abb. 4).



**Abbildung 4.** Galvanostatische Impedanzmessung

Aufgrund des geringen elektrischen Widerstands der Zelle und den daraus resultierenden hohen Empfindlichkeiten des Signalanalysators sind geeignete Maßnahmen zur Steigerung des Signal-Rausch-Verhältnisses zu treffen, um zuverlässige Messergebnisse zu generieren. Weitere Maßnahmen zur Steigerung der Qualität der Messergebnisse betreffen die Minimierung der Übergangswiderstände.

Für die Durchführung der Messungen wurde ein Frequenzbereich von 50 mHz bis 2 kHz gewählt. Die Messung erfolgt an jeweils 10 Frequenzstützpunkten pro Dekade mit 10 Wiederholungen.

Da der Ladezustand einen wesentlichen Einfluss auf die Dynamik der Lithium-Ionen-Zelle ausübt, werden die Messungen bei unterschiedlichen Ladezuständen wiederholt. Besonders in den Bereichen extremer Ladezustände, in denen ein besonders starkes nichtlineares Verhalten zu beobachten ist, wurde das Raster der Messungen verfeinert.

#### 4.4 Optimierung

Zur Berechnung der Parameter wird eine entsprechende Optimierungsaufgabe formuliert. Hierzu wird das in [5] beschriebene Verfahren erweitert.

Sämtliche Forderungen werden in einem  $n_z$ -dimensionalen Zielvektor  $\underline{z}(\underline{p})$  (5) und alle freien Parameter im  $n_p$ -dimensionalen Parametervektor  $\underline{p}$  (6) zusammengefasst.

$$\underline{z}(\underline{p}) = [\underline{z}_1(\underline{p}) \quad \underline{z}_2(\underline{p}) \quad \dots \quad \underline{z}_{n_z}(\underline{p})]^T \quad (5)$$

$$\underline{p} = [p_1 \quad p_2 \quad \dots \quad p_{n_p}]^T \quad (6)$$

Zur Begrenzung des Parametervektors auf sinnvolle Werte werden die  $n_p$ -dimensionalen Vektoren  $\underline{p}_{lb}$  und  $\underline{p}_{ub}$  eingeführt, die die Zwangsbedingungen enthalten.

$$\underline{p}_{lb} = [p_{1,lb} \quad p_{2,lb} \quad \dots \quad p_{n_p,lb}]^T \quad (7)$$

$$\underline{p}_{ub} = [p_{1,ub} \quad p_{2,ub} \quad \dots \quad p_{n_p,ub}]^T \quad (8)$$

Die Optimierungsaufgabe besteht darin, die  $n_p$  Komponenten des Parametervektors  $\underline{p}$  innerhalb der Grenzvektoren  $\underline{p}_{lb}$  und  $\underline{p}_{ub}$  so anzupassen, dass für alle  $n_z$  Komponenten des Zielvektors am Ende der Optimierung möglichst minimale Werte erreicht werden.

Das Ziel der Identifikation ist die Minimierung des Modellfehlers. Im Frequenzbereich bedeutet dies eine Minimierung des Fehlers von Amplituden- und Phasengang für alle  $n$  Frequenzstützpunkte.

Nach [5] werden die durch Gewichte  $k_{eval}$  bewerteten quadratischen Fehlerflächen zwischen gemessenem und berechnetem Amplituden- und Phasengang als Zielfunktionen formuliert.

$$\varepsilon_{mag} = \frac{1}{n} \left( \sum_{i=1}^n [20 \log_{10} |G_{Messung}(j\omega_i)| - 20 \log_{10} |G_{Modell}(j\omega_i)|]^2 \cdot k_{eval,i} \right) \quad (9)$$

$$\varepsilon_{phase} = \frac{1}{n} \left( \sum_{i=1}^n \left[ 20 \log_{10} \left( 1 + \left| \angle G_{Messung}(j\omega_i) - \angle G_{Modell}(j\omega_i) \right| \frac{\pi}{180} \right) \right]^2 \cdot k_{eval,i} \right) \quad (10)$$

Über  $k_{eval}$  erfolgt eine Gewichtung bei unterschiedlichen Frequenzen.

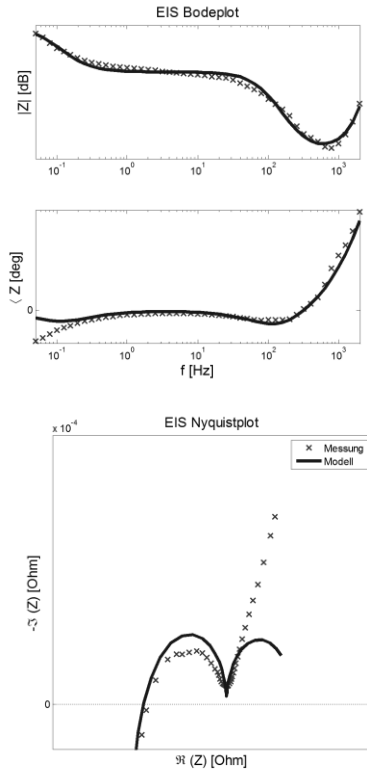
Die Lösung dieser Optimierungsaufgabe erfolgt durch Anwendung des Downhill-Simplex-Algorithmus nach Nelder/Mead (vgl. [6], S. 477) unter Berücksichtigung der definierten Zwangsbedingungen. Der Algorithmus zeichnet sich nach [6] durch seine Einfachheit bei gleichzeitig hoher Robustheit aus.

#### 4.5 Durchführung des Identifikationsverfahrens

Zu Beginn des Identifikationsverfahrens wird die Li-Ionen-Zelle als Ersatzschaltbild mit 2 RC-Gliedern modelliert. Eine Frequenzgangmessung an der realen Li-Ionen-Zelle wird ebenfalls durchgeführt. Die gewichtete quadratische Fehlerfläche der Frequenzkennlinien von Modell und Messung wird ermittelt und mittels Optimierung minimiert.

Das Ergebnis der Parameteroptimierung ist in Abbildung 5 in Form des Bodediagramms (oben) und der Wurzelortskurve (unten, invertierte Imaginärachse) dargestellt. Über einen weiten Frequenzbereich kann das verwendete Modell die Messungen gut approximieren. Gerade jedoch im niederfrequenten Bereich (näherungsweise linearer Anstieg in der komplexen Ebene) zeigt das Modell Schwächen auf. Eine Ver-

besserung der Genauigkeit über den gesamten betrachteten Frequenzbereich ist nur durch Steigerung der Modellordnung also die Änderung der Modellstruktur möglich.



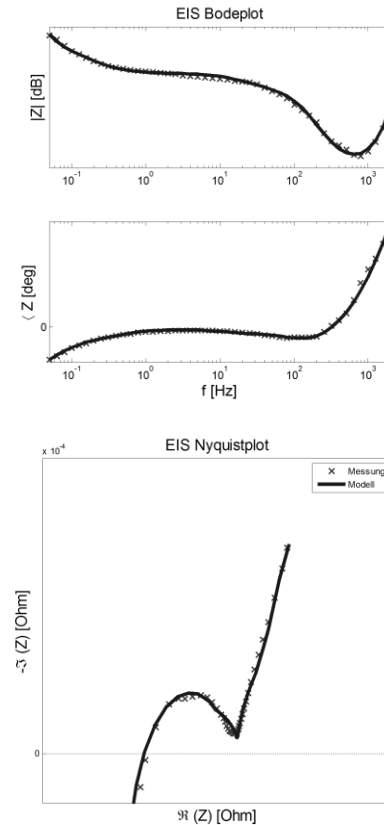
**Abbildung 5.** Identifikation Doppel-RC-Modell

Zwei weitere Iterationsschritte mit einer Änderung der Modellstruktur durch Erhöhung der Modellordnung (Steigerung der Anzahl von RC-Gliedern) folgen, bis ein Modell vorliegt, das die reale Li-Ionen-Zelle über den relevanten Frequenzbereich mit sehr hoher Genauigkeit repräsentieren kann. Das Ergebnis der letzten Iteration ist ein Ersatzschaltbildmodell mit vier RC-Gliedern. In Abbildung 6 erfolgt der Vergleich mit der Frequenzgangmessung.

## 5 Verifikation

Das erstellte Simulationsmodell wird anhand definierter Testzyklen verifiziert. Dafür werden die Simulationsergebnisse den Messwerten einer realen Lithium-Ionen-Zelle gegenübergestellt. Zunächst wird eine geladene, reale Lithium-Ionen-Zelle mit einem Testzyklus belastet und entladen. Der die Zelle belastende Strom wird außerdem als Anregungssignal für das Simulationsmodell verwendet.

Zur Verifikation wird zunächst ein Entladezyklus mit konstantem Strom gewählt. Die Verläufe von simulierter und gemessener Klemmenspannung für ein Anregungssignal durch einen Stromimpuls können nahezu identisch abgebildet werden.



**Abbildung 6.** Identifikation ESB mit 4 RC-Gliedern

Außerdem soll das dynamische Verhalten des Modells validiert werden. Dafür wird als Testzyklus der DST (dynamic stress test) power cycle verwendet. Dieser ist speziell für den Test von Batterien unter Laborbedingungen für den Automobileinsatz entwickelt worden.

Die Spitzenleistung für den Test wird so gewählt, dass der relevante Betriebsbereich durch die Messungen abgedeckt wird. Der Test wird periodisch wiederholt bis die Entladeschlussspannung der Zelle erreicht ist. Abbildung 7 zeigt drei Zyklen dieses Tests bei unterschiedlichen Ladezuständen.

In Abbildung 8 sind der gemessene und simulierte Spannungsverlauf für eine Anregung durch den DST power cycle dargestellt. Es sind dem realen Spannungsverlauf die simulierten Verläufe eines Modells

mit lediglich einem RC-Glied und eines Modells mit vier RC-Gliedern gegenübergestellt.

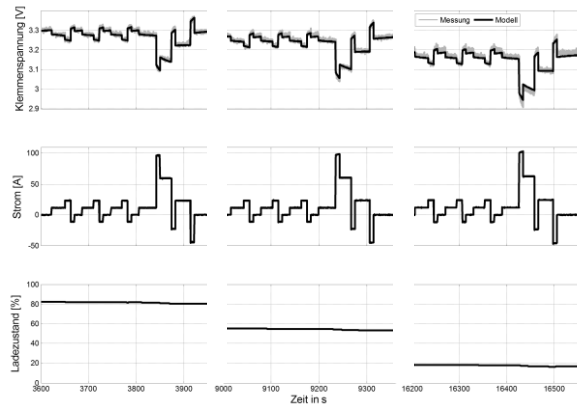


Abbildung 7. Entladung der Li-Ionen-Zelle mittels DST-Zyklus

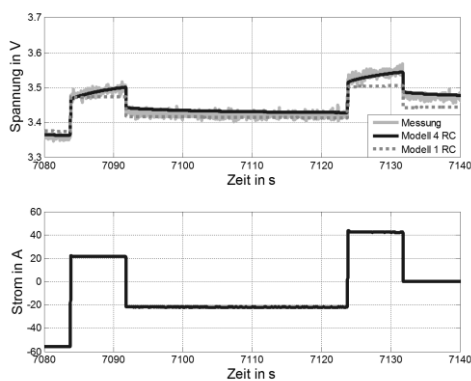


Abbildung 8. Validierung mittels DST-Zyklus

Es ist zu erkennen, dass die Dynamik des gemessenen Spannungsverlaufs infolge einer sprunghaften Änderung des Anregungssignals deutlich besser durch ein Modell mit 4 RC-Gliedern abgebildet werden kann. Das Modell mit einem RC-Glied ist hingegen nicht in der Lage den Spannungsverlauf nachzubilden.

## 6 Zusammenfassung

In diesem Beitrag wurde die Modellbildung und Identifikation einer Lithium-Ionen-Batterie für Elektrofahrzeuge beschrieben. Nach der Vorstellung verschiedener Verfahren zur Modellierung von Batteriezellen, wurde die Modellbildung am Beispiel eines elektrischen Ersatzschaltbildmodells beschrieben. Anschließend wurde ein geeignetes Verfahren zur Parameteridentifikation im Frequenzbereich und die Vorgehensweise bei der Messung des Frequenzgangs erläutert. Zur Berechnung der Parameter aus den gewonnenen Messwerten wurde ein entsprechender

Optimierungsalgorithmus beschrieben. Dabei konnte festgestellt werden, dass zur hinreichend genauen Abbildung des Frequenzgangs der realen Messungen ein Modell höherer Ordnung notwendig ist. Ein sinnvoller Kompromiss zwischen Identifikationsaufwand und Modellgüte stellt für die zugrundeliegende Li-Ionen-Zelle ein Modell 4. Ordnung dar. Dieses Ergebnis hat sich bei der abschließenden Validierung anhand definierter statischer und dynamischer Testzyklen bestätigt.

## 7 Quellen

- [1] M. A. Roscher. *Zustandserkennung von LiFe-PO<sub>4</sub>-Batterien für Hybrid- und Elektrofahrzeuge*. Dissertation, RWTH Aachen, Deutschland, 2011.
- [2] K. M. Tsang, W. L. Chan, Y. K. Wong und L. Sun. *Lithium-ion Battery Models for Computer Simulation*. Proceedings of the 2010 IEEE International Conference on Automation and Logistics, August 16-20, Hong Kong and Macau.
- [3] U. Tröltzsch. *Modellbasierte Zustandsdiagnose von Gerätebatterien*. Dissertation, Universität der Bundeswehr München, Deutschland, 2005.
- [4] F. Quantmeyer, X. Liu-Henke, W. Diehl und S. Bode. *Parametrierung von Batteriesimulationsmodellen mithilfe eines geeigneten Prüfstands*. Tagungsband zum ASIM/GI-Fachgruppentreffen STS/GMMS, in Wolfenbüttel, S. 209-220, 2012.
- [5] X. Liu-Henke. *Mechatronische Entwicklung der aktiven Feder-/Neigetechnik für das Schienenfahrzeug RailCab*. Dissertation, Uni Paderborn, Deutschland, 2005.
- [6] R. Isermann und M. Münchhof. *Identification of Dynamic Systems*. Springer, Deutschland, 2011.

Die vorgestellten Ergebnisse sind im Rahmen des Projekts *EVM* entstanden. Dieses Projekt wird finanziell durch das Bundesministerium für Bildung und Forschung (BMBF) unterstützt. Förderkennzeichen: 17N0911.

## Variantenmanagement in der modellbasierten Produktlinienentwicklung von Fahrzeugsystemen

Hannes Holdschick<sup>1</sup>, Walter Commerell<sup>2</sup>

<sup>1</sup>Daimler AG

<sup>2</sup>Hochschule Ulm

*Hannes.Holdschick@daimler.com*

In den letzten Jahren ist der Umfang softwarebasierter Funktionen stark angestiegen und damit auch die in Software ausgeprägte Variabilität. Ein Lösungsbaustein für die kommenden Herausforderungen liefert eine explizite Variantenmodellierung im Rahmen einer Produktlinienentwicklung. Die im Variantenmodell enthaltenen Merkmale, deren Abhängigkeiten und das durch Regeln formulierte Konfigurationswissen unterstützen bei der Ausprägung einer Variante die vollständige und konsistente Konfiguration. Daher ist unerlässlich, dass das Variantenmodell stets die aktuelle Variabilitätsstruktur des Funktionsmodells abbildet und, sobald sich dieses ändert, einfach und fehlerfrei angepasst werden kann. Konsistenzbedingungen sollen diesen Evolutionsprozess unterstützen, indem fehlerhafte bzw. hinweiswürdige Modellierung im Variantenmodell gekennzeichnet und anschließend bearbeitet werden kann.

### 1 Einleitung

Eine breite Fahrzeugpalette, spezifische Produktanforderungen aus globalen Märkten und eine Vielzahl softwarebasierter Ausstattungsvarianten: In der Automobilindustrie führt die gestiegene Fahrzeug- und Funktionsvielfalt zu einer immer höheren Anzahl von Modellvarianten. So wäre es aus Variantensicht ohne weiteres möglich, dass von den 330000 Einheiten eines Mittelklasse-Fahrzeugs, die im Jahre 2006 gefertigt wurden, keine zwei identisch sind [1]. Darüber hinaus spielt Software in der Fahrzeugindustrie eine immer größere Rolle. So geht man davon aus, dass 90 % aller zukünftigen Innovationen im Fahrzeug von E/E-Systemen inklusive deren Software getrieben werden [2]. Diese Zahlen lassen erahnen, mit welcher Variantenkomplexität wir künftig in den beteiligten Software-Systemen rechnen müssen. Das Konzept der Software-Produktlinien unterstützt die Entwicklung und Beherrschung variantengeprägter, softwarebasierter Systeme [3]. Die funktionalen Eigenschaften dieser Systeme lassen sich durch Merkmale beschreiben [4]. So können gewisse Software-Varianten mehrere Merkmale gemeinsam haben und sich wiederum in anderen Merkmalen unterscheiden. Dabei ist eine Variante durch die Angabe ihrer Merkmale eindeutig bestimmt.

Unser Fokus liegt auf Software-Variabilität in der modellbasierten Funktionsentwicklung. Dabei wird zuerst ein sogenanntes 150 %-Funktionsmodell erstellt, welches die Funktionalitäten aller vorgesehe-

nen Varianten enthält. Durch verschiedene Variabilitätsmechanismen ist es dem Entwickler möglich, daraus das Funktionsmodell für eine gewünschte Variante zu konfigurieren, welches anschließend für die Simulation bzw. die Codegenerierung genutzt werden kann.

Erreicht die Variabilität in einem Funktionsmodell eine hinreichende Komplexität, sollte sie extern dokumentiert werden. In einem merkmalsbasierten Variantenmodell werden einerseits die gemeinsamen und unterschiedlichen funktionalen Eigenschaften der Software-Varianten in einem Merkmalmodell [4] festgehalten. Soll andererseits anhand dieser Merkmale auch die Konfiguration des Funktionsmodells erfolgen, muss zusätzlich das Mapping zwischen diesen beiden Artefakten erstellt werden, welches im sogenannten Konfigurationsmodell abgebildet wird. In Abschnitt 3 werden diese Teilmodelle des Variantenmodells näher erläutert, wobei uns das Modell der Fahrerassistenzsysteme als Beispiel dient.

Ist ein Variantenmodell erstellt, besteht es parallel neben dem Implementierungsartefakt, welches einer ständigen Evolution unterliegt. Dadurch entsteht die Herausforderung, die Konsistenz dieser beiden Modelle zu gewährleisten, welche auch schon in anderen Arbeiten thematisiert wurde [5]. Eine dafür grundlegende Eigenschaft des Variantenmodells ist die Wartbarkeit. Unsere Definition dieser Eigenschaft ist angelehnt an die ISO-Richtlinie 9126 über Softwarequa-

lität [6], welche auch das Kriterium Wartbarkeit beschreibt:

Die **Wartbarkeit eines Variantenmodells** beschreibt dessen Fähigkeit, modifiziert zu werden. Dazu gehören Korrekturen, Verbesserungen oder Anpassungen des Modells an Änderungen der Umgebung.

Aus unserer Sicht sind der Wartbarkeit zwei Aspekte besonders zuträglich: Einerseits muss bekannt sein, welche Änderungen in Funktions- bzw. Simulationsmodellen häufig auftreten, die anschließend im Variantenmodell abgebildet werden müssen. Eine frühere Arbeit [7] zeigt dazu typische Änderungen auf und gibt jeweils eine entsprechende Modellierung für die Anpassung des Variantenmodells vor. Außerdem muss nach einer Änderung des Variantenmodells sichergestellt werden können, dass keine Inkonsistenzen bzw. Fehlmodellierungen aufgetreten sind. Wir haben dafür Konsistenzbedingungen definiert, die in Abschnitt 4 beschrieben werden.

## 2 Variabilität im Simulationsmodell

Wie bereits erwähnt, wird bei dem Variantenmanagement in der Entwicklung der Fahrerassistenzsysteme der 150%-Ansatz verfolgt. Das bedeutet, dass zunächst die gesamte Funktionalität aller Varianten bereitgehalten wird und die Konfiguration darin besteht, an jedem Variationspunkt ein Modul aus mehreren auszuwählen, um die gewünschte Variante des Systems zu erzeugen. Angelehnt an bestehende Definitionen [8] stellt ein **Variationspunkt** für uns eine Stelle im Modell dar, an der die Auswahl einer oder mehrerer Ausprägungen für unterschiedliche Produktvarianten möglich ist. In diesem Abschnitt soll beschrieben werden, welche in der Literatur [9] dargelegten Mechanismen im Umfeld der Fahrerassistenzsysteme oft genutzt werden, um einen Variationspunkt in Matlab / Simulink technisch zu realisieren.

### 2.1 Enabled Subsystem

Mit dem *Enabled Subsystem*-Block kann man ein Subsystem aktivieren bzw. deaktivieren. Durch ein Kontrollsignal wird entschieden, ob die Funktionalität des Subsystems ausgeführt wird. Dieser Mechanismus eignet sich daher sehr gut, um eine optionale Auswahl einer Funktion zu modellieren. Grundsätzlich ist es auch möglich, mit dem *Enabled Subsystem*-Block die Auswahlmöglichkeit zwischen alternativen Funktionsmodulen abzubilden. Dazu müssen jedoch mehrere *Enabled Subsystem*-Blöcke verschachtelt werden, was die Modellierung unnötig unübersicht-

lich macht. Ein großer Vorteil des *Enabled Subsystem*-Mechanismus ist dagegen, dass man damit durch die entsprechende Konfiguration des Kontrollblocks alle Bindungszeiten abdecken kann. Die Variabilität kann somit entweder bereits zur Erstellung des Simulink-Modells gebunden werden oder auch zu einem späteren Zeitpunkt, beispielsweise zum Start-up des Steuergerätes. In Abb. 1 ist ein *Enabled Subsystem*-Block abgebildet, der durch einen *Constant*-Block (*VAR\_Schalter*) gesteuert wird.

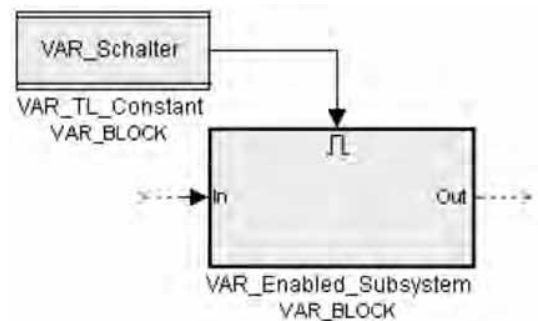


Abbildung 1: Ein *Enabled Subsystem*-Block

### 2.2 Model Load Save

*Model Load Save* ist ein Daimler-intern entwickelter Mechanismus, der es erlaubt, Funktionsmodule in ein Simulink-Modell einzubinden. Dafür müssen diese Funktionsmodule extern als \*.mdl-Dateien zur Verfügung stehen und werden in der Ordnerstruktur verwaltet. Welche Module für welche Variante des Gesamtsystems nötig sind, wird in einer Konfigurationstabelle dokumentiert. Aus dieser Tabelle wird eine Konfigurationsdatei generiert, die auf Basis der Modulnamen die entsprechenden Subsysteme in den Modellrahmen einfügt. Daher ist *Model Load Save* sehr gut für die Modellierung einer alternativen Auswahl an Funktionalitäten geeignet. Mit dem Mechanismus lässt sich auch eine optionale Auswahl abbilden. Dies ist jedoch nicht so komfortabel wie z.B. mit einem *Enabled Subsystem*-Block, da man dabei auch für den Fall, dass die betroffene Funktionalität deaktiviert werden soll, ein eigenes Subsystem erstellen muss. Dadurch, dass *Model Load Save* mit externen Funktionsmodulen arbeitet, ist eine Bindung der Variabilität nur zur Modellerstellung möglich.

### 2.3 Switch bzw. Multiport-Switch Blöcke

*Switch*- bzw. *Multiport-Switch*-Blöcke eignen sich besonders für die Modellierung von Variabilität auf Signalebene. Beide Blocktypen wählen aus mehreren Eingangssignalen eine Variante aus und geben diese

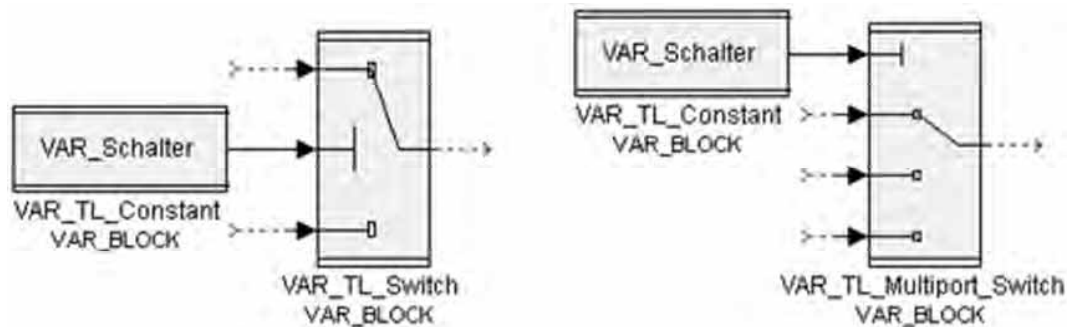


Abbildung 2: Ein Switch- bzw. Multiport-Switch-Block

als Ausgangssignal wieder aus. Die Auswahl des gewünschten Eingangssignals wird mithilfe des Kontrollsignals gesteuert, welches dafür mit einem konfigurierbaren Schwellwert verglichen wird. Der *Switch*-Block wählt dabei aus zwei Eingangssignalen aus, wohingegen bei dem *Multiport-Switch*-Block die Anzahl der Eingangssignale konfiguriert werden kann. Dadurch eignen sich beide Blocktypen besonders gut für die Modellierung einer Auswahl an alternativen Signalen. In Abb. 2 ist auf der linken Seite ein *Switch*-Block und auf der rechten Seite ein *Multiport-Switch*-Block dargestellt, die jeweils über einen *Constant*-Block gesteuert werden.

## 2.4 Logische Gatter

Eine weitere Möglichkeit, Variabilität auf Signalebene abzubilden, sind logische Gatter. Durch einen *Logical Operator*-Block wird dabei aus mehreren Eingangssignalen ein Ausgangssignal ermittelt. Dafür werden die Eingangssignale als Wahrheitswerte (*true* bzw. *false*) interpretiert und daraus das Ausgangssignal entsprechend des jeweiligen Operators (z.B. AND oder OR) errechnet. Interpretiert man eines der Eingangssignale als Kontrollsignal, so lassen sich diese Blöcke als Variabilitätsmechanismen verwenden. Liegt beispielsweise auf dem Kontrollsignal eines logischen AND-Blocks der Wert 0, welcher als *false* interpretiert wird, so haben die übrigen Eingangssignale keinen Einfluss auf den Output. Ist der Wert dagegen ungleich Null, was als *true* interpretiert wird, so hängt das Ausgangssignal tatsächlich von den weiteren Eingangssignalen ab.

## 2.5 Variant Subsystem

Dieser Mechanismus eignet sich für die Auswahl zwischen alternativen Blockvarianten. Innerhalb des *Variant Subsystem*-Blocks werden die Alternativen

als Subsysteme modelliert. Über ein variantenspezifisches Datenobjekt, das sogenannte *Variant Object*, entscheidet sich, welche Modellvariante ausgewählt wird. Dabei ist jedem *Variant Object* eine Bedingung zugeordnet. Ist diese Bedingung erfüllt, so wird die zugehörige Modellvariante aktiviert. Bei der Modellierung muss beachtet werden, dass die Schnittstellen zwischen dem *Variant Subsystem*-Block und seinen Varianten identisch sein müssen.

## 3 Variantenmodellierung

In diesem Abschnitt wird näher auf das Merkmalmodell und das Konfigurationsmodell eingegangen. Als Beispiel dient das Variantenmodell der Fahrerassistenzsysteme.

### 3.1 Das Merkmalmodell

Im Merkmalmodell sind die funktionalen Eigenschaften des Systems hierarchisch als Baum strukturiert [4]. Dabei ist jedem Merkmal ein Variationstyp zugeordnet, welcher die hierarchische Beziehung zu dessen Vatermerkmal beschreibt. Wir unterscheiden in Anlehnung an bestehende Ansätze [10, 11] die folgenden Variationstypen:

- *Obligatorisch*: Ein obligatorisches Merkmal muss genau dann ausgewählt werden, falls dessen Vatermerkmal gewählt wurde. In Abb. 3 ist ein Merkmalmodell für Fahrerassistenzsysteme dargestellt. Darin sind beispielsweise die Merkmale *Limiter* und *Radarsysteme* obligatorisch und werden gekennzeichnet durch ein Ausrufezeichen vor dem „M“, welches für Merkmal steht.
- *Optional*: Ein optionales Merkmal kann ausgewählt werden, falls dessen Vatermerkmal ausgewählt wurde. In Abb. 3 sind beispiels-

weise die Merkmale *Tempomat* und *Bremsassistent* optional und werden daher durch ein Fragezeichen gekennzeichnet.

- *Alternativ*: Aus einer Gruppe alternativer Merkmale muss genau eines gewählt werden, falls deren Vatermerkmal ausgewählt wurde. In Abb. 3 bilden die Merkmale *einstufiger\_Tempomathebel* und *zweistufiger\_Tempomathebel* die einzige Alternativmerkmalgruppe. Alternative Merkmale werden in diesen Modellen durch einen Doppelpfeil gekennzeichnet.
- *Oder*: Aus einer Gruppe von Oder-Merkmalen muss mindestens eines gewählt werden, falls deren Vatermerkmal ausgewählt wurde. Oder-Merkmale werden durch ein „X“ gekennzeichnet. Dieser Variations-typ wird selten verwendet und ist daher auch nicht Teil unseres Beispiels.



Abbildung 3: Ausschnitt des Merkmalmodells der Fahrerassistenzsysteme

Das Wurzelement des Merkmalmodells, in unserem Beispiel in Abb. 3 das Merkmal *Fahrerassistenzsysteme*, muss dabei grundsätzlich ausgewählt werden. Die Eltern-Kind-Beziehungen können formal als sogenannte hierarchische Constraints ausgedrückt werden.

Zusätzlich schränken auch die Cross-Tree-Constraints, wie z.B. Implikation oder gegenseitiger Ausschluss, die möglichen Varianten ein. Zum Beispiel wird im Merkmalmodell in Abb. 3 durch den Constraint „*Bremsassistent requires Nahbereichsradar*“ ausgedrückt, dass man für die Wahl des Bremsassistenten auch den Nahbereichsradar benötigt.

Der Variantenraum definiert die Menge aller möglichen Varianten, ohne die eben beschriebenen Constraints zu berücksichtigen:

Sei  $n$  die Anzahl der Merkmale im Merkmalmodell, so nennen wir  $V = \{(v_1, \dots, v_n) : v_i \in \{true, false\}\}$  den **Variantenraum**.

Die oben erwähnten Constraints beschränken diesen Variantenraum  $V$ . Durch sie wird eine *gültige* Merkmalauswahl definiert: Wir nennen  $v \in V$  eine **Variantenkonfiguration**, falls für  $v$  alle hierarchischen und alle Cross-Tree-Constraints erfüllt sind. Legt man das Merkmalmodell aus Abb. 3 zugrunde, so ist weder eine Merkmalauswahl ohne das obligatorische Merkmal *Limiter* eine Variantenkonfiguration, noch wäre eine Auswahl gültig, die das Merkmal *Bremsassistent* ohne das Merkmal *Nahbereichsradar* beinhaltet. Die Menge aller Variantenkonfigurationen für ein gegebenes Merkmalmodell wird in der Literatur auch als die *Semantik des Merkmalmodells* bezeichnet [12].

### 3.2 Das Mapping

Das Mapping stellt die Verbindung zwischen dem Funktionsmodell und dem Merkmalmodell dar und dient dazu, eine Merkmalauswahl zur Konfiguration einer Systemvariante zu nutzen. Dieser Zusammenhang ist im Konfigurationsmodell dokumentiert. Für jeden Variationspunkt aus dem Funktionsmodell enthält es die Konfigurationslogik. In Abb. 4 sieht man einen Ausschnitt eines Konfigurationsmodells, welcher die Variationspunkte *VAR\_Bremsassistent* und *VAR\_Tempomat* enthält. Der Konfigurationsprozess sieht vor, dass eine Systemvariante spezifiziert wird, indem man eine gültige Merkmalmenge, also eine Variantenkonfiguration, definiert.

Die Konfigurationsaufgabe besteht nun darin, aus einer solchen Variantenkonfiguration die Menge jener Variationspunktausprägungen zu ermitteln, die dafür sorgen, dass das Funktionsmodell mit diesen Ausprägungen die spezifizierte Funktionalität aufweist. Jede Ausprägung steht dabei für eine Möglichkeit, den Variationspunkt zu binden. Nehmen wir beispielsweise an, dass der für den Bremsassistent verantwortliche Variationspunkt im Simulink durch ein Enabled-Subsystem Block realisiert wurde, so existiert dafür im Konfigurationsmodell eine Ausprägung für den Fall, dass das entsprechende Subsystem im Modell aktiviert wird (im Bsp. „1 (Vollmodul)“) und eine Ausprägung für den Fall, dass es deaktiviert wird (im Bsp. „0 (Leermodul)“).



Variation Point	Default Assignment/Assignment
<div> <div> </div> <div> </div> <div> </div> </div> <b>VAR_Bremsassistent</b>	
<div> </div> 0 (Leermodul)	NOT(Bremsassistent)
<div> </div> 1 (Vollmodul)	Bremsassistent
<div> <div> </div> <div> </div> <div> </div> </div> <b>VAR_Tempomat</b>	
<div> </div> 0 (Leermodul)	NOT(Tempomat)
<div> </div> 1 (einstufiger Tempomathebel)	einstufiger_Tempomathebel
<div> </div> 2 (zweistufiger Tempomathebel)	zweistufiger_Tempomathebel

Abbildung 4: Ausschnitt des Konfigurationsmodells der Fahrerassistenzsysteme

Die Konfigurationslogik, also das Wissen, welche Merkmalauswahl zu welchen Ausprägungen an den Variationspunkten führt, ist in den Konfigurationsregeln formuliert:

Eine **Konfigurationsregel** ist eine boolesche Formel, deren Literale aus Merkmalen des Merkmalmodells bestehen.

Jeder Ausprägung ist genau eine Konfigurationsregel zugeordnet. In dem Konfigurationsmodell in Abb. 4 hat am Variationspunkt des Bremsassistenten die Ausprägung für das Leermodul die Konfigurationsregel *NOT(Bremsassistent)* und die Ausprägung des Vollmoduls die Konfigurationsregel *Bremsassistent*. Liegt eine Variantenkonfiguration  $v \in V$  vor, lässt sich anhand der Konfigurationsregeln für jeden Variationspunkt die gewünschte Ausprägung bestimmen, indem man den Wahrheitswert der Konfigurationsregeln auswertet. Dabei werden gewählte Merkmale in der Formel durch *true* und nicht gewählte Merkmale durch *false* ersetzt. An jedem Variationspunkt wird dann die Ausprägung gültig, deren Konfigurationsregel zu *true* evaluiert.

Jeder Variationspunkt kann beliebig viele Ausprägungen haben und zu jeder Ausprägung gehört genau eine Konfigurationsregel. Es ist daher wichtig, dass die Konfigurationsregeln so formuliert sind, dass an jedem Variationspunkt für jede theoretisch mögliche Merkmalauswahl immer nur genau eine dieser Regeln wahr werden kann. Dadurch ist sichergestellt, dass bei der Konfiguration für jeden Variationspunkt genau eine Ausprägung gültig wird. Diese Eigenschaft kann durch Konsistenzbedingungen gesichert werden, wie wir im nächsten Abschnitt sehen werden.

## 4 Die Konsistenzbedingungen

In diesem Abschnitt werden Konsistenzbedingungen für das Variantenmodell vorgestellt. Sie verfolgen für uns zwei Ziele: Einerseits soll es möglich sein, ein im Entwicklungsprozess befindliches Variantenmodell nach Inkonsistenzen bzw. bemerkenswerten Zuständen zu durchsuchen und diese anzuzeigen. Andererseits wird der Änderungsprozess dieses Modells abgesichert, indem man nach der Durchführung einer Änderung ermitteln kann, ob dabei Fehlmodellierungen oder ein inkonsistenter Modellzustand aufgetreten sind.

### 4.1 Konsistenzbedingung 1

Diese Konsistenzbedingung soll dafür sorgen, dass der Teil der Konfiguration, bei dem die Konfigurationsregeln an den Variationspunkten ausgewertet werden, reibungslos abläuft. Dafür muss folgende Eigenschaft sichergestellt werden:

*An jedem Variationspunkt muss für jede theoretisch mögliche Variantenkonfiguration genau eine Konfigurationsregel wahr werden.*

Sollte eine Merkmalauswahl existieren, für die an einem Variationspunkt keine oder mehr als eine Konfigurationsregel wahr wird, so kann in dieser Situation keine Ausprägung ermittelt werden. Das kann zur Folge haben, dass das Zielartefakt nicht vollständig konfiguriert werden kann, sodass unter Umständen eine inkonsistente bzw. syntaktisch fehlerhafte Artefaktvariante entsteht.

Diese Bedingung wird in zwei Schritten geprüft. Zuerst wird ermittelt, ob ein Variationspunkt existiert, an dem für eine bestimmte Variantenkonfiguration keine Konfigurationsregel wahr wird. Dafür muss die

☐ QP VAR_Tempomat	
⓪ 1 (einstufiger Tempomathebel)	Tempomat AND einstufiger_Tempomathebel
⓪ 2 (zweistufiger Tempomathebel)	Tempomat AND zweistufiger_Tempomathebel

Abbildung 5: Dieser Variationspunkt verletzt die Konsistenzbedingung 1

folgende Formel für jeden Variationspunkt einzeln auf Erfüllbarkeit geprüft werden:

$$MM \wedge \bigwedge_{i=1}^N (\neg KR_i)$$

Dabei bezeichnet  $MM$  die aussagenlogische Repräsentation des Merkmalmodells und die  $KR_i$  durchlaufen die Konfigurationsregeln des gerade untersuchten Variationspunktes. Ist der entstandene Ausdruck lösbar, existiert für diesen Variationspunkt eine Merkmalbelegung, sodass keine Konfigurationsregel wahr wird.

Wäre der Variationspunkt  $VAR\_Tempomat$  aus Abb. 5 Teil unseres Konfigurationsmodells, so würde er die beschriebene Konsistenzbedingung verletzen: wählt man das optionale Merkmal  $Tempomat$  nicht aus, so wird der Ausdruck in beiden Konfigurationsregeln durch *false* ersetzt und keine der beiden Regeln evaluiert zu *true*. Dadurch kann an diesem Variationspunkt nicht entschieden werden, welche Ausprägung ausgewählt wird.

Im zweiten Schritt wird geprüft, ob ein Variationspunkt existiert, an dem für eine bestimmte Variantenkombination mehr als eine Konfigurationsregel wahr wird. Dabei muss für jeden Variationspunkt und für jedes Paar von Konfigurationsregeln  $KR_i$  und  $KR_j$  mit  $i \neq j$  an diesem Variationspunkt die folgende Formel auf Erfüllbarkeit geprüft werden:

$$MM \wedge KR_i \wedge KR_j$$

Ist einer dieser Ausdrücke erfüllbar, so existieren zwei Konfigurationsregeln, die für eine bestimmte Merkmalauswahl gleichzeitig wahr werden.

Existiert nun ein Variationspunkt, für den entweder die erste oder die zweite Prüfung anschlägt, so verletzt dieser die Konsistenzbedingung und muss überarbeitet werden, damit die Konfiguration wieder reibungslos ablaufen kann.

#### 4.2 Konsistenzbedingung 2

Die folgende Konsistenzbedingung soll für verständlichere Konfigurationsregeln sorgen:

*Konfigurationsregeln enthalten keine obligatorischen Merkmale.*

Ein obligatorisches Merkmal muss für jede Variante ausgewählt werden, wodurch im Konfigurationsprozess dessen Bezeichner in der Konfigurationsregel stets als *true* ausgewertet wird. Das Merkmal hat somit keinen Einfluss auf die Konfiguration und kann daher in der Konfigurationsregel auch vernachlässigt werden. Die Lesbarkeit und Verständlichkeit der Konfigurationsregeln soll somit erhöht werden, da nur noch für die Konfiguration relevante Merkmale darin auftreten. In bestimmten Ausnahmefällen kann eine solche Modellierung durchaus Sinn machen, etwa wenn bereits abzusehen ist, dass in naher Zukunft das betroffene Merkmal variabel wird. Ein obligatorisches Merkmal in einer Konfigurationsregel ist daher weniger ein Modellierungsfehler, sondern viel mehr ein Modellzustand, auf den wir den Entwickler hinweisen wollen, sodass er ggf. Anpassungen vornehmen kann.

Um diese Konsistenzbedingung zu prüfen, müssen alle Konfigurationsregeln dahingehend untersucht werden, welchen Variationstyp die in den Regeln auftretenden Merkmale aufweisen.

#### 4.3 Konsistenzbedingung 3

Diese Konsistenzbedingung soll für aussagekräftiges Konfigurationswissen sorgen:

*Keine Konfigurationsregel ist vor dem Hintergrund der Merkmalmodellierung eine Tautologie bzw. eine Kontradiktion.*

Diese Modellierung führt dazu, dass die Ausprägung der betroffenen Konfigurationsregel entweder immer oder nie gültig wird. Der Entwickler sollte daher die Möglichkeit haben, solche Modellzustände zu finden, um die entsprechende Konfigurationsregel anzupassen.

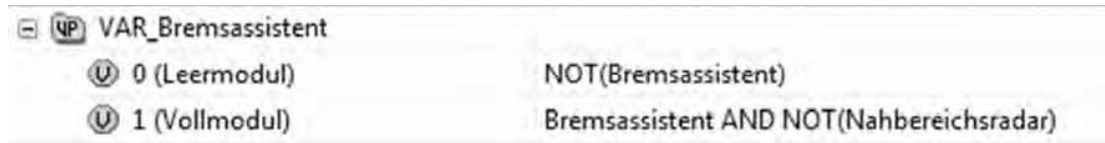


Abbildung 6: Dieser Variationspunkt verletzt Konsistenzbedingung 3

Die Prüfung dieser Konsistenzbedingung läuft in zwei Schritten ab. Zunächst wird geprüft, ob sich unter den Konfigurationsregeln eine Tautologie bzgl. der Merkmalmodellierung befindet. Dafür muss für jede Konfigurationsregel  $KR$  die folgende Formel auf Lösbarkeit getestet werden, wobei  $MM$  wieder die aussagenlogische Formel des Merkmalmodells repräsentiert:

$$MM \wedge \neg(KR)$$

Findet man für diesen Ausdruck keine Lösung, so existiert für die entsprechende Konfigurationsregel keine Merkmalbelegung, für die ihre Negation wahr wird. Somit wird diese Regel für keine Belegung falsch, ist also eine Tautologie vor dem Hintergrund der Merkmalmodellierung.

Im zweiten Schritt wird nach Widersprüchen bzgl. der Merkmalmodellierung gesucht. Dafür muss für jede Konfigurationsregel  $KR$  die folgende Formel auf Lösbarkeit getestet werden:

$$MM \wedge KR$$

Ist dieser Ausdruck nicht lösbar, so existiert für die entsprechende Konfigurationsregel keine Merkmalbelegung, für die sie wahr wird. Somit ist diese Regel ein Widerspruch zum Merkmalmodell.

Der Variationspunkt *VAR\_Bremsassistent* in Abb. 6 verletzt die beschriebene Konsistenzbedingung. Da im Merkmalmodell (s. Abb. 3) durch einen Cross-Tree-Constraint ausgedrückt wird, dass man den *Bremsassistent* nur zusammen mit dem Merkmal *Nahbereichsradar* auswählen kann, stellt die Konfigurationsregel für die Ausprägung *1 (Vollmodul)* einen Widerspruch zur Merkmalmodellierung dar. Unabhängig von der Merkmalauswahl kann diese Konfigurationsregel somit nie als *true* ausgewertet werden. Bedenkt man dabei, dass diese Inkonsistenz nur durch Kenntnis des kompletten Merkmalmodells identifiziert werden kann, so ist eine manuelle Suche bei großen Modellen kaum vorstellbar.

## 5 Related Work

Es existieren viele Ansätze, die Analysen des Merkmalmodells beschreiben, um generelle Eigenschaften oder auch Inkonsistenzen zu ermitteln. Benavides et al. [13] haben verschiedene Arbeiten in diese Richtung zusammengefasst. Es finden sich dort Analysen, die ein Merkmalmodell z.B. auf die Anzahl bzw. die Existenz gültiger Belegungen prüfen. Eine weitere Arbeit [14] beschreibt explizit Fehlmodellierungen in Merkmalmodellen. Diese werden in drei verschiedene Klassen eingeteilt. Als *Redundanzen* werden mehrfach formulierte Variabilitätsinformationen bezeichnet, welche z. B. bei der Kombination von hierarchischen und Cross-Tree-Constraints entstehen können. Die Klasse der *Anomalien* fasst alle Fälle zusammen, in denen die Domäne widersprüchlich modelliert wurde. Muss etwa ein optionales Merkmal durch einen Cross-Tree-Constraint stets gewählt werden, so bildet das Modell nicht die eigentliche Variabilität der Funktionalität ab, die an dem betroffenen Merkmal hängt. Außerdem wird noch auf *Inkonsistenzen* eingegangen. Damit sind Fehlmodellierungen gemeint, wie z. B. ein Cross-Tree-Constraint, der den Widerspruch zwischen zwei obligatorischen Merkmalen ausdrückt. Solche Inkonsistenzen führen in der Regel dazu, dass das Merkmalmodell nicht mehr lösbar ist und dadurch keine Variante konfiguriert werden kann.

Durch den Fokus auf das Merkmalmodell sind die präsentierten Analysen für uns nur teilweise relevant. Viele unserer Konsistenzbedingungen beziehen sich auf den Zusammenhang zwischen Merkmalen und Konfigurationswissen, worauf in beiden Arbeiten nicht eingegangen wird.

## 6 Zusammenfassung

Die Konsistenz zwischen dem Variantenmodell und der tatsächlichen Variabilität in der Implementierung ist im industriellen Umfeld eine große Herausforderung. Ein initial erstelltes Variantenmodell wird nach seiner Integration in den Entwicklungsprozess Teil der Evolution des Funktionsmodells und muss daher wartbare Strukturen aufweisen, die Anpassungen erleichtern. Konsistenzbedingungen der hier be-

schriebenen Art bieten uns eine Möglichkeit, diese Strukturen herzustellen und zu erhalten. Wir können dadurch Variantenmodelle vor deren Integration in den Entwicklungsprozess auf Inkonsistenzen untersuchen. Außerdem wird der Prozess einer Änderung des Modells um die Prüfung der Konsistenzbedingungen erweitert, um Fehlmodellierungen bzw. ungewollte Modellzustände zu vermeiden. Wir haben hier einen Teil unserer Bedingungen zur Sicherung der Konsistenz des Variantenmodells vorgestellt, die sich auf das Merkmal- und Konfigurationsmodell, aber vor allem auch auf die Interaktion dieser beiden Modelle beziehen.

In unserer zukünftigen Arbeit werden wir die Sammlung an Konsistenzbedingungen für das Variantenmodell weiter ausbauen. Die gesammelten Erfahrungen sollen die Grundlage für Modellierungsrichtlinien sein, die zu einem wartbaren Variantenmodell führen. Um die Arbeit der Entwickler weiter zu erleichtern, arbeiten wir momentan daran, die Konsistenzprüfungen des Variantenmodells in unseren Entwicklungsprozess und damit auch in unsere Werkzeugkette zu integrieren. Dadurch wird eine automatische Prüfung der Konsistenzbedingungen möglich sein.

## 7 References

- [1] J. Weiland. *Variantenkonfiguration Eingebetteter Automotive Software mit Simulink*. Dissertation, Universität Leipzig, Deutschland, 2008.
- [2] M. v. d. Beeck. *Development of logical and technical architectures for automotive systems*. In: *Software and Systems Modeling Vol 6, No 2*, S. 205-219, 2007.
- [3] P. C. Clements und L. Northrop. *Software Product Lines: Practices and Patterns*. SEI Series in Software Engineering. Addison-Wesley, August 2001.
- [4] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak und A. S. Peterson. *Feature-oriented Domain Analysis*. Technischer Bericht. Carnegie-Mellon University Software Engineering Institute, November 1990.
- [5] R. Lotufo, S. She, T. Berger, K. Czarnecki und A. Wąsowski. *Evolution of the Linux Kernel Variability Model*. In: *Proceedings of the 14th international Conference on Software Product Lines: going beyond*. S. 136-150, Springer-Verlag, 2010.
- [6] ISO/IEC. *ISO/IEC 9126. Software engineering – Product quality*. ISO/IEC, 2001.
- [7] H. Holschick. *Challenges in the Evolution of Model-Based Software Product Lines in the Automotive Domain*. In: *Proceedings of the 4th International Workshop on Feature-Oriented Software Development (FOSD)*, S. 70–73, USA, 2012.
- [8] G. Böckle, P. Knauber, K. Pohl und K. Schmid. *Software-Produktlinien: Methoden, Einführung und Praxis*. Dpunkt Verlag, 2004.
- [9] B. Gutekunst und J. Weiland. *Handhabung von Varianz in Simulink aus funktionsorientierter Sicht*. Informatik 2011 – 9. Workshop Automotive Software Engineering, Berlin, 2011.
- [10] K. Czarnecki und U. Eisenecker. *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley, USA, 2000.
- [11] D. S. Batory. *Feature Models, Grammars, and propositional Formulas*. SPLC. Volume 3714 of *Lecture Notes in Computer Science*, S. 7-20. Springer, 2005.
- [12] P. Borba. *An Introduction to Software Product Line Refactoring*. GTTSE. Volume 6491 of *Lecture Notes in Computer Science*, S. 1-26. Springer, 2009.
- [13] D. Benavides, S. Segura und A. Ruiz-Cortés. *Automated Analysis of Feature Models 20 Years Later: a Literature Review*. *Information Systems*, 35(6), 2010.
- [14] T. von der Maßen und H. Lichter. *Deficiencies in Feature Models*. In: *Proceedings of SPLC'04 Workshop on Software Variability Management for Product Derivation – Towards Tool Support*, 2004.

# Thermische Modellierung eines doppelstöckigen Wagenkastens für Next Generation Train.

Ivan Windemut, Mathias Böhm  
Deutsches Zentrum für Luft- und Raumfahrt  
Institut für Fahrzeugkonzepte  
*ivan.windemut@dlr.de*

Die wachsenden Komfortansprüche der Reisenden, die aufgrund des Klimawandels immer extremer werdenden Witterungsbedingungen sowie der grenzüberschreitende Schienenverkehr über verschiedene Klimazonen erfordern einen erhöhten Energieaufwand für die Klimatisierung der Fahrgasträume und der Führerstände. Dem gegenüber steht die Forderung der Betreiber und der Politik nach energieeffizienteren Schienenfahrzeugen. Darüber hinaus müssen thermische Eigenschaften von neuartigen Materialien, die durch den angestrebten Leichtbau vermehrt in Schienenfahrzeugen Anwendung finden, untersucht werden.

Zur Ermittlung äußerer und innerer Einflussfaktoren, wie Wärme- und Feuchtigkeitslasten, wurde mittels der Human Comfort Bibliothek der XRG Simulation GmbH ein thermisches Wagenkastenmodell entwickelt, welches neben der Geometrie, den verwendeten Materialien und den bahnspezifischen Normen und Vorschriften das Wohlbefinden der Fahrgäste berücksichtigt. Das thermische Wagenkastenmodell wurde in einen virtuellen Prüfstand, der mit Hilfe der auf der DLR-Bibliothek Alternative Vehicles [1] basierenden Railway Vehicles Bibliothek aufgebaut ist, integriert. Durch die Kopplung der thermischen Modelle mit den Schienenfahrzeugmodellen werden strecken-, geschwindigkeits- und wetterabhängige Simulationen ermöglicht.

In Anbetracht der Herausforderungen, die heute an moderne Schienenfahrzeuge gestellt werden, wurden mehrere Situationen wie z.B. Fahrt an einem sonnigen Sommertag, Fahrt in einer eisigen Winternacht sowie Fahrten mit Voll-, Teil- und Nullbesetzung untersucht und Heiz- und Kühlleistungsbedarfe ermittelt. Aus den Simulationsergebnissen werden Anforderungen an die Auslegung der elektrischen Bordnetze abgeleitet um somit zu einer sicheren Versorgung der Komforteinrichtungen beizutragen.

## 1 Einleitung

Wachsendes Passagieraufkommen im Personennah- und Fernverkehr und steigende Energiepreise machen es notwendig, effizientere Antriebs- und Energiekonzepte auf die Schiene zu bringen. Daher wird am Institut für Fahrzeugkonzepte des Deutschen Zentrums für Luft- und Raumfahrt e. V. (DLR), in Zusammenarbeit mit acht weiteren Instituten an einem neuartigen Zug für den Hochgeschwindigkeitsverkehr, dem Next Generation Train (NGT), geforscht.

Ziele dieser Forschung sind unter anderem:

- Niedrigerer spezifischer Energieverbrauch (50 %) gegenüber dem ICE3
- Verringerung des Verschleißes und der Lebenszykluskosten
- Höhere betriebliche Fahrgeschwindigkeit von 400 km/h
- Intelligentes Energiemanagement
- Konsequenter Leichtbau unter Einsatz modernster Materialien

Der Ultrahochgeschwindigkeitszug wird für transeuropäische (oder vergleichbar lange) Strecken konzipiert und muss in Sachen Fahrgastkomfort den höchsten Ansprüchen genügen. Da der Energieverbrauch der Klimatisierungsanlagen in diesem Zusammenhang einen nicht unerheblichen Teil des Gesamtenergieverbrauches darstellt (beim ICE3 5-10% [2]), müssen die thermischen Vorgänge im Wagenkasten des Next Generation Trains quantifiziert werden. Dafür werden äußere und innere Einflussfaktoren identifiziert und bei der Modellierung berücksichtigt. Zu solchen Faktoren zählt vor allem die Geometrie des Wagenkastens, die Materialzusammensetzung und Schichtdicke der Wände, klimatische Einsatzbedingungen, Fahrgeschwindigkeit, aber auch die Wärme- und Feuchtigkeitsabgabe der Fahrgäste und der Einrichtungsgegenstände.

Um konkrete Aussagen über den streckenabhängigen Energieverbrauch der Heating, Ventilation and Air Conditioning (HVAC)-Anlagen eines Zuges machen zu können, wurde mit Hilfe von Modelica/Dymola

ein thermisches Wagenkastenmodell aufgebaut, welches die zuvor genannten Einflussfaktoren berücksichtigt. Dieses Modell wurde nachfolgend in ein energetisches Gesamtzugmodell aus der auf der DLR-Bibliothek Alternative Vehicles (AV) basierenden Railway Vehicles Bibliothek (RV) integriert.

Mit Hilfe des energetischen Gesamtzugmodells lassen sich Maßnahmen zur Verbesserung der Energiebedarfswerte und ggf. zur Minderung einiger gravierender Einflussfaktoren identifizieren. Dieses gilt sowohl für Neuentwicklungen wie NGT, als auch für bestehende Schienenfahrzeuge.

## 2 Thermisches Wagenkastenmodell

Das thermische Wagenkastenmodell (TWKM) wurde in der Simulationsumgebung Dymola erstellt. Dieser Simulationssoftware von Dassault Systemes liegt die objektorientierte und gleichungsbasierte Programmiersprache Modelica zugrunde. Für die Simulation werden in einem Modelica-Modell Komponenten basierend auf Differential-, algebraischen und diskreten Gleichungen aufgebaut. Danach werden die Gleichungen aller Komponenten und aller Verbindungen zu einem Gesamtgleichungssystem zusammengefasst. Da die direkte numerische Lösung bezüglich der Rechenzeit in der Regel ineffizient ist, werden Transformationsalgorithmen angewendet, um das Ausgangsgleichungssystem in eine durch ein Standard-Integrationsverfahren lösbare Form zu transformieren [3].

Grundelemente aus verschiedenen Fachdisziplinen wie Thermodynamik, Elektrotechnik oder Mechanik stehen dem Anwender durch eine kostenfreie Bibliothek, die Modelica Standard Library, zur Verfügung. Darüber hinaus existieren kommerzielle Modelica-Bibliotheken, wie z. B. die HumanComfort Library von XRG Simulation GmbH zur Simulation der thermischen Behaglichkeit, die Alternative Vehicles Library vom DLR-Institut für Fahrzeugkonzepte zur Simulation von neuartigen Straßenfahrzeugen oder auf der letzteren basierende Railway Vehicles Library zur Simulation von Schienenfahrzeugen.

