

ASIM-Workshop STS/GMMS 2014

Treffen der ASIM/GI-Fachgruppen:

Simulation technischer Systeme

Grundlagen und Methoden in Modellbildung und Simulation

**20. bis 21. Februar 2014 in
Reutlingen-Rommelsbach**

Tagungsband

Jürgen Scheible (Hrsg.)
Ingrid Bausch-Gall (Hrsg.)
Christina Deatcu (Hrsg.)



Arbeitsgemeinschaft Simulation ASIM in der Gesellschaft für Informatik GI

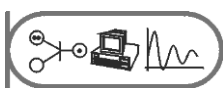
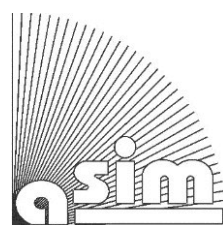


Hochschule Reutlingen
Reutlingen University



Robert Bosch Zentrum für Leistungselektronik

Hochschule Reutlingen • Universität Stuttgart • Robert Bosch GmbH



ARGESIM Reports

Published by **ARGESIM** and **ASIM**, Arbeitsgemeinschaft Simulation,
Fachausschuss 4.5 der GI

Series Editor:

Felix Breiteneker (ARGESIM / ASIM)
Div. Simulation, Vienna University of Technology
Wiedner Hauptstrasse 8 - 10, A - 1040 Vienna
Tel: +43-1-58801-10115, Fax: +43-1-58801-10199
Email: Felix.Breiteneker@tuwien.ac.at

ARGESIM Report 42 ASIM Mitteilung AM 149

Titel: ASIM-Workshop STS/GMMS 2014
Treffen der ASIM/GI-Fachgruppen:
Simulation technischer Systeme
Grundlagen und Methoden in Modellbildung und Simulation

Herausgeber: Jürgen Scheible
Ingrid Bausch-Gall
Christina Deatcu
Email: christina.deatcu@hs-wismar.de

ISBN print 978-3-901608-82-7, 2014

ISBN CD 978-3-901608-81-0, 2014

ISBN Ebook 978-3-901608-42-1, 2014

DOI 10.11128/arep.42

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

© by ARGESIM / ASIM, Wien, 2014 – Hochschule Reutlingen

ARGE Simulation News (ARGESIM)
c/o F. Breiteneker, Div. Simulation, Vienna Univ. of Technology
Wiedner Hauptstrasse 8-10, A-1040 Vienna, Austria
Tel.: +43-1-58801-10115, Fax: +43-1-58801-10199
Email: info@argesim.org; WWW: <http://www.argesim.org>

Druck:

WENZEL GmbH druck . kopie . media
München

Vorwort

Mit der Verfügbarkeit leistungsfähiger Computer haben rechnergestützte Simulationsverfahren überall in Wissenschaft und Technik Einzug gehalten. Die modellbasierte Simulation als „virtuelles Experiment“ stellt insbesondere im Entwurf technischer Systeme ein wirksames und längst unverzichtbares Hilfsmittel dar, um Entwicklungsergebnisse hinsichtlich gewünschter Eigenschaften abzusichern. Die Möglichkeiten heutiger Simulationsmethoden sind faszinierend, weshalb gerade Anfänger (aber nicht nur diese) der Gefahr ausgesetzt sind, deren Ergebnisse unkritisch zu übernehmen. Besondere Bedeutung kommt hier der Lehre zu. Neben der Anwendung der Simulationswerkzeuge ist es wichtig, den Studierenden auch deren theoretische Grundlagen nahe zu bringen und damit ihr Bewusstsein hinsichtlich der Grenzen der Simulation zu schärfen.

Der Workshop der ASIM/GI-Fachgruppen »Simulation technischer Systeme« und »Grundlagen und Methoden in Modellbildung und Simulation« bringt Fachleute aus Wirtschaft und Wissenschaft zum Erfahrungsaustausch rund um die Simulation zusammen. Hierbei werden alle Aspekte von den Grundlagen über die Methoden bis hin zu Werkzeugen und Anwendungsbeispielen angesprochen. Wir freuen uns sehr, dass wir am Robert Bosch Zentrum für Leistungselektronik der Hochschule Reutlingen die diesjährige ASIM ausrichten können. Die Studierenden und Doktoranden erhalten so die Gelegenheit, das wichtige Thema Simulation in seinen vielen Facetten vor Ort zu vertiefen.

Ein besonderes Merkmal der ASIM ist die breite thematische Streuung, die sich in 14 Sitzungen zu unterschiedlichen Schwerpunktthemen mit insgesamt 44 angenommenen Beiträgen widerspiegelt. Die Besucher haben so die Gelegenheit zum Informationsaustausch auch über Anwendungsgebiete und Ingenieursdisziplinen hinweg. Daneben wird das Treffen wieder begleitet von einer Ausstellung, auf der namhafte Firmen ihre Werkzeuge und Lösungen präsentieren.

Der vorliegende Tagungsband enthält die Langfassungen der Beiträge, für deren Bereitstellung wir uns bei allen Autoren bedanken. Weiterhin danken wir allen Mitorganisatoren der ASIM/GI-Fachgruppen STS und GMMS für die im Rahmen der ASIM geleistete Arbeit. Ganz besonderer Dank gebührt hier Frau Dr. Ingrid Bausch-Gall für ihre vielfältige Unterstützung in der Organisation und ihren unermüdlichen Einsatz rund um die ASIM. Darüber hinaus danken wir Frau Susan Krause und Frau Tatjana Mayer für die Übernahme des Organisationsbüros vor Ort, sowie der Robert Bosch GmbH für die Unterstützung der Veranstaltung.

Allen Besuchern der ASIM wünschen wir zwei spannende Konferenztage, interessante Vorträge und einen regen fachlichen Gedankenaustausch.

Herzlich willkommen zur ASIM2014 am rbz in Reutlingen-Rommelsbach!

Heinz-Theo Mammen

ASIM/GI-Fachgruppe »Simulation technischer Systeme«

Thorsten Pawletta

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

Jürgen Scheible

Hochschule Reutlingen – Robert Bosch Zentrum für Leistungselektronik

Fachbereich Electronic Design Automation

Reutlingen, Februar 2014

Tagungsleitung:

Ingrid Bausch-Gall, BAUSCH-GALL GmbH, München

Christina Deatcu, Hochschule Wismar

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

Thorsten Pawletta, Hochschule Wismar, ASIM/GI-Fachgruppe GMMS

Jürgen Scheible, Hochschule Reutlingen, Robert Bosch Zentrum für Leistungselektronik

Programmorganisation:

Ingrid Bausch-Gall, BAUSCH-GALL GmbH, München

Walter Commerell, Hochschule Ulm

Leo Gall, BAUSCH-GALL GmbH, München

Joachim Haase, Fraunhofer IIS/Institutsteil EAS Dresden

Daniel Lückerrath, Uni Köln

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

Klaus Panreck, Fachhochschule Bielefeld

Thorsten Pawletta, Hochschule Wismar, ASIM/GI-Fachgruppe GMMS

Jürgen Scheible, Hochschule Reutlingen, Robert Bosch Zentrum für Leistungselektronik

Michael Striebel, ZF-Lenksysteme GmbH, Schwäbisch Gmünd

Tagungsorganisation:

Tatjana Mayer

Hochschule Reutlingen

Robert Bosch Zentrum für Leistungselektronik

Oferdingerstr. 50, 72768 Reutlingen-Rommelsbach

Tel: +49 (7121) 271-7085

www.reutlingen-university.de

www.rbzentrum.de

Tagungsort:

Robert Bosch Zentrum für Leistungselektronik

Oferdingerstr. 50

72768 Reutlingen-Rommelsbach

Veranstalter:

ASIM/GI-Fachgruppe Simulation technischer Systeme

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

www.asim-gi.org/sts

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

Die Fachgruppe *Grundlagen und Methoden in Modellbildung und Simulation* (GMMS) befasst sich in enger Zusammenarbeit von Industrie und Forschungseinrichtungen mit neuen methodischen Entwicklungen zu Modellierungsansätzen, numerischen und softwaretechnischen Verfahren, Algorithmen, Simulationswerkzeugen sowie mit Problemen der simulationsgestützten Optimierung. Von besonderem Interesse sind Methoden und Werkzeuge, die über mehrere Anwendungsdomänen hinweg eingesetzt werden.

Einen weiteren Schwerpunkt bildet die Anwendung von Methoden der Modellbildung und Simulation in der Lehre und in der Weiterbildung. Darüber hinaus sind die Arbeitsgruppen Simulation und KI und Verkehrssimulation in die Fachgruppe integriert, die sich spezifischen Problemstellungen widmen. Eine lange Tradition hat der Vergleich von Modellierungs- und Simulationsansätzen sowie deren Unterstützung durch Simulationswerkzeuge, welcher mit den ARGESIM Comparisons on Modeling and Simulation Techniques and Simulation Software verbunden ist. Die Comparisons werden regelmäßig in den Simulation Notes (ehemals Simulation News) Europe (SNE) publiziert.

Die Fachgruppe ist Mitorganisator des jährlichen Doktorandenworkshops Trends in Computing and Scientific Engineering (TCSE) sowie des gemeinsamen Jahresworkshops der Fachgruppen STS und GMMS und organisiert weitere themenspezifische Workshops.

www.asim-gi.org/gmms

Hochschule Reutlingen – Robert Bosch Zentrum für Leistungselektronik

Die Hochschule Reutlingen entstand aus einer 1855 gegründeten Webschule. Starkes Wachstum führte 1975 zum Bau eines eigenen Campus „auf dem Hohbuch“, wo heute über 5000 Studierende in den fünf Fakultäten Technik, Informatik, Angewandte Chemie, Textil und Design und der European Business School ESB untergebracht sind. Die Hochschule erzielt regelmäßig exzellente Rankings.

Der stark forcierte Ausbau der Leistungselektronik in der Robert Bosch GmbH führte 2009 zur Stiftung des Robert Bosch Zentrums für Leistungselektronik (rbz). Das rbz ist ein Forschungs- und Lehrverbund, in dem sich die Bosch-Gruppe, die Hochschule Reutlingen und die Universität Stuttgart zusammen geschlossen haben mit dem Ziel, exzellente Ingenieure in diesem Fachbereich auszubilden. Diese Kooperation ist die erste und bisher einzige dieser Art in Deutschland.

Am rbz-Standort Reutlingen-Rommelsbach wird seit 2010 der Master-Studiengang Leistungs- und Mikroelektronik angeboten. Hier können Studierende beispielweise ihren eigenen IC entwickeln, der in der Bosch Waferfab gefertigt wird. In Forschungsprojekten zu Leistungshalbleitern, Integrierten Schaltungen und Entwurfsautomatisierung arbeiten heute 13 Doktoranden und weitere Mitarbeiter.

www.reutlingen-university.de, www.rbzentrum.de

Tagungsband

ASIM-Workshop STS/GMMS 2014

Treffen der ASIM/GI-Fachgruppen:

Simulation technischer Systeme

Grundlagen und Methoden in Modellbildung und Simulation

Inhalt

Inhalt

Plenarvorträge

- Aktuelle Trends in der FE-Simulation von Fertigungsprozessen in der Automobilindustrie 1
Dimitrij Tikhomirov

Grundlagen und Methoden I

- Advancing Virtual Commissioning with Variant Handling 7
Johannes Möck, Jens Weiland
- Anforderungen an State Event Charakterisierungen für die Simulation hybrider Modelle anhand von Benchmarks 15
Andreas Körner, Horst Ecker, Felix Breitenecker
- Methodenvergleich in der Simulation von Grundwasserverschmutzung 17
Stefanie Winkler, Martin Bicher, Felix Breitenecker

Simulation in der Elektronikentwicklung I

- Ansätze der Modellierung einer induktiven Übertragungsstrecke mit dem Simulationstool OctaveFEMM 19
Thomas Lang
- Reduzierte Modelle für elektromechanische Bauteile mit Berücksichtigung von Wirbelströmen 25
Ursula Voß, Martin Hanke
- Kühlungsoptimierung von elektronischen Komponenten mittels Strömungssimulation 31
Bernd Scheiderer, Wolfgang Schlüter, Ansgar Ringleb
- Thermische Simulation von Bonddrähten in verpackten Chips unter Berücksichtigung der Draht-Package-Interaktion 37
Carl Christoph Jung, Christian Silber, Jürgen Scheible

Physikalische Modelle & Modelica

- Integrierte Analyse von Modelica-Modellen mit MapleSim und Maple 49
Johannes Friebe, Thomas Richard
- Using Functional Mock-up Interface in Software-in-the-Loop Simulation Applications 51
Michael Seibt
- Physikbasierte Simulation im Anlagenbau 57
Stefan Gulan, Ulrich Odefey, Thomas Bär, Herbert Beesten, Holger Hämmerle, Bernd Kärcher, Matthias Riedl
- Fallstudien zur Modellierung physikalischer Systeme in der experimentellen Archäologie 63
Johannes Tanzler, Philipp Pichler, Bernhard Heinzl, Hans Reschreiter, Kerstin Kowarik, Felix Breitenecker

Mathematische Verfahren in Modellbildung und Simulation

- State Estimation with Unscented Kalman Filter for Higher Index Nonlinear Differential-Algebraic Systems 65
Ilja Alkov, Dirk Weidemann
- Convergence of Dynamic Iteration for Coupled Engineering Problems 73
Sebastian Schöps, Andreas Bartel, Michael Günther
- Über die Simulation differential-algebraischer Modelle mit nichttrivialem Index 81
Carina Pöll, Irene Hafner, Bernhard Heinzl, Felix Breitenecker

Simulation in der Elektronikentwicklung II

Universelle OTA-Testbench	83
<i>Andreas Gerlach, Moritz Junge, Jürgen Scheible</i>	
Ein Effizienzmodell für getaktete Schaltwandler im Multi-MHz-Bereich	89
<i>Achim Seidel, Jürgen Wittmann, Bernhard Wicht</i>	
Effektive Fehlerdiagnose von Strukturen mit Digital-Analog-/Analog-Digital-Konvertern mit einem Cloud-basierten Workflow	99
<i>Matthias Gulbins, André Schneider, Steffen Rülke</i>	

Simulation mechatronischer Systeme

Gekoppelte vs. integrierte Simulation der Steuerungs-, Regelungstechnik und Strukturodynamik mechatronischer Systeme am Beispiel von Werkzeugmaschinen	109
<i>Gerhard Kehl, Peter Wagner</i>	
Theoretische und experimentelle Untersuchungen der Vertikaldynamik eines elektrischen Rollstuhls mit dem Ziel der Optimierung des Fahrverhaltens durch semiaktive Dämpfer	113
<i>Sönke Lück, Rolf Naumann</i>	
Konzeption eines hochdynamischen Systems mit sphärischem Elektroantrieb	119
<i>Marian Göllner, Xiaobo Liu-Henke</i>	
Nutzung einer Parametererregung in MEMS	127
<i>Till Kniffka, Johannes Welte, Horst Ecker</i>	

Simulation thermischer Systeme

Model Based Optimization of Building Control Systems	131
<i>Christoph Clauss, Jürgen Haufe, Torsten Blochwitz, Edgar Liebold, Ullrich Hintzen, Volker Klostermann</i>	
Dynamische Simulation von Gebäuden und Anlagen für mehr Komfort und mehr Effizienz	139
<i>Sven Rutkowski</i>	

Simulation in der Automobilindustrie

Mechatronische Entwicklung eines HiL-Prüfstands zur Erprobung einer aktiven Federung	151
<i>Michael Scheele, Sven Jacobitz, Xiaobo Liu-Henke</i>	
Ein skalierbares Echtzeitsystem zur Erprobung des Batteriemanagements in Elektrofahrzeugen	161
<i>Florian Quantmeyer, Matthias Roch, Waldemar Diehl, Xiaobo Liu-Henke</i>	
Powertrain Cosimulation in a MiL and SiL Setup using the AUTOSAR and FMI Standards	167
<i>Christoph Störmer, Christoph Malz, Corina Mitrohin</i>	

Simulation von Verkehrssystemen

A model for airline personnel schedule simulation	171
<i>Patrick Kuckertz, Hubert Randerath</i>	
3D-Simulation und Scheduling mit Simio	181
<i>Markus Bans</i>	
Multi-depot multi-vehicle-type vehicle scheduling for Cologne's tram network	191
<i>Daniel Lückcrath, Oliver Ullrich, Aleksander Kupicha, Ewald Speckenmeyer</i>	

Grundlagen & Methoden II

Modellbibliothek für die Interaktion von Robotern in der MATLAB/DEVS-Umgebung auf Basis des SBC-Frameworks <i>Birger Freymann, Thorsten Pawletta, Tobias Schwatinski, Sven Pawletta</i>	199
2Simulate: A Distributed Real-Time Simulation Framework <i>Jürgen Gotschlich, Torsten Gerlach, Umut Durak</i>	209
Henon - eine alternative numerische Methode zur Ereignislokalisierung <i>Franz Preyser, Bernhard Heinz, Felix Breitenecker</i>	215

Simulation und Energieeffizienz

Control Strategies for Energy-Optimized Operation of Assembly Systems <i>Robin Diekmann, Joscha Heinze, Ilja Alkov, Dirk Weidemann</i>	217
Interdisziplinäre Optimierung der Energieeffizienz in Fertigungsbetrieben mittels Co-Simulation <i>Bernhard Heinzl, Wolfgang Kastner, Matthias Rößler, Friedrich Bleicher, Fabian Dür, Iva Kovacic, Ines Leobner, Niki Popper</i>	225

Simulation in der Elektronikentwicklung III

A Modelica Based Lithium Ion Battery Model <i>Imke Krüger</i>	229
Untersuchung eines alternativen Aktorkonzepts für aktive Bedienelemente auf Basis von Open-Source Simulationsprogrammen <i>Nanno Peters, Gerhard Stebner, Christoph Hartwig</i>	237

Technische Simulationsanwendungen

Simulation of hybrid Railway Vehicles and Comparison of electrical Energy Storage Systems <i>Sebastian Schmid</i>	243
Hybride Simulation des Materialflusses in Aluminium-Druckgussbetrieben <i>Sven Hirschberg, Wolfgang Schlüter, Ansgar Ringleb</i>	257

Analyse von Simulationsergebnissen

Automatisierte Auswertung von DYMOLA-Simulationsergebnissen mit MATLAB <i>Ivan Windemut, Mounir Nasri, Holger Dittus</i>	263
---	-----

Aktuelle Trends in der FE-Simulation von Fertigungsprozessen in der Automobilindustrie

Dmitrij Tikhomirov¹

¹Hochschule Hamm-Lippstadt, Marker Allee 76-78, 59063 Hamm

Dmitrij.Tikhomirov@hshl.de

Im globalen Wettbewerb werden die Automobilhersteller und –zulieferer vor neuen Herausforderungen gestellt. Zur Erhöhung ihrer Wettbewerbsfähigkeit sind die Unternehmen der Automobilindustrie angehalten, die Produktentwicklungszeiten bei steigenden Qualitätsanforderungen deutlich zu senken. Insbesondere die Zeiträume für die Fertigungsplanung sind stark eingeschränkt.

In diesem Beitrag wird vorgestellt, wie die Finite-Elemente-Simulation die Planungsprozesse unterstützen kann. Anhand von Beispielen wird die Anwendung der Schweißsimulation bei der Fertigungsplanung aufgezeigt. Durch eine frühzeitige Ermittlung des Schweißverzugs können die Fertigungsschritte sicherer geplant und die Prototypenanzahl reduziert werden.

Des Weiteren wird die Abbildung von Fertigungsprozessketten in der FE-Simulation diskutiert. Anhand eines Praxisbauteils wird die Bedeutung der Berücksichtigung von Fertigungsprozessen bei der Vorhersage von Bauteileigenschaften demonstriert. Die Realisierung der virtuellen Prozesskette für unterschiedliche Fertigungsprozesse wird diskutiert.

1 Einleitung

Für die Erhöhung der Wettbewerbsfähigkeit produzierender Unternehmen ist eine nachhaltige Reduktion der Produktentwicklungszeiten erforderlich. Insbesondere die Zeiträume für die Fertigungsplanung sind stark eingeschränkt. Hierbei kann die Finite-Elemente-Simulation von Fertigungsprozessen einen frühzeitigen Einblick in das Verhalten und die Eigenschaften von Bauteilen ermöglichen. Außerdem kann die Machbarkeit einzelner Fertigungsschritte überprüft werden.

Immer mehr unterstützt die numerische Simulation die Fertigungsplanung bei der Absicherung von Bauteiltoleranzen während aller Fertigungsschritte sowie bei der Vorhersage von Eigenschaften gefertigter und zusammengefügter Baugruppen. Wenn die Einflüsse einzelner Fertigungsschritte, z.B. Umformen oder Fügen, auf die Bauteileigenschaften vor der Werkzeugherstellung bekannt sind, können die Fertigungsprozesse ohne aufwändige Prototypentests optimiert werden. Dadurch können die Fertigungskosten und die –zeit reduziert werden.

Bei der Herstellung von Karosserie- und Fahrwerksbauteilen spielen die Umform- und die Fügeprozesse eine große Rolle. Heute unterstützt die FE-

Umformsimulation die Produktionsplanung bei der Ermittlung wesentlicher Einflüsse des Umformprozesses auf die Blechbauteile (Blechdickenverteilung, Faltenbildung, Versagen, u.s.w.) [1]. Eine weitere Herausforderung stellt die Vorhersage der Rückfederung (Formänderung nach der Blechentnahme aus dem Umformwerkzeug) und deren Kompensation dar [2].

Mit zunehmender Verwendung von höher- und hochfesten Stählen und der damit verbundenen Reduktion der Blechdicken in der Automobilindustrie und anderen Industriebranchen gewinnt das Problem des thermischen Verzugs infolge der Schweißprozesse immer mehr an Bedeutung. Durch den Schweißverzug können die Toleranzanforderungen an die gefügten Baugruppen verletzt werden. Dies führt möglicherweise zur Notwendigkeit der Überbrückung größerer Spalte beim Zusammenbau von Komponenten. Dabei ist die Anwendung von hochpräzisen Fügeverfahren, z.B. Laserstrahlschweißen, ohne vorherige Nachbearbeitung nicht möglich. Der Aufwand zur Feststellung solcher Abweichungen in der Praxis und zur deren Beseitigung kann ganz oder teilweise entfallen, wenn die entstehenden Schweißverzüge bereits während der Fertigungsplanung bekannt sind und entsprechend berücksichtigt werden.

Die konventionellen Maßnahmen zur Schweißverzugsminimierung in der Produktion umfassen die Optimierung der Schweißfolge und die Veränderung der Spannbedingungen. Das Auffinden einer optimalen Kombination aus Schweißfolge und Spannbedingungen für komplexe Bauteile mit mehreren Schweißnähten ist nicht trivial und erfordert viele Versuche an Realbauteilen. Außerdem gibt es noch die weiteren Randbedingungen, die bei der Prozessauslegung berücksichtigt werden müssen, wie beispielsweise Einhaltung von Taktzeiten, Restriktionen bei der Zugänglichkeit einzelner Schweißnähte, Gewährleistung einer optimalen RoboterAuslastung, Vermeidung von möglichen RoboterKollisionen. Die Aufgabe der Schweißverzugsreduktion in der Produktion ist daher stets unter Berücksichtigung von genannten Aspekten sowie anderen technologischen und organisatorischen Anforderungen zu betrachten. Die optimalen Randbedingungen für verzugsarme Schweißungen in der Praxis müssen umsetzbar sein.

Können die Bauteiltoleranzanforderungen durch Planungsmaßnahmen nicht erfüllt werden, so ist es notwendig, die Einzelteilkonstruktionen zu ändern. Dies bringt allein einen hohen Aufwand mit sich, denn dabei meist auch die Werkzeuggeometrien für die Umformwerkzeuge geändert und diese erneut hergestellt werden müssen. Solche Iterationsschleifen in der Fertigung erhöhen vielfach die Produktionszeit und –kosten. Hierbei kann die numerische Simulation eine erhebliche Hilfe leisten. Wenn die entstehenden Schweißverzüge frühzeitig mittels FE-Simulation ermittelt werden, können die Änderungen der Einzelteilgeometrien bereits in den frühen Planungsphasen, noch vor der Herstellung der Umformwerkzeuge, berücksichtigt werden.

Die Einhaltung der Bauteiltoleranzen und die mechanischen Eigenschaften von Blechbauteilen können innerhalb der virtuellen Prozesskette untersucht werden [3][4]. Unterschiedliche Werkstoff- und Prozessmodelle sowie FE-Netze bei jeder Prozesssimulation werden innerhalb der entwickelten Softwaretools gekoppelt, so dass die Datenübertragung zwischen einzelnen Prozessschritten möglich ist. Hierbei ist es wichtig, die Kompatibilität mit verschiedenen kommerziellen FE-Softwareprodukten, die bei den jeweiligen Prozesssimulationen verwendet werden, einzuhalten. Dabei sollten auch neue, auf den Softwaremarkt kommende FE-Tools durch diese Schnittstellen in die Prozesskette integriert werden können.

Dieser Beitrag besteht aus den folgenden Kapiteln. Kapitel 2 beschreibt die Anforderungen der Fertigungsplanung an Simulationstools. Im Kapitel 3 wird anhand einiger Beispiele die Anwendung der Schweißsimulation mit dem Softwaretool Weld Planner® für die schnelle Ermittlung des Schweißverzugs in der Fertigungsplanung vorgestellt. Kapitel 4 befaßt sich mit der virtuellen Fertigungsprozesskette. Die Bedeutung der Berücksichtigung der Fertigungsprozesse für die Festigkeits- und Lebensdaueranalyse wird anhand eines Bauteils demonstriert. Die Zusammenfassung findet sich im Kapitel 5.

2 Anforderungen der Fertigungsplanung an Simulationstools

Die Vision einer durchgängigen digitalen Planung komplexer Fertigungsketten und der benötigten Produktionssysteme ist heute unter dem Stichwort „Digitale Fabrik“ ein Gegenstand umfangreicher Forschungsaktivitäten [5].

Eine wesentliche Voraussetzung zur angestrebten Kopplung der digitalen Fabrikplanung und der computergestützten Prozesssimulation ist die Verfügbarkeit leistungsfähiger Simulationstools. Die wesentlichen Kriterien dabei sind die Qualität der Simulationsergebnisse, ihre Aussagefähigkeit in Bezug auf die realen Planungsprozesse sowie der Datenaufbereitungs- und Simulationsaufwand in konkreten Planungsprojekten. Die Simulationstools für die einzelnen Fertigungsprozesse sollen einerseits die genannten Kriterien erfüllen und andererseits eine prozesskettenübergreifende Kopplung von einzelnen Fertigungssimulationen ermöglichen.

Die ständig wachsenden Genauigkeitsanforderungen auf die Modellierung von Fertigungsprozessen und die Abbildung von Prozessketten führen zur Entwicklung neuer Softwaretools für die Produktionsplanung. Für den Einsatz in der Fertigungsplanung sollen diese Tools im Allgemeinen die folgenden Bedingungen erfüllen [6]:

- gute Integrierbarkeit in die Routineprozesse der Fertigungsplanung
- möglichst universeller Einsatz
- leichte Kopplung mit anderen Simulationsverfahren der Fertigungsprozesskette
- leichte Erlernbarkeit und hohe Benutzerfreundlichkeit

- kurze Simulationszeiten

Die in den Kapiteln 3 und 4 genannten Softwaretools wurden unter Berücksichtigung dieser Kriterien entwickelt.

3 Effiziente Schweißverzugsvorhersage in der Fertigungsplanung

Das Softwaretool Weld Planner[®] wurde von der Firma INPRO [7] in Kooperation mit der Firma ESI Group [8] entwickelt. Seit mehreren Jahren ist Weld Planner[®] ein Bestandteil der Softwarepalette der Firma ESI Group und wird von der ESI Group weltweit vertrieben.

Die Simulation mit dem Weld Planner[®] basiert auf den bereits vernetzten Geometrien (FE-Netze). Es können sowohl Schalen- als auch Volumenelemente verwendet werden. In der frühen Planungsphase, nachdem die ersten Festigkeitsberechnungen durchgeführt wurden, sind solche FE-Netze bereits vorhanden.

Dieses Softwaretool ist sehr benutzerfreundlich und verfügt über die für einen Planungsingenieur notwendigen Funktionalitäten, darunter automatische Schweißnahterkennung, mehrere Möglichkeiten der Schweißnaht- und Schweißpunktdefinition, eine Auswahl industrietypischer Spannvorrichtungen sowie die Möglichkeit der Schweißfolgedefinition als ein Roboterplan. Mit dem Weld Planner[®] können Schutzgas-, Laser- und Punktschweißverfahren modelliert werden.

Sollte der ermittelte Schweißverzug nicht den Toleranzanforderungen genügen, muss der Anwender seinen Schweißplan verändern und eine neue Berechnung durchführen. Die Berechnungszeit hängt stark von der Modellgröße ab und kann ab einigen Sekunden bis zu einigen Stunden betragen. Weld Planner[®] eignet sich gut für die schnelle qualitative Ermittlung des Schweißverzugs und die Bestimmung der optimalen Schweißfolge sowie Spannbedingungen für ein verzugsarmes Bauteil. Bei geeigneter Kalibrierung des Modells anhand einfacher Testbauteile können die belastbaren quantitativen Vorhersagen erzielt werden [6].

Anhand einiger Praxisbauteile kann die Anwendung des Weld Planners[®] für die Zwecke der Fertigungsplanung demonstriert werden. Die vorgestellten Bauteile stammen aus den Bereichen Karosserie und Fahrwerk, aus welchen die meisten schweißverzugs-

relevanten Bauteile der Automobilindustrie kommen. Aus Gründen der Geheimhaltung können hier nur Verzugstrends und keine Werte gezeigt werden.



Abbildung 1. PKW-Heckdeckel: oben – globaler Schweißverzug nach der Ausspannung, unten – lokale Verzugsgradienten bei der Kleinskalendarstellung [6]

Abbildung 1 zeigt einen PKW-Heckdeckel eines Daimler-Fahrzeugs[6]. Seine Komponenten bestehen aus Aluminiumblechen, welche mit dem MIG-Verfahren geschweißt werden. Die Spannbedingungen und die Schweißfolge entsprechen denen der Produktionsplanung. Nach der Ausspannung wird das Bauteil in der entsprechenden Lage in der Messvorrichtung positioniert. Sowohl der Verzugstrend als auch die –werte wurden von der Produktionsplanung bestätigt. Durch Verwendung der Kleinskalendarstellung konnte die unsichtbare lokale Welligkeit des Randes des oberen Bleches festgestellt werden. Dieses Phänomen kann in der Praxis auftreten und wird durch entsprechende technische Maßnahmen vermieden.

Eine weitere Anwendung auf die Karosseriestrukturen ist eine PKW-B-Säule von Volkswagen, Abbildung 2 [6]. Zur Gewährleistung der Crash-Performance wird die B-Säule häufig mit Elementen aus hochfesten warmumgeformten Stählen verstärkt. Diese Verstärkungen werden mit Hilfe des Laserstrahlschweißens mit Innen- und Außenblechbauteilen gefügt. Dabei wird häufig die Vorhaltung notwendig, d.h. die B-Säule wird in der Gegenrichtung zum erwarteten Verzug vorverformt. Nach der Ausspannung weist dann das Bauteil einen geringeren Verzug als ohne Vorhaltung auf. Die Verzugswerte und –richtungen bei der Simulation dieses Bauteils stimmen mit den Messungen an Prototypenbauteilen gut überein. Ba-

sierend auf der Schweißverzugssimulation können die Werte der Vorhaltung ermittelt werden.

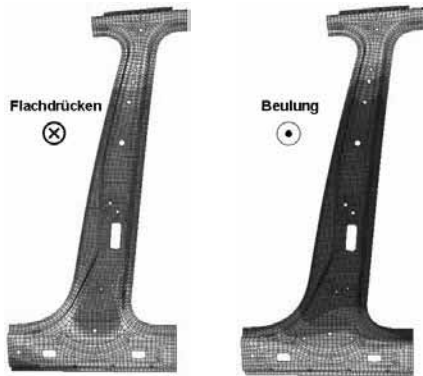


Abbildung 2. PKW-B-Säule: links – Deformation der Verstärkung bei der Vorhaltung, rechts – Verzug nach dem Schweißen und der Ausspannung [6]

Die Validierung des Softwaretools Weld Planner® erfolgte anhand weiterer Bauteile aus dem Karosserie- und Fahrwerksbereich. Einige Beispiele dazu können [6][9] entnommen werden. Neben der Vorhersage des Schweißverzugs bei gegebenen Spannbedingungen und Schweißfolge wurden noch weitere Empfehlungen für die Fertigungsplanung ausgearbeitet. Dazu gehört die Optimierung der Schweißfolge im Hinblick auf den minimalen Verzug, die Untersuchung der Einflüsse der Spannbedingungen auf den Verzug sowie die Optimierung der Spannpositionen.

4 Simulationsketten von Fertigungsprozessen für die Blechbauteile

Die wesentlichen Fertigungsprozesse der Blechbauteilgruppen der Automobilindustrie umfassen das Umformen (Kalt- oder Warmumformen), das Trennen und Fügen (thermisch oder mechanisch), das Lackieren und den Zusammenbau. Es ist heute mittels numerischer Simulation möglich, die Effekte aller dieser Teilschritte auf die Komponenten zu untersuchen. Für die neuen Bauteile müssen dabei mehrere Berechnungen, z.B. statische, dynamische (NVH), Lebensdauer- und Crashanalysen, durchgeführt werden. Alle genannten Fertigungsschritte leisten ihre Beiträge sowohl zur Maßhaltigkeit (Toleranzen) als auch zu den Bauteileigenschaften (Festigkeit, Crash), s. Abbildung 3.

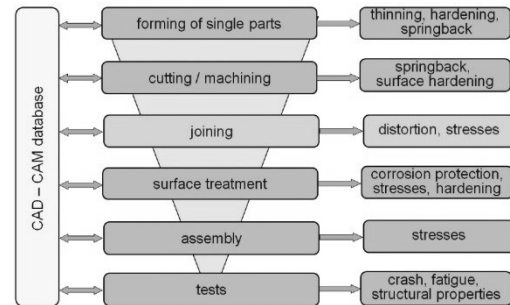


Abbildung 3. Fertigungsprozesskette im Karosseriebau sowie die wichtigsten Einflüsse einzelner Fertigungsschritte auf ein Bauteil [9]

In [10] wurde ein neues Planungstool für die Definition, Untersuchung und Ergebnisauswertung von integrierten Simulationsketten vorgestellt. Mit diesem Tool ist es möglich, einzelne Fertigungsprozesssimulationen mit unterschiedlichen kommerziellen FE-Softwareprogrammen durchzuführen. Die folgende Simulationskette wurde damit aufgebaut und untersucht: die Umformsimulation mit PAM-STAMP®, die Schweißsimulation mit Weld Planner®, die Festigkeitsanalyse mit ABAQUS® und die Lebensdaueranalyse mit FEMFAT® [10]. Dabei fließen die Ergebnisse aus einer vorhergehenden Simulation in die nachfolgende Berechnung ein. Für die Demonstration der virtuellen Prozesskette wurde ein Hilfsrahmen, bestehend aus mehreren umgeformten und miteinander verschweißten Blechbauteilen, verwendet.

Abbildung 4 zeigt welche Effekte die Umform- und die Schweißsimulationen auf die Eigenspannungen in diesem Bauteil im beanspruchten Zustand aufweisen. Die Farbskalen aller Darstellungen sind identisch, die Spannungswerte sowie die Belastungen wurden aus Geheimhaltungsgründen weggelassen. Die Darstellung in Abbildung 4, oben zeigt das niedrige Eigenspannungsniveau im Bauteil bei einem typischen Lastfall ohne Berücksichtigung von Fertigungsprozessen. Bei der mittleren Darstellung erkennt man ein deutlich höheres Eigenspannungsniveau infolge der Berücksichtigung der Umformsimulation. Bei der unteren Darstellung liefert die Berücksichtigung der Umform- und der Schweißsimulation noch eine weitere Eigenspannungszunahme, insbesondere im Bereich der Schweißnähte. Es soll an dieser Stelle angemerkt werden, dass die Untersuchung der Eigenspannungen nur ein Aspekt einer Festigkeitsanalyse darstellt. Insgesamt kann festgestellt werden, dass die Fertigungsprozesse wie das Umformen und das Schweißen einen nicht vernachlässigbaren Einfluss

auf die gesamte Spannungsverteilung im Betriebszustand haben.

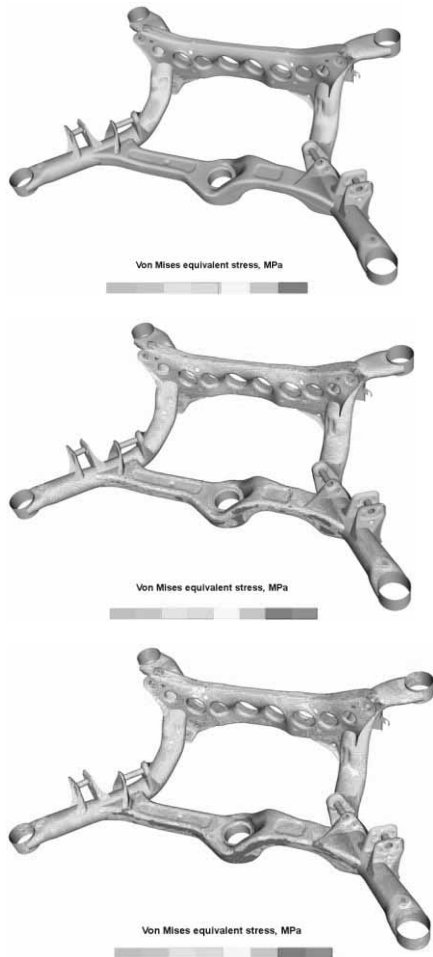


Abbildung 4. Von Mises Spannungen, MPa, nach einer statischen Berechnung eines Lastfalls: oben – ohne Ergebnisse der Umform- und der Schweißsimulation, Mitte – mit Ergebnissen der Umformsimulation und ohne Berücksichtigung der Schweißsimulation, unten – mit Berücksichtigung der Ergebnisse der Umform- und der Schweißsimulation [10].

In [10] wurde ein weiteres Beispiel einer Prozesskette, eine Lebensdaueranalyse mit und ohne Berücksichtigung der Fertigungsprozesse, diskutiert. Auch dabei konnte beobachtet werden, dass die Berücksichtigung der Fertigungsprozesse eine Schädigungszunahme im Bauteil hervorruft.

Ein weiterer Schritt besteht in der Validierung von Prozessketten. Hierfür müssen die Kriterien erarbeitet werden, nach welchen alle Schritte im Einzelnen und das Gesamtergebnis anhand von entsprechenden Messwerten validiert werden können. Hierfür sollten

mehrere unterschiedliche Realbauteile miteinbezogen werden.

5 Zusammenfassung

In diesem Beitrag wurden einige aktuelle Trends in der FE-Simulation von Fertigungsprozessen in der Automobilindustrie diskutiert. Während die Umformsimulation seit mehreren Jahren eine zuverlässige Unterstützung der Fertigungsplanung und einen festen Bestandteil des Planungsprozesses darstellt, steht die Schweißsimulation erst am Anfang ihrer Einbindung in die Planungsprozesse. Unterschiedliche Schweißsimulationstools sind heute auf dem Markt vorhanden. Nur einige von denen sind für die Aufgaben und Rahmenbedingungen der Fertigungsplanung geeignet (s. Kapitel 2). Das Softwaretool Weld Planner[®], welches im Kapitel 3 vorgestellt wurde, ist ein Beispiel einer auf die Anforderungen der Fertigungsplanung zugeschnittenen Software.

Die Anwendung dieses Tools wurde anhand von einigen Beispielen demonstriert. Dabei wurde gezeigt, dass die frühzeitige Abschätzung des Schweißverzugs die Fertigungsplanung bei der Auswahl einer optimalen Schweißfolge und Spannbedingungen sowie bei der Prozessauslegung unterstützen kann. Dabei kann die Fertigung der Einzelteile besser und prozesssicherer geplant und die Anzahl der üblichen Prototypenschweißungen bis auf ein Minimum verringert werden.

Ein weiterer Trend in der Simulation von Fertigungsprozessen besteht in der Verkettung von einzelnen Prozesssimulationen zu einer durchgängigen Simulationskette. Aufgrund einer Vielfalt an Simulationstools in diesem Bereich stellt dies eine Herausforderung dar. Bisher wurden nur einzelne Teilketten erfolgreich realisiert.

Für die Bereiche Karosserie und Fahrwerk ist die Berücksichtigung der Umform- und der Schweißprozesse innerhalb der Prozesskette notwendig. Im Kapitel 4 wurde darauf eingegangen, welche Auswirkung diese Fertigungsprozesse auf die Eigenspannungen in einem Bauteil im belasteten Zustand haben. Ohne die Berücksichtigung von Fertigungsprozessen wird das Spannungsniveau im Allgemeinen unterschätzt. Aus diesem Grund sollten im Rahmen eines Festigkeits- oder eines Lebensdauernachweises die Ergebnisse der vorhergehenden Umform- und Schweißsimulation berücksichtigt werden.

Bei den Blechbauteilen der Automobilindustrie ist heute die Vorhersage des Bauteilverhaltens und der -eigenschaften ohne Berücksichtigung von Fertigungsprozessen undenkbar. Für eine zuverlässige Vorhersage mittels Simulation ist eine Validierung von allen Schritten der Fertigungsprozesskette notwendig. Diese Validierung sollte anhand von realen Bauteilen und realen Prozess- und Werkstoffdaten durchgeführt werden. Auf diese Weise können sowohl die entsprechenden Modelle als auch die Methoden der Datenübertragung zwischen verschiedener Software validiert werden.

6 Referenzen

- [1] K. Roll. *Simulation der Blechumformung – neue Anforderungen und Tendenzen*. Tagungsband 12. Dresdner Werkzeugmaschinen-Fachseminar Simulation von Umformprozessen unter Einbeziehung der Maschinen- und Werkzeugeinflüsse, Dresden, 2007.
- [2] K. Roll, T. Lemke und K. Wiegand. *Possibilities and strategies for simulations and compensation of springback*. Im Tagungsband NUMISHEET 2005, Part A, Herausgeber: L.M. Smith et al., American Institute of Physics, USA, S. 295-302, 2005.
- [3] Weiher, J. *Die prozesskettenübergreifende Simulation – Motivation und Anforderungen*, Tagungsband Presswerk im Wandel der Anforderungen, Bad Nauheim, 2008.
- [4] M.F. Zäh und M. Langhorst. *Durchgängige Bauteil-Struktur-Simulation für Fertigungsprozessketten*, Im Tagungsband Integration von Umformen, Trennen und Fügen für die flexible Fertigung von leichten Tragwerkstrukturen, Herausgeber: A.E. Tekkaya, VDI Reihe Fertigungstechnik Band 668, Düsseldorf, S. 331-348, 2009.
- [5] T. Bär und S. Haasis. *Perspektiven für die Simulation*. In Simulation in der Automobilproduktion, Herausgeber: J. Bayer et al, Springer Verlag, Berlin, Deutschland, 2003.
- [6] D. Tikhomirov, J. Weiher, K. Roll, T. Franz und M. Kröger. *Fast welding distortion prediction for the production planning in automotive industry*. Tagungsband Große Schweißtechnische Tagung, DVS-Bericht 258, Düsseldorf, S. 90-94, 2009.
- [7] <http://www.inpro.de/>
- [8] <https://www.esi-group.com/>
- [9] D. Tikhomirov, G. Eßer und H.-W. Scholz. *Towards FE-simulation based production planning and development*. Production Engineering – Research and Development, S. 185-191, 2010.
- [10] D. Tikhomirov, H. Wessels, J. Weiher and G. Eßer. *Industrial challenges of virtual manufacturing process chains for sheet metal automotive components*, Tagungsband 1st Conference on Multiphysics Simulation. Advanced Methods for Industrial Engineering, Bonn, 2010.