### 2.1 Ziele der Modellierung

Zur Abschätzung des Energiebedarfs zur Klimatisierung von unterschiedlichen Mittelwagenkonzepten aber auch zur Identifikation und Quantifizierung von äußeren und inneren Einflussfaktoren wurde das thermische Wagenkastenmodell als Umgebung für die simulationstechnische Untersuchung unterschiedli-

cher Klimaanlageentypen entwickelt. Dieses Modell kann mit wenig Aufwand an unterschiedliche doppelstöckige Züge angepasst werden. Das liegt zum einen an der guten Ausnutzung des bahnspezifischen Lichtraumprofils durch den NGT und den darauf basierenden geometrischen Vereinfachungen im Rahmen der Modellbildung, zum anderen aber an der Integration der Materialdaten als Parametersätze (Records).

Durch die Verwendung der HumanComfort Library (HCL), war eine genaue Nachbildung des mehrschichtigen Wand- und Fensteraufbaus möglich [4]. Die Modelle der HCL sind bereits verifiziert und wurden daher als Referenz herangezogen.

Die große Anzahl der Einflussfaktoren erfordert komplexe Modelle mit mehrstufiger Parametervererbung, was die Simulationszeit verlängert und Maßnahmen zur Minimierung der Rechenzeit notwendig macht. Eine solche Maßnahme stellt die vereinfachte Betrachtung thermodynamischer Vorgänge dar. Als Beispiel kann das Außerachtlassen von Wärmebrücken oder die eindimensionale Betrachtung der Wärmeleitung im Festkörper genannt werden. Solche Vereinfachungen sind im Fall NGT durchaus zulässig, da die aus CFK aufgebaute Wagenhülle schlechter Wärmeleiter ist. Für jedes weitere Wagenkastenkonzept müssen diese aber einzeln geprüft werden.

### 2.2 Modellaufbau

#### 2.2.1 Einflussfaktoren

Die inneren Einflüsse auf einen Fahrgastraum sind die Wärme- und Feuchtigkeitsabgabe der Fahrgäste und der inneren Einrichtung. Diese sind in der HCL implementiert und wurden im Wagenkastenmodell als fertiger Baustein verwendet [4]. Da für die Inneneinrichtung des NGT kein Wärmemanagement notwendig ist, wird das Untermodell auch entsprechend parametrisiert.

Die äußeren Einflüsse sind hauptsächlich durch die DIN EN 13129 - „Luftbehandlung in Schienenfahrzeugen des Fernverkehrs“ festgelegt [5]. In der Norm sind für Europa drei klimatische Zonen definiert, wobei für jede Zone die niedrigste und die höchste Außentemperatur, die relative Luftfeuchtigkeit und die Strahlungsleistung der Sonne vorgegeben sind (vgl. Tabelle 1). Diese Faktoren werden durch das Untermodell „Weather“ aus HCL miteinbezogen. Weitere Außenbedingungen werden dem TWKM von dem virtuellen Prüfstand zur Verfügung gestellt.

Klima- zone	Niedrigste Außen- tempera- tur	Höchste Außen- tempe- ratur	Relative Luft- feuch- tigkeit	Strah- lungslei- stung Son- ne
Zone I	-10 °C	+40 °C	40%	800 W/m <sup>2</sup>
Zone II	-20 °C	+35 °C	50%	700 W/m <sup>2</sup>
Zone III	-40 °C	+28 °C	45%	600 W/m <sup>2</sup>

**Tabelle 1.** Klimazonen in Europa [5]

Zusammenfassend wird die Wärmebilanz (1) des Fahrgastraums in Abbildung 1 dargestellt.

$$\dot{Q}_{zu} = \dot{Q}_{Tr} + \dot{Q}_L + \dot{Q}_P + \dot{Q}_{St} + \dot{Q}_M \quad (1)$$

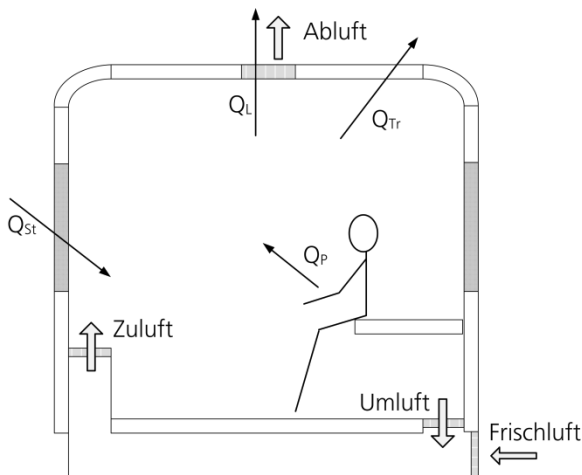
Dabei bedeuten:

$\dot{Q}_{Tr}$  - Transmissionswärmestrom;

$\dot{Q}_L$  - Verlustwärmestrom durch Abluft;

$\dot{Q}_P$  - Wärmestrom einer Person;

$\dot{Q}_{St}$  - Strahlungswärmestrom.



**Abbildung 1.** Wärmebilanz des Fahrgastraums [6]

### 2.2.2 Geometrische Vereinfachung für NGT

Für die thermische Modellierung des NGT ist es ausreichend einen Doppelstockwagenkasten, nicht aber den Gesamtzug zu betrachten. Der Energieverbrauch des Gesamtzuges kann anschließend durch lineare Interpolation bestimmt werden. Durch diese Vereinfachung wird der Rechenaufwand minimiert und der Modellierungsaufwand verringert. In diesem Zusammenhang werden die beiden Stirnseiten des Wagens als adiabate Grenzen angesehen. Es findet also kein Wärmeaustausch zwischen diesen Flächen und der Umgebung statt.

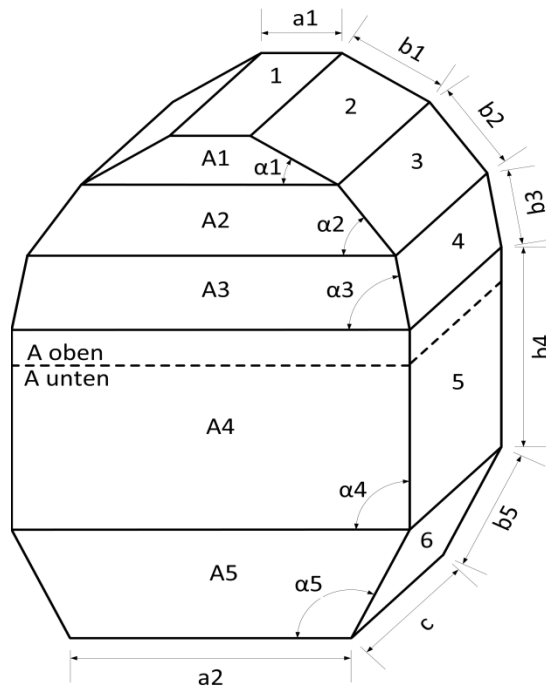
Der Wagenkasten wurde in einen Fahrgastraum und zwei Fahrwerkssegmente aufgeteilt (Abbildung 2).



**Abbildung 2.** Aufteilung des NGT-Wagenkastens

Die Gründe für die Aufteilung sind der unterschiedliche geometrische Aufbau der Teilbereiche, die Position der Türen, die Aussparungen für die Fahrwerke sowie die Verwendung unterschiedlicher Materialien der Fahrzeughülle. Außerdem wurde der Einfluss der Fahrgäste ausschließlich im Fahrgastraum betrachtet.

Der verwendete Querschnitt des Wagenkastens ist in der Abbildung 3 dargestellt.



**Abbildung 3.** Querschnitt des NGT-Wagenkastens

Da mit den Wand- und Fenstermodellen aus HCL nur ebene Flächen dargestellt werden können, die des NGT jedoch gewölbt sind, war es notwendig, Korrekturfaktoren für Flächen und Volumina einzuführen. Die realen Flächeninhalte wurden dazu aus dem CAD-Modell bestimmt.

Mit den Winkeln  $\alpha_1$  bis  $\alpha_5$  wurde die Flächenausrichtung gegenüber dem Horizont dargestellt, welche für die Berechnung der Sonneneinstrahlung maßgeblich ist.

In der Abbildung 4 ist das Modell des Fahrgastraums dargestellt. Das Fahrwerksegment unterscheidet sich von diesem durch fehlende Fahrgäste und vorhandene Türen. Die Abbildung 5 zeigt abschließend das Gesamtmodell.

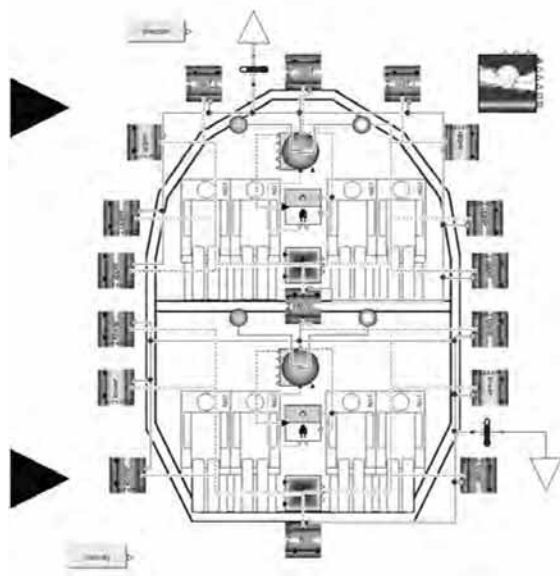


Abbildung 4. Fahrgastraummodell

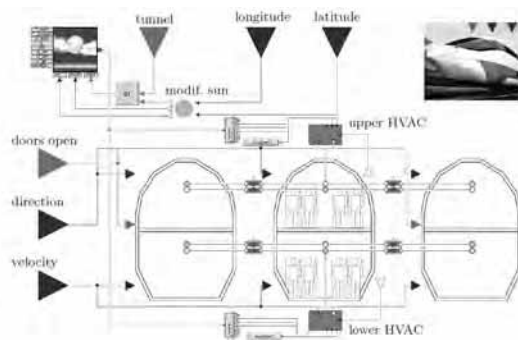


Abbildung 5. Thermisches Wagenkastenmodell NGT

## 2.3 Virtueller Prüfstand

Dem in der Abbildung 6 dargestellten virtuellen Prüfstand für das thermische Wagenkastenmodell liegt das nach NGT-Spezifikationen aus [7] und [8] parametrisierte SimpleTrain2-Modell aus der RV-Bibliothek zugrunde. Das SimpleTrain2-Modell ist ein Zugmodell für die Simulation und Bewertung von Triebzügen mit elektrischen, diesel-elektrischen sowie diesel-hydraulischen Antrieben. Mit Hilfe der SimpleTrain2-Architektur werden am DLR unter anderem energetische Untersuchungen zur Antriebsleistung in Abhängigkeit von fahrzeugcharakteristischen Eigenschaften wie z.B. Fahrzeuggewicht, Roll- und Luftwiderstand durchgeführt.

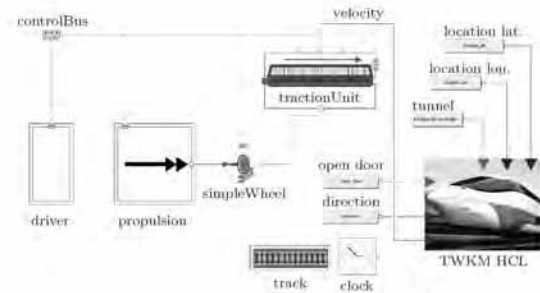


Abbildung 6. TWKM im virtuellen Prüfstand

Durch das Zusammenfügen beider Modelle war es möglich, die wetterabhängige Simulationen streckenbezogen zu gestalten. Dem Wagenkastenmodell werden die Informationen über aktuelle Fahrgeschwindigkeit, Zugausrichtung, Tunnel und Stationen bezogen auf die Zugposition zur Verfügung gestellt. Aus diesen berechnet das Wagenkastenmodell die Konvektionsverluste, die Sonneneinstrahlung sowie die Wärmeverluste durch offene Türen und zieht die Bilanz unter Berücksichtigung der Fahrgastraumerwärmung durch Fahrgäste. Die Ermittlung der Sonneneinstrahlungsleistung erfolgt aus den geografischen Zugkoordinaten und der Tageszeit mit Hilfe eines modifizierten SUN-Modells aus der HCL. Daher ist die Angabe der Startzeit notwendig.

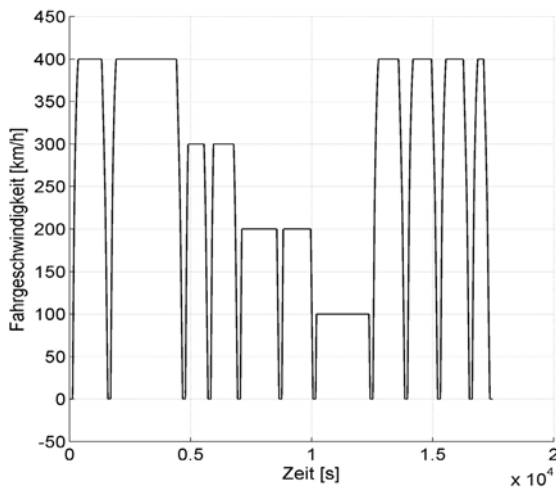
## 2.4 Ergebnisse der Simulationen

Den hier vorgestellten Simulationen liegen folgende Parameter zugrunde: NGT-HGV-Geometrie, Materialien und Abmessungen; jede der zwei Ebenen ist mit einer auf die Innenraumtemperatur von 22°C ideal geregelten Klimaanlage im Außenluftbetrieb ausgestattet; Sitzplatzkapazitäten sind 64 auf der oberen Ebene und 48 auf der unteren; der Zug ist zu Anfang weder vorgeheizt noch vorgekühlt. Als Referenzstrecke dient eine hypothetische Strecke von Paris nach Wien. Sie ist ca. 1230 km lang und die Fahrzeit beträgt mit dem gewählten Geschwindigkeitsprofil ca. 5 Stunden. Der Streckenverlauf und das Geschwindigkeitsprofil sind in Abbildung 7 und Abbildung 8 dargestellt.



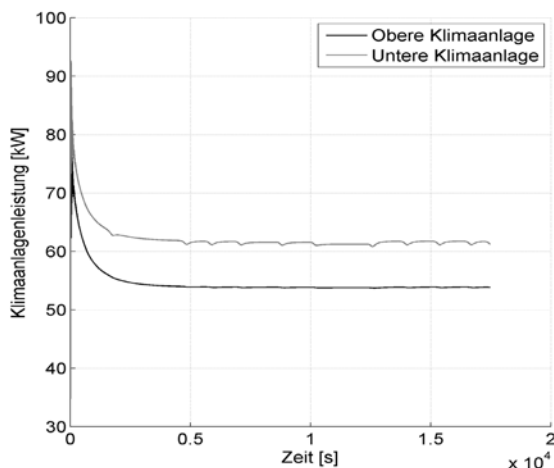
Abbildung 7. Referenzstrecke NGT





**Abbildung 8.** Fahrprofil NGT

Die Abbildung 9 zeigt den Energiebedarf der Klimaanlage im Heizbetrieb. Dabei handelt es sich um thermische Leistung. Die Außenlufttemperatur von  $-10^{\circ}\text{C}$ , relative Luftfeuchtigkeit von 50% und die Sonneneinstrahlung von 0 Watt werden zusammenfassend als Fall „Winter Nacht“ bezeichnet. Dabei ist der Zug auf beiden Ebenen vollbesetzt.



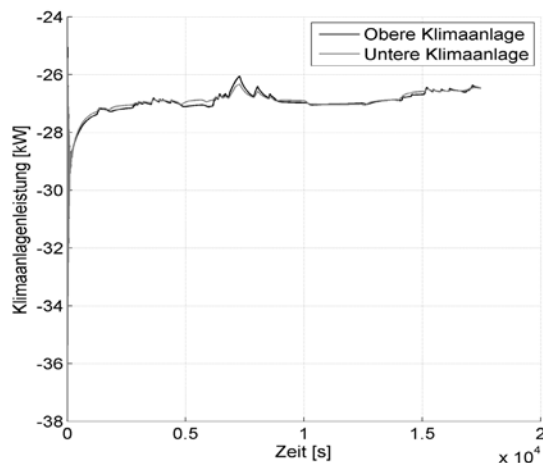
**Abbildung 9.** Heizbetrieb; Winter Nacht, vollbesetzt

Die obere Klimaanlage benötigt weniger Heizleistung als die untere. Das ist zum einen durch das kleinere Volumen zum anderen durch die größere Anzahl der Fahrgäste im oberen Bereich zu erklären. Aus dem gleichen Grund sind die geschwindigkeitsabhängigen Konvektionsverluste im unteren Bereich stärker ausgeprägt.

In der Abbildung 10 ist der Energiebedarf der Klimaanlage im Kühlbetrieb dargestellt. Unter der Be-

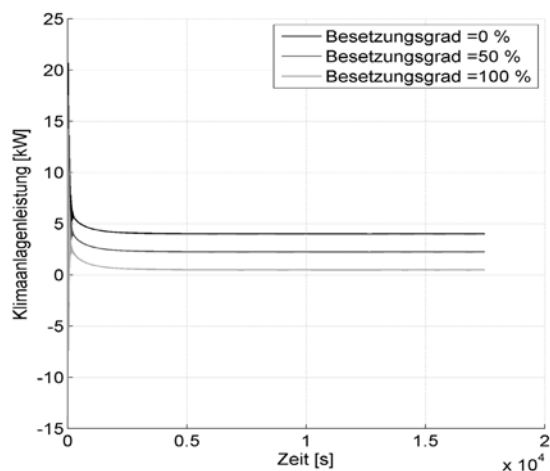
zeichnung „Sommer Tag“ sind die Außentemperatur von  $28^{\circ}\text{C}$ , relative Luftfeuchtigkeit von 70% und die Sonneneinstrahlung von max. 700 Watt zusammengefasst.

Der Sonnengang wird ausgehend von der Startzeit am 21. Juni um 12:00 Uhr und der aktuellen geographischen Zugposition berechnet. Der Besetzungsgrad beträgt auf beiden Ebenen 100%.



**Abbildung 10.** Kühlbetrieb; Sommer Tag, vollbesetzt

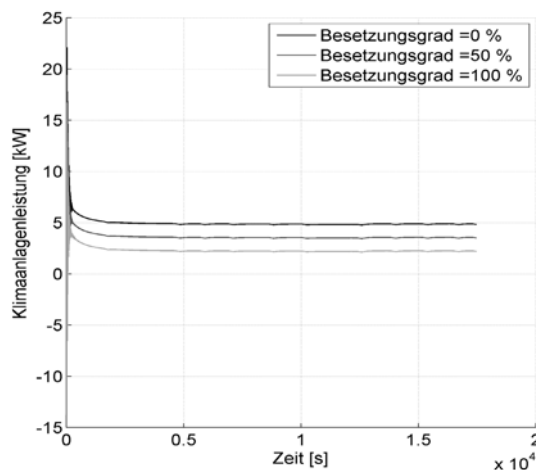
Die Energiebedarfe der Klimaanlage sind in diesem Fall nahezu identisch. Die Spitzen kommen durch die fehlende Sonneneinstrahlung in den auf der Strecke vorhandenen Tunnel zustande. Die große Spitze im mittleren Bereich kennzeichnet den neuen Stuttgarter Hauptbahnhof, einen Durchgangsbahnhof in Tieflage mit unterirdischen Zulaufstrecken [9], in den der NGT einen planmäßigen Halt hat.



**Abbildung 11.** Obere Klimaanlage bei variablem Besetzungsgrad (Sommer Nacht)

Die Abbildung 11 und die Abbildung 12 zeigen exemplarisch die Abhängigkeit der Klimaanlageleistungen vom Besetzungsgrad. Als Außenbedingung wurde der Fall „Sommer Nacht“ (Außenlufttemperatur von 18°C, relative Luftfeuchtigkeit von 80%, keine Sonneneinstrahlung) gewählt.

Der Besetzungsgrad wurde in zwei Schritten von 0% auf 100% erhöht. Dabei fällt der Bedarf an Heizleistung bei der oberen Klimaanlage von 4 kW auf 0,5 kW und bei der unteren Klimaanlage von 5 kW auf 2 kW. Der Grund dafür ist die Wärmeabstrahlung der Fahrgäste. Die Leistungsdifferenz zwischen den Ebenen lässt sich durch die Geometrieunterschiede und ungleiche Sitzplatzkapazitäten erklären.



**Abbildung 12.** Untere Klimaanlage bei variablem Besetzungsgrad (Sommer Nacht)

### 3 Ausblick

Das vorgestellte thermische Wagenkastenmodell erlaubt es, thermodynamische Vorgänge in einem beliebigen doppelstöckigen Wagenkasten zu untersuchen. Die hierzu notwendigen Anpassungen und Umparametrisierungen sind mit geringem Aufwand durchführbar. Durch die erfolgreiche Kopplung der Bibliotheken RV und HCL wurde nicht nur die komplexe Geometrie des NGT-Wagenkastens abgebildet, sondern auch schienenfahrzeugspezifische äußere Einflüsse (Tunnelfahrt, schneller Wechsel der Witterungsbedingungen, Bahnhofshalt, etc.) wurden berücksichtigt.

Einer der weiteren Schritte ist der Ausbau des Fahrgastmodells, welcher die Ergebnisse der momentan im Rahmen des NGT-Projekts laufenden psychologischen Untersuchungen des Komfortempfindens mit-

einbezieht. Des Weiteren werden die Temperaturschichtung im Fahrgastraum und ein komplexeres Klimaanlageanlagenmodell mit Möglichkeit des Umluftbetriebs implementiert. Darüber hinaus ist es angedacht, die Simulationsergebnisse mit einem realen physikalischen Versuchsaufbau zu überprüfen.

### 4 References

- [1] J. Ungethüm, D. Hülsebusch, H. Dittus und T. Braig. *Simulation von alternativen Fahrzeugantrieben in Modelica*. Workshop der ASIM/GI-Fachgruppe, 2010.
- [2] C. Klose. *Reduzierung des Energiebedarfs von Hochgeschwindigkeits-Triebzügen durch ein Verfahren zur Systemoptimierung der Steuerung von Triebfahrzeugen*. Dissertation, TU Dresden, Deutschland, 1999.
- [3] M. Otter und C. Schweiger. *Modellierung mechatronischer Systeme mit Modelica*. Mechatronischer Systementwurf, S. 39-50, 2004.
- [4] XRG Simulation GmbH. *Human Comfort Library 1.2, User's Guide*, 2012.
- [5] DIN EN 13129-1. *Bahnanwendungen-Luftbehandlung in Schienenfahrzeugen des Fernverkehrs-Teil 1: Behaglichkeitsparameter*. Deutsches Institut für Normung e.V., 2003.
- [6] K. Töpfer. *Grundausrüstungen*. VEB Verlag für Verkehrswesen Berlin, DDR, 1983.
- [7] T. Weiler. *Electric propulsion concept for an energy-efficient high speed train*. International Conference of Ecological Vehicles and Renewable Energies, 2011.
- [8] T. Braig. *Bremskonzept NGT HGV*. Next Generation Train - Bericht AP 1411. DLR intern. 2011.
- [9] DB AG und Projektpartner. *Finanzierungsvertrag Stuttgart 21, Anlage 3.2a*. [Online] [www.bahnprojekt-stuttgart-ulm.de/uploads/tx\\_smediamediathek/Finanzierungsvertrag\\_An1\\_3\\_2a.pdf](http://www.bahnprojekt-stuttgart-ulm.de/uploads/tx_smediamediathek/Finanzierungsvertrag_An1_3_2a.pdf), 2009.

# Nichtlineare Modelle hydraulischer Komponenten zur energieflussbasierten Modellierung

Ilja Alkov<sup>1</sup>, Dirk Weidemann<sup>1</sup>

<sup>1</sup>Fachhochschule Bielefeld

*il a.alkov, dirk.weidemann @fh-bielefeld.de*

In diesem Beitrag wird die Entwicklung von energieflussbasierten Modellen hydraulischer Komponenten und des daraus aggregierten Modells eines Hydrauliksystems beschrieben. Mit besonderem Hinblick auf nichtlineare Zusammenhänge in den Modellen hydraulischer Prozesse werden Problematiken hervorgehoben, die die Lösbarkeit und die Stabilität der Lösung der Modelle beeinflussen. Abschließend werden Vorschläge zur Verbesserung der Modelle mit Fokus auf die energieflussbasierte Formulierung vorgestellt und Erfahrungen aus der Simulation artgleicher Modelle präsentiert.

## 1 Einleitung

Die energieflussbasierte Modellierung ist eine Methode zur Formulierung mathematischer Modelle ausgehend von der Charakterisierung physikalischer Vorgänge durch Energieflussgrößen. Mit Energiefluss wird hierbei die zeitliche Veränderung einer als Energie definierten Größe umschrieben. Für verschiedene physikalische Vorgänge lassen sich jeweils zwei Variablen definieren, deren Produkt die ausgetauschte Leistung oder den Energiefluss beschreibt. Somit ist die energieflussbasierte Modellierung eine Anwendung der Hamilton-Gleichungen. Einen umfassenden Überblick zu diesem Thema geben Jeltsema und Scherpen in [1]. Darüber hinaus ist die energieflussbasierte Modellierung auch für Vorgänge ökonomischer bzw. soziologischer Natur und bei der Warteschlangentheorie anwendbar, falls geeignete Ersatzgrößen definiert werden. Auf die Vorteile des Ansatzes der energieflussbasierten Modellierung wird in [1] und [2] eingegangen.

Der im Rahmen dieses Beitrags gewählte Modellierungsansatz für die Beschreibung hydraulischer Komponenten basiert auf der Reduktion detaillierter dreidimensionaler Strömungsmodelle auf eindimensionale. Die Beeinflussung der Strömungsgrößen durch die Komponenten wird durch konzentriertparametrische Modelle, d.h. durch gewöhnliche Differentialgleichungen beschrieben. Ferner erfolgt die Beschreibung der einzelnen Komponenten im Sinne einer objektorientierten Modellierung. Entsprechend leitet

sich das Gesamtmodell des hydraulischen Systems aus der Verknüpfung der einzelnen Teilmodelle ab.

Die energieflussbasierte Betrachtung der Modelle hydraulischer Systeme ist aus mehreren Gesichtspunkten vorteilhaft. Die interessierenden Größen der Strömung in den Komponenten sind der Druck und der Volumenstrom. Das Produkt der Größen gibt die hydraulische Leistung wieder. Die Klassifikation der hydraulischen Vorgänge nach der Leistungsbilanz in Widerstände, induktive Speicher, kapazitive Speicher, Leistungswandler und Quellen ist möglich. Allerdings führt die Übernahme der in der energieflussbasierten Modellierung etablierten linearen Gleichungen für die Beschreibung der Zusammenhänge in den hydraulischen Komponenten für verbreitete Modellanforderungen zu einer sehr groben Näherung. Ein allgemein bekanntes Beispiel für das Modell eines hydraulischen Widerstandselementes ist die Proportionalitätsbeziehung zwischen dem Volumenstrom in einer hydraulischen Blende und der Quadratwurzel des zugeordneten Druckabfalls. Darüber hinaus beeinflussen gegebenenfalls die druck- und temperaturabhängigen Eigenschaften des hydraulischen Fluides das Verhalten der Komponenten maßgebend.

In diesem Beitrag wird die Entwicklung von energieflussbasierten Modellen hydraulischer Komponenten und des daraus aggregierten Modells eines Hydrauliksystems beschrieben. Mit besonderem Hinblick auf nichtlineare Zusammenhänge in den Modellen hydraulischer Prozesse werden Problematiken hervorgehoben, die die Lösbarkeit und die Stabilität der Lö-

sung der Modelle beeinflussen. Abschließend werden Vorschläge zur Verbesserung der Modelle mit Fokus auf die energieflussbasierte Formulierung vorgestellt und Erfahrungen aus der Simulation artgleicher Modelle präsentiert.

## 2 Modellierung

Die Beschreibung fluidmechanischer Prozesse mit mathematischen energieflussbasierten Modellen ist, wie Modellbildung jeder anderer Art, auf der Grundlage geeigneter Annahmen vorzunehmen. Die Eignung der Annahmen ist bzgl. des Verwendungszweckes des Modells, des Genauigkeitsanspruches, des zulässigen Modellierungs- und Simulationsaufwandes zu überprüfen.

Die energieflussbasierten Modelle können unterschiedlich dargestellt werden. Beispielsweise gelingt die Darstellung der Modelle in der Port-Hamilton-Form [3], als Bondgraphen [4], sowie durch verschiedene Arten generalisierter Netzwerke. In diesem Beitrag werden Bondgraphen zur graphischen Darstellung für die energieflussbasierten Modelle angewandt. Die Implementierung der Modelle erfolgt in der objektorientierten Modellierungssprache Modelica.

### 2.1 Bondgraphen

Bondgraphen sind eine graphische Modellsynthese-, Modelldarstellungs- und Modelanalysemethodik [5]. Grundlage der Methodik bilden die Definitionen der Knoten und der gerichteter Kanten der Bondgraphen. Entsprechend dem energieflussbasierten Ansatz werden die Leistungsvariablen (engl. power variables) effort und flow definiert. Den Knoten, auch Elemente der Bondgraphen genannt, sind Gleichungen zur Verknüpfung der Leistungsvariablen zugeordnet. Die gerichteten Kanten sind namensgebend für die Methodik und werden als Bonds bezeichnet. Die Bonds kennzeichnen den energieerhaltenden und speicherfreien Leistungsaustausch zwischen den Interface-Strukturen der Elemente. Die Richtung der Bonds gibt hierbei die positiv definierte Richtung des Energieflusses an. Die Interface-Strukturen der Elemente sind Ports zum definierten Leistungsaustausch. Über die Ports werden Informationen sowohl über die effort- als auch über die flow-Variable übergeben.

In der Methodik der Bondgraphen werden die Einport-Grundelemente Se effort-Quelle, Sf flow-Quelle, C kapazitiver Speicher, I induktiver Speicher, R Widerstand, die Zweiport-Grundelemente TF Transfor-

mator, GY Gyrator und die Multiport-Grundelemente 0 und 1 Junction definiert. Die Gleichungen zur Beschreibung der Zusammenhänge der Variablen in den Elementen werden als konstitutive Gleichungen bezeichnet. Die Quellen geben den Wert der entsprechenden Leistungsvariablen fest vor. Im kapazitiven Speicher ist der Wert der effort-Variablen proportional zum zeitlichen Integral der flow-Variablen, wobei der Kehrwert des Proportionalitätsparameters als Kapazität bezeichnet wird. Im induktiven Speicher ist dagegen der Wert der flow-Variablen proportional zum zeitlichen Integral der effort-Variablen, wobei der Kehrwert des Proportionalitätsparameters als Induktivität bezeichnet wird. Für den Widerstand ist die effort-Variable über den Widerstandsparameter proportional der flow-Variablen. Der Transformator und der Gyrator stellen energieerhaltende und speicherfreie Leistungstransformation dar. Im Fall des Transformators ist die effort-Variable eines Anschlusses proportional zur effort-Variablen des anderen, wobei das Entsprechende für die flow-Variablen gilt. Beim Gyrator ist die effort-Variable eines Anschlusses proportional zur flow-Variablen des anderen. Den Junctionen (engl. junction) wird in der Methodik eine besondere Bedeutung zugeteilt. Diese Multiportelemente stellen Verknüpfungen oder Zwangsbedingungen zwischen den Variablen der Elemente dar. Im Einzelnen können die Junctionen als Formulierung der Gleichgewichtsbedingungen oder Erhaltungssätze aufgefasst werden. Die Vorzeichen in Summationsätzen werden durch die Richtung der Bonds an den Ports der Junctionen vorgegeben.

Element	Konstitutive Gleichung
Se	$e = e_s$
Sf	$f = f_s$
C	$e = \frac{1}{c} \cdot \int_{t_0}^t f \cdot d\tau + e(t_0)$
I	$f = \frac{1}{i} \cdot \int_{t_0}^t e \cdot d\tau + f(t_0)$
R	$e = r \cdot f$
TF	$e_1 = r \cdot e_2$ $f_2 = r \cdot f_1$

GY	$e_1 = r \cdot f_2$ $e_2 = r \cdot f_1$
0	$\sum_{k \in {}_0} s_k \cdot f_k = 0, \quad s_k \in \{1, -1\}$ $e_k = e \quad \forall k, \quad {}_0 \subset N$
1	$\sum_{k \in {}_1} s_k \cdot e_k = 0, \quad s_k \in \{1, -1\}$ $f_k = f \quad \forall k, \quad {}_1 \subset N$

**Tabelle 1.** Grundelemente der Bondgraphen

Die vorangestellte Tabelle erfasst die konstitutiven Gleichungen der Grundelemente der Bondgraphen. Darin bezeichnen  $e$  und  $f$  die effort- und die flow-Variable entsprechend. Mit  $e_s$ ,  $f_s$ ,  $c$ ,  $i$ ,  $r$  sind jeweils die Parameter der Elemente bezeichnet. Die Indizierung der Leistungsvariablen kennzeichnet unterschiedliche Ports der Elemente. Die Junktionen sind Multiport-Elemente, entsprechend bezeichnen  ${}_0$  bzw.  ${}_1$  die Menge der Portindizes. Die Multiplikatoren  $s_k$  nehmen den Wert 1 an, wenn der zugeordnete Bond zur Junktion gerichtet ist oder  $-1$  im anderen Fall. Die Richtung der Bonds ist durch den Modellierer vorzugeben.

Mit den Grundelementen der Bondgraphen sind lineare Modelle mit konstanten Eingangsgrößen darstellbar. Eine wesentliche Erweiterung stellen modulierte Elemente dar. Die Kurzbezeichnung der modulierten Elemente setzt sich aus der Kurzbezeichnung des entsprechenden Grundelementes und eines vorangestellten M-Buchstaben zusammen. Die modulierten Elemente sind Verallgemeinerungen für parametrierbare Grundelemente, deren Parameter als externe Variable vorgegeben wird. Durch Kombination von Grundelementen, modulierten Elementen und Sensorelementen, zur Feststellung der Werte der Leistungsvariablen, können mit einer fallbezogenen Definition der notwendigen Signalschaltungen beliebige Modelle beschrieben werden. Im Allgemeinen führt die Definition anwendungsspezifischer Elemente mit der Bondgraph-Interface-Struktur zum einen zu einer erheblich einfacheren Modellbildung und zum anderen zu einer besseren Übersichtlichkeit der Modelle.

## 2.2 Modelle der Fluideigenschaften

Die Eigenschaften des Fluides, vor allem die Dichte und die Viskosität, und deren Modellierung in Ab-

hängigkeit u.a. von Druck und Temperatur nehmen bei der Beschreibung hydraulischer Vorgänge eine zentrale Rolle ein. In diesem Beitrag werden die Wärmeströme und die Temperaturverteilung in dem zu modellierenden System nicht betrachtet. Die Angabe der Temperatur erfolgt in Form eines zeit- und zustandsinvarianten Parameters. Die Variation des Druckes wird im großen Bereich  $[1; 10^8]$  Pa zugelassen. Das führt zur Notwendigkeit der Definition des elementbezogenen Druckes und der Bestimmung der Fluideigenschaften für jedes Element. Darüber hinaus wird die Abhängigkeit der Fluideigenschaften von anderen Größen, wie etwa die Abhängigkeit der Viskosität von der Schergeschwindigkeit, nicht betrachtet. Als Arbeitsfluid wird ein homogenes Gemisch aus Mineralöl und Luft angenommen. Der Massenanteil der Luft im Gemisch wird als zeit- und zustandsinvariant betrachtet. Ferner werden Entmischungsvorgänge nicht berücksichtigt.

Die Dichte des Mineralöls wird mit der Kompressibilität  $\kappa_{oil}$  und dem Wärmeausdehnungskoeffizient  $\gamma_{oil}$  durch

$$\rho_{oil} = \rho_{oil,ref} \cdot \exp(\kappa_{oil} \cdot (p - p_{ref}) - \gamma_{oil} \cdot (T - T_{ref})) \quad (1)$$

beschrieben. Darin ist  $p$  der Druck,  $T$  die Temperatur,  $p_{ref}$  der Bezugsdruck,  $T_{ref}$  die Bezugstemperatur,  $\rho_{oil}$  die Dichte des Mineralöls und  $\rho_{oil,ref} = \rho_{oil}(p_{ref}, T_{ref})$  die Bezugsdichte des Mineralöls. Das Modell wurde mit vorhandenen Messwerten parametrisiert und verifiziert.

Die Dichte der Luft wird mit der Zustandsgleichung idealer Gase

$$\rho_{air} = \rho_{air,ref} \cdot \left( \frac{p}{p_{ref}} \right)^1 \cdot \left( \frac{T}{T_{ref}} \right)^{-1} \quad (2)$$

modelliert. Darin ist  $p$  der Druck,  $T$  die Temperatur,  $p_{ref}$  der Bezugsdruck,  $T_{ref}$  die Bezugstemperatur,  $\rho_{air}$  die Dichte der Luft und  $\rho_{air,ref} = \rho_{air}(p_{ref}, T_{ref})$  die Bezugsdichte der Luft.

Die Dichte des Gemisches  $\rho$  errechnet sich entsprechend

$$\rho = \frac{1}{\frac{\mu_{oil}}{\rho_{oil}} + \frac{\mu_{air}}{\rho_{air}}} \quad (3)$$

aus den Dichten des Mineralöls und der Luft gewichtet mit den relativen Massenanteilen  $\mu_{oil}$  des Mineralöls und  $\mu_{air} = 1 - \mu_{oil}$  der Luft. Aus der Definition der Dichte des Gemisches kann der sich in der Literatur etablierte Ersatzkompressibilitätsmodul (vgl. [4]) des hydraulischen Arbeitsfluides abgeleitet werden. Dieser wird zur Verifikation des Dichtemodells des Gemisches verwendet.

Die Viskosität des Mineralöls wird Roelands-Beziehung

$$\eta_{oil} = \eta_{oil,ref} \cdot \exp\left(\ln\left(\frac{\eta_{oil,ref}}{6.315 \cdot 10^{-5} \text{ Pa} \cdot \text{s}}\right) \cdot \psi_{roel}\right) \quad (4)$$

$$\psi_{roel} = -1 + \left(1 + \frac{p - p_{ref}}{1.96 \cdot 10^8 \text{ Pa}}\right)^\zeta \cdot \left(\frac{T - 138\text{K}}{T_{ref} - 138\text{K}}\right)^{-\xi}$$

gemäß [7] beschrieben. Darin ist  $p$  der Druck,  $T$  die Temperatur,  $p_{ref}$  der Bezugsdruck,  $T_{ref}$  die Bezugstemperatur,  $\eta_{oil}$  die Viskosität des Mineralöls und  $\eta_{oil,ref} = \eta_{oil}(p_{ref}, T_{ref})$  die Bezugsviskosität des Mineralöls,  $\zeta$  und  $\xi$  Parameter.

Die Viskosität der Luft wird mit der Approximationsfunktion

$$\eta_{air} = \eta_{air,ref} \cdot \frac{\theta(p, T)}{\theta(p_{ref}, T_{ref})} \quad (5)$$

$$\theta(p, T) = p^{0.01} \cdot T^{0.75} + 0.132 \cdot p \cdot T^{-2}$$

der aus [8] entnommenen Daten bestimmt. Darin ist  $p$  der Druck,  $T$  die Temperatur,  $p_{ref}$  der Bezugsdruck,  $T_{ref}$  die Bezugstemperatur,  $\eta_{air}$  die Viskosität der Luft und  $\eta_{air,ref} = \eta_{air}(p_{ref}, T_{ref})$  die Bezugsviskosität der Luft.

Für die Bestimmung der Viskosität des Gemisches werden zunächst die relativen Volumenanteile der Bestandteile entsprechend

$$\varphi_{oil} = \frac{\frac{\mu_{oil}}{\rho_{oil}}}{\frac{\mu_{oil}}{\rho_{oil}} + \frac{\mu_{air}}{\rho_{air}}}, \quad \varphi_{air} = \frac{\frac{\mu_{air}}{\rho_{air}}}{\frac{\mu_{oil}}{\rho_{oil}} + \frac{\mu_{air}}{\rho_{air}}} \quad (6)$$

errechnet. Darin sind  $\varphi_{oil}$  und  $\varphi_{air}$  die relativen Volumenanteile,  $\mu_{oil}$  und  $\mu_{air}$  die relativen Massenanteile,  $\rho_{oil}$  und  $\rho_{air}$  die Dichten des Mineralöls und der Luft entsprechend.

teile,  $\rho_{oil}$  und  $\rho_{air}$  die Dichten des Mineralöls und der Luft entsprechend.

Mit Bezug auf den physikalischen Charakter der Viskosität wird die Viskosität des homogenen Gemisches durch

$$\eta = \frac{\eta_{oil} \cdot \varphi_{oil}^{2/3} + \eta_{air} \cdot \varphi_{air}^{2/3}}{\varphi_{oil}^{2/3} + \varphi_{air}^{2/3}} \quad (7)$$

modelliert. Darin ist  $\eta$  die Viskosität des Gemisches,  $\varphi_{oil}$  und  $\varphi_{air}$  die relativen Volumenanteile,  $\eta_{oil}$  und  $\eta_{air}$  die Viskositäten des Mineralöls und der Luft entsprechend.

### 2.3 Modelle hydraulischer Prozesse

Die hydraulischen Prozesse sind Teilvorgänge beim fluidmechanischen Leistungsaustausch. Diese werden durch die energieflussbasierten Modelle in diesem Abschnitt beschrieben. Der Leistungsaustausch wird hierbei transient räumlich eindimensional entlang einer Hauptstromrichtung betrachtet. Folgende hydraulische Prozessarten werden abgegrenzt: kapazitive Energiespeicherung, induktive Energiespeicherung und Dissipation. Das Verhalten der im Weiteren behandelten hydraulischen Komponenten wird durch geeignete Kombination dieser Prozesse beschrieben. Die Variablen und Parameter der hydraulischen Prozesse werden mit Bezug auf die Komponenten, die sie einbeziehen, erläutert. Als Leistungsvariablen  $e$  effort und  $f$  flow für die Modellierung der hydraulischen Prozesse werden der Druck und der Volumenstrom entsprechend gewählt.

Das dynamische Verhalten der hydraulischen Komponenten wird maßgeblich durch deren kapazitive Speicherfähigkeit beeinflusst. Diese Eigenschaft folgt aus der Elastizität der Komponenten und damit aus der Variabilität deren Fassungsvermögens. Die Kompressibilität des Arbeitsfluides stellt vor allem bei niedrigem Druck einen bedeutenden Beitrag der kapazitiven Speicherfähigkeit bereit. Beide Effekte werden durch

$$f = \frac{1}{\rho} \cdot \frac{d}{dt}(\rho \cdot v) \quad (8)$$

$$v = v_0 \cdot \exp(c_v \cdot (e - e_0))$$

modelliert. Darin ist  $e$  der Druck und  $e_0$  der Initialdruck in der Komponente,  $f$  der durch die Komponente auf Grund des kapazitiven Speichervor-

ganges dem Hauptstrom entnommene Differenzvolumenstrom,  $v$  das Volumen,  $v_0$  das Initialvolumen und  $c_v$  die Volumensteifigkeitskennzahl der Komponente,  $\rho$  bezeichnet die Dichte des Arbeitsfluides.

Aus der Sicht der energieflussbasierten Modellierung ist die induktive Speicherung der zur kapazitiven Energiespeicherung duale dynamische Effekt. Diese Eigenschaft fluidmechanischer Prozesse resultiert aus der Trägheit des Fluides und der das Fluidvolumen eingrenzenden Wandung. Die Auswirkungen der Wandungsträgheit sind für übliche Verhältnisse in den hydraulischen Komponenten von untergeordneter Bedeutung. Der Zusammenhang zwischen den Leistungsvariablen für den Prozess der induktiven Energiespeicherung wird aus dem Impulssatz für die in der hydraulischen Komponente eingeschlossene Flüssigkeitssäule (vgl. [6]) durch

$$e = \frac{d}{dt} \left( \frac{l \cdot \rho}{A} \cdot f \right) \quad (9)$$

bestimmt. Darin ist  $e$  die zur Veränderung des Volumenstromes notwendige Druckdifferenz, der  $f$  Volumenstrom in der Komponente,  $l$  die Länge und  $A$  die Querschnittsfläche der Flüssigkeitssäule in der Komponente,  $\rho$  die bezeichnet die Dichte des Arbeitsfluides.

Die Beschreibung dissipativer hydraulischer Vorgänge ist komplex. Diese Prozesse sind nichtlinear und verändern ihre Charakteristik abhängig von der Dynamik der instationären Strömungen, vgl. [9]. Für die angestrebten Betrachtungen wird der instationäre frequenzabhängige Durchflusswiderstand nicht berücksichtigt. Die Dissipation in den fluidmechanischen Vorgängen wird durch mehrere Modelle beschrieben. Grundsätzlich wird zwischen hydraulischem Durchflusswiderstand gerader Rohre und Rohreinbauten differenziert.

Der dissipative Widerstand in geraden Rohren ist aus theoretischer und experimenteller Sicht extensiv untersucht [10], [11], [12]. Das im Folgenden dargestellte Modell basiert auf der Darcy-Weisbach-Gleichung und verwendet eine aus der Haaland-Approximation [13] entwickelte explizite stetige Form für die Berechnung des Rohrreibungsfaktors.

$$\begin{aligned} e &= \lambda \cdot \frac{l \cdot \rho}{2 \cdot d_h \cdot A^2} \cdot |f| \cdot f \\ Re &= \frac{d_h \cdot \rho \cdot |f|}{A \cdot \eta} \\ \lambda &= \lambda_{laminar} \cdot 0.5 \cdot \left( 1 - \tanh \left( 2.5 \cdot \frac{Re - Re_{crit}}{Re_{range}} \right) \right) + \\ &\quad + \lambda_{haaland} \cdot 0.5 \cdot \left( 1 + \tanh \left( 2.5 \cdot \frac{Re - Re_{crit}}{Re_{range}} \right) \right) \\ \lambda_{laminar} &= \frac{64}{Re + Re_{min} \cdot \left( 1 - \tanh \left( \frac{Re}{Re_{min}} \right) \right)} \\ \lambda_{haaland} &= \left( \left( \frac{-1.8}{\ln(10)} \cdot \psi_{haaland} \right) + 10^{-38} \right)^{-2} \\ \psi_{haaland} &= \ln \left( \left( \frac{r_h}{3.7} \right)^{1.11} + \frac{6.9}{Re - 10^{-38} \cdot Re_{min}} \right) \end{aligned} \quad (10)$$

Darin ist  $e$  die zur Aufrechterhaltung des Volumenstromes notwendige Druckdifferenz,  $f$  der Volumenstrom in der Komponente,  $l$  die Länge,  $A$  die Querschnittsfläche und  $d_h$  der hydraulische Durchmesser der Flüssigkeitssäule in der Komponente,  $r_h$  die relative hydraulische Rauigkeit,  $\lambda$  der Rohrreibungsfaktor,  $Re$  die Reynolds-Zahl,  $Re_{crit} \approx 2320$  die Reynolds-Zahl und  $Re_{range}$  der Reynolds-Zahlbereich für den Übergang von laminarer zu turbulenter Strömung,  $Re_{min}$  der Mindestwert der Reynolds-Zahl für den stetigen Übergang zum ruhenden Fluid,  $\rho$  und  $\eta$  bezeichnen die Dichte und die Viskosität des Arbeitsfluides.

Hydraulische Komponenten, wie Krümmer, Blenden, Drosseln oder Querschnittsänderungen, werden als Rohreinbauten bezeichnet. Dissipative Widerstände in solchen Komponenten werden mit

$$e = \frac{e_{ref} \cdot \rho}{\rho_{ref} \cdot f_{ref}^\gamma} \cdot |f|^{\gamma-1} \cdot f \quad (11)$$

und

$$e = \frac{e_{ref} \cdot \eta}{\eta_{ref} \cdot f_{ref}^\gamma} \cdot |f|^{\gamma-1} \cdot f \quad (12)$$

beschrieben (vgl. [6]). Darin ist  $e$  die zur Aufrechterhaltung des Volumenstromes notwendige Druckdif-

ferenz,  $f$  der Volumenstrom in der Komponente,  $\gamma$  der Volumenstromexponent,  $\rho$  und  $\eta$  die Dichte und die Viskosität des Arbeitsfluides,  $e_{ref}$ ,  $f_{ref}$ ,  $\rho_{ref}$ ,  $\eta_{ref}$  bezeichnen entsprechende Bezugswerte zur Dimensionierung des Widerstandes.

In fluidmechanischen Vorgängen treten oft zwei Prozesse mit Quellcharakteristik auf. Zum einen ist das die Veränderung des Druckniveaus in einem konstanten homogenen Massenkraftfeld mit parallelen gradlinigen Feldlinien, wie annähernd im Gravitationsfeld der Erde.

$$e = \rho \cdot a \cdot d \quad (13)$$

Zum anderen ist das die Druckänderung bedingt durch die Querschnittsänderung der Leitung.

$$e = \frac{\rho}{2} \cdot |f|^2 \cdot \left( \frac{1}{A_1^2} - \frac{1}{A_2^2} \right) \quad (14)$$

Darin ist  $e$  die Druckveränderung,  $f$  der Volumenstrom,  $a$  die Intensität des Massenkraftfeldes,  $d$  der Weg des Fluides entlang der Richtung der Feldlinien,  $A_1$  und  $A_2$  die Querschnittsflächen,  $\rho$  bezeichnet die Dichte des Arbeitsfluides.

Die druck- und temperaturabhängigen Eigenschaften des Fluides gehen in alle oben dargestellten Modelle ein. Die Temperatur wird, wie bereits erläutert, als invarianter Parameter vorgegeben und kann damit in jedem modellierten Prozess als solcher implementiert werden. Für die Druckabhängigkeit ist die Definition des für die Eigenschaften des Fluides maßgeblichen Druckes notwendig. Zum Erhalt der Unabhängigkeit der modellierten Prozesse von der Durchflussrichtung wird für die Bestimmung der Fluideigenschaften der arithmetische Mittelwert der Drücke vor und nach dem hydraulischen Prozess verwendet. Eine kompakte Implementierungsform und graphische Darstellung werden bei diesem Ansatz durch die Ausführung aller hydraulischen Prozesse als Zweiport-Elemente erreicht. Der für die Eigenschaften des Fluides maßgebliche Druck ist damit der Mittelwert der effort-Variablen an den Ports des Elementes.

## 2.4 Modellierung hydraulischer Komponenten

Als hydraulische Komponenten sollen technische Einheiten verstanden werden, in denen fluidmechanische Leistungsaustauschvorgänge ablaufen können. Das Verhalten der hydraulischen Komponenten wird

durch geeignete Kombination hydraulischer Prozesse mit Bondgraphen beschrieben. Im Folgenden wird die Modellbildung für eine begrenzte Auswahl verbreiteter hydraulischer Komponenten beschrieben.

In hydraulischen Systemen werden oft Blenden und Drosseln eingesetzt. Diese Dämpfungskomponenten zeichnen sich vor allem durch geringe räumliche Ausdehnung, durch hohen Widerstand gegen Volumenänderung und durch geringe induktive Speichereigenschaft aus. Die prägende Eigenschaft der Blenden und Drosseln ist der dissipative Widerstand. In der Literatur (vgl. [6]) finden sich die Formen (11) und (12) zur Beschreibung der Blenden und der Drosseln. Durch die Kombination dieser Formen können weitere Effekte für besondere Bauformen und Druck-Volumenstrom-Kennlinien abgebildet werden.

Krümmen und Verzweigungen besitzen meistens ein nicht vernachlässigbares Fassungsvermögen, jedoch eine geringe Längenausdehnung entlang der Hauptstromrichtung. Diese Komponenten werden daher durch die Kombination von zwei hydraulischen kapazitiven Speicherelementen entsprechend Form (8) und eines zwischen diesen eingesetzten dissipativen Widerstandes der Form (11) modelliert (vgl. [14]).

Hydraulische Leitungen, Rohre und Schläuche sind komplexe dynamische Systeme. Aus der Literatur sind verschiedene Modellierungsansätze bekannt. Häufig wird in der Modellierung Bezug auf den Aufbau der Leitung genommen [9]. Im Rahmen der hier beschriebenen Methodik wird eine Leitung durch das Aneinanderreihen der hydraulischen kapazitiven Energiespeicher, der hydraulischen induktiven Energiespeicher und der hydraulischen dissipativen Elemente der Form (10) modelliert. Diese Kombination wird in Form eines Bondgraphen dargestellt. Der gewählte Ansatz lässt Erweiterungen des Leitungsmodells und Verdichtung der räumlichen Diskretisierung der Leitung ohne wesentlichen Aufwand zu.

Hydraulische Ventile können durch eine Kombination steuerbarer Blenden und Drosseln beschrieben werden (vgl. [15]). Die Modelle der Blenden und Drosseln werden, wie oben beschrieben, ausgeführt. Der Widerstandskennwert der Dämpfungskomponenten und damit der zugeordneten Durchflussmöglichkeiten wird hierbei durch eine geeignete Signalschaltung vorgegeben. Die kapazitive und die induktive Energiespeichereigenschaften der hydraulischen Ventile können meistens vernachlässigt werden. Für Sonderfälle können Speicherelemente dem Widerstandsmodell



dell nachgeschaltet werden. Die Abb. 1 stellt ein Modell eines 4/3-Wegeventils dar, das als Bondgraph mittels Modelica-Sprache implementiert wurde. Darin stellen die MHRT-Elemente die steuerbaren Blenden dar, die 0-Elemente sind Junktionen, pt1 bis pt4 sind Bondgraph-Ports, die restlichen Elemente bilden die Signalschaltung ab. Die Bonds sind in der Abb. 1 durch  $\bullet \Rightarrow$  symbolisiert und  $\Rightarrow$  bezeichnen die Signale.

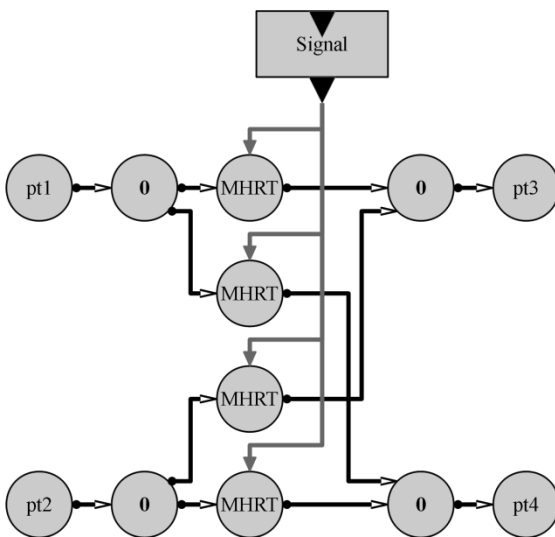


Abbildung 1. 4/3-Wegeventil

## 2.5 Modelle hydraulischer Systeme

Hydraulische Systeme sind technische aus hydraulischen Komponenten bestehende Einheiten. Die Modelle dieser Systeme können entsprechend durch Kombination der Modelle der hydraulischen Komponenten erstellt werden. Das entwickelte Modell besteht aus mehreren untereinander vernetzten Energieflusssträngen. Dieses Vorgehen, sowie die bereits erläuterte Aggregation der Modelle der hydraulischen Prozesse zu Modellen der hydraulischen Komponenten, zeichnet die Methodik der Bondgraphen aus. Die Energieflussstränge in den Modellen der hydraulischen Systeme können ohne wesentlichen Aufwand erweitert und verzweigt werden, wodurch nach Bedarf das Modell um weitere Effekte ergänzt werden kann. Die Objektorientierung der gewählten Modellierungssprache begünstigt diesen Prozess und bietet alle entsprechenden Vorteile.

Das hierarchische Modell wird zur Lösung in ein System differentialer und algebraischer Gleichungen überführt. Vor diesem Hintergrund ist die Interaktion der Teilmodelle, vor allem in Bezug auf deren Nichtlinearität, zu untersuchen.

## 3 Nichtlinearität energieflussbasierter Modelle

Einer der wesentlichen Vorteile der energieflussbasierten Modellierung und als Sonderfall der Bondgraphen ist die Möglichkeit der akasalen Portdefinition. Die Ports der Elemente übergeben Informationen sowohl über die effort- als auch über die flow-Variable im Element auf die Modellierungsebene außerhalb des Elementes. Abhängig von der Kombination der Elemente im Modell werden die Gleichungen der Elemente entweder nach der effort- oder der flow-Variablen aufgelöst. Der Prozess der Identifikation der Variablen, nach der die Gleichungen des Elementes aufgelöst werden, wird als Kausalisierung bezeichnet. Die Ermittlung der beiden möglichen kausalisierten Formen erfordert eine Inversion der Gleichungen der Elemente. Die linearen Grundelemente der Bondgraphen, mit Ausnahme der Quellelemente, können automatisiert symbolisch in beide kausalisierte Formen überführt werden. Für nichtlineare Elemente ist dieser Vorgang im Allgemeinen symbolisch nicht möglich. Eine numerische iterative Inversion der nichtlinearen Gleichungen der Elemente führt für größere Modelle zum hohen Rechenaufwand und numerischen Instabilität. Drei verschiedene Möglichkeiten zur effizienten Modellbildung sind mit Hinblick auf die dargestellte Problematik möglich. Die Ports der nichtlinearen Elemente können kausal definiert werden. Das führt zu erheblichen Einschränkungen in der Entwicklung der zusammengesetzten Modelle. Wenn möglich sollten beide kausalisierten Formen der Gleichungen in dem Element implementiert werden. Die Verwendung einer der Formen kann über einen Elementparameter manuell angefordert werden. Das erfordert eine manuelle Auswertung der automatisierten Analyse des Gleichungssystems des Gesamtmodells. Aufgrund der Lösungsverfahren der eingesetzten Simulationsumgebung wird dieser Ansatz verfolgt. Eine weitere Möglichkeit ist die automatisierte Initiierung der Verwendung einer der kausalisierten, im Element implementierten Form. Dieses Vorgehen erfordert bei der eingesetzten Modellierungsumgebung einen erheblichen Implementierungsaufwand. Bei der Umsetzung müssen die Anforderungen der Lösungsverfahren vor allem für die differential-algebraische Gleichungssysteme berücksichtigt werden. In (16) ist der dissipative Prozess der Form (11) in beiden kausalisierten Formen dargestellt. Formelzeichen der Gleichung (11) wurden übernommen.

$$e = \frac{e_{ref} \cdot \rho}{\rho_{ref} \cdot f_{ref}^\gamma} \cdot |f|^{\gamma-1} \cdot f$$

$$f = \left( \frac{e_{ref} \cdot \rho}{\rho_{ref} \cdot f_{ref}^\gamma} \right)^{\frac{-1}{\gamma}} \cdot |e|^{\frac{1}{\gamma}-1} \cdot e$$
(15)

Die Gleichungen der jeweils betrachteten nichtlinearen Elemente können entsprechend der Charakteristik dieser Elemente in die Form der entsprechenden Grundelemente überführt werden.

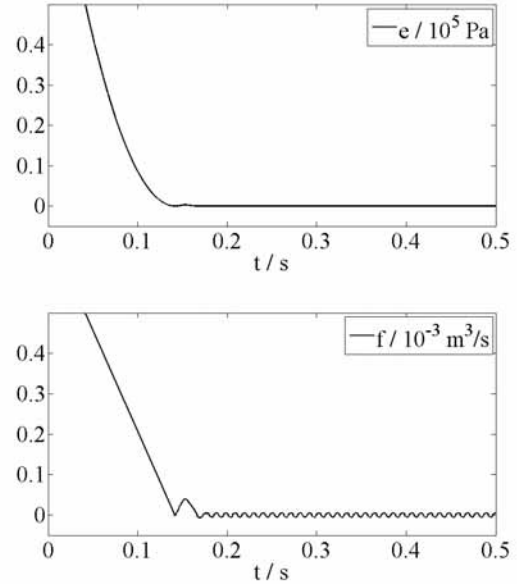
$$e = r \cdot f$$

$$r = \begin{cases} \frac{e_{ref} \cdot \rho}{\rho_{ref} \cdot f_{ref}^\gamma} \cdot |f|^{\gamma-1} \wedge p_{caus} = 1 \\ \left( \frac{e_{ref} \cdot \rho}{\rho_{ref} \cdot f_{ref}^\gamma} \right)^{\frac{1}{\gamma}} \cdot |e|^{\frac{1}{\gamma}-1} \wedge p_{caus} = 2 \end{cases}$$
(16)

Diese Umformung ist für den dissipativen Prozess der Form (11) in (16) dargestellt. Darin ist  $p_{caus}$  der Parameter zur Unterscheidung der kausalisierten Form des Elementes und  $r$  der Widerstandsbeiwert des dissipativen Prozesses. Andere Formelzeichen wurden übernommen.

Offensichtlich variiert der Widerstandsbeiwert in Abhängigkeit der Leistungsvariablen und kann unter Umständen den Wert 0 oder annähernd 0 erreichen. Das gleiche gilt auch für den Kehrwert. Diese Grenzwerte führen zu erheblichen numerischen Problemen bei der Lösung des Modells. Die Auswirkungen sind bei der Simulation eines vereinfachten Modells eines Krümmers deutlich zu beobachten. Das Modell besteht aus zwei linearen kapazitiven Speicherelementen und einem dissipativen hydraulischen Element der Form (16). Der simulierte Vorgang bildet den Druckausgleich der kapazitiven Elemente ab. Die Druckdifferenz an dem dissipativen Element strebt stetig gegen 0, wodurch der Widerstandsbeiwert für den gewählten Volumenstromexponenten  $\gamma = 2$  ebenfalls gegen 0 strebt. Für einen geringen Widerstandsbeiwert gehen die Verläufe der Variablen in einen nicht abklingenden Schwingungszustand über, dessen Amplitude und Frequenz von dem gewählten Lösungsverfahren und dem Toleranzwert des Lösungsverfahrens abhängt. Die Schwingung entsteht aus der numerischen Lösung der Differential-Algebraischen-Gleichungen. Die Verläufe der Leistungsvariablen sind in der Abb. 2 dargestellt. Als

Lösungsverfahren wird der DASSL-Algorithmus mit der relativen Toleranz von  $10^{-4}$  verwendet.



**Abbildung 2.** Simulation eines Krümmers

Zur Verbesserung des Modellverhaltens wird der Widerstandsbeiwert beschränkt. Die Einführung dieser Beschränkung ermöglicht eine vollständige Vermeidung der numerischen Schwingung.

Andere nichtlineare Modelle hydraulischer Prozesse, vorausgesetzt die Möglichkeit einer Überführung in eine mit Grundelementen der Bondgraphen korrespondierende Form besteht, können entsprechend behandelt werden. Wesentlich ist die Betrachtung der in den nichtlinearen Modellen veränderlichen Proportionalitätsparameter in den konstitutiven Gleichungen der Elemente. Einen unmittelbaren Zugang zu diesen Größen bietet die energieflussbasierte Modellierung.

## 4 Zusammenfassung

In diesem Beitrag wurde eine Methodik zur Erstellung von Modelle hydraulischer Komponenten und Systeme beschrieben. Basis für die entwickelten Modelle ist die Betrachtung der Energieflüsse. Die Teilprozesse des hydraulischen Leistungsaustausches sind aus dieser Sicht modelliert. Die Aggregation der Prozessmodelle zu Modellen der Komponenten wurde an mehreren Beispielen erläutert. Der Aufbau von Modellen hydraulischer aus Komponenten zusammengesetzter Systeme wurde dargestellt.

Abschließend wurden ausgewählte Problematiken nichtlinearer Modelle hydraulischer Komponenten beschrieben. Auswirkungen der Nichtlinearität auf die Modellbildung und numerische Lösung wurden skizziert. Vorschläge zur Behandlung der dargestellten Probleme wurden präsentiert und erläutert.

## 5 Quellen

- [1] D. Jeltsema und J. M. A. Scherpen. *Multidomain modeling of nonlinear networks and systems*. IEEE Control Systems Magazine, S. 28–59, 2009.
- [2] R. A. Layton. *Principles of Analytical System Dynamics*. Springer-Verlag, USA, 1998.
- [3] A. J. van der Schaft. *Port-Hamiltonian systems: network modeling and control of nonlinear physical systems*. Advanced Dynamics and Control of Structures and Machines, CISM Courses and lectures No. 444, S. 127–168, 2004.
- [4] W. Borutzky. *Bond Graph Methodology*. Springer-Verlag, UK, 2010.
- [5] H. M. Paynter. *Analysis and Design of Engineering Systems*. M.I.T. Press, USA, 1961.
- [6] H. Murrenhoff. *Grundlagen der Fluidtechnik: Teil 1: Hydraulik*. Shaker Verlag, Deutschland, 2012.
- [7] C. J. A. Roelands, J. C. Vlugter und H. I. Watterman. *The Viscosity-Temperature-Pressure Relationship of Lubricating Oils and its Correlation with Chemical Constitution*. ASME Journal of Basic Engineering, Vol. 11, S. 601–611, 1963.
- [8] J. H. Spurk und N. Aksel. *Strömungslehre, Einführung in die Theorie der Strömungen*. Springer Verlag, Deutschland, 2010.
- [9] R. Etlender. *Modellierung und Simulation der Wellenausbreitung in flexiblen hydraulischen Leitungen*. Dissertation, Universität Stuttgart, Deutschland, 2012.
- [10] J. Nikuradse. *Strömungsgesetze in Rauhen Röhren*. V.D.I Forschungsheft, Nr. 361, S. 1–22, 1933.
- [11] C. F. Colebrook. *Turbulent Flow in Pipes, with particular Reference to the Transition Region Between the Smooth and Rough Pipe Laws*. Journal of Institution of Civil Engineers, Vol. 11, S. 133–156, 1938–1939.
- [12] L. F. Moody. *Friction factors for pipe flow*. Transactions of the ASME, Vol. 66, No. 8, S. 671–684, 1944.
- [13] S. E. Haaland. *Simple and explicit formulas for the friction factor in turbulent pipe flow*. Transactions of the ASME, Journal of Fluids Engineering Vol. 105, No. 1, S. 89–90, 1983.
- [14] G. Bauer. *Hydraulik, Grundlagen, Bauelemente, Anwendungen*. Vieweg+Teubner Verlag, Deutschland, 2011.
- [15] I. Alkov. *Verallgemeinerndes Modell hydraulischer Ventile*. Tagungsband FLUIDON Conference, Aachen, 2012.



## Echtzeitsimulation hydraulischer Antriebe Vom Analogrechner zum FPGA

**Peter Beater**  
beater@fh-swf.de

### Zeitlicher Ablauf

- Analogrechner
- Digitalrechner
- FPGA
  
- Hydrostatisches Getriebe
- PC-Simulation (offline / real time)
- FPGA in Echtzeit

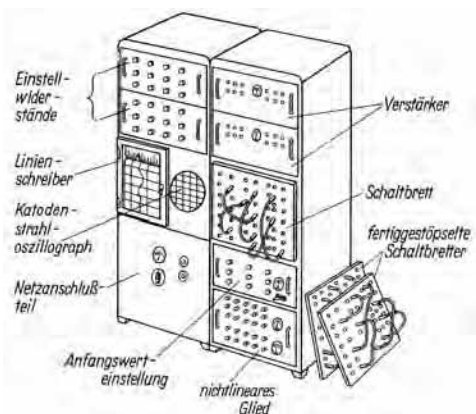
ASIM 28. 2. 2013 – P. Beater – FH-SWF - 1

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die Anfänge - Analogrechner - 60er bis 80er

### Signale durch Analogspannungen dargestellt - Kabelprogrammierung



w. Oppelt Kleines Handbuch ...

ASIM 28. 2. 2013 – P. Beater – FH-SWF - 2

### Gerätetechnik

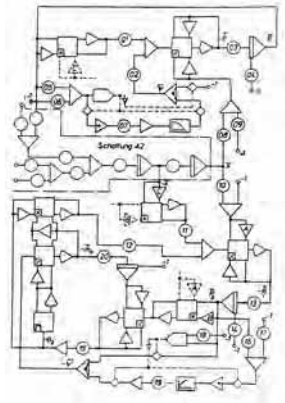
- Verstärker
- Integrierer
- Summierer
- Potentiometer
- Schaltbrett zum Programmieren
- Voltmeter
- Linienschreiber
- Oszillograph
- Nichtlinearitäten machbar

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die Anfänge - Analogrechner - 60er bis 80er

Signale durch Analogspannungen dargestellt



### Einsatz

- Lineare Systeme
- Geringe Systemordnung
- Begrenzter Wertebereich
- Begrenzte Auflösung
- ECHTZEIT
- Parallele Arbeitsweise
- Programmdokumentation?

I. Forster Diss RWTH Aachen

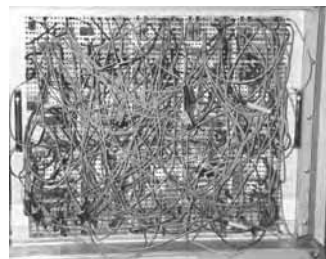
ASIM 28. 2. 2013 – P. Beater – FH-SWF - 3

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die Anfänge - Analogrechner - 60er bis 80er

Signale durch Analogspannungen dargestellt



ASIM 28. 2. 2013 – P. Beater – FH-SWF - 4

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Der nächste Schritt - Digitalrechner - 80er bis heute

Signalverläufe durch numerische Lösung von Differentialgleichungen



### Gerätetechnik

- Frei programmierbarer Prozessor (Boolesche Verknüpfungen)  
Sequentielle Arbeitsweise – ein Befehl nach dem nächsten
- Tastatur, Bildschirm
- Massenspeicher, Plotter, Drucker
- Gleitkommaeinheit
- AD- und DA-Wandler

ASIM 28. 2. 2013 – P. Beater – FH-SWF - 5

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Der nächste Schritt - Digitalrechner - 80er bis heute

Signalverläufe durch numerische Lösung von Differentialgleichungen



### Einsatz

- Nichtlineare Systeme
- Beliebige Systemordnung
- Sehr großer Wertebereich bei Gleitkommadarstellung
- Sehr große Auflösung
- Spezielle Sprachen zur Simulation
- ECHTZEIT ????

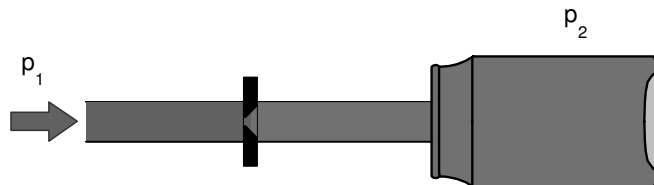
ASIM 28. 2. 2013 – P. Beater – FH-SWF - 6

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Der nächste Schritt - Digitalrechner - 80er bis heute

### Dynamik hydraulischer Systeme



Die Zeitkonstante für den Druckaufbau ergibt sich bei einem Volumen  $V = 330 \text{ ml}$  und einem Blendendurchmesser  $d = 1 \text{ mm}$  bei üblichem Öl zu:

$$T = \frac{2 \cdot V \cdot k_l \cdot v \cdot \rho}{d^3 \cdot \beta \cdot \pi} = 30 \mu\text{s}$$

ASIM 28. 2. 2013 – P. Beater – FH-SWF - 7

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die Gegenwart - Field Programmable Gate Array

### Softwaremäßig konfigurierbare Digitaltechnik „digitaler Analogrechner“



### Gerätetechnik

- Boolesche Algebra
- Addition, Subtraktion
- Einige Multiplizierer (18\*18 Bit)
- Taktfrequenz 40 MHz
- VHDL „natürliche Programmiersprache“
- Programmierung durch Bitfile

ASIM 28. 2. 2013 – P. Beater – FH-SWF - 8

Fachhochschule  
Südwestfalen  
University of Applied Sciences



## Echtzeitsimulation Hydraulik

Die Gegenwart - Field Programmable Gate Array

Softwaremäßig konfigurierbare Digitaltechnik –  
Ansatz von National Instruments Corp.



### Einsatz

- Graphische Programmierung LabVIEW
- Festkommazahlen
- Kontrollstrukturen
- Modularisierung durch SubVI
- Echtzeituhren
- AD- DA-Wandler im CompactRIO

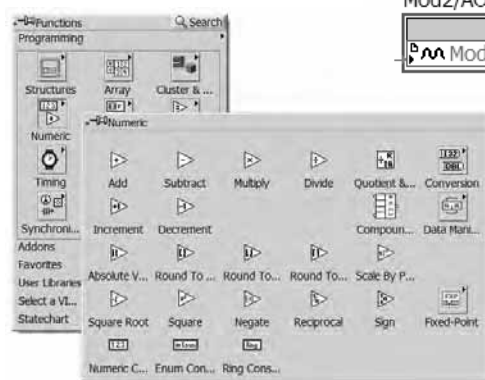
ASIM 28. 2. 2013 – P. Beater – FH-SWF - 9

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die Gegenwart - Field Programmable Gate Array

Softwaremäßig konfigurierbare Digitaltechnik – LabVIEW

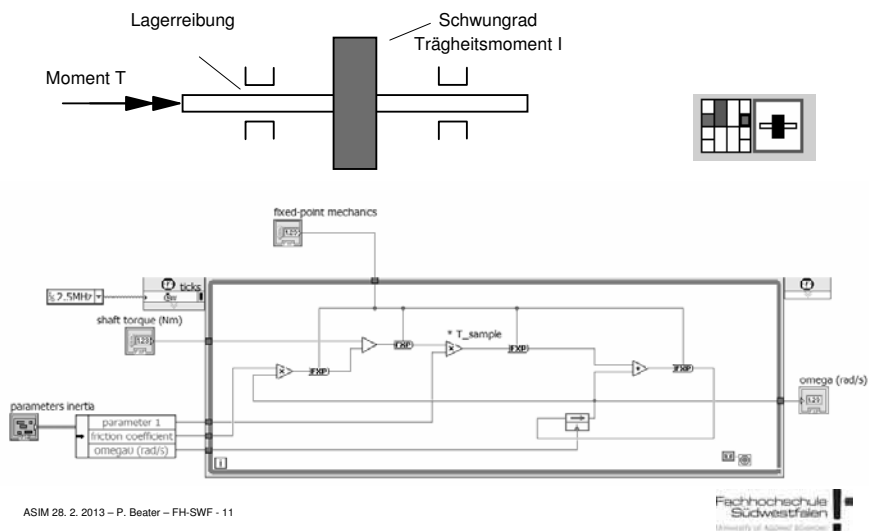


ASIM 28. 2. 2013 – P. Beater – FH-SWF - 10

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die Gegenwart - Field Programmable Gate Array



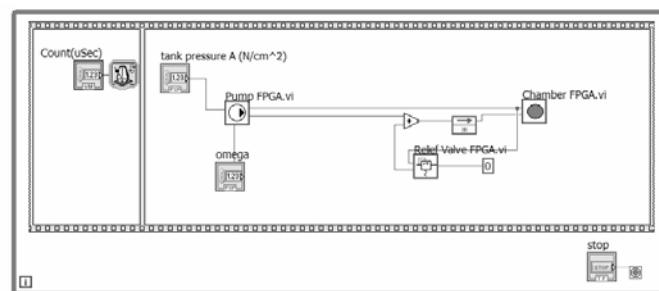
ASIM 28. 2. 2013 – P. Beater – FH-SWF - 11

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die Gegenwart - Field Programmable Gate Array

### Softwaremäßig konfigurierbare Digitaltechnik – LabVIEW



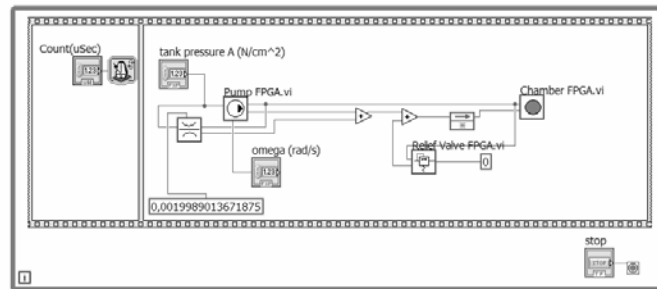
ASIM 28. 2. 2013 – P. Beater – FH-SWF - 12

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die Gegenwart - Field Programmable Gate Array

### Softwaremäßig konfigurierbare Digitaltechnik – LabVIEW



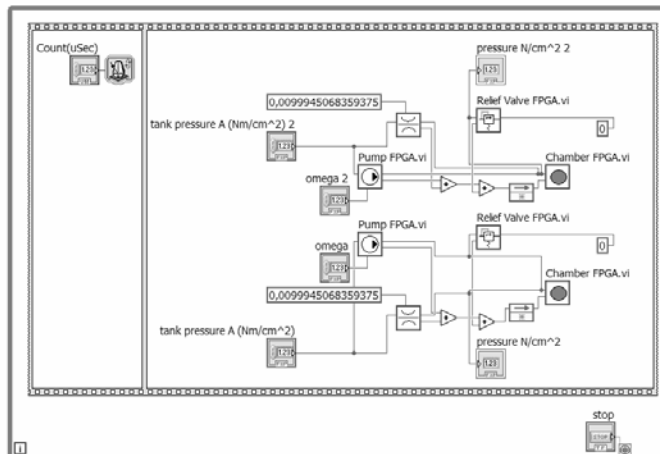
ASIM 28. 2. 2013 – P. Beater – FH-SWF - 13

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die Gegenwart - Field Programmable Gate Array

Softw



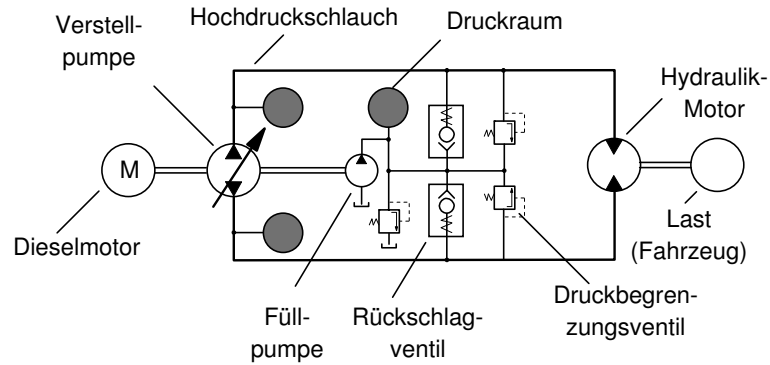
ASIM 28. 2. 2013 – P. Beater – FH-SWF - 14

Fachhochschule  
Südwestfalen  
University of Applied Sciences

# Echtzeitsimulation Hydraulik

Die Gegenwart - Field Programmable Gate Array

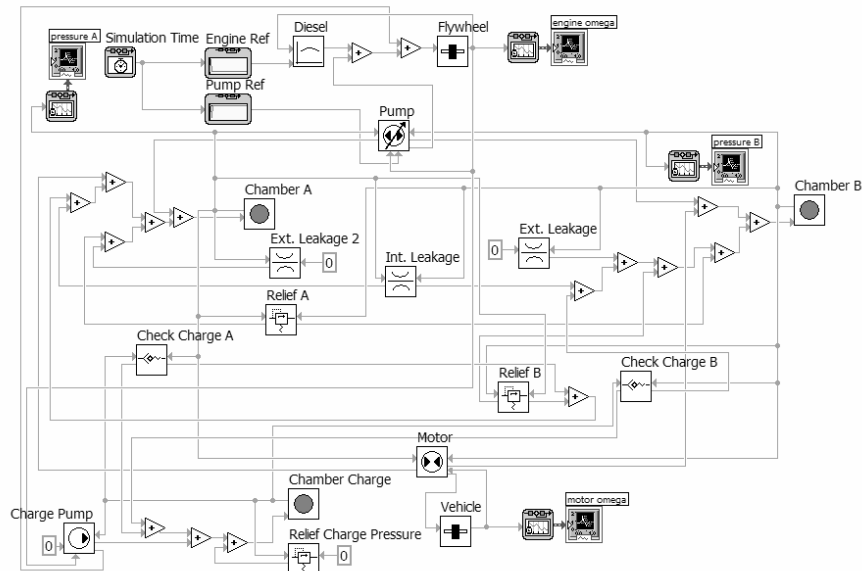
## Beispiel: Hydrostatisches Getriebe



ASIM 28. 2. 2013 – P. Beater – FH-SWF - 15

Fachhochschule  
Südwestfalen  
University of Applied Sciences

# Echtzeitsimulation Hydraulik

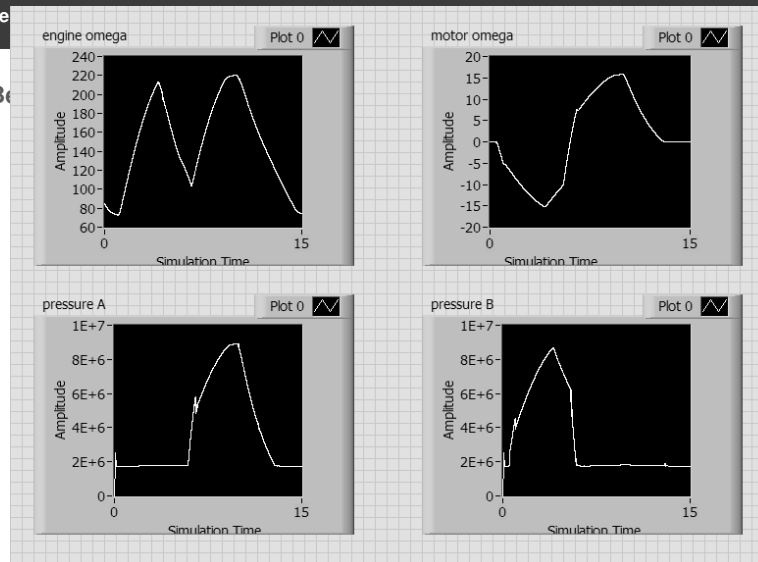


University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die

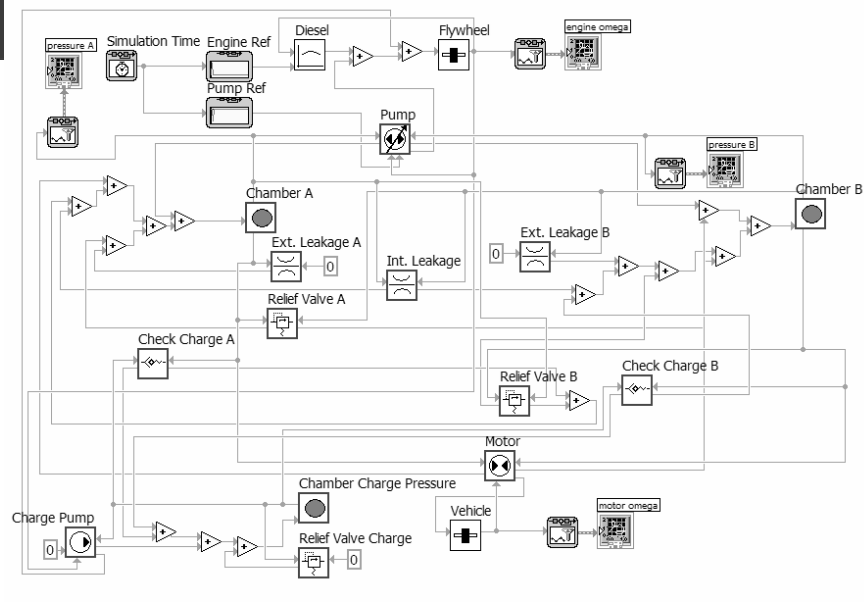
Be



ASIM 28. 2. 2013 – P. Beater – FH-SWF - 17

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

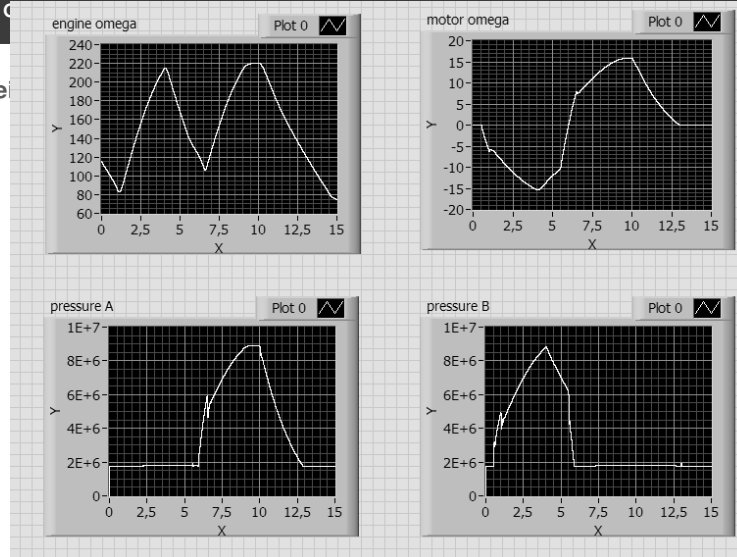


Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die C

Be

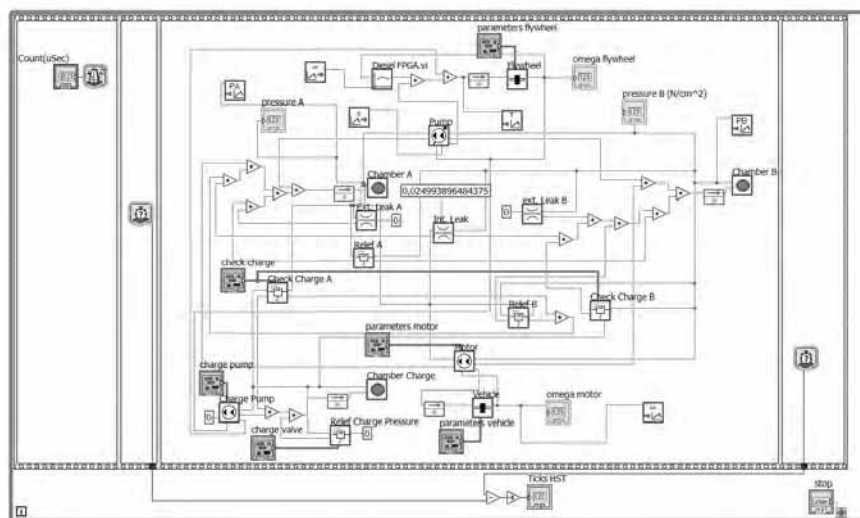


ASIM 28. 2. 2013 – P. Beater – FH-SWF - 19

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die Gegenwart – Field Programmable Gate Array

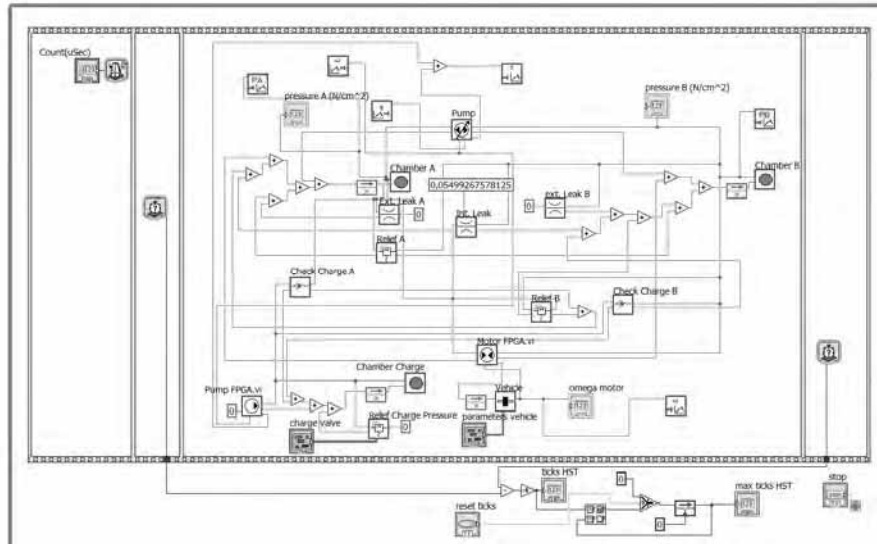


ASIM 28. 2. 2013 – P. Beater – FH-SWF - 20

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die Gegenwart - Field Programmable Gate Array

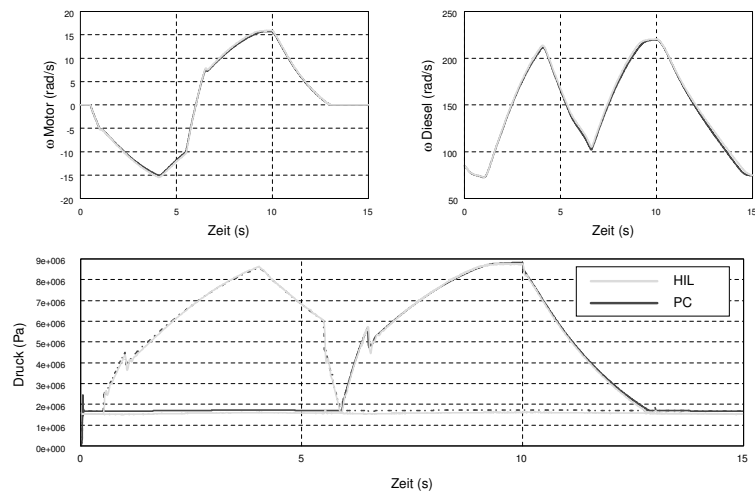


ASIM 28. 2. 2013 - P. Beater - FH-SWF - 21

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die Gegenwart - Field Programmable Gate Array



ASIM 28. 2. 2013 - P. Beater - FH-SWF - 22

Fachhochschule  
Südwestfalen  
University of Applied Sciences

## Echtzeitsimulation Hydraulik

Die Gegenwart - Field Programmable Gate Array

### Literatur

- W Oppelt Kleines Handbuch technischer Regelvorgänge Verlag Chemie
- I Forster Elektrohydraulisch Lastsimulation 1988
- P Beater Entwurf hydraulischer Maschinen Springer 1999
- P Beater Modeling and Real-Time Simulation of Hydraulic Drives – An Introduction to the Fluid Power Toolset for LabVIEW, CompactRIO and LabVIEW-FPGA, Kindle eBook, [www.amazon.com](http://www.amazon.com), geplant Frühjahr 2013
- N Kehtarnavaz, S Mahotra Digital signal processing laboratory – LabVIEW Based FPGA Implementation BrownWalker Press 2011
- Fluid Power Toolset for LabVIEW [www.ni.com/labview-tools-network](http://www.ni.com/labview-tools-network) geplant Frühjahr 2013



# Multiphysikalische 1D-Simulation hybrider Antriebssysteme für Schienenfahrzeuge

Dipl.-Ing. (FH) Sebastian Schmid

Engineering Service, Voith Turbo GmbH & Co. KG, Heidenheim

*zzSebastian.zzSchmid@voith.com*

Prototypversuche verschiedener Hersteller zeigen dass die Hybridisierung von Schienenfahrzeugen die Möglichkeit bietet den Kraftstoffverbrauch deutlich zu senken. Aus Sicht eines Antriebslieferanten für Schienenfahrzeuge erfordert dabei jede Kombination aus Fahrzeug, Antriebssystem und Streckenprofil die genaue Analyse ob eine Hybridisierung wirtschaftlich ist. Deshalb entwickelt Voith Turbo im Rahmen eines im Oktober 2012 gestarteten Projekts mit der University of Bradford ein Simulationstool um hybride Antriebssysteme für spezifische Schienenfahrzeuganwendungen auszulegen und Wirtschaftlichkeitsanalysen durchzuführen. Der erste Anwendungsfall für das Simulationstool ist ein paralleles Hybridsystem für einen diesel-hydraulischen Triebwagen. Für dieses System werden ein Doppelschichtkondensator- und ein Batteriemodul als Energiespeicher mit dem Simulationstool untersucht. Als Fahrprofile werden zwei Strecken aus dem EU Projekt CleanER-D verwendet. Die Funktionsweise des Simulationstools und erste Simulationsergebnisse werden dargestellt und das weitere Vorgehen erläutert.

## 1 Einleitung

Die Bundesregierung von Deutschland verfolgt im Rahmen ihres nationalen Entwicklungsplans Elektromobilität das ehrgeizige Ziel, dass bis zum Jahr 2020 eine Million Elektrofahrzeuge auf Deutschlands Straßen unterwegs sind [1]. Der Erfolg dieses Vorhabens hängt dabei wesentlich von der zukünftigen Entwicklung leistungsfähiger, leichter und für den Endkunden finanziell erschwinglicher Energiespeicher ab. Die weltweite Forcierung an Entwicklungsvorhaben im Bereich Lithium-Ionen-Akkumulatoren zeigt dass diese Technologie dabei eine Schlüsselrolle für die Elektromobilität der Zukunft einnehmen wird. Derzeitige Energiespeicher in bereits erhältlichen Hybridfahrzeugen wie z.B. dem Toyota Prius oder dem Opel Ampera bieten jedoch nur einen begrenzten Energiegehalt um das zusätzliche Batteriegewicht möglichst niedrig zu halten. Vor allem bei Autobahn- und Überlandfahrten mit wenigen Bremsvorgängen verbrauchen diese Fahrzeuge dabei oft mehr wie vergleichbare konventionelle PKW mit modernen hocheffizienten Dieselmotoren.

Im Gegensatz zum Straßenverkehr bietet der Schienenverkehr mit seinen definierten Fahrplänen und damit vorhersagbaren zyklischen Lastprofilen ideale Voraussetzungen für die Hybridisierung von Antriebssträngen. Durch Rekuperation von Bremsenergie kann der Kraftstoffverbrauch von dieselmotorisch angetriebenen Triebwagen reduziert werden. Vor

allem bei Triebwagen im öffentlichen Personennahverkehr (z. B. S- und Regionalbahnen) konnten hier im Regelbetrieb bei einem japanischen Fahrzeugbetreiber und bei Prototypversuchen eines deutschen Herstellers von Antriebssystemen Verbrauchseinsparungen im teilweise zweistelligen Prozentbereich erzielt werden [2,3]. Ein weiterer großer Vorteil, bzgl. der Hybridisierung von Schienenfahrzeugen im Gegensatz zu PKW, ergibt sich durch die Antriebssystemarchitektur von dieselektrisch angetriebenen Fahrzeugen. Durch den verwendeten Zwischenkreis können elektrische Energiespeicher einfach in das Fahrzeug integriert werden. Hinzukommt, dass das zusätzliche Gewicht durch einen Energiespeicher in Relation zum Fahrzeuggewicht einen geringeren negativen Einfluss auf den Kraftstoffverbrauch ausübt wie bei einem PKW.

Aus diesem Grund befindet sich derzeit bei Voith Turbo ein Simulationstool im Aufbau, welches es ermöglicht zu einem frühen Entwicklungszeitpunkt verschiedene hybride Antriebssysteme für Schienenfahrzeuge simulativ miteinander vergleichen zu können. Das Hauptziel besteht dabei darin, für eine vorgegebene Kombination aus Fahrzeug, Fahrprofil und Fahrplan, eine in Bezug auf Wirtschaftlichkeit optimierte Antriebsarchitektur zu bestimmen. Um dieses Vorhaben zu unterstützen ist die Voith Turbo GmbH & Co. KG im Oktober 2012 eine Kooperation mit dem HyPER-C Institut unter der Leitung von Professor Kambiz Ebrahimi von der University of Bradford

eingegangen. Der aktuelle Stand des Vorhabens wird in diesem Beitrag aufgezeigt.

## 2 Simulationsmodell eines hybriden Triebwagens

Auf Basis eines konkreten Entwicklungsprojekts hat man sich bei Voith Turbo dafür entschieden, zunächst einen zweiteiligen Triebwagen, im Folgenden als DMU (Diesel Multiple Unit) bezeichnet, mit einem hybriden diesel-hydraulischen Antriebssystem (DIWAhybrid) in dem Simulationstool abzubilden. Das Fahrzeug besitzt zwei Antriebseinheiten, welche jeweils aus einem Voith V2876T3 Reihen-6-Zylinder Dieselmotor (Emissionsstufe Euro Stage IIIB) mit 390 kW Leistung bei 1800 1/min und einem hydraulischen Voith DIWA 864.5 Getriebe mit 320 kW Eingangsleistung bestehen. Das Getriebe besitzt einen 1. Gang in welchem leistungsverzweigt über einen Wandler und mechanisch über die Planetenradsätze angefahren wird, mit zunehmender Geschwindigkeit werden drei mechanische Gänge durchfahren bei denen der Wandler überbrückt wird um einen höheren Wirkungsgrad zu erzielen [4]. Jede Antriebseinheit treibt das jeweilige Drehgestell am Anfang und am Ende des Fahrzeugs an. Die beiden Fahrzeughälften sind über ein Jakobs-Drehgestell miteinander verbunden. Ein vergleichbarer zweiteiliger DMU mit zwei Antriebseinheiten ist in Abbildung 1 dargestellt.



Abbildung 1: Beispiel eines zweiteiligen DMU

Am Abtrieb des Getriebes befindet sich eine zusätzliche Zahnradstufe über welche ein Elektromotor verbunden ist. Dieser besitzt ein Anfahrtdrehmoment von 1050 Nm und eine maximale Leistung 180 kW. Damit bietet sich mit dem Fahrzeug die Möglichkeit im Traktionsbetrieb das Fahrzeug zusätzlich zum Dieselmotor elektrisch anzutreiben und bei Verzögerungen Bremsenergie zu rekuperieren um den elektrischen Energiespeicher zu laden. Es handelt sich somit per Definition um ein paralleles Hybridsystem. Über das Radsatzgetriebe und die Räder wird schließlich die Zugkraft erzeugt um das Schienenfahrzeug anzutreiben.

Mit Hilfe des multiphysikalischen Simulationstool LMS Imagine.Lab AMESim wurde ein Modell des Antriebssystems aufgebaut. Die Besonderheit ist

dabei, dass alle für die Erzeugung der Übersetzungen in dem DIWA Getriebe verwendete Bauteile (Wandler, Kupplungen und Planetenradsätze) ebenfalls in dem Modell mit der entsprechenden Parametrierung zu finden sind. In Abbildung 2 ist für das bessere Verständnis der Teil des Antriebssystems dargestellt, in welchem das DIWA Getriebe modelliert wurde.

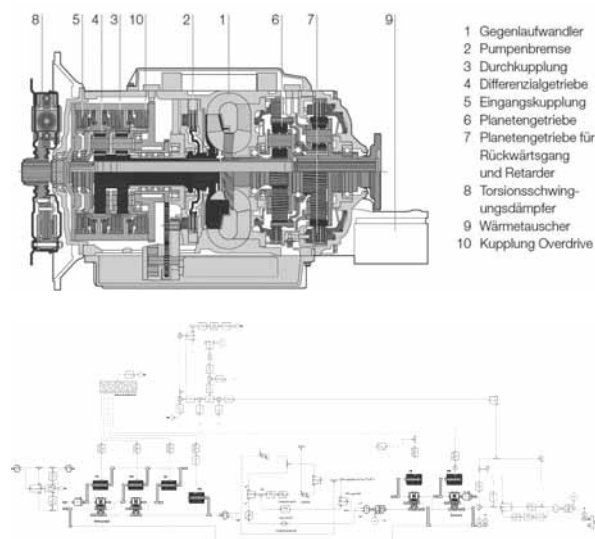


Abbildung 2: Querschnitt des hydraulischen Getriebes Voith DIWA 864.5 [4] und Modellierung in AMESim

Alle weiteren für das Antriebssystem und das Fahrzeug charakteristischen Größen, wie z.B. Schaltpunkte, Momenten- und Leistungsbegrenzungen des Getriebes, oder der Drehmomentenverlauf und das Verbrauchskennfeld des Dieselmotors, sind ebenfalls in dem Modell hinterlegt. Darüber hinaus lassen sich alle an den Dieselmotor angeschlossenen Nebenverbraucher entsprechend parametrieren.

## 3 Fahrplanfahrt mit dem Simulationstool

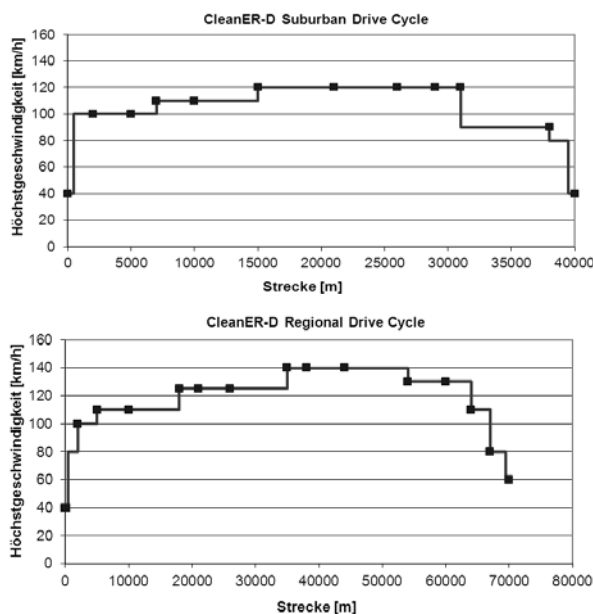
Wie in der Einleitung bereits angesprochen, soll es mit dem Simulationstool ermöglicht werden, verschiedene hybride Antriebssysteme für Schienenfahrzeuge simulativ miteinander vergleichen zu können. Dazu müssen zunächst Fahrprofile und Fahrweisen definiert werden, um mit dem Simulationstool eine Fahrsimulation durchzuführen.

### 3.1 Fahrprofile

Im Automobilbereich werden normierte Fahrzyklen (z.B. Neuer Europäischer Fahrzyklus) mit fest definierten Geschwindigkeitsverläufen für die Ver-

brauchsermittlung von PKW herangezogen. Im Gegensatz dazu verwenden Anbieter von Schienenfahrzeugantriebssystemen bei ihrer Auslegung das konkrete Fahrzeug mit den entsprechenden Fahrzeugdaten und Fahrprofile und Fahrplänen, welche von dem zukünftigen Fahrzeugbetreiber zur Verfügung gestellt werden. Desweiteren werden in den meisten Fällen vom dem Fahrzeugbetreiber ein Zugkraftverlauf und eine bestimmte Anfahrbeschleunigung gefordert, was ebenfalls bei der Auslegung beachtet werden muss.

Um hybride Antriebssysteme für Schienenfahrzeuge herstellerunabhängig miteinander vergleichen zu können, wurden in dem seit März 2009 laufendem EU-Projekt CleanER-D generische Fahrprofile für verschiedene Fahrzeuggattungen definiert. Für das im Rahmen des Projekts mit der University of Bradford geplante Simulationstool wurden zwei dieser Strecken verwendet: Suburban und Regional Drive Cycle [5]. Der Suburban Drive Cycle entspricht dabei mit einer Länge von 40 km, einem mittleren Haltestellenabstand von 3.6 km und einer Höchstgeschwindigkeit von 120 km/h einem typischen S-Bahnprofil. Der Regional Drive Cycle hat eine Länge von 70 km, einen mittleren Haltestellenabstand von 5.0 km und eine Höchstgeschwindigkeit auf der Strecke von 140 km/h und ist daher einem typischen Regionalbahnprofil gleichzusetzen. Die auf beiden Strecken jeweils erlaubte Höchstgeschwindigkeit ist in Abbildung 3 dargestellt.

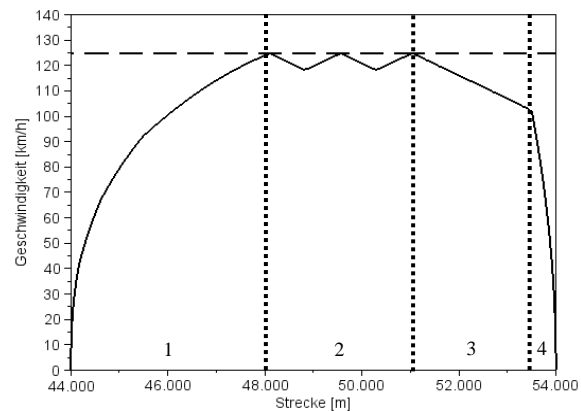


**Abbildung 3:** Zulässige Höchstgeschwindigkeit und Haltestellenpositionen für die beiden CleanER-D Strecken Suburban und Regional Drive Cycle [5]

Für beide Fahrprofile wurden darüber hinaus für jede einzelne Haltestellenposition die zugehörige Ankunfts- und Abfahrtszeit definiert. Bei beiden Strecken sind keine Steigungen und Bogenradien der Gleise vorhanden. In der Fahrsimulation werden die Strecken zunächst von der Anfangs- bis zur Endhaltestelle zurückgelegt, danach macht das Fahrzeug eine 20-minütige Pause bevor die Strecke in die andere Richtung durchfahren wird. Während den Haltestellenstops und innerhalb der 20-minütigen Pause an der Endhaltestelle wird das Fahrzeug nicht abgestellt. Dies entspricht nach Rücksprache mit Fahrzeugbetreibern dem allgemeinen Betriebsverhalten.

### 3.2 Fahrweise

Im Folgenden wird das typische Fahrverhalten eines Schienenfahrzeugführers aufgezeigt, welches sich in der Praxis als die verbrauchsgünstigste Fahrweise erwiesen hat und dementsprechend in dem Simulationstool hinterlegt ist. Eine Fahrt zwischen zwei Haltestellenpunkten gliedert sich demnach in vier Abschnitte: Traktion, Sägezahnfahrt, Segeln und Bremsen. In Abbildung 4 sind anhand eines Simulationsergebnisses diese einzelnen Abschnitte gekennzeichnet.



**Abbildung 4:** Typisches Fahrprofil eines Schienenfahrzeugs zwischen zwei Haltestellen mit den dazugehörigen Abschnitten 1-4

Zunächst beschleunigt das Fahrzeug mit der maximal möglichen Zugkraft bis die zulässige Höchstgeschwindigkeit auf der Strecke erreicht wird (1), danach wird anhand einer Sägezahnfahrt die Geschwindigkeit gehalten (2). Dabei lässt der Fahrer zunächst das Fahrzeug rollen in dem er den Lasthebel zurücknimmt. Wenn die aktuelle zulässige Höchstgeschwindigkeit um 5 % unterschritten wird, beschleunigt der Fahrer das Fahrzeug erneut. Aufgrund der hohen Trägheit eines Schienenfahrzeugs und dem damit verbundenem geringen Geschwindigkeitsabfall in

Rollphasen erweist sich dieses Verhalten als sehr verbrauchsgünstig. Das Fahrprofil erinnert dabei an ein Sägezahnprofil, was der Fahrweise die Bezeichnung „Sägezahnfahrt“ gegeben hat. In einem definierten Abstand zur nächsten Haltestelle nimmt der Fahrer den Lasthebel auf Nullstellung zurück und lässt das Fahrzeug bis zu einer bestimmten Geschwindigkeit ausrollen, dies wird als Segeln bezeichnet (3). Anschließend findet die Bremsung in den Zielbahnhof statt (4). Die Distanzen zur Haltestelle in welcher das Segeln (3) und die Bremsung (4) eingeleitet werden, sind von der zu erreichenden Fahrzeit abhängig und werden entsprechend der Fahrpraxis des Fahrers gewählt.

### 3.3 Fahrsimulation

Für die Fahrsimulation bei Voith Turbo wird das Simulationsmodell des Antriebssystems mit einem Modell des Fahrzeuges verknüpft. Dieses lässt sich mit den notwendigen Parametern, welche für die Fahrwiderstandsberechnung notwendig sind, parametrisieren. Ein Simulationsmodell eines virtuellen Fahrers bildet die bereits beschriebene Fahrweise mit den einzelnen Abschnitten zwischen den Haltestellen ab. In der folgenden Tabelle 1 sind die wichtigsten für die Fahrsimulation verwendeten Parameter des betrachteten DMU aufgeführt.

<b>Beschreibung</b>	Zweiteiliger DMU mit 2 Antriebsanlagen
<b>Motor</b>	Voith R2876T3
<b>Motorleistung</b>	390 kW bei 1800 1/min
<b>Getriebe</b>	Voith Diwa 864.5
<b>Eingangsleistung</b>	320 kW bei 2000 1/min
<b>Fahrzeugmasse</b>	105 t
<b>Adhäsionsmasse</b>	72 t
<b>Radsatzgetriebe-übersetzung</b>	3.0492
<b>E-Motorübersetzung</b>	2.0348
<b>Treibraddurchmesser</b>	0.79 m
<b>Max. Geschwindigk.</b>	124.8 km/h
<b>Mittl. Verzögerung</b>	0.8 m/s <sup>2</sup>

**Tabelle 1:** Parameter des zweiteiligen DMU

Für die Fahrsimulation werden das Standardfahrzeug (Abk. S), welches keine Hybridkomponenten besitzt, und zwei hybride Fahrzeugkonfigurationen (Abk. K1 DSK und K2 LiIon) mit dem am Getriebeausgang angebrachtem Elektromotor und Energiespeicher betrachtet. Als Energiespeicher kommen in der ersten Konfiguration zwei DSK-Module mit insgesamt 1 kWh Energieinhalt und in der zweiten Konfiguration zwei Li-Ionen-Batteriemodule mit insgesamt 26 kWh Energieinhalt zum Einsatz.

	<b>Konfiguration 1 (K1 DSK)</b>	<b>Konfiguration 2 (K2 LiIon)</b>
<b>Energiespeicher</b>	2 DSK-Module	2 Li-Ionen-Batteriemodule
<b>Energieinhalt</b>	2 · 0.5 kWh	2 · 13 kWh
<b>Zus. Gewicht</b>	2 · 2.0 t	2 · 1.5 t

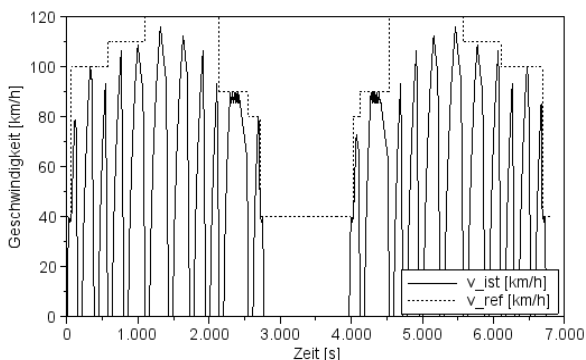
**Tabelle 2:** Parameter der beiden hybriden Konfigurationen K1 DSK und K2 LiIon

### 3.4 Hybride Antriebsstrategie

Für hybride Antriebssysteme ergibt sich ein weiteres Kriterium, neben dem Einhalten der vorgeschriebenen Fahrzeiten, um auf einen aussagekräftigen Kraftstoffverbrauch zu kommen. In der Fahrsimulation wird zusätzlich berücksichtigt, dass am Ende der Hin- und Rückfahrt beim erneuten Erreichen der Starthalttestelle der Ladezustand des Energiespeichers dem zu Beginn der Simulation entspricht. Dies ist notwendig da bei den ersten Betrachtungen keine externe Ladung vorgesehen ist. Um dies zu erreichen wird ein Parameter *boost* eingeführt. Während Traktionsphasen (1 und 2 in Abbildung 4) gibt der Elektromotor zusätzlich zum Dieselmotor Moment an den Getriebeausgang ab. Um den Energiespeicher während der Simulation nicht zu tief zu entladen und um auf einen ausgeglichenen Ladezustand zu kommen, wird das abgegebene Moment über den Faktor *boost* begrenzt, welcher das Verhältnis zwischen dem abgegebenen Drehmoment zum aktuell möglichem Drehmoment des Elektromotors zum angibt. Bei der Konfiguration mit Doppelschichtkondensator (DSK) als Energiespeicher wird der Faktor *boost* auf 1 gesetzt, da bei jedem Abbremsvorgang gewährleistet ist dass der Speicher wieder vollständig geladen wird.

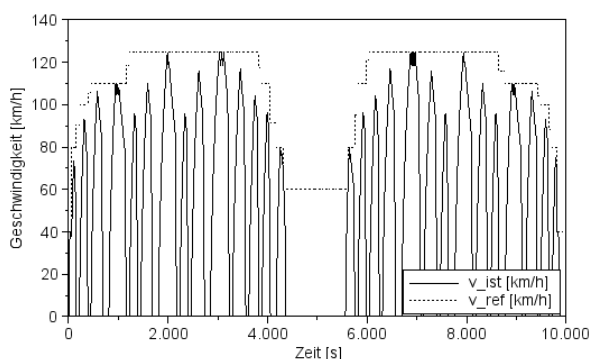
## 4 Simulationsergebnisse

Um eine Referenz für die beiden hybriden Fahrzeugkonfigurationen zu erzeugen, wurde in den ersten beiden Simulationen die Strecken Suburban Drive Cycle und Regional Drive Cycle mit dem Standardfahrzeug ohne Hybridkomponenten durchfahren. Abbildung 5 und Abbildung 6 zeigen den Geschwindigkeitsverlauf des Standardfahrzeugs auf den beiden betrachteten Strecken.