Advancing Virtual Commissioning with Variant Handling

Johannes Möck¹, Jens Weiland¹

¹Hochschule Reutlingen, School of Engineering

Alteburgstr. 150

72762 Reutlingen, Germany

{johannes.moeck, jens.weiland}@reutlingen-university.de

Abstract: Nowadays the software development plays an important role in the entire value chain in production machine and plant engineering. An important component for rapid development of high quality software is the virtual commissioning. The real machine is described on the basis of simulation models. Therefore, the control software can be verified at an early stage using the simulation models. Since production machines are produced highly individual or in very small series, the challenge of virtual commissioning is to reduce the effort to the development of simulation models. Therefore, a systematic reuse of the simulation models and the control software for different variants of a machine is essential for an economic use. This necessarily requires a consideration of the variability which may occur between the production machines. This paper analyzes the question of how to systematically deal with the software-related variability in the context of virtual commissioning. For this purpose, first the characteristics of the virtual commissioning and variability handling are considered. Subsequently, the requirements to a so-called variant infrastructure for virtual commissioning are analyzed and possible solutions are discussed.

1 Introduction

Nowadays the software development plays an important role in the entire value chain in production machine and plant engineering. The low-cost and rapid development of high quality software has become a crucial success factor. The machine control software (NC, PLC, HMI), which has to be solved computationally, can no longer be seen as only an appendage of a machine. Special software engineering methods are necessary to master the complexity of the control software.

The virtual commissioning is an important step to shorten developmental times (time-to-market) and to improve the quality of the control software. In virtual commissioning the real machine is described based on virtual simulation models, including kinematic and behavioral models. Using this technique the control software can be verified on an early stage by means of the simulation models. In addition, simulation results can flow back into the machine design. Thus, virtual commissioning allows an early validation and optimization of the control software and the entire machine behavior. This leads to a significant reduction of commissioning time and to a considerably higher quality of the machine. In many places, the virtual commissioning is at the threshold to productive use.

Despite the great progress that has been made in recent years in the field of virtual commissioning, there are still aspects which counteract their economic productive use. An essential aspect is the effort of developing simulation models: Production machines and plants are often highly individually and produced in very small batches. As a result, the functionality of a machine has to be customized for the respective customer. In addition, from a technical point of view, the machine-specific sensors and actuators lead to an individual control of the machine. Functional and technical differences between the different machine variants are reflected inevitably in the control software and the simulation models of each machine variant.

Depending on the developmental phase and the scope of use, the developmental effort can be reduced if parts of the control software and the simulation models could be reused for different variants of the machine. However, a systematic reuse requires consideration of the variability that can occur between the production machines. As it has been used in mechanical and plant engineering for a long time now, a systematic handling of the variability is still lacking adequate concepts for the associated software and model-side variability. This concerns in particular dependencies between variability in the control software and the simulation models. In the following

sections, the question is discussed, how to deal systematically with the occurring variability between the simulated production machines in software within the virtual commissioning¹. Firstly, the paper gives a brief introduction into the Virtual Commissioning. In the following the systematic handling of variability in the context of virtual commissioning is discussed. Afterwards the different requirements for a so-called variant infrastructure within the virtual commissioning are mentioned and briefly explained. Finally a summary and an outlook about advancing the virtual commissioning with variant handling are given.

2 Virtual Commissioning (VC)

In contrast to the classical sequential development of machine and plant engineering, the virtual commissioning (VC) follows a different methodological approach. Parts of the commissioning are brought forward by using a virtual machine of the necessary system components. Thereby the software development can start early in the developmental process (Figure 1). Through the virtual commissioning the classic commissioning is not completely replaced, but it can be significantly shortened and simplified. By parallelizing the developmental steps and by utilizing the simulation feedback, VC significantly shortens developmental times, reduces developmental costs and improves product quality.

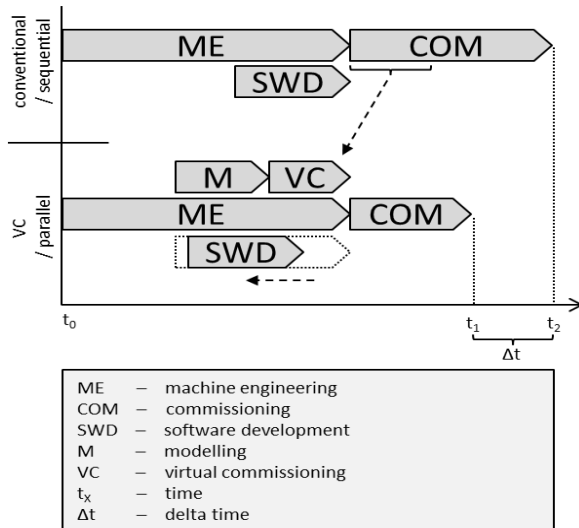


Figure 1: Concept of the Virtual Commissioning

¹ This work is funded by the Federal Ministry of Education and Research within the project Virtual Commissioning of Variant-Rich Systems (VivaSys) under the reference number 03FH085PX2.

Depending on the objectives of the VC, the components are displayed as simple as possible and as accurate as necessary in various domain specific simulation models. For example, a virtual machine tool is represented by kinematics and behavioral models. In this case the three-dimensional kinematics model shows the geometry, the kinematics and the collision calculation of the machine. Whereas the behavioral model describes the physical characteristics of the machine, such as the timing or the switching behavior. Simulation models can be connected to real control software via a (simulated or real) field bus system. The control software can be tested in this way at an early stage compared with the simulation model (Figure 2).

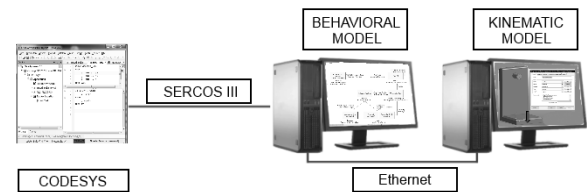


Figure 2: Integration of simulation models and control software

Usually there are different simulation tools for development and simulation of kinematics and behavioral models. Thus, for example the multibody simulation can be modeled in the simulation tool RecurDyn from FunctionBay [8] and the behavioural model can be modeled in SimulationX from ITI [9] or Simulink from The Mathworks [11]. The control software can be realized by the use of the developmental environment CODESYS [17] from 3S-Smart Software Solutions.

Simulation models, which have been developed in this way can be executed and synchronized at the same time as part of the co-simulation. Concepts that allow such coupling between simulation tools are, for example:

- The Functional Digital Mock-up (FunctionalDMU): An initiative of the Fraunhofer Gesellschaft with the aim to create a bridge between different simulations and visualization [3]. The runtime environment consists of a master simulator and simulators with appropriate wrappers that are connected to the master simulator interfaces, called slots.
- The Functional Mock-up Interface (FMI), which was developed in the context of MODELISAR

[12]. This concept was developed for the exchange of dynamic models, which enables the coupling of different simulation and modeling environments for cross-domain simulation [3].

These above concepts allow the exchange of information between simulation models during the simulation time. The focus of the following requirements is ensuring the consistency between simulation models before executing a possible simulation.

3 Handling of variability

Current simulation tools like RecurDyn, SimulationX or Simulink only support the modeling of individual systems and do not know concepts regarding variability. In industry, the current procedure in the context of VC is therefore that for each component individual simulation models have to be developed. Alternatively existing models are adapted by duplicating (so-called Clone and Own). In particular, the duplication can lead to increased maintenance costs and increased expenditure of time throughout the software life cycle. As a change in one version may lead to changes in all other copies.

An economic use of VC requires inevitably a systematic observation and handling of variability within and between simulation models and associated control software.

A suitable concept for the systematic handling of variability and for the efficient production of highly individualized systems represents the product line engineering ([4], [6]). Instead of developing individual systems, which are independent of each other, the focus is in product line engineering from the outset on developing a set of systems that are associated with a particular application domain. This development is mainly done in two parallel processes: The domain engineering and application engineering. During domain engineering, a product line infrastructure for the systematic reuse of software artifacts is developed. This includes, for example, the analysis of variable requirements between systems in terms of features as well as a cross-system reference architecture or reusable implementation components. During the application engineering these artifacts are the basis, in order to generate specific members of the product line.

The generative software development builds upon the product line engineering. Goal of generative software development is to generate automatically highly cus-

tomized and optimized systems from defined reusable components on the basis of a concrete system specification [7]. Core concept represents the generative domain model (Figure 3). This separates application-oriented concepts in the problem space from the concepts of the implementation in the solution space.

It allows separate development of domain concepts and reusable components and thus their individual modeling, implementation and evolution. Both models can be developed independently of each other in this way. The configuration knowledge maps the problem space to the solution space and represents the relationship between the two models explicitly.

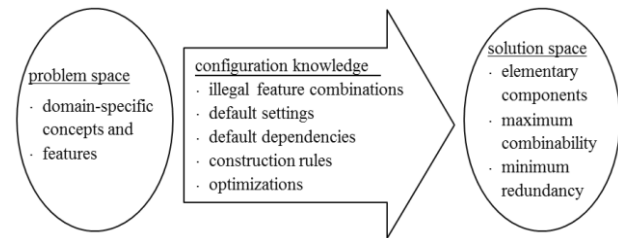


Figure 3: Elements of the generative domain model (see [7])

The product line engineering and the generative software development, thus, form a basis for the systematic handling of variability in the context of VC.

4 Requirements for variants infrastructure within the VC

Taken the product line engineering and generative software development as the basis for a software-based handling of variability, the following aspects have to be considered. These aspects are the essential requirements for a so-called variant infrastructure within the VC.

4.1 Development of feature models in the problem space

For a function-based approach of variant-rich embedded systems, feature models turned out to be successful ([2], [7], [5], [9]). In feature models variable requirements of a product line are managed and hierarchically structured in the form of features of the studied domain. The result is a comprehensive model of common and variable features and their dependencies between product variants. A distinction is made between mandatory and optional features, and (1..n):m-group relations that are realized by the feature types Mandatory, Optional, Alternative, and Or. Each fea-

ture can have a set of attribute-value pairs. Dependencies between features and attributes are defined by restrictions. For feature modeling, the tool pure::variants from the company pure-systems [13] can be used. Figure 4 shows, for example, the variable characteristics of a speed picker (using notation from [7]).

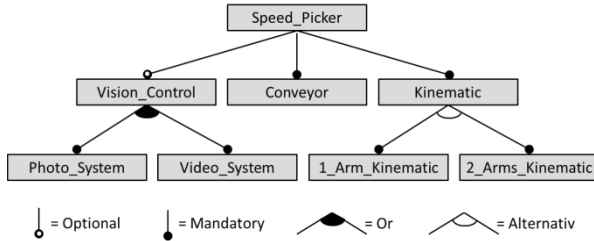


Figure 4: Extract of the speed picker feature diagram

In this feature tree, *Vision_Control* represents an optional feature, which could or could not be part of the *Speed_Picker* system. The *Kinematic* could be achieved by choosing one of two alternative features *1_Arm_Kinematic* or *2_Arms_Kinematics*. Finally, *Photo_System* and *Video_System* are Or-features. *Vision_Control* could be implemented either on the basis of using either one of them or both.

4.2 Development of reusable components in the solution space

In the context of VC, the solution space includes the control software and the simulation models. Therefore the question arises which aspects - such as technique, information, and procedure - have to be considered in the implementation of variability in control software and simulation models. In the following, essential aspects and the resulting requirements for the development tools are discussed in more detail.

a) Starting point for the implementation of variability in control software and simulation models is the variation point. This defines a separate, clearly identifiable area of the software or model, in which adjustments can be made for a specific system variant [1]. Variability is therefore clearly localized.

For the introduction of variability in control software and simulation models it is essential that the development tools contain language elements by which variation points can be realized. Here, the significant factor is, that these "model-specific" language elements differ from "regular" language elements:

- A variation point must be clearly visible for the developer for manual and graphical development.

Eg by using a variant-specific color or a graphical/textual label of the variation point.

- A variation point must be clearly identifiable for automated processing. Eg via an annotation as a unique identifier. This is necessary for automated configuration and communication with the control software and the simulation models.

In addition, the potential language elements should ensure a uniform handling for realizing variability within the considered development tool.

b) The variation point essentially consists of a variability mechanism and the possible variants aside from a unique identification. The variability mechanism determines how a variation point is removed and is replaced by an associated variant. Figure 5 shows an example of a variation point in SimulationX.

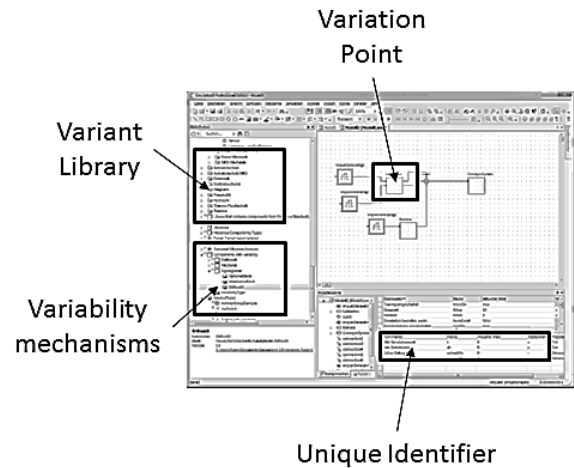


Figure 5: Variation Point in SimulationX

The development tools must provide mechanisms by which a variation point can be removed from the software or models in terms of configuration of a specific variant. Basically, the following mechanisms can be distinguished:

- During configuration a specific variant is selected from a set of predefined variants based on a specification. The variants are included in the control software and the behavioral models. Eg such variants could be selected by a signal routing.
- Within the substitution a variation point is replaced by a variant. The variation point specifies the condition under which this variant has to be inserted in the variation point. Such variants could be managed in the development

tool within a library or by a file system which is under a separate and external version control.

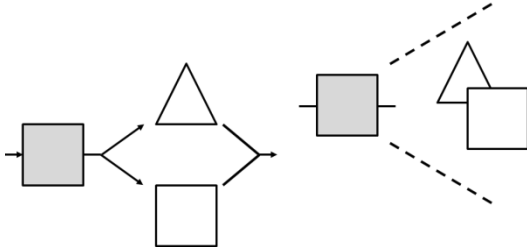


Figure 6: Variability mechanisms „selection“(left) and "substitution"(right)

The selection and substitution can be controlled via a specific set of variant configuration parameters. Meanwhile, a number of object-oriented development tools are offered for the modeling of control and simulation software. Therefore, the inheritance is another application for the selection and substitution of variants.

- The generation is based on a specification that eg may take the form of a blueprint. From this specification, the system variants will be generated by a generator.

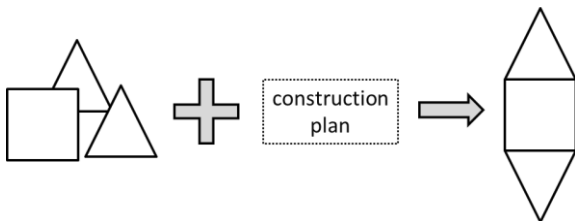


Figure 7: Variability mechanism "generation"

Not every concept is supported by each development tool. On the one hand, in the modeling of a variation point in a behavioral model, like SimulationX, it is possible to model all variants through signal routing, whereby a selection can be realized. On the other hand within the modeling of kinematic models, eg of multi-body simulation models in RecurDyn, only one variant may be part of the current model (see Figure 8). In this case, the variation point requires detailed information by which variants it may be replaced and where to find them.

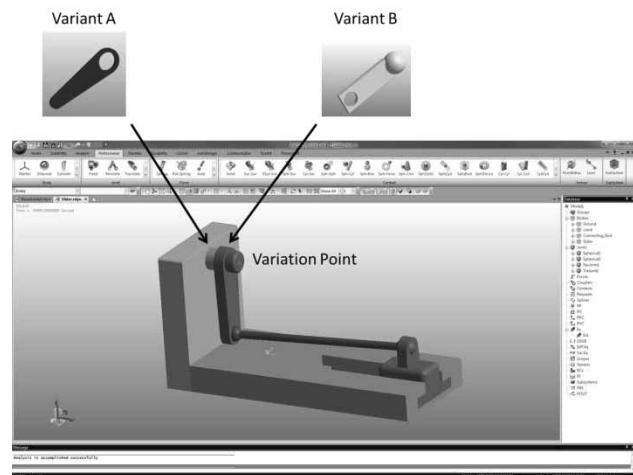


Figure 8: Modeling a variant-rich multibody model in RecurDyn

c) As part of the mapping of the problem space to the solution space, the features are mapped on variation points in the control software and the simulation models. Therefore, feature types *Optional*, *Alternative*, and *Or* have a strong influence on the realization of these variation points: Feature types have to be mapped to variability mechanisms. Experience shows that not every variability mechanism supports each feature type. Therefore, the tool side must provide potential variability mechanisms for the various feature types.

d) Depending on the size of the variant, that is associated with a variation point in the control software and the simulation model, different granularities of variability can be distinguished. The simplest form of variability is the so-called data variability. Variants describe the parameter values, which represent application-specific thresholds or characteristic curves. Another form of variability is the variant-specific signal routing within an application function. In the third form, variants consist of code blocks or model components that encapsulate variant-specific application functions. This enables the separate and possibly parallel development of functions. Such application functions may also be reused in a different context. In the development of control software by using CODESYS, a code block could be a class, a function, a function block or a set of parameters. For the relevant granularity the potential variability mechanisms has to be considered on the tool side.

4.3 Modeling of configuration knowledge

In the context of the VC the features from the feature model have to be mapped to variation points from the control software and simulation models by the configuration knowledge. For this mapping the knowledge about the features, the variation points in the control software and simulation models, as well as the dependencies between features and variation points (both among themselves and between problem and solution space) are required.

In the feature model dependencies between features can be specified as following: dependencies can depend on the position of a feature in the model hierarchy, the feature type or the special relations to other features. On the basis of the tool `pure::variants` it is possible to manage this knowledge.

With respect to the variability in the control software and simulation models various questions have to be answered:

- What comprises the knowledge to manage variability? Among others, this affects the information about the variation points, information about variants and information about the used variability mechanisms.
- Where to store this variability knowledge? Eg within the means of the development tools or outside the tools in a central repository.
- How to save the knowledge about the variability? This aspect relates to the structure of the data model and its implementation.

The mapping of features out of the feature model to variation points enables tracing of the distributed variation and automated configuration of the variant-rich control software and simulation models based on a feature selection. Here it is necessary to analyze the following aspects:

- What comprises the knowledge to manage dependencies? Eg the assignment of features to variation points of the control software and the simulation models, default settings, etc.
- Where and in what type the dependencies are to be saved? Eg within the respective development tools or in a central repository.

4.4 Synchronization using a variant manager

As part of the VC several software tools are working on the development of the control software and the

kinematics and behavioral models closely together. Knowledge of the variability and dependencies within and between the control software, the simulation models and the feature model must necessarily be synchronized for the management and configuration of variant-rich systems. This is the central task of the so-called *variant manager* (Figure 9).

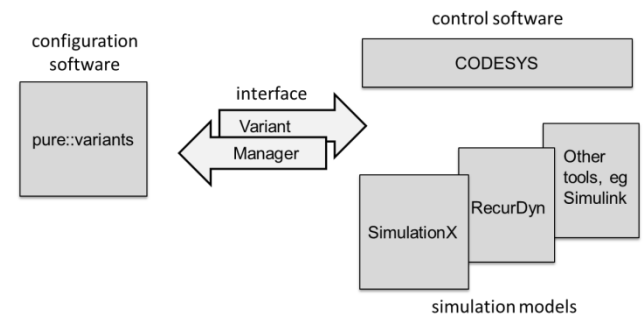


Figure 9: Variant infrastructure with variants manager

The variant manager provides the functionality to automatically share variability and dependency knowledge between different VC-development tools in the variant infrastructure. It ensures consistency of variability knowledge across the used tools. Due to different interfaces of the development tools it is up to the variant manager to provide an appropriate interface technology for the exchange of information. At the logical level the variant Manager must provide the following functionality to the development tools:

- To log on (and off) to the variant manager.
- To define the possible communication techniques at the technical level for integration into the infrastructure. This communication could be realized by the exchange of formatted files (XML, CSV, etc.), remote procedure calls, or an object broker.
- To consistently exchange the tool intrinsic variability knowledge on the basis of a defined set of operations.

Part of the development is to evaluate to what extent such a variant manager should be realized central in the form of an information broker or decentralized distributed in the VC-development tools.

5 Summary and Outlook

In the development of production machines, the virtual commissioning (VC) plays an increasingly important role. On the basis of virtual simulation mod-

els, the control software can be verified at a very early stage and can back flow simulation insights into the design of the machine. This leads to shorter development time and a higher product quality.

The challenge of the VC is to reduce the high costs of developing the simulation models. These costs can be significantly reduced by considering variability in the control software and the simulation models. A consideration of variability enables the systematic reuse of common parts of the model.

On the basis of product line engineering concepts, the requirements are analyzed for a variant infrastructure in this paper. This is the basis for a systematic presentation, management and configuration of variability in control software and simulation models. As part of the VC several software development tools work together. For a multi-tool synchronization of variability knowledge the requirements for a so-called variant manager are analyzed within the variant infrastructure.

The requirements have been determined as part of the BMBF-funded project Virtual Commissioning of Variant-Rich Systems (VivaSys). Currently a data model and the architecture for the variant infrastructure as well as a procedure for handling variability in the context of VC are developed.

6 References

- [1] Becker, M.: Anpassungsunterstützung in Software-Produktfamilien. Dissertation, Technische Universität Kaiserslautern, 2004.
- [2] Beuche, D.: Composition and Construction of Embedded Software Families. Dissertation, Universität Magdeburg, 2003.
- [3] Blochwitz, T.; Otter, M.; Arnold, M.; Bausch, C.; Clauß, C.; Elmqvist, H.; Junghanns, A.; Mauss, J.; Monteiro, M.; Neidhold, T.; Neumerkel, D.; Olsson, H.; Peetz, J.-V.; Wolf, S.: The Functional Mockup Interface for Tool independent Exchange of Simulation Models. 8th International Modelica Conference 2011, 20-22 March 2011, Dresden, Germany, 2011.
- [4] Böckle, G.; Knauber, P.; Pohl, K.; Schmid, K.: Software-Produktfamilien – Methoden, Einführung und Praxis. dPunkt Verlag, 2004.
- [5] Clauss, M.: Untersuchung der Modellierung von Variabilität in UML. Diplomarbeit. Technische Universität Dresden, 2001.
- [6] Clements, P.C.; Northrop, L.: Software Product Lines – Practices and Patterns. SEI Series in Software Engineering, Addison-Wesley, 2001.
- [7] Czarnecki, K.; Eisenecker, U.W.: Generative Programming – Methods, Tools, and Applications. Addison-Wesley, 2000.
- [8] FunctionBay GmbH: RecurDyn, www.functionbay.de/, 23.01.2014.
- [9] ITI Gesellschaft für ingenieurtechnische Informationsverarbeitung mbH: SimulationX, <http://www.simulationx.com/>, 23.01.2014.
- [10] Lee, K.; Kang, K.C.; Koh, E.; Chae, W.; Kim, B.; Choi, B.W.: Domain-Oriented Engineering of Elevator Control Software: A Product Line Practice. In: Donohoe, P. (Edit.): Software Product Lines – Experience and Research Directions. Kluwer Academic Publishers, 2000.
- [11] Mathworks: Simulink. www.mathworks.de/products/simulink/, 23.01.2014.
- [12] MODELISAR (07006): Functional Mock-up Interface for Model Exchange. Document version 1.0, 2010.
- [13] pure-systems GmbH: pure::variants Eclipse Plugin User Guide. 2012.
- [14] Relovski, B.; Neumerkel, D.; Kleiner, N.; Welakwe, N.-A. S.: The MODELISAR Project and the Functional Mockup Interface. 1st Int. Conference on Multiphysics Simulation – Advanced Methods for Industrial Engineering, 22.-23. June 2010, Bonn.
- [15] Wenk, M.: Virtuelle Inbetriebnahme. www.oth-aw.de/wenk/forschung/virtuelle_inbetriebnahme/, 03.01.2014.
- [16] Wünsch, G.: Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme. Herbert Utz Verlag, 2007.
- [17] 3S-Smart Software Solutions GmbH: CODESYS, <http://www.codesys.com/>, 03.01.2014.

Anforderungen an State Event Charakterisierungen für die Simulation

Andreas Körner¹, Horst Ecker², Felix Breiteneker¹

¹Institute für Analysis und Scientific Computing, Technische Universität Wien

²Institut für Mechanik und Mechatronik, Technische Universität Wien

andreas.koerner@tuwien.ac.at

State Events eröffnen einen Ansatz der effizienten Beschreibung von mathematischen Modellen. Dieser Modellierungszugang ermöglicht es, verschiedene Modelle, die jedes für sich betrachtet mit minimalem Zustandsraum sowie Gleichungsstrukturen auskommen, miteinander zu verbinden. Für den Übergang von einem Modellteil zu anderen sind sogenannte State Events verantwortlich für deren Beschreibung mathematische Formulierung vorliegen. Da die verwendeten Modelle jedoch an die zugrundeliegende Aufgabenstellung des technischen Systems oder des naturwissenschaftlichen Prozesses angepasst sind, ist es hier interessant zu untersuchen, ob die entsprechenden Beschreibungsformen das Auffinden von State Events effizient gestalten lässt. Hierfür werden im vorliegenden Beitrag einige Benchmarks betrachtet, die hinsichtlich dieser effizienten Formulierung einige Hinweise bieten. Des Weiteren wird eine Klassifikation von State Events motiviert, um in diesen Klassen eine an das Modell und den Modellübergang angepasste Betrachtung der Events zu ermöglichen.

1 Einleitung und Motivation

Im Allgemeinen kann ein mathematisches Modell eines beliebigen Systems oder Prozesses durch die Gleichung

$$F(x, \dot{x}, w) = 0 \quad (1)$$

beschrieben werden. Dabei stellt x den Zustandsvektor, \dot{x} dessen Ableitung sowie w den Vektor der externen Variablen dar. Im Zusammenhang mit State Events ist der Grundgedanke jener, dass es mehrere Systemteile gibt, welche für sich betrachtet i.A. mit einem kleineren Satz an Gleichungen auskommen als in (1) benötigt wird. Daraus resultieren die Gleichungen

$$F_{l_i}(x, \dot{x}, w) = 0, \quad (2)$$

für je einen Zustand l_i , $i = 1, 2, \dots, k$. Das determinierte Umschalten zwischen den Zuständen wird durch eine Übergangsrelation a_i verwaltet, die sogenannte State Event Transition. Dieses System kann mit Hilfe eines hybriden Automaten, wie in Abbildung 1 dargestellt, visualisiert werden.

In Abbildung 1 beschreibt Inv die Zuordnung eines Zustandes l_i zu einem zugehörigen stetig differenzierbaren Zustandsvektor x .

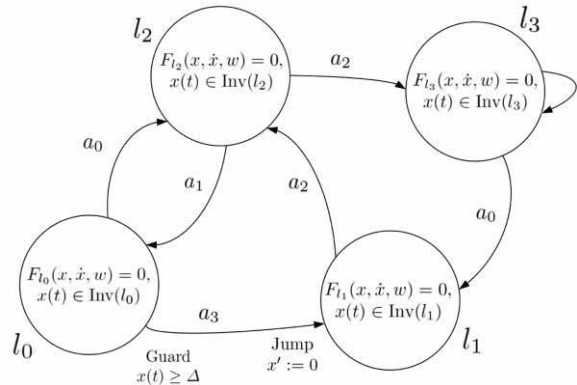


Abbildung 1. Beispiel eines hybriden Automaten

Jeder Übergang eines Zustandes in einen weiteren wird durch eine Guard und eine Jump Funktion begleitet. Eine vereinfachte Beschreibung zum Auffinden eines State Events ist durch die Nullstellen der sogenannten Eventfunktion h gegeben. Hierbei wird durch das Aufsuchen der Nullstelle aus der Gleichung

$$h(x, \dot{x}, w) = 0 \quad (3)$$

der Zeitpunkt des Events bestimmt bzw. numerisch iteriert. An diesen Prozess angeschlossen ist die sogenannte Event Action

$$E(x, \dot{x}, w) = 0, \quad (4)$$

welche Veränderungen in der Systembeschreibung nach sich ziehen. Eine detaillierte Auflistung und

Erläuterung dieser Fragestellung ist in [1] nachzulesen.

2 Charakterisierung und Klassifikation

Für die Simulation ist die Behandlung von State Events durch das in [2] angesprochene Schema gegeben:

- Detektierung des Events
- Lokalisierung des Events
- Event Action
- Neustart der Simulation

Die grundlegende Idee ist die Einteilung von State Events nach der folgenden Charakterisierung:

- Änderung des Ausgangs
- Änderung von Parametern
- Änderungen des Eingangs
- Änderungen des Wertes des Zustandsvektors
- Änderungen des Wertes der Ableitung des Zustandsvektors
- Änderungen des gesamten Modells

Eine detaillierte Beschreibung der einzelnen Klassen würde für eine Gruppe von oft auftretenden State Events eine Strukturierung und daher bessere Ausbildung für die Anforderungen der Simulation ergeben. Der folgende Abschnitt soll durch Benchmarks die unterschiedlichen Anforderungen verschiedener Modelle verdeutlichen

3 Benchmarks

3.1 Hybrid Electrical Circuit

In Abbildung 2 ist die Schaltung eines Schwingkreises mit einer Diode dargestellt, welche ein hybrides Szenario zur Folge hat. Ein Zustand wirkt sich auf den anderen aus bzw. zieht eine Änderung desselben nach sich.

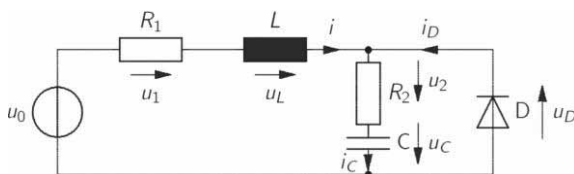


Abbildung 2. Hybrider elektrischer Schwingkreis

3.2 Rotierendes Pendel

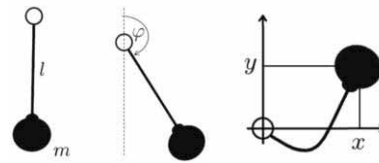


Abbildung 3. Rotierendes Pendel

In Abbildung 3 ist die Prinzipskizze eines rotierenden Pendels angeführt. Hier besteht das Event darin, dass das Modell zwischen einer Masse an einem gespannten Faden und einer Masse im (Erd-) Schwerfeld umschaltet. Dieses Beispiel hat noch als Zusatz, dass die Beschreibung des Modells einmal in Kreiskoordinaten und einmal in kartesischen koordinaten erfolgt, bei der Umschaltung ist also auch eine Koordinatentransformation erforderlich ist.

4 Diskussion

Eine Klassifikation von Zustandsereignissen und deren Behandlung bedarf einer sauberen mathematischen Definition der einzelnen Ereignisklassen, welche hinsichtlich der daran anschließenden Simulation zu erfolgen hat. Die simplen einführenden Aufgaben der einzelnen Benchmarks zeigen, dass die Einteilung und gesonderte Behandlung effizientere Simulationen nach sich zieht. Eine Liste von Benchmarks, welche einige Fragestellungen in diesem Bereich aufwerfen oder helfen diese zu bearbeiten, findet sich auf [3]. Diese werden als Ausgangspunkt für weitere Betrachtungen herangezogen.

5 References

- [1] Körner A., et. Al, About an alternative Method of numerical Iteration for State Event Finding and Handling in System Simulation of Hybrid Dynamical Systems, UKSIM 15th International Conference on Modelling and Simulation 2013, IEEE, Cambridge
- [2] Breitenacker F. et. al, Change of independent Variables for State Event Detection in System Simulation, I3M 9th International Multidisciplinary Modeling & Simulation Multiconference, 2012, Vienna
- [3] ARGESIM Benchmark-List 2014 – www.argesim.org

Methodenvergleich in der Simulation von Grundwasserverschmutzung

Stefanie Winkler¹, Martin Bicher², Felix Breiteneker¹

¹Technische Universität Wien, Institut für Analysis und Scientific Computing

²dwh GmbH Simulation Services, Wien

stefanie.winkler@tuwien.ac.at

In dieser Arbeit werden zwei Verfahren bzw. Methoden der Simulation der Diffusionsgleichung anhand eines einfachen Beispiels beschrieben. Dabei betrachte man eine beschränkte Wasserfläche, gegeben durch ein Rechteck, auf der sich Schmutzpartikel von einer Quelle aus ausbreiten. Diese Ausbreitung wird durch die Diffusionsgleichung beschrieben und im folgenden durch eine approximierte Lösung und dem Differenzenverfahren simuliert.

1 Einleitung

Die Grundwasserverschmutzung dient hier als anschauliches Beispiel bzw. als Motivation für die Diffusionsgleichung. Das folgende Beispiel soll eine vereinfachte Anwendung dieser mathematischen Gleichung zeigen. Hierbei kann man eine rechteckige Fläche betrachten. In dieser ist an einer bestimmten Stelle eine Schmutzquelle vorhanden. Aus dieser Schmutzquelle strömt pro Zeiteinheit, beispielsweise pro Minute, eine bestimmte Menge an Schmutzpartikel. Die Diffusionsgleichung beschreibt dann im Folgenden wie sich diese Schmutzpartikel von dieser Quelle aus auf der gesamten Fläche verteilen. In diesem speziellen Anwendungsbeispiel wird auch noch von einer gewissen Strömung in x-Richtung ausgegangen. In diesem Abstract werden verschiedene Methoden zur Simulation der Grundwasserverschmutzung beschrieben.

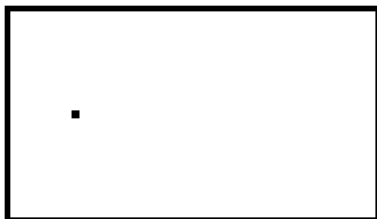


Abbildung 1. Vereinfachte Darstellung des Beispiels der Grundwasserverschmutzung (40LE hoch und 70LE breit). [2]

2 Methoden

Die allgemeine Diffusionsgleichung in zwei Dimensionen [1] ist gegeben durch

$$\frac{\partial c}{\partial t} = D \cdot \Delta c. \quad (1)$$

Diese Gleichung kann man nun auf verschiedene Arten lösen. Die naheliegendste Lösung wäre eine analytische Lösung. In der Simulation bedient man sich aber auch oft numerischer Verfahren zur Lösung von Differentialgleichungen. Für dieses Anwendungsbeispiel ziehen wir bestimmte Parameter in Betracht. Daher lässt sich die spezielle Diffusionsgleichung für diesen Fall schreiben als:

$$\frac{\partial c}{\partial t} \cdot \left(1 - \frac{u}{R}\right) = \frac{\alpha u}{R} \cdot \Delta c. \quad (2)$$

Diese Gleichung lässt sich natürlich auch umformen in die Form (1) wobei der Faktor D durch

$$D = \frac{\alpha u}{R-u} \quad (3)$$

gegeben wäre.

2.1 Approximierte Lösung

Unter Einbeziehung der Parameter aus Tabelle 1 kann man, beschränkt auf die endliche Fläche aus Abbildung 1, die folgende approximierte Lösung der Gleichung betrachten [2]:

$$c(x, y, t) = \frac{C_0}{4\sqrt{\alpha\pi t}} e^{\frac{x-r}{2\alpha}} \operatorname{erfc}\left(\frac{r-ut}{\sqrt{2\alpha ut}}\right). \quad (4)$$

Hierbei ist r und C_0 gegeben durch $r = \sqrt{x^2 + y^2}$ und $C_0 = \frac{M}{h n_e u}$. Die Funktion kann man, nur für ein beliebig feines Gitter, an jeder Stelle in Abbildung 1 auswerten.

Geschwindigkeit in x-Richtung	u	$10^{-5} \frac{m}{s}$
Dispersionslänge	$\alpha_T = \alpha_L$	0,05 m
Abbaufaktor	λ	0
Verzögerungsfaktor	R	1
Dicke des gesättigten Flusses	h	10 m
Effektives poröses Volumen	n_e	0,25
Inputrate der Verschmutzung	M	$2 \frac{mg}{s}$

Tabelle 1. Tabelle der verwendeten Parameter des Fallbeispiels.

2.2 Differenzenverfahren

Eine weitere Methode bietet die Anwendung des zentralen Differenzenverfahrens. In diesem Fall approximiert man die Ableitung einer Funktion an der Stelle x_i mit Hilfe der Funktionswerte an x_{i-1} und x_{i+1} wie folgt:

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1}))}{2h} \quad (5)$$

wobei $h = x_i - x_{i-1}$ gilt. In der Diffusionsgleichung benötigt man allerdings die 2. Ableitung. Diese erhält man durch erneute Anwendung des Differenzenverfahrens auf die 1. Ableitung. Mit diesem Verfahren ist eine numerische Lösung der PDE (2) möglich.

$$\frac{\partial c}{\partial t} \cdot \left(1 - \frac{u}{R}\right) = \frac{\alpha u}{R} \cdot \left(\frac{c_{x-1,y} + c_{x+1,y} + c_{x,y-1} + c_{x,y+1} - 4c_{x,y}}{h^2}\right) \quad (6)$$

Zur Lösung dieser Differentialgleichung benötigt man aber ein weiteres numerisches Verfahren. In diesem Fall wurde das Eulerverfahren eingesetzt. In diesem Verfahren wird mit Hilfe des bekannten Zustands und deren zeitlicher Änderung der neue Zustand berechnet. In diesem Fall lässt sich das durch folgende Gleichung beschreiben

$$c(t + \Delta t) = c(t) + \dot{c}(t) \cdot \Delta t \quad (7)$$

Wobei $c(t)$ den bekannten Zustand und $c(t + \Delta t)$ den neuen Zustand beschreibt.

3 Ergebnisse

Die folgende Abbildung 2 zeigt uns das Ergebnis der approximierten Lösung nach 100 Tagen. Die Feinheit des Gitters wurde dabei auf $h = \frac{1}{30}$ festgesetzt.

Den Einfluss der Strömung in x-Richtung erkennt man sehr gut. Der dunkle Bereich zeigt dabei den sauberen Teil des Wassers an. Je heller die Stellen des Wassers sind, desto größer ist dort die Verschmutzung.

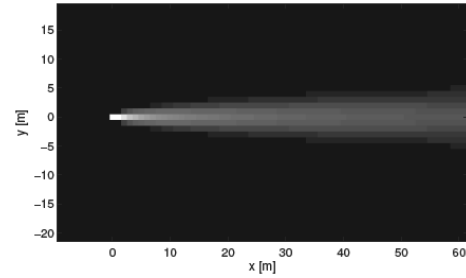


Abbildung 2. Simulationsdurchlauf der approximierten Lösung nach 150 Tagen.

Abbildung 3 zeigt einen Durchlauf der Simulation mit dem Differenzenverfahren. Man kann sehr gut erkennen, dass sich die Ausbreitung des Schmutzes ähnlich verhält.

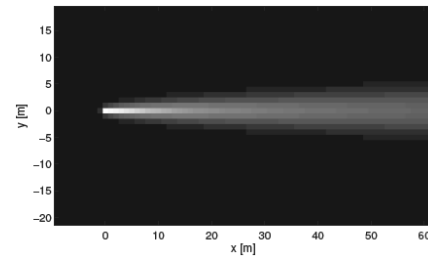


Abbildung 3. Simulationsdurchlauf der Differenzenmethode nach 150 Tagen.

4 Ausblick

In dieser Arbeit wurden bisher nur zwei verschiedene Simulationsmethoden in Betracht gezogen. Im weiteren sollen noch andere Verfahren bzw. Simulationsumgebungen eingebunden werden. Es gibt verschiedene Ansätze die Diffusionsgleichung numerisch aber auch analytisch zu lösen. Die Verwendung von PDE-basierender Software könnte noch einen weiteren Blickwinkel eröffnen.

Eine weitere Aufgabe besteht darin, nicht nur die Verschmutzung des Grundwassers, sondern auch die Auswirkungen von Reinigungsmaßnahmen, wie zum Beispiel einer Pumpe, zu untersuchen.

5 References

- [1] J.Cran. *The Mathematics of Diffusion*. Oxford University Press, United States, 1975.
- [2] ARGESIM. *Benchmark-List 2014*. www.argesim.org

Ansätze der Modellierung einer induktiven Übertragungsstrecke mit dem Simulationstool OctaveFEMM

Thomas Lang

Robert Bosch GmbH, Tübinger Str. 123, 72762 Reutlingen

Thomas.Lang3@de.bosch.com

Die Auslegung induktiver Leistungsübertragungssysteme für die kabellose Ladung von Elektrofahrzeugen stellt eine Herausforderung dar. Am Beispiel eines Übertragungssystems kleiner Dimension wird gezeigt, welche Unterstützung die Simulation bei der Auslegung einer magnetischen Übertragungsstrecke leisten kann. Für diesen Zweck wurden Modelle von Sende- und Empfängerspulen erstellt und deren elektrisches und magnetisches Verhalten simuliert. Die Simulationsergebnisse wurden mit Messungen der magnetischen Flussdichte und elektrischer Kenngrößen an einem mit Hardware aufgebauten Übertragungssystem verglichen.

1 Einleitung

Unter den Ladetechniken für elektrisch angetriebene Fahrzeuge gewinnen induktive Energieübertragungsverfahren zunehmend an Bedeutung. Vorteile solcher Systeme sind beispielsweise der Komfort in der Nutzung, da hier der Verwender keine Kabelverbindung zwischen netzgebundener Energiequelle und Fahrzeug herstellen muß.

Induktive Übertragungssysteme sind bei Kleingeräten wie in elektrisch betriebenen Zahnbürsten bereits täglich im Gebrauch und nicht mehr wegzudenken. Für mobile Telefone sind solche Systeme auch bereits auf dem Markt erhältlich. Dabei befinden sich im mobilen Gerät eine induktive Empfängerspule und eine in Leiterplattenbauweise hergestellte Empfänger-elektronik. Die zugehörige Sendespule und die Senderelektronik sind in eine Ladematte integriert welche meist über ein Netzteil mit dem Stromnetz verbunden wird. Auf diese Ladematte wird das Gerät zur Ladung aufgelegt, die Ladeelektronik erkennt das Gerät und steuert die Energieübertragung.

Aufgrund der Bauart dieser Ladesysteme für Kleingeräte beträgt die Übertragungsstrecke nur wenige Millimeter. Für die genannten Übertragungssysteme sind auch entsprechende Standardisierungen in Arbeit [1].

Eine besondere Herausforderung bilden Übertragungsstrecken bei denen die Energie über mehrere Zentimeter übertragen werden muß. Dies ist der Fall bei elektrisch angetriebenen Fahrzeugen. Aufgrund des Einbauortes der Empfängerseite im Chassis von Elektrofahrzeugen ergibt sich dieser größere Abstand

zur Sendeinheit die üblicherweise im Boden oder eingelassen ist.

Die Dimensionierung von Sende- und Empfängerspulen stellt für eine solche Anordnung eine wesentliche Schwierigkeit dar. Zumindest die magnetische Kopplung zwischen den Spulen stark von deren vertikaler Distanz abhängig ist und Werte einnimmt, die wesentlich kleiner sind als beispielsweise die eines Transformators.

An einem Beispiel wird die Dimensionierung einer magnetischen Übertragungsstrecke anhand der Finite-Elemente Simulation gezeigt und mit Messungen an einem realen Prototypen verglichen.

1.1 Grundlagen

Wird an eine Sendespule TX ein elektrisches Wechselfeld angelegt, so entsteht in dieser Spule ein magnetischer Fluss Φ_{11} . Dieser magnetische Fluss durchsetzt die Empfängerspule RX in welcher nach dem Induktionsgesetz eine Spannung induziert wird. Da diese Spulen jedoch räumlich getrennt sind, durchdringt jedoch nur ein Teil Φ_{21} des von der ersten Spule erzeugten magnetischen Flusses auch die Empfängerspule. Aus dem Verhältnis dieser Größen wird die Kopplung wie folgt definiert.

$$k_{21} = \frac{\Phi_{21}}{\Phi_{11}} \quad (1)$$

Umgekehrt bei Beaufschlagung der zweiten Spule mit einem elektrischen Feld ergibt sich die Kopplung zu

$$k_{12} = \frac{\Phi_{12}}{\Phi_{22}} \quad (2)$$

Der Gesamtkopplungsfaktor ist der geometrischen Mittelwert der Einzelkopplungsfaktoren k_{21} und k_{12} .

$$k = \sqrt{k_{12} \cdot k_{21}} \quad (3)$$

Der Kopplungsfaktor kann Werte zwischen 0 und 1 annehmen. Bei der kontaktlosen Energieübertragung sind je nach Abstand Werte zwischen 0,3 bis 0,8 zu erwarten [4].

Das Vorhandensein eines weiteren Stromkreises erzeugt jedoch auch eine Gegeninduktivität M . Für die Gegeninduktivität von Sende- zur Empfängerseite und umgekehrt wird analog zum Kopplungsfaktor auch wieder der Mittelwert gebildet. Die Zusammenhänge beschreibt die folgende Gleichung

$$M = k \sqrt{L_1 \cdot L_2} \quad (4)$$

Zur Berechnung der Gegeninduktivität mittels der Simulation wird noch der magnetische Fluss Ψ durch die Spulen benötigt.

Der magnetische Fluss durch eine Einzelspule ist proportional zur Induktivität, der Windungszahl n , und dem Strom I durch die Spule.

$$\Psi = L \cdot I = n \cdot \Phi \quad (5)$$

Die Induktivität L einer Spule lässt sich durch Umformung von Gleichung (5) berechnen

$$L = n \cdot \frac{\Phi}{I} \quad (6)$$

In der Simulation mittels FEMM lässt sich für eine gewählte Anordnung zweier rotationssymmetrischer Spulen der verkettete Fluss für jede Spule berechnen. Bei der Bestimmung des verketteten Flusses ist die Gegeninduktivität mit einzubeziehen. Es ergibt sich der nachfolgende Zusammenhang.

$$\Psi_1 = L_1 \cdot I_1 + M \cdot I_2 \quad (7)$$

Durch geeignetes Einsetzen der Spulenströme I_1 und I_2 in das Simulationsmodell wird der verkettete Fluss Ψ_1 berechnet.

In gleicher Weise werden der verkettete Fluss und die Induktivität der zweiten Spule ermittelt.

Die Kopplung k ergibt sich durch Umformung von Gleichung (4) und einsetzen von L_1 und L_2 .

1.2 Aufbau des Simulationsmodells

Die gewünschte Spulengeometrie und die Übertragungsstrecke wurden in das Simulationsmodell übernommen. Die Betriebsfrequenz der Anordnung lag zwischen 30kHz und 200 kHz. Deshalb wurde als Leitermaterial Kupferlitze verwendet um die ohmschen Widerstände der Spulen klein zu halten. Zur Führung des magnetischen Flusses wurden Ferritmaterialien eingesetzt.

Abbildung 1 zeigt schematisch die verwendete Geometrie.

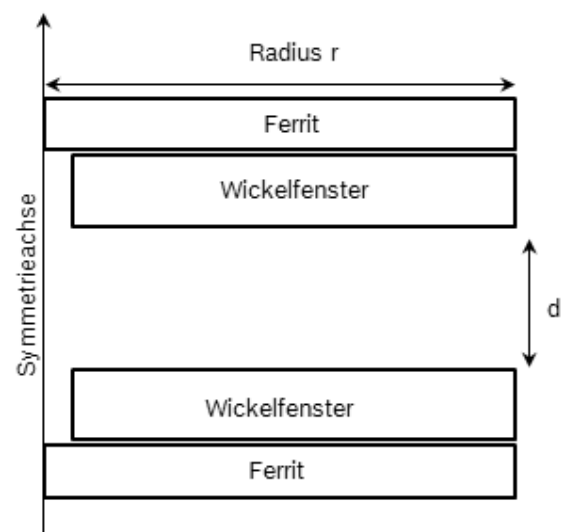


Abbildung 1. Geometrie (Halbmesser) und Anordnung der beiden Spulen.

Bei der Erstellung der Modelle wurde wie folgt vorgegangen. Es wurde der Programmteil für „magnetostatische Probleme“ ausgewählt. Als nächstes wurde im FEMM Editor der „Problemtyp“ als achsensymmetrisch definiert und eine Arbeits-Frequenz eingegeben.

Die gezeigte Spulengeometrie und die Abstände der Spulen d [mm] wurde dann in die Eingabemaske von

FEMM übertragen. Anschließend wurden die Systemgrenzen für den Lösungsraum definiert. Für die Eigenschaft des Systemrands („boundary properties“) wurden die Standardeinstellungen verwendet. Nun wurden die verwendeten Materialeigenschaften definiert. Das Ferritmaterial war ein Weichferrit. Als Umgebungsmedium wurde Luft gewählt. Danach wurden die Materialeigenschaften der verwendeten Litzen eingegeben, wie beispielsweise die Anzahl und der Durchmesser der Einzeladern.

Im Anschluss daran wurden die Sende- und Empfängerspule parameterisiert indem im Wickelfenster entsprechende Block-Eigenschaften definiert wurden. Die Windungszahl der Spulen wurde durch die äußeren Abmessungen, sowie durch den Gesamtdurchmesser der Litze vorgegeben.

Schließlich wurden die Ferrit-Eigenschaften als „Block“ dem entsprechenden geometrischen Modell zugewiesen [3].

1.3 Durchführung der Simulationen

Mittels einer ersten Reihe von Simulationen wurden die Werte für L_1 , L_2 und der Gegeninduktivität M ermittelt. Die Selbstinduktivität der gezeigten Spulen von war annähernd gleich, da Sende- und Empfängerspule ungefähr die gleiche Anzahl von Windungen hatten.

Mit einem in Matlab geschriebenen Programm wurde anschließend der Kopplungsfaktor k durch Umformung von Gleichung (4) ermittelt. In dieses Programm wurde als Parameter die zu erzielende Leistung P sowie die erwünschten Strom- und Spannungswerte auf Empfängerseite nach der Gleichrichtung des Sinussignals vorgegeben. Anhand dieser Daten errechnet das Programm die resultierenden sinusförmigen Ströme im Zwischenkreis, d.h. den Strom in Spule TX und den Strom in Spule RX. Diese errechneten Ströme werden nun in das FEMM Modell der Übertragungsstrecke eingetragen. Dazu wird für die Spule TX der errechnete Strom in das Feld „properties circuit“ → „circuit current“ eingegeben. In gleicher Weise wird für die Spule RX vorgegangen.

Im Simulationsprogramm wird zunächst ein Netz mit dem Maschengenerator erstellt. Sollten bestimmte Eigenschaften der Materialien in den einzelnen Bereichen nicht zugewiesen sein erscheint einen Fehlermeldung und erlaubt die entsprechende Korrektur.

Ist das Netz erstellt kann die Simulation gestartet werden. Als Parameter wurde die Solver Genauigkeit

von $1E-8$ verwendet sowie der Lösungsalgorithmus „succ. approx“. Nach Ablauf der Simulation stellt das Programm nun die Feldlinien und der Verlauf der Flussdichte „field density plot“ dar. Die Abbildung 2 zeigt ein typisches Ergebnis.

Die Feldlinien durchdringen beide Spulen. Es zeigt sich, daß die größte Flussdichte in den Ferriten sowie im Luftspalt im Zentrum der Spulen herrscht. Einige Feldlinien verlaufen weit außerhalb der Spulen. Diese tragen zur Streuinduktivität bei.

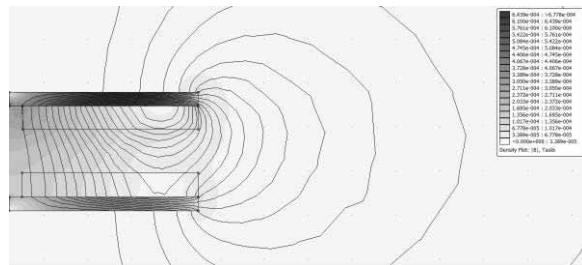


Abbildung 2. Darstellung der Feldliniendichte (dunkelgrau: Bereiche hoher Dichte; hellgrau: Bereiche geringer Dichte)

Um aus der Simulation Daten zu erhalten, die mit realen Messungen an Spulen vergleichbar sind wurden Flussdichteprofile entlang des Spulendurchmessers von Innen nach Außen mittels der Auswerteoption „contour plot“ erzeugt. Diese Profillinien wurden in z-Richtung in Höhe des halben Abstands der Luftstrecke d gezogen. Beispielsweise bei einem Spulenabstand $d = 1\text{ cm}$ wurde die Flussdichte in einem Vertikalen Abstand von 5 mm ermittelt.

Abbildung 3 zeigt ein typisches Profil der zugrunde liegenden Spulenordnung.

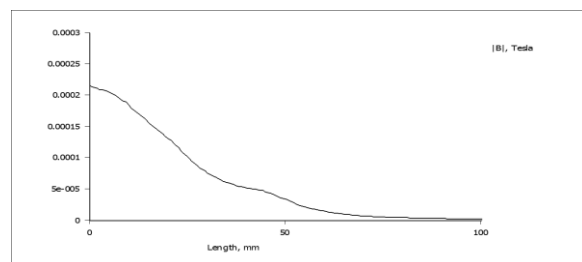


Abbildung 3. Flussdichte entlang der ausgewählten radialen Profillinie.

1.4 Simulationsergebnisse

Für verschiedene vertikale Abstände der Spulen d wurden nun in FEMM die im Matlab-Programm

errechneten Spulenströme eingegeben und in der Simulation die Flussdichten berechnet. Tabelle 1 zeigt die maximalen Werte von B im Luftspalt im Spulenabstand $d/2$ und beim Radius $r=0$; d.h. in der Spulenmitte.

Abstand d [mm]	Ausgangsleistung P [W]	Kopplungsfaktoren k -	Bmax @d/2;r=0 [μT]
10	23	0,61	480
20	22	0,37	303
30	24	0,23	232

Tabelle 1. In der Simulation berechnete Flussdichten Bmax und berechnete Kopplungsfaktoren k als Funktion des Abstands d.

1.5 Aufbau der Hardware und Messungen der Flussdichte

Für den späteren Vergleich zwischen Messung und Simulation wurde ein Prototyp gefertigt.

Die zuvor beschriebenen Spulen wurden mittels Litze auf den Ferritkörper gewickelt und anschließend verklebt. Abbildung 4 zeigt das Beispiel einer für die Messung verwendeten Induktionsspule.

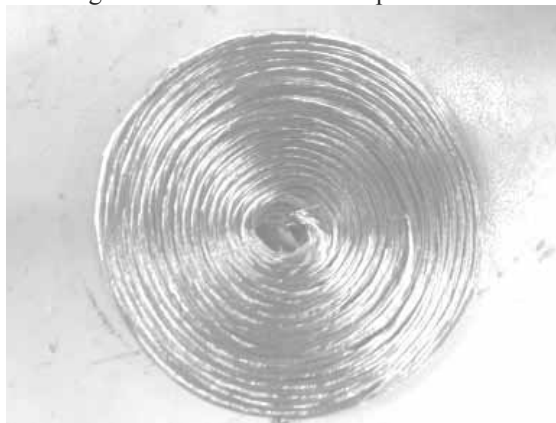


Abbildung 4. Typische für die Messung verwendete Induktionsspule.

Der Betrieb der Spulen, erfolgte an einem durch eine Spannungsquelle (0-100V) gespeisten Resonanzwandler mit einstellbarer Frequenz (30-200 kHz) und mit serieller-serieller Blindleistungskompensation. Sekundärseitig wurde ein Zweiweg-Gleichrichter verwendet, als Last diente eine Stromsenke mit konstanten Widerstandswert.

Die Primärspule wurde direkt am Resonanzwandlerausgang betrieben. Mittels Abstandshalter aus

nichtmagnetischem Material wurde ein definierter Abstand von Primär- zur Sekundärspule geschaffen, der variiert wurde. Die Sekundärspule wurde mit dem Gleichrichter verbunden. Am Ausgang des Gleichrichters befand sich die elektronische Last. Es war im Versuch wichtig die Spulen genau übereinander zu zentrieren. Zur Einstellung des Resonanzpunktes wurden jeweils zuvor errechnete Kondensatoren auf TX- und RX-Seite eingebaut. Die Ausgangsspannung des Systems konnte über die Wahl der Eingangsspannung des Wandlers variiert werden.

Im Betrieb der Anordnung, d.h. bei laufender induktiver Energieübertragung wurde die Flussdichte mittels eines Teslameters mit Transversalsonde im Luftspalt gemessen. Zunächst wurde die Sonde kalibriert, so daß sie bei nicht vorhandenem AC-Magnetfeld den Wert 0 anzeigte. Zu Messung wurde die Sonde in definiertem Vertikalen Abstand von $d/2$ in den Spalt gebracht. Es wurde so vorgegangen, daß die Sonde zunächst in einem radialen Abstand von 220mm platziert wurde und anschließend langsam nach Innen geschoben wurde. Dies hatte den Vorteil, daß die Sonde von Bereichen niedriger Flussdichte in Bereiche hoher Flussdichte verschoben wurde, damit war sichergestellt, daß sich eine mögliche Vormagnetisierung der Sonde nicht auf die Messergebnisse auswirkte.

1.6 Validierung des Modells mit Messergebnissen

Für alle simulierten Übertragungsstrecken wurde auch das B-Feld während des Betriebs des Prototypen gemessen. Abbildung 5. zeigt ein typisches Diagramm welches die erzielten Simulationsergebnisse und die Messwerte gegenübergestellt.

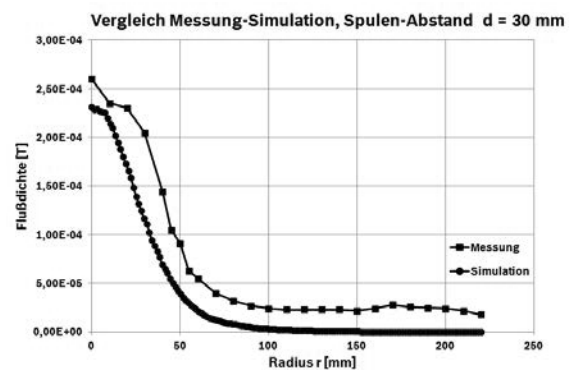


Abbildung 5. Gegenüberstellung von Messung und Simulation entlang der Profilinie

Es zeigt sich, daß der Verlauf der durch die Simulation gefundenen Flussdichte sehr derjenigen der Messung gleicht. Die Messung enthält allerdings weniger Messpunkte, da von Hand gemessen wurde.

Es zeigte sich, daß die durch Messung ermittelten Werte etwas streuen. Da die Sonde etwa eine Messfläche von 5 mm^2 und eine Dicke von $1,5 \text{ mm}$ hatte, war eine exakte Positionierung im Luftspalt schwierig. Zudem musste die Transversal-Sonde genau senkrecht zur Spulenoberfläche gehalten werden um die Flussdichte am gewählten Punkt richtig zu erfassen.

Die absoluten Flussdichten der Messungen lagen jedoch etwas über den mit der Simulation ermittelten. Im gezeigten Beispiel liegt die Abweichung innerhalb des Luftspalts durchschnittlich bei ca. $+11\%$. Außerhalb des Luftspalts ist die Abweichung etwas größer, da Flussdichten in Bruchteilen von μT an die Auflösungsgrenzen des Teslameters stoßen.

Neben den Flussdichten wurden auch die Kopplungsfaktoren an der Spulenanordnung mittels Messungen bestimmt. Tabelle 2 zeigt die Gegenüberstellung von Messung und Simulation. Auch hier ist die Abweichung zu den mittels der Simulation gefundenen Werten gering. Maßgebend für die Größe der Kopplungsfaktoren ist der Abstand der Spulen. Mit steigendem Abstand werden diese geringer.

Abstand d [mm]	Kopplungsfaktoren k Messung	Kopplungsfaktoren k Simulation
10	0,54	0,61
20	0,35	0,37
30	0,22	0,23

Tabelle 2. Vergleich der Kopplungsfaktoren Messung und Simulation

1.7 Zusammenfassung und Diskussion

Das mit der Simulation aufgebaute geometrische Modell der Spulenanordnung mit den zugewiesenen Eigenschaften der Ferrite und der Spulen liefert ein hinreichend genaues Modell des Magnetfelds im Luftspalt und bildet ebenso die in der Praxis gefundenen Kopplungsfaktoren ab.

Die gemessenen Flussdichten liegen etwas über denen der Simulation. Ein Grund kann darin liegen daß die in der Simulation verwendeten Spulenströme etwas geringer waren als die in den Versuchen tat-

sächlichen durch die Spulen fließenden Ströme. Die in der Simulation gewählten Abstände zwischen den einzelnen Windungen der Litzen könnten in der Simulation größer angenommen sein, als sie tatsächlich gewickelt wurden. Dahingehend könnte das Modell noch weiter verfeinert werden.

Aufgrund der auch ohne Optimierung erzielten hohen Übereinstimmung von Simulation und Messung ist es möglich die entsprechenden Auslegungsparameter für die Spulen und insbesondere der Übertragungsstrecke mittels der Simulation zu bestimmen, ohne daß diese Spulen vorher hergestellt werden müssen.

Mit der Simulation lässt sich die Parameterisierung der Modelle und die gezeigte Vorgehensweise auch auf induktive Übertragungssysteme übertragen, die wesentlich größere Abmessungen und Leistungen haben, wie sie beispielsweise zur Ladung von Elektrofahrzeugen in Entwicklung sind. Dabei sind vertikale Abstände im Bereich von $12\text{--}18 \text{ cm}$ un bei Spulen Durchmesser von $70\text{--}80 \text{ cm}$ gebräuchlich.

Es ist bei der Dimensionierung solcher Systeme zu beachten, daß hierbei die den einschlägigen Regularien und Richtlinien festgelegten maximalen Feldstärken berücksichtigt werden [5].

1.8 Danksagung

Herrn Tobias Diekhans, Robert Bosch GmbH, danke ich für die Unterstützung in der Modellierung.

2 References

- [1] See for Reference.
<http://www.wirelesspowerconsortium.com>.
- [2] Eberhard Waffenschmidt, Toine Staring. *Limitation of inductive power transfer for consumer applications*. 13th European Conference on Power Electronics and Applications (EPE 2009), Barcelona, Spain, 8.-10.Sep. 2009, paper #0607.
- [3] David Meeker. *Finite Element Method Magnetics User's Manual Version 4.2*. dmeeker@ieee.org.
- [4] Dominik Huwig und Peter Wambsganß. *Modulares Plattformkonzept für die kontaktlose Übertragung von Energie und Daten*. RRC power solutions GmbH. www.rrc-wireless-power.de, 2010

- [5] ICNIRP GUIDELINES for *limiting exposure to time-varying electric and magnetic fields (1Hz-100kHz)*.
Published in: Health Physics 99(6):818-836;
2010.

Reduzierte Modelle für elektromechanische Bauteile mit Berücksichtigung von Wirbelströmen

Ursula Voß¹, Martin Hanke²

¹Hochschule Reutlingen

²CADFEM GmbH

Ursula.voss@reutlingen-university.de

Die sogenannte Systemsimulation, bei der mehrere physikalische Domänen gemeinsam simuliert werden, erlaubt die Analyse komplexer und damit realitätsnaher Systeme und spielt eine zunehmend größere Rolle bei der Auslegung von Komponenten. Enthält das System Teile, die durch Feldgrößen aus unterschiedlichen physikalischen Domänen beschrieben werden müssen, kann man Co-Simulationen einsetzen, die allerdings zeitaufwändig sind. Für die Auslegung des Systems ist es dagegen notwendig, Systemsimulationen schnell durchzuführen zu können. Hierfür können für ausgewählte Bauteile oder Domänen schnellere reduzierte Ersatzmodelle (ROM) eingesetzt werden. In dieser Arbeit stellen wir ein reduziertes Modell für elektromechanische Bauteile mit Berücksichtigung von Wirbelströmen vor. Wirbelstromeffekte hängen nicht nur vom aktuellen Zustand, sondern auch von der Geschichte der elektromagnetischen Domäne ab. Das vorgestellte Ersatzmodell basiert auf Daten, die mit einer Reihe von stationären Feldsimulationen vorab erzeugt werden. Für die Modellierung der geschichtsabhängigen Wirbelstromeffekte wird ein Konvolutionsansatz (Faltungsansatz) verwendet. Vergleiche mit entsprechenden Co-Simulationen in ANSYS Maxwell und Simplorer zeigen am Beispiel eines Hubankers, dass das Ersatzmodell in der Lage ist, die wesentlichen Eigenschaften des Bauteils physikalisch korrekt abzubilden.

1 Einleitung

Die Systemsimulation, bei der mehrere physikalische Domänen gemeinsam simuliert werden, erlaubt die Analyse komplexer und damit realitätsnaher Systeme und spielt eine zunehmend größere Rolle bei der Auslegung von Komponenten. Dafür ist es notwendig, Systemsimulationen schnell durchzuführen zu können. Hierfür können für ausgewählte Bauteile oder Domänen schnellere reduzierte Ersatzmodelle (ROM) eingesetzt werden [1].

1.1 Wirbelströme in elektromechanischen Systemen

Das Ziel dieser Arbeit ist die Berücksichtigung von Wirbelstromeffekten in der Systemsimulation elektromechanischer Komponenten. Eine solche Komponente könnte beispielsweise der in einem Magnetventil verbaute Hubanker sein, siehe Abb. 1. Typisch für solche Komponenten ist, dass die mechanische Bewegung eine Rückwirkung auf den elektrischen Kreis hat. Die Pulsweite der elektrischen Ansteuerung und die charakteristischen Bewegungszeiten des Ankers haben dieselbe Größenordnung im Bereich einer Zehntelmillisekunde, siehe Abb. 2 oben. Daher ist es wichtig, die Zusammenhänge genau abbilden zu können, um z.B. ein „Klebenbleiben“ des Ankers im gepulsten Betrieb vermeiden zu können oder ein solches Kleben gerade sicherzustellen. Wirbel-

stromeffekte können das Verhalten dabei deutlich ändern.

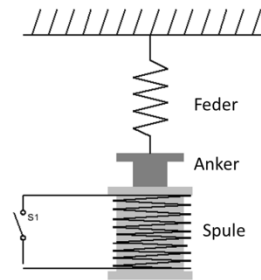


Abbildung 1. Schematische Darstellung eines Hubankers. Wird der Schalter geschlossen, wird der Anker in die Spule hineingezogen. Wird der Schalter geöffnet, bringt die Feder den Anker in die Ausgangslage zurück.

Wirbelströme treten auf, wenn sich ein Leiter in einem zeitlich konstanten Magnetfeld bewegt, oder wenn ein Leiter sich in einem zeitlich veränderlichen Magnetfeld befindet. Sie erzeugen ein Magnetfeld, das dem ursprünglichen Feld entgegenwirkt und damit seine Wirkung schwächt. Hat das Material einen elektrischen Widerstand, erwärmt es sich [2]. Am Beispiel des Hubankers kann man die Auswirkung von Wirbelströmen deutlich erkennen: Der Maximalstrom wird deutlich höher, aber die auf den Anker

wirkende Kraft wird kleiner, dadurch wird der Anker später und in manchen Fällen, wie hier dargestellt, nicht vollständig angezogen, siehe Abb. 2. unten. Trotz der geringeren mechanischen Bewegung ist die aufzuwendende elektrische Energie höher.

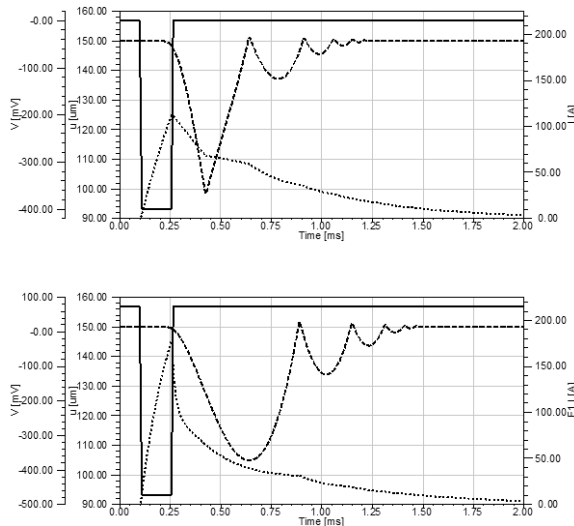


Abbildung 2. Typische Verläufe von Spannung (durchgezogene Linie), Strom (gepunktet) und Verschiebung des Ankers (gestrichelt) ohne Berücksichtigung von Wirbelstromeffekten (oben) und mit Wirbelstromeffekten (unten).

1.2 Systemsimulation

Wegen der Komplexität der Geometrie und der vielfältigen Ursachen für Wirbelströme kann der gesamte Effekt nur mithilfe der numerischen Simulation quantitativ dargestellt werden. Hier verwendete Werkzeuge dafür sind ANSYS Maxwell, ein auf der Finite-Elementmethode basierendes Programm für die Simulation elektromagnetischer Felder und ANSYS Simplorer, ein Simulationswerkzeug zur Systemsimulation von Leistungselektronik und elektrisch kontrollierten Systemen.

Bisher gibt es mit diesen Werkzeugen zwei Methoden, die geometrische Komplexität in der Systemsimulation abzubilden: Zum einen die sogenannte Co-Simulation, bei der eine Maxwell – und eine Simplorer-Simulation in jedem Zeitschritt im Wechsel durchgeführt werden. In der Maxwellrechnung wird die tatsächliche detaillierte Geometrie verwendet und so die geometrische Komplexität, Nichtlinearitäten und Wirbelströme abgebildet. Simplorer dagegen bildet Leistungselektronik und das mechanische Verhalten konzentrierter Elemente ab [3]. Mit der Co-simulation kann man Wirbelstromeffekte abbilden, aber die Berechnungen sind zeitaufwändig und damit ungeeignet ist für die Auslegung von Parametern wie

zum Beispiel Pulsweite, maximale Spannung und Vorspannung der Feder.

Eine schnelle Alternative ist die Verwendung des Equivalent Circuit Extractor (ECE), der reduzierte Modelle direkt aus Maxwell für Simplorer erzeugt. Hierbei wird aus einer Serie stationärer, d.h. nicht zeitabhängiger Maxwell-Rechnungen ein Kennfeld erzeugt, aus dem während der Systemsimulation der verkettete magnetische Fluss und die erzeugte Kraft gelesen werden. Dieses Verfahren ist nicht in der Lage Wirbelstromeffekte abzubilden.

2 Optislang MOP als Kennfeld

Um später auch ein Modell der Wirbelstromeffekte in das Ersatzmodell integrieren zu können, wird der Ansatz des Equivalent Circuit Extractor mit einem flexiblen Kennfeld in Simplorer implementiert. Zugrunde gelegt werden wie im ECE-Modell die mechanische Bewegungsgleichung

$$m \ddot{u} + c \dot{u} + k u = F \quad (1)$$

und die elektrische Bewegungsgleichung

$$U = RI + n \frac{d\Phi}{dt} \quad (2)$$

Diese sind über die Kraft

$$F = F(u, I) \quad (3)$$

und den magnetischen Fluss

$$\Phi = \Phi(u, I) \quad (4)$$

gekoppelt. Diese beiden Größen werden für verschiedene Werte der mechanischen Verschiebung u und des elektrischen Stromes I tabelliert.

Als erweiterbare Tabelle wird ein sogenanntes Meta-model of Optimal Prognosis (MOP) aus der Software Optislang verwendet. Optislang ist eine Software für Sensitivitätsanalyse, Optimierung und Robustheitsanalyse, die in ANSYS Workbench integriert ist. Ein MOP ist eine Response-Surface, die Funktionswerte für eine Anzahl von Design Points optimal interpoliert. MOPs sind insbesondere für Sensitivitätsanalysen und Optimierung bei einer großen Anzahl von Parametern entwickelt worden [4]. Solche MOPs können innerhalb der ANSYS Workbench aus beliebigen Analysen erzeugt werden. In der vorliegenden Anwendung werden sie aus magnetostatischen Maxwellrechnungen bedatet und mit einer geringen Anzahl von Parametern als Kennfeld verwendet. Eine C-Bibliothek, die es erlaubt, für beliebige Eingangsparameter den Wert der Antwortfläche zu ermitteln, steht als dynamic link library (dll) zur Verfügung

[5]. Diese Programmbibliothek kann in Simplerer integriert werden. Für aktuelle Werte von u und I werden so $\Phi(u, I)$ als rechte Seite der elektrischen Bewegungsgleichung und $F(u, I)$ als äußere Kraft des mechanischen Teilmodells ermittelt. Die Auswertung des MOP muss in C erfolgen, während die Bewegungsgleichungen dann sowohl in C oder in VHDL-Block gelöst werden können, siehe Abb. 3 und 4.

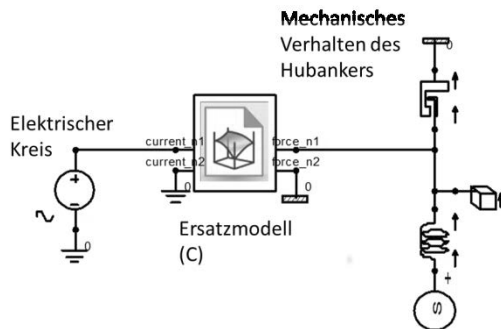


Abbildung 3. Simplermodell, in dem die Auswertung des MOPs in das Ersatzmodell integriert ist.

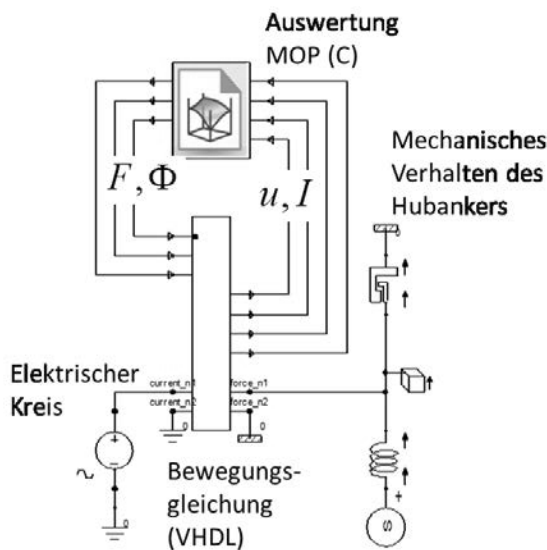


Abbildung 4. Simplermodell, in dem die Bewegungsgleichungen in einem VHDL-Block und gelöst werden, während die Auswertung des MOPs davon getrennt stattfindet.

3 Modellierung von Wirbelstromeffekten

Wirbelstromeffekte sind nicht nur von aktuellen Werten von Verschiebung und Strom bzw. Erregung abhängig, sondern auch von der Stromanstiegsgeschwindigkeit und der Vorgeschichte der Systems.

In Abb. 5 sind verketteter Fluss und Kraft für verschiedene Stromanstiegsgeschwindigkeiten \dot{I} bei konstanter Spaltöffnung aufgetragen. Der Strom wurde dabei linear bis 200 A erhöht und wieder auf 0 reduziert. Für Werte von $\dot{I} = 1 \text{ e}^{-3} \text{ A/s}$ und kleiner erhält man dasselbe Verhalten wie ohne Berücksichtigung von Wirbelstromeffekten. In diesem Fall ist der Kurvenverlauf symmetrisch, es gibt keine Abhängigkeit von der Vorgeschichte. Je größer die Stromanstiegsgeschwindigkeit ist, desto kleiner sind die Maximalwerte. Das Maximum wird verzögert angenommen und der Verlauf ist nicht mehr symmetrisch.

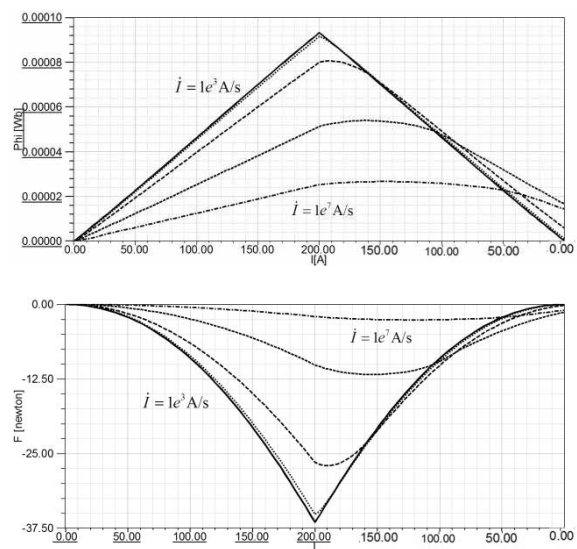


Abbildung 5. Verketteter Fluss Φ (oben) und Anzugskraft F im Hubanker (unten) für feste Ankerpositionen und verschiedene Stromanstiegsgeschwindigkeiten. $\dot{I} = 1 \text{ e}^{-3} \text{ A/s}$ ist dabei ein Wert, bei dem keine Wirbelstromeffekte mehr sichtbar werden.

Diese Wirbelstromeffekte werden in erster Näherung mit Hilfe der gewöhnlichen Differentialgleichung

$$\tau \frac{d\Phi(t)}{dt} + \Phi(t) = \Phi_{MOP}(t), \quad \Phi(0) = 0 \quad (5)$$

modelliert. Φ_{MOP} bezeichnet dabei den verketteten Fluss, der sich ohne Berücksichtigung von Wirbelstromeffekten ergibt. Die Lösung der Differentialgleichung ist

$$\Phi(t) = e^{-\frac{t}{\tau}} \int_0^t \frac{\Phi_{MOP}(t')}{\tau} e^{\frac{t'}{\tau}} dt'. \quad (6)$$

Mittels partieller Integration erhält man daraus

$$\Phi(t) = \Phi_{MOP}(t) - \int_0^t \Phi_{MOP}(t') e^{-\frac{t-t'}{\tau}} dt'. \quad (7)$$

Der letzte Term beschreibt eine Faltung der rechten Seite der Differentialgleichung mit einer abklingenden Exponentialfunktion. Für

$$\Delta \Phi(t) = \int_0^t \Phi_{MOP}(t') e^{-\frac{t-t'}{\tau}} dt' \quad (8)$$

erhält man mit $t = n \Delta t$ und mit der zusammengesetzten Rechteckregel

$$\Delta \Phi(n \Delta t) \approx \left[\sum_{k=1}^n \Phi_{MOP}(k \Delta t) e^{-\frac{k \Delta t - n \Delta t}{\tau}} \right] \Delta t. \quad (9)$$

Damit erhält man

$$\begin{aligned} \Delta \Phi(n \Delta t) &= \Delta t \Phi_{MOP}(n \Delta t) e^{-\frac{n \Delta t - n \Delta t}{\tau}} \\ &\quad + \Delta t \sum_{k=1}^{n-1} \Phi_{MOP}(k \Delta t) e^{-\frac{k \Delta t - n \Delta t}{\tau}} \\ &= \Delta t \Phi_{MOP}(n \Delta t) \\ &\quad + \Delta t e^{-\frac{\Delta t}{\tau}} \sum_{k=1}^{n-1} \Phi_{MOP}(k \Delta t) e^{-\frac{k \Delta t - (n-1) \Delta t}{\tau}} \\ &= \Delta t \Phi_{MOP}(n \Delta t) + e^{-\frac{\Delta t}{\tau}} \Delta \Phi((n-1) \Delta t). \end{aligned}$$

Φ_{MOP} wird im Vorfeld für gegebene Werte von u und I erzeugt und als Kennfeld zur Verfügung gestellt. Mit einer numerischen Berechnung der Zeitableitung $\dot{\Phi}_{MOP}$ zum jeweils aktuellen Zeitpunkt erhält man so eine rekursive Formel zur Berechnung der Korrektur, die gut implementiert werden kann. Eine entsprechende Gleichung gilt für die Korrektur der Kraft F . Die korrigierten Werte werden in Gleichung (1) und (2) verwendet.

4 Ergebnisse

Das entwickelte Ersatzmodell ist Wirbelstromeffekten wurde auf zwei verschiedene Bauarten eines Hubankers angewendet, wobei die zweite Variante konstruktive Elemente zur Verringerung der Auftreffgeschwindigkeit und des Gesamtenergiebedarfs enthält. Das Ersatzmodell liefert für Vergleichsprobleme dieselben Bewegungsabläufe, Stromanstiegszeiten und Maximalströme wie die Cosimulation, die als Referenz herangezogen wird, siehe Abb. 6. Dabei liegen die Rechenzeiten im Bereich der Rechenzeiten des ECE-Modells, während die Cosimulation wesentlich zeitaufwändiger ist.

Vergleiche der aufgewendeten elektrischen Energie (siehe Tabelle 1) zeigen wie erwartet, dass Wirbelstromeffekte dazu führen, dass mehr Energie aufgewendet werden muss. Die Fehler des neuen Verfahrens sind deutlich kleiner als die durch Wirbelströme hervorgerufenen Effekte. Der etwas größere Unterschied bei Geometrie 2 erklärt sich dadurch, dass in diesem Fall der Anker die untere Position nur kurz

erreicht und sofort in die Ausgangsposition zurückkehrt.

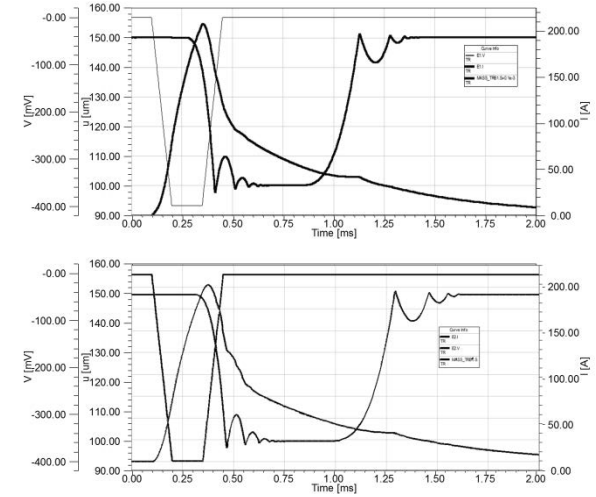


Abbildung 6. Vergleich des zeitlichen Ablaufs zwischen Cosimulation (oben) und der Simulation mit dem Ersatzmodell (unten) für Geometrie 1. Dargestellt sind der vorgegebene Spannungsverlauf, der Stromverlauf und die Bewegung des Ankers.

Geometrie 1	Verfahren	E_in [mJ]
Ohne Wirbelstromeffekte	Cosimulation	8,41
	Ersatzmodell	8,31
Mit Wirbelstromeffekten	Cosimulation	13,60
	Ersatzmodell	13,06

Geometrie 2	Verfahren	E_in [mJ]
Ohne Wirbelstromeffekte	Cosimulation	5,99
	Ersatzmodell	5,77
Mit Wirbelstromeffekten	Cosimulation	12,70
	Ersatzmodell	9,50

Tabelle 1. Vergleich der aufgewendeten elektrischen Energie für zwei verschiedene Ankergeometrien.

5 Zusammenfassung

Insgesamt wurde ein schnelles Ersatzmodell entwickelt, mit dem elektromechanische Bauteile auch unter Berücksichtigung von Wirbelstromeffekten in einer Systemsimulation simuliert werden können. Damit steht ein Werkzeug für Parameterauslegung und Optimierung in Simpler zur Verfügung. Dieses Verfahren verwendet das Metamodell of Optimal Prognosis (MOP) aus Optislang als Kennfeld. Dieses wird in einem Simplerblock ausgewertet und die Werte als Eingangsdaten für das Lösen der Bewegungsgleichungen in Simpler verwendet. Wirbelstromeffekte werden über eine mit Hilfe eines Faltungsintegrals beschriebene Korrektur von Kraft und Fluss berücksichtigt. Damit werden Simulationser-

gebnisse erzielt, die hinsichtlich Maximalstrom, Anzugszeit und Energiebilanz mit der entsprechenden Cosimulation vergleichbar sind.

6 References

- [1] W.H. Schilders. *Introduction to Reduced Order Modelling*. In: Model Order reduction: Theory, Research Aspects and Applications, Herausgeber: W.H. Schilders, H.A. Forst und J. Rommes, Springer, S. 1-31, 2008.
- [2] E. Kallenbach et al. *Elektromagnete: Grundlagen, Berechnung, Entwurf und Anwendung*, Springer 2012.
- [3] P. Zhou, D. Lin, W.N. Fu, B. Ionescu und Z.J. Cendes. *A general cosimulation approach for coupled field-circuit problems*, IEEE Transactions on Magnetics, 42, 4, S. 1051-1054, 2006.
- [4] R. Schirrmacher, Ch. Kubala, S. Kandaswamy und H. Ulmer. *Use of Metamodels for the Interdisciplinary Multi-Objective-Optimization (MOO)*, 8. Weimarer Optimierungs- und Stochastiktag, Weimar, 2011.
- [5] O. Siegemund. *Multiscale System Simulation*. Master-Thesis, Landshut University of Applied Science, Ingolstadt University of Applied Science, Deutschland, 2013.

Kühlungsoptimierung von elektronischen Komponenten mittels Strömungssimulation

B.Eng. Bernd Scheiderer, Prof. Dr. phil. nat. Wolfgang Schlüter, Dipl.-Ing. (FH) Ansgar Ringleb
Kompetenzzentrum Strömungssimulation - KSTS
Hochschule Ansbach
b.scheiderer@hs-ansbach.de

Im vorliegenden Beitrag wird die grundsätzliche Methodik bei der Modellbildung einer Kühlungsoptimierung mittels Strömungssimulation (engl.: Computational Fluid Dynamics, CFD) erläutert. Dabei wird verstärkt auf die Geometriemodellierung eingegangen. Neben einem flexibel gestalteten CAD-Modell ist darauf zu achten, einen CFD-gerechten Modellaufbau zu gewährleisten. Ein weiterer Aspekt ist eine mögliche Vereinfachung hinsichtlich der zu analysierenden Messgröße innerhalb des CFD-Modells. Ein dritter und sehr wichtiger Modellierungsbaustein ist die Vernetzung des zu simulierenden Systems. Je nach Intention des Simulationsvorhabens und den gegebenen Systemeigenschaften werden unterschiedliche Anforderungen an das Netz gestellt, welche bei der Methodik der Vernetzung beachtet werden müssen.

Die genannten Modellierungsschritte werden hinsichtlich ihrer Randbedingungen und der angewandten Systematik dargestellt. Zur Veranschaulichung der Vorgehensweise werden die Modellierungsschritte am Beispiel eines Projekts veranschaulicht. Projektziel stellte die Optimierung der Komponentenkühlung innerhalb eines Schaltschranks dar. Projektpartner waren die Hochschule Ansbach und die Firma R.Stahl Schaltgeräte GmbH, Waldenburg.

1 Einleitung

Die Kühlung von elektronischen Komponenten stellt eine wichtige Teilaufgabe in der Elektronikentwicklung dar. Hohe Temperaturen gefährden die Funktionalität und verringern die Haltbarkeit. Bei der Optimierung der Kühlung bietet die Strömungssimulation (engl.: Computational Fluid Dynamics, CFD) erhebliche Vorteile. So können Kausalitäten visualisiert und analysiert werden, die messtechnisch, wenn überhaupt, nur mit sehr hohem Aufwand darstellbar wären. Untersuchungen hinsichtlich Verbesserungen der Strömungsführung anhand der Variation der Systemgeometrie oder durch technische Eingriffe in das System lassen sich ebenfalls mit geringerem zeitlichen Aufwand mittels CFD realisieren.