**Abbildung 5:** Geschwindigkeitsverlauf für das Standardfahrzeug auf dem Suburban Drive Cycle

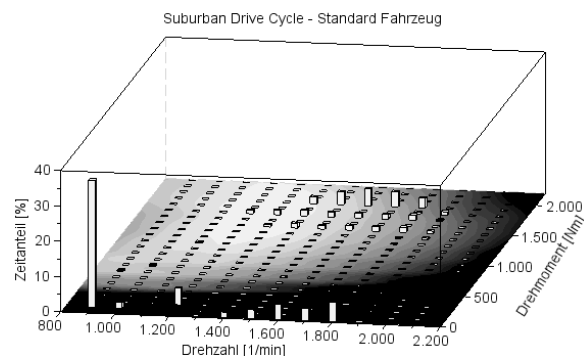
Für den Suburban Drive Cycle ergab sich dabei ein kumulierter Kraftstoffverbrauch von 107.4 l und für den Regional Drive Cycle 155.7 l. Auf die Strecke bezogen entsprechend 1.34 l/km bzw. 1.11 l/km. Bei den Darstellungen über der Zeit ist deutlich die 20-minütige Pause vor der Rückfahrt zu erkennen. Da der simulierte DMU auf eine maximale Geschwindigkeit von 124.8 km/h ausgelegt ist (vgl. mit Tabelle 1), wurde die zulässige Höchstgeschwindigkeit des Regional Drive Cycle auf diese Geschwindigkeit begrenzt (s. Abbildung 6).



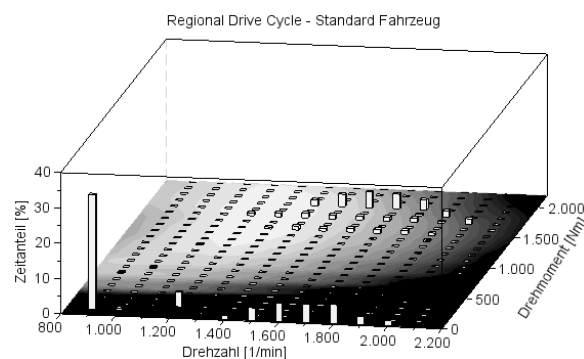
**Abbildung 6:** Geschwindigkeitsverlauf für das Standardfahrzeug auf dem Regional Drive Cycle

Aufgrund des größeren mittleren Haltestellenabstandes von 5.0 km des Regional Drive Cycle im Ver-

gleich zu den 3.6 km des Suburban Drive Cycle ergibt sich ein um 17 % niedrigerer Durchschnittsverbrauch. Ein Grund dafür ist die an der Gesamtfahrzeit prozentual geringere Verweildauer in Haltestellen. Dies wird aus den Motorhistogrammen in Abbildung 7 und Abbildung 8 deutlich. Die beiden höchsten Balken stehen dabei für den Zeitanteil in Haltestellen, bei denen der Dieselmotor im Leerlauf mit 900 1/min läuft und entsprechend definierten Nebenverbraucher betreibt.



**Abbildung 7:** Motorhistogramm für das Standardfahrzeug im Suburban Drive Cycle



**Abbildung 8:** Motorhistogramm für das Standardfahrzeug im Regional Drive Cycle

Im nächsten Schritt wurden die beiden hybriden Fahrzeuge für den Suburban und den Regional Drive Cycle simuliert. Es ergeben sich die in Tabelle 3 dargestellten Verbräuche und im Vergleich zum Standardfahrzeug prozentuale Einsparungen.

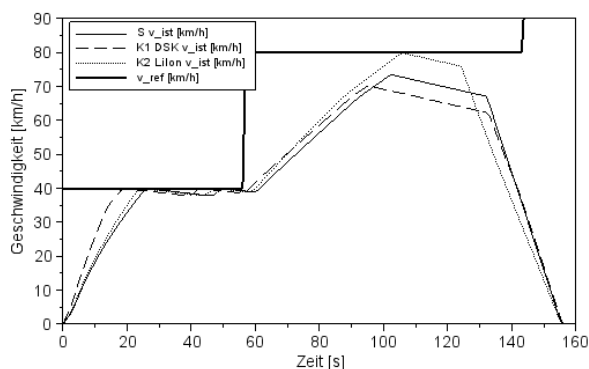
Für das betrachtete Fahrzeug und die analysierten Strecken zeigt sich, dass der Energiespeicher mit DSK-Modulen einen größeren Verbrauchsvorteil erzielt als der Energiespeicher auf LiIon-Basis. Dies liegt zum einen an dem um 2 t geringeren Fahrzeuggewicht (K1 DSK gegenüber K2 LiIon) und zum anderen an der verwendeten hybriden Antriebsstrategie. Die Elektromotoren des mit DSK ausgestatteten Fahrzeug-

ges geben beim Anfahren ihr volles Moment ab, da der *boost* Wert auf 1 gesetzt ist. Dadurch erreicht das Fahrzeug schneller die geforderte Geschwindigkeit und hat dadurch die Möglichkeit über eine größere Distanz zu segeln, was zu einem Verbrauchsvorteil gegenüber dem Standardfahrzeug führt. Darüber hinaus wird der Geschwindigkeitsbereich in welchem der Wandler des DIWA.5 Getriebes arbeitet, schneller durchfahren was ebenfalls zu einem Verbrauchsvorteil führt.

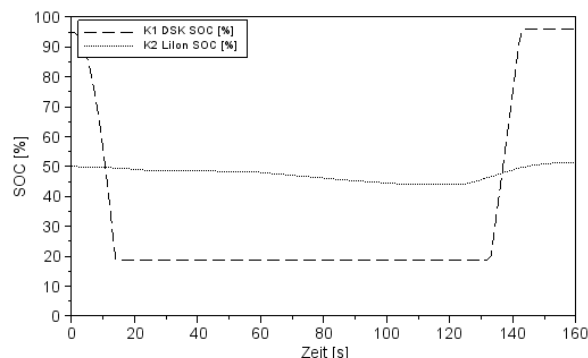
Strecke	Fahrzeug		
	S	K1 DSK	K2 LiIon
<b>Suburban Duty Cycle</b>	107.4 l	93.2 l (13.2 %)	101.5 l (5.5 %)
<b>Regional Duty Cycle</b>	155.7 l	141.9 l (8.9 %)	149.9 l (3.7 %)

**Tabelle 3:** Kumulierter Kraftstoffverbrauch [l] und prozentuale Einsparung [%] auf den beiden betrachteten Strecken für das Standardfahrzeug und die beiden Hybridkonfigurationen

Die größere Segeldistanz wird aus Abbildung 9 deutlich, in welcher der Geschwindigkeitsverlauf aus der Simulation für den Abschnitt 2 der Hinfahrt des Regional Drive Cycle für alle Fahrzeuge dargestellt wird. Beim Anfahren werden die DSK vollständig auf den minimal zulässigen Wert von 20 % Ladezustand (SOC) entladen (s. Abbildung 10). Beim Bremsen werden diese wieder vollständig auf 95 % SOC aufgeladen.



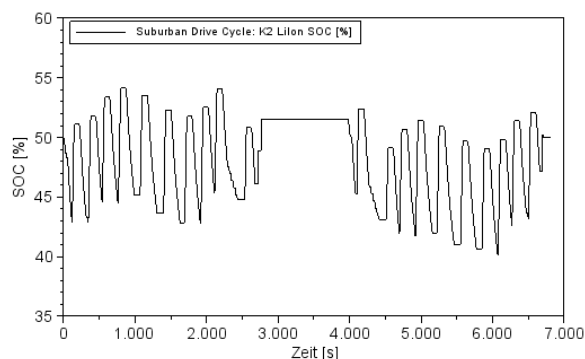
**Abbildung 9:** Geschwindigkeitsverlauf Regional Drive Cycle Hinfahrt Abschnitt 2 für die betrachteten Fahrzeugvarianten



**Abbildung 10:** SOC-Verlauf Regional Drive Cycle Hinfahrt Abschnitt 2 für die betrachteten Energiespeicher

Bei den LiIon-Batterien als Energiespeicher ist der *boost* Faktor herabgesetzt. Damit wird während Traktionsphasen der Dieselmotor unterstützt und am Ende der Simulation ist der ursprüngliche SOC von 50 % wiederhergestellt. (s. Abbildung 11 für den Suburban Drive Cycle).

Aus den Simulationen wird deutlich, dass sich die für die DSK verwendete Betriebsstrategie mit einem *boost*-Wert von 1 besser für das verwendete DIWA.5 Getriebe eignet als die Li-Ion-Module mit einem reduzierten *boost*-Wert. Allerdings wurde nur eine Betriebsstrategie untersucht. So bieten LiIon-Batterien aufgrund ihres großen Energiegehalts die Möglichkeit die Nebenverbraucher des Dieselmotors zu elektrifizieren und über die Batterie anzutreiben. Dadurch kann in Stillstandszeiten der Dieselmotor ausgeschaltet werden, was zu einer Verbrauchsreduzierung führt. Dies wäre auch eine Voraussetzung um das Fahrzeug rein elektrisch über die Batterie anzutreiben.



**Abbildung 11:** SOC-Verlauf Suburban Drive Cycle K2 LiIon

## 5 Zusammenfassung und Ausblick

Die multiphysikalische Simulation bietet die Möglichkeit hybride Antriebssysteme frühzeitig im Entwicklungsprozess auszulegen. Mithilfe eines Simulationstools wird Voith Turbo für verschiedene Kombinationen aus Fahrzeug, Antriebssystem, Fahrprofil und elektrischem Energiespeicher eine optimierte Konfiguration bestimmen. In einer ersten Untersuchung wurde ein zweiteiliger diesel-hydrmechanischer Triebwagen mit einem parallelem Hybridsystem (DIWAhybrid) untersucht. Als Energiespeicher wurden DSK-Module und Li-Ion Batterien gegenübergestellt. Für das betrachtete System ergibt sich mit DSK-Module auf zwei Referenzstrecken ein geringerer Kraftstoffverbrauch als mit LiIon Batterien.

In den nächsten Arbeitspaketen wird das Simulationstool um weitere Energiespeicher (z.B. Schwungradspeicher), Antriebssysteme (z.B. Diesel-Elektrik) und hybride Antriebsstrategien (z.B. rein elektrisches Anfahren) erweitert. Desweiteren wird das Post Processing um eine Wirtschaftlichkeitsbetrachtung ergänzt um die Mehrkosten durch ein hybrides Antriebssystem, die Lebensdauer des Energiespeichers und den reduzierten Kraftstoffverbrauch gegenüberzustellen.

## 6 References

- [1] Die Bundesregierung. *Nationaler Entwicklungsplan Elektromobilität der Bundesregierung*, S.18, [www.bmbf.de](http://www.bmbf.de) (Aufruf 2013-01-24), 2009.
- [2] Shiraki, N., Satou, H. und Arai, S.. *A Hybrid System for Diesel Railcar Series Ki-ha E200*, The 2010 International Power Electronics Conference, S. 2853-2858, 2010.
- [3] Lehmann, Ingo et al. *Hybrid-Powerpack für nachhaltigen und umweltfreundlichen Triebwagenantrieb*, Eisenbahntechnische Rundschau Nr. 9, September 2011, S. 18-22, 2011.
- [4] Voith. *DIWA.5 Fahrkomfort mit Wirtschaftlichkeit verbinden*, Druckschrift, [www.voith.com](http://www.voith.com) (Aufruf 2013-01-24), 2013.
- [5] CleanER-D. *Deliverable 7.2.1: Detailed Specification: Parameters definition*, <https://secure.cnc.it/cleaner-d/Docs/CLD-D-UIC-012-03.zip> (Aufruf 2013-01-24), S.6-7, 2011.





# Auslegung eines RangeExtender-Antriebsstrangs mit Hochtemperatur-PEM-Brennstoffzelle

Mounir Nasri<sup>1</sup>, Holger Dittus<sup>1</sup>

<sup>1</sup>Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR) (German Aerospace Centre)

Institut für Fahrzeugkonzepte (Institute of Vehicle Concepts)

*mounir.nasri@dlr.de*

Batterieelektrische Fahrzeuge, die für den urbanen Verkehr ebenso wie für längere Strecken geeignet sein sollen, werden mit einem RangeExtender ausgerüstet sein. Die Brennstoffzelle als RangeExtender stellt eine vielversprechende Lösung zur Substituierung des verbrennungsmotorisch betriebenen RangeExtenders dar. In dieser Arbeit wird ein solches System ausgelegt. Im Fokus stehen die Kopplung zwischen Batterie, Hochtemperatur-PEM-Brennstoffzelle und einem Wasserstoff-Metallhydridspeicher.

Anhand vorhandener Systemmodelle wie z. B. Brennstoffzelle, Batterie und Metallhydridspeicher wird mittels der AlternativeVehicles-Bibliothek ein Gesamtfahrzeugmodell entwickelt, wobei insbesondere thermische Modelle zu erweitern und parametrieren sind. Für die thermische Kopplung zwischen Hochtemperaturbrennstoffzelle und Wasserstoffspeicher wird ein Hochtemperatur-Kühlkreislauf modelliert. Die Batterietemperatur hingegen wird im Modell mit Hilfe eines Niedertemperatur-Kühlkreislaufs abgebildet. In Betracht der Herausforderungen, die heute an moderne KFZ gestellt werden, können mehrere Szenarien wie z.B. Kaltstart oder Warmbetrieb erstellt werden. Für diese Szenarien werden geeignete Betriebsstrategien entwickelt und in das Gesamtfahrzeugmodell integriert. Anhand verschiedener Fahrzyklen wird das thermische und elektrische Gesamtsystem-Verhalten untersucht.

## 1 Einleitung

Um die von der EU-Kommission vorgegebene Emissionsminderung von 60 % gegenüber 1990 im Verkehrsbereich zu erreichen, sind alternative Antriebskonzepte mit geringeren Emissionen als die bisherigen, auf Verbrennungsmotoren basierenden Antriebe, erforderlich.

In dem von der Europäischen Kommission vorgelegten Weißbuch Verkehr 2011 konkretisiert die EU-Kommission die Ziele im Verkehrsbereich, danach soll bis 2030 der Anteil der mit konventionellen Kraftstoffen betriebenen PKW halbiert und in den Städten bis 2050 auf null reduziert werden [1].

Eine vielversprechende Lösung sind batterieelektrische Fahrzeuge mit Brennstoffzellen-RangeExtender, die für den urbanen Verkehr ebenso wie für längere Strecken geeignet sind. Sie vereinen die Vorteile von verbrennungsmotorischen Hybridantrieben hinsichtlich Komfort und Reichweite mit denen von reinen Elektrofahrzeugen hinsichtlich der Emissionsfreiheit.

Am Institut für Fahrzeugkonzepte des Deutschen Zentrums für Luft- und Raumfahrt (DLR) wird in Zusammenarbeit mit dem Institut für Technische

Thermodynamik ein fahrzeugtauglicher Antriebsstrang mit Hochvolt-Batterie, Hochtemperatur-PEM-Brennstoffzelle (HT-BZ) und Metallhydridspeicher, für die Wasserstoffspeicherung, entwickelt und getestet. Auf Grundlage vorhandener Systemmodelle wird mit Hilfe der DLR-eigenen Modelica-Bibliothek AlternativeVehicles (AV) [2] der RangeExtender-Antriebsstrang modelliert und das dazu passende Thermomanagementkonzept entwickelt. Um die thermischen und elektrischen Anforderungen verschiedener Fahrzyklen an das Gesamtsystem zu ermitteln, wird der Antriebsstrang in das Hotzenblitz-Fahrzeugmodell [3] integriert. Die technischen Daten von Hotzenblitz sind in der Tabelle 1 zusammengefasst.

Klasse	Kleinstwagen
Motorleistung	12 kW
Leergewicht	830 kg

**Tabelle 1.** Kenndaten Hotzenblitz

## 2 Komponenten des RangeExtender-Antriebsstrangs

### 2.1 Brennstoffzellensystem

Das Brennstoffzellen-Gesamtsystem besteht aus 4 Modulen (gestrichelter Kasten) mit jeweils drei Stacks à 89 Zellen. Wie in Abbildung 1 dargestellt, sind die Stacks eines Moduls untereinander in Reihe geschaltet. Ein Strang besteht aus der seriellen Anordnung von zwei Modulen. Die beiden resultierenden Einzelstränge sind parallel verschaltet.

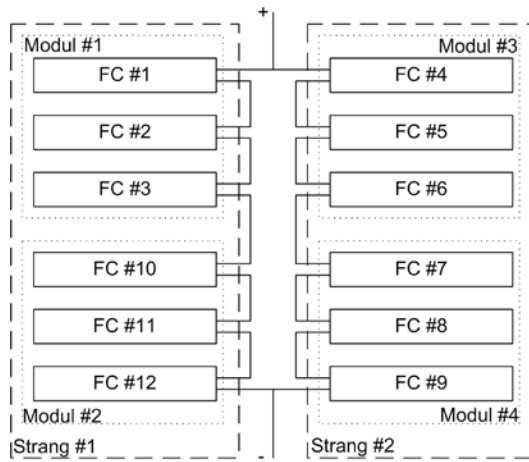


Abbildung 1. Verschaltung der Brennstoffzellen-Module

Für die Modellierung der Brennstoffzelle wird ein kennfeldbasiertes Modell verwendet [4]. Dieses Modell ist für Systemsimulationen, bei denen eine hohe Rechengeschwindigkeit gefordert ist, gut geeignet [2]. Die Polarisationskennlinie wird aus Messungen und Datenblättern übernommen. Für die thermische Simulation ist die Vorgabe der thermischen Masse von großer Bedeutung, da die Verlustleistung zur Änderung der Temperatur der thermischen Masse führt. Die thermische Verlustleistung der Brennstoffzelle ist proportional zur Spannungsdifferenz von Leerlaufspannung ( $U_0$ ) und tatsächlicher Spannung ( $U_{Stack}$ ) [5] und wird wie folgt berechnet

$$P_{Verlust} = (N \cdot U_0 - U_{Stack}) \cdot I$$

mit  $N$  Anzahl der Zellen,  $U_0 = 1,23$  V und  $I$  der Batteriestrom

### 2.2 Metallhydridspeicher

Das Sorptionsspeicher-Modell ist ein 0-dimensionales Speichermodell und basiert auf Gleichgewichtsreak-

tionen [6]. Die Massenbilanz wird durch die folgende Gleichung beschrieben:

$$\frac{1}{RT} \frac{\partial p}{\partial t} - \frac{p}{RT^2} \frac{\partial T}{\partial t} + \frac{\dot{n}}{\pi(R_o^2 - R_i^2)L} + \frac{1-\varepsilon}{\varepsilon} \frac{\rho_s}{2MMH} \frac{\partial q}{\partial t} = 0$$

wobei  $p$  der Druck im Reaktionsbett,  $T$  die gemittelte Wasserstofftemperatur im Metallhydridbett,  $R$  die universelle Gaskonstante,  $\dot{n}$  der Wasserstoffmassenstrom,  $R_i$  der Innenradius,  $R_o$  der Außenradius,  $L$  die Länge des Reaktionsbets,  $\varepsilon$  die Porosität des Metallhydridbets,  $\rho_s$  die Dichte des Metallhydrids,  $MMH$  die gemittelte molekulare Masse des Metallhydrids und  $q$  die Wasserstoffkonzentration im Metallhydrid ist.

Die Energiegleichung lautet:

$$\left( \frac{p}{RT} C_p H_2 + \frac{1-\varepsilon}{\varepsilon} \rho_s C_{ps} \right) \frac{\partial T}{\partial t} + \frac{\partial p}{\partial t} + \frac{1-\varepsilon}{\varepsilon} \frac{\rho_s \Delta H}{2MMH} \frac{\partial q}{\partial t} + \frac{2h_{bed} R_i}{\varepsilon(R_o^2 - R_i^2)L} (T - T_m) = 0$$

wobei  $C_p H_2$  die Wärmekapazität von Wasserstoff,  $C_{ps}$  die Wärmekapazität des Metallhydrids,  $H$  die isostere Adsorptionswärme des Wasserstoffs im Metallhydrid,  $h_{bed}$  der Gesamtwärmeübertragungskoeffizient zwischen Kühlmedium und Metallhydridbett und  $T_m$  die Temperatur des Kühlmediums ist.

Durch Anpassung der geometrischen Größen Länge oder Radius kann der Metallhydridspeicher skaliert werden.

### 2.3 Batterie

Parallel zu dem BZ-Gesamtsystem sind Akkus geschaltet. Da die Systemspannung des BZ-Systems im Bereich 350...400 V liegt, wurde das Akkupack aus 93 seriell verschalteten Li-Ion-Zellen aufgebaut. Die Ladeschlussspannung der Zellen liegt bei 4,0 V, die gesamte Ladeschlussspannung damit bei 372 V

Die AV-Bibliothek stellt zwei unterschiedliche Batteriemodelle zur Verfügung (ein kennfeldbasiertes und ein impedanzbasiertes Modell) [6]. Das kennfeldbasierte Batteriemodell wird hier für die Modellierung der Einzelzellen benutzt. Mit Hilfe von Kennfeldern aus Datenblättern werden abhängig von Ladezustand, Strom und Temperatur die entsprechenden Werte für den Innenwiderstand und die Ruhespannung vorgegeben.

### 3 Thermisches Modell

Das Kühlsystem für den RangeExtender-Antriebsstrang besteht aus einem Hochtemperatur- und Niedertemperatur-Kühlkreislauf und wird in der Simulationsumgebung Dymola umgesetzt. Der Hochtemperatur-Kühlkreislauf stellt den thermischen Kreislauf des Brennstoffzellensystems dar und besteht aus Hochtemperaturbrennstoffzelle, Metallhydridspeicher, Pumpe, Hochtemperaturkühler und den zugehörigen Leitungen (siehe Abbildung 2). Die Arbeitstemperaturen der HT-BZ liegen zwischen 100 °C und 160 °C. Der Auslösetemperaturbereich, in dem Wasserstoff aus dem Alanat des Metallhydridspeichers gelöst wird, beginnt bei 100 °C. Durch einen Thermoölkreislauf werden die beiden Komponenten miteinander gekoppelt. Als Kühlmittel wird UCOTHERM S-15-A verwendet. UCOTHERM ist eine synthetische, wasserfreie, jedoch wasserlösliche Wärmeträgerflüssigkeit aus Polyalkylenglykol-Derivaten und speziell für den Einsatz mit Vorlauf-Temperaturen zwischen +100 °C und +205 °C vorgesehen. Wegen des produktspezifisch sehr hohen Viskositätsindexes ist die Flüssigkeit auch bei tiefen Temperaturen verwendbar [7].

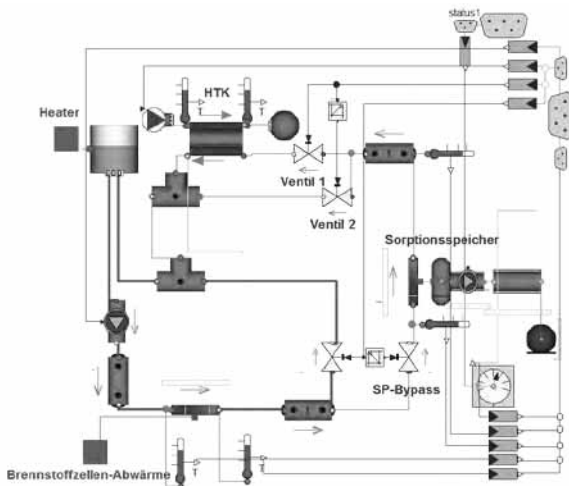


Abbildung 2. Hochtemperatur-Kühlkreislauf

Die Batterietemperaturierung wird im Modell mit Hilfe eines Niedertemperatur-Kühlkreislaufts mit dem Kühlmittelgemisch Wasser/Glykol abgebildet. Aus Leistungs- und Lebensdauergründen soll die Batterie nur in einem bestimmten Temperaturfenster betrieben werden: Ab Betriebstemperaturen von 50 °C wird die Batterielebensdauer reduziert und eine Leistungsmin- derung ist feststellbar. Bei sehr niedrigen Temperaturen unter ca. 10 °C lässt die Leistung der Batterie ebenfalls deutlich nach und der Wirkungsgrad sinkt.

Wie es in **Abbildung 3** zu erkennen ist, verbindet der Batteriekreislauf die elektrischen Komponenten Batterie und Motor (EM) mit einem Niedertemperatur-kühler (NTK), der die überschüssige Verlustwärme an die Umgebung überträgt. Die dargestellten Ventile (Ventil 1, 2 und 3) werden für die Zu- und Abschal- tung einiger Kühlkreiszeige abhängig vom aktuel- len Betriebszustand verwendet: Wenn ein schnelles Aufheizen der Batterie bei fehlender Motorabwärme gewünscht ist, wird nur das Ventil 2 geöffnet. Zur Abfuhr der Batterieabwärme wird nur das Ventil 1 geöffnet.

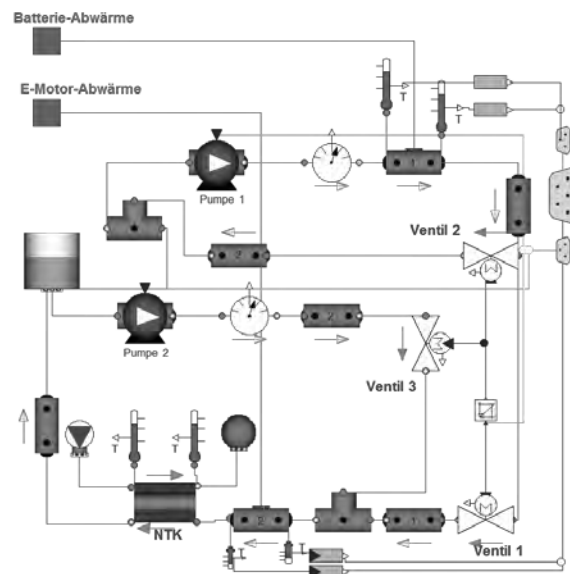


Abbildung 3. Niedertemperaturkreislauf

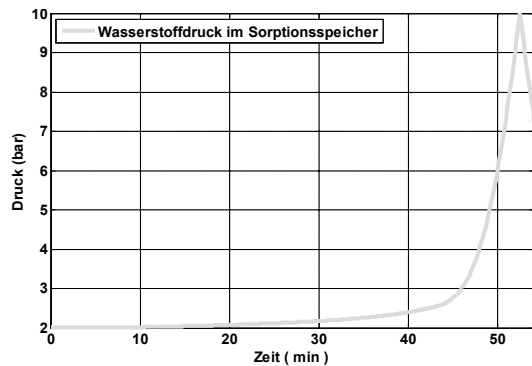
### 4 Betriebsstrategien

Für die folgenden thermischen Szenarien Kaltstart, hohe Betriebstemperaturen und Leistungssprung wer- den Betriebsstrategien entwickelt.

#### 4.1 Kaltstart

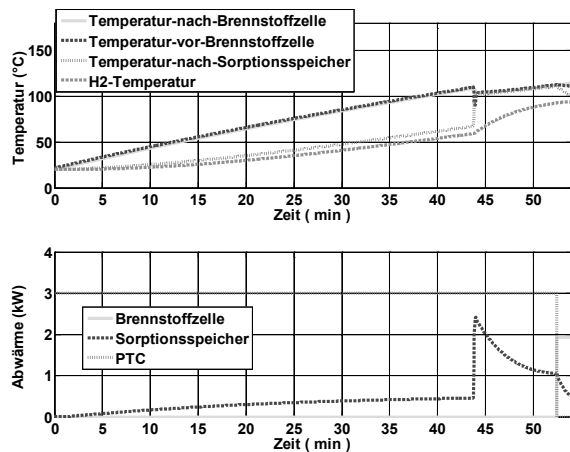
Um die Temperatur der Brennstoffzelle zuerst auf 110 °C zu erhöhen, wird in der Startphase nur der rot eingezeichnete kleine Kühlkreis zugeschaltet (Abbildung 2). Durch eine externe Energiequelle (z. B. PTC-Heater) wird das Kühlmittel geheizt. Danach wird der Sorptionsspeicher-Bypass (SP-Bypass) voll geöffnet, um den Speicher schnell zu erwärmen. In- folge der Erwärmung wird Wasserstoff freigesetzt. Da die Brennstoffzelle noch nicht aktiv ist und damit kein Wasserstoff verbraucht wird, erhöht sich der Druck im Tank (siehe Abbildung 4). Bei einem Was- serstoffdruck von ca. 10 bar im Speicher erfolgen die

Abschaltung der elektrischen Heizung und der Start der Brennstoffzelle.



**Abbildung 4.** Wasserstoffdruckverlauf beim Aufheizen des Metallhydridspeichers

Wie in Abbildung 5 zu erkennen ist, dauert es mit einer 3 kW PTC-Leistung ungefähr 43 Minuten, bis die Brennstoffzelle eine Betriebstemperatur von ca. 110 °C erreicht. Die Aufheizzeit eines Brennstoffzellensystems mit 50 kg Gesamtmasse und 900 J/kgK thermischer Kapazität von Umgebungstemperatur auf 110 °C ist von der Heizleistung abhängig und liegt im Falle 20 °C Umgebungstemperatur zwischen 5 Minuten für 18 kW Heizleistung und 30 Minuten für 4 kW.



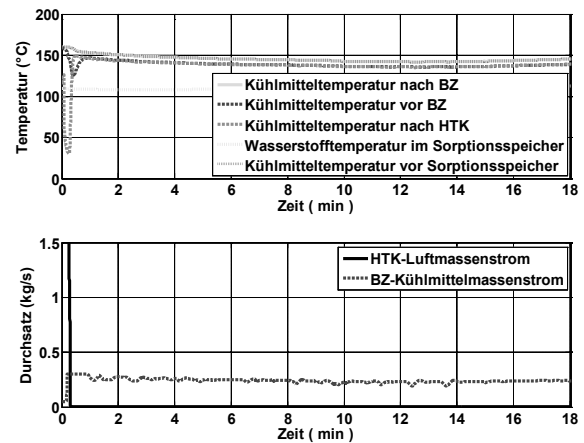
**Abbildung 5.** Temperatur- und Wärmeentwicklung während der Start-Phase

#### 4.2 Warmbetrieb

Die Hauptaufgabe in der Warmbetrieb-Phase besteht darin, die Temperatur der Brennstoffzelle im empfohlenen Betriebsbereich zu halten. Damit die obere Grenztemperatur (160 °C) nicht überschritten wird, öffnet sich der Thermostat (Ventil 1 in der Abbildung 2) bei 150 °C. Damit wird der Hochtemperaturkühler

mit in den Kreislauf eingeschlossen und die überschüssige Wärme an die Außenluft abgegeben. Alternativ kann diese Abwärme auch für die Aufheizung des Innenraums verwendet werden. Der Luftmassenstrom durch den Hochtemperaturkühler wird so geregelt, dass die Kühlmittelintrittstemperatur der Brennstoffzelle 155 °C beträgt.

Abbildung 6 zeigt, dass nachdem die Betriebstemperatur der Brennstoffzelle 160 °C erreicht hat, eine zusätzliche Kühlung durch den Luftwärmetauscher im Falle einer simulierten Fahrt auf dem Artemis-Road-Fahrzyklus nicht notwendig ist (der Luftmassenstrom durch den Hochtemperatur ist gleich 0 kg/s).



**Abbildung 6.** Temperatur und Massenstrom des Kühlmittels, Fahrprofil: ARTEMIS\_Road

#### 4.3 Leistungssprung

Das untere Diagramm der Abbildung 7 zeigt, dass bei einem Speichervolumen von ungefähr 0.022 m<sup>3</sup> die Artemis-Motorway-Fahrt nach 6 Minuten abgebrochen wird, da der Druck des gasförmigen Wasserstoffs zu gering ist und damit der Sorptionsspeicher nicht in der Lage ist, den für die BSZ zusätzlich benötigten Wasserstoff zu liefern. Die Verdopplung des Speichervolumens führt dazu, dass genügend gasförmiger Wasserstoff während der Fahrt im Sorptionsspeicher freigesetzt wird.

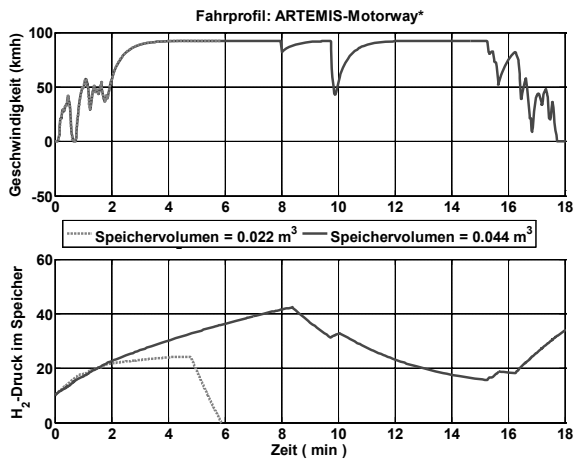


Abbildung 7. Ist-Geschwindigkeit, H<sub>2</sub>-Druck im Speicher, Fahrprofil: ARTEMIS\_Motorway<sup>1</sup>

## 5 Variantenrechnung Gesamtsystem

Wie in Abbildung 8 gezeigt, werden in das Gesamtfahrzeugmodell vom Hotzenblitz-Fahrzeug mit einer Gesamtmasse von ungefähr 1050 kg die beiden Kühlmittel-Kreisläufe, der RangeExtender-Antriebsstrang und das ThermoRegler-Modul integriert. Der originale Elektromotor GT 20 mit einer Dauerleistung von 12 kW wird nicht ersetzt.

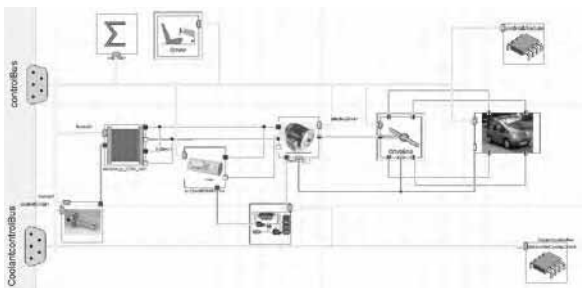


Abbildung 8. Gesamtfahrzeugmodell inklusiv Kühlmittelkreisläufen und ThermoRegler

Für die Simulation wird beispielhaft der neue Europäische Fahrzyklus (NEFZ) gewählt. Der Geschwindigkeitsverlauf über die Zeit ist in Abbildung 9 dargestellt. Wie es im Bild zu erkennen ist, beträgt die maximale erreichbare Geschwindigkeit aufgrund der geringen Antriebsleistung des Hotzenblitz-Fahrzeugs lediglich 90 kmh.

Als beispielhaftes Ergebnis sind in Abbildung 10 die berechneten Leistungen der Brennstoffzelle und der Batterie aufgetragen.

<sup>1</sup> Aufgrund 12 kW Motor-Dauerleistung beträgt die maximale Geschwindigkeit des Hotzenblitz ca. 90 kmh

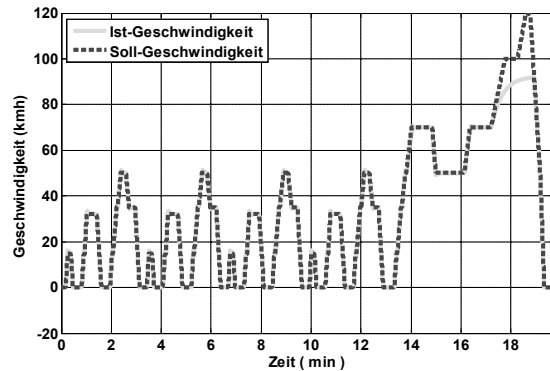


Abbildung 9. Geschwindigkeitsverlauf im NEFZ

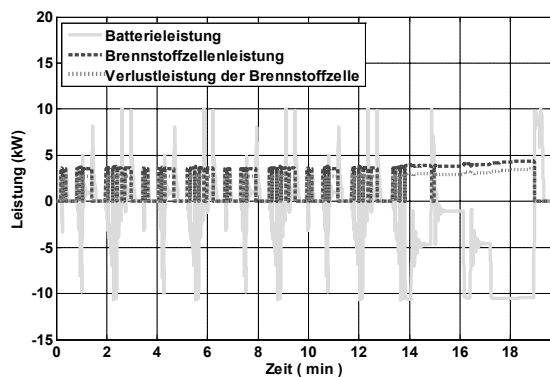


Abbildung 10. Leistungen Batterie und Brennstoffzellensystem (Fahrprofil NEFZ)

In Abbildung 11 ist der Verlauf der Kühlmitteltemperatur am Ein- und am Austritt des Brennstoffzellensystems aufgetragen. Wie man erkennt, ist die Temperatur der Brennstoffzelle in dem vom Hersteller empfohlenen Betriebsbereich (zwischen 100 °C und 160 °C) gehalten. Die Ergebnisse verschiedener Fahrprofile wie z. B. der H<sub>2</sub>-Verbrauch und die elektrische Energie aus der Batterie bezogen auf 100 km sind in der Tabelle 2 zusammengefasst.

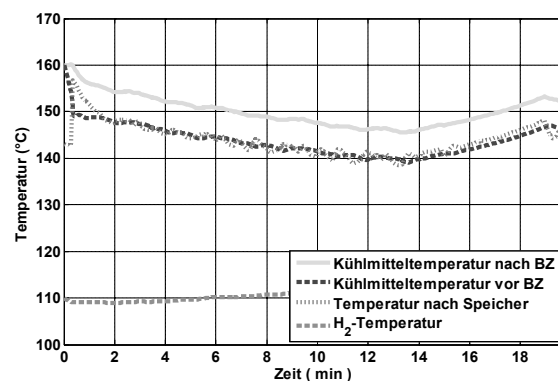


Abbildung 11. Temperaturverläufe im HT-Kühlkreis (Fahrprofil NEFZ)

	NEFZ	ARTEMIS Urban	FTP75
<b>H<sub>2</sub>-Verbrauch (kg/100 km)</b>	0.47	0.62	0.46
<b>Wärme für H<sub>2</sub>- Freisetzung (kwh/100 km)</b>	8.77	16.60	8.23
<b>Energie aus der Batterie (kwh/100 km)</b>	7.17	10.80	7.20
<b>Gesamtenergie- bedarf (kwh/100 km)</b>	13.40	18.26	13.60

**Tabelle 2.** Ergebnisse verschiedener Fahrprofile bezogen auf 100 km

## 6 Zusammenfassung

Mit der Alternative Vehicles-Bibliothek des DLR wurde ein kompletter Range Extender-Antriebsstrang mit Hochtemperatur-Brennstoffzelle und thermischem Kreis modelliert und in ein Fahrzeugmodell integriert. Mit dem erstellten Modell wurden verschiedene Szenarien wie Kaltstart und Warmbetrieb untersucht und die dazu passenden Betriebsstrategien entwickelt.

Das termische und elektrische Gesamtsystemverhalten wurde anhand verschiedener Fahrzyklen untersucht. Dabei konnte gezeigt werden, dass je nach Fahrzyklus die Abwärme der Brennstoffzelle nicht ausreicht, um bei einem Leistungssprung schnell genug eine ausreichende Menge Wasserstoff aus dem Metallhydridspeicher auszulösen. Dieses Problem konnte durch Vergrößerung des Speichervolumens gelöst werden.

Zur Berücksichtigung der für die Kabinenklimatisierung notwendigen Energie ist in einem nächsten Schritt geplant, zusätzlich ein Innenraummodell zu erstellen und ins Gesamtfahrzeugmodell zu integrieren. Mit diesem thermischen Gesamtmodell können dann weiterführende Untersuchungen zum Energie- und Thermomanagement an Brennstoffzellenfahrzeugen durchgeführt sowie geeignete Betriebsstrategien, die auch die Beheizung der Kabine berücksichtigen, entwickelt werden.

## 7 Abkürzungen

AV	Alternative Vehicles-Bibliothek
EM	Elektrischer Motor
HT-BZ	Hochtemperaturbrennstoffzelle
HTK	Hochtemperaturkühler
HV	Hochvolt
NEFZ	Neuer Europäischer Fahrzyklus
NTK	Niedertemperaturkühler
PEM	Polymer-Elektrolyt-Membran
PTC	Positiver Temperaturkoeffizient
SOC	State of Charge (Ladezustand)

## 8 Referenzen

- [1] EU-Kommission. *Fahrplan zu einem einheitlichen europäischen Verkehrsraum - Hin zu einem wettbewerbsorientierten und ressourcenschonenden Verkehrssystem*. Brüssel, 28.3.2011.
- [2] J. Ungethüm; D. Hülsebusch; H. Dittus und T. Braig. *Simulation von alternativen Fahrzeugantrieben in Modelica*. ASIM Conference, Ulm 2010.
- [3] H. E. Friedrich. und P. Treffinger. *Hylite Technologieplattform für Brennstoffzellenhybride*. ATZ05, 2006.
- [4] D. Hülsebusch; J. Ungethüm und et al. *Multi-disziplinäre Simulation von Fahrzeugen*. ATZ10, 2009.
- [5] D. Wenger. *Metallhydridspeicher zur Wasserstoffversorgung und Kühlung von Brennstoffzellenfahrzeugen*. Dissertation, Technische Universität München, 2009.
- [6] Z. Jianga; R.A. Dougala und et al. *Simulation of a thermally coupled metal-hydride hydrogen storage and fuel cell system*. Journal of Power Sources 142, S. 92–102, 2005.
- [7] Fragol Schmierstoff GmbH & Co. KG. *Thermöl UCOTHERM Datenblatt*.

# Prozess kalibriertes Materialmodell für die Spritzgießsimulation von hochgefüllten, elektrisch und thermisch leitfähigen Thermoplasten

J. Dörner<sup>1</sup>, A. Swoboda<sup>1</sup>, J. Wortberg<sup>1</sup>

<sup>1</sup>Universität Duisburg-Essen, Institut für Produkt Engineering,

Lehrstuhl für Konstruktion und Kunststoffmaschinen

*antje.swoboda@uni-due.de*

## Zusammenfassung

Die Anwendung der Spritzgießsimulation in der Produktentwicklung ist eine gängige Methode, um beispielsweise Visualisierungseffekte sowie Zeit- und Kosteneinsparung zu erzielen. Die heutige Simulationssoftware kann angesichts vielfältiger Auswahlmöglichkeiten auf den entsprechenden Anwendungsfall zugeschnitten werden und liefert geeignete Ergebnisse.

Dieser Beitrag widmet sich der Spritzgießsimulation im Rahmen der Herstellung spritzgegossener Bipolarplatten. Diese stellen eine wichtige Wiederholeinheit in der Brennstoffzelle, welche zur Wandlung von chemischer in elektrische Energie ihren Einsatz findet, dar. Die Hauptfunktion einer Bipolarplatte innerhalb einer Brennstoffzelle lässt sich auf die thermische und elektrische Leitfähigkeit reduzieren. Zur Realisierung dieser wird ein überaus hoher Anteil verschiedener kohlenstoffbasierter Füllstoffe in das thermoplastische Matrixmaterial eingebracht. Dadurch werden Anpassungen bei der Spritzgießsimulation als Konsequenz von veränderten Werkstoffeigenschaften nötig. Besonderes Augenmerk wird auf die Messdatenüberführung mittels mathematischer Modelle beziehungsweise Funktionen der Scherviskosität, des pvT-Verhaltens, der spezifischen Wärmekapazität und der Wärmeleitfähigkeit gelegt, aus denen das materialspezifische Materialmodell erstellt wird. In diesem Beitrag werden die Ergebnisse eines prozess kalibrierten Materialmodells, bei dem die physikalischen Messdaten um Prozessdaten erweitert wurden, präsentiert. Durch die Bestimmung der effektiven Viskosität und den Effektivwerten des Wärmübergangs gelang es, den Werkzeuginnendruckverlauf in der Einspritz- und Nachdruckphase in einem Versuchsfenster aus Schneckenvorlaufgeschwindigkeit und Werkzeugtemperatur abzubilden. Die Prognosefähigkeit innerhalb der Kalibrierungsgrenzen wird anhand einer unkalibrierten Werkzeugtemperatur nachgewiesen.

## 1 Einleitung

In diesem Beitrag stehen thermoplastische Kunststoffe im Vordergrund, die durch die Zugabe von elektrisch leitfähigen Füllstoffen optimierte Eigenschaften erhalten. Bipolarplatten (BPP) erfordern für die Materialauswahl eine gute elektrische Leitfähigkeit, weshalb die eingesetzten thermoplastischen Kunststoffe ebenfalls elektrisch leitfähig sein müssen. BPP sind Serienartikel für Polymer-Elektrolyt-Membran-Brennstoffzellensysteme (PEM-BZ) und sollen vorzugsweise kostengünstig mit dem Urformverfahren Spritzgießen hergestellt werden. Bislang wurden BPP durch Heißpressen oder mechanische Bearbeitung hergestellt, wodurch der Stückpreis für einen Serienartikel unangemessen hoch ist. Die Veränderung der Eigenschaften typischer Thermoplaste durch die Füllung mit leitfähigen Füllstoffen erzeugt

neue Herausforderungen in der Prozessbeherrschung, der Maschinen- und Verfahrenstechnik sowie dem Rheologieverständnis. Das Thema findet internationales Interesse, da BPP zurzeit einen kostenintensiven Bestandteil einer Brennstoffzelle darstellen. Gelingt es, BPP serienmäßig mit den geforderten Eigenschaften im Spritzgießverfahren zu produzieren, könnte die PEM-BZ gerade für den Markt Elektromobilität interessant werden. Zu hohe Kosten und eine zu geringe Leistungsdichte verhindern seit einigen Jahren den vollständigen Durchbruch und die nötige Popularität der Brennstoffzelle [1]. Um diese Entwicklung zu unterstützen und geeignete Spritzgießwerkzeuge auf diese Anwendung auslegen zu können, wird die Spritzgießsimulation eingesetzt, womit Prozessdrücke während der Einspritzphase und der Abkühlvorgang berechnet werden können. Da es sich nicht mehr um reine Thermoplaste handelt, ist die Prognosefähigkeit

der Simulationssoftware eingeschränkt und die Spritzgießsimulation erfolgt mit Hilfe eines prozess kalibrierten Materialmodells.

### 1.1 Hochgefüllte Thermoplast-Compounds

Zur Herabsetzung des elektrischen Widerstandes von Thermoplasten kommen elektrische Füllstoffe zum Einsatz. Gegenstand der Untersuchung ist Polypropylen (PP) gefüllt mit einem großen Anteil Graphitpulver und Hochleitfähigkeitsruß. Die Füllstoffe bringen Perkolation mit sich, also einen starken Anstieg der elektrischen Leitfähigkeit ab einer bestimmten Füllstoffkonzentration, und Erhöhung der Wärmeleitfähigkeit. Sie führen aber auch zu höherer Viskosität und damit verschlechterter Fließfähigkeit und Verarbeitbarkeit sowie Problemen bei der Materialcharakterisierung und damit zur Beeinträchtigung der Simulationsfähigkeit dieser Materialien. Als Formteil wurde eine Rechteckplatte ( $60 \times 40 \times 4 \text{ mm}^3$ ) mit einem einfachen Kaltkanal in Stangenform ausgewählt. Die Rechteckplatte wurde bevorzugt, da weniger Material im Vergleich zur BPP benötigt wird und das Prozessfenster und die maschinenseitigen Variationsmöglichkeiten deutlich größer sind. Zur Überwachung des Formfüllvorgangs und Aufzeichnung von Vergleichsdaten für die Simulation wurde ein Werkzeuginnen-drucksensor nahe am Anguss positioniert.

### 1.2 Strategie der Materialmodellbildung: Prozess kalibriertes Materialmodell

Zur Abbildung des charakteristischen Materialverhaltens innerhalb der Simulation werden im ersten Schritt prozessrealistische Messungen der Viskosität, der Wärmekapazität, der Wärmeleitfähigkeit und des pVT-Verhaltens durchgeführt. Aufbauend auf diesen Stoffdaten, ergänzt durch Messdaten aus dem Prozess, wird ein Materialmodell erstellt, das nicht nur für einen Betriebspunkt Gültigkeit hat, sondern in einem Fenster mit definierten Grenzen gültig ist. Hierzu ist es notwendig, eine Auswahl wichtiger Spritzgießparameter zu definieren, die als Variablen für ein Versuchsfenster dienen. In dieser Arbeit wird diese Methodik fortlaufend anhand eines zweidimensionalen Versuchsfensters mit den Faktoren Schneckenvorlaufgeschwindigkeit und Werkzeugtemperatur exemplarisch beschrieben. Im nächsten Schritt werden die Grenzen (Levelstufen) der einzelnen Faktoren gewählt. Hierbei ist es wichtig, die Grenzen an den realen Spritzgießprozess anzupassen. Die Grenzen sollen daher nur so weit gewählt werden, wie der Spritzgießprozess realistisch durchgeführt werden

kann. Anhand einiger Variationsmöglichkeiten von Materialparametern ist es möglich, die Güte des Simulationsergebnisses anhand eines Optimierungskriteriums zu bewerten. In dieser Arbeit wurde der Fokus auf den Verlauf des Werkzeuginnendruckes während der Einspritz- und Nachdruckphase gelegt. Dieser Verlauf ist ohne die Kalibrierungsfunktion eine Schwachstelle der Simulation und führt zu einer fehlerhaften Berechnung der zeitlichen Ausbreitung der Schmelze. Es wird daher davon ausgegangen, dass zunächst dieser berechnete Verlauf dem gemessenen angepasst werden muss, bevor weitere Optimierungsschritte erfolgen. Ausführungen zu anderen üblichen Strategien der Materialmodellbildungen finden sich in [2].

## 2 Ermittlung der Material- und Prozessdaten

Viskositätsdaten in einem Schergeschwindigkeitsbereich zwischen  $10^1$  und  $3 \cdot 10^4 \text{ 1/s}$  wurden mit einem Hochdruckkapillarrheometer (HKR) für verschiedene Temperaturen ermittelt und Einlaufdruckverluste in Form der Bagley-Korrektur berücksichtigt. Die Messdaten zeigen einen rein strukturviskosen Verlauf und deuten auf eine sehr geringe Temperaturabhängigkeit des Materials hin. Innerhalb der Simulationssoftware wird die Viskosität des Materials durch das Cross-Arrhenius-Modell beschrieben. Zur Abbildung von hohen (Füllphase) und niedrigen (Nachdruckphase) Schergeschwindigkeiten gilt es einen Kompromiss der verwendeten Fits zu erreichen. Da mit dem HKR nur zuverlässige Daten für die Füllphase erzielt werden können, stellt der Bereich niedriger Schergeschwindigkeiten einen hohen Unsicherheitsfaktor innerhalb der Simulation dar.

Mit einem Kolbengerät wurden isobar für fünf verschiedene Druckstufen vT-Kurven des Abkühlprozesses, welcher der Abkühlphase von heißer Schmelze im Werkzeug des Spritzgießprozesses nahe kommt, ermittelt. Da mit diesem Charakterisierungsverfahren keine prozessnahe Abkühlung, also Kühlraten wie im Spritzgießwerkzeug, realisiert werden können, liegt die ermittelte Kristallisationstemperatur ( $148^\circ\text{C}$ ) über der bei realistischen Prozessbedingungen erwarteten. Die Schnittstelle zur Simulationssoftware stellt die Beschreibung der Messdaten mit dem Modified-Tait-Modell dar.

Mit Hilfe der dynamischen Differenzkalometrie (DSC) wurde die spezifische Wärmekapazität während der Abkühlung aufgezeichnet. Die charakteristi-



sche Peaktemperatur bei 132 °C bildet die erfasste Kristallisationstemperatur ab. Die Überführung in die Simulationssoftware erfolgte über 25 Datenpunkte, wobei für die Beschreibung des Kristallisationsbereich der Großteil der Datenpunkte verwendet wurde.

Die bei der pvT- bzw. DSC-Messung aufgezeichneten Kristallisationstemperaturen weichen aufgrund von unterschiedlichen Kühlraten von einander ab. Zur Vermeidung von Messfehlern innerhalb des Materialmodells wurde eine *effektive Kristallisationstemperatur*, welche der ermittelten aus der DSC-Messung entspricht, festgelegt. Die Kristallisationstemperatur wurde entlang der pvT-Kurven zu der tieferen Temperatur verschoben. Für die Modellbildung wird demnach nicht der komplette Kurvensatz verschoben. Druckabhängige Daten für die effektive Kristallisationstemperatur konnten nicht berücksichtigt werden.

Zur Vervollständigung aller relevanten Materialdaten wurde die Wärmeleitfähigkeit quer zur Richtung der Formfüllung mit der Laser-Flash-Methode ermittelt. Mit Hilfe der Wärmeleitfähigkeit können Annahmen und Korrekturen des Wärmeübergangskoeffizienten, welcher die Einstellung des Wärmeaustausches mit dem Spritzgießwerkzeug quer zur Fließrichtung beschreibt, getroffen werden und so das anisotrope Verhalten des Materials indirekt berücksichtigt werden. Eine Temperaturabhängigkeit der Wärmeleitfähigkeit kann ausgeschlossen werden [3].

Für den Aufbau eines zweidimensionalen Versuchsfensters wurden für die Faktoren Schneckenvorlaufgeschwindigkeit und Werkzeugtemperatur für je zwei Levelstufen (40 bzw. 80 mm/s; 110 bzw. 130 °C) alle relevanten Prozessdaten (Einspritzdruck-, Werkzeuginnendruckverlauf) zum Abgleich von der Spritzgießsimulation mit dem Experiment ermittelt. Für eine ausführliche Beschreibung zur Ermittlung der Material- und Prozessdaten sei auf [2] verwiesen.

### 3 Festlegung der Simulationsvariablen

Um die Ergebnisse der Spritzgießsimulation den gemessenen Zielgrößen anzupassen, werden einige Parameter im Material- und Prozessmodell als in Grenzen variabel definiert. Zu diesen Optimierungsvariablen zählen die Viskosität der Schmelze sowie der Wärmeübergangskoeffizient. Im Folgenden werden die ausgewählten Variablen vorgestellt.

#### 3.1 Viskosität

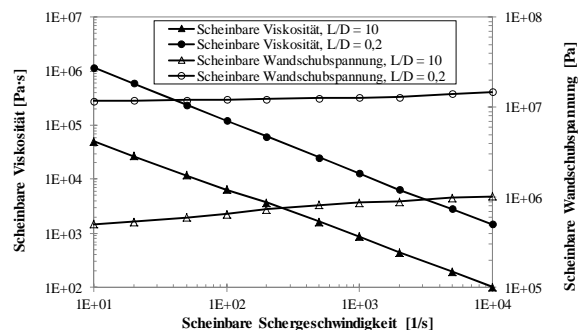
In Abschnitt 2 wurde das Cross-Arrhenius-Modell zur Beschreibung der Viskositätsfunktion innerhalb der Simulationssoftware genannt. Die folgenden Erläuterungen sollen eine Erklärung liefern, in welcher Form und mit welchem Hintergrund die Viskositätsdaten für die Simulation anhand des enthaltenen Modellparameters  $T_b$  verändert wurden.

Eine Messung bei 260 °C ergibt die in **Abbildung 1** dargestellten Mess- und Berechnungswerte für die Viskosität und die Wandschubspannung unter Anwendung der für newtonsche Flüssigkeiten gültigen Gleichungen (1) und (2) zur Ermittlung der scheinbaren Größen für die Schubspannung und die Schergeschwindigkeit.

$$\tau_{ap} = \frac{r}{2 \Delta l} \cdot \Delta p \quad (1)$$

$$\dot{\gamma}_{ap} = \frac{4}{\pi \cdot r^3} \cdot Q \quad (2)$$

Zur Berücksichtigung des strukturviskosen Verhaltens des Kunststoffes, wird anschließend die Weissenberg-Rabinowitsch Korrektur angewendet. Nach der Durchführung des Korrekturverfahrens werden die berechneten Viskositäten und Schubspannungen als wahre Größen bezeichnet. Neben der Weissenberg-Rabinowitsch Korrektur gibt es weitere prozessabhängige Korrekturverfahren. Bei der Ermittlung rheologischer Daten wird i.d.R. die Bagley-Korrektur vorgenommen, um Einlaufdruckverluste beim Einstromen in die Kapillare zu berücksichtigen. Ohne die Korrektur wird die Kraft, die zur elastischen Verformung der Schmelze erforderlich ist gemessen und es werden als Folge zu hohe Werte für die Viskosität angenommen.

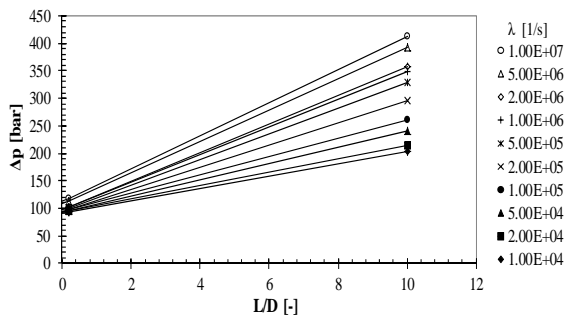


**Abbildung 1.** Am HKR ermittelte, scheinbare Viskositätsdaten des Compounds

Ohne Bagley-Korrektur ist demnach ein prozessabhängiger Widerstand begriffen, der sich aufgrund

geometrischer Bedingungen einstellt. Nach der Anwendung der Bagley-Korrektur beziehen sich die berechneten Daten rein auf das Materialverhalten und berücksichtigen keine Prozessabhängigkeiten. Da Spritzgießprozesse hochdynamisch sind und auch mehrere Querschnittsveränderungen zwischen dem Flansch der Plastifiziereinheit und der Kavität auftreten, sind Daten ohne Verknüpfung mit dem Verarbeitungsprozess nicht zielführend (Vergleich Einlauf in eine Kapillare). Das eingesetzte hochgefüllte Compound verstärkt diese Abhängigkeit zusätzlich. Die scheinbaren Viskositätsdaten liegen auf einem höheren Niveau als die nach Bagley- und Weissenberg-Rabinowitsch korrigierten Werte.

Bei zusätzlicher Betrachtung der ebenfalls bei 260 °C ermittelten Bagley-Geraden in **Abbildung 2**, wird der große Unterschied zwischen der wahren Viskosität und der prozessabhängigen Viskosität, hier HKR-Messung, deutlich. Zwei verschiedene Kapillaren, die jeweils für zehn verschiedene Schergeschwindigkeiten eingesetzt wurden, verdeutlichen den Druckverlust aufgrund der Geometrie bzw. des Prozesses. Abhängig von der eingestellten Schergeschwindigkeit liegen Druckdifferenzen zwischen 95 bar und 120 bar bei Verwendung der „Nulldüse“ vor.



**Abbildung 2.** Bagley-Geraden zur Korrektur der Einlaufdruckverluste

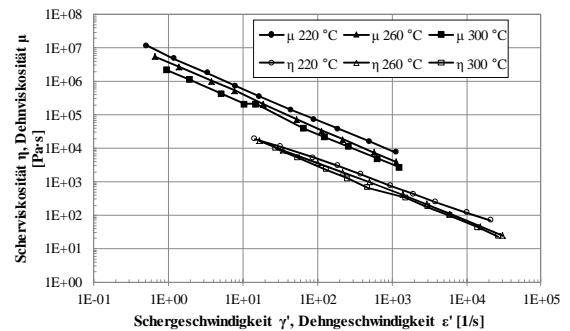
Eine weitere Prozessabhängigkeit wird im Cross-Arrhenius-Modell nicht berücksichtigt. Die angesprochenen Veränderungen im Strömungsquerschnitt sowie das Angussverteilersystem der Rechteckplatte verursachen nicht nur Scherung sondern auch Dehnung. Das Viskositätsmodell ermöglicht nur die Eingabe gemessener Scherviskositätsdaten und nicht die der Dehnviskosität. Neben dem grundsätzlich höheren Niveau der Dehnviskosität gegenüber der Scherviskosität kann je nach Polymer eine zusätzliche Dehnverfestigung bei höheren Schergeschwindigkeiten auftreten [4], [5], [6]. Die beiden beschriebenen Diskrepanzen zwischen der Abbildungsmöglichkeit

(Querschnittveränderungen und Dehnviskosität) des Cross-Arrhenius-Modells und den tatsächlich auftretenden, prozessabhängigen Strömungsverhältnisse führen dazu, dass höhere Viskositätsdaten als gemessen eingesetzt werden müssen. Das Viskositätsmodell, das zu einer bestmöglichen Abbildungsgenauigkeit des Werkzeuginnendruckverlaufes führt, wird nachfolgend als *effektive Viskosität* bezeichnet. Die effektive Viskosität ist kein Stoffwert, sondern ein Modellparametersatz, der die genannten Diskrepanzen bestmöglich berücksichtigt.

Zu höheren Werten verschobene Viskositätsdaten werden zusätzlich bei der Betrachtung der berechneten Cogswell-Dehnviskosität bekräftigt. **Abbildung 3** stellt die korrigierte Scherviskosität und die berechnete Dehnviskosität nach *Cogswell* des gleichen Materials im Vergleich dar [7], [8]. Der Zusammenhang zwischen den Scherviskositätsdaten und der berechneten Dehnviskosität nach *Cogswell* wird in den Gleichungen (3) und (4) wiedergegeben. Dabei wird deutlich, dass sich die Kurven parallel verschieben und im betroffenen Schergeschwindigkeitsbereich die Dehnviskosität im Vergleich eine Zehnerpotenz höher liegt als die Scherviskosität.

$$\mu = \frac{9 \cdot (n+1)^2 \cdot \Delta p_{ent}^2}{32 \cdot \eta_{ap} \cdot \dot{\gamma}_{ap}^2} \quad (3)$$

$$\dot{\epsilon} = \frac{4 \cdot \eta_{ap} \cdot \dot{\gamma}_{ap}^2}{3 \cdot (n+1) \cdot \Delta p_{ent}} \quad (4)$$



**Abbildung 3.** Wahre Scherviskosität und Dehnviskosität nach Cogswell im Vergleich

Im Folgenden wird die Auswirkung des Faktors  $T_b$  grafisch verdeutlicht. Für eine bessere Übersicht sind in **Abbildung 4** für die gemessene Scherviskosität und die berechnete Dehnviskosität jeweils nur für eine Temperatur aufgetragen. Beide auf Messdaten basierende Datenreihen sind zusätzlich mit einem Curve-Fitting anhand des Cross-Arrhenius-Modells versehen. Die Parameter  $n$ ,  $\tau^*$  sowie  $B$  wurden kon-

stant gehalten. Nur der Parameter  $T_b$  wurde von 8800 K auf 18000 K erhöht und führt zu einer guten Abbildung der Dehnaviskosität.

Diese Veränderung soll verdeutlichen, dass die Variation dieses Faktors ausreichend ist, um eine plausible Veränderung im Viskositätsmodell zu erzielen. Die leichte Steigungsabweichung kann anhand des Faktors  $n$  korrigiert werden, die Korrektur wird aufgrund der undurchsichtigeren Vergleichbarkeit jedoch nicht vorgenommen. Zuvor wurde beschrieben, dass beide Verformungsarten (Scherung und Dehnung) aufgrund des Prozesses und der Geometrie vorliegen, und beide Arten anteilig an den resultierenden Eigenschaften beteiligt sind. Dies bedeutet, dass die effektive Viskosität zwischen den beiden Kurven erwartet wird.

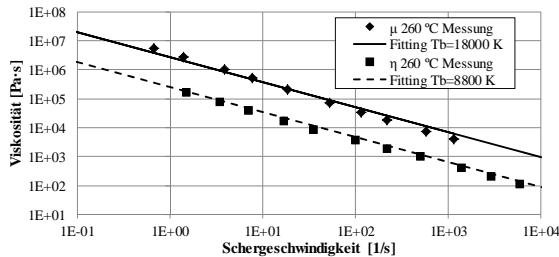


Abbildung 4. Auswirkung des Cross-Arrhenius-Parameters  $T_b$

### 3.2 Wärmeübergangskoeffizient

Gängige Simulationssoftware rechnet meist mit festen Werten zwischen ca. 5.000 W/(m<sup>2</sup>·K) für die Füllphase und ca. 25.000 W/(m<sup>2</sup>·K) für die Nachdruckphase. In der Realität handelt es sich jedoch um dynamische und lokal unterschiedliche Werte, die von komplexen Zusammenhängen zwischen Einflussgrößen wie Einspritzgeschwindigkeit, Druck, Temperatur, Schwindungsverhalten, Viskosität und Oberflächenbeschaffenheit des Werkzeuges abhängig sind [3].

Bisher ist es nicht möglich, die Größe des Wärmeübergangskoeffizienten quantitativ zu berechnen. Qualitativ lässt sich jedoch festhalten, dass die Größe des Wärmeübergangskoeffizienten maßgeblich von der lokalen Kontaktsituation zwischen Schmelze und Kavitätsoberfläche abhängig ist. Bei idealem Kontakt zwischen Schmelze und der Kavitätsoberfläche entspräche die wahre Kontaktfläche der theoretisch maximalen Kontaktfläche, sprich der gesamten Oberfläche der Kavität. In der Realität entspricht die wahre Kontaktfläche jedoch nicht der theoretischen Kontaktfläche. Daher stellen sich entsprechend der Prozess- und Materialparameter in der Regel geringere

Wärmeübergangskoeffizienten als 25.000 W/(m<sup>2</sup>·K) ein. Darüber hinaus ist eine veränderte Situation zwischen Füllphase und Nachdruckphase zu betrachten. In der Füllphase wird die tatsächliche Wärme übertragende Fläche nicht nur durch ein Rauigkeitsprofil der Kavitätsoberflächen beeinträchtigt, sondern auch von dem zu jedem Zeitpunkt bereits eingespritzten Volumen und der Strömungsgeschwindigkeit. Eine feinere zeitliche Diskretisierung als die Unterscheidung zwischen Füll- und Nachdruckphase ist seitens der Software nicht realisierbar.

Die Oberflächenrauheit der Kavität und die hohe Viskosität der Schmelze führen zu einem nicht idealen Kontakt zwischen Schmelze und Werkzeug. Je größer die Rauheit und die Viskosität, desto geringer ist die wahre Kontaktfläche. Zur Überwindung dieser Rauigkeitstäler ist vorrangig der lokal vorherrschende Druck bestimmend. Je höher der Druck ist, desto größer ist auch die Kontaktfläche [9],[10]. Es sind aber auch weitere Größen zu beachten, welche sich indirekt auf diesen Sachverhalt auswirken. Mit steigender Temperatur, sowohl der Schmelze, als auch des Werkzeuges, verringert sich die Viskosität der Schmelze in der Grenzschicht wodurch die Kontaktfläche größer wird. Auch die Einspritzgeschwindigkeit hat einen entscheidenden Einfluss. Es ist also davon auszugehen, dass eine höhere Geschwindigkeit des schmelzeartigen Kunststoffes wiederum zu einem schlechteren Kontakt zum Werkzeug führt [9].

Während des gesamten Abkühlprozesses der Schmelze ist zudem das Schwindungsverhalten des Compounds zu beachten. Je größer die Schwindung ausfällt, desto schlechter gestaltet sich die Kontaktsituation im Zuge der Abkühlung, da das Material von der Werkzeugwand abhebt [9]. Aufgrund des hohen Füllstoffgehaltes im Compound ist nur eine sehr geringe Druckabhängigkeit bzgl. des spezifischen Volumens vorhanden. Anhand des beschriebenen pvT-Verhaltens und des pvT-Diagramms ist das Schwindungsmaß vernachlässigbar klein. Gleichung (5) verdeutlicht den Zusammenhang zwischen dem Wärmeübergangskoeffizienten und der zu übertragenden Wärmemenge. Anhand dieser Gleichung wird die Wirkungsweise veranschaulicht, wie sich Unterschiede zwischen wahren und theoretischen Kontaktflächen auswirken.

$$Q = \alpha \cdot A \cdot (T_2 - T_1) \cdot \Delta t \quad (5)$$

Anhand der Biot-Zahl, welche in Gleichung (6) definiert ist, kann abgeschätzt werden, ob der Wärme-

übergangskoeffizient vernachlässigt werden kann. Dies ist der Fall wenn die Biot-Zahl einen Wert größer 100 annimmt [10], [10]. Neben dem vorliegenden Wärmeübergangskoeffizient und der Wärmeleitfähigkeit des betrachteten Materials wird die charakteristische Länge ( $D/2$ ) des betrachteten festen Körpers eingesetzt.

$$Bi = \frac{\alpha \cdot \frac{D}{2}}{\lambda} \quad (6)$$

Bei Betrachtung eines ungefüllten PP mit einer angenommenen Wärmeleitfähigkeit von 0,3 W/(m·K) und der 4 mm dicken Rechteckplatte, ergeben sich Biot-Zahlen zwischen 30 und 150 bei Variation des Wärmeübergangskoeffizienten zwischen 5.000 W/(m²·K) und 25.000 W/(m²·K). Die Biot-Zahlen überlappen somit den kritischen Bereich. Eine Berücksichtigung des Wärmeübergangskoeffizienten ist anhand dieser Betrachtung in vielen Fällen nicht notwendig.

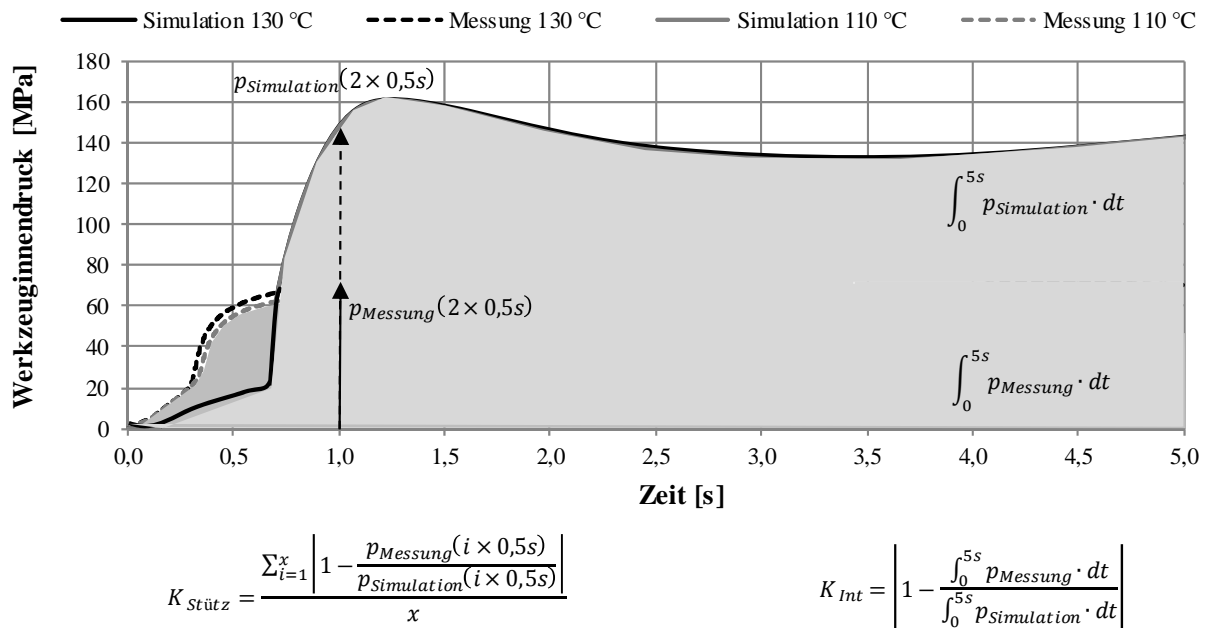
Das eingesetzte leitfähige Compound jedoch hat eine deutlich höhere Wärmeleitfähigkeit und es treten geringere Wärmeübergangskoeffizienten auf. Bei einer Wärmeleitfähigkeit von 17 W/(m·K) und bei Anwendung des gleichen Formteils ergeben sich Biot-Zahlen zwischen 0,58 und 2,94 bei Variation des Wärmeübergangskoeffizienten zwischen 5.000 W/(m²·K) und 25.000 W/(m²·K). Eine Berücksichtigung muss daher gewährleistet werden.

Aufgrund der beschriebenen Unsicherheiten hinsichtlich der Wärmeübergangskoeffizienten für die Füll- und Nachdruckphase und den zusammengetragenen, publizierten Erfahrungswerten, werden beide Koeffizienten als Simulationsvariable eingeführt, die eine zusätzliche Feinabstimmung hinsichtlich des simulierten Werkzeuginnendruckverlaufs ermöglichen. Der Wertebereich der Koeffizienten wird in **Tabelle 1** dargestellt.

Parameter	Bedeutung	Wertebereich
$\alpha_F$	Wärmeübergangskoeffizient Füllphase	5.000-10.000
$\alpha_P$	Wärmeübergangskoeffizient Nachdruckphase	7.000-12.000

**Tabelle 1.** Verwendete Wertebereiche der Wärmeübergangskoeffizienten

Die Koeffizienten, die in Kombination mit dem Simulationsparameter Viskosität die Zielfunktion bestmöglich beschreiben, werden als *effektive Wärmeübergangskoeffizienten* bezeichnet. Sie sind ein Ergebnis der iterativen Spritzgießsimulation.



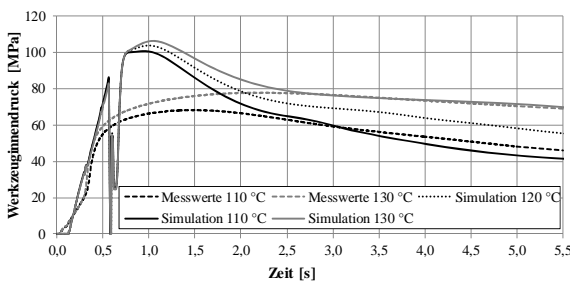
**Abbildung 5.** Simulationsergebnis ohne Kalibrierungsfähigkeit und Visualisierung der Bewertungskriterien

## 5 Ergebnisse

Einen Einstieg in die Darstellung der Ergebnisse soll **Abbildung 5** liefern. Sie zeigt ein Berechnungsergebnis, das ohne Kalibrierung erzeugt wurde. Alle beschriebenen Stoffdaten wurden unverändert der Simulations-Software bereitgestellt. Es wird deutlich, dass der gemessene Werkzeuginnendruckverlauf weder qualitativ noch quantitativ wiedergegeben werden kann. Leichte Korrekturmaßnahmen, die auf mögliche Diskretisierungsfehler eingehen können, sind bei der großen Diskrepanz der Ergebnisse nicht zielführend. Daher wurde der Ansatz eines kalibrierfähigen Material- und Prozessmodells verfolgt.

Nachfolgend sollen die Ergebnisse dargestellt werden, die anhand der vorgestellten Effektivwerte für die Kristallisationstemperatur, die Viskosität und die Wärmeübergangskoeffizienten berechnet worden sind. Im ersten Schritt wird ein Vergleich zwischen gemessenen und berechneten Werkzeuginnendruckverläufen bei 40 mm/s Einspritzgeschwindigkeit dargestellt (siehe **Abbildung 6**).

Um gleichzeitig den Gültigkeitsbereich des Simulationsmodells zu verdeutlichen, sind zwei verschiedene Werkzeugtemperaturen als Prozessvariable im Diagramm eingetragen. Beim Vergleich der jeweils zueinander gehörenden Kurven (berechnet und gemessen) kann eine sehr gute qualitative Übereinstimmung während des Großteils der Nachdruckphase festgestellt werden. Die Steigung während der Einspritz- und der Nachdruckphase ist ein weiteres Merkmal der Übereinstimmung. Anhand der berechneten Kurvenverläufe ist bei der Variation der Werkzeugtemperatur weiterhin eine übereinstimmende Temperaturabhängigkeit gegenüber den gemessenen Kurvenverläufen zu beobachten. Der Umschaltzeitpunkt wird nicht zufriedenstellend abgebildet. Auf die Bewertungskriterien wird im Folgenden noch näher eingegangen.



**Abbildung 6.** Werkzeuginnendruck für zwei verschiedene Werkzeugtemperaturen (40 mm/s)

Effektive Viskosität		Effektiver Wärmeübergangskoeffizient	
$B [Pa \cdot s]$	650	$\alpha_F [W/m^2 \cdot K]$	6.000
$\tau^* [Pa]$	48.000	$\alpha_P [W/m^2 \cdot K]$	8.000
$n [-]$	0,137		
$T_b [K]$	15.000		
$d [\frac{1}{Pa}]$	0		

**Tabelle 2.** Effektivwerte für die Simulation bei 40 mm/s Einspritzgeschwindigkeit

**Tabelle 2** zeigt die errechneten Parameter, die zur besten Übereinstimmung zwischen Simulation und Messung geführt haben. Die Viskosität wurde, wie in den Grundlagen aufgezeigt, zu höheren Werten durch eine Steigerung des Parameters  $T_b$  verschoben. Alle weiteren Parameter zur Definition der Viskosität blieben unverändert. In Kombination mit den effektiven Wärmeübergangskoeffizienten kann gezeigt werden, dass das aufgebaute Simulationsmodell den Anwender in die Lage versetzt, stark veränderte Materialeigenschaften von gefüllten Compounds sowie prozessabhängige Einflüsse richtig zu berücksichtigen. Zur objektiven Bewertung der Qualität der Simulationsergebnisse werden zwei Bewertungskriterien herangezogen, die kombiniert und zu jeweils 50 % gewichtet einen Kennwert über die Qualität der Simulation liefern. Die Bewertungskriterien sind das integrale Vergleichskriterium und das Stützstellenkriterium, welche in **Abbildung 5** visualisiert werden.

Einstellung	$K_{Int}$	$K_{Stütz}$	$K_{Gesamt}$
40 mm / 110 °C	0,174	0,120	0,147
40 mm / 130 °C	0,197	0,083	0,140

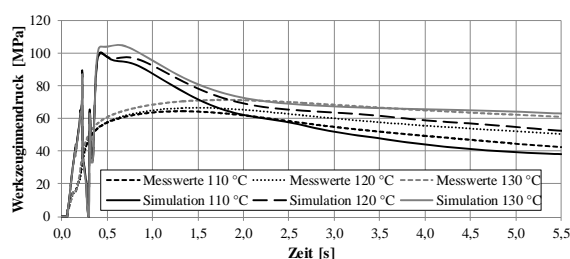
**Tabelle 3.** Auswertung der Simulationsergebnisse bei 40 mm/s Einspritzgeschwindigkeit

Die ermittelten Kennzahlen aus **Tabelle 3** können als Abweichungen in Prozent ausgedrückt werden. Die Übereinstimmung der Simulationsergebnisse wird demnach bei 40 mm/s Einspritzgeschwindigkeit bei niedriger Werkzeugtemperatur mit 14,7 % beziffert. Die Abweichung für die hohe Werkzeugtemperatur liegt darunter und beträgt 14,0 %.

Im nächsten Schritt soll die weitere Übertragbarkeit des Simulationsmodells veranschaulicht werden. Dazu werden in **Abbildung 7** Messdaten sowie berechnete Werkzeuginnendruckverläufe für eine höhe-

re Einspritzgeschwindigkeit (80 mm/s) dargestellt. Auch in diesem Fall besteht eine gute Übereinstimmung bzgl. der Parallelität der Vergleichskurven. Die Nachdruckphase kann sehr genau abgebildet werden.

Anhand der beiden Werkzeugtemperaturen 110 °C und 130 °C ist zu erkennen, dass wiederum eine übereinstimmende Temperaturabhängigkeit gegenüber den Messdaten besteht. Interessant ist allerdings die Überprüfung der Prognosefähigkeit des Modells. Nach der Kalibrierung der hohen und niedrigen Werkzeugtemperatur wurde mit dem gleichen Parametersatz der Werkzeuginnendruckverlauf für 120 °C Werkzeugtemperatur berechnet. Wie in der Modellvorstellung beschrieben, kann mit diesem Resultat bestätigt werden, dass Prognoseberechnungen innerhalb des Kalibrierungsbereiches möglich sind.



**Abbildung 7.** Werkzeuginnendruck für zwei verschiedene Werkzeugtemperaturen (80 mm/s)

Aus **Tabelle 4** können die gewählten Parameter als Grundlage der dargestellten Simulationsergebnisse abgelesen werden. Die Parameter für das Viskositätsmodell blieben dafür unverändert. Da der Prozess bei Steigerung der Einspritzgeschwindigkeiten einige Änderungen aufweist, die sich auf das Abkühlverhalten der Schmelze und dem Kontakt zu den Kavitätsoberflächen beziehen, wurden Änderungen an den Wärmeübergangskoeffizienten vorgenommen. Wie bereits dargestellt, nimmt der Wärmeübergangskoeffizient der Füllphase nicht nur Einfluss auf die Füllphase, sondern auch auf das Niveau des weiteren Verlaufes während der Nachdruckphase. Das beste Resultat konnte daher erzielt werden, wenn für  $\alpha_F$  10.000 W/(m<sup>2</sup>·K) und für  $\alpha_P$  8.250 W/(m<sup>2</sup>·K) gewählt wird.

Effektive Viskosität		Effektiver Wärmeübergangskoeffizient	
$B [Pa \cdot s]$	650	$\alpha_F [W/m^2 \cdot K]$	10.000
$\tau^* [Pa]$	48.000	$\alpha_P [W/m^2 \cdot K]$	8.250
$n [-]$	0,137		
$T_b [K]$	15.000		
$d [\frac{1}{Pa}]$	0		

**Tabelle 4.** Effektivwerte für die Simulation bei 80 mm/s Einspritzgeschwindigkeit

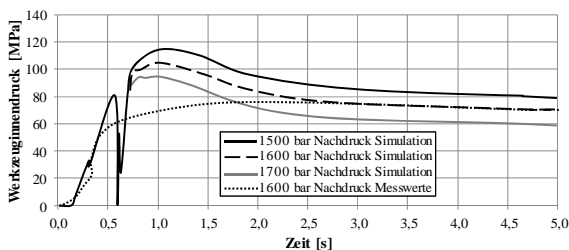
Der visuelle Eindruck aufgrund der Darstellung im Diagramm wird zusätzlich mit den bekannten Koeffizienten zur Bewertung der Qualität der Ergebnisse ergänzt. **Tabelle 5** beinhaltet für alle drei gemessenen und berechneten Werkzeuginnendruckverläufe die berechneten Koeffizienten. Die Abweichungen des Simulationsmodells nach Übertragung auf eine höhere Einspritzgeschwindigkeit liegen zwischen 11,5 % und 16,5 % für das integrale Kriterium und zwischen 9,7 % und 13,0 % für das Stützstellenkriterium. Die Ergebnisse sind ein zahlenmäßiger Nachweis, dass Erweiterungen des Simulationsmodells mit mehreren Prozessvariablen möglich sind, obwohl das Materialmodell unverändert bleibt. Die Abbildungsgenauigkeit erzielt eine ähnliche Genauigkeit, der ursprünglich kalibrierten Daten.

Einstellung	$K_{Int}$	$K_{Stütz}$	$K_{Gesamt}$
80 mm / 110 °C	0,115	0,130	0,122
80 mm / 120 °C	0,138	0,112	0,125
80 mm / 130 °C	0,165	0,097	0,131

**Tabelle 5.** Auswertung der Simulationsergebnisse bei 80 mm/s Einspritzgeschwindigkeit

Durch ergänzende Messungen und Simulationen können zukünftig noch weitere Informationen gewonnen werden, welche die Prozessdynamik aufgreifen. In den Grundlagen wurde erläutert, dass aufgrund der Dynamik des Spritzgießprozesses eine Erhöhung der Viskosität des Materials vorgenommen werden muss. Im vorgestellten Fall wurde die Dynamik in Form der Einspritzgeschwindigkeit beeinflusst ohne die Viskositätsdaten anzupassen. Die Einführung eines von der Einspritzgeschwindigkeit abhängigen Parameters  $T_b$  ist eine mögliche Erweiterung.

Abschließend wird ein Simulationsergebnis dargestellt, das auf die Veränderung eines weiteren Spritzgießparameters eingeht, ohne diesen kalibriert zu haben. Dieser Parameter ist der Nachdruck, der abhängig von den Prozessparametern Schmelze- und Werkzeugtemperatur in realen Prozessen verändert wird. **Abbildung 8** zeigt den Einfluss auf das Simulationsergebnis bei Erhöhung bzw. Reduzierung des Nachdruckes um 100 bar. Das Modell berechnet eine Änderung des Werkzeuginnendruckverlaufes während der Nachdruckphase von ca. 90 bar.



**Abbildung 8.** Änderung eines nicht kalibrierten Parameters (Nachdruck)

Um einen größeren Gültigkeits- und Funktionsbereich des Modells zu ermöglichen, müssten weitere Parameter wie der Nachdruck oder die Schmelzetemperatur anhand weiterer Messungen ergänzt werden. Da der Zeitaufwand deutlich steigen wird, ist die Einführung eines Automatismus nützlich. Ist die präzise Auskunft über das Formfüllverhalten für einen Prozess wichtig, ist die vorgestellte Methodik korrekt. Wenn ein breiter aufgestelltes Materialmodell angestrebt wird, können weitere Parameter wie beschrieben ergänzt werden; die Genauigkeit wird durch die Ergänzung für jeden einzelnen Betriebspunkt jedoch abnehmen.

## 6 Zusammenfassung und Ausblick

Anhand einiger Beispiele und der Variation mehrerer Spritzgießparameter konnte die Kalibrierfähigkeit sowie die Verbesserung der Abbildungsgenauigkeit der Spritzgießsimulation nachgewiesen werden. Hohe Füllgrade, stark veränderte thermodynamische Eigenschaften des Compounds sowie starke Unterschiede zwischen Charakterisierungsprozessen und Produktionsprozessen erforderten eine neue Vorgehensweise, die auf die genannten Unterschiede Rücksicht nehmen kann.

Der gängige Umgang mit Ergebnissen aus der Spritzgießsimulation besteht in der Auswertung des Formfüllverhaltens, der Ermittlung des erforderlichen

Einspritzdruckes sowie die lokale Untersuchung des Formteils hinsichtlich der Bindenähte und Entlüftungskanäle. Eine grundlegende Voraussetzung zur Analyse einer der genannten oder anderen Eigenschaften ist die zeitlich richtige Berechnung des Formfüllverhaltens. Diese wurde in dem aufgebauten Prozess- und Materialmodell anhand der Kalibrierung des Werkzeuginnendruckverlaufes ermöglicht. Die Kalibrierung selbst erfordert ein Mindestmaß an Variabilität im Prozess- sowie im Materialmodell. Mit den ausgewählten Variablen Viskosität (Variation  $T_b$ ) sowie den Wärmeübergangskoeffizienten für die Füll- und Nachdruckphase konnten einige prozessbedingte Einflüsse berücksichtigt werden, welche bei der Ermittlung der rheologischen Stoffdaten nicht erfasst werden konnten. Ein Fokus muss auf die Viskositätsdaten bei sehr kleinen Schergeschwindigkeiten gelegt werden. Die Ermittlung dieser Daten ist jedoch aufgrund von Blockströmung und des hohen Viskositätsniveaus schwierig.

Die Variationsparameter, die zu einer Lösung bzw. zur besten Lösung führten, wurden als effektive Viskosität und effektiver Wärmeübergangskoeffizient bezeichnet. Damit sind keine wahren Stoffwerte, sondern prozessabhängige Simulationsparameter gemeint. Mit dem Einsatz dieser Effektivwerte gelang die Kalibrierung des Werkzeuginnendruckverlaufes für verschiedene Prozessvariationen.

Anhand der Parameter Einspritzgeschwindigkeit und Werkzeugtemperatur wurde das Modell kalibriert. Gleichzeitig konnte dargestellt werden, dass nach erfolgreicher Kalibrierung auch Prognoseberechnungen für Einstellungen innerhalb der Kalibrierungsgrenzen durchführbar sind. Diese Erkenntnis ist besonders wichtig, da hiermit beim Aufbau anderer Kalibrierungsaufgaben oder bei der Erweiterung für andere Variablen der experimentelle Aufwand minimiert werden kann. Ein Abfahren der Grenzen zur Aufzeichnung der notwendigen Daten ist ausreichend.

Um die Erweiterbarkeit zu testen, wurde mit einem weiteren Spritzgießprozessparameter das Modell beansprucht. Die Steigerung des Nachdruckes führt zu einer plausiblen Reaktion des Modells. Eine Änderung der Geometrie (z. B. die Wanddicke) erfordert weitere Analysen und ist nicht mit dem gleichen Aufwand zu kalibrieren, wie veränderte Prozessparameter.

Für die Simulationssoftware wird der Aufbau eines automatischen Simulationsalgorithmus angestrebt, um Bedienungsfehler zu minimieren und Zeit einzusparen. Die Erweiterung auf mehr als zwei Faktoren zur Kalibrierung, um den Gültigkeitsbereich zu vergrößern, ist durch den Einsatz einer Automatisierung deutlich leichter zu erreichen

Das Modell bzw. das Modellierungsverfahren ist mit konkreten Beispielen vorgestellt worden. Für andere Spritzgießprozesse können vom Beispiel abweichende Prozessparameter in den Fokus gelangen, die ebenfalls anstelle der Parameter Werkzeugtemperatur oder Einspritzgeschwindigkeit eingesetzt werden können. Auch eine Erweiterung auf mehrere Dimensionen kann durch den gegebenen Ausblick optimistisch betrachtet werden.

Das erarbeitete Simulationsmodell leistet einen Beitrag zur Optimierung von PEM-BZ oder anderen Anwendungen, die hochgefüllte und elektrisch leitfähige Kunststoffe einsetzen.

## 7 Anmerkungen

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 03X0048C gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor. Die Autoren bedanken sich für die Unterstützung im Projekt CarboPlate bei der Bayer Technology Services GmbH, Clariant Masterbatches GmbH, Evonik Degussa GmbH und dem Zentrum für BrennstoffzellenTechnik GmbH.

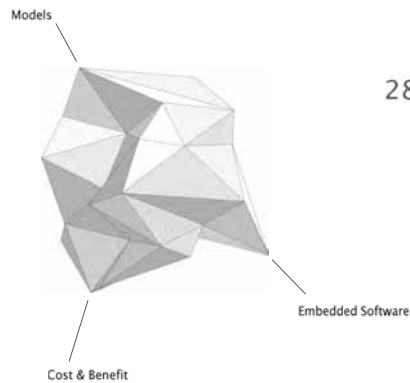


## 8 References

- [1] Chen, S. C.; Shih, M. Y.; Lee, P. H.: Effects of Molding Conditions on the Conductivity Properties of Injection Molded Bipolar Plate used for Fuel Cell. In: 64th Annual Technical Conference of the Society of Plastics Engineers (ANTEC 2006), S. 1168–1171.
- [2] Dörner, J.: Spritzgießen elektrisch leitfähiger Thermoplaste - Prozesstechnik und Modellbildung. Dissertation Universität Duisburg-Essen, 2012.
- [3] Amberg, J.; Brokamp, S.; Derieth, T.: Charakterisierung hochgefüllter Graphit-Compounds zur Auslegung und Optimierung des Spritzgießprozesses für Bipolarplatten in Brennstoffzellen-Anwendungen. AiF-Forschungsvorhaben (IGF-Vorhaben 304 ZN). 2012.
- [4] Münstedt, H.: New Universal Elongational Rheometer for Polymer Melts. In: Journal of Rheology, Jg. 23, S. 421–436, 1979.
- [5] Barnes, H. A.; Roberts, G. P.: A Simple Empirical Model Describing the Steady-State Shear and Extensional Viscosities of Polymer Melts. In: Journal of Non-Newtonian Fluid Mechanics, Jg. 44, S. 113–126, 1992.
- [6] Kurzbeck, S.: Dehnrheologische Eigenschaften von Polyolefinschmelzen und Korrelation mit ihrem Verarbeitungsverhalten beim Folienblasen und Thermoformen. Dissertation Universität Erlangen-Nürnberg, 1999.
- [7] Cogswell, F. N.: Converging Flow of Polymer Melts in Extrusion Dies. In: Polymer Engineering and Science, Jg. 12, H. 1, S. 64–73, 1972.
- [8] Kwag, C.; Vlachopoulos, J.: An Assessment of Cogswell's Method for Measurement of Extensional Viscosity. In: Polymer Engineering and Science, Jg. 31, H. 14, S. 1015–1021, 1991.
- [9] Nguyen-Chung, T.; Jüttner, G.; Löser, C.: Determination of the Heat Transfer Coefficient from Short-Shots Studies and Precise Simulation of Microinjection Molding. In: Polymer Engineering and Science, Jg. 50, H. 1, S. 165–173, 2010.
- [10] Brunotte, R.: Die thermodynamischen und verfahrenstechnischen Abläufe der in-situ-Oberflächenmodifizierung beim Spritzgießen. Dissertation Technische Universität Chemnitz, 2006.
- [11] Kalwa, M.: Anwendung der Finiten Elemente Methode zur Simulation von Wärmetransportvorgängen in der Kunststoffverarbeitung. Dissertation RWTH Aachen, 1990.



# Cost-Benefit Analysis of Model-Based Software Development



ASIM Workshop  
28.02.2013 – 01.03.2013 FH Düsseldorf

Dr. Jens Zimmermann, Altran

F 084 D R06 2012-07

altran

## Agenda

1. Models in Software Development
2. Established Model Use Cases
3. Model Use Cases with Potential
4. Cost-Benefit Analysis

F 084 D R06 2012-07

altran

# 1 // Models in Software Development

F 084 D R06 2012-07

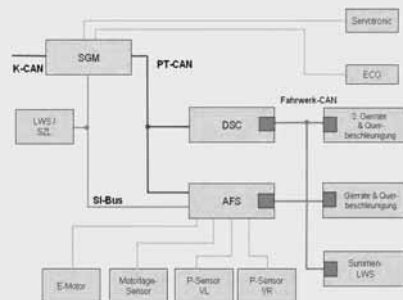
altran

## Architectural Model

### Reality



### Architectural Model



4

F 084 D R06 2012-07

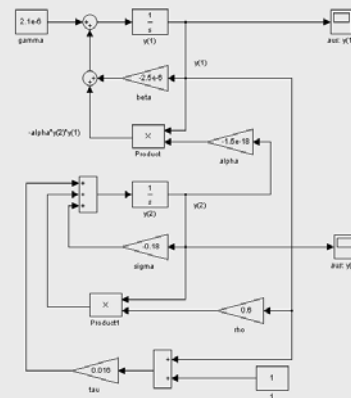
altran

## Function Model

Reality



Function Model



5

F 084 D R06 2012-07

ALTRAN

## Simulation Model

Reality



Simulation Model



6

F 084 D R06 2012-07

ALTRAN

## Models



7

F 084 D R06 2012-07

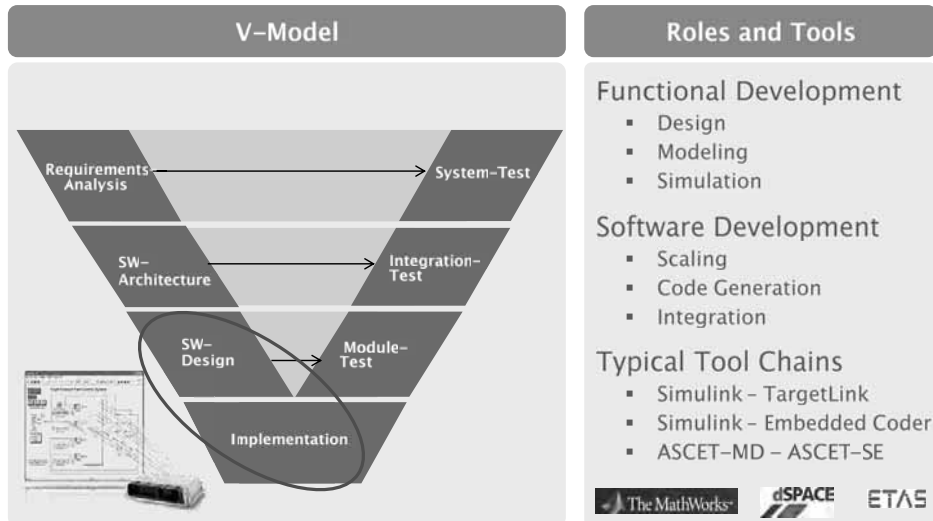
altran

## 2// Established Model Use Cases

F 084 D R06 2012-07

altran

## The best-explored Usage of Models: Design & Implementation

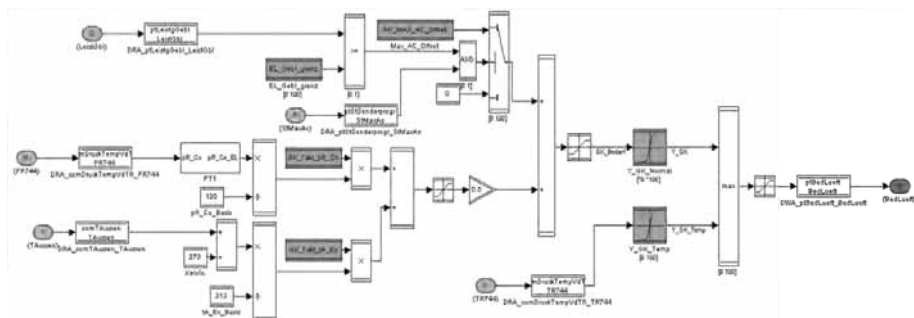


9

F 084 D R06 2012-07

altran

## Typical Models: Signal-Flow-Oriented Block-Diagrams



MATLAB  
SIMULINK

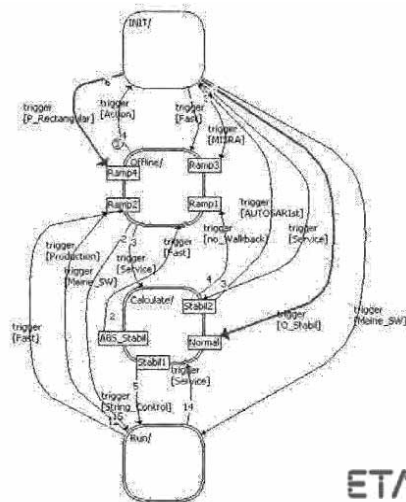
TargetLink  
3.0

10

F 084 D R06 2012-07

altran

## Typical Models: State Machines



ETAS

11

F 084 D R06 2012-07

altran

## 3 // Model Use Cases with Potential

F 084 D R06 2012-07

altran

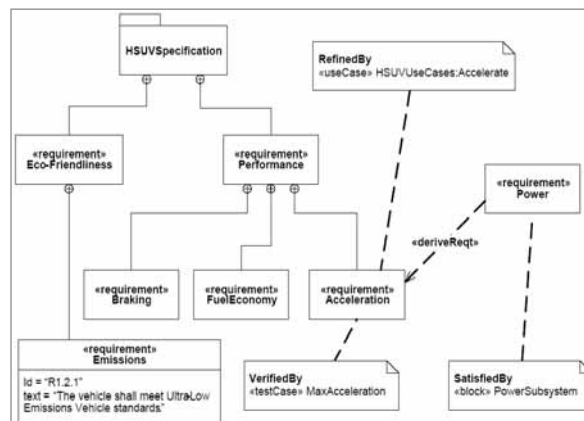
## Modeling Requirements with SysML

### Origin of SysML

Extensions of UML  
New Elements / Diagrams  
Focus on System Engineering

### Advantages

Improved Traceability  
Automated Consistency Checks  
Better Overview



13

F 084 D R06 2012-07

ALTRAN

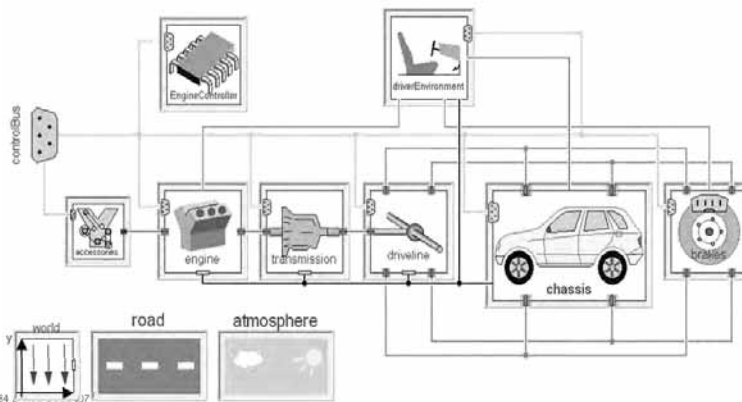
## Model Based Testing

### Early Validation by

Model in the Loop (MiL),  
Software in the Loop (SiL),  
Processor in the Loop (PiL)  
Models simulating the Environment

### Automated Generation of Test Cases

e.g. based on  
State Machines describing  
expected Functionality



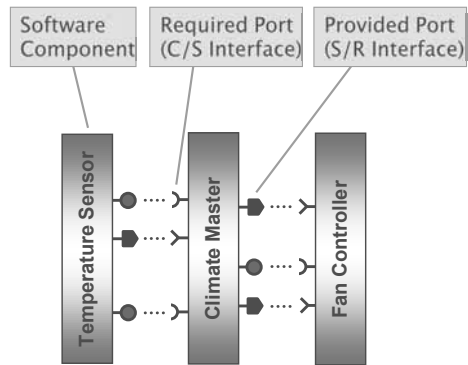
Modelica  
Multi-Engineering  
Modeling and Simulation  
MODELICA

14

F 084 D R06 2012-07

ALTRAN

## Meta Modelling



**M2: Meta-Model-Level**  
Language basis for functions:  
Software-Component-Template

**M1: Model-Level**  
Model of the climate functionality

**M0: Implementation-Level**  
Climate system running



15

F 084 D R06 2012-07

ALTRAN

## 4// Cost-Benefit Analysis

F 084 D R06 2012-07

ALTRAN



## Study about Model-Based Software Development

*What is the real quantitative potential of model based software development?*

**Study “Costs and benefits of model-based development of embedded software systems in the automotive domain”**

In cooperation with the Technical University of Munich

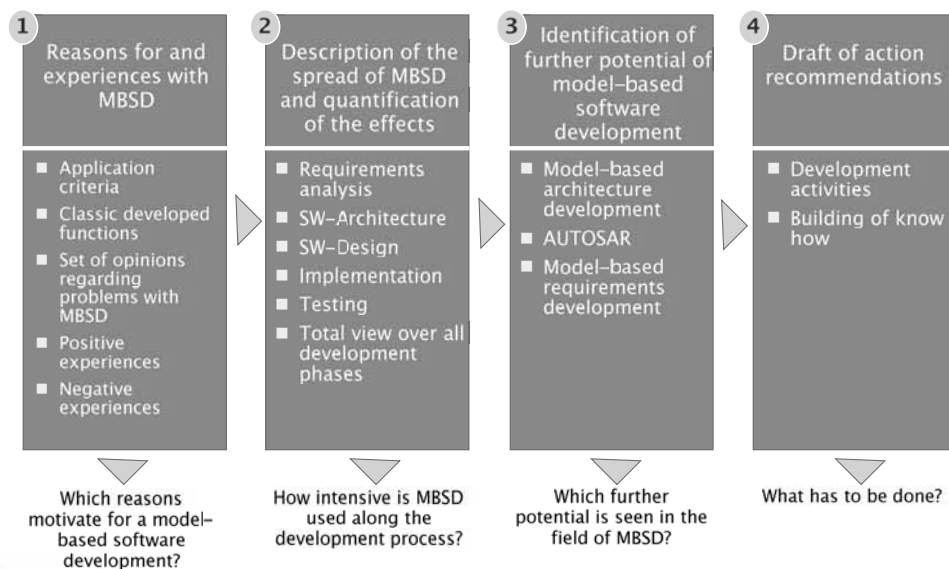
Lehrstuhl für  
Wirtschaftsinformatik

17

F 084 D R06 2012-07

ALTRAN

## Contribution of the study

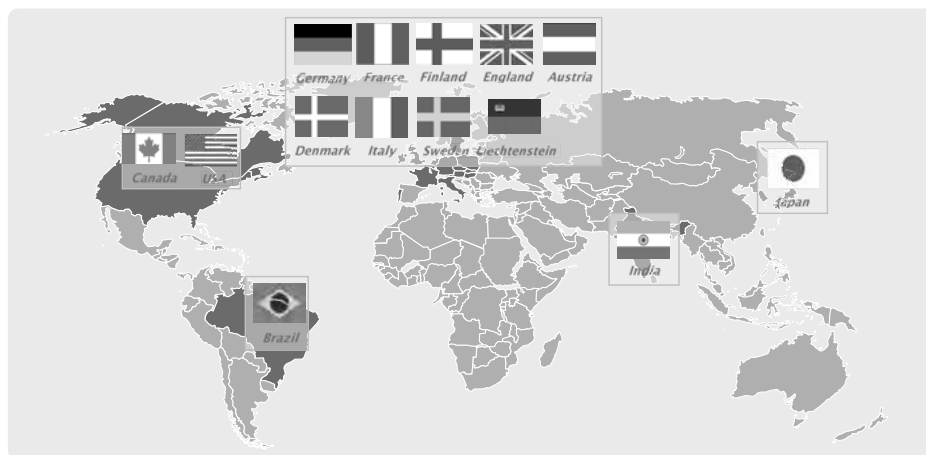


18

F 084 D R06 2012-07

ALTRAN

## Country of origin of the participants of the study

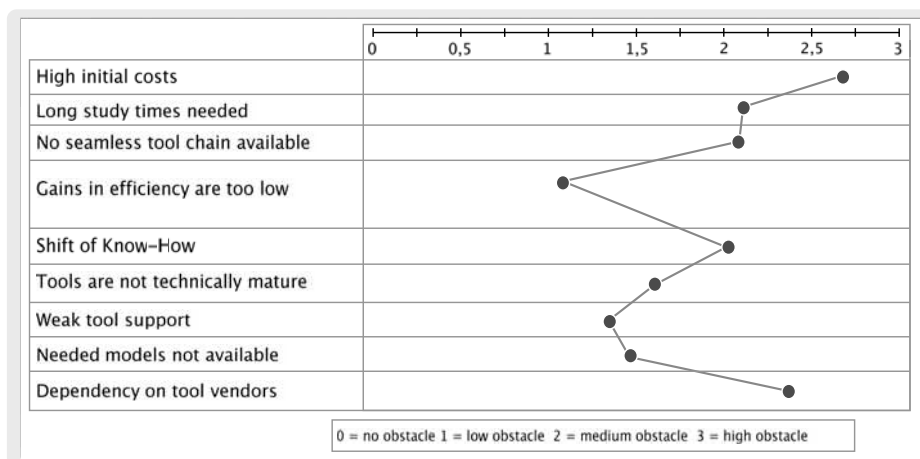


19

F 084 D R06 2012-07

altran

## Result of the Study: Overview about problems with MBSE



20

F 084 D R06 2012-07

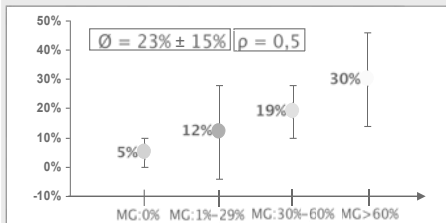
altran

## Result of the Study: Cost & Benefit of RapidControlPrototyping

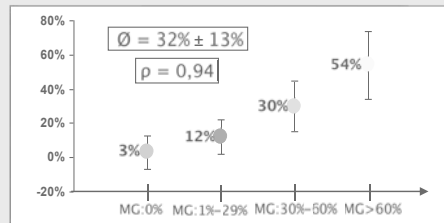
### RapidControlPrototyping as part of Requirements Analysis

The more intensive RapidControlPrototyping is used during requirements analysis, the higher are the costs but also the number of detected errors.

#### Changes in costs



#### Changes in number of detected errors



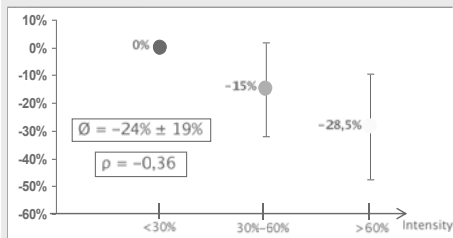
21

F 084 D R06 2012-07

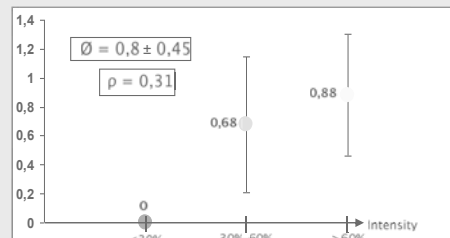
ALTRAN

## Result of the Study: Impact of modeling Degree and Simulation

### Modeling degree and test intensity vs. overall costs



### Modeling degree and test intensity vs. product quality



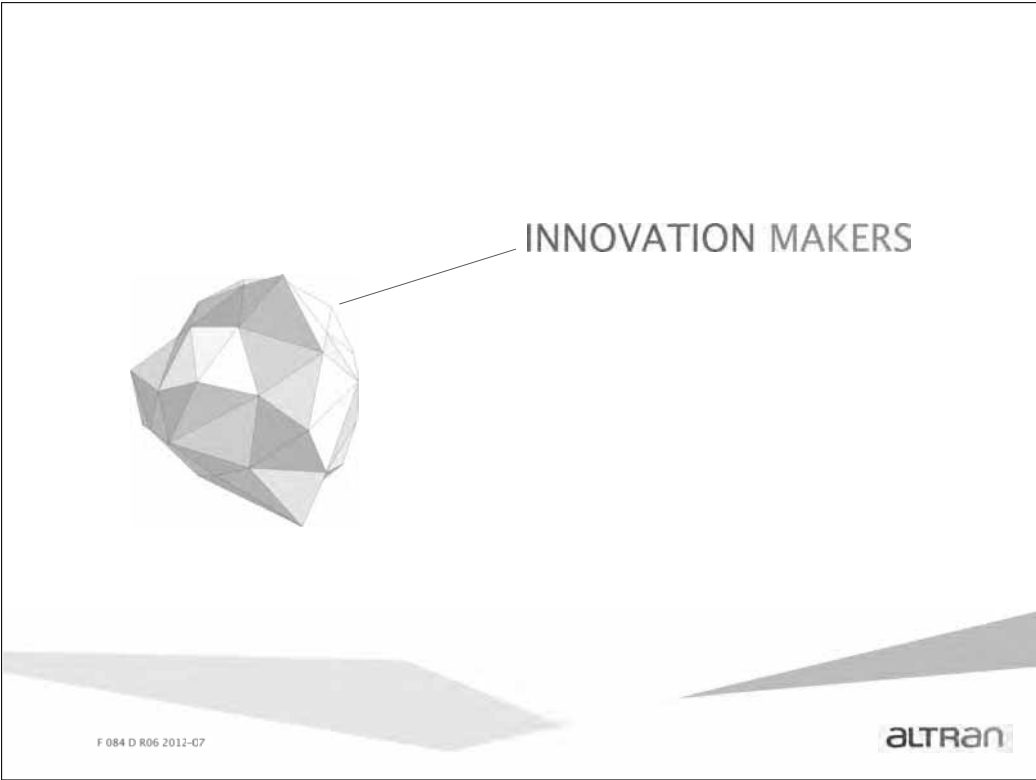
### Using Models as much as possible (Modeling and MiL-Simulation) pays off

- The higher the modeling degree and the test intensity on function models is the more the overall costs decrease.
- The higher the degree of modeling and the test intensity on function models is the bigger are the improvements in product quality.

22

F 084 D R06 2012-07

ALTRAN



# Multigrid-based Solution Methods for the Optimal Control of Nonstationary Flows

Köster, M.<sup>1</sup>; Hinze, M.<sup>2</sup>; Turek, S.<sup>1</sup>

<sup>1</sup>Institute of Applied Mathematics, TU Dortmund

<sup>2</sup>Department of Mathematics, University of Hamburg

*michael.koester@mathematik.tu-dortmund.de*

While simulation methods in the field of Computational Fluid Dynamics (CFD) are already quite advanced, efficient and well developed numerical approaches for optimisation and optimal control in CFD are still under development. Particular challenges in this context are optimisation problems with instationary flows, resulting in discrete nonlinear systems of equations with an extremely high number of unknowns. We give a short overview about current developments in the field of solvers for such situations. Our emphasis will be placed on special space-time multigrid based solution strategies which are able to solve the underlying nonlinear system of equations with linear complexity.

## 1 Introduction

The optimal control of incompressible, nonstationary flow problems belongs to today's most challenging problems in the field of optimisation. Modeling leads to minimisation problems whose necessary first order optimality conditions form the so-called Karush-Kuhn-Tucker (KKT-) systems. These is the starting point for numerical solution approaches. All variables in the KKT system are fully coupled, so that its discretisation leads to a very high-dimensional nonlinear system of equations.

There are different discretisation and solver approaches available to tackle KKT systems numerically, and a fairly good overview can be found in [1] and [2]. Newer works (cf. [3], [5]) start to use multigrid methods to prevent performance degeneration for fine discretisations. The trick is to apply multigrid methods simultaneously in space and time. We will sketch this approach for the optimal distributed control of the nonstationary Navier-Stokes equations.

## 2 Distributed flow control

A typical model problem for optimisation in CFD is optimal distributed control of the nonstationary Navier-Stokes equations, which is modelled as follows. The spatial domain is denoted by  $\Omega \subset \mathbb{R}^d$  ( $d=2,3$ ), with boundary  $\Gamma$ . For  $T>0$ ,  $Q:=(0,T) \times \Omega$  defines a space-time cylinder and  $\Sigma:=(0,T) \times \Gamma$  its space-time boundary. With  $\alpha>0$  a regularisation parameter,  $z:Q \rightarrow \mathbb{R}^d$  a given 'target' function and

$y^0: \mathbb{R}^d \rightarrow \mathbb{R}^d$  a given initial condition, we consider the minimisation problem

$$J(y, u) = \frac{1}{2} \|y - z\|_{L^2(Q)}^2 + \frac{\alpha}{2} \|u\|_{L^2(Q)}^2 \rightarrow \min \quad (1)$$

such that

$$y_t - \Delta y + y \nabla y + \nabla p = u \quad \text{in } Q, \quad (2a)$$

$$-\nabla \cdot y = 0 \quad \text{in } Q, \quad (2b)$$

$$y = 0 \quad \text{on } \Sigma, \quad (2c)$$

$$y(0) = y^0 \quad \text{in } \Omega, \quad (2d)$$

where  $\|\cdot\|_{L^2(Q)}$  denotes the  $L_2$ -norm over the space-time cylinder  $Q$ . The aim of this problem is to find an optimal control  $u$  such that the flow  $y$  is as close as possible to the given target flow  $z$  in the mean-square sense.