Das zu optimierende System ist ein Schaltschrank, der temperaturempfindliche elektronische Komponenten der Firma R.Stahl Schaltgeräte GmbH enthält. Zur geometrischen Modellierung des Schaltschranks wurde ein CAD-Programm eingesetzt. Zur Vernetzung und Simulation wurde eine kommerzielle CFD-Software verwendet.

Der Workflow des Projekts (Abbildung 1) zeigt den iterativen Ablauf einer Optimierung. Nach der Aus-

wertung der Simulationsergebnisse wird in jedem Optimierungsschritt die Geometrie neu angepasst und der Vorgang muss von hier aus von neuem begonnen werden.

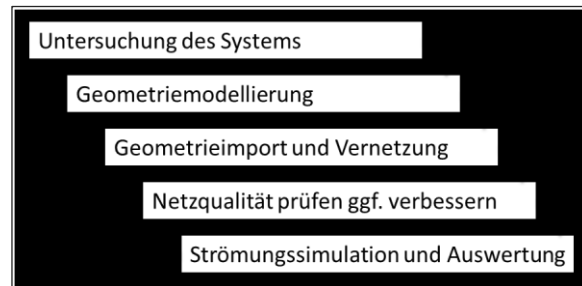


Abbildung 1: Workflow des Optimierungsprojekts

Zur Gewährleistung einer zeiteffizienten Systemoptimierung mittels CFD muss das Simulationsmodell ausreichend flexibel für Geometrievariationen und hinsichtlich der Simulationszeit hinreichend leistungsfähig gestaltet werden. Da viele Schritte zur Vorbereitung der CFD Simulation in jedem Optimierungszyklus gleich sind, sollten diese automatisierbaren Schritte mittels Makrobefehlen ausführbar gestaltet werden.

Um beides im Optimierungsprojekt zu erreichen, werden die folgenden Vorgehensweisen empfohlen:

- Wahl einer sinnvollen zu analysierenden Messgröße
- Intelligente Geometriemodellierung
- Systemspezifische Vernetzung

2 Wahl einer sinnvollen Messgröße

Während der Analyse des zu simulierenden Systems, sollte neben den zu variierenden Parametern auch untersucht werden, anhand welcher Messgröße die Optimierung des Systems erfolgen soll. Dabei muss die in der Praxis physikalisch interessierende Größe nicht unbedingt die für die Simulation sinnvollste Lösung darstellen, was am folgenden Beispiel näher betrachtet werden soll.

Die Kühlung der Elektronikkomponenten erfolgt mit Hilfe von Zwangskonvektion, dabei steht die Optimierung des Wärmeübergangs \dot{q} (Formel 1) von den elektrischen Komponenten auf das Fluid im Fokus.

$$\dot{q} = \alpha \cdot (T_K - T_\infty) \quad (1)$$

Der Wärmeübergang \dot{q} ist linear vom Wärmeübergangskoeffizienten α bzw. der Nusselt-Zahl Nu (Formel 2) abhängig.

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

Die Nusselt-Zahl Nu (Formel 3) wiederum ist geometrieabhängig und ist eine Funktion der Prandtl-Zahl Pr , die einen Stoffwert darstellt, und der Reynolds-Zahl Re , die die Strömung beschreibt.

$$Nu \approx C \cdot Re^m \cdot Pr^n \quad (3)$$

Die Temperaturdifferenzen im System steigen nicht über 50 K, infolgedessen werden die Stoffwerte nicht wesentlich beeinflusst ($\Delta Pr < 2\%$). Damit es kann die Nu -Zahl und somit der Wärmeübergang \dot{q} als Funktion von der Reynolds-Zahl Re angenommen werden. Unter Vernachlässigung des leicht exponentiellen Einflusses der Re -Zahl (i.d.R. gilt $m \approx 0,8$), kann vereinfacht festgehalten werden, dass der Wärmeübergang linear mit der Re -Zahl zunimmt. Damit kann das vorliegende Kühlproblem zur effizienten Untersuchung und Optimierung mittels CFD-Simulation auf ein Strömungsproblem reduziert werden.

Die Reynolds-Zahl Re , welche als das Verhältnis der Trägheits- zu den Reibungskräften interpretiert werden kann [1], ist proportional zur Strömungsgeschwindigkeit u (Formel 4) beziehungsweise zum Volumenstrom \dot{V} (Formel 5).

$$Re = \frac{u \cdot L}{\nu} \quad (4)$$

$$u = \frac{\dot{V}}{A} \quad (5)$$

Je größer das durch und um die elektronischen Komponenten strömende Volumen \dot{V} des Fluids ist, desto höher ist somit die abgeführte Wärme \dot{q} und desto niedriger bleibt die Komponententemperatur T_K .

Die vorgenommene Vereinfachung des ursprünglichen Kühlproblems auf ein Strömungsproblem, sprich die vorgenommene Reduzierung auf eine Kennzahl (Re -Zahl), bietet zudem den Vorteil, dass während der Optimierung nur ein Parameter betrachtet und ausgewertet werden muss. Würde im System auch die Temperatur analysiert werden, müsste mindestens die Nusselt-Zahl zusätzlich bei allen Auswertungen beachtet werden.

Die rein strömungstechnische Analyse des Systems bietet hinsichtlich des Simulationsmodells drei entscheidende Vorteile. Erstens muss dadurch im Modell nur der Raum, der die Komponenten umgibt und durchströmt, modelliert werden, nicht aber die elektrischen Komponenten selbst. Dies verringert den Modellierungsaufwand beträchtlich und verringert zusätzlich die Anzahl der Simulationszellen und damit der Simulationszeit. Zweitens kann so auf die thermische Modellierung und somit auf das Lösen der numerisch aufwendig zu behandelnden Energiegleichung verzichtet werden, wodurch die Simulationszeit etwa um den Faktor 2,5 reduziert wird. Drittens würde die Simulation des Wärmeübergangs die genaue Bestimmung der Wärmeleitfähigkeitswerte λ aller elektronischen Komponenten erfordern, welche aufgrund der Vereinfachung entfallen kann.

Ein Nachteil steht diesen gravierenden Vorteilen gegenüber. Anhand der Simulationsergebnisse lassen sich hinsichtlich der Komponententemperatur nur qualitative Aussagen treffen. Dies erschwert auch die Validierung des Modells, da keine quantitativen Temperaturwerte vorliegen, welche wesentlich einfacher und präziser zu messen sind als Volumenströme oder Strömungsgeschwindigkeiten.

3 Intelligente Geometriemodellierung

Um Variationen in der Systemgeometrie zu ermöglichen, sollte das CAD-Modell flexibel und mittels Geometrievariablen änderbar gestaltet werden. Darüber hinaus ist darauf zu achten, strömungssimulationsoptimierte Rechengebiete zu modellieren. Dazu kann die Geometrie in weniger wichtigen Bereichen mit geringerem Detaillierungsgrad ausgeführt werden.

3.1 Korrelierte Bemaßung im CAD-Modell

Eine korrelierte Bemaßung im CAD-Modell verringert die Zeit, die für wiederkehrende Anpassungen des Modells zur Optimierung des Systems nötig sind, erheblich. Deshalb sollte zumindest jeder Parameter, den es im Laufe der Optimierungen zu variieren gilt, als Variable x_i vorhanden sein. Diese Bemaßungen bzw. Variablen gehören formal einer Menge M_1 an. Da sich häufig viele geometrische Größen eines Systems mit der Änderung einer anderen Größe verändern, sollte im Vorfeld der Modellierung darauf geachtet werden, welche Größen y_j von welchen Variablen x_i abhängig sind. Alle abhängigen Größen y_j gehören einer Menge M_2 an. Das CAD-Modell muss so gestaltet werden, dass alle diese abhängigen Größen y_j automatisch berechnet (Formel 6) und im Modell verwendet werden.

$$y_j = f(x_i), \quad x_i \in M_1, y_j \in M_2 \quad (6)$$

Ist dies nicht der Fall, droht für jede Variation der Modellgeometrie erheblicher Zeitaufwand, um die nicht automatisch berechneten Maße an unterschiedlichen Stellen zu finden und manuell zu ändern. Bei konsequenter Anwendung der korrelierenden Bemaßung, nimmt eine Geometrieänderung im CAD-Modell nur wenige Minuten in Anspruch. Es müssen nur die Variablen x_i geändert sowie die daraufhin automatisch erzeugte Systemgeometrie gespeichert werden.

3.2 Vereinfachungen im CAD-Modell

Der Detaillierungsgrad des CAD-Modells wirkt sich direkt auf die Zellenanzahl im späteren CFD-Modell aus. Deshalb sollte in Bereichen, die für die zu untersuchende oder optimierende Größe belanglos sind, das CAD-Modell vereinfacht werden. Beispiele sind etwa Kabelkanäle, die an den Seiten geöffnet sind, aber im Modell besser lediglich als Quader abgebildet werden (Abbildung 2).

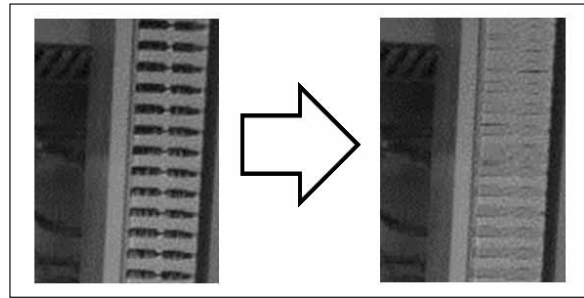


Abbildung 2: Offene und geschlossene Kabelkanäle

Kann im Vorfeld nicht ausgeschlossen werden, dass diese Vereinfachung einen Einfluss auf die Messgröße nimmt, ist die Durchführung einer diesbezüglichen Geometriestudie erforderlich. Im CAD-Modell ist vorzusehen, dass schnell zwischen detailliertem Modell und Vereinfachung umgeschaltet werden kann. Dies macht eine modulare Konstruktion der Details notwendig, so dass diese im CAD-Modell aktiviert/deaktiviert werden können, ohne einen Einfluss auf das übrige Modell nehmen. Im Schaltschrank wurde zum Beispiel als Vereinfachung die Modellierung eines Kabelstrangs entfernt. Diese Vereinfachung hatte bei der nachfolgenden Vernetzung eine Reduzierung der Zellenanzahl um 20 % zur Folge. Die Analyse, ob diese Vereinfachung angewendet werden kann, war jedoch negativ (Tabelle 1). Der Kabelstrang beeinflusst die Ergebnisse der Strömungsgeschwindigkeit nur geringfügig, die Ergebnisse der Volumenstromwerte ändern sich hingegen zum Teil deutlich.

Tabelle 1: Abweichung durch die Vereinfachung im CAD-Modell

Messgröße	Abweichung [%]
Volumenstrom \dot{V}_1	3,4
Volumenstrom \dot{V}_2	5,4
Volumenstrom \dot{V}_3	6,3
Strömungsgeschwindigkeit u_1	1,7

4 Vernetzung des Simulationsgebiets

Je nach Intention des Simulationsvorhabens und den vorliegenden Systemeigenschaften müssen bei der Vernetzung des Simulationsgebietes unterschiedliche Schwerpunkte gesetzt werden, denn die Simulationsdauer hängt nahezu linear von der Zellanzahl (Diskretisierung) ab. Bei einem Optimierungsvorhaben sollte die Simulationszeit eines Simulationsdurchlaufes und damit die Zellanzahl möglichst gering sein. Hinzu

kommt bei der Simulation von elektronischen Komponenten, dass häufig feine geometrische Strukturen vorhanden sind, die ein sehr feines Netz erfordern, was die Zellanzahl stark erhöht [2].

Um diese konträren Anforderungen bestmöglich zu erfüllen, empfiehlt sich die Anwendung geeigneter Vernetzungsmethoden und deren Überprüfung mittels einer sog. Netzstudie.

4.1 Vernetzungsmethodik

Ist die Geometrie des Strömungsgebietes grundsätzlich rechteckig oder kreisförmig, kann je nach Geometrieform ein Netz mit vierflächigen (tetrahedral) oder sechsfächigen (hexahedral) Zellen verwendet werden. Sind die Strömungsgebiete von komplexer geometrischer Form, wie beispielsweise der zu simulierende Schaltschrank, sind vielblächige (polyhedral) Zellen zu bevorzugen. Diese bieten Vorteile hinsichtlich des Konvergenzverhaltens und damit der Simulationszeit [3].

Vor jeder Vernetzung sollte die Ausnutzung von Symmetrieeigenschaften und deren strömungsmechanischer Einfluss geprüft werden. Eine zweidimensionale Simulation kann je nach Geometrie und physikalischen Bedingungen eine etwa 95 % geringere Simulationszeit gegenüber einer dreidimensionalen Strömungssimulation aufweisen. Zudem ist das System auf Symmetrieachsen hin zu untersuchen. Sind Symmetrien im System vorhanden, braucht, je nach Anzahl der Symmetrien nur ein Bruchteil des Systems vernetzt und simuliert werden, entsprechend kürzer ist der benötigte Zeitaufwand.

Eine sehr effektive Methode zur Reduzierung der Simulationszeit unter Gewährleistung belastbarer Ergebnisse ist die Anwendung regional verfeinerter Netze [4]. Dabei kann das Rechenggebiet in Bereichen, in denen niedrige Geschwindigkeitsgradienten auftreten, gröber aufgelöst werden. Bis zu einem gewissen Grad hat diese Vereinfachung keine gravierenden Auswirkungen auf die Simulationsergebnisse. Da es jedoch häufig geometrisch einfacher ist, die Gebiete, an denen das Netz feiner sein sollte, zu definieren, als das Gebiet in dem das Netz gröber sein kann, werden normalerweise die Gebiete mit feinerem Netz definiert (Abbildung 3).

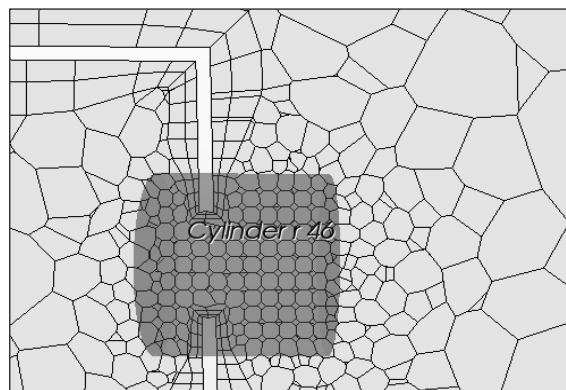


Abbildung 3: Beispiel einer Netzverfeinerung

Diese können je nach CFD-Software unterschiedliche geometrische Ausmaße haben (z. B. quader-, kegel-, zylinder- oder kugelförmig). Innerhalb dieser Bereiche kann das Netz hinsichtlich der Zellgröße und anderer Kriterien, wie die Anzahl und Dicke der wandnahen Zellen, verfeinert werden. Alle Größenangaben können relativ zum Wert einer Basisgröße oder in absoluten Längenmaßen definiert werden, wobei dies individuell abzuwägen ist. Beispielsweise sollten die Zellgrößen relativ definiert werden, um bei einer Skalierung des Netzes auch die Zellgrößen in den feiner vernetzten Bereichen zu verändern. Bei Größen, die der wandnahen Vernetzung und somit der Auflösung der Grenzschicht dienen, sind hingegen absolute Vorgaben empfehlenswert, da die Grenzschicht physikalische Eigenschaften aufweist und somit unabhängig von der Vernetzung zu sehen ist [5].

Bei der vorliegenden Schaltschranksimulation konnte auf keine Symmetrieeigenschaft zurückgegriffen werden, so dass ein volles 3D-Modell untersucht wurde. Zur Auflösung der klein strukturierten Lüftungsschlitze, Bauteilzwischenräume und Bereichen mit hoher Strömungsgeschwindigkeit wurde ein regional verfeinertes Netz eingesetzt. Hohe Strömungsgeschwindigkeiten sind beispielsweise bei den Lüftern und geometrischen Eingriffen in die Strömungsführung vorhanden. Insgesamt konnte die globale Netzgröße von 25 mm (Netzgeneration 1) auf 200 mm (Netzgeneration 4) erhöht werden, bei gleichbleibender Netzauflösung in den relevanten Strömungsbereichen. Dadurch reduzierte sich die Zellanzahl bei uniformer Vernetzung von 18,2 Mio. Zellen (Netzgeneration 1) um den Faktor 30 auf lediglich 663.000 Zellen bei regional verfeinerter Vernetzung (Netzgeneration 4). Die Simulationszeit wurde infolge dessen

von über 7 Stunden auf weniger als 30 Minuten verkürzt (Abbildung 4).

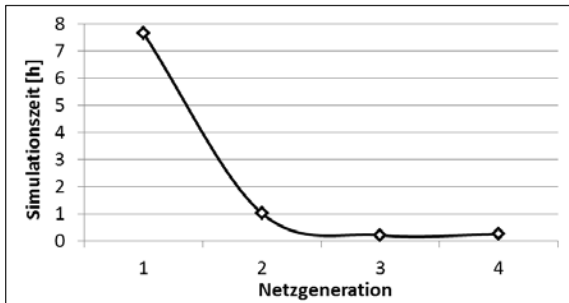


Abbildung 4: Simulationszeiten der Netzgenerationen

4.2 Analyse der Netzqualität

Bei Optimierungsproblemen, die mittels Strömungssimulation gelöst werden sollen, ist stets auf einen Kompromiss zwischen niedriger Zellanzahl im CFD-Modell und noch vertretbarer Qualität der Ergebnisse zu achten, da sonst aufgrund der vielen Variationen und damit notwendigen hohen Simulationsanzahl nicht in einem angemessenen Zeitrahmen simuliert werden kann. Mittels einer Netzstudie kann der Einfluss der Diskretisierung auf die Ergebnisqualität untersucht werden, dabei sollte das Netz jeweils um den Faktor 2 verfeinert werden [6].

In der Netzstudie des Schaltschranks wird deutlich, dass die Netzgröße 100 mm Abweichungen kleiner 5% gegenüber der feinsten Auflösung aber eine um den Faktor 12-16 geringere Rechenzeit aufweist. Eine gröbere Vernetzung würde zwar eine weitere Reduzierung der Rechenzeit um den Faktor 6-8 bedeuten, weist aber Abweichungen größer 10 % gegenüber der feinsten Auflösung auf (Abbildung 5).

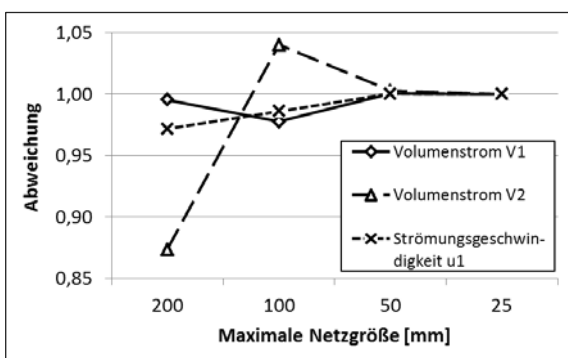


Abbildung 5: Netzstudie mit variierter Zellgröße

Somit bietet die Netzgröße 100 mm einen guten Kompromiss zwischen hoher Genauigkeit sowie niedriger Rechenzeit und empfiehlt sich daher zur Durchführung von Simulationen zur Optimierung der

Kühlung der elektronischen Komponenten im Schaltschrank.

Zur Validierung der Simulationen anhand experimenteller Daten würde sich die Vernetzung 50 mm eignen, da kaum noch Unterschiede zu einer noch feineren Vernetzung festgestellt werden können.

5 Zusammenfassung

Im Beitrag konnte gezeigt werden, dass es bereits beim Design des CAD-Modells wichtig ist, unnötige Details zu vereinfachen, da diese die Zellzahl in der späteren Simulation stark erhöhen. Weitere Rechenzeit kann durch die Wahl der Messgröße und somit der notwendigen physikalischen Modellierung, anhand derer das System optimiert werden soll, eingespart werden. Durch die dargestellte Zeiteinsparung durch Anwendung regional verfeinerter Vernetzung wurde deutlich, dass hier neben der Nutzung von Symmetrieeigenschaften das größte Potenzial hinsichtlich Minimierung der Simulationszeiten liegt.

Schlussendlich kann gesagt werden, dass mittels Strömungssimulation komplexe Problemstellungen mit strömungsbedingten Wärmetransporten effizient optimiert werden können, sofern schon von Beginn an konsequent auf einen Kompromiss zwischen vertretbarer Ergebnisqualität und niedriger Simulationszeit geachtet wird.

6 References

- [1] J. Kopitz und W. Polifke. *Wärmeübertragung: Grundlagen, analytische und numerische Methoden*. Pearson Deutschland GmbH, 2009.
- [2] M. Casey und T. Wintersgerste. *ERCOFTAC Special Interest Group on Quality and Trust in Industrial CFD-Best Practice Guidelines*. European Research Community on Flow, Turbulence and Combustion, 2000.
- [3] M. Perić und S. Ferguson. *The advantage of polyhedral meshes*. CD-adapco, 2007.
- [4] J. F. Thompson, B. K. Soni und N. P. Weatherill. *Handbook of Grid Generations*. CRC press, 1999.
- [5] H. Schlichting und K. Gersten. *Boundary-layer theory*. Springer, 2000.
- [6] J. H. Ferziger und M. Perić. *Numerische Strömungsmechanik*. Springer DE, 2007.

Thermische Simulation von Bonddrähten in verpackten Chips unter Berücksichtigung der Draht-Package-Interaktion

Carl Christoph Jung ¹, Christian Silber ², Jürgen Scheible ¹

¹ Robert Bosch Zentrum für Leistungselektronik, Hochschule Reutlingen, Reutlingen, Deutschland

² Robert Bosch GmbH, Reutlingen, Deutschland

Carl_Christoph.Jung@Reutlingen-University.de

In der Mikroelektronik werden Chips häufig in Mold-Gehäusen verpackt. Die elektrischen Verbindungen vom Chip zu den Anschlussbeinchen des Gehäuses werden mit Bonddrähten realisiert. Für die Berechnung der Gleichgewichtstemperatur in einem Bonddraht bei konstantem Strom sowie von Temperaturverläufen bei transienten Strömen ist die herkömmliche FEM-Methode langsam und unhandlich. Daher wurde der Bondrechner entwickelt, der ein zylindersymmetrisches Ersatz-Modell für das Package in geeigneten mathematischen Gleichungen abbildet.

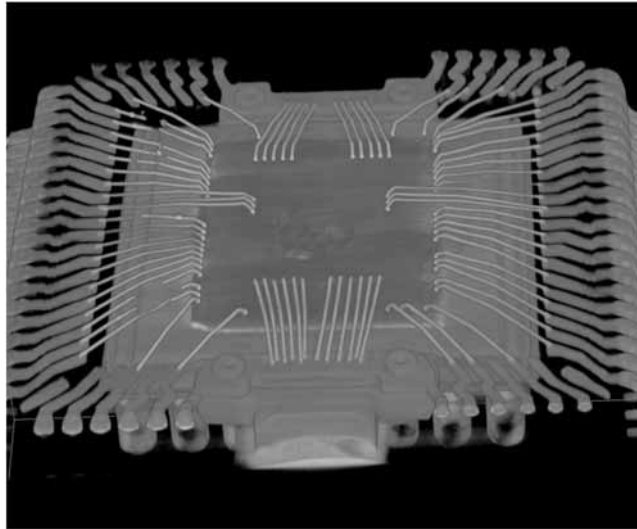
Im Gegensatz zum Bondrechner der ersten Generation [1], der auf den Gleichungen von [2] basiert, bietet ein neuer mathematischer Ansatz die Möglichkeit, eine endliche effektive Package-Größe, sowie einen endlichen Wärmeübergang zwischen Bonddraht und Mold-Masse zu berücksichtigen. Ebenso wurde die Berechnung der Interaktion von mehreren benachbarten Drähten verfeinert. Die Berechnung von beliebigen transienten Pulsformen mittlerer Länge wurde ebenfalls verbessert. Eine quadratische Komponente in der Temperaturabhängigkeit des spezifischen Widerstandes des Drahtmaterials kann jetzt ebenfalls berücksichtigt werden.

Die Ergebnisse wurden erfolgreich mit FEM-Berechnungen verglichen und die Geschwindigkeit der Berechnung ist um Größenordnungen schneller als mit kommerziellen FEM-Programmen.

Problemstellung

Problem:
Durchbrennen
eines oder mehrerer
Bonddrähte bei
elektrischer
Überlastung

Gesucht:
Berechnungstool
zur Dimensionierung
der Stromtragfähigkeit



3D X-ray image, copyright Robert Bosch GmbH

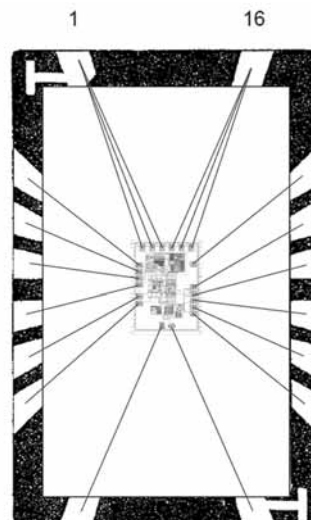


Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Eingangsgrößen

Eingangsgrößen aus
Bondplan:

- > Länge der Bonddrähte
- > Abstand der Bonddrähte (pitch)
- > Strombelastung pro Draht
- > Grundtemperatur Chip+Lead-frame



Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Historie des Bondrechners

Anwendung:

Dimensionisierung von Bonddrähten für verpackte Chips

(Bonddrähte sind in Moldmasse eingebettet)

Berechnung der Gleichgewichtstemperatur in einem Bonddraht bei konstantem Strom sowie von Temperaturverläufen bei transienten Strömen: Herkömmliche FEM-Methode langsam und unhandlich.

2000 Theoretischer Ansatz Nöbauer basierend auf zylindersymmetrischem Ersatz-Modell für das Package.

2012 Bondrechner der 1. Generation [A. Gerlach et al., 2012], auf den Gleichungen von [G. T. Nöbauer und H. Moser, 2000] basiert.

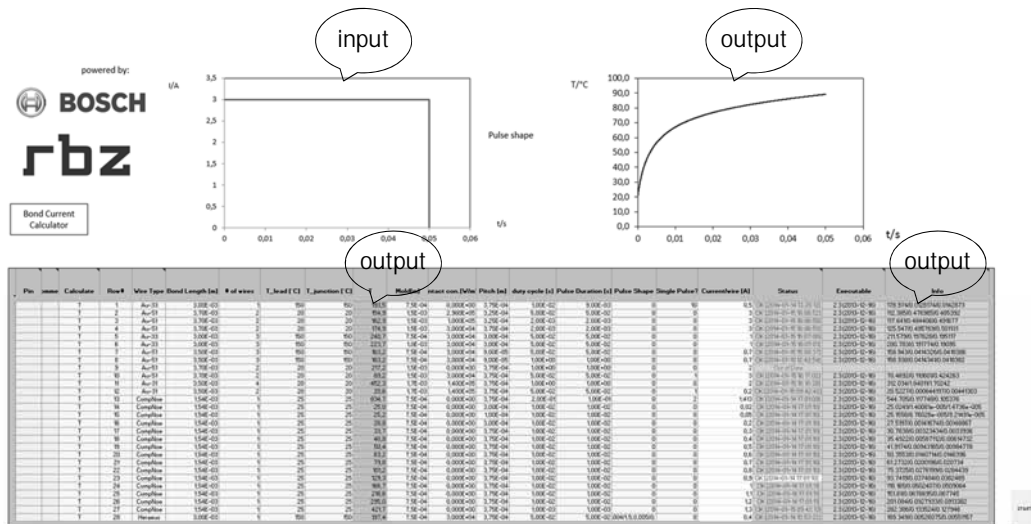
2013 Bondrechner der 2. Generation mit neuem mathematischen Ansatz: endliche effektive Package-Größe, endlicher Wärmeübergang zwischen Bonddraht und Mold-Masse. Interaktion von mehreren benachbarten Drähten verfeinert. Beliebige transiente Pulsformen mittlerer Länge verbessert.

Status: Erfolgreicher Vergleich mit FEM-Berechnungen, Geschwindigkeit der Berechnung um Größenordnungen schneller als mit kommerziellen FEM-Programmen.



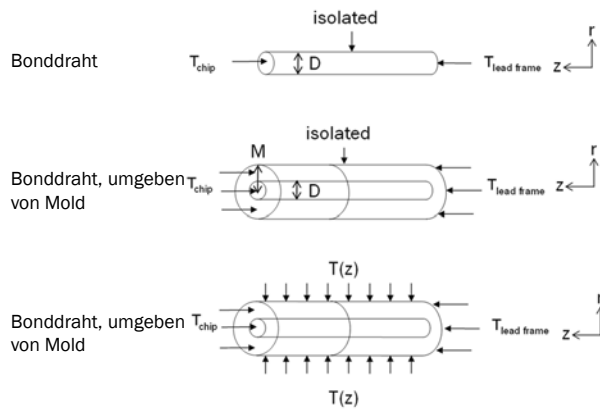
Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Oberfläche des Tools in Microsoft Excel®



Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Zylindersymmetrisches Modell



Modell

Longitudinal in Kontakt mit Wärmebad

verschiedene Spezialfälle:

- ohne Mold, im Vakuum

- Mold radial in Kontakt mit Wärmebad, oder isoliert



Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

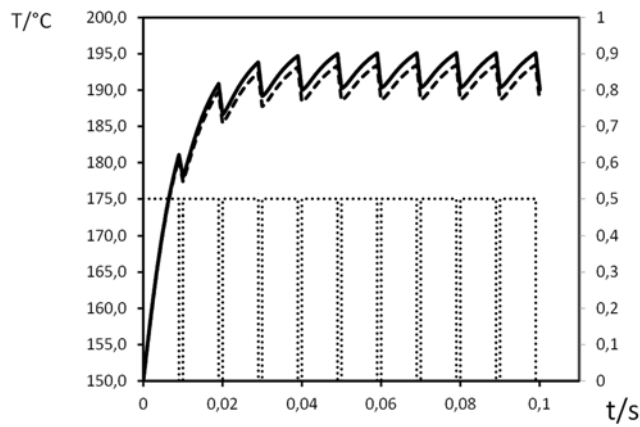
Berücksichtigte spezielle Effekte

- Temperaturabhängigkeit der elektrischen Leitfähigkeit des Drahtmaterials
- Temperaturabhängigkeit der thermischen Leitfähigkeit des Drahtmaterials
- Thermischer Kontaktwiderstand zwischen Draht und Mold-Material
- Endliches Package mit effektiver Größe M
- Wärmeerzeugung im Draht durch unterschiedliche Stromprofile (Zeitauflösung 1 μ s)
- Mehrere Drähte und deren Interaktion über die Mold-Masse



Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Simulationsergebnis für transienten Input



Temperatur-Transienten-
Berechnung möglich unter
Berücksichtigung von
Nichtlinearitäten

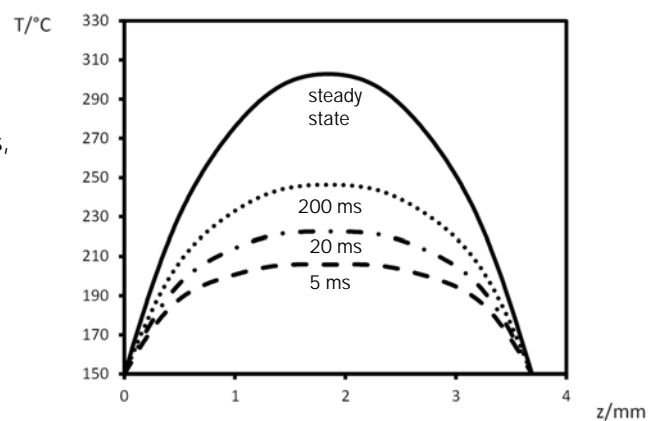
: Temperaturkoeffizient der thermischen Leitfähigkeit
Gestrichelt: 0 . Durchgezogen: nicht verschwindend. Gepunktet: Strom.



Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Temperaturprofil entlang des Drahtes

Anregung:
Stromsprung bei $t = 0$ s,
Dann konstanter Strom



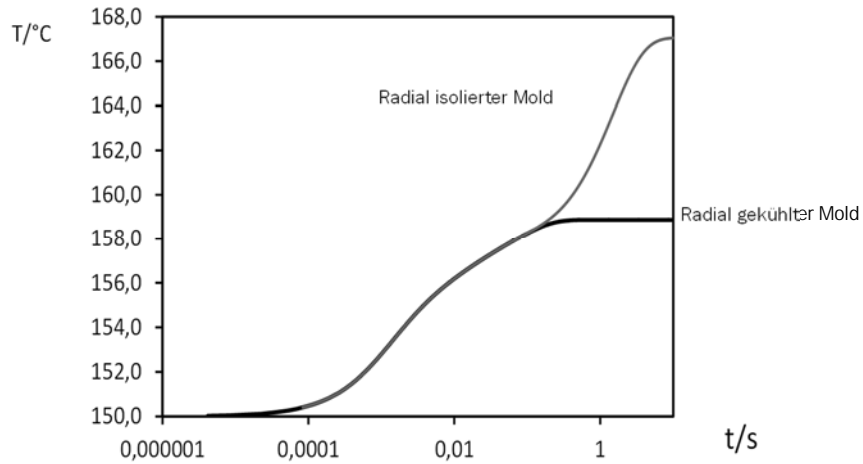
Mit einem begrenzten Mold-Durchmesser, radial isoliert



Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

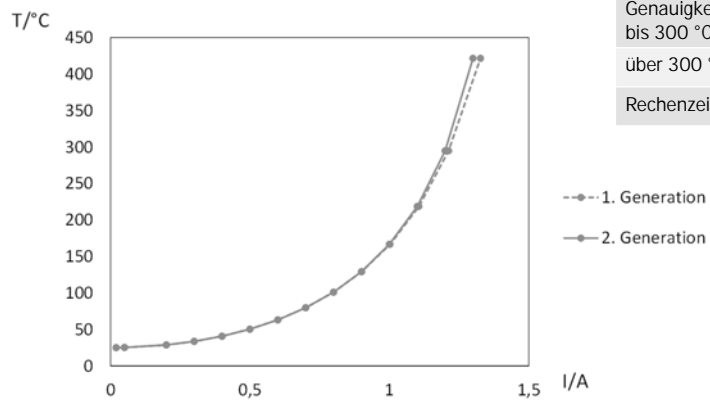
Zeitlicher Anstieg der Temperatur

Maximums-Temperatur im Draht



Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Vergleich Bondrechner 1. Generation/2. Generation

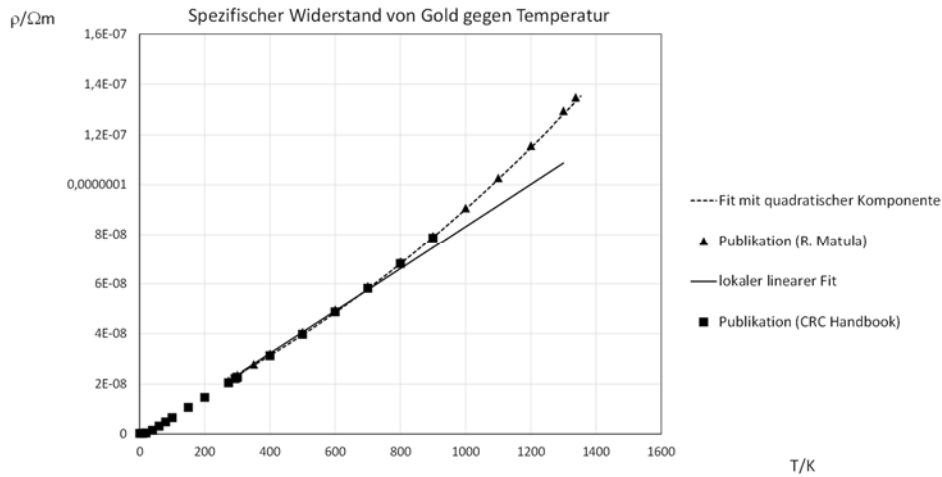


	1. Generation	2. Generation
Genauigkeit bis 300 °C	++	++
über 300 °C	+	++
Rechenzeit	++	++

Fazit: für Bond-Draht im Vakuum sehr vergleichbare Resultate. oberhalb 300 °C: 2. Generation genauer

Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Spez. Widerstand von Gold als f(T)



Ab 700 °C quadratische Komponente notwendig

Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

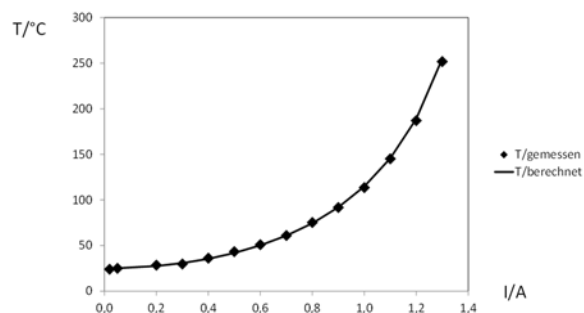
Vergleich mit Messung

Wie groß ist der Temperaturkoeffizient des spezifischen Widerstands von Gold?

Direkt gemessen (rbz, bis 175 °C): 0,0038 1/K; CRC Handbook (lokaler Fit: Bosch): 0,0039 1/K; R. Matula: 0,0037 1/K

Mittlere Temperatur eines bestromten Drahts (Länge 1,54 mm, Durchmesser 25 µm) in Luft verglichen mit theoretischen Werten (therm. Leitfähigkeit Luft: Wert bei 100 °C)

Hieraus indirekt auch
Temperaturkoeffizient
bestimmbar:

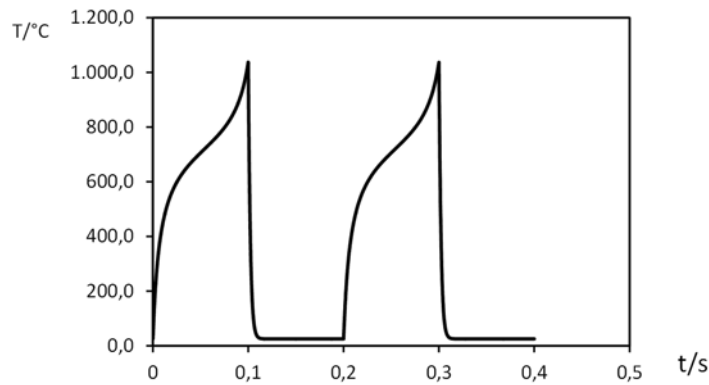


Resultat: lokal linear: 0,0039 1/K, bezogen auf 20 °C

Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Selbstverstärkung der Temperatur

Selbstverstärkung für Bond-Draht im Vakuum bei hohen Temperaturen aufgrund der Temperaturkoeffizienten von spez. Widerstand und thermischer Leitfähigkeit bei Rechteck-Pulsen

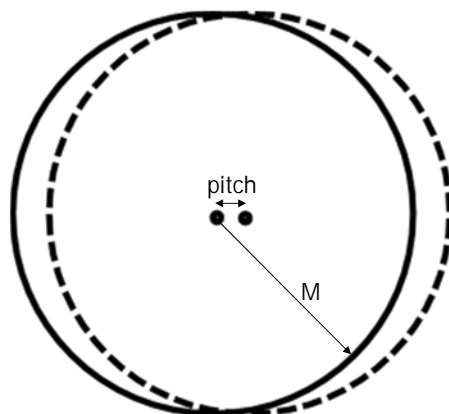


Quadratischer Ansatz bildet thermal runaway besser ab



Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Parallel bestromte Drähte erwärmen sich gegenseitig



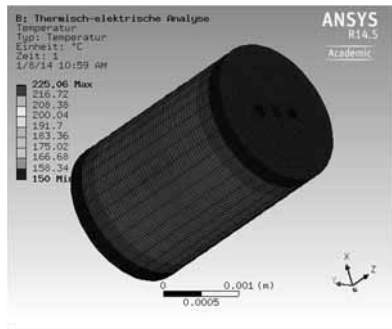
Gegenseitige Erwärmung wird angenähert durch:
Jeweils lineare Superposition der Mold-Temperaturen trotz unterschiedlicher Randbedingungen, wenn $\text{pitch} \ll M$. In den Drähten nichtlineare Superposition.



Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Vergleich mit FEM

#	D/ μm	l/mm	T (Chip) / °C	M/mm	h_c	pitch/ μm	I/A	Tmax (Draht) / °C Unsere Simulation	Tmax (Draht) / °C FEM	Berechnungszeit FEM
2	51	3,7	20	1,5	ideal	325	3	155	157	6h
2	51	3,7	20	1,5	1×10^5	325	3	163	165	7,5h
2	51	3,7	20	1,5	3×10^4	375	3	175	175	11h
3	33	3	150	0,75	3×10^4	300	1	241	241	1,8h
3	33	3	150	1	3×10^4	300	1	224	225	1,5h
3	51	3,5	150	0,75	3×10^4	90	0,7	163	163	0,8h



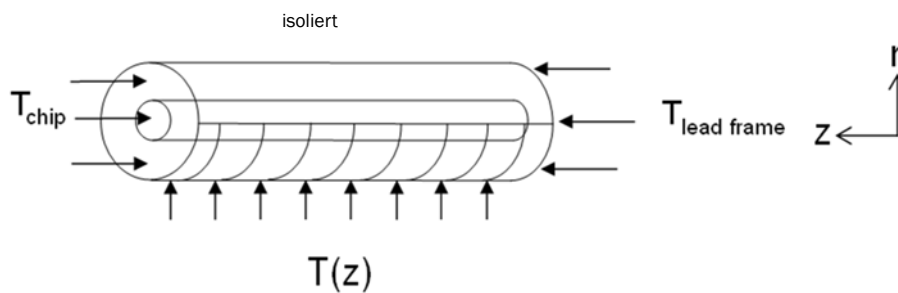
Summarische Berechnungszeit: 37 sec 29 h

Bondrechner 2. Generation ist für mehrere Drähte über drei Größenordnungen schneller als FEM



Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Ausblick: die „Dachrinne“



Ziel: Bessere Abbildung der thermischen Umgebung
unten: Leadframe, oben: Luft

und außerdem: Einfluss des Skin-Effekts



Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Zusammenfassung

Software Tool für die schnelle und exakte Berechnung des Wärmetransfers von einem Bonddraht zu einem zylindrischen endlichen Mold-Package

- Berechnung basierend auf **numerischer Integration von Differential-Gleichungen**
- Erfassung von **nicht-idealem Wärme-Transfer** vom Draht zum Mold
- Erfassung von **beliebigen Bond und Mold-Materialien, Drahtlänge, -durchmesser**
- Erfassung von beliebigen transienten Stromprofilen -> **beliebige Pulsformen**, die industriell relevant sind, **können innerhalb von Sekunden bis Minuten berechnet werden**
- **Erfolgreiche Validierung** des Tools durch Vergleich mit finite-Elemente-Simulationen und Messungen



Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Vielen Dank

für die Aufmerksamkeit

und an

Michael Ebli (rbz) für das Bonden,
Sebastian Buck (rbz) für Messungen,
Daniel Marolt (rbz) für das Adaptieren der Excel-Oberfläche,
sowie Bernhard Wicht (rbz) und Martin Pfost (rbz) für die Benutzung
ihrer Labors

Die Entwicklung des Bondrechners wurde in Zusammenarbeit mit der
Robert Bosch GmbH, Bereich Automotive Electronics, Reutlingen
durchgeführt.