### 2.1 The KKT system

Using the Lagrange multiplier approach [3], the necessary optimality conditions of the above minimization can be formulated as a nonlinear system of equations:

- State, adjoint and control equation:

$$y_t - \Delta y + y \nabla y + \nabla p = u \quad \text{in } Q, \quad (3a)$$

$$-\nabla \cdot y = 0 \quad \text{in } Q, \quad (3b)$$

$$-\lambda_t - \Delta \lambda - y \nabla \lambda + (\nabla y)^T \lambda + \nabla \xi = (y - z) \quad \text{in } Q, \quad (4a)$$

$$-\nabla \cdot \lambda = 0 \quad \text{in } Q, \quad (4b)$$

$$\alpha u + \lambda = 0 \quad \text{in } Q \quad (5)$$

- Boundary and initial/end time conditions

$$\begin{aligned} y &= \lambda = 0 && \text{on } \Sigma, \\ y(0) &= y^0, \quad \lambda(T) = 0 && \text{in } \Omega, \end{aligned}$$

In this system,  $\lambda: Q \rightarrow \mathbb{R}^d$  denotes the so-called adjoint velocity and  $\xi: Q \rightarrow \mathbb{R}$  the adjoint pressure. Equation (3) proceeds forward in time, equation (4) backward in time and the control equation (5) couples the forward and backward equation. Therefore, the system represents a fully coupled boundary value problem in space and time.

## 2.2 Space-time discretisation and hierarchy

For the discretisation in space, we choose the finite element space  $Q_2$  of quadratic finite elements on quadrilaterals for the velocities and controls. The pressures are discretised using  $P_1^{disc}$ , the space of piecewise discontinuous linear finite elements. Time discretisation is carried out, e.g., with the implicit Euler scheme. Higher-order schemes like Crank-Nicolson are also possible, but lead to more complicated numerical algorithms (see [3]).

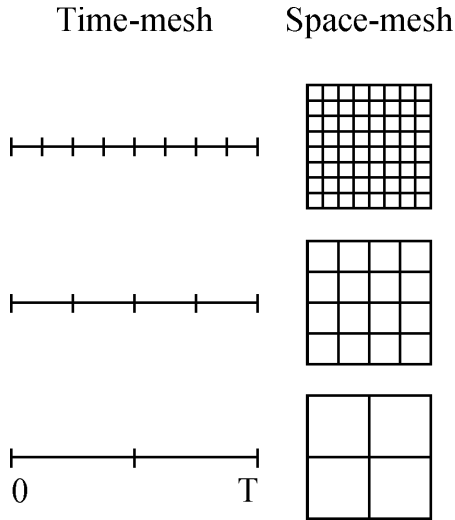


Figure 1: Hierarchy of three space-time meshes. Coarse mesh (level 1) on the bottom.

A space-time mesh hierarchy of  $L \in \mathbb{N}$  levels is obtained as follows:

1. Spatial discretisation: Let  $T_1, \dots, T_L$  define a hierarchy of meshes in space generated by regular refinement of a basic mesh. On each level, a finite element discretisation is carried out, resulting in a hierarchy of finite element spaces  $V_1, \dots, V_L$  for velocity/control and  $Z_1, \dots, Z_L$  for the pressure.

2. Time discretisation: The time interval  $[0, T]$  is subdivided into  $N$  time intervals of equal length  $k = T/N$  with gridpoints  $t_0 = 0 < t_1 < \dots < t_N = T$ . A time hierarchy is generated by bisection, so that time level  $l$  contains  $2^{l-1}N$  time intervals.

The space-time space  $W_l$ ,  $l=1, \dots, L$ , is defined by the combination of  $(V_l, Z_l)$  in space and  $2^{l-1}N$  intervals in time. Figure 1 sketches a hierarchy of three space-time levels. On the finest space  $W_L$ , a solution to the optimal control problem (1) shall be computed.

## 3 Solver design

Different solver approaches have been developed in recent years to tackle the above KKT system (3)-(5) numerically, see [1], [3], [4], [5]. In the following, we sketch two types of solvers which couple Newton techniques for the nonlinear parts with multigrid techniques for the linear subproblems.

### 3.1 The Newton approach

The Newton approach is based on equation (5) which we write in operator form as

$$F(u) := \alpha u + \lambda = 0 \quad \text{in } Q. \quad (6)$$

In this equation,  $\lambda = \lambda(y(u))$  is a solution of the adjoint equation (4) and  $y = y(u)$  a solution of the primal equation (3). Given an initial iterate  $u_0$ , the Newton iteration for finding the solution of (6) reads

$$u_{n+1} = u_n - F'(u_n)^{-1} F(u_n), \quad (7)$$

which consists of three steps:

1. Form the residual

$$d_n = -F(u_n). \quad (8a)$$

This necessitates three substeps:

- One forward iteration to solve (3) with given  $u_n$ ; this provides an intermediate primal velocity  $y_n$  and pressure  $p_n$ .
- One backward iteration to solve (4) with given  $y_n$ ; this results in an intermediate adjoint velocity  $\lambda_n$  and pressure  $\xi_n$ .
- Form

$$d_n = -(\alpha u_n + \lambda_n).$$

2. Calculate  $\bar{u}$  by solving the linear equation:

$$F'(u_n) \bar{u} = \alpha \bar{u} + \bar{\lambda} = d_n \quad (8b)$$

3. Update the solution:

$$u_{n+1} = u_n + \bar{u} \quad (8c)$$

In step (8b),  $\bar{\lambda} = \bar{\lambda}(\bar{y}(\bar{u}))$  is the solution of the adjoint equation

$$-\bar{\lambda}_t - \Delta \bar{\lambda} - y \nabla \bar{\lambda} + (\nabla y)^T \bar{\lambda} + \nabla \bar{\xi} = \bar{y} - (\nabla \bar{y})^T \lambda + \bar{y} \nabla \lambda \quad \text{in } Q, \quad (9a)$$

$$-\nabla \cdot \bar{\lambda} = 0 \quad \text{in } Q, \quad (9b)$$

$$\bar{\lambda} = 0 \quad \text{on } \Sigma, \quad (9c)$$

$$\bar{\lambda}(T) = 0 \quad \text{on } \Omega, \quad (9d)$$

with  $\bar{y} = \bar{y}(\bar{u})$  the solution of the linearised primal equation

$$\bar{y}_t - \Delta \bar{y} + y \nabla \bar{y} + \bar{y} \nabla y + \nabla \bar{p} = \bar{u} \quad \text{in } Q, \quad (10a)$$

$$-\nabla \cdot \bar{y} = 0 \quad \text{in } Q, \quad (10b)$$

$$\bar{y} = 0 \quad \text{on } \Sigma, \quad (10c)$$

$$\bar{y}(0) = 0 \quad \text{in } \Omega. \quad (10d)$$

Step (8b) is numerically expensive as a linear system in space and time has to be solved. This can be done iteratively by a multigrid approach.

### The linear solver in the Newton approach

For the linear equation (8b), an arbitrary iterative algorithm based on defect corrections can be used. For example, given a damping parameter  $\omega > 0$  and an initial guess  $\bar{u}_0$  on level L, a simple Richardson iteration reads

$$\bar{u}_{m+1} = \bar{u}_m + \omega(u_n - F'(u_n)\bar{u}_m). \quad (11)$$

Similarly, it is also possible to apply a BiCGStab or GMRES algorithm. All such iterative algorithms produce a sequence of iterates  $\bar{u}_m$  that converge to the solution  $\bar{u}$  of (8b), which is a function on the space-time cylinder. One has to note here that the creation of the defect is the expensive part: To calculate the matrix-vector product  $F(u_n)\bar{u}_m$ , the following three steps have to be done:

- Solve the primal equation (10); this is a simulation of a linear equation forward in time and computes an auxiliary  $\bar{y}_m$  and  $\bar{p}_m$ .
- Solve the adjoint equation (9); this is a simulation of a linear equation backward in time and computes an auxiliary  $\bar{\lambda}_m$  and  $\bar{\xi}_m$ .

- In each timestep, apply the linear combination

$$F(u_n)\bar{u}_m = \alpha \bar{u}_m + \bar{\lambda}_m. \quad (12)$$

### Multigrid in the Newton approach

Multigrid is a special type of iterative algorithm and can be embedded in the above iterative structure as well. It builds upon the hierarchy  $W_1, \dots, W_L$  of spaces in space and time. The basic idea reads as follows:

1. Take an initial guess  $\bar{u}_0$  as before.
2. **Smoothing:** On level L, carry out a fixed number of iterations with an iterative algorithm (CG, GMRES) as above, e.g., NSM=4 steps. This gives an intermediate solution  $\bar{u}_m^{NSM}$ .
3. **Coarse grid correction:** On level L-1, use  $\bar{u}_m^{NSM}$  to calculate a 'coarse grid solution'  $\tilde{u}$  to  $\bar{u}$ . Combine  $\tilde{u}$  and  $\bar{u}_m^{NSM}$  to get a new approximate  $\bar{u}_{m+1}$ . More precisely, one applies the defect correction formula

$$\bar{u}_{m+1} := \bar{u}_m^{NSM} + P(F'_{L-1}(u_n))^{-1} R(u_n - F'(u_n)\bar{u}_m^{NSM})$$

with R a suitable restriction operator from level L to L-1, P a prolongation operator from level L-1 to L and  $F'_{L-1}(u_n)$  an approximation to the operator  $F'(u_n)$  on level L-1.

This process can be carried out recursively: The approximate solution on level L-1 can be calculated by a couple of smoothing steps on level L-1 followed by a coarse grid correction from level L-2, etc. On level l=1, the iterative algorithm (CG, GMRES,...) has to be carried out until convergence to calculate a basic coarse grid solution. A good overview about multigrid can be found in [6] and [7].

### 3.2 The SQP approach

The above Newton approach can be interpreted as an elimination of the equations (3) and (4) from the KKT system. An iteration is carried out only in based on (5), using (3) and (4) as auxiliary problems. The SQP approach is an alternative approach which does the other way around, formulating an iteration in  $y$ ,  $p$ ,  $\lambda$ ,  $\xi$  and  $u$  simultaneously. It is usually more effective than the Newton approach. However, the approach is very intrusive and requires a certain amount of coding, as it cannot be realised with black-box solvers.

We consider a special SQP approach which eliminates the control. Elimination of equation (5) leads to the following set of equations:

- State equation:

$$y_t - \Delta y + y \nabla y + \nabla p = -\frac{1}{\alpha} \lambda \quad \text{in } Q, \quad (13a)$$

$$-\nabla \cdot y = 0 \quad \text{in } Q, \quad (13b)$$

- Adjoint equation:

$$-\lambda_t - \Delta \lambda - y \nabla \lambda + (\nabla y)^T \lambda + \nabla \xi = (y - z) \quad \text{in } Q, \quad (14a)$$

$$-\nabla \cdot \lambda = 0 \quad \text{in } Q, \quad (14b)$$

- Boundary conditions, initial/end time conditions (see above).

This equation can be expressed in operator form by

$$G(y, p, \lambda, \xi) = (0, 0, 0, 0) \quad (15)$$

with  $G=(G_1, G_2, G_3, G_4)$ ,  $G_i=G_i(y, p, \lambda, \xi)$  the defects of (13) and (14) given by

$$G_1 = y_t - \Delta y + y \nabla y + \nabla p + \frac{1}{\alpha} \lambda$$

$$G_2 = -\nabla \cdot y$$

$$G_3 = -\lambda_t - \Delta \lambda - y \nabla \lambda + (\nabla y)^T \lambda + \nabla \xi - y + z$$

$$G_4 = -\nabla \cdot \lambda$$

Given an initial iterate  $x_0:=(y_0, p_0, \lambda_0, \xi_0)$ , a Newton iteration has now to be applied to (15). With  $x_n:=(y_n, p_n, \lambda_n, \xi_n)$ , this reads

$$x_{n+1} = x_n - G'(x_n)G(x_n). \quad (16)$$

Again, this can be decomposed into three steps:

1. Create the residual

$$d_n = -G(x_n). \quad (17a)$$

2. Solve the linear equation

$$G'(x_n)\bar{x} = d_n. \quad (17b)$$

3. Update the solution by

$$x_{n+1} = x_n + \bar{x}. \quad (17c)$$

Here, (17b) means that  $\bar{x}=(\bar{y}, \bar{p}, \bar{\lambda}, \bar{\xi})$  is a solution of the equations (9) and (10), respectively, with  $y=y_n$ , and  $\lambda=\lambda_n$ .

### The defect

In contrast to the Newton approach, the creation of the residual as well as the solution of the linear equation is fundamentally different. The creation of the residual in (17a) is rather cheap. For a given iterate  $x_n$ , no system has to be solved. Instead, the operator has to be *applied* to  $x_n$ , which means a couple of matrix-vector products (for the Laplace/gradients/divergence operators) or an assembly (for the nonlinearities), respectively, in each timestep.

### The linear equation

The main difficulty in this approach is the linear system (17b) which has to be solved for all primal/adjoint variables simultaneously. This system has a very specific form. For example, if the implicit Euler scheme is used for the time discretisation, one can write the discrete solution vector containing all timesteps in the form

$$\bar{x}_n = (x_n^0, x_n^1, \dots, x_n^N)$$

with  $x_n^i = (y_n^i, p_n^i, \lambda_n^i, \xi_n^i)$  the solution at  $t_i$ ,  $i=0, \dots, N$ .

A matrix-vector product like in (17b) is then carried out with the discrete counterpart of  $G'(x_n)$ , given by a (block) tridiagonal matrix of the form

$$G'(\bar{x}_n) = \begin{pmatrix} A_0 & M_d & & & \\ M_p & A_1 & M_d & & \\ & M_p & A_1 & \ddots & \\ & & \ddots & \ddots & \\ & & & M_p & A_N \end{pmatrix}$$

The matrices  $M_p$  are weighted mass matrices, introduced by the implicit Euler scheme in the primal equation. Similarly, the matrices  $M_d$  are weighted mass matrices from the adjoint equation. The diagonal blocks  $A_i$  contain for each timestep a (linearised) Navier-Stokes operator that represents the coupled equations (13) and (14), as well as weighted mass matrix from the timestepping scheme.

### The linear solver in the SQP approach

As described above, applying the operator  $G'(x_n)$  corresponds in the discrete world to a matrix-vector product with a (block) tridiagonal matrix. Therefore, it is possible to use any iterative algorithm for linear systems (BiCGStab, GMRES) in order to solve (17b). For example, given an initial iterate



$\bar{x}_0 := (\bar{y}_0, \bar{p}_0, \bar{\lambda}_0, \bar{\xi}_0)$  and a damping parameter  $\omega > 0$ , a Richardson iteration reads

$$\bar{x}_{m+1} = \bar{x}_m + \omega(d_n - G'(x_n)\bar{x}_m). \quad (18)$$

Provided  $\omega$  is small enough, this should converge for  $m \rightarrow \infty$  to a solution  $\bar{x}$  of (17b). Using a BiCGStab or GMRES algorithm however is expected to converge much more rapidly.

### Preconditioning

A crucial point concerning the SQP approach is that one can easily derive efficient preconditioners. For example, the preconditioned version of the Richardson iteration above reads

$$\bar{x}_{m+1} = \bar{x}_m + \omega C^{-1}(d_n - G'(x_n)\bar{x}_m) \quad (19)$$

with  $C$  an approximation to  $G'(x_n)$ . Due to the fact that  $G'(x_n)$  can be represented as a tridiagonal matrix, straightforward and efficient preconditioners are the Block-Jacobi method, realised by

$$C_J = \begin{pmatrix} A_0 & & & & \\ & A_1 & & & \\ & & A_2 & & \\ & & & \ddots & \\ & & & & A_N \end{pmatrix},$$

the forward Block Gauß-Seidel preconditioner, realised by

$$C_{GS}^f = \begin{pmatrix} A_0 & & & & \\ M_p & A_1 & & & \\ & M_p & A_2 & & \\ & & \ddots & \ddots & \\ & & & M_p & A_N \end{pmatrix},$$

the backward Block Gauß-Seidel  $C_{GS}^b$  (realised by using  $M_d$  on the upper diagonal instead of  $M_p$  on the lower one) or combinations of these. Applying  $C^{-1}$  that way reduces to the application of the  $A_i^{-1}$  in all timesteps  $i=1, \dots, N$ . These problems, however, are problems in space, and efficient monolithic spatial multigrid solvers are available.

### Monolithic Multigrid in the SQP approach

For an efficient application of a monolithic multigrid strategy, one has to carry out the discretisation of the KKT system (i.e., equation (15)) on all levels  $1, \dots, L$ .

Multigrid is used to solve equation (17b) as a special type of an iterative solution algorithm. The basic idea reads as follows, similar to the Newton approach.

1. Take an initial guess  $\bar{x}_0$  on level  $L$  as before.
2. **Smoothing:** On level  $L$ , carry out a fixed number of iterations with an iterative algorithm (BiCGStab, GMRES) as above, e.g., NSM=4 steps. This gives an intermediate solution  $\bar{x}_m^{NSM}$ .
3. **Coarse grid correction:** On level  $L-1$ , use  $\bar{x}_m^{NSM}$  to calculate a 'coarse grid solution'  $\tilde{x}$  to  $\bar{x}$ . Combine  $\tilde{x}$  and  $\bar{x}_m^{NSM}$  to get a new approximate  $\bar{x}_{m+1}$ . More precisely, one applies the defect correction formula

$$\bar{x}_{m+1} := \bar{x}_m^{NSM} + P(G'_{L-1}(u_n))^{-1} R(d_n - G'(x_n)\bar{x}_m^{NSM})$$

with  $R$  a suitable restriction operator from level  $L$  to  $L-1$ ,  $P$  a prolongation operator from level  $L-1$  to  $L$  and  $G'_{L-1}(u_n)$  an approximate solution operator for  $G'(u_n)$  on level  $L-1$ .

Again, the computation of the coarse grid correction  $\tilde{x}$  can iteratively be carried out using a smoothing/coarse grid correction strategy that involves level  $L-2$ ,  $L-3$ , etc. On level  $l=1$ , a coarse grid solver has to be used, i.e., the iterative algorithm has to iterate until convergence.

## 4 Numerical examples

In this section, we present some numerical examples to demonstrate the applicability of the proposed solver approaches. We consider the following test example, based on the Flow-Around-Cylinder benchmark problem from [8], see also [9].

The spatial domain is described by a rectangle without an inner cylinder,

$$\Omega := (0, 2.2) \times (0, 0.41) \setminus B_r(0.2, 0.2), \quad r := 0.05$$

The boundary of this domain is decomposed into five parts:

$$\begin{aligned} \Gamma_1 &:= \{0\} \times [0, 0.41], & \Gamma_2 &:= (0, 2.2) \times \{0\}, \\ \Gamma_3 &:= \{2.2\} \times (0, 0.41), & \Gamma_4 &:= \{0, 2.2\} \times \{0.41\}, \\ \Gamma_5 &= \partial B_r(0.2, 0.2). \end{aligned}$$

Boundary conditions are defined as  $y(x, t) := (0, 0)$  for  $x \in \Gamma_2 \cup \Gamma_4 \cup \Gamma_5$ , do-nothing boundary conditions on

$\Gamma_3$  and a parabolic profile with maximum inflow velocity  $Y_{\max}=1.5$  at  $\Gamma_1$ . Using  $v=1/1000$ , this results in a  $Re=100$  optimisation. The time interval for this test case is  $[0, T]$  with  $T=0.35$  which roughly corresponds to one oscillation in the uncontrolled flow.

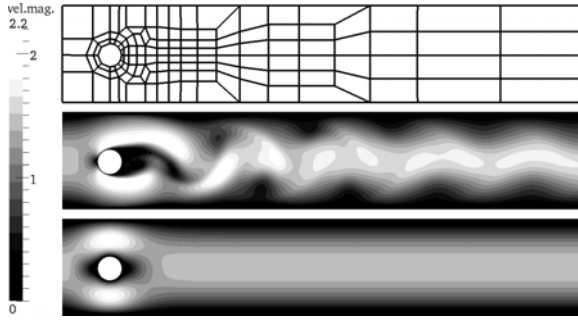


Figure 2: Mesh (top),  $y^0$  (centre) and  $z$  (bottom).

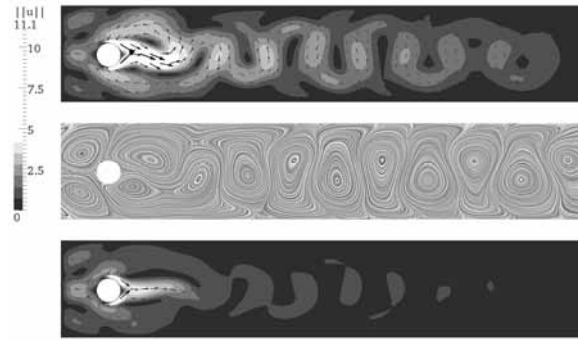


Figure 3:  $u$  at  $t=0.05$  (top) and corresponding SurfaceLIC representation (centre). Bottom:  $u$  at  $t=0.5$ .

The initial flow  $y^0$  is the fully developed nonstationary Navier-Stokes flow, the target flow  $z$  is the stationary Stokes flow. Figure 2 depicts the mesh, the initial condition and the target flow, Figure 3 the control at different points in time. Table 1 gives an overview about the problem size.

Space-Lev.	#vertices	#edges	#elements
1	156	286	130
2	572	1 092	520
3	2 184	4 264	2 080
4	8 528	16 848	8 320

Table 1: Mesh size for different refinement levels in space.

#### 4.1 Newton solver

For the following tests, the described space-time Newton algorithm preconditioned with a space-time

multigrid solver in the control space is chosen. The space-time Newton algorithm was configured to reduce the initial residual by six digits. The space-time multigrid algorithm used in every Newton step reduced its initial residual by two digits using a V-cycle. For smoothing, four steps of a space-time CG method are applied. The coarse grid solver is a CG solver as well which damps its initial residual by two digits. During the calculation of the nonlinear and linear residuals, forward and backward iterations have to be performed. Their local systems in space are solved such that the residual drops below  $10^{-14}$ .

#### Single-grid preconditioning

The first test considers a simple one-level solver configuration, see Table 2. On different refinement levels in space and time the Newton algorithm with embedded CG solver is applied. We use the following notations:

- ‘SLv’ = refinement level in space
- ‘N’ = number of intervals in time
- $T_{\text{opt}}$  = time for the the optimisation.
- $T_{\text{sim}}$  = time needed for the computation of the first forward simulation
- #NL = number of Newton iterations
- $\Sigma_{\text{lin}}$  = sum of all iterations of the linear solver

The number of CG iterations increases slightly with increasing the number of timesteps, so level-independent convergence is nearly lost. The refinement in space does not seem to be relevant for the efficiency of the CG solver. CG only needs about 7-10 iterations to reduce the residual by a factor of  $10^{-2}$  in every Newton iteration, which leads in this example to a ratio of about 30-50 between the simulation and the optimisation.

SLv	N	$T_{\text{opt}}$	$T_{\text{sim}}$	#NL	$\Sigma_{\text{lin}}$	$\frac{\#NL}{\Sigma_{\text{CG}}}$	$\frac{T_{\text{opt}}}{T_{\text{sim}}}$
3	40	4:12:29	6:38	5	35	7.0	38.1
3	80	9:01:58	12:17	5	43	8.6	44.1
3	160	18:38:29	23:56	5	47	9.4	46.7
2	40	42:55	1:09	5	36	7.2	37.1
3	40	4:12:29	6:38	5	35	7.0	38.1
4	40	15:36:08	25:48	5	36	7.2	36.3
2	20	20:33	0:40	5	32	6.4	31.1
3	40	4:12:29	6:38	5	35	7.0	38.1

Table 2: Newton with single-grid CG solver

### Multigrid preconditioning

Table 3 depicts results for different levels using a multigrid solver for the linear subproblems. For the coarse mesh, space level two with 20 timesteps is chosen.

The total number of multigrid iterations is rather constant or even reducing for higher levels, which is an advantage to the one-level solver. However, multigrid needs about 1.5-2.5 iterations per nonlinear iteration, which corresponds to 6-10 CG iterations on the finest level (due to four smoothing steps per multigrid iteration). This is rather similar to the one-level solver test. The ratio between optimisation and simulation is with a factor of 50 higher than in the one-level case. So the additional overhead will pay off only for meshes with a finer mesh resolution in time.

SLv	N	T <sub>opt</sub>	T <sub>sim</sub>	#NL	$\Sigma \text{lin}$	$\frac{\#NL}{\Sigma CG}$	$\frac{T_{opt}}{T_{sim}}$
3	40	5:40:00	6:38	4	8	2.0	51.3
4	80	46:03:22	52:24	5	9	1.8	52.7
5	160	297:26:50	6:13:18	5	8	1.6	47.8

Table 3: Newton with multigrid solver

### 4.2 SQP solver

In this section, the same example is calculated using the SQP approach. The basic test settings are the same. However, it is possible to use different solver settings here as the SQP approach allows solving subproblems inexactly without destroying the solution.

- The outer Newton solver is configured to reduce the nonlinear residual by six digits which is similar to the configuration of the Newton above
- For the inner linear solver, we apply on the one hand a single-grid space-time BiCGStab solver, on the other hand a space-time multigrid solver. Both reduce the residual by two digits. If a space-time multigrid solver is used, the coarse grid solver and the smoother is BiCGStab with a forward-backward simulation used for preconditioning. The smoother applies four (post-) smoothing steps.
- For the solver in space in each time interval, we apply a spatial monolithic multigrid algorithm which reduces its initial residual by two digits. Smoothing is carried out with a local pressure-Schur-Complement ('VANKA'-) like technique.

### Single-grid preconditioning

Table 4 shows the results using the single-grid BiCGStab solver. The time  $T_{sim}$  for the simulation was taken from the Newton approach in the control space, see Table 2.

The number of nonlinear iterations is independent of the level. The number of iterations of the linear solver is depending on the time interval length, it rises slightly if the number of time intervals increases. The refinement level in space does not seem to have an influence to the solver efficiency. This behaviour has also been seen in the test for the Newton approach in the control space.

Comparing optimisation and simulation, the optimisation only needs about 10-20 times longer than the simulation. So in this example, the control-eliminated SQP approach seems to be more efficient than the Newton approach in the control space, compare Table 2.

SLv	N	T <sub>opt</sub>	T <sub>sim</sub>	#NL	$\Sigma \text{lin}$	$\frac{\#NL}{\Sigma CG}$	$\frac{T_{opt}}{T_{sim}}$
3	40	1:53:48	6:38	6	31	5.2	17.2
3	80	2:16:00	12:17	6	31	5.2	11.1
3	160	4:40:35	23:56	6	33	5.5	11.7
2	40	16:12	1:09	6	32	5.3	14.0
3	40	1:53:48	6:38	6	31	5.2	17.2
4	40	7:28:09	25:48	5	26	5.2	17.4
2	20	9:05	0:40	5	25	5.0	13.8
3	40	1:53:48	6:38	6	31	5.2	17.2
4	80	12:09:20	52:19	6	34	5.7	13.9

Table 4: SQP with single-grid solver

### Multigrid preconditioning

Table 5 depicts the results in the case where a multigrid solver is used for preconditioning in space and time.

The convergence behaviour of the multigrid-based approach does not depend on the refinement level in contrast to the single-grid approach. Concerning the absolute time and the ratio between optimisation and simulation, this approach is slightly more expensive than the single-grid approach. One can see a factor of about 20. In total, the complete approach is still only half as expensive in this configuration compared to the Newton approach in the control space, see Table 2 and Table 3.

SLv	N	T <sub>opt</sub>	T <sub>sim</sub>	#NL	$\Sigma \text{lin}$	$\frac{\#NL}{\Sigma CG}$	$\frac{T_{opt}}{T_{sim}}$
3	40	2:22:30	6:38	6	9	1.5	21.5
4	80	16:41:27	52:24	6	10	1.7	19.1

Table 5: SQP with multigrid solver

## 5 Conclusions and Outlook

We presented two modern approaches for the optimal control of nonstationary, incompressible flows. The methods are rather general and can in a similar way also be applied to other types of problems. It can be said that the complete approach can also be extended to boundary control problems. Furthermore, bounds in the control are usually also possible, cf. [3].

The basic idea is to derive a KKT system which models the first order necessary optimality conditions of a given minimisation problem. Depending on the approach, some equations are eliminated, and a Newton solver is applied to the remaining equations. The linear system inside of the Newton is solved with a space-time multigrid approach.

The standard Newton approach in the control space has the advantage that it can be built upon an existing flow solver, as it only needs a forward and a backward solver for the primal/adjoint equations. The SQP approach on the other hand is more intrusive, but (due to more freedom in the choice of the stopping criteria) also more efficient.

In both cases, the multigrid approach leads to level-independent convergence rates. This means that the effort necessary for solving the optimisation problem grows linearly with the problem size, which is optimal. However, the additional overhead renders this approach only useful for very fine discretisations, in particular, with a lot of timesteps.

There are even more methods available, which can be seen as modifications of the above two. In the ‘reduced SQP’ approach for example, one combines the above two approaches (see [1]). Furthermore, special BFGS-based variants are also available which are applicable, e.g., for boundary control problems with only few unknowns. Numerical results for these approaches are, however, not yet available.

## 6 References

- [1] M. Hinze. Optimal and instantaneous control of the instationary Navier–Stokes equations.

Habilitation thesis, Institut für Numerische Mathematik, Technische Universität Dresden, 2000.

- [2] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. Optimization with PDE Constraints, Volume 23 of Mathematical Modelling: Theory and Applications. Springer, Berlin, 2009. ISBN 9781402088384.
- [3] M. Köster. A Hierarchical Flow Solver for Optimisation with PDE Constraints. Phd thesis, TU Dortmund, Lehrstuhl III für Angewandte Mathematik und Numerik, 2011.
- [4] M. Hinze, M. Köster, and S. Turek. A space-time multigrid solver for distributed control of the time-dependent Navier–Stokes system. Preprint SPP1253-16-02, SPP1253, 2008.
- [5] G. Büttner. Ein Mehrgitterverfahren zur optimalen Steuerung parabolischer Probleme. PhD thesis, Fakultät II – Mathematik und Naturwissenschaften der Technischen Universität Berlin, 2004. [http://edocs.tu-berlin.de/diss/2004/buettner\\_guido.pdf](http://edocs.tu-berlin.de/diss/2004/buettner_guido.pdf).
- [6] W. Hackbusch. Multi-Grid Methods and Applications. Springer Series in Computational Mathematics. Springer, Berlin, 1985. ISBN 3-540-12761-5.
- [7] R. E. Bank and T. F. Dupond. An optimal order process for solving finite element equations. Math. Comput., 36(153):35–51, 1981.
- [8] S. Turek and M. Schäfer. Benchmark computations of laminar flow around cylinder. In E.H. Hirschel, editor, Flow Simulation with High-Performance Computers II, Volume 52 of Notes on Numerical Fluid Mechanics, pages 547–566. Vieweg, 1996.
- [9] M. Hinze and K. Kunisch. Three control methods for time-dependent fluid flow. Flow Turbulence Combust., 68:273–298, 2000.

# Mathematical Modelling and Numerical Algorithms for Simulation of Oil Pollution

QUANG A DANG<sup>1</sup>, M. EHRHARDT<sup>2</sup>, GIA LICH TRAN<sup>3</sup>, DUC LE<sup>4</sup>

<sup>1</sup>Institute of Information Technology, 18 Hoang Quoc Viet, Ha noi, Vietnam,  
dangqa@ioit.ac.vn

<sup>2</sup>Lehrstuhl für Angewandte Mathematik und Numerische Analysis, Fachbereich C – Mathematik und Naturwissenschaften, Bergische Universität Wuppertal, Gaußstr. 20, 42119 Wuppertal, Germany,  
ehrhardt@math.uni-wuppertal.de

<sup>3</sup>Institute of Mathematics, 18 Hoang Quoc Viet, Ha noi, Vietnam,  
tranglich@math.ac.vn

<sup>4</sup>Buro of Hydrology and Meteorology, Dang Thai Than, Ha noi, Vietnam,  
Leduc@nchmf.gov.vn

## Abstract

This talk deals with the mathematical modeling and algorithms for the problem of oil pollution. For solving this task we derive the adjoint problem for the advection-diffusion equation describing the propagation of oil slick after an accident, which we call the main problem.

We prove a fundamental equality between the solutions of the main and the adjoint problems. Based on this equality we propose a novel method for the identification of the pollution source location and the accident time of oil emission.

This approach is illustrated on an example for an accident in the offshore of the central part of the Vietnamese coast. Numerical simulations demonstrate the effectiveness of the proposed method. Besides, the method is verified for 1D model of substance propagation.

## References

- [1] A.C. Bagtzoglou, J. Atmadja, *Mathematical Methods for Hydrologic Inversion: The Case of Pollution Source Identification*, Handb. Environ. Chem. **5** (2005), 65–96.
- [2] Q.A. Dang, M. Ehrhardt, *Adequate numerical solution of air pollution problems by positive difference schemes on unbounded domains*, Math. Comput. Modelling **44** (2006), 834–856.
- [3] Q.A. Dang, M. Ehrhardt, G.J. Tran, D. Le, *On the numerical solution of some problems of environmental pollution*, In C. Bodine, ed., Air Pollution Research Advances, Nova Science, 2007, 171–200.
- [4] Q.A. Dang, M. Ehrhardt, G.J. Tran, D. Le, *Mathematical Modelling and Numerical Algorithms for Simulation of Oil Pollution*, Environmental Modeling and Assessment **17** (2012), 275–288.
- [5] Q.A. Dang, M. Ehrhardt, *On Dirac delta sequences and their generating functions*, Appl. Math. Lett. **25** (2012), 2385–2390.
- [6] G.I. Marchuk, *Adjoint Equations and Analysis of Complex Systems*, Kluwer, Dordrecht, 1995.
- [7] E. Milnes, P. Perrochet, *Simultaneous identification of a single pollution point-source location and contamination time under known flow field conditions*, Adv. Water Resources, **30** (2007), 2439–2446.
- [8] Y.N. Skiba, *Direct and adjoint estimates in the oil spill problem*, Rev. Int. Contam. Ambient. **11** (1995), 69–75.
- [9] Y.N. Skiba, *Direct and adjoint oil spill estimates*, Environmental Monitoring and Assessment **59** (1999), 95–109.
- [10] Y.N. Skiba, D. Parra-Guevara, *Mathematics of oil spills: existence, uniqueness, and stability of solutions*, Geofísica Internacional **38** (1999), 117–124.



# Methods for modelling and numerical simulation of fluid flow problems in open channels and closed pipes

T. Jax<sup>1</sup>, G. Steinebach<sup>2</sup>

<sup>1,2</sup>Hochschule Bonn-Rhein-Sieg, FB 03 (EMT), Grantham-Allee 20, 53757 Sankt Augustin, Germany

*tim.jax@smail.emt.h-brs.de*

The simulation of fluid flows is of importance to many fields of application, especially in industry and infrastructure. The modelling equations applied describe a coupled system of non-linear, hyperbolic partial differential equations given by one-dimensional shallow water equations that enable the consistent implementation of free surface flows in open channels as well as pressurised flows in closed pipes. The numerical realisation of these equations is complicated and challenging to date due to their characteristic properties that are able to cause discontinuous solutions.

In the following methods will be presented to solve the system of hyperbolic problems by using the method of lines approach. The semidiscretisation in space is realised by finite volumes and the local Lax-Friedrichs scheme in combination with a central WENO interpolation of third order. To solve the resulting stiff system of differential-algebraic equations integration in time is based on implicit and semi-implicit methods.

## 1 Introduction

The one-dimensional assumption of the shallow-water equations is sufficient for multiple fields of application and can be found especially in flow simulations for large networks of open channels or closed pipes. Here, e.g., the one-dimensional shallow-water equations are used for planning water supply and sewerage in urban infrastructure, for water-level forecast in high water situations and for modelling fluids in process plants. By supplementing additional convection-diffusion-reaction equations it is possible to evaluate transport of soluble components and chemical-biological reactions. Furthermore, gas flows can be simulated due to formal similarity [1].

An established method to solve the space- and time-dependent system of partial-differential equations (PDEs) given by the one-dimensional shallow water equations is the application of the method of lines (MOL) [2]. The MOL is based on a semidiscretisation in space and time: Approximating derivations in space by appropriate discretisation schemes derivations in time remain thus leading to a system of ordinary differential-equations (ODEs) that can be solved by already existing integrators.

In addition, correct implementation of boundary and coupling conditions for the simulation of flows in single reaches and whole networks requires the definition of algebraic equations that convert the system

of ODEs into a system of differential-algebraic equations (DAEs). Due to its stiff properties the DAE system requires the application of suitable implicit and semi-implicit methods [3]. Therefore ROW-methods [2], [4] have been chosen as preferred integrators, but other schemes like PEER-methods show positive results, too [5].

The intention of this article is to present a current approach to evaluate the one-dimensional shallow water equations by using the MOL. Because of the application of existing integrators to solve the resulting DAE system the focus will be on the description of methods for semidiscretisation in space that are capable to solve possible discontinuous solutions due to the non-linear, hyperbolic properties with adequate accuracy and that satisfy the problem's special requirements.

In the following section the applied modelling equations are given. A common formulation in literature will be presented as well as a description to evaluate alternative state variables. Section 3 deals with the methods used for semidiscretisation in space. Section 4 summarises the realisation of algebraic equations to describe boundary and coupling conditions. The application to a test problem given by the dam break problem will be considered in section 5.

## 2 Modelling equations

### 2.1 One-dimensional shallow water equations

The system of one-dimensional shallow water equations considered below allows the consistent simulation of free surface flows in open channels and pressurised flows in closed pipes. It results from an extension of the so-called Saint-Venant equations that are restricted to one-dimensional free surface flows [6].

A common formulation in literature is based on a calculation of the state variables  $(\rho A)$  and  $(\rho q)$  [3]:

$$(\rho A)_t + (\rho q)_x = q_s \quad (1)$$

$$(\rho q)_t + \left( \frac{(\rho q)^2}{(\rho A)} \right)_x + g(\rho A) \times \varphi_x + \frac{A}{\beta \rho_0} \times \rho_x = -g(\rho A) \times S_f \quad (2)$$

Here, equation (1) describes the conservation of mass and equation (2) the conservation of momentum. Parameter  $A$  denotes the flooded cross-sectional area,  $\rho$  the flowing fluid's density and  $q$  the volume flow, i.e. the product by the flooded area  $A$  and the average flow velocity  $u$  ( $q=Au$ ). By using  $q_s$  lateral incoming and outgoing mass-flows can be considered.

The parameter  $z$  is the water surface elevation above reference level  $NN$  (fig. 1). The water level elevation is the sum given by water depth  $h$  and bottom elevation  $S_0$  ( $z=h+S_0$ ). It must be a distinct function of place  $x$  and flooded cross-section  $A$  [3].

The constants  $g$ ,  $\beta$  and  $\rho_0$  denote the gravitational acceleration ( $g=9.81\text{m/s}^2$ ), the coefficient of isothermal compressibility ( $\beta=5 \cdot 10^{-10}\text{m}^2/\text{N}$ ) and the density of the fluid for free surface flows, i.e.  $\rho_0=1000\text{kg/m}^3$  for water. In the case of free surface flows in open channel geometries a constant density  $\rho=\rho_0$  is assumed.

The friction slope  $S_f$  considers effects due to turbulence, friction and viscosity. According to the empirical formula by Manning-Strickler the friction slope is [3]:

$$S_f = u|u| \frac{1}{k_{St}^2 r_{hydr}^{4/3}} \quad (3)$$

The Strickler coefficient  $k_{St}$  describes a friction parameter whose values can be found in [7]. The hydraulic radius  $r_{hydr}$  denotes the relation of the flooded cross-sectional area  $A$  and the wetted perimeter  $U$  ( $r_{hydr}=A/U$ ).

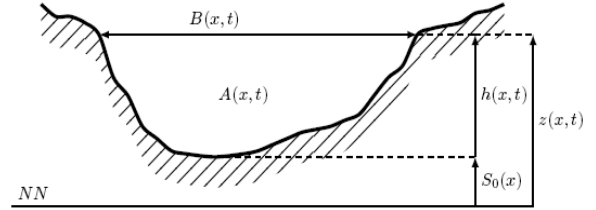


Figure 1. Channel cross-section.

### 2.2 Alternative formulation

In many applications the water surface elevation  $z$  at a specific time is more interesting than the flooded cross-sectional area  $A$ . Furthermore, the correct simulation of the balance solution of the so-called still-water test problem requires an equivalent formulation of the conservation of mass (1) that depends on the water surface elevation  $z$  explicitly [3], [8]. Therefore a formulation allowing the direct evaluation of the state variables  $(\rho z)$  and  $(\rho q)$  seems to be recommendable and will be considered below.

The evaluation of the new state variables results from the relation

$$(\rho A)_t = B(\rho z)_t \quad (4)$$

that allows the mentioned equivalent formulation of the conservation of mass (1) [3], [8]. Equation (4) is valid for free surface fluid flows in open channel geometries with rectangular, trapezoidal and circular cross-section as well as pressurised flows in circular closed pipes. The proof is given in [9].

Substituting equation (4) into (1) the one-dimensional shallow water equations to evaluate the state variables  $(\rho z)$  and  $(\rho q)$  write [9]:

$$(\rho z)_t + \frac{1}{B}(\rho q)_x = \frac{1}{B}q_s \quad (5)$$

$$(\rho q)_t + \left( \frac{(\rho q)^2}{(\rho A)} \right)_x + g(\rho A) \times \varphi_x + \frac{A}{\beta \rho_0} \times \rho_x = -g(\rho A) \times S_f \quad (6)$$

In the case of free surface flows  $B$  denotes the real water surface width (fig. 1) that can be determined by the given channel geometries.

For pressurised flows in completely flooded circular pipes the real water surface width  $B$  becomes zero. Therefore it can not be applied to the conservation of mass (5). Hence an alternative formulation for  $B$  is required to ensure consistent implementation. Assuming inelastic pipes, thus constant water level elevation  $z$  and flooded cross-sectional area  $A$  in time, it becomes  $B = A/z$  from equation (4) [9].



Fitting factor  $B$  to the given flow conditions and channel geometries the alternative formulation of the one-dimensional shallow water equations (5), (6) to evaluate the state variables  $(\rho z)$  and  $(\rho q)$  can be generally applied to free surface and pressurised flows. In the following the description of these modelling equations in matrix and vector notation is relevant. It writes:

$$\mathbf{u}_t + \mathbf{N}(\mathbf{u}) \mathbf{f}_x(\mathbf{u}) + \mathbf{M}(\mathbf{u}) \mathbf{g}_x(\mathbf{u}) = \mathbf{s}(\mathbf{u}) \quad (7)$$

with

$$\mathbf{u} = \begin{pmatrix} (\rho z) \\ (\rho q) \end{pmatrix}, \quad \mathbf{N}(\mathbf{u}) = \begin{bmatrix} 1/B & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{f}(\mathbf{u}) = \begin{pmatrix} (\rho q) \\ (\rho q)^2 / (\rho A) \end{pmatrix},$$

$$\mathbf{g}(\mathbf{u}) = \begin{pmatrix} z \\ \rho \end{pmatrix}, \quad \mathbf{M}(\mathbf{u}) = \begin{bmatrix} 0 & 0 \\ g(\rho A) & \frac{0}{\beta \rho_0} \end{bmatrix}, \quad \mathbf{s}(\mathbf{u}) = \begin{pmatrix} q_s / B \\ -g(\rho A) \times S_f \end{pmatrix}.$$

### 3 Semidiscretisation in space

#### 3.1 Requirements

The one-dimensional shallow water equations describe a numerical ambitious system of non-linear, hyperbolic PDEs. Due to these non-linear, hyperbolic properties discontinuous solutions that complicate the numerical simulation can be induced after a finite time even for given smooth initial values. Therefore the methods for semidiscretisation in space applied using the MOL must be capable of reproducing discontinuous solutions with sufficient accuracy and evaluating their velocity of propagation correctly. Furthermore, the possible simultaneous propagation of wave fronts into different directions requires the application of direction-independent methods that consider the coupling between conservation of mass and momentum in addition.

Besides, system (7) is not given in typical form of conservation laws. The conservative formulation is perturbed by  $\mathbf{N}(\mathbf{u})$  due to factor  $1/B$  in the alternative continuity equation (5) that is not allowed to be differentiated and the so-called non-conservative product  $\mathbf{M}(\mathbf{u}) \mathbf{g}_x(\mathbf{u})$  due to parts in the momentum equation (6). To ensure a complete semidiscretisation in space these components must be considered separately.

An appropriate approach to realise the semidiscretisation of the one-dimensional shallow water equations accounting for these specific requirements is the method described by Kurganov and Levy [10]. It is based on finite volumes with local Lax-Friedrichs scheme and a central WENO interpolation of third-order.

#### 3.2 Finite volumes and WENO-interpolation

To evaluate hyperbolic problems a discretisation with finite volumes is most suitable for most cases. Contrary to other methods like finite differences or finite elements it requires no continuous solution and therefore is advantageous.

Applied to the conservation law

$$u_t + f_x(u) = 0 \quad (8)$$

the finite volumes lead to a general conservative scheme [11], [12] that reads in semi-discretised form:

$$\frac{du_i}{dt} = -\frac{1}{\Delta x} [F(u_{i+1/2}) - F(u_{i-1/2})]. \quad (9)$$

The conservative scheme holds the conservation properties of equation (8) thus reproducing the correct propagation velocity of discontinuous solutions [12]. Value  $u_i(t)$  denotes the integral over cell  $V_i$ . The function values  $u_{i\pm 1/2}$  at the cell boundaries must be computed from the cell means  $u_i(t)$ , which leads to local Riemann problems with left and right intermediate values  $u_{i\pm 1/2}^\pm$  at the cell boundaries (fig. 2).

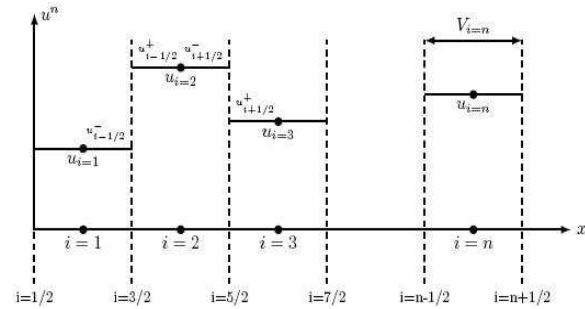


Figure 2. Intermediate values  $u_{i\pm 1/2}^\pm$  at cell boundaries.

Left and right intermediate values can be determined by an interpolation of the cell means onto the cell boundaries. Below, a central WENO interpolation of third order will be applied. It is based on the construction of interpolation polynomials for each cell using the values  $u_i(t)$ . These polynomials can be evaluated at the cell boundaries  $x_{i\pm 1/2}$ . Referring to this, values  $u_{i+1/2}^+$  are determined using the cell means  $u_i$ ,  $u_{i+1}$  and  $u_{i+2}$ . The values  $u_{i+1/2}^-$  result from  $u_{i-1}$ ,  $u_i$  and  $u_{i+1}$  [3], [10].

Compared to interpolation methods of lower order the WENO interpolation of third order reproduces discontinuous solutions with sufficient accuracy by using conservative schemes [10]. Numerical examples and the implementation of the central WENO interpolation are given in [3], [10].

### 3.3 Local Lax-Friedrichs scheme

Semidiscretisation in space of the one-dimensional shallow water equations requires the use of direction-independent conservative schemes that can be applied to coupled systems. Here, the local Lax-Friedrichs scheme (Rusanov scheme) proves to be adequate.

Considering just the discretisation in space according to MOL approach the local Lax-Friedrichs scheme referred to a hyperbolic system of conservation laws

$$\mathbf{u}_t + \mathbf{f}_x(\mathbf{u}) = 0 \quad (10)$$

is according to Kurganov and Levy [10] given by

$$\frac{d\mathbf{u}_i}{dt} = -\frac{1}{\Delta x} [\mathbf{F}_{i+1/2}^{LF} - \mathbf{F}_{i-1/2}^{LF}] \quad (11)$$

with the numerical fluxes

$$\mathbf{F}_{i\pm 1/2}^{LF} = \frac{1}{2} [\mathbf{f}(\mathbf{u}_{i\pm 1/2}^+) + \mathbf{f}(\mathbf{u}_{i\pm 1/2}^-)] - \frac{\lambda_{i\pm 1/2}}{2} [\mathbf{u}_{i\pm 1/2}^+ - \mathbf{u}_{i\pm 1/2}^-].$$

Here,  $\lambda$  denotes the so-called numerical viscosity. It must be chosen identically for the conservation equations of mass and momentum to consider their mutual coupling. Formulation (11) allows the application of the interpolation of higher order described in section 3.2 to determine the intermediate values  $\mathbf{u}_{i\pm 1/2}^\pm$ .

The local Lax-Friedrichs scheme is a centred method and thus independent of the wave fronts' propagation directions [13]. In contrast to other schemes it requires no approximate Riemann solvers. Therefore its implementation is comparatively simple [10].

### 3.4 Discretisation of the modelling equations

The application of the local Lax-Friedrichs scheme is restricted to the conservative parts of the one-dimensional shallow water equations. Given non-conservative parts in system (7) must be considered separately to obtain the complete semidiscretisation in space. Referring to this, the realisation of factor  $1/B$  that is not differentiated according to equation (5) requires the separation of flux and flux correction of the local Lax-Friedrichs scheme. Components of the non-conservative product  $\mathbf{M}(\mathbf{u})\mathbf{g}_x(\mathbf{u})$  must be realised by appropriate difference quotients. Thus, the complete semidiscretisation in space yields [9]

$$\begin{aligned} \frac{d\mathbf{u}_i}{dt} = & \mathbf{s}(\mathbf{u}_i) - \mathbf{N}(\mathbf{u}_i) \frac{1}{\Delta x} [\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}] + \dots \\ & \dots \frac{1}{\Delta x} [\mathbf{LF}_{i+1/2} - \mathbf{LF}_{i-1/2}] - \mathbf{M}(\mathbf{u}_i) \frac{1}{\Delta x} [\mathbf{G}_{i+1/2} - \mathbf{G}_{i-1/2}] \end{aligned} \quad (12)$$

with flux terms

$$\mathbf{F}_{i\pm 1/2} = \frac{1}{2} [\mathbf{f}(\mathbf{u}_{i\pm 1/2}^+) + \mathbf{f}(\mathbf{u}_{i\pm 1/2}^-)],$$

flux corrections

$$\mathbf{LF}_{i\pm 1/2} = \frac{\lambda_{i\pm 1/2}}{2} [\mathbf{u}_{i\pm 1/2}^+ - \mathbf{u}_{i\pm 1/2}^-],$$

and the components of the non-conservative produkt

$$\mathbf{G}_{i\pm 1/2} = \frac{1}{2} [\mathbf{g}(\mathbf{u}_{i\pm 1/2}^+) + \mathbf{g}(\mathbf{u}_{i\pm 1/2}^-)].$$

Equation (12) leads to an ODE for each cell. Values  $\mathbf{u}_{i\pm 1/2}^\pm$  to the left and right of the cell boundaries result from the central WENO interpolation. The matrices  $\mathbf{N}(\mathbf{u})$  and  $\mathbf{M}(\mathbf{u})$  as well as the source term  $\mathbf{s}(\mathbf{u})$  must be applied to the cell centres.

The identical numerical viscosity  $\lambda$  for conservation of mass and momentum results from the eigenvalues given by the Jacobian matrix of the one-dimensional shallow water equations. It is for modelling equations (1), (2) as well as (5), (6)  $\lambda = u \pm \sqrt{gA/B}$  for free surface and  $\lambda = u \pm \sqrt{1/(\beta\rho_0)}$  for pressurised flows.

In implementation the largest absolute eigenvalue given locally at the cell boundaries is considered [3].

## 4 Differential-algebraic equations

The evaluation of the one-dimensional shallow water equations requires the realisation of algebraic equations to describe boundary and coupling conditions for the simulation of single reaches and whole networks [1]. The algebraic equations define a stiff semi-explicit DAE system of the form  $\mathbf{M}\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y})$  that is solved using implicit and semi-implicit integrators [3].

In implementation numerical and physical boundary conditions must be distinguished: First, numerical boundary conditions are realised considering function values  $\mathbf{u}_L = \mathbf{u}(x_{i=1/2}, t)$  and  $\mathbf{u}_R = \mathbf{u}(x_{i=n+1/2}, t)$  at the external cell boundaries of the single reaches as additional unknown parameters and subtracting extrapolation values. Thus the algebraic equations yield [3]

$$0 = \mathbf{u}_L - \mathbf{u}_L^{ex} \quad \text{and} \quad 0 = \mathbf{u}_R - \mathbf{u}_R^{ex},$$

with  $\mathbf{u}_{L,R}^{ex}$  resulting from linear extrapolation:

$$\mathbf{u}_L^{ex} = \frac{1}{2} (3\mathbf{u}_1 - \mathbf{u}_2) \quad \text{and} \quad \mathbf{u}_R^{ex} = \frac{1}{2} (3\mathbf{u}_n - \mathbf{u}_{n-1}).$$

Physical boundary conditions are realised by replacing the extrapolation values by prescribed boundary conditions at corresponding points [3].

Coupling conditions can be defined in different ways [14]. A rather simple method is given by using algebraic equations to equalise the state variables at the exit of a reach with the corresponding values at the entrance of the following reach [3]:

$$0 = (\rho z)_{reachI,out} - (\rho z)_{reachII,in}$$

$$0 = (\rho q)_{reachI,out} - (\rho q)_{reachII,in}$$

## 5 Numerical application

An appropriate test problem to analyse the validity of the alternative one-dimensional shallow water equations (5), (6) and the efficiency of the methods described above is the dam break problem. It considers a dam causing two different water depths thus leading to a Riemann problem with discontinuous distribution of initial values for the water level elevation  $z$  (fig. 3). The water is assumed to be initially at rest ( $q(x, t_0)=0$ ) over the entire interval  $x=[0,1000]$  of a rectangular open channel with constant width  $B=1$  and bottom elevation  $S_0=0$ .

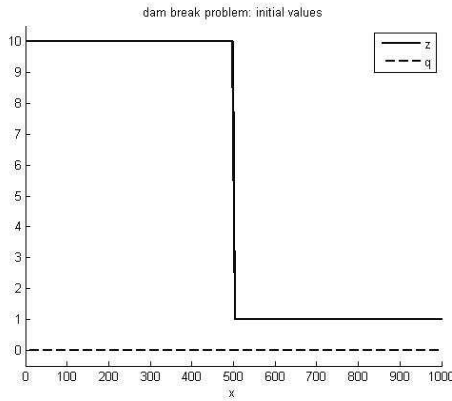


Figure 3. Dam break problem: Initial values.

At the initial time  $t_0=0$  the dam is destroyed and the temporal development of water level elevation and volume flow during the period  $t=[0,30]$  is simulated. The problem will show a characteristic behaviour for the water level elevation with a rarefaction wave propagating from right to left and a discontinuity in form of a shock wave moving from left to right simultaneously. Therefore the test problem is able to verify if the methods for discretisation in space are adequate to simulate wave propagation into different directions and to reproduce discontinuous solutions.

The analytical solution to this test problem can be found in [3], [5].

A comparison of numerical and analytical solutions for water level elevation  $z$  and volume flow  $q$  at final time  $t=30$  is given in fig. 4 and fig. 5. Results were obtained by implementing the described methods in MATLAB using  $n=200$  cells and integrator *rodasp* [15] for integration in time. The analytical solutions are given by a solid line, circles mark numerical solutions at the discrete points. The numerical results are in good accordance with the analytical ones.

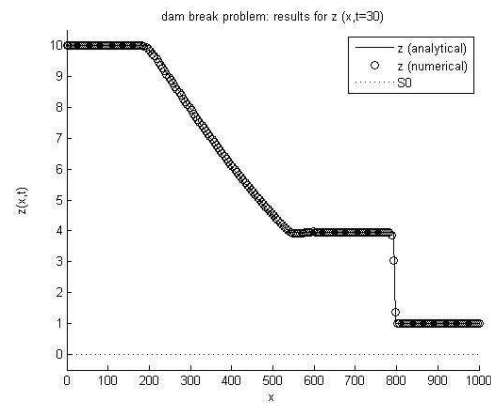


Figure 4. Dam break problem: Results for  $z$ .