Dr. rer. nat. Carl Christoph Jung, Robert-Bosch-Zentrum für Leistungselektronik

Literaturverzeichnis

- [1] A. Gerlach, D. Marolt und J. Scheible. *Der Bond-Rechner – ein Werkzeug zur Dimensionierung von Bonddrähten*. 6. GMM/GI/ITG-Fachtagung „Zuverlässigkeit und Entwurf“, Bremen, Germany, 25.-27.09.2012, GMM-Fachbericht 73, VDE-Verlag, 2012, ISBN 978-3-8007-3445-0, ISSN 1432-3419, S. 35-38, 2012.
- [2] B G. T. Nöbauer und H. Moser. *Analytical Approach to Temperature Evaluation in Bonding Wires and Calculation of Allowable Current*. IEEE Trans. Adv. Packaging, vol. 23, S. 426-435, 2000.
- [3] R. A. Matula, *Electrical Resistivity of Copper, Gold, Palladium and Silver*. J. Phys. Chem. Ref. Data, Vol. 8, S. 1147-1298, 1979.
- [4] Lide, David R. *Handbook of Chemistry and Physics. 75th Edition*. New York: CRC Press, S.11-41, 1996-1997.

Integrierte Analyse von Modelica-Modellen mit MapleSim und Maple

Dr. Johannes Friebe¹, Thomas Richard¹,

¹Maplesoft Europe GmbH

jfriebe@maplesoft.com

Wir demonstrieren neue Features in MapleSim 6.3 zu den Themengebieten Modelica und Analyse in Maple, außerdem Anbindung an Noesis Optimus.

1 Einleitung

Die aktuelle MapleSim-Version 6.3 bringt neue Features in Bezug auf Modelica und erweiterte Analyse-Optionen für alle MapleSim-Modelle auf Maple-Ebene sowie umfangreiche Optimierungsmöglichkeiten mit dem Produkt Optimus unserer Partnerfirma Noesis.

1.1 Neue Modelica-Features

Neben einigen Updates der Code-View-Ansicht für Modelica (Annotationen ein-/ausblenden, Zeilenumbruch, Zeilennummern, usw.) wurde die symbolische Vorverarbeitung aller Modelle beschleunigt, was sich insb. auf benutzerdefinierte Komponenten auf Modelica-Basis auswirkt. Hinzu kommt die Unterstützung weiterer Sprach-Features von Modelica. Mit einem neuen Befehl des MapleSim-Pakets kann man direkt aus Maple heraus Modelica-Code erzeugen. Außerdem wurde in Zusammenarbeit mit Modelon AB die *Hydraulics Library* auf MapleSim (ab Version 6.2) portiert.

1.2 Modell-Analyse auf Maple-Ebene

Für alle MapleSim-Modelle werden automatisch interaktive Maple-Worksheets generiert anhand von *Templates*. Sie umfassen die Themen Gleichungs-Extraktion und -verarbeitung, Mehrkörpermechanik, Sensitivitätsanalyse, Linearisierung, Code-Generierung (insb. für ANSI C), Regelungstechnik für lineare Systeme (Übertragungsfunktion, Zustandsraum-Darstellung, Bode-Plots, Wurzelortskurven, usw.), Monte-Carlo-Simulation Parameter Sweep, Optimierung, und andere. Alle diese vorgefertigten Templates sind vom Benutzer an eigene Aufgabenstellungen anpassbar, und mithilfe der MapleSim-API-Befehle kann er auch von Grund auf eigene Worksheets erstellen. Befehle wie *MonteCarlo* und

ParameterSweep arbeiten per Default multi-threaded, nutzen also die Parallelverarbeitung moderner Prozessoren.

Die Vorstellung dieser Templates und API-Befehle stellt den Hauptteil des Vortrags dar.

1.3 Kopplung mit Noesis Optimus

Ebenso wie Maplesoft gehört die belgische Firma Noesis Solutions seit einiger Zeit zur Cybernet-Gruppe. Ihr Kernprodukt ist die *Optimus*-Suite für industrielle Optimierungsaufgaben. Für MapleSim-Anwender hat dies zwei Auswirkungen: Mit Maple 17 wurde die *Global Optimization Toolbox* auf die Methoden OptimusDEVOL („differential evolution“) und OptimusEGO (hybride Methode mit interpolierenden Response Surface Modellen) umgestellt. Diese Toolbox bietet globale Methoden für skalare Zielfunktionen. Benötigt man hingegen Mehrgrößen-Optimierung, statistische Versuchsplanung / Design of Experiments, Unsicherheits-Modellierung und ähnliches, so bietet sich die Kopplung von Maple und MapleSim mit dem Vollprodukt Optimus an. Hierzu stellen wir exemplarisch die simultane Optimierung von Verbrauchs- und Leistungsparametern aus einem Fahrzeugmodell vor.

2 References

- [1] MapleSim User's Guide, © Maplesoft 2013, ISBN 978-1-926902-32-6
- [2] Maple User Manual, © Maplesoft 2013, ISBN 978-1-926902-23-4
- [3] Maple Programming Guide, © Maplesoft 2013, ISBN 978-1-926902-25-8

- [4] <http://www.maplesoft.com/products/maplesim/new/index.aspx>
- [5] <http://www.noesissolutions.com/Noesis/>

Using Functional Mock-up Interface in Software-in-the-Loop Simulation Applications

Michael Seibt

Dassault Systemes Deutschland GmbH

michael.seibt@3ds.com

Due to lack of standards for both, modeling languages and simulation interfaces, model exchange and consequently the reuse of simulation models was barely possible. FMI (Functional Mock-up Interface), introduced in the ITEA2/MODELISAR project, is an evolving standard that now allows connecting simulation models from different sources and achieving a virtual system integration. This paper illustrates how based on FMUs (Functional Mock-up Units), AUTOSAR Software Components including RTE (Run-time Environment) and OS can be connected with a Modelica plant model running a closed-loop simulation.

Keywords:

Functional Mock-up Interface (FMI), MODELISAR, AUTOSAR, Software-in-the-Loop (SiL), Co-Simulation, Model Exchange

What is FMI?

Due to lack of standards for both, modeling languages and simulation interfaces, model exchange and consequently, the reuse of simulation models were barely possible. Functional Mock-up Interface (FMI), introduced in the ITEA2/MODELISAR project, is an evolving standard that now allows connecting simulation models from different sources and achieving virtual system integration.

FMI provides a standardized simulation interface supporting two dedicated use cases:

- **FMI for Model-Exchange**

In this use case a FMU is exported from a simulation environment without a solver. This FMU is then imported into a Master Simulator which provides the solver to all FMUs participating in the simulation.

- **FMI for Co-Simulation**

In contrast to FMI for Model-Exchange, in this use case the FMU is exported with its solver. Once imported into a Master Simulator each FMU will use its own solver. The Master Simulator provides a simulation backplane handling the synchronization and data exchange between all FMUs.

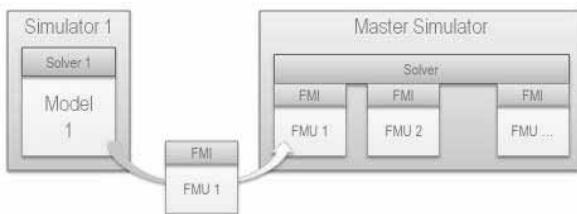


Fig.1 FMI for Model-Exchange

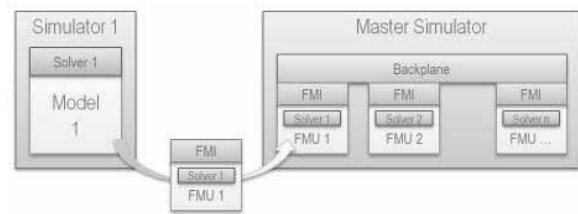


Fig.2 FMI for Co-Simulation

A FMU (Functional Mock-Up Unit) is a package (zip-file) containing different information, such as the model (source code or library), the model description in terms of signal names, attributes, type, unit definitions and documentation.

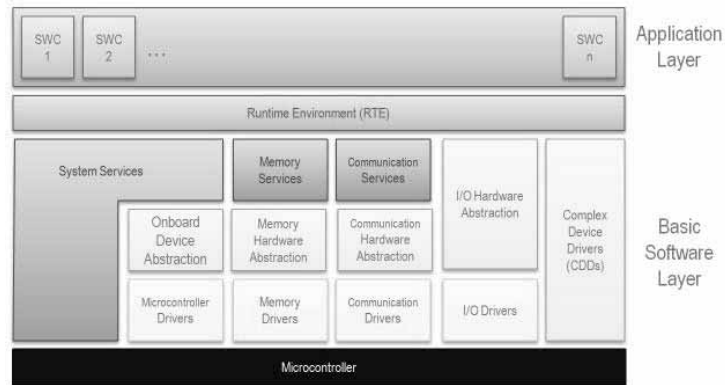
Fig.3
Sample FMI Model Description of an
AUTOSAR Software Component

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<fmiModelDescription
  fmiVersion="1.0"
  modelName="SwcClock"
  modelIdentifier="SwcClock"
  guid="e2007bbc-dc69-464c-9075-79c3fdb41b48"
  description="Generated from AUTOSAR model in release 3.X for Model-Exchange"
  author="Michael Seibt"
  version="1.0"
  generationTool="AUTOSAR BUILDER ASim"
  generationDateAndTime="Tue Jan 28 10:17:34 CET 2014"
  variableNamingConvention="structured"
  numberOfContinuousStates="0"
  numberOfEventIndicators="0">
  <TypeDefinitions>
    <Type name="T_Boolean">
      <BooleanType />
    </Type>
  </TypeDefinitions>
  <DefaultExperiment startTime="0.0"/>
  <ModelVariables>
    <ScalarVariable
      name="SwcClock_Out_StatusClock"
      valueReference="1"
      description=""
      variability="discrete"
      causality="output">
      <Boolean declaredType="T_Boolean" />
    </ScalarVariable>
  </ModelVariables>
</fmiModelDescription>
```

What is AUTOSAR?

AUTOSAR (AUTomotive Open System ARchitecture) is an automotive standard which defines a layered architecture, separating hardware, application software and basic software through standardized interfaces. The interface between Application Software Components and between Application Software and Basic Software (BSW) is handled through the Runtime Environment (RTE), which implements different types of communication mechanisms.

Fig.4
AUTOSAR Layered Architecture



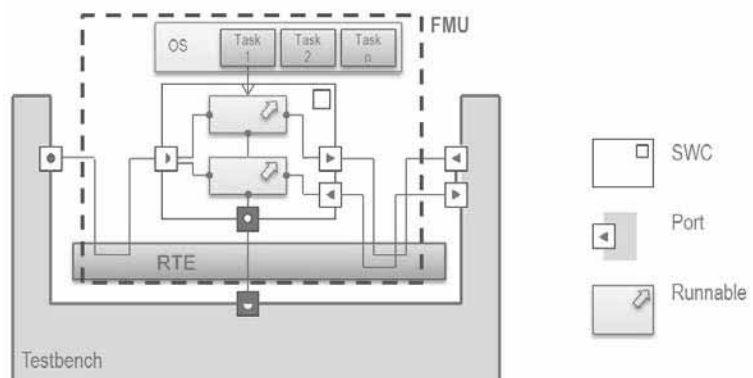
Software-in-the-Loop (SiL) Simulation of AUTOSAR ECUs

In the software development process, unit test is a quiet common way of testing individual Software Components against the specification by applying pre-defined test vectors. To test AUTOSAR Software Components and Compositions at a satisfactory level, it is important to take the AUTOSAR OS and the RTE (Runtime Environment) into account.

- **AUTOSAR OS**
The AUTOAR OS determines (with the RTE_Event-to-Task Mapping) the software scheduling in terms of execution period and execution order
- **RTE**
The RTE implements different communication mechanisms between Software Components and between Software Components and Service Components, such as implicit/explicit Sender/Receiver communication, queued/non-queued Sender/Receiver communication, synchronous/a synchronous Client/Server communication, etc.

ASim (Autosar SIMulation) from Dassault Systemes allows simulating a virtual ECU on VFB (Virtual Function Bus) Level also considering EcuM (ECU State Manager), OS (Operating System) and RTE (Runtime Environment). Basic Software behavior can be emulated through the Testbench.

Fig.5
ASim – Simulation Abstraction of an
AUTOSAR Software Component



However, unit test is an “open-loop” test where reasonable test vectors must be defined carefully. Timing effects and delays, caused by the surrounding environment, can only be roughly estimated through timing offsets in the test vectors.

Software-in-the-Loop, in contrast, is performed by a “closed-loop” simulation in which the software is connected with the virtual plant model. Only few test vectors must be defined in that case, as most input values of the software are provided by the virtual plant model in the “real” time. This adds an enormous value in software-testing, which is why SiL simulation is a common method in industrial development approaches.

Software is usually designed in different tools than the physical parts of the system. But due to proprietary formats, those tools can neither be easily connected nor import/export models from one to another. FMI's purpose is to address this problem.

By exporting the Application Software Component including OS and RTE as an FMU, it can be easily integrated into every simulation environment supporting the FMI standard. The RTE as part of the FMU plays here the role of the “solver”. In case an exported FMU also contains calibration parameters, those can be accessed as normal model parameters. For instance by applying a parameter sweep on the calibration parameters, during simulation one can perform online software calibration.

In following example of an Air-conditioning System, the controller is implemented as an AUTOSAR Software Component and tested in ASim.

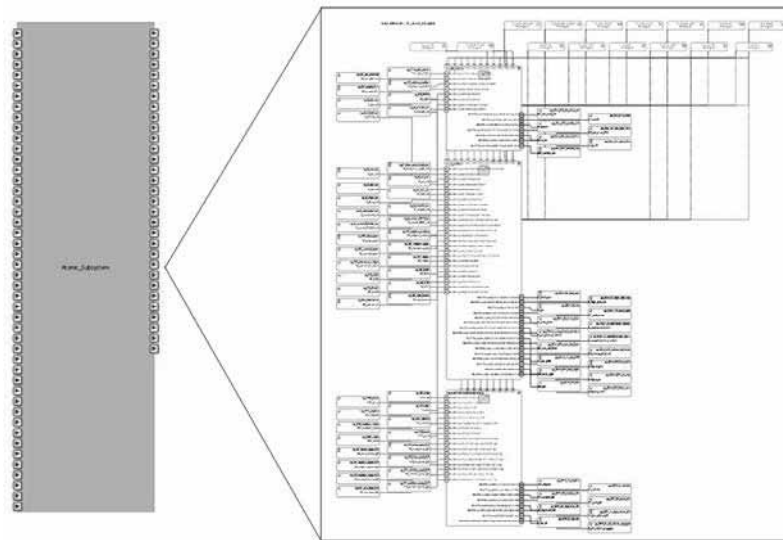


Fig.6 AUTOSAR Software Component and Internal Behavior of the Controller

In a next step the Controller Software Component was exported as a FMU and integrated into an Air-conditioning plant model which is modeled in the Modelica-based simulation tool Dymola.

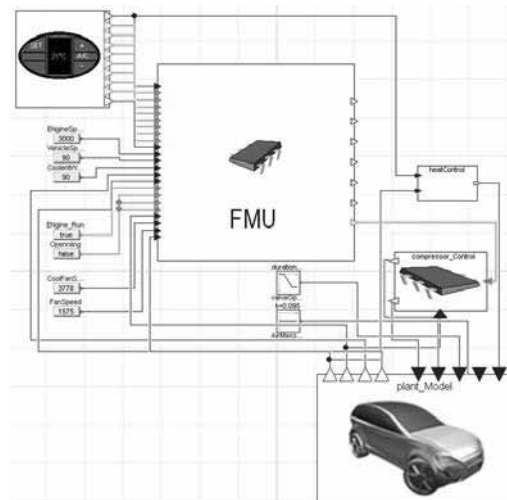
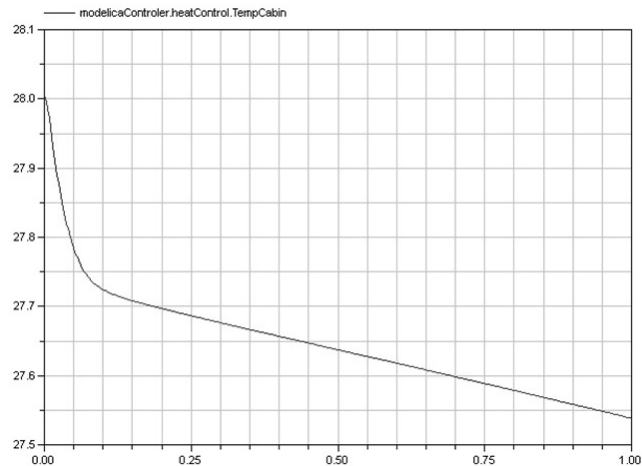


Fig.7
Overall Air-conditioning System
in Dymola

The complete model of the AC-System considers the impact of the plant and the environmental changes such as the ambient temperature. This allows a virtual verification and the tuning of the software controller.

Fig.8
Simulation Result –
controlled Cabin Temperature



FMI and Client/Server Communication

Client/Server communication is used quite extensively in AUTOSAR software development. Typical use cases are master SW components providing a dedicated service to other application SW components, data exchange between sensor/actuator SW components and application SW components or between Service SW components which are part of the basic software stack.

One limitation of FMI today is the lack of support for Client/Server communication. This shortcoming will be addressed with FMI version 2.1 as “physical connectors”. In the meantime this can be worked around by either defining the model scope of a FMU in a way, that it encloses the Client/Server communication within the FMU or by replacing a Client/Server port by a combination of an input and output port (provided and required port with a sender-receiver interface) to mimic the same behavior.

Conclusion

The FMI standard allows virtual system integration by connecting simulation models coming from different sources. This enables not only model exchange between Tier1 and OEM but also within the same organization. Different departments with historically different roles can now exchange on model level.

AUTOSAR Software Components developed by Software developers (at OEMs or Tier1s) can be reused by the system integrator to perform a system simulation or Software Components can be tested in a broader scope by applying Software-in-the-Loop tests.

References

- [1] www.modelica.org
- [2] www.autoar.org
- [3] www.fmi-standard.org
- [4] www.dymola.com
- [5] Functional Mockup Interface (FMI), General Standard for Model Exchange and Simulator Coupling, Linköping University, 2011-02-10, Peter Fritzson

Physikbasierte Simulation im Anlagenbau

Dr. Stefan Gulan¹, Dr. Ulrich Odefey¹, Dr. Thomas Bär², Herbert Beesten³,

Holger Hämmerle⁴, Bernd Kärcher⁵, Dr. Matthias Riedl⁶

¹TWT GmbH, ²Daimler AG, ³tarakos GmbH,

⁴Rücker EKS GmbH, ⁵Festo AG & Co KG, ⁶ifak Magdeburg e.V.

{stefan.gulan,ulrich.odefey}@tw-t-gmbh.de

Die Realisierung von Produktionsanlagen wird in zunehmendem Maße durch Simulationen unterstützt. Die Simulation erstreckt sich dabei über Materialfluss, Kommunikation, Steuerungslogik und graphische Darstellung. Diese Methodik ist als „virtuelle Inbetriebnahme“ bekannt. Das Gesamtverhalten der Anlage kann dabei zeitsparend, kostengünstig und sicher am Rechnermodell überprüft werden. Allerdings werden physikalische Aspekte – insbesondere Kräfteaspekte wie Schwerkraft, Trägheit oder Reibung – wenig betrachtet, obgleich geeignete Methoden bereits aus anderen Gebieten, etwa der Unterhaltungselektronik, bekannt sind.

Im Rahmen des hier vorgestellten AVANTI-Projekts sollen die Möglichkeiten der virtuellen Inbetriebnahme um Kräfteaspekte als Einflussgrößen erweitert werden. Die Erweiterung um physikalische Aspekte zieht sich dabei durch die gesamte virtuelle Wertschöpfungskette; insbesondere Umfang, Notation und Austauschformate der Komponentenmodelle, deren Simulation, sowie die Möglichkeiten funktionaler Anforderungen und deren Testbarkeit werden dadurch beeinflusst. Ein Schwerpunkt liegt dabei auf der Anreicherung etablierter Lösungen um offene Standards, welche die relevanten physikalischen Aspekte bereits unterstützen. Damit zielt das Projekt darauf ab, die Aussagekraft der Modelle zu erhöhen, die Güte der derzeitigen Simulation zu verbessern und den Zugang zu der Methodik für eine Vielzahl industrieller Partner zu ermöglichen.

1 Einleitung

Produktionssysteme in Deutschland zeichnen sich durch einen hohen Automatisierungsgrad aus. Sowohl die Komponentenhersteller als Zulieferer, als auch die Betreiber der Produktionsstätten profitieren dabei von den hohen Qualitätsstandards in Deutschland, weil sie im hohen Maß in die Entwicklung solcher Produkte mit einbezogen sind und damit in der Regel eher am Markt eigene, hochwertige Produkte platzieren können. Ebenso sind für die Sicherung des Produktionsstandorts Deutschland eine flexible Gestaltung der Produktionsanlagen, eine Optimierung des „time to market“ und eine außerordentlich hohe Qualität der Produkte der Zulieferer und der OEMs als Endnutzer ausschlaggebend.

Die in den letzten Jahren von den OEMs vor allem im Automobilbau eingeführte Technologie der virtuellen Inbetriebnahme (VIBN) zielt in die Richtung, die Planungs- und Inbetriebnahmephasen der Produktionsanlage zu verkürzen und frühzeitig Fehler zu beseitigen. Das Forschungsvorhaben AVANTI möchte darauf aufsetzen und entscheidende Verbesserungen bei der Verhaltensbeschreibung der beteiligten me-

chatronischen Komponenten, als auch bei der Testmethodik innerhalb der virtuellen Inbetriebnahme ermöglichen.

1.1 Virtuelle Inbetriebnahme

Die virtuelle Inbetriebnahme wird vorwiegend im Automobilbau, für die Planung, Integration und Verifikation von Produktionsanlagen eingesetzt. Solche Anlagen besitzen eine hohe Komplexität, die sich aus dem Zusammenspiel verschiedener leittechnischer Elemente ergibt. Dies führt zu einer Vielzahl möglicher Fehlerquellen und -einflüssen, die bei der Inbetrieb- und Abnahme der Anlage aufgedeckt und beseitigt werden müssen – was mit hohen Kosten bzw. Zeitaufwand verbunden ist.

Einen – zumindest teilweisen – Ausweg bietet die Verlagerung des Testprozesses auf die Modellebene. Im Zuge der VIBN werden hierbei Elemente auf Feldebene und Sensor-/Aktorebene modelliert und mit realen Steuerungseinheiten (speicherprogrammierbaren Steuerungen, SPS) gekoppelt, was einem „hardware-in-the-loop“ (HIL) -Ansatz in Bezug auf die Steuerungseinheiten entspricht. Die SPS kann damit auf funktionale Korrektheit im Sinne ihrer

Automatisierungsfunktion sowie auf Robustheit gegen bestimmte simulierte Fehlereinflüsse getestet werden. Zur funktionalen Korrektheit gehören etwa die richtige Schrittabfolge, der korrekte Signalfluss, sowie das Einhalten von Taktzeiten, z.B. im Zusammenspiel mit der Dimensionierung der Feldelemente. Als Fehlereinfluss kann im Verhaltensmodell z.B. der Ausfall eines Sensors modelliert werden, auf den die SPS reagieren soll. Hierbei muss betont werden, dass die Verwendung einer realen SPS Echtzeitfähigkeit von der Simulationsumgebung fordert.

Der Methodik der VIBN soll im Rahmen des AVANTI-Projekts um zwei Aspekte erweitert werden, die eine signifikante Verbesserung der derzeitigen Simulations- und Testmöglichkeiten versprechen. Diese bestehen zum einen in der Berücksichtigung von Kraftgrößen, also der Erweiterung der Simulation um ungesteuerte Systemaspekte. Zum anderen soll die Testumgebung das modellbasierte Paradigma unterstützen, das auf dem Vorliegen einer formalisierten Systembeschreibung basiert.

In der vorliegenden Arbeit wird zunächst das Kernstück der VIBN – das *Anlagenmodell* – erläutert. Auf dessen Erweiterungen um physikbasierte Möglichkeiten im Rahmen des AVANTI-Projekts und die Implikationen an den Wertschöpfungsprozess wird nachfolgend eingegangen. Zudem werden die Grundlagen des modellbasierten Testens kurz erläutert.

1.2 Anlagenmodell

Das Modell, das im AVANTI-Projekt als Ausgangsbasis dient und im Zuge des Projekts erweitert werden soll, ist das *mechatronische Anlagenmodell*. Es besteht aus zwei interagierenden Teilen, dem (*kinematisierten*) *Geometriemodell* und dem *Verhaltensmodell*.

Das Geometriemodell beschreibt die räumlichen Ausdehnungen der Anlagenkomponenten und ihre Topologie (die beiden Aspekte werden zur Geometrie zusammengefasst), sowie der kinematischen Kopplung der Einzelteile. Im Geometriemodell können zudem Roboterkomponenten als gekapselte Teilmodelle betrachtet werden, da die Roboterprogramme in einem emulierten Controller tatsächlich ausgeführt werden, um das Verhalten der Roboter darzustellen. Die Elemente des Geometriemodells sind häufig bereits aus dem Planungsprozess bekannt oder können aus Komponentenmodellen der Lieferanten zusammengesetzt werden.

Das Verhaltensmodell simuliert das logische und zeitliche Ein-/Ausgabeverhalten der Anlage gegenüber der SPS. Zudem werden Kommunikation und Materialfluss zwischen den Komponenten modelliert und dem Geometriemodell zur Simulationsdarstellung mitgeteilt. Insbesondere das zeitliche Verhalten der Anlage, etwa die Ausführungszeit einzelner Arbeitsschritte, ist im Verhaltensmodell „hartkodiert“.

Sensoren und deren Signalreaktionen können in beiden Modellen dargestellt werden. So kann wahlweise eine Sensorreaktion nach einer vorgegebenen Zeit vom Verhaltensmodell gegeben werden **Fehler! Verweisquelle konnte nicht gefunden werden.**, oder das Signal wird vom Geometriemodell z.B. bei Durchlauf eines Guts durch einen aktiven Sensorbereich an das Verhaltensmodell weitergegeben [2]. In letztgenanntem Fall spricht man auch von einem *erweiterten Geometriemodell*.

Das Zusammenspiel zwischen SPS und Modellen in der VIBN ist in **Abbildung 1** dargestellt.

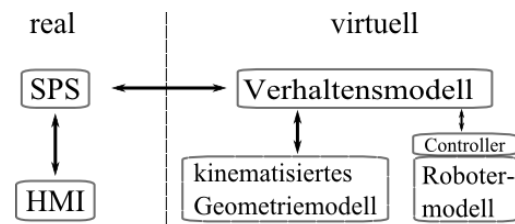


Abbildung 1: Modelle in der VIBN

1.3 Einfluss physikalischer Größen

Die tatsächliche Simulation des Anlagenverhaltens ist derzeit auf das gesteuerte Verhalten beschränkt. So können Komponenten z.B. auf kollisionsfreies Zusammenspiel untersucht werden, das Durchlaufen eines Sensorbereichs kann simuliert werden, woraufhin das dem Sensor zugeordnete Signal über das Verhaltensmodell an die SPS zurückgegeben wird (siehe Abbildung 1: Modelle in der VIBN).

Dynamische, bzw. ungesteuerte Einflüsse werden nicht berechnet. So wird z.B. der Transport auf einem Bandförderer dargestellt, indem das Gut entlang einer vorgegebenen Trajektorie bewegt wird. Diese entspricht zwar in Verlauf und Geschwindigkeit dem Förderband, bietet aber nicht die Möglichkeit, auf die abrupte Änderung der Bandgeschwindigkeit mit Reibungs- und Trägheitsphänomenen zu reagieren.

Die Integration des ungesteuerten Verhaltens in die Simulation ist eines der grundlegenden Ziele des AVANTI-Projekts.

2 Physikbasierte Simulation

Seit einigen Jahren werden im Rahmen der VIBN vermehrt physikbasierte Anlagensimulationen eingesetzt. Hierbei werden bestehende Simulationsmodelle um physikalische Aspekte erweitert, um ein physikalisch korrektes Anlagenverhalten abbilden zu können. In diesem Zusammenhang spielen vor allem Gesetzmäßigkeiten der Mechanik eine tragende Rolle, um Kinematik und Dynamik einzelner Bauteile, daraus zusammengesetzter Komponenten sowie der kompletten Anlage beschreiben zu können.

2.1 Herangehensweise

Grundlage der Beschreibung ist ein geometrisches Ersatzmodell der Anlage, welches aus vereinfachten 3D CAD Daten der einzelnen Bauteile aufgebaut ist. Es folgt eine Klassifizierung der Bauteile und Komponenten in stationäre und bewegliche Elemente sowie die Zuordnung verschiedener physikalischer Eigenschaften [2]. Eine große Rolle spielen Kenngrößen wie Masse, Schwerpunkt, Reibungseigenschaften, Elastizitätsmoduln etc. Das geometrische Modell ermöglicht außerdem die Erfassung und Berechnung von Kollisionen und deren Reaktionen. All diese Informationen ermöglichen eine vollständige physikalische Beschreibung und Simulation der Produktionsanlage.

Ein weiterer wichtiger Gesichtspunkt in diesem Zusammenhang ist die Echtzeitfähigkeit der Simulation. Wie bereits erwähnt erfordert die Verwendung einer realen SPS eine echtzeitfähige Simulation. Diese Bedingung wird vor allem durch zwei Maßnahmen erleichtert, nämlich die Vereinfachung der 3D CAD Daten (konkave Geometrien werden in konvexe Teilgeometrien aufgeteilt) als auch der weitgehende Verzicht auf Modellierung und Simulation biegeschlaffer Elemente. Der erste Schritt beschleunigt die Erfassung und Berechnung von Kollisionen, der zweite verhindert aufwendige Berechnungen mit Hilfe der Methoden finiter Elemente. Hinsichtlich des letzten Punktes gibt es vermehrt Anstrengungen im Bereich Algorithmik, um auch flexible Bauteile effizient in die Simulation miteinbeziehen zu können [3].

2.2 Sprachen zur physikbasierten Modellierung

Die Realisierung der oben beschriebenen Modellierung kann mit unterschiedlichen Software-Werkzeugen erreicht werden. Bis heute hat sich allerdings kein gemeinsamer Standard etabliert, mit dem physikalische Simulationen in der Anlagenplanung durchgeführt werden. Kernziele des Projektes AVANTI sind daher die Festlegung eines leistungsfähigen Beschreibungsformats für physikalische Simulationen, die Identifikation geeigneter Modellierungssprachen und Simulationsumgebungen sowie die Entwicklung notwendiger Schnittstellen zwischen Modellierungssprache und Simulationswerkzeug.

Eine abgeschlossene Entwicklungs- und Simulationsumgebung physik-basierter Simulationen bieten die beiden Softwarepakete MATLAB/Simulink und Modelica. Allerdings besitzen weder MATLAB/Simulink noch Modelica die gesamte Funktionalität, die für physikbasierte Simulationen von Produktionsanlagen wünschenswert wäre. So fehlt MATLAB/Simulink sowohl die Möglichkeit zur Nutzung eines offenen Standards als auch ein objekt-orientierter Ansatz [4]. Modelica hat den zusätzlichen Nachteil, dass es, im Gegensatz zu MATLAB/Simulink, nicht sehr verbreitet ist. Infolgedessen hat sich im Bereich Anlagenplanung ein dualer Ansatz durchgesetzt, welcher auch in AVANTI verfolgt und im Folgenden erläutert werden soll.

Die geometrische und physikalische Beschreibung individueller Bauteile und Komponenten geschieht in XML-basierten Sprachen, die Durchführung der eigentlichen Simulation wird in einer separaten Umgebung durchgeführt. Die derzeit am häufigsten verwendete Beschreibungssprachen werden in einem späteren Abschnitt behandelt.

Die Verwendung eines XML-basierten Formats bringt mehrere Vorteile [5]. Erstens ist XML ein offener Standard, d.h. er ist leicht zugänglich, weiterentwickelbar und einsetzbar. Zweitens unterstützt XML die Einbindung weiterer Dateiformate, z.B. CAD Daten. Drittens ermöglicht solch ein Format den plattform- und implementationsunabhängigen Austausch von Daten. Viertens existieren bereits XML Beschreibungen, welche die Integration physikalischer Aspekte erlauben und für AVANTI benutzt werden können. Fünftens akzeptiert eine Vielzahl der aktuellen Simulations-Umgebungen Datenbeschreibungen im XML-Format. Zusammenfassend lässt sich festhalten, dass XML-Formate ein flexibles, effizientes und ganzheit-

liches Beschreibungswerkzeug zur Anlagenplanung darstellen.

Im nächsten Abschnitt werden verschiedene Werkzeuge zur Durchführung physik-basierter Simulation vorgestellt. Hauptaugenmerk liegt hier auf sogenannten Physik-Engines.

2.3 Physik-Engines

Ein weiterer Ansatz zur physik-basierten Anlagensimulation besteht in der Verwendung von sogenannten Physik-Engines, welche vor allem in der Entwicklung von Computerspielen und der Computergrafik eingesetzt werden. Diese Werkzeuge sollen realistische Bewegungsabläufe vermitteln, indem die zugrunde liegenden physikalischen Prozesse simuliert und objektimmanente Eigenschaften berechnet werden.

Grundlagen der Simulation bilden die Gesetze der Physik der starren und deformierbaren Körper (inklusive Kollisionsdetektion) sowie der Fluidmechanik. Viele der heute erhältlichen Physik-Engines sind nahezu uneingeschränkt echtzeitfähig oder bieten gewisse Beschleuniger an. Aus diesen Gründen drängen sich Physik-Engines als Simulationswerkzeug für die VIBN den Anlagenplanern förmlich auf. In [6] findet man eine ausführliche Zusammenfassung und Evaluierung der gebräuchlichsten Physik-Engines. Bevor eine Physik-Engine jedoch als Ressource für die VIBN genutzt werden kann, muss diese in der Lage sein, die Beschreibung der Anlage zu importieren sowie deren physikalische Eigenschaften zu extrahieren und zu interpretieren. Auf Basis dieser Informationen wird dann ein Modell initialisiert und eine Simulation gestartet. Eine entscheidende Aufgabe im AVANTI Projekt wird also die Entwicklung und Implementierung einer flexiblen Schnittstelle zwischen Beschreibungs- und Simulationsumgebung darstellen.

3 AutomationML

Um eine Produktions- oder sonstige Anlage in einem Detaillierungsgrad zu beschreiben, der eine Simulation im Sinne der Projektziele erlaubt, müssen zumindest die folgenden Eigenschaften der, bzw. Zusammenhänge zwischen den Komponenten mit beschrieben werden:

- *Topologie*, d.h. die hierarchische Komposition der Anlagenkomponenten, die Schnittstellen zwischen diesen für Signal- und Materi-

alfluss, nach Möglichkeit angereichert mit Rolleninformation zur Abstraktion von der individuellen Komponente

- *Geometrie*, d.h. die räumlichen Ausdehnungen der Anlagenkomponenten und ihre relative Ausrichtung zueinander
- *Kinematik*, d.h. die Gelenke, über die Komponenten aneinander gekoppelt sind, sowie die resultierende (geometrische) Relativbewegung der Komponenten
- *Kinetik*, d.h. Kraftfelder, die an den Komponenten anliegen (insbes. Schwerkraft), Materialverhalten (Reibungskoeffizienten) und approximiertes Kollisionsverhalten

Im Anlagenbau wurde hierzu der Sprachstandard *AutomationML* erarbeitet [8], der letztlich eine Zusammenfassung folgender Teilsprachen ist: CAEX zur Beschreibung von Topologie, COLLADA zur Beschreibung von Geometrie, Kinematik und Kinetik, außerdem PLCOpen zur Beschreibung von Ablauf und Verhalten (z.B. Gantt- u. PERT-Charts, aber auch vollständige SPS-Programme). Alle diese Sprachen sind XML-basierte offene Standards.

Die Sprache CAEX fungiert gleichzeitig als „Hauptsprache“ von AutomationML, da sie Assoziationen der Elemente einer Hierarchie untereinander sowie mit beliebigen externen Objekten unterstützt. Insbesondere können Beschreibungen, die durch die AutomationML-eigenen Sprachen nicht ausdrückbar sind, in spezielle Dokumente ausgelagert und verlinkt werden. So kann z.B. komplexes Regelverhalten eines Glieds in einem MathML-Dokument beschrieben werden, welches dem entsprechenden Element in der Topologie zugewiesen wird.

Zusätzlich fasst AutomationML Komponenteneigenschaften, die in der Automatisierung besonders wichtig sind, in semantischen Bibliotheken zusammen. Solche Bibliotheken sind für Systemeinheiten, Schnittstellen und Rollen vordefiniert. Mit Rollenbibliotheken kann z.B. nach fertigungs- und leittechnischen Komponenten unterschieden werden. Weitere Bibliotheken können nach Bedarf hinzugefügt werden. Damit können Anlagen nach abstrakten Eigenschaften hoher Ebene zusammengestellt werden, bevor ein detailliertes Modell gebildet wird.

Aufgrund der aufgeführten Eigenschaften soll im AVANTI-Projekt AutomationML maßgeblich zum Einsatz kommen. Die Festlegung auf diesen offenen Standard soll dazu führen, dass die Komponentenmodelle herstellerübergreifend mit geringem Mehraufwand integrierbar sind.

4 Modellbasiertes Testen

Zur Sicherstellung der Korrektheit des SPS-Programms wird dieses (in der VIBN gegen das Anlagenmodell) getestet. Ein Testfall wird dabei üblicherweise aus einer Anforderung an die Anlage abgeleitet; z.B. wird die prosaisch formulierte Anforderung, dass ein Pufferbereich auf einem Bandförderer nicht überlaufen darf, in einen Test übersetzt, der den Überlauf zu erzwingen versucht. Die Übersetzung, eventuell auch die Durchführung und die Auswertung, der Testergebnisse geschieht manuell von einer Fachkraft. Dies macht den Testprozess fehleranfällig und teuer. Eine Alternative, die sich derzeit durchsetzt und auch im AVANTI-Projekt Anwendung finden soll, liegt im modellbasierten Testen, kurz MBT. Dieser Ansatz verspricht, die genannten Probleme zumindest teilweise zu beseitigen. Eine allgemeine Einführung in MBT findet sich in [9], ausgesuchte Aspekte werden in [10] vertieft.

Im Falle des MBT ist das *Anforderungsmodell* zentrales und namensgebendes Objekt. Es beschreibt Anforderungen und Spezifikationen in einer formalen Notation, die zur automatisierten Weiterverarbeitung, insbesondere der automatisierten Generierung von Testfällen, geeignet ist. Für Anforderungen, die z.B. in Temporallogik formulierbar sind, kann zudem mit *model-checking* – Techniken [11] streng verifiziert werden, ob das formale Anforderungsmodell diese Anforderungen wirklich als Verhaltensaspekte wiedergibt. Die Zusammenhänge sind in Abbildung 2 dargestellt.

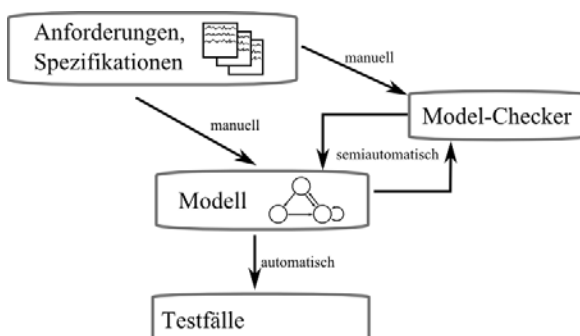


Abbildung 2: modellbasierte Testgenerierung

4.1 Anforderungsmodell

Die Erstellung des Anforderungsmodells ist wiederum einem menschlichen Experten vorbehalten, d.h. gegenüber dem manuellem Erstellen von Tests liegt die interaktive Arbeit hier an einer anderen Stelle. Jedoch bedingt die Erstellung des formalen Anforderungsmodells die implizite Überprüfung der Anforderungen und Verständlichkeit – ohne die die gewissenhafte Modellierung nicht möglich sein kann. Zudem lassen sich Inkonsistenzen der Anforderung am Modell leichter finden, wie z.B. widersprüchliche Angaben zum Eingabeverhalten des Systems. Hier muss bemerkt werden, dass der Vorgang der Modellkonstruktion hohe Qualifikationsanforderungen an den Modellierer stellt, der im besten Fall formal geschult ist und auch domänenspezifisches Wissen einbringt.

Als Modellierungsformalismen, die sowohl die automatisierte Testableitung als auch die Verifikation des Modells gegen formallogische Anforderungen unterstützen, bieten sich zustandsbasierte Systeme, wie statecharts oder Petri-Netze an. Diese sind als Modelle ereignisgesteuerter Systeme etabliert und können um verschiedene Aspekte erweitert werden, ohne die genannten vorteilhaften Eigenschaften zu verlieren.

4.2 Testgenerierung

Bei einem Zustands-/Transitionssystem als zugrundeliegendem Modell läuft die Testgenerierung auf das Auffinden einer Pfadmenge in diesem System hinaus, die alle Zustände und alle Transitionen des Systems abdeckt. Eventuell können hier Sonderbedingungen wie zyklische Pfade berücksichtigt werden. Diese Pfade lassen sich als Kommunikationsfolgen interpretieren, welche die reale SPS implementieren muss.

5 Zusammenfassung

Zwei wesentliche Aspekte, die im Rahmen des AVANTI-Projekts für die virtuelle Inbetriebnahme angegangen werden, sind die Anreicherung der Simulation um physikalische Aspekte und die Verwendung eines Modells zu Testfallerzeugung. Diese Verbesserung sollen die Realitätsnähe der Simulation und die Testabdeckung der Steuerungslogik erhöhen.

Danksagung: Diese Arbeit wurde finanziert durch Mittel des ITEA2-Förderprogramms im Rahmen des AVANTI-Projekts (Förderkennzeichen 01IS13018E).

Literaturverzeichnis

- [1] M. Bergert, S. Höme und L. Hundt. *Verhaltensmodellierung für die virtuelle Inbetriebnahme*. etz, Heft 9, S. 2-9, 2010. VDE Verlag, 2010.
- [2] A. Strahilov, J. Ovtcharova und T. Bär. *Development of the physics-based assembly system model for the mechatronic validation of automated assembly systems*. Proc. 2012 Winter Simulation Conference
- [3] C. Wienns. *Simulation von Kabeln und Schläuchen – Anforderungen und Einflüsse*. Dissertation, Universität Koblenz-Landau, Deutschland, 2008
- [4] L. Ollinger. *Standardisierte Verhaltensmodellierung mechatronischer Betriebsmittel für die Virtuelle Inbetriebnahme im automobilen Karosseriebau*. Diplomarbeit, Technische Universität Kaiserslautern, Deutschland, 2009.
- [5] XML Homepage: <http://www.w3c.org/XML>
- [6] A. Boeing und T. Bräunl. *Evaluation of real-time physics simulation systems*. GRAPH-ITE'07, ACM, 2007.
- [7] M. Bergert, J. Kiefer, S. Hörme und C. Fedrowitz. *Einsatz der Virtuellen Inbetriebnahme im automobilen Karosseriebau – Ein Erfahrungsbericht*. 9. Magdeburger Maschinenbau-Tage, S. 388-397, 2009.
- [8] R. Draht. *Datenaustausch in der Anlagenplanung mit AutomationML*. Springer, 2010.
- [9] I. K. El-Far und J. A. Whittaker. *Model-based Software Testing*. In Encyclopedia of Software Engineering, Wiley, S. 825-837, 2001.
- [10] M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker und A. Pretschner. *Model-Based Testing of Reactive Systems. Advanced Lectures*. LNCS 3472. Springer, 2005.
- [11] C. Baier und J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.

Fallstudien zur Modellierung physikalischer Systeme in der experimentellen Archäologie

Johannes Tanzler¹, Philipp Pichler^{1,2}, Bernhard Heinzl^{2,3}, Hans Reschreiter⁴, Kerstin Kowarik⁴,
Felix Breitenecker¹

¹Institut für Analysis und Scientific Computing, TU Wien

²dwh GmbH Simulation Services, Wien

³Institut für Rechnergestützte Automation, TU Wien

⁴Naturhistorisches Museum Wien

johannes.tanzler@tuwien.ac.at

Als Ergänzung zu klassischen experimentellen Methoden kann auch Computersimulation beitragen bessere Einblicke in verschiedene Aspekte der Archäologie zu gewinnen. In Zusammenarbeit zwischen TU Wien und dem Naturhistorischen Museum Wien wurden mehrere Fallstudien von physikalischen Systemen aus dem prähistorischen Salzbergbau in Hallstatt (Österreich) untersucht. Die Ergebnisse aus diesen Simulationen liefern neue Informationen und quantitative Anhaltspunkte und tragen dazu bei das Verständnis über die prähistorischen Arbeitsprozesse und Werkzeuge zu vertiefen.

1 Einleitung

Der prähistorische Salzbergbau in Hallstatt (Österreich) ist für die Archäologie von großem Interesse. Salzbergbauaktivitäten werden in der Bronzezeit auf 1458-1245 v. Chr. datiert [1]. Die große Menge an archäologischen Funden in den ehemaligen unterirdischen Abbaukammern, u.a. von technischer Ausrüstung und organischen Materialien (Bauholz, Werkzeuge, Bastseile, Felle, etc.) und ihr außerordentlich guter Konservierungszustand durch die konservierenden Eigenschaften des Salzes erlauben es Arbeitsprozesse und Abbaumethoden zu rekonstruieren und so Informationen über das damalige technische Know-how zu gewinnen [2]. Diese Untersuchungen deuten darauf hin, dass der Bergbau in einer sehr effizienten, fast schon industriellen Art und Weise organisiert war und Werkzeuge mit einem hohen Grad an Spezialisierung und Funktionalität zur Verfügung standen. Das Salz wurde in unterirdischen Abbaukammern (vgl. Abbildung 1) mit speziellen Bronzepickeln gebrochen, in Säcke gefüllt und mittels Seilzügen über vertikale Schächte an die Oberfläche transportiert [3].

Zur Untersuchung von physikalischen Aspekten sind in der Archäologie experimentelle Ansätze, die Nachbildungen von Werkzeugen und Gegenständen verwenden, weit verbreitet. Darüber hinaus kann auch

die mathematische Modellierung und Simulation dazu beitragen über technische Geräte und deren Verwendung Informationen zu gewinnen, welche experimentell nur sehr schwer erfassbar wären.

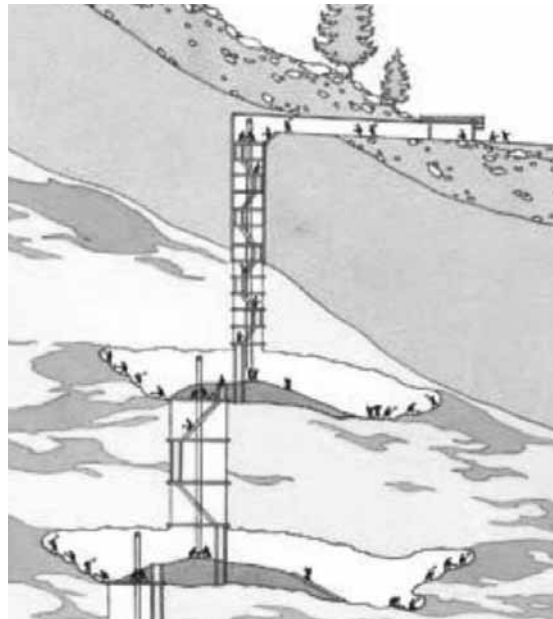


Abbildung 1: Schematische Rekonstruktion des prähistorischen Minen- und Schachtsystems (© D. Gröbner, H. Reschreiter, NHM Wien).

Im Rahmen einer Zusammenarbeit zwischen der TU Wien und dem Naturhistorischen Museum Wien wurden verschiedene Fragestellungen anhand von Fallstudien aus dem prähistorischen Salzbergbau in Hallstatt untersucht und anhand von Simulationsmodellen verschiedene Varianten und Szenarien verglichen [4]. Dabei wurde auf bewährte Methoden aus den Ingenieurwissenschaften (Differentialgleichungen, objektorientierte Starrkörpermodelle) zurückgegriffen. Vorhandene experimentelle Daten lieferten gemeinsam mit Ausgrabungsfunden Parametrisierungen für diese Modelle.

2 Fallstudien

2.1 Seilzug mit Reibung

Bei archäologischen Ausgrabungen wurden Stücke von geflochtenen Seilen aus Lindenbast gefunden, die auf manuelle Seilzüge hindeuten. Es ist allerdings noch ungeklärt, wie lang derartige Seilzüge waren und wie das Seil dabei geführt wurde, da keine Umlenkrollen oder dergleichen gefunden wurden. Um quantitative Anhaltspunkte für diese Fragestellungen zu bekommen, wurden mehrere Varianten von Seilzügen modelliert (Umlenkung an einem Baumstamm oder mit Rolle, offenes oder geschlossenes Seil) und hinsichtlich Kraftaufwand (und damit maximale Länge bzw. Schachthöhe) und Energiebedarf verglichen. Die Simulationsergebnisse zeigen einen dominierenden Einfluss der Seilmasse und des Reibungswiderstandes, welche eine Seilumlenkung mit Gleitreibung über einen Baumstamm für größere Schachthöhen unplausibel machen.

2.2 Schlagbewegung mit Bronzpickel

Zum Brechen des Salzes aus dem Gestein wurden Schlagwerkzeuge (Pickel) mit Holzstiel und aufgesetzter Spitze aus Bronze verwendet. Sehr ungewöhnlich ist allerdings die Form dieser Pickel mit einem Winkel zwischen Stiel und Spitze von ca. 55 bis 75°. Man geht davon aus, dass diese besondere Form an die spezifischen Arbeitsbedingungen angepasst war, zumal keine ähnlichen Geräte an anderen Ausgrabungsorten gefunden wurden. Dieser kleine Winkel und die Länge des Stiels (ca. 1m) erlauben keine typische kreisförmige Schlagbewegung und es ist noch nicht geklärt wie diese Werkzeuge genau verwendet wurden. Mittels Starrkörpermodell des Schlagpickels konnten Vergleiche verschiedene Bewegungsszenarien (mehrere Haltepositionen, Schlag gegen die Wand oder auf den Boden) hinsichtlich

Kraftaufwand und Impuls an der Pickelspitze angestellt werden. So zeigte sich u.a., dass eine Schlagbewegung auf den Boden energieintensiver und ineffizienter ist, da beim Auftreffen der Spitze auf dem Gestein ein enger Winkelbereich (ca. 20 bis 30°) eingehalten werden muss um einen größtmöglichen Effekt zu haben.

3 Fazit

Die Ergebnisse und Vergleiche aus derartigen Simulationen liefern den Archäologen und Archäologinnen neue Informationen und quantitative Anhaltspunkte, die ihnen dabei helfen die Plausibilität verschiedener Theorien zu beurteilen, die Möglichkeiten von Szenarien einzuschränken und gegebenenfalls Theorien zu falsifizieren. Zusammen mit neuen Ausgrabungsfunden und den daraus gewinnbaren Erkenntnissen ist damit das Gesamtbild über den prähistorischen Bergbau in Hallstatt einen ständigen Wandel unterzogen.

Referenzen

- [1] M. Grabner, H. Reschreiter, F.E. Barth, A. Klein, D. Geihofer und R. Wimmer. *Die Denndochronologie in Hallstatt*. Archäologie Österreichs, 17(1), S. 49-58, 2006.
- [2] H. Reschreiter und K. Kowarik. *The Bronze Age*. In: Kingdom of Salt: 7000 Years of Hallstatt, Herausgeber: Kern et al, Wien: VPA3, Naturhistorisches Museum Wien, S. 48-64, 2009.
- [3] K. Kowarik, H. Reschreiter und G. Wurzer. *Modelling Prehistoric Mining*. Tagungsband MATHMOD 2012 - 7th Vienna Conference on Mathematical Modelling, 2012.
- [4] B. Heinzl, E. Auer, B. Slowacki, K. Kowarik, H. Reschreiter, N. Popper und F. Breitenacker. *Mathematical Modelling for Experimental Archaeology: Case Studies for Mechanical Tools in Hallstatt Salt Mines*. Tagungsbeitrag EMSS 2012 - 24th European Modeling and Simulation Symposium, 2012.

State Estimation with Unscented Kalman Filter for Higher Index Nonlinear Differential-Algebraic Systems

Ilja Alkov, Dirk Weidemann
 Institute of System Dynamics and Mechatronics
 University of Applied Sciences Bielefeld
{ilja.alkov, dirk.weidemann}@fh-bielefeld.de

This contribution concerns the state estimation with an unscented Kalman filter (UKF) for higher index nonlinear differential-algebraic (DAE) systems. First, a short introduction to DAE systems is given. A solution concept for nonlinear DAE systems is discussed focusing on properties which are essential for the application to the state estimation. The introduction of stochastic noise in DAE systems and the contrast to stochastic ordinary differential equations (ODE) are described subsequently. Further, the unscented Kalman filter algorithm is reviewed and former stochastic filtering approaches utilizing DAE systems are summarized. Particularly, the formulation of the UKF for the index one DAE systems (cf. [1]) is given. Finally, a direct augmented state estimation approach for higher index nonlinear DAE systems is proposed. Especially, a new concept for the consistent DAE reinitialization after the calculation of the corrected augmented state within the filtering algorithm is developed. Extensions to the UKF algorithm are introduced and discussed, and the capabilities of the state estimation method are explained.

1 Introduction

State estimation is an approach to determine the states of technical systems based on both measurements and structural information of the system. This information is commonly aggregated in an appropriate mathematical model. Technical systems progressively exhibit more complexity, dependencies on environmental systems, and the ability for autonomous operation. Furthermore, phenomena of multiple physical domains are commonly involved in the internal system processes. Thus, in the context of the state estimation, the development of mathematical models is a central and occasionally expensive subtask.

Hierarchical object oriented modeling is one of the most natural and efficient modeling approaches. The basic idea of object orientated modeling is to structure the considered system according to the involved elementary effects and to model these effects using conservation and equilibrium laws. Furthermore, design constraints have to be formulated such that a composition of the submodels yields the model of the complete system. The

result is a differential-algebraic equation (DAE) system describing the considered system behavior within the range of the model validity. Most state estimation algorithms, however, utilize ordinary differential equations (ODE) as underlying mathematical model of the considered technical system. Hence, the application to DAE systems requires adaptations or augmentations to avoid model reformulation, which may be either an expensive or impracticable task.

DAE systems are generalization of ordinary differential equations and are frequently referred to as descriptor systems or singular systems. In contrast to ODE, DAE systems involve more dependent variables than the dimension of the differential state. Furthermore, the state components of a DAE system may depend on each other algebraically. Thus, the state of a DAE system is a combination of a differential state and an algebraic state. Consequently, the state of a DAE is commonly referred to as augmented or descriptor state. The property describing the difference between a DAE and a related ODE system is the index.

This contribution concerns the state estimation with an

unscented Kalman filter (UKF) for higher index nonlinear differential-algebraic systems. In the Section 2, a short introduction to DAE systems is given. A solution concept for nonlinear DAE systems is discussed, focusing on properties which are essential for the application to the state estimation. The introduction of stochastic noise in DAE systems and the contrast to stochastic ODE are described subsequently. Further, the unscented Kalman filter algorithm is reviewed and former stochastic filtering approaches utilizing DAE systems are summarized in the Section 3. Particularly, the formulation of the UKF for the index one DAE systems (cf. [1]) is given. Finally, a direct augmented state estimation approach for higher index nonlinear DAE systems is proposed in Section 4. Especially, a new concept for the consistent DAE reinitialization after the calculation of the corrected augmented state within the filtering algorithm is developed. Extensions to the UKF algorithm are introduced and discussed, and the capabilities of the state estimation method are explained. In the Appendix, notations used in this paper are summarized.

2 Differential-Algebraic Systems

DAE systems are generalization of ordinary differential equations. Consequently, the properties of this system type are more complex. This section concerns the difference between DAE and ODE systems, giving a formal definition of a DAE. Further, the concept of the differentiation index is explained and a solution approach is outlined.

2.1 DAE Systems, Differentiation Index

The following fully implicit system

$$f(x, \dot{x}, u, t) = 0 \quad (1)$$

is considered. Where $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \rightarrow \mathbb{R}^{n_x}$ is a sufficiently differentiable vector valued function implicitly dependent on the augmented state $x \in \mathbb{R}^{n_x}$ of the DAE, the time derivative of the augmented state $\dot{x} \in \mathbb{R}^{n_x}$, the independent input variable $u \in \mathbb{R}^{n_u}$, and the time $t \in \mathbb{R}$. The dimensions of the augmented state and the input variable are denoted by $n_x \in \mathbb{N}$ and $n_u \in \mathbb{N}$, respectively.

The system (1) is an implicit DAE if the Jacobi matrix

with respect to the differentiated augmented state is singular,

$$\det \left(\frac{\partial f}{\partial \dot{x}} \right) = 0 \quad (2)$$

and therefore, according to the implicit function theorem, it is not locally solvable for \dot{x} by algebraic operations. In order to solve for \dot{x} , the function f is differentiated repeatedly forming the derivative array (3).

$$\begin{bmatrix} f(x, \dot{x}, u, t) \\ \frac{d}{dt} f(x, \dot{x}, u, t) \\ \vdots \\ \frac{d^\mu}{dt^\mu} f(x, \dot{x}, u, t) \end{bmatrix} = 0 \quad (3)$$

The differentiation index of a DAE is defined as the smallest number $\mu \in \mathbb{N}$ for which the derivative array is solvable for $\dot{x} = \phi(x, u, \dot{u}, \dots, u^{(\mu)}, t)$, the first time derivative of the augmented state as a function of the augmented state, the input variable, time derivatives of the input variable, and the time. The resulting ODE system is referred to as the underlying ODE. According to this definition, an implicit ODE system is a DAE of differentiation index zero.

2.2 Solution of DAE Systems

The numerical solution of DAE systems commonly consists of two steps. First, the index of the considered DAE is reduced, then, the index reduced system is integrated numerically using algorithms closely related to the integration algorithms for ODE. The index reduction procedure is closely related to the identification of the underlying ODE and algebraic constraints (i.e. restrictive manifold).