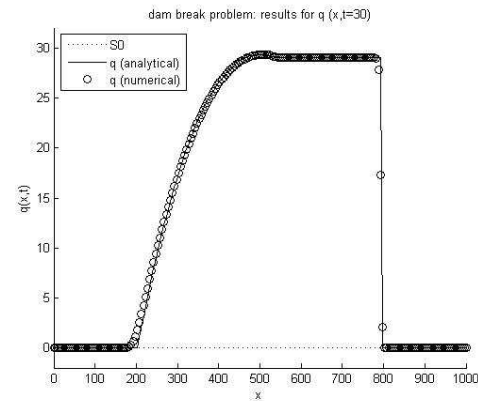


Figure 5. Dam break problem: Results for  $q$ .

## 6 Conclusion and outlook

The presented methods offer convenient numerical behaviour and satisfying results when simulating the dam break problem as well as other test problems given in [3], [8]. In current implementation the usage of the alternative state variables ( $pz$ ) and ( $pq$ ) shows no advantages compared to evaluations based on the established variables ( $\rho A$ ) and ( $pq$ ) up to now.

The local Lax-Friedrichs scheme leads to satisfying results for almost all of the analysed test problems. An exception are problems with transitions from subcritical to supercritical flow conditions [7], [8]. Here, the local Lax-Friedrichs scheme causes incorrect values for volume flow  $q$  at the transition points. To avoid this error other schemes described in [8] can be implemented.

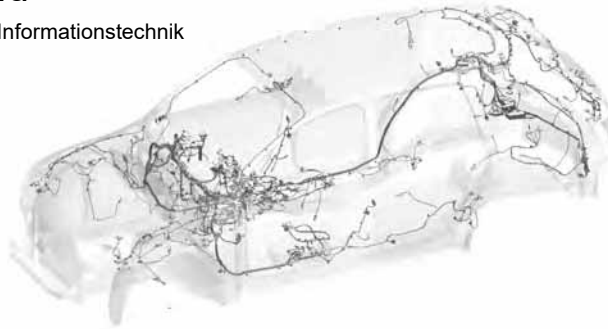
Objective of the continuing project is the optimisation of the applied methods referring to robustness and calculating time to simulate large networks of channels and pipes. Therefore alternative formulations of the modelling equations and the application of adaptive as well as multirate and partitioned methods are analysed presently.

### Acknowledgements

Special thanks to the Graduate Institute of the Bonn-Rhein-Sieg University of Applied Sciences for financial support.

## 7 References

- [1] G. Steinebach: Mathematische Modellbildung und numerische Methoden zur Strömungs-, Transport- und Reaktionssimulation in Netzwerken, in: Forschungsspitzen und Spitzenforschung - Innovationen an der Fachhochschule Bonn-Rhein-Sieg, Ch. Zacharias et al. (eds.), Physica-Verlag, Germany, pp. 151-163, 2009
- [2] M. Hilden, G. Steinebach: Eno-discretizations in MOL-applications: some examples in river hydraulics, Applied Numerical Mathematics, pp. 293-308, 1998
- [3] G. Steinebach, R. Rosen: TestSet zur Evaluierung numerischer Methoden für Kanal- und Rohrströmungsprobleme, Siemens AG (CT PP 2), Germany, 2010
- [4] G. Steinebach, P. Rentrop: An adaptive method of lines approach for modelling flow and transport in rivers, in: Adaptive method of lines, A. Vande Wouwer et al. (eds.), Chapman & Hall/CRC, pp.181-205, USA, 2001
- [5] G. Steinebach, R. Weiner: Peer methods for the one-dimensional shallow water equations with CWENO space discretization, Applied Numerical Mathematics, pp.1567-1578, 2012
- [6] C. Bourdarias, S. Gerbi: A finite volume scheme for a model coupling free surface and pressurised flows in pipes, Journal of Computational and Applied Mathematics, pp. 109-131, 2007
- [7] H. Chanson: The Hydraulics of Open Channel Flow: An Introduction, 2. Edition, Elsevier, UK, 2009
- [8] G. Steinebach, R. Rosen: Stationäre Lösungen und ausbalancierte Diskretisierungen für Rohr- und Kanalströmungsmodelle, Siemens AG (CT PP 2), Germany, 2012
- [9] T. Jax: Modellbildung und numerische Simulation von Kanal- und Rohrströmungsproblemen, Master-Thesis, Hochschule Bonn-Rhein-Sieg, Germany, 2012
- [10] A. Kurganov, D. Levy: A third-order semidiscrete central scheme for conservation laws and convection-diffusion equations, SIAM J. Sci. Comput., pp. 1461-1488, 2000
- [11] C.-D. Munz, T. Westermann: Numerische Behandlung gewöhnlicher und partieller Differenzialgleichungen - Ein interaktives Lehrbuch für Ingenieure, Springer, Germany, 2006
- [12] R. M. M. Mattheij et al. : Partial Differential Equations - Modeling, Analysis, Computation, SIAM, USA, 2005
- [13] E. F. Toro: Shock-Capturing Methods for Free-Surface Shallow Flows, John Wiley & Sons, UK, 2001
- [14] G. Steinebach, S. Rademacher, P. Rentrop, M. Schulz: Mechanisms of coupling in river flow simulation systems, Journal of Computational and Applied Mathematics, pp. 459-470, 2004
- [15] G. Steinebach: Order-reduction of ROW-methods for DAEs and Method of Lines Applications, Preprint 1741, TH Darmstadt, Germany, 1995



**Simulationsbasierte Optimierung von ESD-  
Schutzkonzepten für Integrierte Schaltungen**

Dipl.-Ing. Stanislav Scheier, Dipl.-Ing. Ali Razavi, Prof. Dr.-Ing. Stephan Frei  
Arbeitsgebiet Bordsysteme

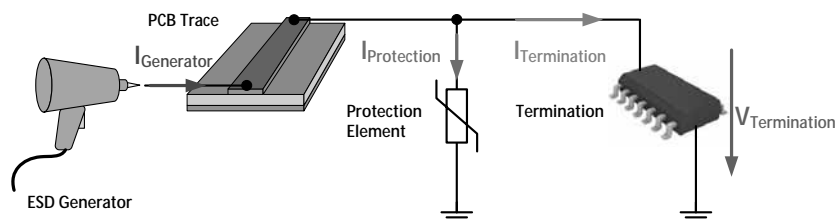


**Gliederung**

- **Einleitung**
- Modellierung
- ESD-Online-Simulator
- Zusammenfassung und Ausblick

### Motivation

- Elektrostatische Entladungen (engl. **ElectroStatic Discharge ESD**) stellen eine ernstzunehmende Bedrohung in der Elektronik-Industrie dar.
- Ein Mensch oder eine Maschine können sich bis auf mehrere kV aufladen und empfindliche Elektronik schädigen. Aufladungen bis ~3 kV nicht spürbar!
- Die ESD-Zerstörfestigkeit vieler Systeme wird durch den Strompfad über die einzelnen Komponenten vom Stecker bis zum Elektronik-IC-Pin bestimmt.
- Durch Simulationen können Bedrohung und Wirksamkeit von Schutzstrategien bereits im Vorfeld untersucht und optimiert werden.



Dipl.-Ing. Stanislav Scheier, Dipl.-Ing. Ali Razavi, Prof. Dr.-Ing. Stephan Frei – Arbeitsgebiet Bordsysteme  
Simulationsbasierte Optimierung von ESD-Schutzkonzepten für Integrierte Schaltungen

3



### Ziele und Anforderungen

- Elektronikentwickler sind meist mit Modellierungs- und Simulationsverfahren nicht vertraut.
- Einfaches Tool zur Simulation von der Belastung in Elektroniksystemen notwendig.
- Geeigneten Modelle notwendig, die mit Messungen verifiziert und parametrisiert wurden.
- Einfache Erweiterbarkeit mit neuen Modellen
- Implementierung in einem Open-Source Simulator (Qucs)
- Hervorhebung fundamentaler Effekte bei ESD

Dipl.-Ing. Stanislav Scheier, Dipl.-Ing. Ali Razavi, Prof. Dr.-Ing. Stephan Frei – Arbeitsgebiet Bordsysteme  
Simulationsbasierte Optimierung von ESD-Schutzkonzepten für Integrierte Schaltungen

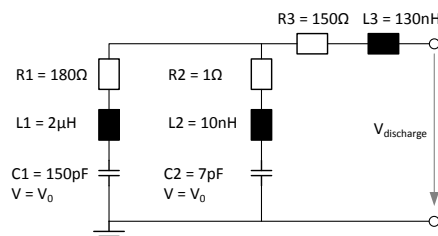
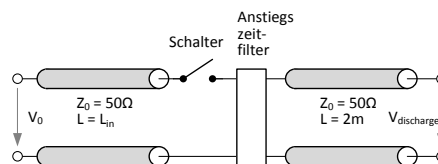
4



- Einleitung
- **Modellierung**
- ESD-Online-Simulator
- Zusammenfassung und Ausblick

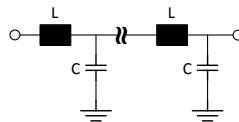
## ESD Generator

- **Transmission Line Pulser (TLP)**
  - Erzeugt einen Rechteckspannungspuls mit variabler Pulshöhe, -breite und Anstiegszeit.
  - Gängiges Messsystem, um die IV-Kurve eines nichtlinearen Bauelements auch bei hohen Spannungen zu messen
  - Prüfmittel / ESD-Nachbildung
- **ESD Generator nach IEC61000-4-2 Standard**
  - Prüfstandard für elektronische Systeme
  - In EU ist die Prüfung Pflicht für nahezu alle elektronischen Geräte.
  - Simuliert eine menschliche Entladung über einen metallischen Gegenstand.



### Platinenleiterbahnen (PCB)

- Platinenleiterbahnen können aufgrund von Reflexionen die Pulsform stark beeinflussen.
- Aufgrund der meist geringen Leiterbahnlänge können die entstehenden Verluste vernachlässigt werden.
- Geringe Leiterbahnlängen können bei den Qucs-Leitungsmodellen zu Instabilitäten führen.
- Approximation mit kaskadierten LC-Netzwerken numerisch günstiger.
- Für jede Länge kann ein entsprechendes LC-Netzwerk erstellt werden.



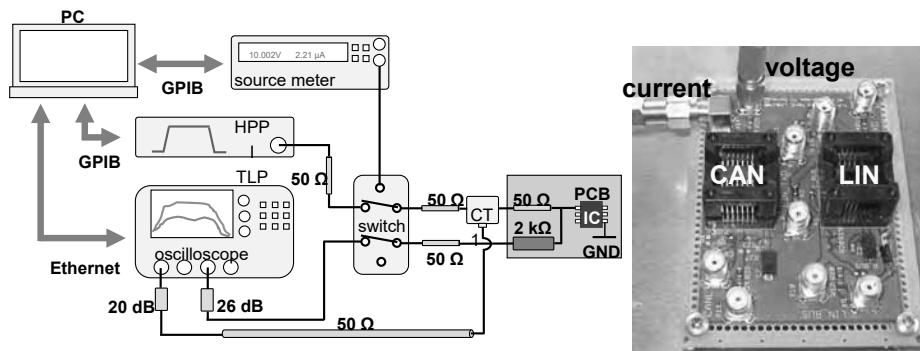
### Lineare Schutzelemente und Terminierungen

- Kondensatoren
  - Bevorzugte Schutzstrategie aufgrund der geringeren Kosten und der guten Eignung für den EMV-Schutz.
  - Beeinflussen negativ die Signalübertragungsrate aufgrund des Tiefpassverhaltens.
- Widerstände
  - Lineare Widerstände veranschaulichen die Funktionsweise verschiedener Strukturen.
  - Können als einfache Approximation für unbekannte Elektronik-ICs verwendet werden.



## Nichtlineare Schutzelemente und Terminierungen, Messaufbau

- Statisches Verhalten wird mit Kennlinienschreiber gemessen
- Verhalten bei höheren Spannungen wird mit TLP gemessen
- IC-Zerstörung wird über eine Veränderung der statischen Kennlinie detektiert
- Parasitäre Elemente werden mit NWA bestimmt.



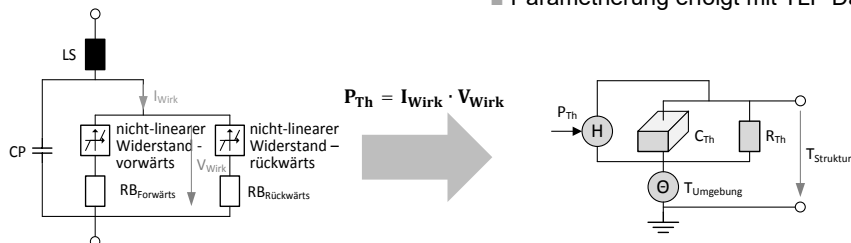
Dipl.-Ing. Stanislav Scheier, Dipl.-Ing. Ali Razavi, Prof. Dr.-Ing. Stephan Frei – Arbeitsgebiet Bordsysteme  
Simulationsbasierte Optimierung von ESD-Schutzkonzepten für Integrierte Schaltungen

9



## Elektronik-IC

- Verschiedene IC-pins wurden bisher untersucht und modelliert
- IC elektrische Domäne
  - Das elektrische Ersatzschaltbild entspricht einem nicht-linearen Widerstand.
- IC thermische Domäne
  - Unter hoher Spannung wird ein Teil des Siliziums leitfähig, die eingepreßte Leistung/Energie wird in diesem Teilvolumen umgesetzt
  - Thermische Ersatzschaltung besteht aus RC Element und gesteuerter Wärmequelle.
  - Schmelztemperatur von Silizium wird als kritischer Grenzwert gesetzt.
  - Parametrierung erfolgt mit TLP Daten



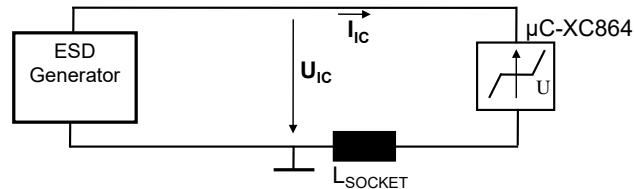
Dipl.-Ing. Stanislav Scheier, Dipl.-Ing. Ali Razavi, Prof. Dr.-Ing. Stephan Frei – Arbeitsgebiet Bordsysteme  
Simulationsbasierte Optimierung von ESD-Schutzkonzepten für Integrierte Schaltungen

10

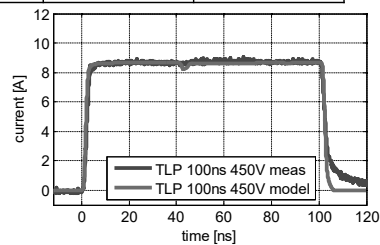
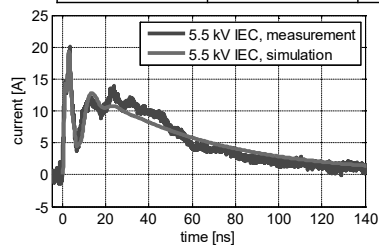


## Elektronik-IC

### ■ Verifikation von einem Mikrokontroller µC-XC864 Modell



ESD pulse	V <sub>charge</sub> [V]	V <sub>IC, max</sub> [V]	I <sub>IC, max</sub> [A]	E <sub>fail</sub> [µJ]
IEC NoiseKen	5500	53.42	27.07	11.30 (0%)
TLP 100 ns	450	95.4	9.0	12.3 (9%)



Dipl.-Ing. Stanislav Scheier, Dipl.-Ing. Ali Razavi, Prof. Dr.-Ing. Stephan Frei – Arbeitsgebiet Bordsysteme  
Simulationsbasierte Optimierung von ESD-Schutzkonzepten für Integrierte Schaltungen

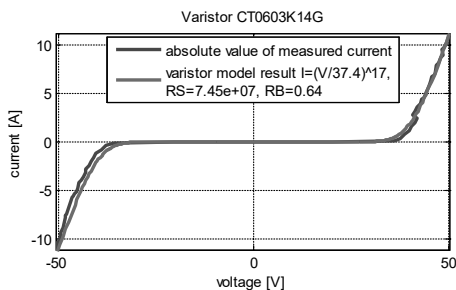
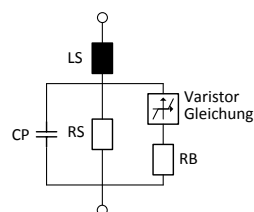
11



## Nichtlineares Schutzelement, Varistor

- Spezielle nichtlineare Komponenten
- Weisen eine geringe parasitäre Kapazität auf
- Varistor Gleichung:

$$I_{var} = \left( \frac{V_{var}}{V_{1A}} \right)^n$$



Dipl.-Ing. Stanislav Scheier, Dipl.-Ing. Ali Razavi, Prof. Dr.-Ing. Stephan Frei – Arbeitsgebiet Bordsysteme  
Simulationsbasierte Optimierung von ESD-Schutzkonzepten für Integrierte Schaltungen

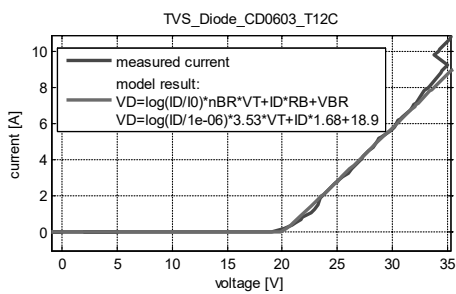
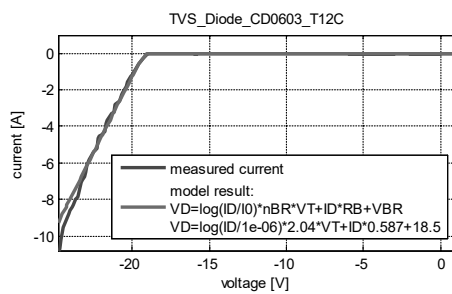
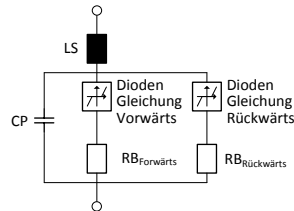
12



### Nichtlineares Schutzelement, TVS Diode

- Spezielle nichtlineare Komponenten
- Weisen eine geringe parasitäre Kapazität auf
- Dioden Gleichung:

$$I_{Diode} = I_S \left( e^{\frac{V_{Diode}}{nV_T}} - 1 \right)$$



Dipl.-Ing. Stanislav Scheier, Dipl.-Ing. Ali Razavi, Prof. Dr.-Ing. Stephan Frei – Arbeitsgebiet Bordsysteme  
Simulationsbasierte Optimierung von ESD-Schutzkonzepten für Integrierte Schaltungen

13



- Einleitung
- Modellierung
- **ESD-Online-Simulator**
- Zusammenfassung und Ausblick

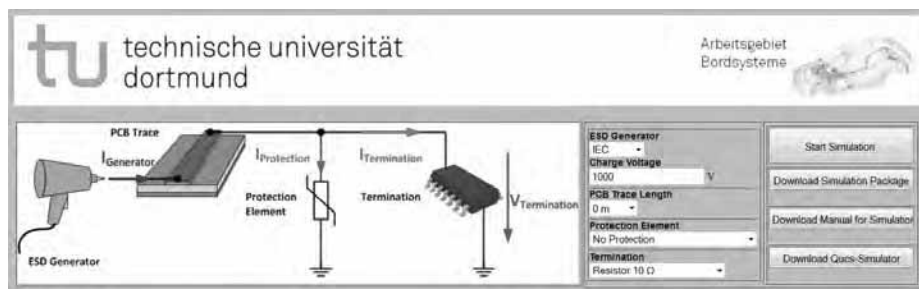
Dipl.-Ing. Stanislav Scheier, Dipl.-Ing. Ali Razavi, Prof. Dr.-Ing. Stephan Frei – Arbeitsgebiet Bordsysteme  
Simulationsbasierte Optimierung von ESD-Schutzkonzepten für Integrierte Schaltungen

14



## Startansicht

- Benutzerfreundliches Tool basierend auf dem Open-Source Simulator Qucs
- Zu finden unter [http://web-bs/esd\\_tool.html](http://web-bs/esd_tool.html) oder über die AG Bordsysteme Homepage <http://www.bordsysteme.tu-dortmund.de>
- Simulationstopologie:
  - Starke Vereinfachung eines Steuergerätes
  - Berücksichtigung relevanter Bauelemente
- Alle Modelle sowie Qucs-Simulator können heruntergeladen werden
  - Komplexere Simulationen sind möglich

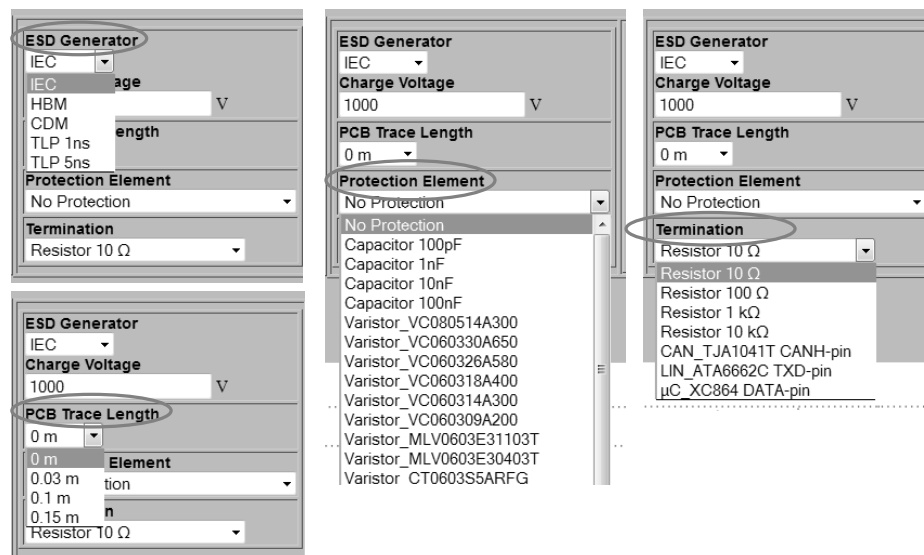


Dipl.-Ing. Stanislav Scheier, Dipl.-Ing. Ali Razavi, Prof. Dr.-Ing. Stephan Frei – Arbeitsgebiet Bordsysteme  
Simulationsbasierte Optimierung von ESD-Schutzkonzepten für Integrierte Schaltungen

15



## Auswahlbereich

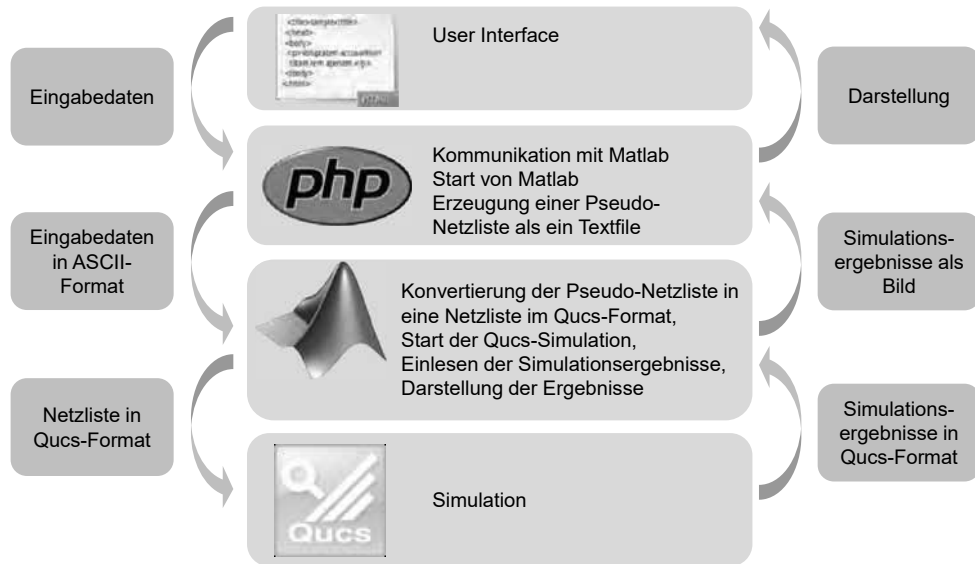


Dipl.-Ing. Stanislav Scheier, Dipl.-Ing. Ali Razavi, Prof. Dr.-Ing. Stephan Frei – Arbeitsgebiet Bordsysteme  
Simulationsbasierte Optimierung von ESD-Schutzkonzepten für Integrierte Schaltungen

16



## Ablauf



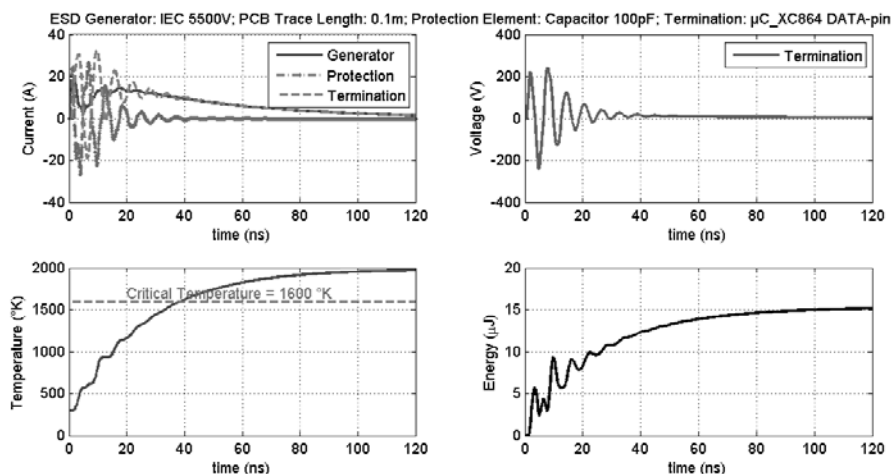
Dipl.-Ing. Stanislav Scheier, Dipl.-Ing. Ali Razavi, Prof. Dr.-Ing. Stephan Frei – Arbeitsgebiet Bordsysteme  
Simulationsbasierte Optimierung von ESD-Schutzkonzepten für Integrierte Schaltungen

17



## Ergebnisse

- Wenn die kritische Temperatur von ca. 1600° C überschritten wird, gilt der IC-Pins als beschädigt



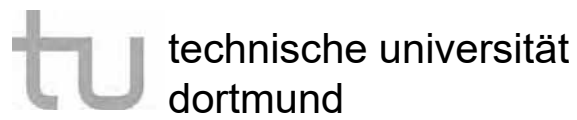
Dipl.-Ing. Stanislav Scheier, Dipl.-Ing. Ali Razavi, Prof. Dr.-Ing. Stephan Frei – Arbeitsgebiet Bordsysteme  
Simulationsbasierte Optimierung von ESD-Schutzkonzepten für Integrierte Schaltungen

18



- Einleitung
- Modellierung
- ESD-Online-Simulator
- **Zusammenfassung und Ausblick**

- Ein benutzerfreundlicher ESD-Simulator wurde entwickelt.
- Abschätzung des realen ESD-Verhaltens von Elektroniksystemen ist möglich.
- Umfangreiche Modellbibliothek verfügbar.
- Erweiterung mit neuen Modellen ist geplant, z.B. spannungsabhängige Kondensatoren



Fakultät für Elektrotechnik und Informationstechnik  
Arbeitsgebiet Bordsysteme

**Vielen Dank für Ihre Aufmerksamkeit**

Dipl.-Ing. Stanislav Scheier  
[stanislav.scheier@tu-dortmund.de](mailto:stanislav.scheier@tu-dortmund.de)  
<http://www.bordsysteme.tu-dortmund.de>







# Systemkonzept einer aktiven Fahrzeugfederung für Elektrofahrzeuge

Xiaobo Liu-Henke, Robert Buchta, Michael Scheele

Ostfalia Hochschule

Fakultät Maschinenbau, Institut für Mechatronik, 38302 Wolfenbüttel

*X.Liu-Henke@ostfalia.de*

Dieser Beitrag fokussiert die Systemkonzeption einer aktiven Fahrzeugfederung für ein Elektro-Quad als Funktionsträger, welcher an der Ostfalia Hochschule aktuell mittels der mechatronischen Methodik entwickelt wird. Es wird der aktuelle Stand der Technik bezüglich Fahrzeugfederungen für konventionelle Fahrzeuge dargestellt, um dann mit Hilfe der mechatronischen Entwicklungsmethodik das Funktionsmodul aktive Federung für Elektrofahrzeuge zu konzipieren. Dabei wird mit dem aktiven Federungssystem der Zielkonflikt aus Fahrkomfort und Fahrsicherheit entschärft. Die Konzipierung erfolgt unter der Berücksichtigung der Anforderungen an ein Elektrofahrzeug modellbasiert. Die Auslegung der Informationsverarbeitung erfolgt hierarchisch, bestehend aus globaler Kompensationsregelung in Kombination mit einer Skyhook-Dämpfung. Entsprechend der mechatronischen Entwicklungsmethodik beinhaltet die Konzeption die Echtzeit-Realisierung mittels HiL-Prüfstands eines Viertelfahrzeugs. Abschließend wird das Potential der Energierückgewinnung bei der Fahrzeugfederung dargestellt.

## 1 Einleitung

Zur Erfüllung einer sicheren, komfortablen und nachhaltig umweltschonenden Mobilität steigen die Anforderungen in der Fahrzeugentwicklung. Nur mit weitreichenden Maßnahmen kann dem global steigenden Bedarf an individueller Mobilität und der zunehmenden Erschöpfung fossiler Ressourcen begegnet werden. Als Lösungsansatz wird die teilweise oder vollständige Elektrifizierung des Antriebsstrangs mit einer ergänzenden Speisung aus regenerativen Energien gesehen. Mit einem bei Elektrofahrzeugen vorhandenen elektrischen Energiespeicher eröffnet sich die Möglichkeit die Dissipationsenergie bei der Dämpfung der Aufbaubewegung zurückzugewinnen und zur Erhöhung der Reichweite zu nutzen. Das Potential zur Energierückgewinnung wurde bereits mehrfach untersucht.

An der Ostfalia Hochschule wird der Funktionsträger **Fredy** (Funktionsträger für regenerative Elektromobilität und Fahrdynamik) zur Integration von Elektromobilität und intelligenten Fahrwerksystemen entwickelt werden. Hierbei soll das System so konzipiert werden, dass das Elektrofahrzeug aktiv gefedert wird, darüber hinaus soll eine Energie-Rekuperation ermöglicht werden.

Das Feder-Dämpfer-System realisiert Relativbewegungen zwischen Fahrwerk und Fahrzeugaufbau und muss dabei Fahrsicherheit und -komfort garantieren.

Über die Feder und den Dämpfer sollen nicht nur die statischen und die dynamischen auf den Aufbau wirkenden Kräfte gestellt werden, sondern gleichzeitig soll eine Isolation hochfrequenter Schwingungen aus der Fahrbahnanregung gewährleistet werden. Im Allgemeinen ist bei der Systemauslegung einer passiven Fahrzeugfederung stets ein Kompromiss, mit einem Zielkonflikt zwischen Fahrsicherheit und Fahrkomfort, zu wählen. Dabei ist zwischen der Höhe der Radlastschwankungen und der Dämpfung der Aufbaubewegung abzuwägen [1]. Zwar soll das Fahrzeug dem Straßenprofil gut folgen, aber auch weitgehend von den Straßenunebenheiten isoliert werden. Passive Feder-Dämpfer-Systeme versteifen sich im hohen Frequenzbereich, da die Stellkraft sich auf eine Linearkombination von Relativweg und Relativgeschwindigkeit zwischen Aufbau und Rad beschränkt. Folglich verfügt das passive Feder-Dämpfer-System über einen sehr schlechten Fahrkomfort. Die Straßenunebenheit führt zu Schwingungen am Fahrzeugaufbau, die den Fahrkomfort des Passagiers und die Fahrsicherheit gefährden. Diese Eigenschaft kann mit Hilfe von aktiver Federungstechnik wesentlich verbessert werden [2].

Bei dem hier fokussierten Systemkonzept für eine aktive Federung für Elektrofahrzeuge werden zwei verschiedene Betriebsstrategien verfolgt. Einerseits soll der Fahrkomfort und die Fahrsicherheit bewähr-

ter aktiver Systeme in Kombination mit einer Energierückgewinnung erzielt werden. Diese erfolgt immer dann, wenn dem System Energie entzogen wird. Andererseits soll mit einer anderen Betriebsstrategie das Maximum an Energie zurückgewonnen werden bei einem gegenüber passiven Systemen überlegenen Systemverhalten. Durch eine veränderliche Dämpfung wird das aktive Stellglied stets generatorisch betrieben mit dem Ziel eines hohen Fahrkomforts bei hoher Fahrsicherheit.

## 2 Lösungsansätze und Stand der Technik

Der wesentliche Unterschied zwischen einem passiven und einem aktiven System kann an dem Vorhandensein einer externen Energiezufuhr festgemacht werden. Während ein passives System keine externe Energiezufuhr besitzt, ist beim aktiven System eine externe Energiezufuhr zur direkten Kompensation der Störungen erforderlich. Die Störgrößen können selbsttätig ohne Eingriff des Fahrers ausgeregelt werden. Hierbei werden die passiven Feder-Dämpfer-Elemente durch mechatronische Elemente - Aktoren, Sensoren und Regler - ersetzt.

Abbildung 1 stellt die typische Konfiguration der passiven und aktiven Federungssysteme von konventionellen Fahrzeugen dar [2].

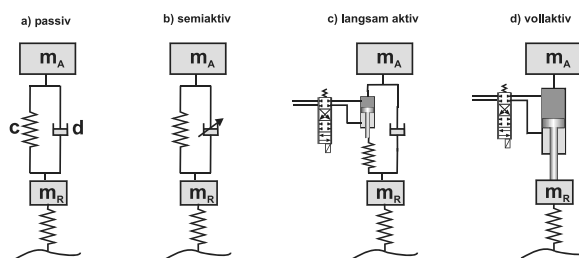


Abbildung 1. Konfigurationen von Federungssystemen

Als **passive Systeme** werden im Automobilbau klassische, konventionelle Fahrwerke, die keine von außen zugeführte Energie benötigen, bezeichnet. Sie weisen konstante Feder- und Dämpfungseigenschaften gemäß einer Kennlinie auf. Bei **semi-aktiven Systemen** handelt es sich um eine Variation der Charakteristik passiver Elemente. Die Stellglieder und damit die Eigenschaften der Federn und Dämpfer können innerhalb der charakteristischen Schwingungsdauer von Aufbau und Rad umgeschaltet werden. Die dafür nötigen kurzen Stellzeiten solcher

Systeme stoßen in dieser Hinsicht an ihre Grenzen. In **langsam-aktiven Systemen mit niedriger Bandbreite** sind die Aktoren in Serie mit passiven Federelementen verschaltet. Der Vorteil dieser Systeme ist ein geringer Energiebedarf, denn die vorhandene passive Feder kompensiert die Schwingung im hohen Frequenzbereich ohne zusätzliche Energiezufuhr. Die Aktoren müssen nur die niederfrequenten Schwingungen des Aufbaus bis max. 5 Hz ausregeln, so dass das System wesentlich kostengünstiger ausgeführt werden kann als ein vollaktives System. Bei **vollaktiven Systemen mit hoher Bandbreite** werden die passiven Elemente gänzlich durch Stellglieder ersetzt. Durch Aufwendung entsprechender Energie können die Kräfte zwischen Rad und Aufbau des Fahrzeugs stark variiert werden. Ein solches System ist jedoch aufgrund der sehr hohen Anforderung an die Aktordynamik und aufgrund des hohen Energiebedarfs kaum sinnvoll realisierbar.

## 3 Mechatronische Entwicklungsmethodik

Zur Entwicklung eines solchen Systems mit sehr hoher Komplexität muss die mechatronische Entwicklungsmethodik herangezogen werden. Zur Gewährleistung eines systematischen Entwurfs ist die Beherrschung der Systemkomplexität elementar. Nach [2] hat sich hierbei folgende mechatronische Methodik mittels Modularisierung und Hierarchisierung bewährt: „Ein komplexes System wird mittels „Top-down-Verfahren“ in Teilsysteme mit mechatronischen Komponenten aufgeteilt.

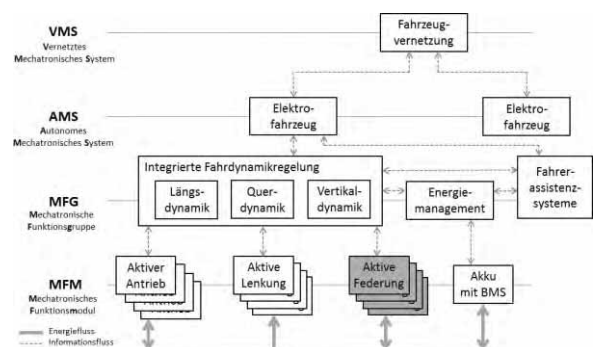


Abbildung 2. Hierarchie der Gesamtsysteme Elektrofahrzeuge nach der mechatronischen Strukturierung, Top-Down

Hierbei ergeben sich folgende vier Ebenen. Die unterste Strukturebene, **Mechatronisches Funktionsmodul (MFM)**, stellt das vitalste Element des Gesamtsystems dar und ist mit einer Funktionalität ge-

kapselt. Es verfügt über eindeutig definierte physikalische und informationstechnische Schnittstellen zu weiteren MFM. Auf dieser Ebene ist das Aktive-Federungssystem einzuordnen (Abbildung 2). Es besteht entsprechend der Charakteristik eines MFM aus einer Tragstruktur, Aktor-/Sensorgruppe und lokaler Informationsverarbeitung.

Die zweite Strukturebene, **Mechatronische Funktionsgruppe (MFG)**, besitzt keine eigene Aktorik und keine mechanische Tragstruktur. Eine MFG kann ihre Funktionen nur erfüllen, indem sie auf die unterlagerte Aktorik der MFM zurückgreift. Hier ist beispielhaft für ein Elektrofahrzeug die Fahrdynamikregelung einzuordnen, die sich im Informationsaustausch mit dem Elektroantrieb befindet.

Die Gesamtheit aus MFG und MFM stellt das **Autonome Mechatronische System (AMS)** dar, also das Elektrofahrzeug im physikalischen Sinn. Die oberste Ebene bildet durch die Kombination mehrerer AMS ein **Vernetztes Mechatronisches System (VMS)**. Die Aufgabe eines VMS liegt in der Koordination des Informationsflusses zwischen den AMS und in der Weitergabe von Entscheidungen, die alle AMS betreffen. Dies entspricht der Vernetzung von mehreren Elektrofahrzeugen wie bei der Car-to-Car- bzw. Car-to-Infrastructure-Communication.

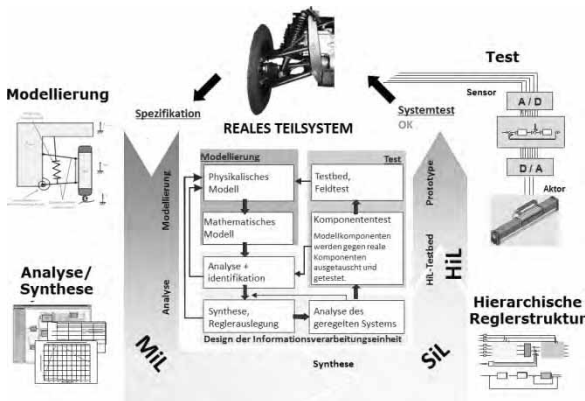


Abbildung 3. Auslegung des Teilsystems nach der mechatronischen Komposition, Bottom-Up

Der Vorteil der mechatronischen Strukturierungsmethode ist der hierarchische Aufbau der Systemfunktionen mit den definierten Schnittstellen auf der horizontalen und der Kommunikation mit der Umgebung auf der vertikalen Ebene. Man erhält somit in einer effizienten Art einen Überblick über die Systemkomplexität.

Nach der Strukturierung und Schnittstellendefinition von oben, erfolgt anschließend der sukzessive Aufbau der modellbasierten Teilsysteme in einem „Bottom-up-Verfahren“. Abbildung 3 stellt die sogenannte mechatronische Komposition des Teilsystems Aktive Federung im Entwicklungsprozess mittels Model-in-the-Loop (MiL), Software-in-the-Loop (SiL) und Hardware-in-the-Loop (HiL) dar.

## 4 Systemkonzept

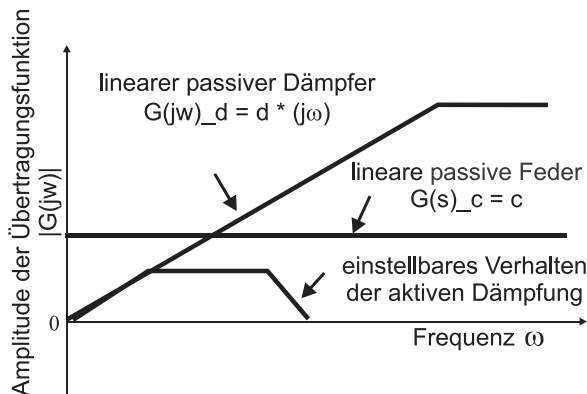
### 4.1 Anforderungen

Es gibt eine Fülle von konstruktiven Anforderungen an eine Fahrzeugfederung wie Abstützen der Aufbaumasse, hinreichender Federweg, eine beladungsunabhängige Charakteristik etc.. Die wesentlichen Anforderungen an eine moderne Fahrzeugfederung sind jedoch die Verbesserung von Fahrkomfort und Fahrsicherheit.

Als Maß für den Fahrkomfort dient die jeweilige frequenzabhängige Aufbaubeschleunigung. Die Beschleunigungskräfte werden von den Insassen als Störung empfunden. Je kleiner die Beschleunigung ist, desto besser der Fahrkomfort. Das kann durch schwingungstechnische Entkopplung des Aufbaus vom Fahrwerk mittels einer weichen Aufhängung in den betrachteten Bewegungsrichtungen erreicht werden.

Eine weitere wichtige Anforderung betrifft die Fahrsicherheit. Diese kann durch eine starke Dämpfung des Wagenkastens zur Stabilisierung aller Aufbaubewegungen erreicht werden. Allerdings übertragen Dämpfer die hochfrequenten Anteile der Fahrwerkschwingungen. Die Abstützung der Dämpfungskräfte erzeugt Beschleunigungskräfte im Wagenkasten. Hierdurch vermindert sich wiederum der Fahrkomfort.

Der hieraus entstehende Zielkonflikt zwischen Komfort und Sicherheit kann mit passiven Systemen nur unzureichend gelöst werden, denn die Abstimmung solcher Systeme ist mit den passiven Feder- und Dämpferelementen bis an die physikalische Grenze ausgeschöpft [3]. Hingegen kann durch aktive Federungstechnik diese Grenze verschoben und damit ein besseres bzw. völlig neues Systemverhalten erreicht werden, da die gewünschte Dämpfung- bzw. Federungscharakteristik im entsprechenden Frequenzbereich durch aktive Eingriffe erzielt werden kann (Abbildung 4). Somit können ein hoher Fahrkomfort und gleichzeitig eine hohe Fahrsicherheit durch diese aktive Stabilisierung aller Aufbaubewegungen gewährleistet werden.



**Abbildung 4.** Übertragungsverhalten der Feder- und Dämpfungssysteme

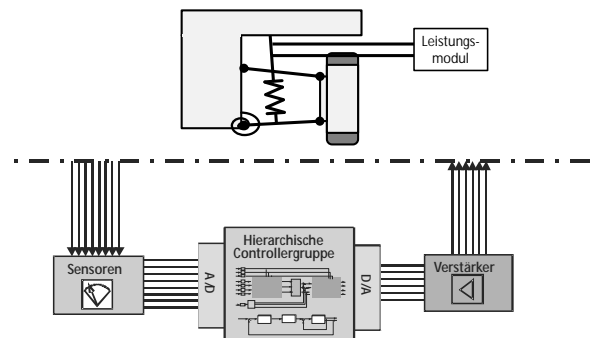
Für ein Elektrofahrzeug sind jedoch weitere Anforderungen zu berücksichtigen:

- Aufnahme der dynamischen Kräfte mittels Aktor und somit unter geringem Energieaufwand (motorischer Betrieb des aktiven Systems)
- Generatorischer Betrieb des Stellgliedes zur Energierückgewinnung der Dämpferdissipationsenergie
- Hohe Energieeffizienz der Komponenten auf Grund des begrenzten Energiereservoirs bei aktuellen Elektrofahrzeugen
- Gleichteilestrategie des aktiven Federungs-systems für alle vier Radlagerungen
- Bauraumbegrenzung auf vergleichbare konventionelle Feder-Dämpfer-Systeme

Auf Grund des angestrebten Ziels der Energierückgewinnung als Beitrag der Reichweitenverlängerung bieten sich somit rein elektrische oder elektromechanische Aktoren für eine aktive Federung eines Elektrofahrzeugs an. Zusätzlich werden für einen Forschungsträger auf Grund der geringen Masse nur geringe Stellkräfte benötigt. Luft-Dämpfer und hydraulische Systeme sind daher nicht zwingend als Lösungen erforderlich.

#### 4.2 Konzeption der aktiven Federung

Unter Verwendung der mechatronischen Methodik wird die aktive Fahrzeugfederung konzipiert. Dabei wird von Anfang an nicht nur die mechanische Tragstruktur betrachtet, sondern auch die dazugehörige Aktorik, Sensorik und Informationsverarbeitung als Teil des Gesamtsystems (Abbildung 5).



**Abbildung 5.** Prinzipieller Aufbau der aktiven Federung

Die Fahrzeugfederung besteht aus zwei Teilen:

- einer aktiven Federung
- einer Niveauregulierung durch Drehstabfeder.

##### 4.2.1 Aktive Federung

Die aktive Feder-Dämpfer-Einheit, die sowohl im generatorischen als auch motorischen Betrieb arbeiten kann, ist zum Stellen der dynamischen Kräfte zuständig.

Das Kernelement bildet die Federung und deren Fußpunktverstellung. Mit Hilfe von Aktoren wird eine zusätzliche Verschiebung am Fußpunkt der Feder erzeugt. Somit entsteht zwischen Aufbau und Fahrwerk eine zusätzliche Kraft, welche die Dynamik positiv beeinflussen kann. Folglich kann auf einen Großteil der passiven Dämpfung verzichtet und die Federung weich konzipiert werden.

Die weiche Federung sorgt dafür, dass die hochfrequenten Anteile der Schwingungen, die oberhalb ihrer Eckfrequenz liegen, nicht weiter an den Aufbau übertragen werden. Somit kann die Anregung im oberen Frequenzbereich allein durch die Federung passiv isoliert werden. Die niederfrequenten Schwingungen des Aufbaus können durch die aktive Fußpunktverstellung der Federung bedämpft werden, damit die gewünschte aktive Dämpfung im unteren Frequenzbereich (ca. 4-5 Hz) realisiert werden kann.

Die notwendige Dämpfung zur Kompensation der Radlastschwankung und damit zur Gewährleistung einer hohen Fahrsicherheit wird durch die geeigneten Regelalgorithmen sichergestellt.

Die für die Regelung der Fußpunktverstellung der Luftfeder benötigten Informationen werden durch die eingesetzten Sensoren bereitgestellt und in einer hierarchisch aufgebauten Mehrgrößenregelung verarbeitet. Die zur Realisierung erforderliche Verarbeitung der komplexen Algorithmen kann durch Einsatz von Echtzeithardware erfolgen. Die Stellsignale werden über einen Leistungsverstärker an das Stellsystem weitergegeben.

#### 4.2.2 Niveauregulierung

Die Drehstabfeder nimmt die statischen Radlasten auf. Mit dem geringen Leergewicht eines Elektrofahrzeugs ist das Verhältnis von Beladung zu Leergewicht äußerst ungünstig. Mit einer Niveauregulierung kann dieser Umstand teilweise entschärft werden. Dazu ist das am Fahrzeugaufbau befestigte Drehstabende mit einem Aktuator zum Klemmen des Drehstabs verbunden. Damit wird das gewünschte Fahrzeugniveau von der aktiven Federdämpfereinheit gestellt und dann für einen stationären Zustand ohne weiteren Energiebedarf durch den Drehstabaktor geklemmt. Dieser benötigt nur zum Lösen und Klemmen Energie und ist äußerst klein dimensioniert, da die Energie zum Stellen des Fahrzeugniveaus von der aktiven Federdämpfereinheit aufgebracht wird.

#### 4.3 Modellbildung und Analyse

Entsprechend der vorgestellten mechatronischen Entwicklungsmethodik erfolgt zu Beginn die Modellierung des MFM aktive Federung. Mit einer ganzheitlichen Betrachtung ist für die Synthese neben der Dynamik des mechanischen Systems auch das Verhalten der Aktorik, Sensorik und Informationsverarbeitung in der Gesamtsystemdynamik zu berücksichtigen.

Für die lokale Regelung des MFM genügt es dieses Modul getrennt vom Gesamtfahrzeug als Viertelfahrzeug zu betrachten, da die gegenseitigen Wechselwirkungen aus der Nick- und Wankbewegung des Fahrzeugaufbaus in der überlagerten Globalregelung berücksichtigt werden. Das physikalische Modell des mechanischen Systems einer Radeinheit als „Zwei Massen-Schwinger“ zeigt Abbildung 6.

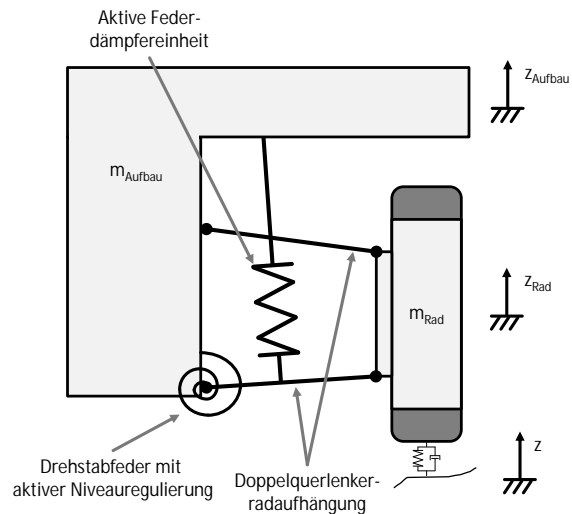


Abbildung 6. Physikalisches Modell der aktiven Federung

Die für die Analyse verwendeten Fahrzeugdaten entstammen einem Funktionsdemonstrators, der für eine prototypische Umsetzung dient. Zur Erzielung einer hohen Reichweite und guten Fahrleistungen hat dieses batterieelektrische Fahrzeug nach den geplanten Modifikationen ein Fahrzeuggesamtgewicht von 650 kg (Leergewicht mit zwei Personen). Die ungefederten Radmassen betragen 20 kg je Rad. Die Gesamtfedersteifigkeit der Aufbaufederung liegt bei 13.400 N/m je Rad. Das Fahrzeug verfügt über eine elektrische Antriebsleistung von 10 kW. Seine Höchstgeschwindigkeit ist auf 60 km/h limitiert.

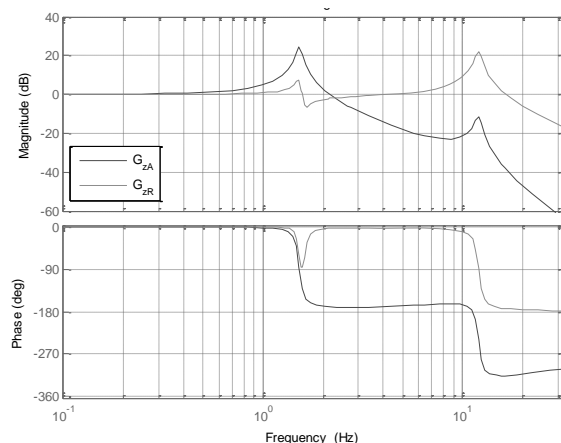


Abbildung 7. Übertragungsverhalten  $G_{zA}=z_A/z$  und  $G_{zR}=z_R/z$  für das ungedämpfte System

Abbildung 7 stellt die Frequenzkennlinien zur Analyse des Übertragungsverhaltens dar. Dieses umfasst die Übertragungsfunktionen für das ungedämpfte System

von der Straßenanregung  $z$  zur Radauslenkung  $z_R$  bzw. Aufbauauslenkung  $z_A$ .

Äußerst auffällig sind die Überhöhungen im Amplitudengang von über 20 dB. Diese lassen auf einen Dämpfungsgrad von  $D < 0,1$  schließen, der bei dem ungedämpften System aus der Systemreibung resultiert.

Die charakteristische Eigenfrequenz von etwa 2 Hz kann dem Fahrzeugaufbau zugeordnet werden, die weitere Eigenfrequenz von fast 11 Hz ist der Radmasse zuzuordnen. Die ungewöhnlich niedrige Eigenfrequenz des Rades ist auf das sehr geringe Fahrzeuggewicht mit entsprechend niedrigen ungefederten Massen eines batterieelektrischen Fahrzeugs zurückzuführen.

#### 4.4 Informationsverarbeitung

Nachdem das dynamische Verhalten des Systems untersucht wurde, müssen Regelstrukturen ausgelegt werden, damit die mechatronischen Funktionen des Systems das gewünschte Systemverhalten erzielen.

Um den Aufwand des Reglerentwurfs für komplexe mechatronische Systeme zu reduzieren und die Entwurfsprozesse besser zu beherrschen, soll die Informationsverarbeitung in besonderer Form strukturiert ausgelegt werden. Die Verallgemeinerung des Kaskadenprinzips [2] wird für die Auslegung der Mehrgrößenregelung angewendet.

Als Grundlage dient die hierarchische Mechatronikstruktur MFM / MFG / AMS / VMS (Kap. 3). Durch die hierarchische Anordnung des betrachteten Gesamtsystems kann eine der entsprechenden hierarchisch angeordneten Reglerstrukturen konzipiert werden.

Da jede Strukturebene ein Teilsystem darstellt, kann für jede Ebene eine jeweils geeignete Controllereinheit ausgelegt werden. Dabei müssen dem Controller ebenenbezogen geeignete Signale zugänglich gemacht werden. Das kann durch Messung oder Beobachtung geschehen. Der jeweilige Ebenencontroller kann neben seiner Strukturebene nur das ihm unterlagerte Teilsystem inklusive der unterlagerten Controllermodelle „sehen“. Die Informationsverarbeitung wird damit konsequent in den Regelprozess eingebunden. Somit entstehen Kaskaden von Regelkreisen und die Auslegung der gesamten Mehrgrößenregelung kann in eine Auslegung der jeweiligen „Teilcontroller“ ebenenweise aufgeteilt werden.

Der Mehrgrößenregler soll hierarchisch in globale, zentrale und lokale, dezentral arbeitende Controllergruppen gegliedert werden.

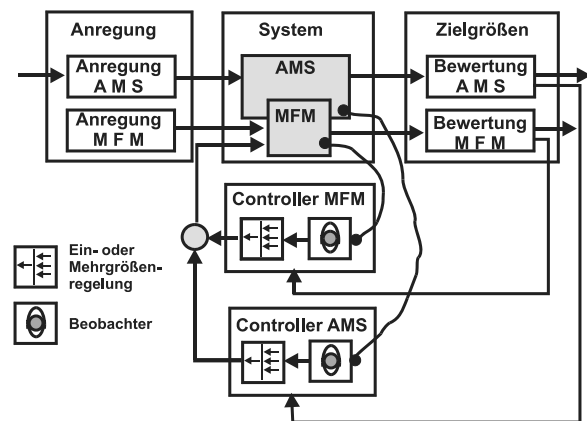


Abbildung 8. Hierarchisch angeordnete Reglerstrukturen

Der übergeordnete Globalregler AMS wird mit einer Kompensationsregelung in Kombination mit einer Skyhook- Dämpfung vorgesehen.

Als lokale Regelungen dienen die Kaskadenregelungen, welche die Dynamik des einzelnen Aktors hinreichend schnell ausregeln können.

Abbildung 9 stellt die Aufbaubewegung des aktiven und des passiven Systems in Vertikalrichtung dar. Das Simulationsergebnis zeigt, dass das Schwingungsverhalten des aktiven Systems eine wesentlich bessere Dämpfung als das des passiven Systems aufweist.