The reformulation of (1) starts with the consideration of $\frac{\partial f}{\partial \dot{x}}$. Since this Jacobi matrix is assumed to be singular, there exist two permutation matrices $P_{x,0}$ and $P_{f,0}$ such that

$$P_{x,0} \cdot x = [x_0^T, z_0^T]^T \quad (4)$$

$$P_{f,0} \cdot f = [f_0^T, g_0^T]^T \quad (5)$$

$$P_{f,0} \cdot \frac{\partial f}{\partial \dot{x}} \cdot P_{x,0}^T = \begin{bmatrix} \frac{\partial f_0}{\partial \dot{x}_0} & \frac{\partial f_0}{\partial \dot{z}_0} \\ \frac{\partial g_0}{\partial \dot{x}_0} & \frac{\partial g_0}{\partial \dot{z}_0} \end{bmatrix} \wedge \det \left(\frac{\partial f_0}{\partial \dot{x}_0} \right) \neq 0, \quad (6)$$

and thus the DAE system can be decomposed to

$$f_0(x_0, \dot{x}_0, z_0, \dot{z}_0, u, t) = 0 \quad (7)$$

$$g_0(x_0, \dot{x}_0, z_0, \dot{z}_0, u, t) = 0. \quad (8)$$

Since $\frac{\partial f_0}{\partial \dot{x}_0}$ is regular, there exists a local solution

$$\dot{x}_0 = \phi_0(x_0, z_0, \dot{z}_0, u, t). \quad (9)$$

Further, \dot{x}_0 is substituted in g_0 . The obtained function may not depend on \dot{z}_0 or in this case the previous decomposition would result in additional components in x_0 . Therefore,

$$g_0(x_0, z_0, u, t) = 0. \quad (10)$$

Thereafter, g_0 is differentiated with respect to time obtaining

$$\frac{\partial g_0}{\partial x_0} \cdot \dot{x}_0 + \frac{\partial g_0}{\partial z_0} \cdot \dot{z}_0 + \frac{\partial g_0}{\partial u} \cdot \dot{u} + \frac{\partial g_0}{\partial t} = 0. \quad (11)$$

Equation (11) is an implicit DAE if $\frac{\partial \frac{dg_0}{dt}}{\partial \dot{z}_0}$ is singular. Again, proceeding with decomposition, two permutation matrices $P_{x,1}$ and $P_{f,1}$ exist such that

$$P_{x,1} \cdot z_0 = [x_1^T, z_1^T]^T \quad (12)$$

$$P_{f,1} \cdot \frac{dg_0}{dt} = [f_1^T, g_1^T]^T \quad (13)$$

$$P_{f,1} \cdot \frac{\partial \frac{dg_0}{dt}}{\partial \dot{z}_0} \cdot P_{x,1}^T = \begin{bmatrix} \frac{\partial f_1}{\partial \dot{x}_1} & \frac{\partial f_1}{\partial \dot{z}_1} \\ \frac{\partial g_1}{\partial \dot{x}_1} & \frac{\partial g_1}{\partial \dot{z}_1} \end{bmatrix} \wedge \det \left(\frac{\partial f_1}{\partial \dot{x}_1} \right) \neq 0, \quad (14)$$

and

$$f_1(x_0, x_1, \dot{x}_0, \dot{x}_1, z_1, \dot{z}_1, u, \dot{u}, t) = 0 \quad (15)$$

$$g_1(x_0, x_1, \dot{x}_0, \dot{x}_1, z_1, \dot{z}_1, u, \dot{u}, t) = 0. \quad (16)$$

Since, $\frac{\partial f_1}{\partial \dot{x}_1}$ is regular, there exists a local solution

$$\dot{x}_1 = \phi_1(x_0, x_1, \dot{x}_0, \dot{z}_1, \dot{z}_1, u, \dot{u}, t). \quad (17)$$

Further, \dot{x}_0 and \dot{x}_1 are substituted in g_1 . The obtained function may not depend on \dot{z}_1 or in this case the previous decomposition would result in additional components in x_1 . Therefore,

$$g_1(x_0, x_1, z_1, u, \dot{u}, t) = 0. \quad (18)$$

Repeated μ times differentiation of g_m , whereas μ is the differentiation index, results in

$$f_m(x_0, \dots, x_m, \dot{x}_0, \dots, \dot{x}_m, z_m, \dot{z}_m, u, \dots, u^{(\mu)}, t) \wedge \det \left(\frac{\partial f_m}{\partial \dot{x}_m} \right) \neq 0 \quad (19)$$

$$\dot{x}_m = \phi_m(x_0, \dots, x_m, \dot{x}_0, \dots, \dot{x}_{m-1}, z_m, \dot{z}_m, u, \dots, u^{(\mu)}, t) \quad (20)$$

$$g_m(x_0, \dots, x_m, z_m, u, \dots, u^{(\mu)}, t) = 0 \quad (21)$$

$$\forall m \in \{0, 1, \dots, \mu\}$$

$$\dim(g_\mu) = 0 \wedge \dim(z_\mu) = 0. \quad (22)$$

Furthermore, repeated substitution

$$z_{m-1} = P_{x,m}^T \cdot [x_m^T, z_m^T]^T \forall m \in \{1, 2, \dots, \mu\} \quad (23)$$

$$\dot{x}_m = \phi_m \forall m \in \{0, 1, \dots, \mu\} \quad (24)$$

in (19) to (21) and reconstruction

$$x = \prod_{m=0}^{\mu} \begin{bmatrix} I_{n_x - \dim(x_m) - \dim(z_m)} & 0 \\ 0 & P_{x,m} \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ \dots \\ x_\mu \end{bmatrix} \quad (25)$$

yields in

$$\dot{x} = \phi(x, u, \dots, u^{(\mu)}, t) \quad (26)$$

$$g(x, u, \dots, u^{(\mu)}, t) = 0, \quad (27)$$

whereas

$$\phi = \prod_{m=0}^{\mu} \begin{bmatrix} I_{n_x - \dim(x_m) - \dim(z_m)} & 0 \\ 0 & P_{x,m} \end{bmatrix} \cdot \begin{bmatrix} \phi_0 \\ \dots \\ \phi_\mu \end{bmatrix} \quad (28)$$

$$g = [g_0^T, \dots, g_\mu^T]^T. \quad (29)$$

The presented reformulation determines the underlying ODE (26) and the restrictive manifold (27) of the considered DAE (1). The restrictive manifold imposes an algebraic constraint on the components of the augmented state. Therefore, the system is obviously overdetermined. However, only a subset of the augmented state may be initialized arbitrarily. Further, the augmented state x can be decomposed to the differential state $x_d \in \mathbb{R}^{n_{x,d}}$ and the algebraic state $x_a \in \mathbb{R}^{n_{x,a}}$, with corresponding dimensions $n_{x,d} \in \mathbb{N}$ and $n_{x,a} \in \mathbb{N}$, thus a common form used for direct numerical integration is a semi-explicit index one DAE given as

$$\dot{x}_d = \phi_d(x, u, \dots, u^{(\mu)}, t) \quad (30)$$

$$g_a(x_a, x_d, u, \dots, u^{(\mu)}, t) = 0 \wedge \det \left(\frac{\partial g_a}{\partial x_a} \right) \neq 0. \quad (31)$$

This decomposition is not unique and is addressed by the index reduction procedure (e.g. [2], [3], [4]). A more formal analysis of nonlinear DAE systems is given in [5].

2.3 Introduction of Stochastic Noise

The previously discussed reformulation discovers another important property of DAE systems. The underlying ODE as well as the algebraic constraints depend on input variable and corresponding time derivatives. Therefore, sufficient differentiability of the input variable is required. Furthermore, this has to be considered particularly in the introduction of the stochastic noise in DAE. The usual assumption of a vector valued independent Gaussian white noise process $v \sim \mathcal{N}(0, C_{v,v})$ affecting (1) leads to

$$f(x, \dot{x}, v, u, t) = 0. \quad (32)$$

Obviously, stochastic noise is subjected to the same requirements as the input variable. Further, additional assumptions concerning stochastic properties of the noise signal are to be introduced (cf. [6], [7]).

Considering the discrete-time unscented Kalman filter, the stochastic noise can be introduced without affecting the solution procedure of the DAE. A regular DAE (1) has the solution x for a consistent initial augmented state $x_i = x(t = t_i)$ at the time instant t_i , i.e. the initial state satisfying the algebraic constraints (27). Therefore, a state transition function exists

$$x_j = \xi_j(x_i, u(\cdot), \dots, u^{(\mu)}(\cdot), t_i), \quad (33)$$

where $x_j = x(t = t_j)$ is the augmented state at the time instant t_j . The state transition can be calculated utilizing an appropriate numerical algorithm for higher index DAE systems. Consequently, the noise signal may be introduced as

$$x_j = \xi_j(x_i, u(\cdot), \dots, u^{(\mu)}(\cdot), t_i) + \sigma_j(x_i, u(\cdot), \dots, u^{(\mu)}(\cdot), t_i) \cdot v_i, \quad (34)$$

where σ_j is an appropriate matrix function.

3 Unscented Kalman Filter

The unscented Kalman filter algorithm is an extension of the Kalman filter for the application to nonlinear systems. The essential substructure of the algorithm is the unscented transformation (UT). The UT was originally introduced in [8] and extended to the scaled UT in [9]. Especially the UT is an approach for the approximation

of the expectation and covariance of stochastic variables involved in a nonlinear transformation. Therefore, a deterministically obtained set of function evaluations involved in the particular nonlinear transformation is considered. Thus, the UKF has several advantages over comparable estimation algorithms. First, the computational complexity order $\mathcal{O}(L^3)$ of the UKF, where L is the dimension of the estimated variable, is the same as for the extended Kalman filter (EKF) (c.f. [10]) and thus lower than for a particle filter with comparable estimation accuracy. Further, in contrast to the EKF, the UKF does not involve computation of Jacobians of the estimated system. Besides this, the UKF enables estimation with higher accuracy than the EKF for stiff and highly nonlinear systems in consequence of the application of the UT instead of equation linearization as in EKF.

Indeed, filtering methods often refer to ODE systems, whereas the application to DAE systems requires significant augmentations to the algorithms. However, several approaches are introduced in former publications. The state estimation for DAE systems utilizing an extended Kalman filter probably the most frequently used state estimation algorithm for nonlinear ODE systems, is addressed in [11]. Particle filtering applied to nonlinear higher index DAE systems is analyzed in [6], where the index reduced form is implicitly considered. Mandela et al. propose in [12] a method for the application of the UKF to nonlinear index one DAE systems. A revised formulation of this algorithm is presented in [1].

The semi-explicit index one DAE is of particular importance. On the one hand, higher index DAE are commonly reduced to this form for numerical solution. On the other hand, this DAE system type frequently results from the modeling approach directly. Finally, every sufficiently differentiable structurally regular higher index DAE can be reformulated to this form as shown in [5].

The semi-explicit index one DAE (30) and (31) is considered for state estimation. For the process, an output equation is given describing measurements taken at the actual technical system. Further, a stochastic noise signal is introduced in the state transition function as well as in the output equation. The state equation is discretized applying appropriate numerical solver. This results in the estimation model given as

$$x_{d,j} = \xi_{d,j}(x_{d,i}, x_{a,i}, u_{\mu}(\cdot), t_i) + v_{d,i} \quad (35)$$

$$x_{a,j} = \xi_{a,j}(x_{d,j}, u_{\mu,j}, t_j) \quad (36)$$

$$y_j = h_j(x_{d,j}, x_{a,j}, u_{\mu,j}, t_j) + v_{y,j} \quad (37)$$

$$u_\mu(t) = [u(t), \dot{u}, \dots, u^{(\mu)}(t)], \quad (38)$$

with time instants t_i and t_j , $\{i, j\} \in \mathbb{N}$, $j = i + 1$, differential states $x_{d,\{i,j\}} \in \mathbb{R}^{n_{x,d}}$, algebraic states $x_{a,\{i,j\}} \in \mathbb{R}^{n_{x,a}}$, output variables $y_{\{i,j\}} \in \mathbb{R}^{n_y}$, differential state transition functions $\xi_{d,\{i,j\}}$, algebraic state functions $\xi_{a,\{i,j\}}$, output functions $h_{\{i,j\}}$, stochastic state noise $v_{d,\{i,j\}} \in \mathbb{R}^{n_{x,d}}$, and stochastic output noise $v_{y,\{i,j\}} \in \mathbb{R}^{n_y}$ at this time instants respectively. Further, the time dependent input variable assigned by $u_\mu \in \mathbb{R}^{n_u \times \mu+1}$ is considered as a set with corresponding time derivatives as defined in (38), as well as the input variables at respective time instants $u_{\mu,\{i,j\}}$. Furthermore, $\mu \in \mathbb{N}$ is the index of the original higher index DAE reduced to the form (30) and (31) as discussed before. Stochastic processes $v_{d,\{i,j\}} \sim \mathcal{N}(0, C_{v_d,v_d,\{i,j\}}) \in \mathbb{R}^{n_{x,d}}$ and $v_{y,\{i,j\}} \sim \mathcal{N}(0, C_{v_y,v_y,\{i,j\}}) \in \mathbb{R}^{n_y}$ are assumed to be Gaussian white noise processes independent of each other and other system quantities with zero expectation and known covariances.

The state estimation with the UKF applied to a nonlinear index one DAE system is realized by the calculation of the system state estimate with minimum mean-square error for the model equation system given in (35) to (37). This estimation requires an observation of the actual system at discrete time instants. The UKF algorithm is based on the scaled UT, thus the weights

$$w_m = \frac{[2 \cdot \alpha^2 \cdot (n_{x,d} + \gamma) - 2 \cdot n_{x,d}, 1_{1 \times 2 \cdot n_{x,d}}]^T}{2 \cdot \alpha^2 \cdot (n_{x,d} + \gamma)} \quad (39)$$

and

$$w_c = w_m + [(1 - \alpha^2 + \beta), 0_{1 \times 2 \cdot n_{x,d}}]^T \quad (40)$$

are defined for the mean and the covariance calculation of the considered samples, respectively, whereas $0 < \alpha \leq 1$, $0 \leq \beta$, $0 \leq \gamma$ are real scalar filter set-up parameters. More convenient is the definition of the matrix valued weight W_c for the covariance calculation, thus

$$W_c = \Psi \cdot \text{diag}(w_c) \cdot \Psi^T \quad (41)$$

$$\Psi = [I_{2 \cdot n_{x,d}+1} - w_m \cdot 1_{1 \times 2 \cdot n_{x,d}+1}] \quad (42)$$

is used in the subsequently discussed algorithm.

The UKF estimation algorithm is structured into the steps *initialization*, *prediction*, and *correction* as given below, whereas introduced quantities are explained in the Appendix.

Unscented Kalman Filter Algorithm for Index one Differential-Algebraic Systems

Initialization:

$$i = i_0 \quad (43)$$

$$\mathbb{E}(x_{d,i|i}) = x_{d,0} \quad (44)$$

$$C_{x_d,x_d,i|i} = C_{x_d,x_d,0} \quad (45)$$

Prediction:

$$d_{d,i|i} = [0_{n_{x,d} \times 1}, -\text{msr}(C_{x_d,x_d,i|i}), \text{msr}(C_{x_d,x_d,i|i})] \quad (46)$$

$$s_{d,i|i} = \mathbb{E}(x_{d,i|i}) \cdot 1_{1 \times 2 \cdot n_{x,d}+1} + \alpha \cdot \sqrt{n_{x,d} + \gamma} \cdot d_{d,i|i} \quad (47)$$

$$s_{a,i|i} = \xi_{a,i}(s_{d,i|i}, u_{\mu,i}, t_i) \quad (48)$$

$$s_{d,j|i}^* = \xi_{d,j}(s_{d,i|i}, s_{a,i|i}, u_{\mu}(\cdot), t_i) + \mathbb{E}(v_{d,i}) \cdot 1_{1 \times 2 \cdot n_{x,d}+1} \quad (49)$$

$$\mathbb{E}(x_{d,j|i}) = s_{d,j|i}^* \cdot w_m \quad (50)$$

$$C_{x_d,x_d,j|i} = s_{d,j|i}^* \cdot W_c \cdot s_{d,j|i}^{*T} + C_{v_d,v_d,i} \quad (51)$$

$$d_{d,j|i} = [0_{n_{x,d} \times 1}, -\text{msr}(C_{x_d,x_d,j|i}), \text{msr}(C_{x_d,x_d,j|i})] \quad (52)$$

$$s_{d,j|i} = \mathbb{E}(x_{d,j|i}) \cdot 1_{1 \times 2 \cdot n_{x,d}+1} + \alpha \cdot \sqrt{n_{x,d} + \gamma} \cdot d_{d,j|i} \quad (53)$$

$$s_{a,j|i} = \xi_{a,j}(s_{d,j|i}, u_{\mu,j}, t_j) \quad (54)$$

$$s_{y,j|i} = h_j(s_{d,j|i}, s_{a,j|i}, u_{\mu,j}, t_j) + \mathbb{E}(v_{y,j}) \cdot 1_{1 \times 2 \cdot n_{x,d}+1} \quad (55)$$

$$\mathbb{E}(y_{j|i}) = s_{y,j|i} \cdot w_m \quad (56)$$

$$C_{y,y,j|i} = s_{y,j|i} \cdot W_c \cdot s_{y,j|i}^T + C_{v_y,v_y,j} \quad (57)$$

$$C_{x_d,y,j|i} = s_{d,j|i} \cdot W_c \cdot s_{y,j|i}^T \quad (58)$$

Correction:

$$K_{d,j} = C_{x_d,y,j|i} \cdot C_{y,y,j|i}^{-1} \quad (59)$$

$$\mathbb{E}(x_{d,j|j}) = x_{d,j|i} + K_{d,j} \cdot (y_{s,j} - y_{j|i}) \quad (60)$$

$$C_{x_d,x_d,j|j} = C_{x_d,x_d,j|i} - K_{d,j} \cdot C_{y,y,j|i} \cdot K_{d,j}^T \quad (61)$$

4 UKF for Higher Index DAE

The algorithm discussed previously is conceptualized for index one DAE systems or systems reduced to this form. Thus, the algebraic state is considered to be sufficiently

determined by the differential state, input variable, and time. Therefore, a deterministic function is used to calculate the algebraic state. Furthermore, model uncertainties are considered to be existent in the differential state transition function and the output function. The corresponding approach is discussed in e.g. [1], [7], and [12].

The algebraic state function naturally includes model uncertainties. The reasons for these are obviously the same as for the uncertainties in the differential state transition equation and the output equation. Frequently, the algebraic state equation represents idealistic formulations of conservation laws, equilibrium principles, and design constraints. Further, measurements taken at the actual technical system may depend on both the differential and the algebraic state. Therefore, since the output equation as well as the algebraic state equation are purely algebraic expressions, the stochastic noise signal should be introduced in both in the exact same manner. Furthermore, since the index reduction generally results in a nonunique decomposition to differential and algebraic states, different processing of these states within the estimation algorithm may be difficult to justify.

The UKF estimation of the augmented state of higher index DAE systems can be succeeded utilizing a slightly adapted algorithm as described before for index one DAE. Thus, the corresponding augmented state transition function can be used instead of the differential state transition function and the algebraic state function as proposed in (34). Further, augmented state noise signal can be introduced representing model uncertainties in both differential and algebraic states. The calculation of the augmented state transition with an appropriate DAE solution algorithm requires in general the use of a consistent initial augmented state (cf. Section 2). Within the initialization procedure of the DAE for the prediction in the UKF algorithm, the augmented state can not be considered completely as discussed in Section 2.2. Therefore, the proposed approach is not the common estimation procedure yet.

However, several appropriate initialization approaches can be provided to include the consideration of the algebraic state noise and thus to enhance the capabilities of sample based discrete-time stochastic filtering algorithms for higher index DAE systems.

Differential State Initialization. DAE solution algorithms commonly reduce the index of the DAE and provide an index one DAE for numerical integration. Con-

sequently, the decomposition to the arbitrarily initializable differential state and the corresponding dependent algebraic state is proceeded. Therefore, the augmented state sample generated by the filtering algorithm can be decomposed to the differential state sample and the algebraic state sample, accordingly. Further, a consistent algebraic state sample can be calculated using the algebraic state function available from the DAE solution algorithm. Thus, the augmented state transition function can be executed with the obtained consistent augmented state sample within the prediction phase.

The differential state initialization obviously presumes an implicit discrete-time state transition of the algebraic state. Therefore, this approach is appropriate for discrete-time filtering only. The proposed adaptation results in a significant contribution of algebraic state model uncertainties to recursive estimation. The best advantage of this initialization approach is the low computational complexity. However, a certain decomposition to the differential and the algebraic states is considered. Since the decomposition is generally not unique, different processing of these states as proposed may be disadvantageous.

Augmented State Initialization. The augmented state sample generated by the filtering algorithm is a set of DAE inconsistent elements. However, the consistency can be achieved by an optimization task using the augmented states in the sample as the optimization initial guesses. The corresponding optimization problem and algorithm are described in [13].

Substantially, the augmented state initialization does not distinguish between the differential and algebraic states. Equivalent processing of differential and algebraic states and thus consideration of the complete augmented state is more natural and corresponds the original higher index DAE. The obvious disadvantage of this approach is the high computational complexity. Further, the original distribution of the augmented state sample may be modified by the optimization task significantly. Therefore, the applicability in the particular filtering algorithm should be verified. Analogous to the approach proposed above, this approach presumes an implicit discrete-time state transition of the augmented state due to the enforcement of consistency. Again, the proposed extension results in an essential contribution of algebraic state model uncertainties to recursive estimation.

Optimal Sub-State Initialization. The concept of the optimal sub-state initialization is based on the decompo-

sition of the higher index DAE to the underlying ODE (26) and the restrictive manifold (27) (cf. Section 2.2). The consistent initialization of a higher index DAE corresponds to the calculation of the states and differentiated states from the underlying ODE, the algebraic constraints, and feasible initialization equations at the initial time instant. In case, the considered DAE has degrees of freedom, which means a sub-state may be initialized arbitrarily, the system (26)-(27) is underdetermined according to the calculation of the augmented state and the corresponding time derivatives. Therefore, a nonempty set of algebraic initialization equations is required to achieve determined conditions. The specification of the complete augmented state estimate corresponds to the provision of initialization equations. This set of initialization equations is, however, not feasible since for a generally nonempty set of algebraic constraints defining the restrictive manifold, the resulting system is overdetermined. Thus, the initialization problem may be described using (26)-(27) as

$$\begin{bmatrix} \dot{x}_i - \phi(x_i, u_{\mu,i}, t_i) \\ g(x_i, u_{\mu,i}, t_i) \\ P_i \cdot [x_i - x_{est,i}] \end{bmatrix} = \Phi_i(x_i, \dot{x}_i, u_{\mu,i}, x_{est,i}, P_i, t_i) \quad (62)$$

$$\Phi_i(x_i, \dot{x}_i, u_{\mu,i}, x_{est,i}, P_i, t_i) = 0 \quad (63)$$

with the augmented state x_i , the estimate of the augmented state $x_{est,i}$, the input variable, and corresponding derivatives $u_{\mu,i} = [u_i, \dot{u}_i, \dots, u_i^{(\mu)}]$, at the initial time instant t_i , where μ is the differentiation index and n_x is the dimension of the augmented state. Further, matrix $P_i \in \mathbb{R}^{n_x - \dim(g) \times n_x}$ is a partition of an appropriate permutation matrix $[P_i^T, Q_i^T]^T \in \mathbb{R}^{n_x \times n_x}$. The initialization problem is solved by the determination of P_i , such that

$$\det\left(\frac{\partial \Phi_i}{\partial [x_i^T, \dot{x}_i^T]^T}\right) \neq 0 \quad (64)$$

and therefore Φ_i can be solved for $[x_i^T, \dot{x}_i^T]^T$. Since P_i is generally nonunique for the described restrictions, a discrete optimization problem can be formulated, such that

$$\min_{[P_i^T, Q_i^T]^T} \|\Phi_i \cdot (X_{est,i} - X_i)\| \quad (65)$$

s.t.

$$X_i = \arg(\Phi_i(X_i, \dot{X}_i, u_{\mu,i}, X_{est,i}, P_i, t_i) = 0), \quad (66)$$

where $X_{est,i}$ is the sample of augmented states generated by the filtering algorithm, X_i and \dot{X}_i are sets of consis-

tently calculated states and differentiated states, respectively. Thus, a sub-state obtained by the proposed optimization task can be initialized with exact values from the estimation procedure. Further, the difference between the consistently initialized augmented state and the estimation is optimal in the sense of the formulated optimization problem.

The optimal sub-state initialization implies equivalent processing of the differential and the algebraic states. Consequently, the estimation algorithm utilizes a model formulation close to the original higher index DAE system. The computational complexity depends on the employed discrete optimization algorithm and may be of lower order than the global search in the augmented state initialization approach described previously. Again, analogous to the proposals discussed first, this approach presumes an implicit discrete-time state transition of the augmented state due to the enforcement of consistency and thus it is appropriate for discrete-time filtering only. Furthermore, the proposed augmentation results in a essential contribution of algebraic state model uncertainties to recursive estimation.

5 Conclusion

This contribution concerns the state estimation with an unscented Kalman filter for higher index nonlinear differential-algebraic systems. A short introduction to nonlinear differential-algebraic systems is given explaining the properties of this system type. Subsequently, the unscented Kalman filter algorithm for index one differential-algebraic systems is presented. Further, this algorithm is extended for the application to higher index nonlinear differential-algebraic systems. Substantially, a new concept for the consistent DAE reinitialization after the calculation of the corrected augmented state within the filtering algorithm is developed. Extensions to the UKF algorithm are introduced and discussed, and the capabilities of the state estimation method are explained. The introduced extensions enable the consideration of augmented state noise and therefore model uncertainties in both differential and algebraic equations of the DAE system. The verification of the proposed estimation methods by the application to appropriate benchmark systems is a subject of the further work.

Appendix

$(\cdot)^T$	transposed matrix
$\dim(\cdot)$	dimension
$\text{diag}(\cdot)$	diagonal matrix
$\text{msr}(\cdot)$	matrix square root, thus $\text{msr}(A) \cdot \text{msr}(A)^T = A$
$E(\cdot)$	expectation
$0_{n_1 \times n_2}$	matrix of size $n_1 \times n_2$ filled with zeros
$1_{n_1 \times n_2}$	matrix of size $n_1 \times n_2$ filled with ones
I_{n_1}	identity matrix of size n_1
i_0	initial time instant index
$i i$	index assigning quantity at the time instant i calculated using actual system observa- tions up to time instant i
$j i$	index assigning quantity at the time instant j calculated using actual system observa- tions up to time instant i
$j j$	index assigning quantity at the time instant j calculated using actual system observa- tions up to time instant j
τ	time instant index replacement $\tau \in \{i, j, i i, j i, j j\}$ used in this table for compact description of quantities
$x_{d,0}$	initial estimation of differential state
$x_{d,\tau}$	differential state
y_τ	output
$y_{s,\tau}$	actual system output
$d_{d,\tau}$	differential state differences matrix
$s_{d,\tau}$	differential state set matrix
$s_{d,\tau}^*$	auxiliary differential state set matrix
$s_{d,\tau}$	algebraic state set matrix
$s_{y,\tau}$	output set matrix
$C_{x_d, x_d, 0}$	initial estimation of differential state co- variance matrix
$C_{x_d, x_d, \tau}$	differential state covariance matrix
$C_{y, y, \tau}$	output covariance matrix
$C_{x_d, y, \tau}$	differential state and output cross covari- ance matrix
$K_{d,\tau}$	Kalman gain matrix corresponding to dif- ferential state

Remark: The application of a function defined for a vector argument to a matrix argument with equal number of rows means the application of this function to each column of the matrix. The result is a matrix with same number of columns as the input matrix. If several matrix arguments are used instead of vector arguments, the column numbers of the matrices have to be equal.

Acknowledgment

This work was funded by the European Union and the federal state of North Rhine-Westphalia, Germany.

References

- [1] I. Alkov and D. Weidemann, "Improved Unscented Kalman filter formulation for nonlinear differential-algebraic systems," in *Proc. of the REM 2013 – 14th International Workshop on Research and Education in Mechatronics*, Vienna, Austria, 2013, pp. 149–154.
- [2] C. Gear, "Differential-algebraic equation index transformations," *SIAM Journal on Scientific and Statistical Computing*, vol. 9, no. 1, pp. 39–47, 1988.
- [3] C. Pantelides, "The consistent initialization of differential-algebraic systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 9, no. 2, pp. 213–231, 1988.
- [4] S. Mattsson and G. Söderlind, "Index reduction in differential-algebraic equations using dummy derivatives," *SIAM Journal on Scientific and Statistical Computing*, vol. 14, no. 3, pp. 677–692, may 1993.
- [5] P. Kunkel and V. Mehrmann, "Analysis of over- and underdetermined nonlinear differential-algebraic systems with application to nonlinear control problems," *Mathematics of Control, Signals and Systems*, vol. 14, no. 3, pp. 233–256, 2001.
- [6] M. Gerdin, "Identification and estimation for models described by differential-algebraic equations," Ph.D. dissertation, University Linköping, Linköping, Sweden, 2006.
- [7] M. Gerdin and J. Sjöberg, "Nonlinear stochastic differential-algebraic equations with application to particle filtering," in *Proc. of the 45th IEEE Conference on Decision and Control*, 2006, pp. 6630–6635.
- [8] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proc. of the American Control Conference*, vol. 3, 1995, pp. 1628–1632.
- [9] S. Julier, "The scaled unscented transformation," in *Proc. of the American Control Conference*, vol. 6, 2002, pp. 4555–4559.
- [10] R. van der Merwe and E. Wan, "The square-root unscented kalman filter for state and parameter-estimation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, 2001, pp. 3461–3464.
- [11] V. M. Becerra, P. D. Roberts, and G. W. Griffiths, "Applying the extended kalman filter to systems described by nonlinear differential-algebraic equations," *Control Engineering Practice*, vol. 9, no. 3, pp. 267–281, 2001.
- [12] R. Mandela, R. Rengaswamy, and S. Narasimhan, "Nonlinear state estimation of differential algebraic systems," in *Proc. of the 7th IFAC International Symposium on Advanced Control of Chemical Processes*, vol. 7, no. 1, Koç, Turkey, 2009, pp. 792–797.
- [13] B. Bachmann, P. Aronsson, and P. Fritzson, "Robust initialization of differential algebraic equations," in *Proc. of the 5th International Modelica Conference*, Vienna, Austria, 2006, pp. 607–614.

Convergence of Dynamic Iteration for Coupled Engineering Problems

Sebastian Schöps^{1,2}, Andreas Bartel³ and Michael Günther³

¹ Graduate School CE, Technische Universität Darmstadt

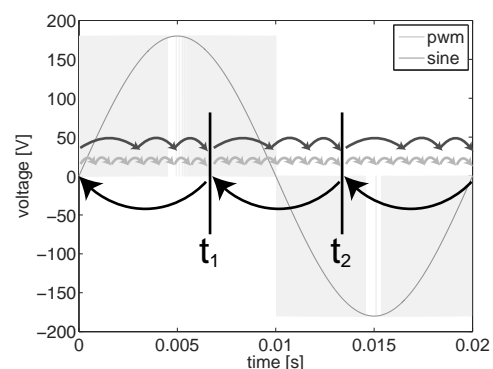
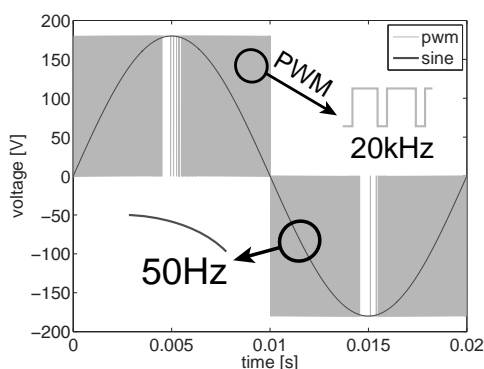
² Theorie Elektromagnetischer Felder, Technische Universität Darmstadt

³ Angewandte Mathematik und Numerische Analysis, Bergische Universität Wuppertal

www.graduate-school-ce.de

February 20, 2014

Multirate Phenomenon in Device Simulation



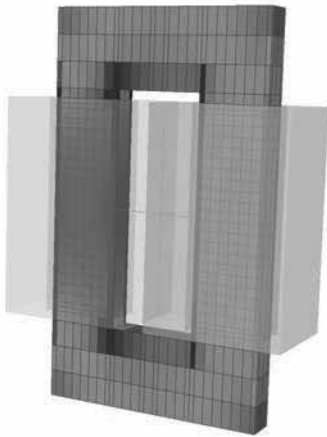
multirate

- fast signals → small steps, slow signals → large steps
- different time steps improve the efficiency

dynamic iteration / waveform relaxation

- decoupling on time windows $H_n = [T_n, T_{n+1}]$
- step sizes h according to signal → different time steps
- exchange waveforms and compute windows iteratively → fixed point

Device: 3D Model of a Transformer



- Maxwell in terms of **magnetic vector potential** (MVP) \vec{A} with $\vec{B} = \nabla \times \vec{A}$
- neglect **displacement currents**
- but core exhibits **eddy currents**
- homogeneous **Dirichlet** BC on Γ
- **excitation** by winding functions
- **discretize** by edge elements

■ curl-curl (differential algebraic) equation

$$\mathbf{M}_\sigma \frac{d}{dt} \mathbf{a} + \mathbf{C}^\top \mathbf{M}_\nu(\mathbf{a}) \mathbf{C} \mathbf{a} = \mathbf{X}_m \mathbf{i}_m$$

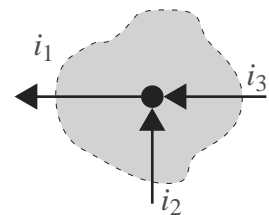
with discrete MVP \mathbf{a} , conductivity \mathbf{M}_σ and curl-curl matrix $\mathbf{K}_\nu := \mathbf{C}^\top \mathbf{M}_\nu(\mathbf{a}) \mathbf{C}$.

Circuit: Model by Modified Nodal Analysis

- lumping: $\mathbf{i}_k := \int_{\Gamma_k} \vec{J} \cdot d\vec{\Gamma}$ and $\mathbf{v}_k := \int_{\ell_k} \vec{E} \cdot d\vec{\ell}$

- KCL: $\oint_{\Gamma} \vec{J} \cdot d\vec{\Gamma} = \sum_{\Gamma_k \subset \Gamma} \mathbf{i}_k = 0 \rightarrow \mathbf{A} \mathbf{i} = 0$

- KVL: $\oint_{\ell} \vec{E} \cdot d\vec{\ell} = \sum_{\ell_k \subset \ell} \mathbf{v}_k = 0$



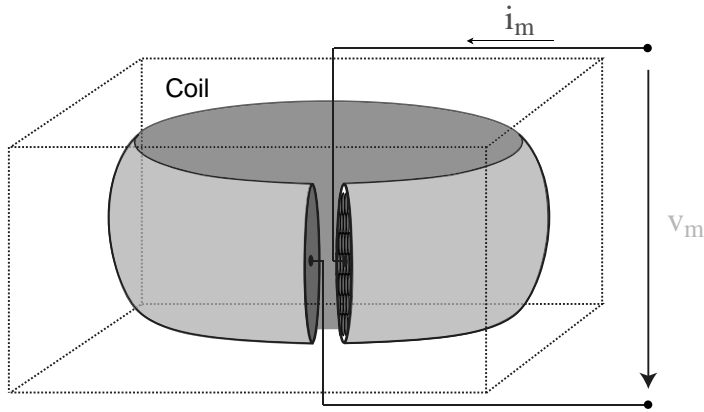
[Chua, Ho/Ruehli/Brennan 1975]

- $\mathbf{A}_C \mathbf{C}_q \mathbf{A}_C^\top \frac{d}{dt} \mathbf{u} + \mathbf{A}_R \mathbf{G}_R \mathbf{A}_R^\top \mathbf{u} + \mathbf{A}_L \mathbf{i}_L + \mathbf{A}_V \mathbf{i}_V + \mathbf{A}_I \mathbf{i}_s(t) = \mathbf{0},$
 $\mathbf{L}_\phi \frac{d}{dt} \mathbf{i}_L - \mathbf{A}_L^\top \mathbf{u} = \mathbf{0},$
 $\mathbf{A}_V^\top \mathbf{u} - \mathbf{v}_s(t) = \mathbf{0},$

topology: incidence matrices \mathbf{A}_\star with $\mathbf{v}_\star = \mathbf{A}_\star^\top \mathbf{u}$, constitutive: capacitances \mathbf{C}_q , inductances \mathbf{L}_ϕ , sources \mathbf{i}_s , \mathbf{v}_s , unknowns: potentials \mathbf{u} , currents \mathbf{i}_L , \mathbf{i}_V

Scale Bridging by Winding Functions

■ Stranded conductor 'coil' model [Bedrosian 1993, Dular 1999, De Gersem et al. 2004]



- many thin wires
- homogeneous current
- divergence freeness
- partition of unity in coil m
 $\sum_i \mathbf{X}_{i,m} = \text{number of turns}$
- orthogonality
 $\mathbf{X}_{:,n}^\top \mathbf{X}_{:,m} = 0 \quad \forall n \neq m$
- curl-curl and coupling equations for each coil

■ circuit coupling via currents/voltages: ($m \in M$)

$$\mathbf{v}_M = \frac{d}{dt} \mathbf{X}^\top \mathbf{a} + \mathbf{R}_M \mathbf{i}_M \quad \text{with} \quad \mathbf{R}_M = \mathbf{X}^\top \mathbf{M}_\sigma^+ \mathbf{X}$$



Interpretation of Coupling Operator

■ Lumped resistance extraction

$$\mathbf{R}_M = \mathbf{X}^\top \mathbf{M}_\sigma^+ \mathbf{X}$$

■ Lumped inductance extraction

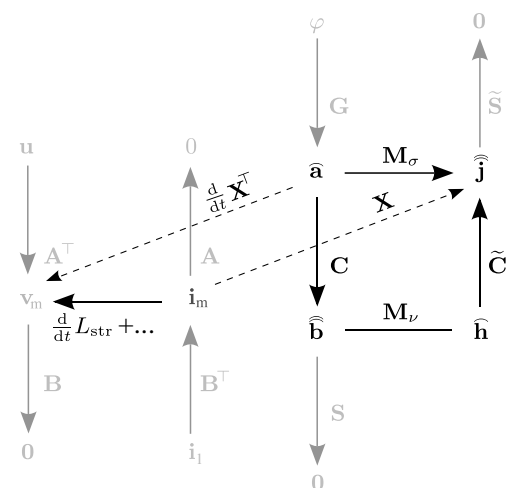
$$\mathbf{L}_M = \mathbf{X}^\top \mathbf{K}_\nu^+ \mathbf{X}$$

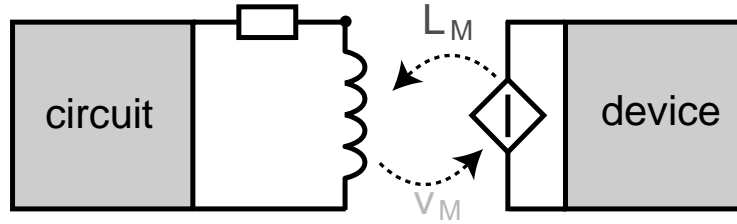
■ solve curl-curl equation for \mathbf{a}

$$\mathbf{a} = \mathbf{K}_\nu^+(\mathbf{a}) \mathbf{X} \mathbf{i}_M - \mathbf{K}_\nu^+(\mathbf{a}) \mathbf{M}_\sigma \frac{d}{dt} \mathbf{a}$$

■ insert into interface \rightarrow circuit description ('Schur complement')

$$\underbrace{\frac{d}{dt} \left(\mathbf{X}^\top \mathbf{K}_\nu^+(\mathbf{a}) \mathbf{X} \mathbf{i}_M \right)}_{:= \mathbf{L}_M(\mathbf{a})} - \underbrace{\frac{d}{dt} \left(\mathbf{X}^\top \mathbf{K}_\nu^+(\mathbf{a}) \mathbf{M}_\sigma \frac{d}{dt} \mathbf{a} \right)}_{:= \phi_M(\mathbf{a})} + \mathbf{R}_M \mathbf{i}_M = \mathbf{v}_M$$





■ Circuit:

$$\mathbf{A}_C \frac{d}{dt} \mathbf{q}_C(\mathbf{A}_C^\top \mathbf{u}, t) + \dots + \mathbf{A}_M \mathbf{R}_M^{-1} \mathbf{A}_M^\top \mathbf{u} + \mathbf{A}_M \mathbf{i}_M = 0$$

$$\frac{d}{dt} (\mathbf{L}_M \mathbf{i}_M + \phi_M) - \mathbf{A}_M^\top \mathbf{u} = 0$$

$$\frac{d}{dt} \phi_L(\mathbf{i}_L, t) - \mathbf{A}_L^\top \mathbf{u} = 0$$

$$\mathbf{A}_V^\top \mathbf{u} - \mathbf{v}_s(t) = 0$$

■ Field:

$$\mathbf{M}_\sigma \frac{d}{dt} \mathbf{a} + \mathbf{C}^\top \mathbf{M}_\nu \mathbf{C}(\mathbf{a}) \mathbf{a} - \mathbf{X} \mathbf{i}_M = 0$$

$$\frac{d}{dt} \mathbf{X}^\top \mathbf{a} + \mathbf{R}_M \mathbf{i}_M - \mathbf{A}_M^\top \mathbf{u} = 0$$

$$\mathbf{L}_M - \mathbf{X}^\top \mathbf{K}_\nu^+(\mathbf{a}) \mathbf{X} = 0 \quad \text{and} \quad \phi_M - \mathbf{X}^\top \mathbf{K}_\nu^+(\mathbf{a}) \mathbf{M}_\sigma \frac{d}{dt} \mathbf{a} = 0$$

Iterative Solution of Coupled Problem

Electromagnetic field problem first

$$\mathbf{M}_\sigma \frac{d}{dt} \mathbf{a}^{(i+1)} + \mathbf{K}_\nu(\mathbf{a}^{(i+1)}) \mathbf{a}^{(i+1)} - \mathbf{X} \mathbf{i}_M^{(i+1)} = 0$$

$$\frac{d}{dt} \mathbf{X}^\top \mathbf{a}^{(i+1)} + \mathbf{R}_M \mathbf{i}_M^{(i+1)} - \mathbf{A}_M^\top \mathbf{u}^{(i)} = 0$$

$$\mathbf{L}_M^{(i+1)} - \mathbf{X}^\top \mathbf{K}_\nu^+(\mathbf{a}^{(i+1)}) \mathbf{X} = 0$$

$$\phi_M^{(i+1)} - \mathbf{X}^\top \mathbf{K}_\nu^+(\mathbf{a}^{(i+1)}) \mathbf{M}_\sigma \frac{d}{dt} \mathbf{a}^{(i+1)} = 0$$

Circuit problem next

$$\mathbf{A}_C \frac{d}{dt} \mathbf{q}_C(\mathbf{A}_C^\top \mathbf{u}^{(i+1)}, t) + \dots + \mathbf{A}_M \mathbf{R}_M^{-1} \mathbf{A}_M^\top \mathbf{u}^{(i+1)} + \mathbf{A}_M \mathbf{i}_M^{(i+1)} = 0$$

$$\frac{d}{dt} (\mathbf{L}_M^{(i+1)} \mathbf{i}_M^{(i+1)} + \phi_M^{(i+1)}) - \mathbf{A}_M^\top \mathbf{u}^{(i+1)} = 0$$

$$\frac{d}{dt} \phi_L(\mathbf{i}_L^{(i+1)}, t) - \mathbf{A}_L^\top \mathbf{u}^{(i+1)} = 0$$

$$\mathbf{A}_V^\top \mathbf{u}^{(i+1)} - \mathbf{v}_s(t) = 0$$

Convergence not guaranteed for DAEs; depends on coupling!

0 Initialization.

Set first time window size H_0 , window counter $n := 0$, sweep counter $k := 0$ and initial values ϕ_0 , \mathbf{q}_0 and \mathbf{a}_0 . Go to Step 1).

1 Guess.

Set window $T_{n+1} = T_n + H_n$ and extrapolate the circuit solution

$$\mathbf{v}_M^{(0)}(t) := \mathbf{A}_M^\top \mathbf{u}^{(k)}(T_n) \quad \text{for } t \in [T_n, T_{n+1}],$$

where $\mathbf{u}^{(k)}$ is the waveform of the node potentials from the last iteration on the previous window. Set $k := 1$ and start dynamic iteration, i.e., go to Step 2).

2 Solve the DAE initial value problems.

- 1 Adaptive time-integration of the field problem on time window $t \in [T_n, T_{n+1}]$ with initial value \mathbf{a}_0

$$\mathbf{M}_{\tilde{\sigma}} \frac{d}{dt} \mathbf{a}^{(k)} + \mathbf{C}^\top \mathbf{M}_\nu \mathbf{C}(\mathbf{a}^{(k)}) \mathbf{a}^{(k)} = \mathbf{X} \mathbf{i}_M^{(k)},$$

$$\frac{d}{dt} \mathbf{X}^\top \mathbf{a}^{(k)} + \mathbf{R}_M \mathbf{i}_M^{(k)} = \boxed{\mathbf{v}_M^{(k-1)}}.$$

- 2 Computing a reduced order representation, e.g.

$$\phi_M(\mathbf{i}_L, t) := \mathbf{L}_M^{(k)}(t) \mathbf{i}_L + \phi_{\text{eddy}}^{(k)}(t).$$

- 3 Adaptive time-integration of the circuit IVP on time window $t \in [T_n, T_{n+1}]$ with initial values ϕ_0 and \mathbf{q}_0

$$\mathbf{A}_C \frac{d}{dt} \mathbf{q}^{(k)} + \mathbf{A}_M \mathbf{R}_M^{-1} \mathbf{A}_M^\top \mathbf{u}^{(k)} + \mathbf{A}_V \mathbf{i}_V^{(k)} + \mathbf{A}_I \mathbf{i}_S^{(k)}(t) + \mathbf{A}_M \mathbf{i}_L^{(k)} = 0,$$

$$\frac{d}{dt} \phi^{(k)} - \mathbf{A}_M^\top \mathbf{u}^{(k)} = 0, \quad \phi^{(k)} - \boxed{\phi_M(\mathbf{i}_L^{(k)}, t)} = 0,$$

$$\mathbf{q} - \mathbf{q}_C(\mathbf{A}_C^\top \mathbf{u}^{(k)}, t) = 0, \quad \mathbf{A}_V^\top \mathbf{u}^{(k)} - \mathbf{v}_s(t) = 0.$$

- 4 Compute the field excitation from the node-potentials and go to Step 3)

$$\mathbf{v}_M^{(k)}(t) := \mathbf{A}_M^\top \mathbf{u}^{(k)}(t).$$

4 Sweep control.

If e.g. $\text{norm}(\mathbf{i}_L^{(k)} - \mathbf{i}_M^{(k)}) > \text{tol}$, then repeat step, i.e., set $k := k + 1$ and go to Step 2), else goto Step 4)

5 Next window.

If $T_{n+1} \geq t_e$ then go to Step 5), else set new initial values

$$\phi_0 := \phi^{(k)}(T_{n+1}), \quad \mathbf{q}_0 := \mathbf{q}^{(k)}(T_{n+1}) \quad \text{and} \quad \mathbf{a}_0 := \mathbf{a}^{(k)}(T_{n+1}).$$

Determine the new window size H_{n+1} , e.g. from the step size predictor of the time-integrators in Step 2a) and 2b). Go to Step 1) with $n := n + 1$.

6 Stop.



Dynamic Iteration: Fixed Point Analysis

Consider DAEs of Index-1

$$\dot{\mathbf{y}}_i^{(k+1)} = \mathbf{F}_i(\mathbf{y}^{(k+1)}, \mathbf{z}^{(k+1)}, \mathbf{y}^{(k)}, \mathbf{z}^{(k)}), \quad 0 = \mathbf{G}_i(\mathbf{y}^{(k+1)}, \mathbf{z}^{(k+1)}, \mathbf{y}^{(k)}, \mathbf{z}^{(k)})$$

Errors

■ differential splitting error

$$\delta_y^{(k)} = \max_{t \in [T_n, T_{n+1}]} |\tilde{\mathbf{y}}^{(k)}(t) - \mathbf{y}(t)|$$

■ algebraic splitting error

$$\delta_z^{(k)} = \max_{t \in [T_n, T_{n+1}]} |\tilde{\mathbf{z}}^{(k)}(t) - \mathbf{z}(t)|$$

Fixed point iteration

$$\begin{bmatrix} \delta_y^{(k+1)} \\ \delta_z^{(k+1)} \end{bmatrix} \leq \begin{bmatrix} \mathcal{O}(H) & \mathcal{O}(H) \\ \beta + \mathcal{O}(H) & \alpha + \mathcal{O}(H) \end{bmatrix} \begin{bmatrix} \delta_y^{(k)} \\ \delta_z^{(k)} \end{bmatrix} + \text{"error in initial values"}$$

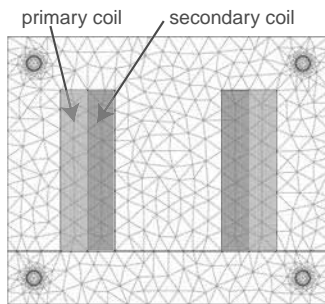
Theorem: Convergence

The iteration converges if the coupling between old and new algebraic variables is weak: $\alpha = \left\| \left(\frac{\partial \mathbf{G}}{\partial \mathbf{z}^{(k+1)}} \right)^{-1} \frac{\partial \mathbf{G}}{\partial \mathbf{z}^{(k)}} \right\| < 1$ since „convergence rate“ = „spectral radius“ < 1

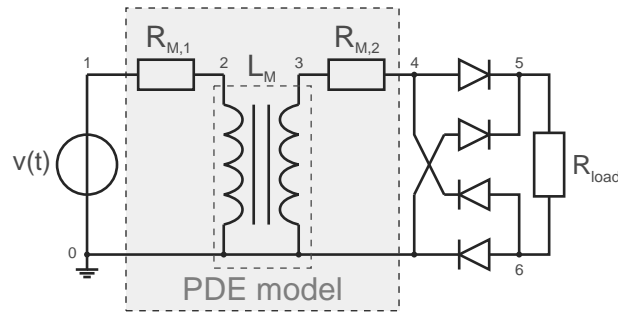


Convergence of Field/Circuit Problem

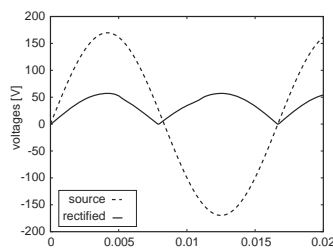
discretization by FEMM/Triangle, www.femm.info



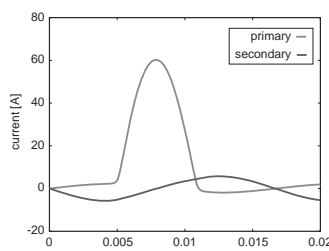
(a) 2D transformer FEMM



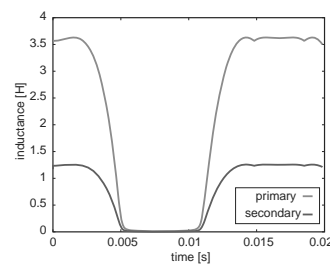
(b) rectifier circuit with MQS field device



(c) voltage drop



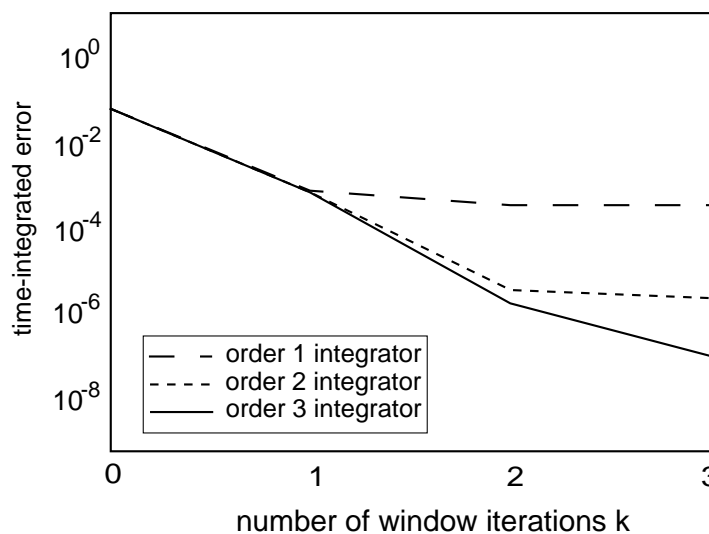
(d) currents



(e) self-inductances



Order of Numerical Time Integration

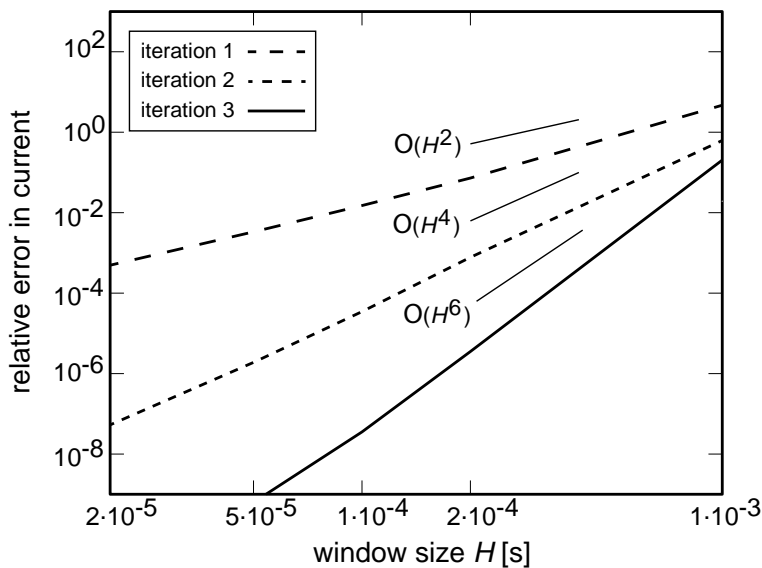


- accuracy depends on time integration order and iterations
- higher order time integrator needs iterations
- low order time integration can be done without iterations

Contraction factor $\alpha = 0 \rightarrow$ convergence guaranteed



Global Convergence in Time with Rate $\mathcal{O}(H^2)$



- fixed windows H
- no iteration control (fixed $k = 1, 2, 3$)
- interpolation by high-order splines
- 5th order time integrator RADAU5

Coupling coefficient $\beta = 0 \rightarrow$ convergence with $\mathcal{O}(H^2)$

Conclusions

Summary

- DAEs are more difficult than ODEs
- index-1 DAEs are mostly harmless
- convergence of dynamic iteration is not unconditional
- convergence is (super) linear at best
- the rate may depend on the window size
- sophisticated interfaces decouple problems better
i.e., larger window size, less communication

Publications (for example):

- (1) Andreas Bartel, Markus Brunk, Michael Günther, and Sebastian Schöps. "Dynamic Iteration for Coupled Problems of Electric Circuits and Distributed Devices". In: SIAM Journal on Scientific Computing 35.2 (Mar. 2013), B315—B335. doi: 10.1137/120867111.
- (2) Andreas Bartel, Markus Brunk, and Sebastian Schöps. "On the Convergence Rate of Dynamic Iteration for Coupled Problems With Multiple Subsystems". In: Journal of Computational and Applied Mathematics (2013). issn: 0377-0427. doi: 10.1016/j.cam.2013.07.031. In press.
- (3) Sebastian Schöps, Andreas Bartel, and Michael Günther. "An Optimal p-Refinement Strategy for Dynamic Iteration of Ordinary and Differential Algebraic Equations". In: Proceedings in Applied Mathematics and Mechanics. Vol. 13. 1. Nov. 29, 2013, pp. 549–552. doi: 10.1002/pamm.201310262.

Über die Simulation differential-algebraischer Modelle mit nichttrivialem Index

Carina Pöll¹, Irene Hafner², Bernhard Heinzl³, Felix Breiteneker¹

¹TU Wien, Institut für Analysis und Scientific Computing

²dwh GmbH Simulation Services, Wien

³TU Wien, Institut für Rechnergestützte Automation

carina.poell@tuwien.ac.at

Durch die Verwendung von komponentenbasierten Simulationswerkzeugen zur Modellierung physikalischer Systeme entstehen Gleichungssysteme mit Index höherer Ordnung. Dieses Paper soll einen Überblick über gängige Methoden der Indexreduktion bieten, wobei eine Klassifikation der verschiedenen Ansätze vorgenommen wird. Im letzten Abschnitt werden verschiedene Methoden anhand eines Beispiels vorgeführt.

1 Einleitung