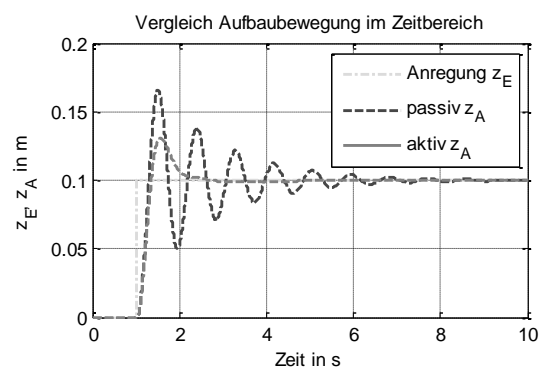


Abbildung 9. Gegenüberstellung passives und aktives System

Die Flächen unter dem Amplitudenspektrum sind ein Maß für die auf den Fahrgast wirkenden Beschleunigungen. Je größer die Fläche ist, desto schlechteren

Fahrkomfort weist ein System auf. Abbildung 10 stellt die Amplitudenspektren für die Aufbaubeschleunigung des aktiven und des passiven Systems gegenüber. Das aktive Fahrwerk schafft hier einen verbesserten Fahrkomfort.

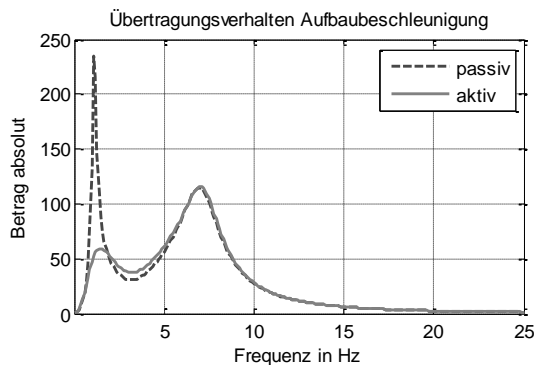


Abbildung 10. Amplitudenspektren passives/aktives System der Aufbaubeschleunigung

## 5 Echtzeit-Realisierung

Um einen Übergang vom theoretischen Rechnermodell zum realen System vorzunehmen, ist ein HiL-Prüfstand vorgesehen.

Abbildung 11 illustriert das Prüfstandkonzept. Es besteht aus drei Teilsystemen: Prüfstand-, Software- und Echtzeithardwaremodul. Die Regelgrößen werden gemessen und die Messsignale werden über A/D-Wandler einem Digitalen-Signal-Prozessor zugeführt. Dort werden die Regelalgorithmen gerechnet und die Signale über D/A-Wandlerkarten an die Ansteuerungseinheiten im realen Subsystem ausgegeben.

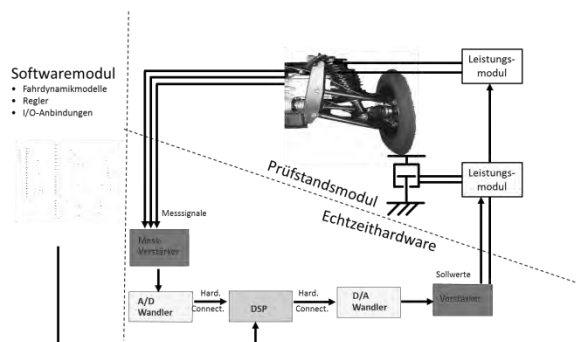


Abbildung 11. Prinzipieller Aufbau des HiL-Prüfstands

Auf diesem Labor-Prüfstand können die ausgelegten hierarchischen Regelstrukturen und die komplette Konfiguration des Systems unter Echtzeitbedingungen

gen messtechnisch analysiert und validiert bzw. getestet werden.

Weiterhin können die System- und Parameteridentifikation unter realitätsnahen Bedingungen sowie die abschließende Implementierung der MFG Aktive-Federung in den Forschungsträger durchgeführt werden.

## 6 Potentialabschätzung der Rekuperation der Stoßdämpferenergie

Aktive Systeme für konventionelle Fahrzeuge sind seit längerem in der Erprobung und bereits zum Teil auf dem Markt verfügbar. Für Elektrofahrzeuge bietet dieses System neben der Überwindung des Zielkonflikts aus Fahrdynamik und Fahrsicherheit ein Potenzial zur Energierekuperation und damit zur Erhöhung der Reichweite dieser. In Abhängigkeit des Fahrbahnprofils (nach ISO 8608) und der Fahrgeschwindigkeit listet [4] die Dämpferleistung bei konventionellen Fahrzeugen auf (Abbildung 12). Im Mittel werden demnach auf deutschen Autobahnen 67 W je konventionellem Fahrzeug im Dämpfer dissipiert.

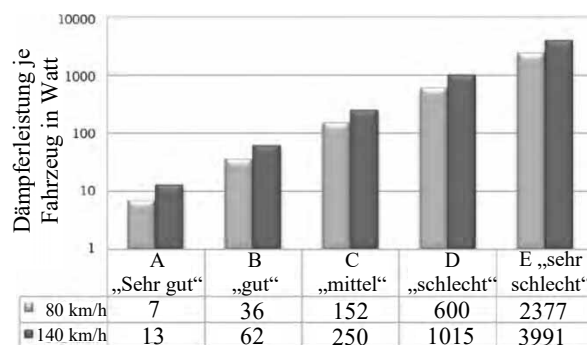


Abbildung 12. Dämpferleistung je konventionelles Fahrzeug in Abhängigkeit der Fahrgeschwindigkeit

Dies bestätigen auch Erfahrungen an der Hochschule Ostfalia an einem Forschungsfahrzeug, M-Mobile [5] und [6], zur Erprobung von Elektrofahrzeugen mit dezentralen Elektroantrieben, dass in der Vertikaldynamik ein Potential zur Verbrauchssenkung besteht.

Das Potential zur Energierekuperation der Dämpferdissipationsenergie bei Elektrofahrzeugen ist somit vorhanden und im praktisch nutzbaren Bereich, wobei sich damit ein neuer Zielkonflikt aus dem Energiebedarf für das aktive Federungssystem und der Energierückgewinnung in Form der Reichweitenverlängerung eröffnet.

## 7 Resümee

Der Beitrag stellt die Systemkonzeption eines aktiven Federungssystems für Elektrofahrzeuge am Beispiel eines Funktionsträgers in Form eines Elektro-Quads, welches an der Ostfalia Hochschule mittels der mechatronischen Methodik entwickelt wird, dar. Mit der Darstellung des aktuellen Stands der Technik bezüglich Federungssystemen von konventionellen Fahrzeugen erfolgt eine Eingruppierung des aktiven Systems. Zur weiteren Detaillierung der Konzeption wird die mechatronische Entwicklungsmethodik vorgestellt, welche einen Überblick über die Systemkomplexität schafft. Diese Methodik ist für die Konzeption essentiell, da sie die Grundlage für alle weiteren Auslegungen (z. B. Informationsverarbeitung/ Reglerauslegung) legt. Mit der Klärung der Anforderungen des aktiven Federungssystems für Elektrofahrzeuge erfolgt die Konzeption. Als Übergang vom theoretischen Rechnermodell zum realen System, erfüllt ein HiL-Prüfstand die Aufgaben der Erprobung und Optimierung unter realitätsnahen Bedingungen. Das theoretisch vorhandene Potenzial zur Energierecuperation der Stoßdämpferenergie eröffnet weitreichende Möglichkeiten in der Praxis. Zugleich entschärft ein aktives Federungssystem den Zielkonflikt aus Fahrkomfort und -sicherheit, eröffnet aber bezüglich der Energierecuperation einen neuen Konflikt zwischen Energiebedarf für das System und der Energierückgewinnung zur Reichweitenverlängerung.

## 8 Referenzen

- [1] M. Mitschke. *Dynamik der Kraftfahrzeuge*. Springer Verlag, Deutschland, 2004
- [2] X. Liu-Henke. *Mechatronische Entwicklung der aktiven Feder-/Neigetechnik für das Schienenfahrzeug RailCab*. Dissertation, Universität Paderborn, Deutschland, 2004.
- [3] W. Darenberg, H. Gall, B. Acker. *Chancen und Probleme aktiver Kraftfahrzeugfederung*, VDI-Berichte, Nr. 515, VDI-Verlag, 1984, S. 187-194.
- [4] M. Willems. *Potenzialabschätzung zur Rekuperation der Stoßdämpferenergie*. ATZ 09/2012, 114. Jahrgang
- [5] X. Liu-Henke, R. Buchta, et al. Simulation eines mechatronischen Lenkungsmoduls für ein Elektrofahrzeug mit dezentralen Direktantrieben. Workshop ASIM/GI-Fachgruppen STS und GMMS, Krefeld, 24./25.02.2011
- [6] X. Liu-Henke, R. Buchta. Regelung der Quer- und Gierbewegung eines Elektrofahrzeugs mittels radnahen Direktantrieben. Workshop ASIM/GI-Fachgruppen STS und GMMS, Wolfenbüttel, 23./24.02.2012



# Modeling Worm Gear Mechanism in Three Architectures, Parameter Identification and Comparison

Xin Zhao<sup>1</sup>, Sergey Petkun<sup>2</sup>

<sup>1,2</sup>Brose Fahrzeugteile GmbH & Co. Kommanditgesellschaft, Hallstadt

*Xin.zhao.temp2@Brose.com*

In this paper we present three architectures of modeling worm gear mechanism in language of VHDL-AMS. The worm gear mechanism is very often used to save the last mechanical state without supplying additional energy. The features of worm gear, such as high reduction rate and self-locking function, are studied and modeled in three architectures. The first is based on behavioral modeling of functionality, which takes the gear ratio and power transmission efficiency as the most important parameters. The second and the third ones realize each 2- and 3-dimensional analysis of force interaction within worm gear mechanism by supplying basic geometrical information and additional dynamical parameters, such as pitch radius, lead angle, friction coefficient and so on. The simulation results with the three architectures and their comparisons between each other show that models of worm gear in this paper reached the expectation to model the movement transferring function and the self-locking function. The amounts of values and quantities calculated in the three architectures are in an ascending order and it meets the basic interests to study the outer behavior and also the inner force distributions of worm gear mechanism.

## 1 Introduction

Worm gear mechanism is widely used in lots of applications. In electrical power window regulator systems of modern vehicles, worm gear mechanism is integrated to Direct-Current Permanent-Magnet motor (DCPM motor) and plays an important role to stabilize the movement of door glass and maintain glass position after electrical power is switched off. The utilization of worm gear mechanism brings advantages, such as good acoustic performance and low cost, while disadvantage is also obvious, that is, low power transmission efficiency [1].

In system simulations [2] of window regulator systems, models of worm gear mechanism are essential and have significant influence on simulation results. Therefore, model of worm gear mechanism is under strong request. However, models are not found until now in existing VHDL-AMS library, for example fundamental VDA library. In this paper, we present worm gear models in three architectures in language of VHDL-AMS. The realizations have the capability to fulfill the basic functionality requirements defined in the next section. The reason for functionality requirements is to constrain the set of behaviors and features to be implemented in models. As next, three architectures are represented and parameter identification is shortly explained. At the end, comparisons are conducted between all three architectures.

## 2 Requirements on worm gear models

Worm gear mechanism, called in another way worm gear drive, is classical part written in academic textbooks[3,4] of mechanical engineering and is also further developed and studied[5,6] in a deeper degree. Mathematical model is studied in paper [7]. Simulation models are found in forms, for example MatLab [8] or in techniques, like FEM [9-12]. However, in the modeling language of VHDL-AMS, there is still a blank.

To model worm gear mechanism, we defined a basic specification, which places the requirements of the model performance and also restricts the features to be implemented and time and effort costed for it. The specification covers two working modes of worm gear mechanism. Under the two working modes, worm gear mechanism shows two different features.

### 2.1 Driving mode

Under driving mode, worm drives gear to rotate. It can be imagined as a gear box with reduction ratio of  $r$  and power transmission efficiency of  $\mu$ . The relationship are described in the following equations,

$$Rot\_Vel\_Worm = Rot\_Vel\_Gear * r \quad (1)$$

$$Trq\_Gear = Trq\_Worm * r * \mu \quad (2)$$

Reduction rate,  $r$ , can be either specified as a parameter or calculated from geometrical information of worm gear design. The efficiency,  $\mu$ , representing loss resulting from friction, can be a parameter as input to models or a result from internal calculation.

## 2.2 Self-locking mode

When torque from gear tries to drive worm, worm gear mechanism goes into self-locking mode. In this mode, worm has no movement if self-locking function is existing and functional. The realization of self-locking function depends on many factors, for example, lead angle, friction at contacting face between worm and gear, temperature, lubrications and so on. Through industrial applications of vehicle window regulator systems with worm gear mechanism, an empirical guideline is summarized, that is, self-locking function of worm gear mechanism exists under the condition of room temperature, lubrication and working range, if power transmission efficiency is smaller than 50 percent; if efficiency is higher than 50 percent, and self-locking function is lost.

As requirement, models of worm gear mechanism in VHDL-AMS should first fulfil the function to transfer movements under driving mode. And models should also realize self-locking function with consideration of power transmission efficiency as threshold. It means that, as guideline, self-locking function should be functional when efficiency is below 50 percent, while no self-locking function when above 50 percent. The models to be introduced in the next section comply with the requirements above.

## 3 Implementations

Three architectures of modeling worm gear mechanism in VHDL-AMS are demonstrated in this section. The first one is based on method of behavioural modeling and it is named in this paper as behavioural architecture. The second and the third analyze each 2D and 3D force interactions and friction at the contacting face between worm and gear and also frictions from bearings. The two architectures dive more into the physical principle of self-locking functions, so we give them the names as 2D physical architecture and 3D physical architecture.

### 3.1 Behavioral Architecture

The behavioural architecture is inspired by worm gear model [13] from Synopsys© Saber® library. The

model is encrypted and is written in language of MAST oriented to Saber®. The MAST model has the problem, not satisfying self-locking function defined in section 2. The phenomenon is that self-locking function is always on, no matter whether power efficiency is below or above 50 percent.

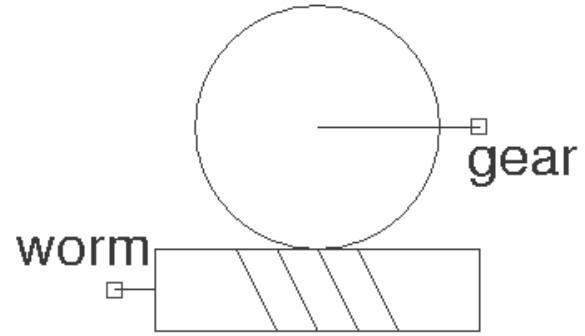


Figure 1: Model of worm gear mechanism

The model realized in this paper has two pins, namely worm and gear, seen in figure 1. Figure 2 and 3 show the internal interpretation of behavioural architecture under driving mode and self-locking mode.

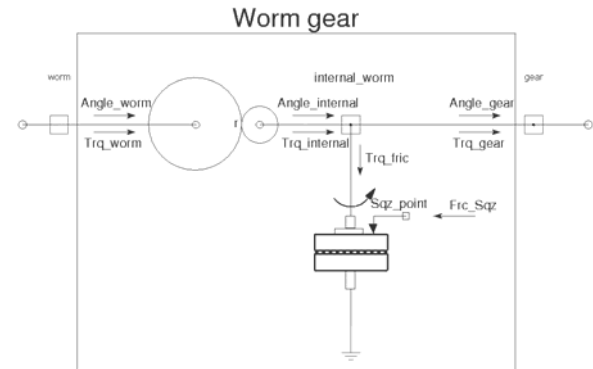


Figure 2: Model interpretation of driving mode

The internal construction of worm gear model consists of an additional pin, internal worm, and two components. One component connecting worm and internal worm is a gear box, which has ratio  $r$  and has no power loss. The gear box ratio inherits worm gear rate. Its function is described by the equations,

$$Angle\_worm = Angle\_internal * r \quad (3)$$

$$Trq\_internal = Trq\_worm * r \quad (4)$$

To bring power loss in model, rotational friction model connects itself between internal worm and ground. It shares movement of internal worm and causes loss indicated as  $Trq\_fric$ , the amplitude of

which is depending on the squeezing force applied at the squeezing point indicated as *Sqz\_point*. The pin of gear has the same angular movement as pin of internal worm and its torque is a result of internal torque and friction torque.

$$Angle\_gear = Angle\_internal \quad (5)$$

$$Trq\_gear = Trq\_internal - Trq\_fric \quad (6)$$

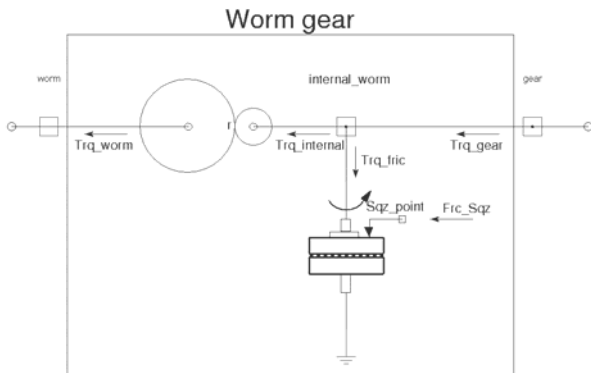
By parameterizing the rotational friction, friction torque could be numerically the same as squeezing force. Under driving mode, squeezing force is configured to be proportional to internal torque in the way to produce designated loss.

$$Frc\_Sqz = Trq\_internal * (1 - \mu) \quad (7)$$

In this way, the torque at gear is

$$Trq\_gear = Trq\_internal * \mu \quad (8)$$

The model behaviours under driving mode are governed by the equations above and these equations comply with the requirement of driving mode, specified by equation 1 and 2 in section 2.1.



**Figure 3:** Model interpretation of self locking

Under self-locking mode, torque is applied at gear but no movement can be produced, if self-locking function is functional. In this case, the angles at gear, internal worm and worm stay the last value. The relation of internal torque changes into

$$Trq\_gear = Trq\_internal + Trq\_fric \quad (9)$$

The self-locking function is realized by increasing the friction torque produced by friction mode until friction torque absorbs all torque from gear so that internal torque is zero. To achieve it, amplitude of squeezing force is configured as

$$Frc\_Sqz = Trq\_gear \quad (10)$$

Therefore, squeezing force takes values depending on working mode. To determine working mode, powers at worm and gear are examined. According to the properties of nodal analysis of simulation and the practice of Synopsys© SaberHDL® simulator, power of worm is negative and power of gear is positive under driving mode, while power of gear is negative when torque is applied at gear, which triggers self-locking function.

```

1 If power_gear > 0.0 then
2   Mode = driving;
3 Else
4   If efficiency < 0.5 then
5     Mode = self_locking;
6   Else
7     Mode = Reverse_driving;
8   End if;
9 End if;
```

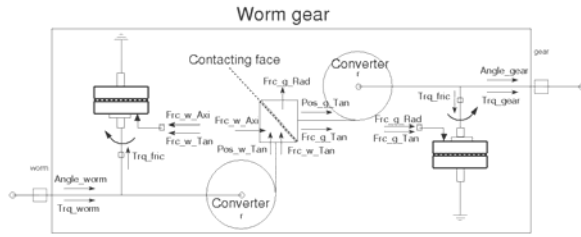
**Code 1:** Working mode determination

Additional to the two working modes, a third one is added, called reversed driving mode, in code 1. Under reversed driving mode, the efficiency, specified as a parameter, has no more practical meaning in calculation, as the power transmission efficiency from gear to worm takes another value, which can not be calculated with limited input parameters.

Behavioural architecture of worm gear model is parameterized mainly by two constants. One is worm gear rate, which can be found through number of teeth at worm and gear. The other parameter is power efficiency, which specifies how much power can be transferred from worm into gear. The identification of efficiency can be theoretically obtained through math equation. Another method is to measure torque of worm and rotational velocity of gear, under given torque at gear and given rotational at worm. Empirically, the efficiency should be a value less than 45 percent under room temperature, so as to insure self-locking functionality under temperature range from -30 to 80.

### 3.2 2D Physical Architecture

The 2D architecture of modeling worm gear mechanism has the same connecting pins as shown in figure 1. The internal interpretation of architecture has components of contacting face, movement converters and rotational frictions.



**Figure 4:** Model interpretation of 2D Architecture

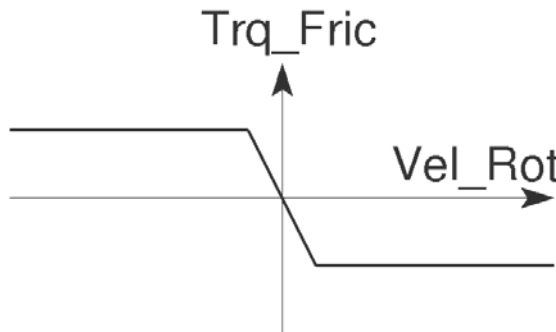
The core of 2D architecture is the component of contacting face, which describes how translational movement is transferred between two sides. One side can move only in vertical direction, the other can move only in horizontal direction and they contact each other with a slope. Translational movements are converted into rotational moving, representing rotational movements of worm and gear. Besides, friction components are attached to worm and gear and the amplitudes of friction torques are dependent on squeezing forces, which are calculated from contacting face.

The function of converter is straightforward.

$$Ang\_Rad * Radius = Pos\_M \quad (11)$$

$$Trq\_NM = Frc\_N * Radius \quad (12)$$

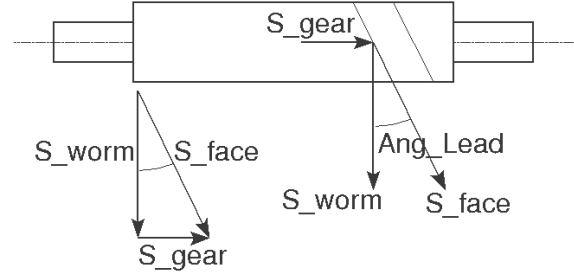
Models of friction in this architecture describe only coulomb friction, without considering viscous part of friction. Amplitude of coulomb friction is determined by friction coefficient, amplitude of squeezing force. From the side of worm, squeezing force is resulting from forces applied to worm in its tangent and its axial directions, while from gear side it is from forces in tangent and radial direction.



**Figure 5:** Friction interpretation

The axial directions of worm and gear are orthotropic. So it is the same with the two translational movements at contacting face, after being converted. Ki-

netic relation at contacting face is shown in figure 6, with  $S\_worm$  indicating translational displacement of worm,  $S\_gear$  for gear and  $S\_face$  for the relevant displacement of contacting face. With lead angle as parameter for worm gear, the relation is as



**Figure 6:** Kinetic relation

$$S\_gear = S\_worm * Tan(Lead) \quad (13)$$

So, the velocity is as

$$Vel\_gear = Vel\_worm * Tan(Lead) \quad (14)$$

As to the kinematic relation,  $frc\_w\_tan$  in figure 7 is the force converted from rotational torque of worm, while  $frc\_g\_tan$  is converted from rotational torque of gear.  $frc\_w\_Axi$  is force from axial direction of worm for supporting, while  $frc\_g\_rad$  is from radial direction of gear.  $frc\_normal$  is normal force applied from one moving part against the other and is vertical to contacting face. Under the condition of no friction, force equilibrium can be built for both worm and gear, while normal forces for them have the same value. However, if friction at contacting face is considered, then the component of friction force will be added into force equilibrium and this friction force is of type of coulomb friction force, depending on normal force and friction coefficient. As in the bottom of figure 7, final amplitudes of forces in tangent, axial and radial directions are resultant from forces in these direction under no friction condition and projected friction forces.

From the analysis above, under the condition of no friction, the following relation can be built,

$$Frc\_w\_Tan = Frc\_g\_Rad \quad (15)$$

$$Frc\_w\_Axi = Frc\_g\_Tan \quad (16)$$

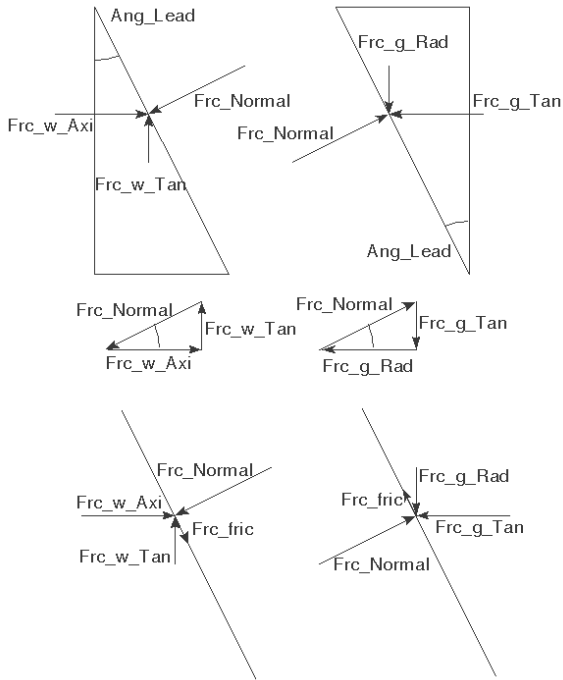


Figure 7: Kinematic relation

With friction added, in driving mode, torque from worm is transferred into gear, with loss from friction of contacting face and bearings. In self-locking mode,  $frc\_g\_Tan$  causes  $frc\_g\_Rad$  to increase. But projection of  $frc\_fric$  on direction of  $frc\_w\_Tan$  could be sufficient to compensate  $frc\_g\_Rad$ . In another case, if  $frc\_fric$  is not sufficient, friction torque between worm and axial bearing is still enough to hold rotational moving tendency of worm. In this manner, self-locking function is realized in 2D architecture.

Parameters	Identifications
Radius worm	Drawing
Radius gear	Drawing
Radius worm bearing	Drawing
Radius gear bearing	Drawing
Lead angle	Drawing
Friction coefficient of worm gear	Handbook; Empirical
Friction coefficient of bearing	Handbook; Empirical

Table 1: Parameters and identifications

The main parameters of 2D architecture are listed in table 1. The parameters, related to the dimensional information, are obtained from drawing. Transmission ratio in this case can be calculated as,

$$ratio = Radius\_w / Radius\_g * Tan(Lead) \quad (17)$$

However, to determine a precise value of friction coefficient beside geometrical parameters, it is not possible. Classical mechanical handbook can give a range of friction coefficient with care of contacting materials, lubrication and temperature. Experiences from industrial applications are also element to help find out appropriate value.

### 3.3 3D Physical Architecture

The 3D interpretation of worm gear is similar to the interpretation of 2D architecture. However, the differences lay on the number of components and their complexity. For 3D interpretation, squeezing forces applying on rotational friction models are from three orthotropic forces in tangent, axial and radial directions of both worm and gear. Converters are the same as in 2D interpretation. However, model of contacting face deals with the force calculation in 3D, as shown in figure 8. Under the condition of no friction considered at contacting face, these forces in orthotropic directions have each relation with normal force, taking parameters of lead angle and press angle.

$$Frc\_w\_Tan = Frc\_w\_Norm * \cos(Press) * \sin(Lead) \quad (18)$$

$$Frc\_w\_Axi = Frc\_w\_Norm * \cos(Press) * \cos(Lead) \quad (19)$$

$$Frc\_w\_Rad = Frc\_w\_Norm * \sin(Press) \quad (20)$$

And, also there are

$$Frc\_w\_Tan = Frc\_g\_Axi \quad (21)$$

$$Frc\_w\_Axi = Frc\_g\_Tan \quad (22)$$

$$Frc\_w\_Rad = Frc\_g\_Rad \quad (23)$$

If friction is considered, friction resulted from normal force will be projected into the orthotropic directions and the procedure to build up equilibrium is similar as in 2D interpretation. All the resultant forces from contacting face are later applied to friction models as squeezing forces, because bearings support both worm and gear in every orthotropic directions. As to the kinetic relation, it remains the same as 2D interpretation and the parameter of press angle has no influence of the transmission ratio.

Compared with 2D architecture, parameter set for 3D architecture requires one additional, which is press angle. Press angle of worm gear can be obtained from drawings. Realizations of driving function and self-locking function are based on the same principle as 2D architecture.

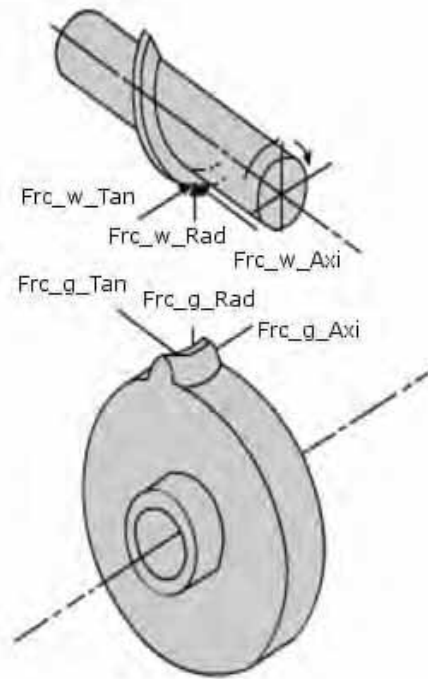


Figure 8: Forces at contacting face

## 4 Comparisons

In this section, we verify the performance of the three architectures of worm gear models, to testify whether they could meet the requirement defined in section 2. A test bench in SaberHDL® sketch is designed, to test functionality of worm gear models under driving mode and self-locking mode.

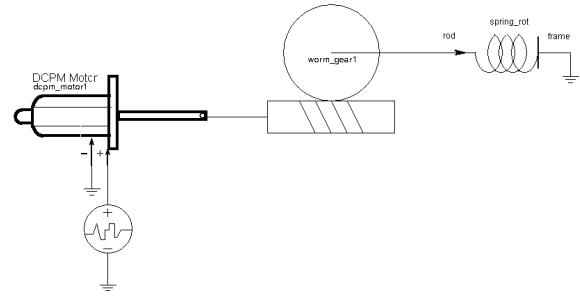


Figure 9: Simulation sketch and components

As in figure 9, worm is connected to a DCPM motor, while gear is simply connected to a rotational spring. Spring simulates a linier load. The characteristic line of DCPM motor is in figure 10. And the motor is driven by a PWL voltage source, the profile is in figure 11.

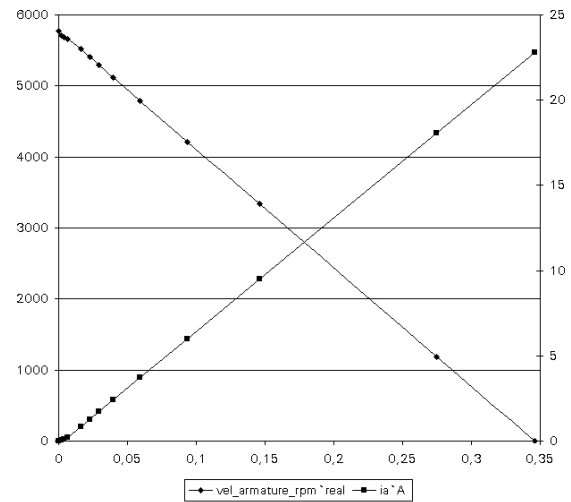
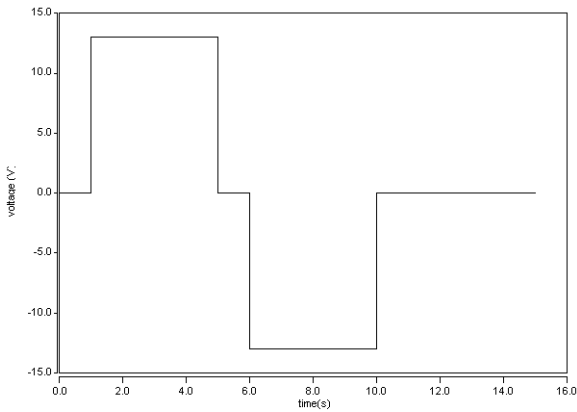


Figure 10: Characteristic lines of DCPM motor

Under PWL wave, motor will first rotate in positive direction for 4 seconds, then stay still for 1 second and then rotate in reversed direction for another 4 seconds. Therefore, driving mode and self-locking mode will be tested in both rotational directions.

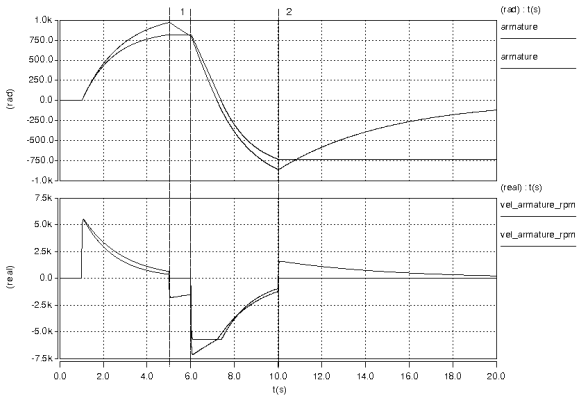
### 4.1 Functionality

The functionality in the sense of verifying worm gear models means that power transmission from worm to gear should not have any problem under driving mode. Self-locking function should be able to be enabled and disabled, corresponding reaction should be revealed.



**Figure 11:** PWL Voltage Profile

The reduction ratio of worm gear models is parameterized to be 1:73 and power transmission efficiency is set to be 46 percent. They can be directly set into behavioural architecture. For 2D and 3D architectures, reduction ratio is realized by geometrical parameters and power efficiency is set through geometrical parameters and friction coefficients.

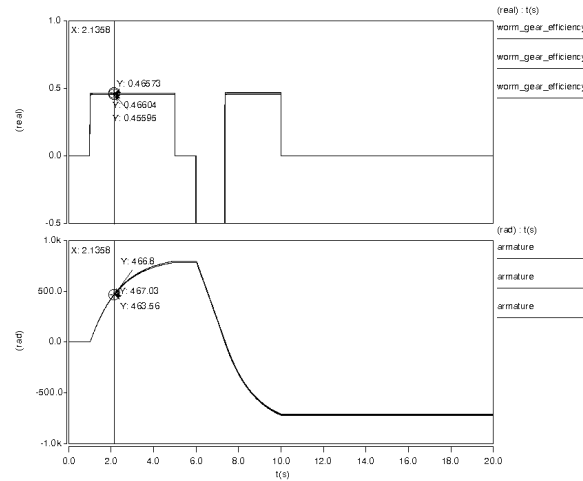


**Figure 12:** Angle and angular velocity of armature under self-locking function enabled and disabled

The first simulation is conducted to check the behaviors of worm gear models under conditions of self-locking function enabled and disabled. For one time, the power transmission efficient is set to be 46 percent and for the second time it is set to be 60 percent.

Figure 12 shows the simulation result of behavioural architecture of worm gear model. Under driving mode, the worm gear with higher power transmission efficient rotated faster than the one with lower efficient. However, self-locking function is dismissed under 60 percent of efficiency and armature of DCPM motor reversed when driving voltage of motor is zero, marked with region 1. For 2D and 3D architectures, the same behaviors can be achieved by

changing friction coefficients in model parameter sets. Changing lead angle can also result into a higher efficiency of power transmission, but it would cause reduction ratio varied.



**Figure 13:** Angle of armature and power transmission efficiency of worm gear models from three architectures

In the second simulation, we compared the behaviors of worm gear models within different architectures. The parameters of them were set so that they would behave in a similar way, which is, reduction rate is 1:73 and efficiency is 46 percent. The simulation result in figure 13 shows that the difference of efficiency between simulation and designated value is in the range of 1.2% to 1.4%.

## 4.2 Simulation efficiency

Secondly, simulation efficiency with three architectures of worm gear models was compared. The setup of simulator is presented in table 2.

Options	Value
End time / s	20
Time step / s	1m
Truncation Error	0.005m
Sample Point Density	1000
Target Iteration	300

**Table 2:** Simulator configuration

The time costed for simulating each worm gear model in sketch as shown in figure 9 differs from each other.

Architecture	Simulation Time
Behavioral	1.48 s
2D	0.625 s
3D	0.312 s

**Table 3:** Simulation time costed for three architectures

It is obvious to see that behavioral model costed relevantly longer time, comparing with the other two. The reason is that the behavioral architecture is achieved by the event-driven process. The simulator has to monitor the event in code 1 to trigger changing of working mode. However, 2D and 3D architectures do not need such process for determining working mode. Therefore, it makes them more “natural”.

## 5 Summary

As the first attempt, we presented three architectures of worm gear models in VHDL-AMS. Under defined requirements, the simulation result proves that worm gear models satisfy a basic analysis of worm gear behaviors.

There are still works to be done in future. The friction models used in all three architectures implement only coulomb friction without considering other components like viscous friction. But, whether it is necessary depends on also the intended investigation degree of friction behaviors. To verify the worm gear models, an improved version of modeling specification should be generated so as to concretely define behaviors to be modelled. From the aspect of validation, proper measurements should be built. Comparison between simulation result and measurement data should be conducted. Such cycle shall be repeated until an acceptable accuracy can be produced.

## 6 References

- [1] I. Dudas. *The Theory and Practice of Worm Gear Drives*. Penton Press, London, 2000.
- [2] S. Petkun. *Mechatronics simulation on system level: BroSAnT approach*. ASIM 2011, 7-9 September 2011, in Winterthur, Switzerland, ISBN: 978-3-89967-733-1, 09.2011.
- [3] J. Volmer. *Getriebetechnik. Grundlage*. Verlag Technik, Deutschland, 1995.
- [4] F. L. Litvin. *Gear geometry and applied theory*. Englewood Cliffs, Prentice Hall, NJ., 1994.
- [5] I. D. Napau. *Contributions to the modeling, simulation and experimentation of Worm-face gear drives with localized contact*. Dissertation, Technical University of Cluj-Napoca, Romania, k2005.
- [6] H. Dierich. *Weiterentwicklung der Theorie zur Ermittlung von Hertzischen Drücken und Reibungszahlen in Verzahnungen von Schneckengetrieben*. Dissertation Ruhr Universität Bochum, 1989.
- [7] R. Stolyarchuk. *Multibody modeling and dynamical analysis of the worm gear drive systems*. The 48th Scandinavian Conference on Simulation and Modeling, in Göteborg, pp. 66-72, 2007.
- [8] Matlab documentation center. *Worm gear with adjustable gear ratio and friction losses*. MathWorks, 2012.
- [9] I. Dudas, K. Banyai and G. Varga. *Simulation of meshing of worm gearing*. ASME 7<sup>th</sup> International Power Transmission and Gearing Conference, San Diego, 1996. Proceedings, pp. 141-146.
- [10] P. Horak. *Computer model of the contact relations of worm gear pairs*. 4<sup>th</sup> World Congress on Gearing and Power Transmission, Paris, 16-18 March, 1999. pp 483-488.
- [11] K. Varadi and R. Poller. *Analysis of gear teeth contact by the finite element method*. Acta Technica Acad. Sci. Hung., 101(4), pp. 397-416, 1988
- [12] T. Bercsey and P. Horak. *Finite Element Analysis of ZTA TYPE Worm Gear Pairs*. Design of machines and structures, vol1 2002, pp 145-150, 2002.
- [13] Saber® MAST library. *Worm gear*. Synopsys© Saber®.



## Practical aspects of modeling of mechatronics system at the example of BroSAnT system

Sergey Petkun

Brose Fahrzeugteile GmbH & Co. Kommanditgesellschaft, Hallstadt

*Sergey.Petkun@Brose.com*

In this paper will be discussed the peculiarities of modeling mechatronics system. An example of modeling mechatronics system is based on BroSAnT (Brose System Analyses Tool) system. But the common features like input and analyses can be generalized for other modeling system. The most important is the modeling the mechatronics devices. The focus of discussion is not evaluation and numerical complexities but rather organization pre- and post- processing processes. The client-server architecture allows using the simulation system at different domains which leads to the increasing the model quality. The main idea going through the paper is that for the simulation of mechatronics system requires new conceptions like pre- and also post-processing. For the preprocessing, the domain dependent component is the primary element. And for post process the elementary unit is single simulation running like minimal set for analyses. The time based analyses is essential feature of mechatronics simulation. The important features as simulation history tracking and precise description of simulation condition are become inevitable for right estimation the simulation results. The animation of result belongs to post process without any doubts. At the simulation of one classic mechatronics system like electric power window regulator is demonstrated the main features with some possible solution of them. The mechatronics modeling is moving in the middle between precise solution and practicable usage.

### 1 Introduction

System simulation is very popular theme at the different conferences and meetings. But the usage of such simulation on daily business projects support is essential seldom [1].

The reason for it is very simple - the simulation processes as simulation construction, calculation and result analysis can not be held in time with project running time. Up to now, there is the popular meaning that the simulation of system can be build from the component simulations. It can have place, but by that, it is often forgotten that the first simulation result for system analysis will be demanded already at acquisition phase as basic for technical and economical feasibility of a project. At this project phase, the components of the system may not exist at all. Sometimes it can be used some existing components. However if the system should be developed and built as new one – it should be presented something what is new in the system or new combination of existing components. The building of the system simulation on the basic of component simulations is technically and economically not reasonable. It will be not awaiting the exact precision on simulation results but the simulation should give some feeling about feasibility and potential problems at the realization phase. At

next project phases the question will be changed in direction concretization of components and the requirement on reliability of results is continually increasing.

### 2 Requirements on construct part of system

#### 2.1 System synthesis

To divide system into simulation components is not so evidently simple task like it seems. There are a number of components which can be separated from the system on simple natural criteria like availability at the commercial market as standalone independable on usage product. Some motor unit can be an example for it. There is another criterion like development or usage product specific unit. The scope of this criterion is changing with growing integration processes. Some example for it can be window regulator drive, where the worm gear is already integrated with motor unit.

By the typical construction of mechatronics system there are components which are presented almost ever. The source of power like battery or the transformer of electrical energy in mechanical one (usually drive unit), and sensors belong to obligatory compo-

nents. The executive mechanism (mechanics) is the most variable part in mechatronics system. The mechanics part is strongly dependable on its task. The different solutions can exist for the same task also. The cable and cross arm systems are widespread at present in the industry for window regulator systems. The both mechanics system are used for the same aim: put up and down the window glass of car.

There are a number of the components which don't represent real physical components of system but rather they belong to technical necessary elements for simulation. For example, the switch sequences during the simulation should be defined before simulation start. The software algorithm module is some component by simulation construction but belongs to electronics and is not physical one.

## 2.2 Parameterisation of component

The definition of parameter set for some component is one of the most important parts in the construction of the system. As a rule, the most of functionality is described in specification on the component. But it is often not complete, because of different reasons like to keep low price or to defend the knowhow. The functional description of the required component covers not always full required functionality. It is connected also with trend to standardisation, which increase commercial success on the market. For some components there are already standard specifications like for drives. For other components, requirements are contract specific and dependent on company-customer agreement. A window regulator mechanics is typical example of such component.

The rough rule by the parameterisation of component sounds like this "the minimal number of parameter which allows simulating the functional behaviour of it". The perfect situation is if the all parameters of component are available from data sheet on it. But such situation is more some exclusion as rule.

The requirement on component will be given very often in form some functional table look up like characteristic line for drive (rotational velocity, current through load torque). As a general rule, the depth of modelling of component is directly connected with a number of its parameters. That is why the depth of system level should not be greater than necessary for proper functionality the whole system. Especially at first phase of system analysis, it is recommended to take the simplest component model. At this point of analysis, it is not required much information about

the component but rather principal behaviour whole system.

For some components like control sequence or software module, the parameterisation process is very natural: software has usually own parameter set and control sequences consist from only event inputs (time – switch state). In BroSAnT realisation, configurator looks like at the Figure.1:

System Configuration
C:/Test/Simulation/asim
Assembly
generic_window_reg
Drive
Sensors
Electronics
Mechanics
Door
Software
Anti_Trap_Spring
Battery_Voltage
Control
Simulation overview
Simulation

Figure 1: Configurator of BroSAnT

## 3 Calculation process

After building the system from components together, the configurator sends the simulation data set to evaluation server, which tests the simulation for the properly content and returns corresponding respond (all is o.k. or something goes wrong)

### 3.1 Server architecture

The advantages of server-client architecture are evidently

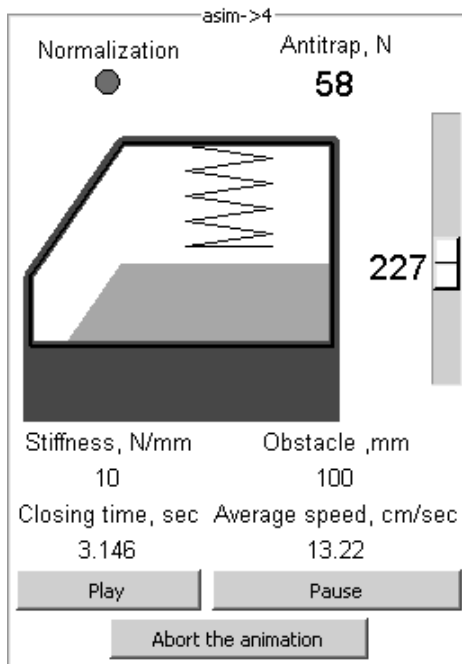
- high performance of calculation
- small number of licences
- central administration of models
- simulation access over the world

The server is running on Linux platform what leads to restriction on choice the available solver. But it simplifies the organisation of many tasks connected with usual client - server architecture. The most of solvers, which are used in electronics, are interactive tools in contrast to mechanics.

That means that the evaluation time remains in reasonable range: not more as five times greater as real time the single simulation. The evaluation time depends very strong from the number and the quality of the models included in the general model of whole system.

### 3.2 Client architecture

The client part is a more complex, because the number of requirements to end user tasks is essential greater. Not only system configurator but also result analyser is placed on the client part. The traceability of simulation running, repeatability of the results and recovering the simulation environment are obligatory features of system simulation. They impose a number of additional requirements to client GUI (Graphical User Interface) and also to its structural organisation. The difficult of organisation such front end consists also in interdisciplinary possessory of user groups.



**Figure 2:** Animated presentation of simulation results

The different thinking in electronics and mechanics make this task enormous complex to find the common basis for all simulation participants.

For example, the animated presentation of result was asked too often from mechanics and not necessary as usual as for electronics. The Figure 2 shows the example of animated result presentation.

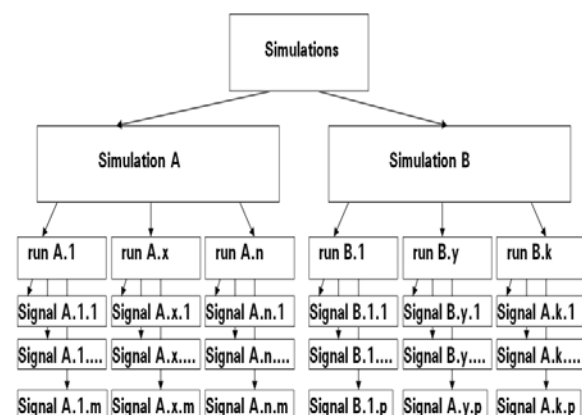
## 4 Organisation of Result Analysis

### 4.1 Common remarks

It is a common situation, when all efforts are given for building the simulation, to obtain the reasonable results. At the end, there is not enough time for its proper detailed analysis. The following question comes immediately: what is the reasonable result. In the case, if simulation target group remains like the only group, the answer is more or less clear. But at system level, the questions are coming normally from different groups. And every group has own meaning what as the first should be rated. For electronics, the electrical values like currents, voltages are the most interesting and important. The mechanics sets value on forces and deformations. The algorithm developer tries to analyse the timings and delays. The common feature by all is transient analyses. This is the speciality of system simulation: the transient curves are analysed. Not all curves are analysed with the same attention. The most attention will be drawn to the values that are familiar from daily business: usually measure results. All of that has the great influence at concept on a scope tool. The graphical scope is the main tool for system simulation analysis.

### 4.2 The main unit of simulation

As already mentioned, the natural feature of system simulation is the transient analysis. Thanks to relative small simulation time and simplicity by the changing the parameters, the big number of simulation runs can appear into relative short period. The structure of result can be presented something like at Figure 3. From the structure one can find out that the reasonable unit of system simulation is the single simulation itself.



**Figure 3:** Structure of simulation results

The simulation **A** was simulated **n** times and every run contains the **m** signals. Another simulation B has **k** runs and every runs contains **p** signals.

The structuring of data set can be continued: the structuring of data set run **A.1** according to signal nature: analog, digital or events or physical container: files.

There are two sets of the result for analysing: the different transient curves for the only simulation and the same values, like e.g. voltage, but from different simulation runs. In our example, the first set can contain the all signals: from **A1.1.1** up to **A1.1.m**. Second set consists of all signals like **A.1.1.1** up to **A.1.n.1** (may be e.g. voltage on the motor pins). The Signal B.1.1 can be also some voltage on motor pin.

It is clear, that the possibility to compare not only the different runs between each other but also the similar signal from different simulation should be available in an analysing system. The two important aspects appear at this moment: simulation run administration and traceability of analysed results. Going out from the first task, the main atomic unit the scope in BroSAnT is a set for the single simulation run. This set can consist from the single or a number of result files.

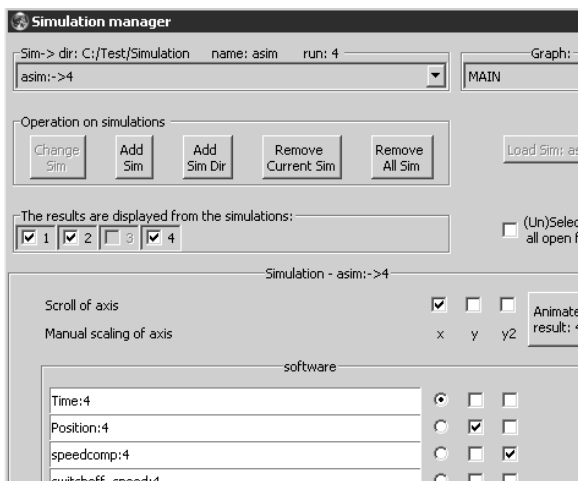


Figure 4: Fragment of simulation manager

The common property for one simulation run (for example, Simulation A – Run A.1.1) is the same parameter set that means the same construction, under the same condition with the equal control sequences within the simulation set. The whole result analysis is managed by simulation manager but rather file manager. The fragment of the BroSAnT simulation manager is shown at the Figure 4.

There are a number of situations which don't suit in common schema like function "change of component". By usage this function, the part of signals will be replace through another one. In our example, the signal number **m** in simulation A can be changed in another number new one like **m\***.

The reasonable organization of different comparisons is enormous difficult because there is no standardization in the scoping processe for analyzing the system.

### 4.3 Result traceability

The classic question on simulation "what will be if ...?" lead the fast growing the number of runs and it is very important not to lose the overview by result comparison. By present concept, there is no difference between the construction and environmental changes by means parameter variation. That is why, the traceability of simulation runs should keep clear way of parameter changing. It is not simple because of useful function "change (replace) component".

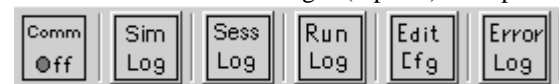


Figure 5: Fragment of control bar

For this purpose, the protocol of parameter changing is writing in some log file, which helps to recover later whole simulation run. Possibilities for traceability in BroSAnT run are shown at the Figure. 5.

Eventual this protocol should be extended for additional information such a who and when changed the simulation. There is no end to protocol the necessary information like which simulator version, under which operational system and so on.

The system simulation result is not the only effort of some single group but the whole project participants.

### 4.4 Parameter variation space

In the [2] the most interesting variation fields like constructive setting, climatic and dynamic environment, working condition and algorithmic sets were already described. If this space will be completed with tolerance requirements and influences tear-wear, the number of possible parameter variations is increased more as exponential. That is why, it is impossible to test all space of interests without exactly restricted plan for essential virtual tests.

## 5 Discussion and outlook

The system simulation is not the same like component simulation. From beginning up to the end, it has its own requirement such as less as possible parameters, not so precise result but reliable and quick. The different interests to simulation results also as different variation on domain specific parameter lead to specific requirements on post analysis tool.

The first analysis occurs at early phase of project what excludes the usage of component simulation. The system simulation focuses more on the interaction between components as on components itself. The next special feature of mechatronics analysis is transient behaviour: all is changing in time. The fast calculation time and huge possibilities for parameter variations can lead to big number of simulation runs in short period of time, which requires the peculiar methods of the transparent traceability of simulation results. The different domains of end users bring some additional difficulties that should be taken into account by results analysis.

The BroSAnT [2] is the example of realisation all this requirements that were applied to company specific products like electric power window regulator. The experience of using the multi-domain simulation, where the mechatronics simulation belongs, shows the most problems are in the intersection of domains but rather in domains itself.

The perfect suitability is not the same as compatibility. Exactly this property brings the optimal performance for whole system. It should not be forgotten, that it is valid for some clear restricted part of application space. There is no universal optimal in global sense. The mechatronics simulation helps to find such optimal combination for best performance. Despite of classical requirement on system simulator like (pre-process, solver, post process), there are a number of specific ones that comes from mechatronics nature (multi-domain possessory, transient analysis). It is important to find properly level of expected satisfaction for all domain participants to get synergy effect by system analysis.

## 6 References

- [1] Petkun, Sergey: *Mechatronics simulation on system level: BroSAnT approach*. Proceeding ASIM 2011, Winterthur, Switzerland, 7-9.09.2011

- [2] Petkun, Sergey: *BroSAnT - an example of the mechatronical way of thinking*. Proceeding ASIM 2012, Wolfenbüttel, 23-24.02.2012



## Autorenverzeichnis

### A

A Dang, Quang - Bergische Universität Wuppertal .....	191
Alkov, Ilja - FH Bielefeld .....	125

### B

Beater, Peter - Fachhochschule Südwestfalen .....	135
Boden, Maik - Technische Universität Dresden .....	35
Buchta, Robert - Hochschule Ostfalia .....	211
Büker, U. - HELLA KGaA Hueck & Co. ....	97

### C

Clauss, Christoph - Fraunhofer IIS/EAS .....	21
Commerell, Walter - Hochschule Ulm .....	111

### D

Decius, N. - HELLA KGaA Hueck & Co .....	97
Diehl, Waldemar - Hochschule Ostfalia .....	89
Diekmann, Robin - FH-Bielefeld .....	75
Dörner, Jens - Universität Duisburg-Essen .....	161

### E

Ebling, Dirk - FH Düsseldorf .....	1
Ehrhardt, Matthias - Bergische Universität Wuppertal .....	191
Engel, Ole - XRG Simulation GmbH .....	31

### F

Frei, Stephan - TU Dortmund .....	199
-----------------------------------	-----

### G

Gottelt, F. - XRG Simulation .....	31
------------------------------------	----

### H

Haase, Joachim - FhG IIS-EAS .....	83
Haubrich, Tobias - Hochschule Bonn-Rhein-Sieg .....	51
Hessel, Ewald - Hella KGaA Hueck & Co. ....	83
Holdschick, Hannes - Daimler AG .....	111

### J

Jax, Tim - Hochschule Bonn-Rhein-Sieg .....	193
---	-----

### K

Kabitzsch, K. - Technische Universität Dresden .....	35
Kießling, Jan - Hochschule Ostfalia .....	105
Köhler, U. - HELLA KGaA Hueck & Co .....	97
Köster, Michael - Technische Universität Dortmund .....	183

---

## L

Le, Duc – Bergische Universität Wuppertal .....	191
Liu-Henke, Xiaobo – Hochschule Ostfalia .....	89, 105, 211
Lückerath, Daniel – Uni Köln .....	43

## M

Majetta, Kristin – Fraunhofer IIS/EAS .....	21
Masjosthusmann, Christopher – HELLA KGaA Hueck & Co .....	97

## N

Nasri, Mounir – DLR .....	155
---------------------------	-----

## R

Razavi, Ali – TU Dortmund .....	199
Renz, Ala – XRG Simulation GmbH .....	31

## S

Scheele, Michael – Hochschule Ostfalia .....	211
Scheier, Stanislav – TU Dortmund .....	199
Schmid, Sebastian – Voith Turbo GmbH & Co. KG .....	147
Schmidt, Artur – Hochschule Wismar .....	65
Schmidt, Thorsten – Hochschule Ansbach.....	21
Schwatinski, Tobias – Hochschule Wismar .....	57
Speckenmeyer, E. – Uni Köln .....	43
Steinebach, G. – Hochschule Bonn-Rhein-Sieg .....	193
Swoboda, Antje – Universität Duisburg-Essen .....	161

## P

Pawletta, Thorsten – Hochschule Wismar .....	57, 65
Petkun, Sergey – Brose Fahrzeugteile GmbH & Co. KG .....	219, 227

## Q

Quantmeyer, Florian – Hochschule Ostfalia .....	89, 105
---	---------

## T

Tran, Gia Lich – Bergische Universität Wuppertal .....	191
--	-----

## U

Ullrich, O. – Uni Köln .....	43
------------------------------	----

## W

Weidemann, Dirk – FH-Bielefeld .....	75, 125
Weger, Heiko – Technische Universität Dresden .....	35
Windemut, Ivan – DLR .....	119
Wortberg, J. – Universität Duisburg-Essen .....	161



---

## **X**

Xin, Zhao - Brose Fahrzeugteile GmbH & Co. KG ..... 219

## **Z**

Zimmermann, Jens - Altran GmbH ..... 7, 171







**ARGESIM**