Eine komponentenbasierte akausale Modellbeschreibung für physikalische Systeme, wie beispielsweise Modelica oder MATLAB/SimScape, führt in der Regel auf differential-algebraische Gleichungssysteme (DAEs) mit nichttrivialem differentiellen Index. Das numerische Lösen von DAEs mit Index höherer Ordnung ist mit herkömmlichen ODE-Lösungsmethoden im Allgemeinen sehr komplex und daher numerisch aufwändig bzw. kann sogar unmöglich sein. Diese Problematik führt auf die sogenannte Indexreduktion, bei dem die gegebene DAE in eine DAE mit Index niedrigerer Ordnung oder eine ODE umformuliert wird. Aufgrund der großen Unterschiede (Struktur, Eigenschaften, etc.) von DAEs findet man in der Literatur eine ganze Reihe verschiedener Ansätze und Methoden zur Reduktion des differentiellen Index. Einige dieser Ansätze sollen in dieser Arbeit gegenübergestellt und anhand von Fallstudien evaluiert werden.

2 Grundlegende Definitionen

Ein differential-algebraisches Gleichungssystem ist im Allgemeinen gegeben durch die implizite Gleichung

$$F(t, x, \dot{x}) = 0, \quad (1)$$

mit $F: I \times D_x \times D_{\dot{x}} \rightarrow \mathbb{R}^n$ ($n \in \mathbb{N}$), wobei I ein reelles Intervall ist, \dot{x} die Ableitung von x nach t symbolisiert und $D_x, D_{\dot{x}}$ offene Teilmengen von \mathbb{R}^n sind (siehe [1]). Ein differential-algebraisches Gleichungssystem zeichnet sich dadurch aus, dass differentielle und algebraische Gleichungen auftreten,

wobei die algebraischen Gleichungen, die die Form $g(x) = 0$ mit einer Funktion $g: \mathbb{R}^n \rightarrow \mathbb{R}^k$ mit $k < n$ besitzen, auch Zwangsbedingung genannt werden. Gleichung (1) besitzt den differentiellen Index $m \in \mathbb{N}$, wenn m die minimale Anzahl an Ableitungen darstellt so, dass sich aus dem System

$$F(t, x, \dot{x}) = 0, \frac{dF(t, x, \dot{x})}{dt} = 0, \dots, \frac{d^m F(t, x, \dot{x})}{dt^m} = 0 \quad (2)$$

ein gewöhnliches Differentialgleichungssystem (ODE) extrahieren lässt, das durch algebraische Umformungen auf die Form $\dot{x} = \varphi(t, x)$ mit $\varphi: I \times D_x \rightarrow \mathbb{R}^n$ gebracht werden kann (siehe [2]).

3 Methoden der Indexreduktion

Die folgende Klassifikation der Methoden zur Indexreduktion ist aus [2] entnommen.

3.1 Indexreduktion durch Differentiation

Ein erster Ansatz zur Reduktion des Index besteht darin, die Zwangsbedingung

$$g(x) = 0 \quad (3)$$

zu differenzieren und die Zwangsbedingung durch ihre Ableitung zu ersetzen. Dieses Vorgehen wird durchgeführt bis das System den Index 1 besitzt. Ein Problem, das sich durch dieses Vorgehen ergibt, ist die Tatsache, dass durch das Ableiten Information verloren geht, da die für die Rückintegration notwendigen Anfangswerte unbekannt sind. Dadurch kann es zum sogenannten numerischen „Drift-off“ kommen, bei dem sich die numerisch berechnete Lösung von der exakten Lösung entfernt.

Ein etwas anderer Ansatz ist die Baumgarte-Methode (siehe [3]). Diese Methode ersetzt die Zwangsbedingung (3) durch eine Linearkombination von g, \dot{g} und \ddot{g} , das heißt statt (3) wird folgende Gleichung verwendet

$$\ddot{g} + 2\alpha\dot{g} + \beta^2 g = 0. \quad (4)$$

Die Methode wird dadurch motiviert, dass durch die Gestalt von Gleichung (4) bzw. das Auftreten der ursprünglichen Zwangsbedingung keine Information verloren geht und daher keine Anfangswerte notwendig sind. Die Konstanten α und β in Gleichung (4) sollen so gewählt werden, dass die Differentialgleichung asymptotisch stabil ist, das heißt $\alpha > 0$. Ein Problem dieser Methode stellt die Wahl der beiden Konstanten dar.

3.2 Stabilisierung durch Projektion

Bei diesem Ansatz wird die numerische Lösung, wenn sie die Zwangsbedingung nicht mehr erfüllt, das heißt von der tatsächlichen Lösung „wegdriftet“, auf die durch die Zwangsbedingung implizit definierte Mannigfaltigkeit projiziert. Diese sogenannte Lösungsmannigfaltigkeit ist daher gegeben durch

$$M = \{x \in \mathbb{R}^n : g(x) = 0\}. \quad (5)$$

Beispielsweise wird bei einem Index-2-Problem das durch Differenzieren der Zwangsbedingung erhaltene Index-1-Problem numerisch gelöst und im Fall, dass die Lösung die ursprüngliche Zwangsbedingung nicht mehr erfüllt, wird die numerische Lösung auf die durch die ursprüngliche Zwangsbedingung definierte Mannigfaltigkeit projiziert.

3.3 Methoden basierend auf lokaler Zustands-transformation

Bei dieser Methode wird die Lösung der DAE statt auf dem gesamten Zustandsraum nur auf einer Mannigfaltigkeit gesucht, die implizit definiert ist durch die Zwangsbedingung (siehe Gleichung (5)). Dadurch vereinfacht sich das Lösen einer DAE zum Lösen einer ODE auf der Mannigfaltigkeit M , die durch Einführung lokaler Koordinaten leichter gelöst werden kann.

4 Fallstudie

Als Anschauungsbeispiel wird die Bewegung eines Pendels auf einer Kreisbahn in kartesischen Koordinaten verwendet, der Einfachheit halber werden die Masse und die Länge des Seils mit 1 angenommen.

Diese ist in expliziter Form gegeben durch die folgenden Gleichungen (siehe [4])

$$\dot{x} = v_x \quad (6)$$

$$\dot{y} = v_y \quad (7)$$

$$\dot{v}_x = - \quad (8)$$

$$\dot{v}_y = g - \quad (9)$$

$$x^2 + y^2 = 1, \quad (10)$$

wobei die Kraft, die das Pendel auf der Kreisbahn hält, und g die Erdbeschleunigung bezeichnet. Mathematisch gesehen kann als Lagrange Multiplikator interpretiert werden. Die Zwangsbedingung ist gegeben durch Gleichung (10). Die DAE (6)–(10) ist ein Index-3-Problem, was typisch für mechanische Systeme ist (siehe [2], [5]). Die Baumgarte-Methode angewendet auf diese Zwangsbedingung, liefert

$$2(v_x^2 + v_y^2 - x^2 + gy - y^2) + 4\alpha(xv_x + yv_y) + \beta^2(x^2 + y^2 - 1) = 0. \quad (11)$$

Das Lösen der DAE (6)–(10) mit Hilfe einer Lösungsmannigfaltigkeit ist in diesem Fall durch Transformation auf Polarkoordinaten sehr einfach möglich (vgl. Abschnitt 3.3). In [5] werden außerdem verschiedene Mehrschrittverfahren zur numerischen Lösung des Gleichungssystems (6)–(10) diskutiert.

5 Referenzen

- [1] P. Kunkel und V. Mehrmann. *Differential-Algebraic Equations: Analysis and Numerical Solution*. European Mathematical Society, Finnland, 2006.
- [2] E. Hairer und G. Wanner. *Solving Ordinary Differential Equations II*. Springer, Deutschland, 2002.
- [3] E. Eich und M. Hanke. *Regularization Methods for Constrained Mechanical Multibody Systems*. ZAMM - Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, S. 761–773, 1995.
- [4] F. E. Cellier und E. Kofman. *Continuous System Simulation*. Springer, USA, 2006.
- [5] J. Frochte. *Evaluation and Adaption of Techniques for Higher Index DAE with Respect to Real-Time Simulation*. SNE Simulation Notes Europe, S. 45–52, 2012.

Universelle OTA-Testbench

Andreas Gerlach¹, Moritz Junge², Jürgen Scheible¹

¹Hochschule Reutlingen, Robert Bosch Zentrum für Leistungselektronik

²Universität Bremen

andreas.gerlach@reutlingen-university.de

Es wird eine universell einsetzbare Testbench zur Simulation von integrierten Schaltungen innerhalb der OTA-Schaltungsklasse (Operational Transconductance Amplifier; Transkonduktanzverstärker) vorgestellt.

Transkonduktanzverstärker sind in der analogen Schaltungstechnik weit verbreitet und daher von großer Bedeutung. Sie treten sowohl als eigenständige Schaltungen innerhalb eines Chips, sowie als Bestandteil anderer Schaltungen (z.B. als erste und zweite Stufe von Operationsverstärkern) auf. Es kann davon ausgegangen werden, dass heute kaum ein analoger oder Mixed-Signal-Chip gefertigt wird, in dem keine Transkonduktanzverstärker verbaut sind.

Die Entscheidungsfindung des Entwicklers bei der Auslegung eines OTAs beruht maßgeblich auf einer anwendungsspezifischen Simulation. Die Erstellung einer eigenen Testbench für jede Anwendung bedeutet allerdings einen hohen Zeitaufwand und erschwert den Vergleich der Simulationsergebnisse unterschiedlicher Schaltungsvarianten. Durch eine universelle Testbench kann zum einen der Zeitaufwand verringert werden, zum anderen können nun Simulationsergebnisse direkt miteinander verglichen werden. Hierdurch wird die Entscheidungsfindung des Entwicklers objektiviert und beschleunigt. Neben dem Vergleich unterschiedlicher Schaltungen innerhalb einer Technologie ist auch der Vergleich einer Schaltung in unterschiedlichen Technologien denkbar.

Die Idee einer universell anwendbaren Testbench lässt sich auch auf andere analoge Schaltungsklassen anwenden und damit als Prinzip verallgemeinern.

1 Einleitung

Beim Entwurf analoger Schaltungen ist die Simulation für die Verifikation der Funktion unersetzlich. Die Testbench als Zusammenfassung der relevanten Simulationen ist somit ein fester Bestandteil eines jeden Entwurfsablaufs. Testbenches werden vom Entwickler schaltungs- und anwendungsspezifisch erzeugt. Momentan bieten Simulationsumgebungen keine fertigen Testbenches (d.h. keine Testbenchbibliothek) zur Simulation ganzer Schaltungsklassen an.

In diesem Artikel wird eine universelle Testbench vorgestellt, die für die Simulation von Schaltungen der Schaltungsklasse Transkonduktanzverstärker (OTA) entwickelt wurde. Die Wahl der Schaltungsklasse wird in Kap. 2 kurz begründet. Kap. 3 zeigt den OTA und seine wichtigsten Varianten. Anschließend wird in Kap. 4 die universelle Testbench und der Arbeitsablauf ihrer Anwendung vorgestellt. Kap. 5 schließt mit einer Zusammenfassung der Vorteile und einem kurzen Ausblick.

2 Anwendungshäufigkeit von OTAs

Bei der Analyse aktueller Chips, in der typische ICs der Automobilelektronik untersucht wurden, wurde festgestellt, dass Schaltungen aus der OTA-Schaltungsklasse die am meisten verbreiteten analogen Schaltungen im IC-Design sind, sieht man von elementaren Grundsaltungen, wie Gate-, Source- und Drain-Schaltung sowie Stromspiegel ab.

OTA-Schaltungen kommen innerhalb eines Projekts oft in mehreren topologisch unterschiedlichen Varianten zum Einsatz. Ihr häufiger Gebrauch prädestiniert sie somit für den Entwurf einer universellen Testbench.

3 OTA-Typen

Der Transkonduktanzverstärker besteht wie ein Operationsverstärker aus einem Differenzeingangspaar, hat aber einen hochohmigen Ausgang. Sein Verhalten wird in [1] so beschrieben:

„Er verhält sich wie eine Stromquelle, deren Strom durch die Eingangsspannungsdifferenz U_D gesteuert wird.“

OTA-Schaltungen lassen sich anhand ihrer Eingangsspannungsbereiche in die drei nachfolgend beschriebenen Typen untergliedern.

- „P-Eingang“-OTA: bedingt durch pnp- oder P-Kanal-Eingangstristoren lassen sich Eingangsspannungen im unteren Versorgungsspannungsbereich analysieren, d.h. vom unteren Rail¹ (auch knapp unterhalb) bis etwa über die halbe Versorgungsspannung. Sie erreichen aber das obere Rail nicht.
- „N-Eingang“-OTA: im Gegensatz zum „P-Eingang“-OTA liegt der Funktionsbereich bei Eingangssignalen im oberen Versorgungsspannungsbereich. Sie erreichen das untere Rail nicht vollständig, funktionieren jedoch bis zur maximalen Versorgungsspannung und häufig auch darüber.
- „Rail-to-Rail-Eingang“-OTA: dieser Typ beschreibt eine Kombination der vorangegangenen OTAs. Er eignet sich für Signale innerhalb des gesamten Versorgungsspannungsbereiches, in Spezialfällen auch ober- und unterhalb der Rails (Over-the-rails-OTA).

Um bereits auf der Symbolebene zwischen den drei Typen unterscheiden zu können, wird eine optionale Erweiterung der OTA-Symbole nach Abb.1 vorgeschlagen.

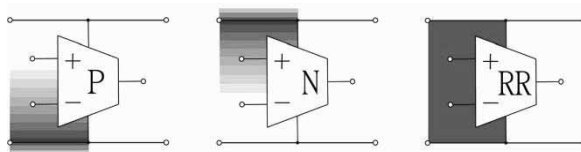


Abbildung 1. Einteilung der OTA-Topologievarianten in drei Gruppen. Von links nach rechts: Symbole für „P-Eingang“-OTA, „N-Eingang“-OTA und „Rail-to-Rail-Eingang“-OTA. Die Schattierungen deuten den Eingangsspannungsbereich des jeweiligen OTA-Typs an.

Die hier betrachteten OTA-Varianten besitzen nur einen Ausgang. Ein derartiger Ausgang wird als un-

symmetrisch (single ended) bezeichnet. Es gibt auch OTA-Schaltungen mit symmetrischen Ausgängen. Diese besitzen zwei Ausgänge.

Die zuvor durchgeführte Chipuntersuchung hat gezeigt, dass die „P-Eingang“-OTAs die am häufigsten verwendete Gruppe bilden. Dies ist damit zu erklären, dass die meisten Signale in der Praxis nahe am unteren Rail liegen. Eine Beispielschaltung eines solchen OTA-Typs zeigt Abbildung 2.

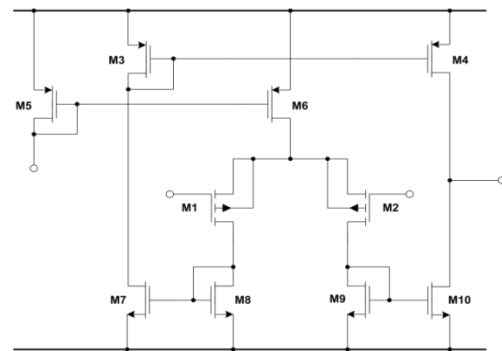


Abbildung 2. Beispiel eines „P-Eingang-OTAs“ mit symmetrischem Eingang und unsymmetrischem Ausgang.

4 Universelle OTA-Testbench

4.1 Aufbau und Anwendung der Testbench

Die hier vorgestellte Testbench besteht im Wesentlichen aus einem Schaltplan und Gleichungen, welche fest in die Simulationsumgebung eingebunden sind.

Der Schaltplan stellt die Testumgebung für die vom Benutzer definierte OTA-Schaltung bereit. Hierzu muss der Benutzer lediglich ein Schaltungssymbol für seinen zu testenden OTA erstellen und dieses in das DUT-Symbol (Device Under Test) innerhalb der Testbench einfügen. Durch diesen einfachen und zeitsparenden Vorgang ist die zu testende OTA-Schaltung in die Testumgebung eingebunden.

Zusätzlich sind in der Testbench alle zur Bestimmung der charakteristischen OTA-Kenngrößen benötigten Gleichungen eingebaut. Nach der Festlegung der arbeitspunktrelevanten Parameter Versorgungsspannung, Referenzstrom und Lastwiderstand in der Testbench werden die Simulationen durchgeführt, welche die Eingangsgrößen für diese Bestimmungsgleichungen liefern.

Die Kenngrößen und Simulationen werden in Kap. 4.3 ausführlich erläutert. Die Reihenfolge der Simula-

¹ Mit unterem und oberem Rail bezeichnet man die unteren und oberen Level der Versorgungsspannung.

tionen ergibt sich hierbei aus einer anwendungsspezifischen Priorisierung der Kenngrößen (s. Kap. 4.3).

Neben der Zeitersparnis bei der Charakterisierung bzw. Dimensionierung einzelner Schaltungen wird durch die Anwendung einer standardisierten Testumgebung auch ein direkter Vergleich der Simulationsergebnisse für unterschiedliche Schaltungen ermöglicht. Der Vergleich wird dabei durch eine tabellarisch einheitliche Darstellung unterstützt. Darüber hinaus sind die Ergebnisse auch grafisch darstellbar.

4.2 Simulationsarten

Die Testbench nutzt folgende Simulationsarten für die Berechnung der charakteristischen Größen des OTAs:

- Transiente Simulation,
- DC-Simulation,
- AC-Simulation,
- Monte-Carlo-Simulation,
- Alterungssimulation.

4.3 Ermittlbare Größen

Die Reihenfolge, in welcher die Simulationen durchgeführt werden, wird – wie bereits angedeutet – anwendungsspezifisch festgelegt. Als erstes erfolgt allerdings grundsätzlich die Berechnung der Steilheit g_m , da sie die primäre Aussage über die Funktion eines OTA macht. Aus den von der Schaltung zu erfüllenden Kriterien ergibt sich eine Gewichtung der weiteren Schaltungskenngrößen, woraus die Reihenfolge der weiteren Simulationen abgeleitet wird.

Die von der Testbench berechenbaren Größen werden nachfolgend aufgeführt und erläutert. Die Reihenfolge der Auflistung entspricht einer beispielhaften Priorisierung.

Steilheit g_m

Die Steilheit beschreibt die Änderung des Ausgangsstroms in Abhängigkeit der Differenzeingangsspannung und stellt das wichtigste Kriterium für den OTA dar. In der Testbench werden drei Methoden zur Bestimmung der Steilheit eingesetzt. Die drei Methoden basieren auf den nachfolgend beschriebenen unterschiedlichen Simulationsarten, welche alle voneinander unabhängig sind.

- Die dc-sweep-Simulation: hierbei wird die Eingangsspannungsdifferenz variiert und die Ausgangsstromänderung ausgewertet.

- Die ac-Simulation.
- Eine transiente Simulation. Aus dieser werden sog. gealterte Transistor-Modelle erzeugt, welche das für den betrachteten Fall resultierende Alterungsverhalten der Transistoren abbilden. Hiermit lässt sich eine Alterungssimulation des OTA durchführen.

Die Anwendung mehrerer voneinander unabhängiger Methoden zur Ermittlung der Steilheit g_m bietet eine Vergleichsmöglichkeit der Ergebnisse. Durch Bestätigung dieser Ergebnisse anhand unterschiedlicher Simulationsarten kann der für die Steilheit berechnete Wert abgesichert werden.

In der bisherigen Anwendung der Testbench wichen die Ergebnisse bei standardmäßigen Einstellungen und eingestelltem Arbeitspunkt über die drei Methoden nur im Promille-Bereich voneinander ab.

Eingangsspannungsbereich

Der Eingangsspannungsbereich ergibt sich aus dem Steilheitsverlauf, aufgetragen über der Gleichakteingangsspannung u_{INCM} .

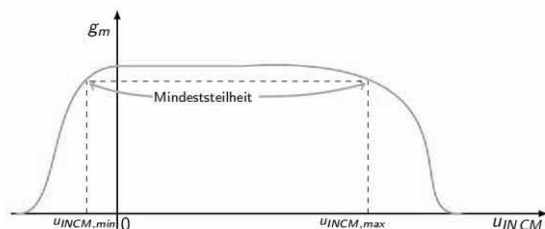


Abbildung 3. Prinzipieller Verlauf der Steilheit einer OTA-Schaltung mit „P-Eingang“.

In der Abbildung 3 wird der prinzipielle Eingangsspannungsbereich einer „P-Eingang-OTA-Schaltung“ dargestellt. Anwendungsabhängig wird eine bestimmte Mindeststeilheit gefordert. Der Schnittpunkt dieser Größe mit dem Steilheitsverlauf der Schaltung definiert den möglichen Eingangsspannungsbereich. Daraus ergeben sich die beiden Grenzen $u_{INCM,min}$ und $u_{INCM,max}$.

Es ergibt sich eine charakteristische „Hut-Funktion“ für den „P-Eingang“-OTA (Abb. 3) und für den „N-Eingang“-OTA. Beim „Rail-to-Rail-Eingang“-OTA ergibt sich eine Kurve, die aus der Superposition der „N-Eingang-OTA“ und „P-Eingang-OTA“-Kurven resultiert.

Maximale Stromaufnahme

Die Gesamtstromaufnahme lässt sich in die Stromaufnahme der ersten und der zweiten Stufe des OTAs aufteilen. Am Beispiel von Abbildung 2 wird eine „P-Eingang“-OTA-Schaltung dargestellt. Hierbei besteht die Eingangsstufe aus den Transistoren M1, M2, M6, M8 und M9. Die zweite Stufe besteht aus den Transistoren M3, M4, M7 und M10.

Darauf aufbauend kann auch die Leistungsaufnahme berechnet werden. Mit Hilfe der Testbench können diese Ergebnisse übersichtlich und standardisiert dargestellt werden.

Offsetspannung

„Sie gibt an, welche Differenzspannung angelegt werden muss, damit die Ausgangsspannungen gleich groß sind oder – bei unsymmetrischen Ausgängen – ein bestimmter Sollwert erreicht wird.“

Diese Definition der Offsetspannung wird in [1] für einen Differenzverstärker (der einfachste OTA) angegeben und ist auf komplexere OTA-Schaltungen übertragbar.

Es gibt zweierlei Ursachen für die Entstehung der Offsetspannung:

- topologische Ursachen,
- durch stochastische Abweichungen verursachte Ungleichheit der Bauelemente, die zum Mismatch führt.

Mit einer nominalen Simulation können eigentlich nur die topologischen Offsetspannungsursachen abgedeckt werden. Es gibt jedoch die Möglichkeit, mit einer DC-Match-Analyse auch die stochastischen Ursachen abzudecken. Eine Monte-Carlo-Simulation liefert jedoch die genaueren Ergebnisse.

Da der stochastische Anteil in der Realität immer auftritt, wird in der hier vorgestellten Testbench bei der Ermittlung der Offsetspannung auf die Monte-Carlo-Simulation zurückgegriffen, deren Ergebnis in Abb. 4 dargestellt ist.

Um einen Vergleich zu ermöglichen, wurde auch die auf der DC-Match-Simulation basierende Methode eingefügt.

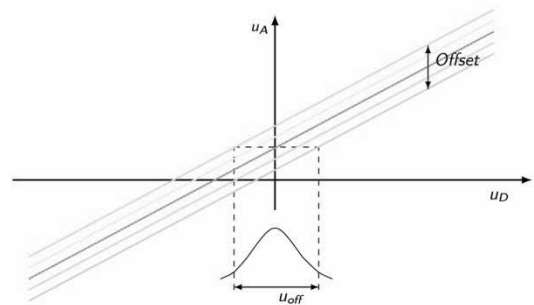


Abbildung 4. Offsetspannung u_{off} wird u.a. mittels einer Monte-Carlo-Simulation als Gauß'sche Verteilung ermittelt.

Weitere Faktoren, die die Offsetspannung beeinflussen und in der Testbench berücksichtigt werden können sind

- die Temperatur,
- Änderungen der Betriebsspannung.

Bode-Diagramm

Das Bode-Diagramm gibt den Betrag der über die Frequenz aufgetragenen Verstärkung in dB, den Phasengang und den markanten -3dB-Punkt wider. Bei einer OTA-Schaltung bestimmt der Eingangswiderstand der nachfolgenden Schaltung den Verlauf des Bode-Diagramms und muss deswegen bei der Simulation berücksichtigt werden.

Anders ist es beim idealen Operationsverstärker; hier ist das Bodediagramm von der nachfolgenden Schaltung unabhängig.

Maximaler Ausgangsstrom

Der maximale Ausgangsstrom ist der Strom, den eine OTA-Schaltung am Ausgang maximal liefern kann. Hierfür wird die Differenz der Eingangsspannung erhöht und der dadurch verursachte Ausgangsstrom grafisch dargestellt. Der Maximalwert wird als Zahlenwert ausgegeben.

Rauschen

Das Rauschen der Transistorschaltungen besteht aus zwei Komponenten: dem $1/f$ -Rauschen und dem thermischen Rauschen. Beide Komponenten werden in dieser Simulation erfasst.

Common Mode Rejection Ratio

Das CMRR ist definiert als Verhältnis von Differenz- und Gleichtaktverstärkung [1].

Es wird aus der ac-Simulation bestimmt, um den kompletten Verlauf über den Frequenzbereich, in dem der OTA arbeitet, zu veranschaulichen.

Power Supply Rejection Ratio

Das PSRR beschreibt die Ausgangsspannungsänderung bei einer Veränderung der Versorgungsspannung. Ihr Wert wird ebenso wie das CMRR aus der ac-Simulation bestimmt. Dadurch wird ersichtlich, wie sich das Brummen von Störquellen unterschiedlicher Frequenzen am Ausgang auswirkt.

Eingangsaussteuerbarkeit

Das „Sweepen“ der Differenzspannung an den Eingangstransistoren hat eine hohe Gate-Source-Spannung zur Folge. Bei der Eingangsaussteuerbarkeit wird überprüft, ab welcher Spannung die Gates durchbrechen. Dies wird ermöglicht durch die in die Simulationsmodelle zusätzlich eingebaute „Smoke Detector“-Funktion.

5 Zusammenfassung und Ausblick

Im vorliegenden Artikel wird eine universelle Testbench für die Simulation von OTA-Schaltungen vorgestellt. Sie besteht aus einem Schaltplan zur Einbindung einer benutzerdefinierten OTA-Schaltung, sowie aus Gleichungen für die Berechnung der charakteristischen Größen einer solchen Schaltung. Hierdurch bietet sie die Möglichkeit einer unkomplizierten, standardisierten und schnellen Überprüfung einer OTA-Schaltung.

Der mit der Erstellung individueller Testbenches verbundene Zeitaufwand lässt sich hiermit deutlich reduzieren. Darüber hinaus bietet die universelle Testbench standardisierte Testbedingungen in einer verifizierten Testumgebung. Dies führt zu einer verbesserten Entwurfssicherheit und ermöglicht zudem einen objektiven Vergleich zwischen verschiedenen Schaltungsvarianten. Sogar ein technologieübergreifender Vergleich ist möglich.

Der Ansatz universeller Testbenches lässt sich prinzipiell auf andere Schaltungsklassen übertragen. Die Entscheidung, den Aufwand zur Erstellung einer universellen Testbench zu betreiben, muss jedoch für weitere Schaltungsklassen individuell getroffen werden. Sie hängt in erster Linie vom Verwendungsgrad der Schaltungsklasse ab, aus der sich die erzielbare Kosten-Nutzen-Relation ergibt. Der erzielbare Zuge-

winn an Entwurfssicherheit kann ebenfalls eine entscheidende Rolle spielen.

In Zukunft wäre denkbar, universelle Testbenches für weitere Schaltungsklassen in Form einer Bibliothek innerhalb der Entwicklungsumgebung anzubieten. Hierdurch hätten Entwickler die Möglichkeit, das Verhalten der unterstützten Schaltungen schnell und unkompliziert zu überprüfen.

6 Literaturverzeichnis

- [1] U. Tietze, Ch. Schenk. *Halbleiter Schaltungstechnik*. Springer Verlag Berlin, Heidelberg, New York, 2002.
- [2] X. Rasmus. *Demystifying the Operational Transconductance Amplifier*. Texas Instruments Application Report, 2013.

Ein Effizienzmodell für getaktete Schaltwandler im Multi-MHz-Bereich

Efficiency Model for Multi-MHz-Converters

Workshop der ASIM/GI-Fachgruppen

20./21. Februar 2014

Achim Seidel

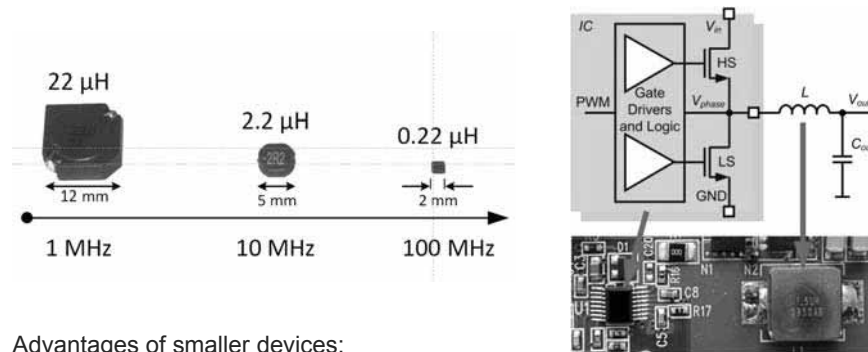
Jürgen Wittmann

Bernhard Wicht

Outline

- Introduction
- Efficiency Model
- Experimental Results
- Analysis with the Efficiency Model
- Conclusion

Cost Reduction by Increasing the Frequency



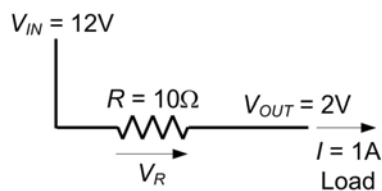
Advantages of smaller devices:

- Less costs, potential for integration
- Better reliability and vibration resistance

How to Convert 12V into 2V?

Using a simple resistor...

It works, but...



Large power loss at the resistor:

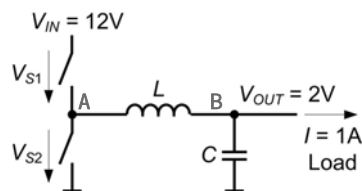
$$P_R = V_R \cdot I = 10V \cdot 1A = 10W$$

(even larger than P_{out} !)

How to Convert 12V into 2V?

Switched mode conversion...

It works and has small losses...



Power loss at the switches: $P_S = V_S \cdot I_S$

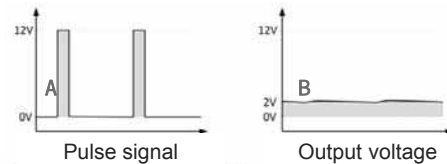
ideally:

off: $I_S = 0 \rightarrow P_S = 0$

on: $V_S = 0 \rightarrow P_S = 0$

in reality >95% power efficiency possible with frequencies in kHz-range

Pulse-Width Modulation...



Achim Seidel, M.Sc., Hochschule Reutlingen, Alteburgstraße 150, 72627 Reutlingen, www.reutlingen-university.de,
T. +49(0)7121 271-7066, F. +49(0)7121 271-90-7066, achim.seidel@reutlingen-university.de

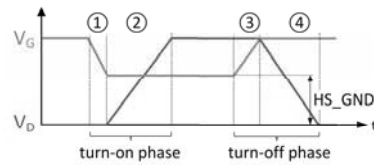
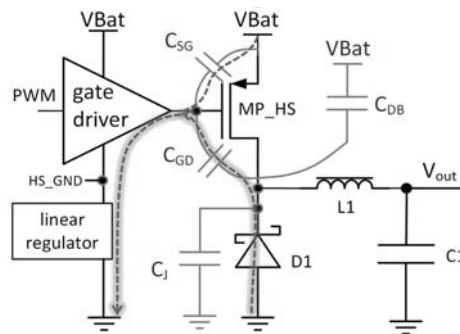
Efficiency Model

Achim Seidel, M.Sc., Hochschule Reutlingen, Alteburgstraße 150, 72627 Reutlingen, www.reutlingen-university.de,
T. +49(0)7121 271-7066, F. +49(0)7121 271-90-7066, achim.seidel@reutlingen-university.de

6

Efficiency Model for Multi-MHz Operation

Example phase ①:



→ Loss cause: C_{GD}
→ Loss location:
Schottky diode, linear
regulator, gate driver

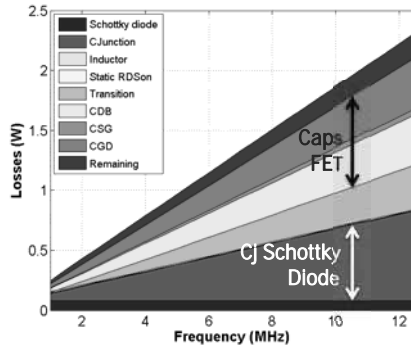
Determining Losses Including Causes and Locations

		loss cause			
		phase 1	phase 2	phase 3	phase 4
loss location	gate driver Low-Side	C_{GD}, C_{SG}			
	gatedriver High-Side			C_{DG}, C_{SG}	
	linear regulator	C_{GD}, C_{SG}	C_{GD}		
	Schottky diode	C_{GD}			
	FET R_{DSon}		C_{GD}, C_{DB}, C_J		
		transition losses, static losses			

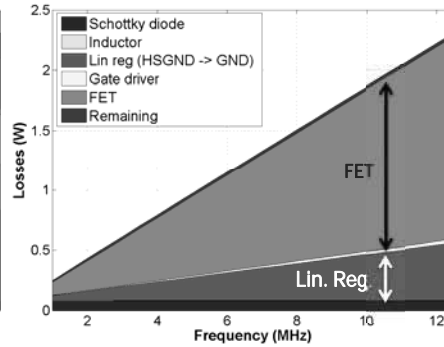
→ Assignment: loss locations and loss causes

Efficiency Model: Analysis of Losses

Loss causes



Loss locations



Iload = 200 mA
Vin = 40 V

Switching losses dominant at high frequency and high input voltage

Achim Seidel, M.Sc., Hochschule Reutlingen, Alteburgstraße 150, 72627 Reutlingen, www.reutlingen-university.de,
T. +49(0)7121 271-7066, F. +49(0)7121 271-90-7066, achim.seidel@reutlingen-university.de

Structure of the Efficiency Model



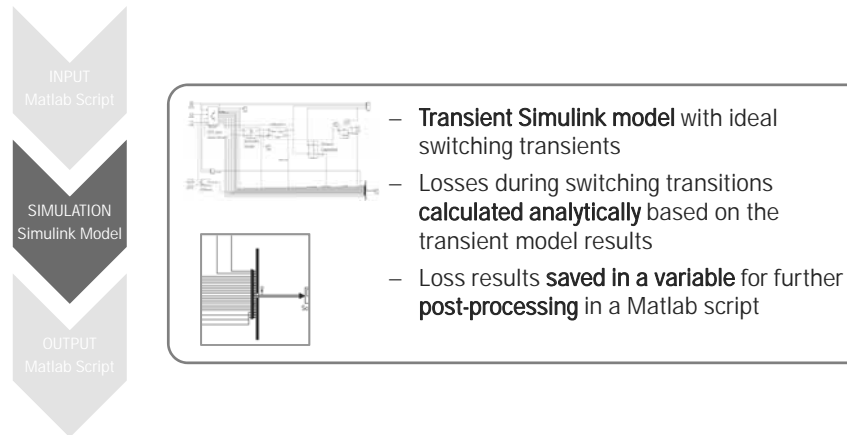
- Input arguments
 - Plot settings
 - Converter parameters (i.e. Iout, Vin, Vout, Rdson, Cpar)
- Pre-calculation, i.e.
 - Initial conditions for the transient Simulink model to achieve faster settling
 - Duty cycle



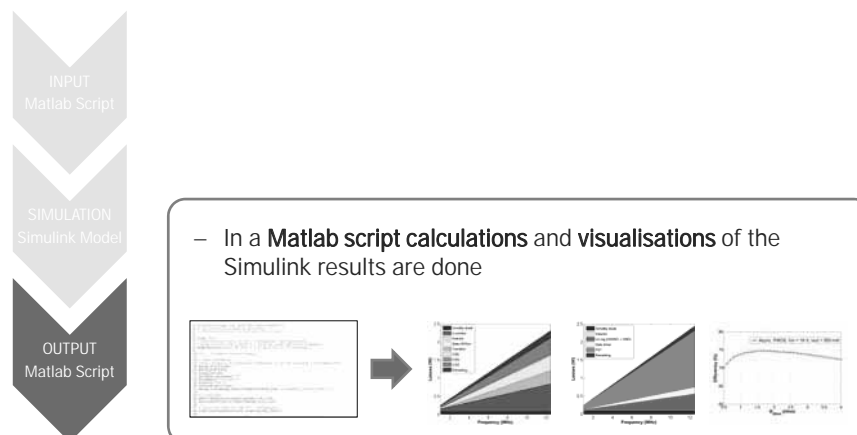
Achim Seidel, M.Sc., Hochschule Reutlingen, Alteburgstraße 150, 72627 Reutlingen, www.reutlingen-university.de,
T. +49(0)7121 271-7066, F. +49(0)7121 271-90-7066, achim.seidel@reutlingen-university.de

10

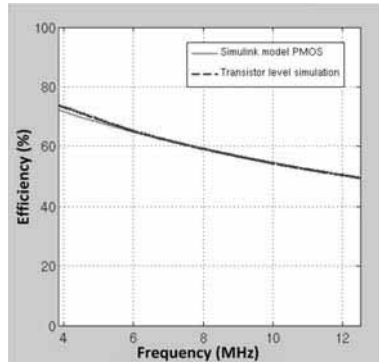
Structure of the Efficiency Model



Structure of the Efficiency Model



Verification: Efficiency Model and Transistor Level Simulation



$V_{in} = 18\text{ V}$
 $I_{load} = 200\text{ mA}$

Deviation between model and simulation < 2 %

Achim Seidel, M.Sc., Hochschule Reutlingen, Alteburgstraße 150, 72627 Reutlingen, www.reutlingen-university.de,
T. +49(0)7121 271-7066, F. +49(0)7121 271-90-7066, achim.seidel@reutlingen-university.de

13

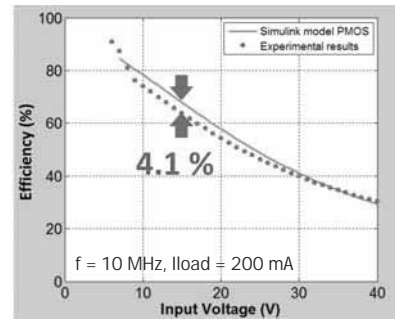
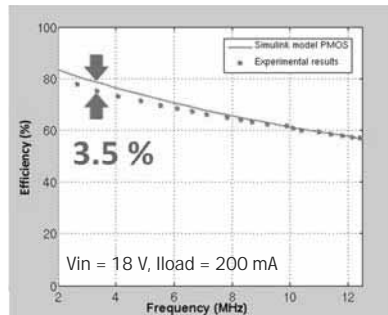
Experimental Results



Achim Seidel, M.Sc., Hochschule Reutlingen, Alteburgstraße 150, 72627 Reutlingen, www.reutlingen-university.de,
T. +49(0)7121 271-7066, F. +49(0)7121 271-90-7066, achim.seidel@reutlingen-university.de

14

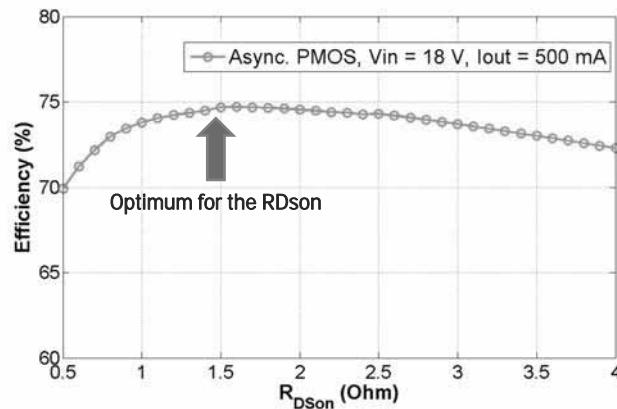
Experimental Results



- Fully operational @ up to 13 MHz and $V_{in} = 40 \text{ V}$
- Efficiency model verified by experimental results

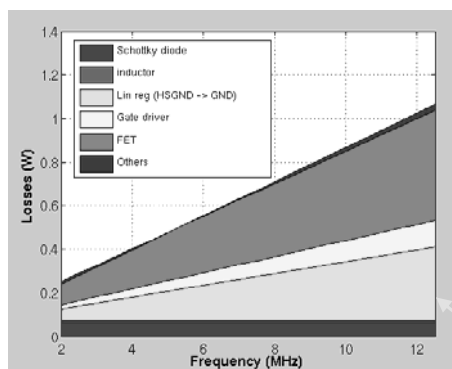
Analysis with the Efficiency Model

Parameter Optimization of a Given Converter Architecture



Achim Seidel, M.Sc., Hochschule Reutlingen, Alteburgstraße 150, 72627 Reutlingen, www.reutlingen-university.de,
T. +49(0)7121 271-7066, F. +49(0)7121 271-90-7066, achim.seidel@reutlingen-university.de

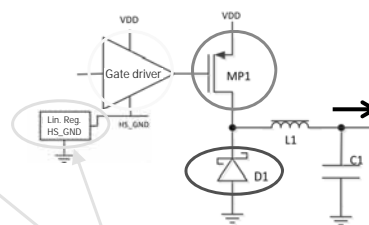
Comparison of Converter Architectures



$V_{in} = 18 \text{ V}$, $I_{load} = 200 \text{ mA}$

Not required with NMOS power FET

→ NMOS typically better than PMOS



Achim Seidel, M.Sc., Hochschule Reutlingen, Alteburgstraße 150, 72627 Reutlingen, www.reutlingen-university.de,
T. +49(0)7121 271-7066, F. +49(0)7121 271-90-7066, achim.seidel@reutlingen-university.de

18

Conclusion

- **MHz converters:**
 - Reduce system size and cost by faster switching (>10 MHz)
 - But: power losses increase, precise efficiency model required
 - Conventional efficiency calculations not suitable anymore
- **The presented efficiency model allows**
 - for dedicated optimization by root cause and loss location
 - an optimization of design parameters
 - to study and compare different converter architectures
- **Measurement results:** Efficiency model matches with less than 3.5 %, main impact from parasitic capacitances

References

- X. Wang, J. Park, E. Van Brunt, and A. Huang, "Switching losses analysis in mhz integrated synchronous buck converter to support optimal power stage width segmentation in cmos technology," in Energy Conversion Congress and Expos. (ECCE), 2010 IEEE, Sept. 2010, pp. 2718 –2724.
- X. Wang and A. Huang, "Capacitor energy variation based designer-side switching losses analysis for integrated synchronous buck converters in cmos technology," in Applied Power Electron. Conf. and Exposition (APEC), 2011 Twenty-Sixth Annual IEEE, March 2011, pp. 1130 –1137.
- J. Wittmann and B. Wicht, "MHz-converter design for high conversion ratio," in 2013 IEEE 25th International Symposium on Power Semi-conductor Devices and ICs (ISPSD), in press.
- J. Wittmann, T. Funk, and B. Wicht, "Gate driver design for fast switching dc-dc converters," in 48th MPC Workshop, Aalen, July 2012, pp. 17–24.
- H. Mattausch et al., Power/HVMOS Devices Compact Modeling. Springer, 2010, ch. 2: "HiSIM-HV: A Scalable, Surface-Potential-Based Compact Model for High-Voltage MOSFETs".

Effektive Fehlerdiagnose von Strukturen mit Digital-Analog-/Analog-Digital-Konvertern mit einem Cloud-basierten Workflow

Matthias Gulbins, André Schneider, Steffen Rülke

Fraunhofer IIS/EAS Dresden

matthias.gulbins@eas.iis.fraunhofer.de

Eine hochkomplexe und zeitaufwändige Aufgabe beim Entwurf integrierter Mixed-Signal-Schaltkreise ist die Fehlerdiagnose. Der vorliegende Beitrag stellt einen auf Cloud-Technologien basierenden Lösungsansatz vor, der Fehler in für solche Schaltkreise typischen Strukturen aus Analog-Digital- und Digital-Analog-Wandlern lokalisiert. Das Diagnoseverfahren (Ergebnis des BMBF-Projektes *DIANA*) beruht auf dem sogenannten Loop-Back-Test, der zwar die Generierung von Testdaten vereinfacht, aber eine Vielzahl von Variantensimulationen mit verschiedenen Simulationsprinzipien und erheblichen Datenmengen erfordert. Diese sollen nunmehr problemangepasst und damit effizient in der Cloud realisiert werden. Für die entsprechende Informationsverarbeitung in der Cloud wurde das in dem Projekte *OptiNum-Grid* entwickelte Framework *GridWorker* adaptiert. Experimente mit ersten Anwendungsbeispielen bestätigen die Leistungsfähigkeit und Praktikabilität des Ansatzes für daten- und verarbeitungsintensive Schaltkreisentwurfsaufgaben.

1 Einleitung

Analog-Digital- und Digital-Analog-Wandler (ADC/DAC) sind typische Schaltungskomponenten vieler Mixed-Signal-Schaltkreise, z.B. im Bereich der Automobilelektronik. Der Test solcher Strukturen und die Diagnose der Fehlerursachen gestalten sich als überaus komplex und aufwendig. Eine Strategie zur Reduktion der Test- und Diagnosekosten ist ein kombiniertes Vorgehen aus strukturellen und simulations-methodischen Ansätzen. Neben einer Loop-Back-Struktur für den Test kommen hierarchische Ansätze und Cloud-Technologien für die Fehlerdiagnose zur Anwendung.

Bei einer Loop-Back-Struktur werden für den Test DAC und ADC in Reihe geschaltet, so dass das Ausgangssignal des ADCs mit dem Eingangssignal des DACs verglichen werden kann. Vorteile sind, dass

- der Test digital abläuft, was eine einfache Stimulierung und eine reproduzierbare Fehlerdiagnose ermöglicht,
- die Fehler der Wandler direkt auf der elektrischen Netzwerkebene definiert werden und
- Schwankungen der Herstellungsparameter von Schaltkreisen unmittelbar beim Test berücksichtigt werden.

Für die Beherrschbarkeit der hochkomplexen und (entsprechend der Parametervariationen und der Vielfalt von möglichen Fehlerorten und -werten) sehr

umfangreichen Fehlersimulationen ist folgendes vorgesehen:

- Einerseits ist ein hierarchisches Vorgehen erforderlich: Alle Fehlersimulationen der Wandler werden auf Netzwerkebene ausgeführt, die Ergebnisse in die funktionelle Ebene überführt und die Simulationen in der Loop-Back-Umgebung sowie die Fehlerdiagnose auf funktioneller Ebene durchgeführt.
- Andererseits werden alle Operationen (Simulationen und Konvertierungen auf Netzwerk- und Funktionsebene) in einer Cloud unter Verwendung geeigneter Virtualisierungstechnologien (*GridWorker*) ausgeführt.

Mit dem vorgestellten Lösungsansatz wird die Leistungsfähigkeit der Cloud-Technologien für die zunehmend daten- und verarbeitungsintensiven Schaltkreisentwurfsaufgaben demonstriert. Gerade aus der immer höheren Integrationsdichte und der zunehmenden Verarbeitungsbreite der Schaltungskomponenten erwachsen diesbezüglich erhebliche Anforderungen.

Weiterhin ist der beschriebene Lösungsansatz auch die Grundlage für künftige Dienstleistungsangebote zur Varianten- bzw. Fehlersimulation, Analyse und Diagnose beim Mixed-Signal-Schaltkreisentwurf. Dabei könnten auch die von den Autoren entwickelten Werkzeuge zur Informationsverarbeitung in der Cloud (*GridWorker*) und zur analogen Fehlersimula-

tion (aFSIM) als *Software-as-a-Service* (SaaS) Verwendung finden.

Schließlich wird mit dem Lösungsansatz auch die Qualität der Fehlerdiagnose signifikant erhöht. Das wird möglich, weil dank der Cloud-Technologien anstelle von relativ groben Verhaltensmodellen nunmehr auch genauere Strukturmodelle auf der elektrischen Netzwerkebene verwendet werden können, die das auszuwertende Datenvolumen und Anzahl der erforderlichen Simulationsläufe drastisch ansteigen lassen.

Der weitere Beitrag ist wie folgt gegliedert: Abschnitt 2 erläutert das Fehlerdiagnoseverfahren von Loop-Back-Strukturen, Abschnitt 3 stellt den Cloud-basierten Workflow dafür vor und Abschnitt 4 die Adaption des *Gridworkers* an diesen Workflow. Abschnitt 5 diskutiert die bei Experimenten erreichten Ergebnisse. Eine Zusammenfassung sowie der Ausblick auf weiterführende Arbeiten schließen den Beitrag ab.

2 Prinzip der Fehlerdiagnose für Loop-Back-Strukturen

Wichtige Schritte bei der Fertigung integrierter nanoelektronischer Systeme (System-on-Chip – SoC) sind Fertigungs- und On-Chip-Test sowie – im Fehlerfall – die Fehlerdiagnose. Insbesondere bei SoC mit Mixed-Signal-Komponenten ist dies meist sehr zeit- und kostenaufwendig. Ein Ansatz zur Aufwandsreduktion ist der Loop-Back-Test.

Grundliegende Idee des Loop-Back-Tests ist es, zwei funktional zueinander inverse Blöcke in Reihe zu schalten und den eigentlichen Test dann für diese Zusammenschaltung durchzuführen. Der Testaufwand kann erheblich reduziert werden, weil Ein- und Ausgangssignale der Zusammenschaltung weitgehend einander entsprechen und damit einfach verglichen werden können

Im Projekt DIANA wurde ein Zugang entwickelt, um den Loop-Back-Test auch für Mixed-Signal-Komponenten vom Typ ADC und DAC anzuwenden, wenn diese gemeinsam in einem System-on-Chip-Halbleiterbaustein integriert sind. Die Zusammenschaltung von DAC und ADC zu einer Loop-Back-Struktur mit digitalen Ein- und Ausgängen ermöglicht den Test auf digitaler Basis. Dabei erfolgt ein statisches Vorgehen, bei dem der ADC mit einer Rampenfunktion und der DAC mit einer Treppenfunktion stimuliert werden. Nach der Stimulierung des DACs

mit einer digitalen Treppenfunktion wird das analoge Ausgangssignal des DACs auf den analogen Eingang des ADCs geschaltet. Ein zwischengeschalteter Tiefpass erzeugt analoge Zwischenwerte, um die Rampenfunktion besser nachzubilden und die erforderliche Testgenauigkeit zu erreichen (vergl. Abbildung 1). Die Ausgabe des ADCs ist eine Folge aus Digitalwerten.

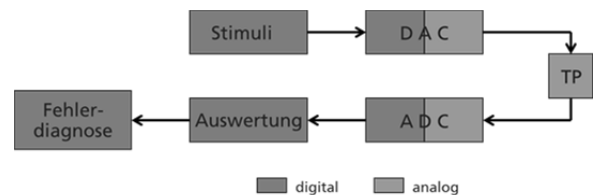


Abbildung 1. Loop-Back-Struktur aus Digital-Analog- und Analog-Digital-Wandler

Aufgrund der großen Datenmenge, werden aus der Ausgabe des ADCs, also der Loop-Back-Struktur, Kenngrößen berechnet, die es auch für jeden der beiden Wandler gibt. Statische Kenngrößen sind z. B. integrale Nichtlinearität (INL) und differentielle Nichtlinearität (DNL) [1]. Zur Verbesserung der Fehlerdiagnose können weitere Kenngrößen wie Monotonie oder fehlende Codes herangezogen werden. Alle Kenngrößen einer Folge aus Digitalwerten werden zu einer Signatur zusammengefasst, die diese Folge charakterisiert.

Im Unterschied zu den aus der Literatur bekannten Verfahren liegt der Schwerpunkt bei unserem Vorgehen auf der Fehlerdiagnose [2], [3]. Aus Stimulus und Ausgabe der Loop-Back-Schaltung soll nicht nur erkannt werden, ob ADC und DAC innerhalb einer vorgegebenen Genauigkeit korrekt funktionieren, sondern im Fehlerfall auch diagnostiziert werden, welcher der beiden Wandler fehlerbehaftet und wo der Fehler lokalisiert ist.

Dafür wird die fehlerfreie Loop-Back-Schaltung simuliert und aus den Simulationsergebnissen die Kenngrößen für die Signatur berechnet. Anschließend wird ein Fehler ausgewählt und in die Schaltung eingefügt, d. h., die Schaltung entsprechend diesem Fehler verändert. Die mit dieser Schaltung ausgeführte Simulation liefert eine Fehlersignatur. Die Signatur der fehlerfreien Schaltung und die Fehlersignaturen aller Fehler werden vor der eigentlichen Fehlerdiagnose berechnet und dann bei der Fehlerdiagnose mit der Signatur der konkreten zu testenden Schaltung verglichen.

Bei der Fehlerdiagnose wird die zu untersuchende Schaltung mit der Treppenfunktion stimuliert und aus der Ausgabefolge die Kenngrößen und damit die „gemessene“ Fehlersignatur berechnet. Diese „gemessene“ Fehlersignatur wird mit den berechneten Fehlersignaturen verglichen und aus dem Vergleich auf den vorliegenden Fehler geschlossen.

Um einen möglichst großen Hardware-Bezug zu erreichen, wurden die Wandler auf der elektrischen Netzwerkebene modelliert. Als Fehler wurden Kurzschlüsse und Unterbrechungen sowie parametrische Veränderungen der Widerstände und Kapazitäten betrachtet.

Da Simulationen auf elektrischer Netzwerkebene sehr zeitaufwändig sind, wurde ein hierarchisches Vorgehen gewählt. Dabei werden ADC und DAC unabhängig voneinander simuliert und die Simulationsergebnisse jeweils in Kennlinien umgewandelt. Die Simulation der Loop-Back-Zusammenschaltung erfolgt unter Verwendung der Kennlinien auf der Verhaltens-ebene z. B. mit MATLAB.

Bei der Durchführung der Fehlerdiagnose hat sich herausgestellt, dass sich viele Fehlersignaturen nur geringfügig unterscheiden, so dass viele Fehler als Fehlerkandidaten für eine gemessene Fehlersignatur in Frage kommen. Ein Abstandsmaß ist schwer zu definieren. Dabei stellt sich die Frage, wie genau die Kennlinien sind, denn diese werden von Parametern, z. B. Kennwerten von Bauelementen wie Widerständen und Transistoren oder von Zellen wie Operationsverstärker, beeinflusst. Infolge von kleinen Veränderungen im Produktionsprozess können diese Parameter variieren. Von ihnen seien jeweils Nominalwert und Varianz bekannt. Die Parametervariationen werden in Monte-Carlo-Simulationen nachgebildet und so Kennlinien erzeugt, die diese Parametervariationen berücksichtigen. Damit sind eine Quantifizierung der Fehlerabstände und eine Verbesserung der Fehlerdiagnose möglich.

Durch die Monte-Carlo-Simulationen erhöht sich der Simulationsaufwand drastisch. Um die damit gleichfalls wachsende Datenmenge zu reduzieren und für die Fehlerdiagnose anwendbar zu machen, wird neben den Nominalwerten nur die Datenverteilung in Form von Histogrammen sowie Min-Max-Intervallen verwendet.

Die Kenngrößen und die Min-Max-Intervalle einer Fehlersignatur bilden die Grundlage für die Fehlerdiagnose. Werden n Kenngrößen als Achsen eines

kartesischen Koordinatensystems betrachtet, so liegen für jeden Fehler alle möglichen Kombinationen der n Kenngrößen innerhalb eines n -dimensionalen Quaders, der durch die Min-Max-Intervalle begrenzt wird. Zur Veranschaulichung werden die beiden Kenngrößen „Gain Error“ und „Offset Error“ betrachtet, wie in Abbildung 2 illustriert. Für den fehlerfreien Fall F0 und sechs mögliche Fehler F1, ..., F6 wurden per Monte-Carlo-Simulation die Intervalle für Offset und Gain Error bestimmt, in denen diese Kenngrößen auftreten. Diese spannen unterschiedlich große Flächen auf. Aufgabe der Fehlerdiagnose ist es, einer gemessenen Fehlersignatur T_i Fehlerkandidaten zuzuordnen. Abbildung 2 zeigt, dass sich diese Aufgabe grafisch lösen lässt: Liegt die Fehlersignatur T_i innerhalb der aufgespannten Fläche bzw. im aufgespannten n -dimensionalen Quader eines Fehlers, so kommt dieser Fehler als Kandidat für die vorliegende Fehlersignatur in Frage. Gibt es nur einen Kandidaten, so ist die Fehlerdiagnose damit beendet. Im vorliegenden Beispiel gemäß Abbildung 2 wird bei den Fehlersignaturen T1 und T2 kein Fehler (F0); bei T3 Fehler F3; bei T4 F5 und bei T5 F4 klassifiziert. Gibt es mehrere Kandidaten, so wird zunächst der Fehlerort (Fehlerlokalisation) bestimmt.

Sind die Fehler F1 und F5 Fehler am gleichen Fehlerort mit unterschiedlichen Fehlerwerten, so ist der Fehlerort gefunden. Im Schritt der Fehleridentifikation muss der Fehlerwert, d. h. die Abweichung von einem Nominalwert, bestimmt werden.

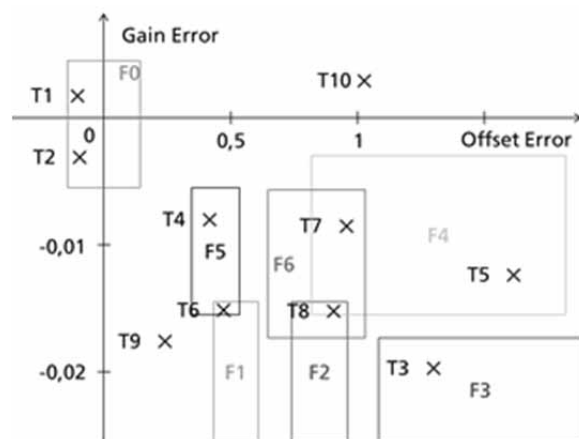


Abbildung 2. Fehlerintervalle F_i der Kenngrößen „Gain Error“ und „Offset Error“ für verschiedene Fehler

Haben die Fehlerkandidaten unterschiedliche Fehlerorte, so können weitere Kenngrößen in Simulation

und Messung einbezogen werden, um so die Fehlerkandidaten voneinander zu unterscheiden.

Unabhängig davon ist eine Bewertung der Kandidaten unter Verwendung der berechneten Verteilungen möglich. Dabei werden Fehler gleichen Fehlerorts zusammengefasst.

In Abbildung 2 gibt es für die Fehlersignaturen T6 und T7 jeweils zwei, F1 und F5 bzw. F4 und F6, sowie für die Fehlersignatur T8 drei Kandidaten: F2, F4 und F6.

Falls einer gemessenen Fehlersignatur kein Fehler zugeordnet werden kann, gehört diese Signatur nicht zu den in den Simulationen untersuchten Fehlern. In Abbildung 2 gilt das für T9 und T10.

3 Workflow zur Erzeugung der Fehlersignaturen

Wie in Abschnitt 2 erläutert, erfordert die Detektion von Fehlern in den ADC/DAC-Komponenten eines konkreten gefertigten Mixed-Signal-Schaltkreises sogenannte Fehlersignaturen, die im Rahmen eines Loop-Back-Tests zu berechnen sind.

Der in diesem Abschnitt vorgestellte Workflow realisiert die algorithmische Umsetzung des Loop-Back-Tests durch Simulatoren und Skripte und liefert im Ergebnis die für die Fehlerdiagnose benötigten Fehlersignaturen. Für berechnungs- und datenintensiven Schritte werden dabei Cloud-Technologien eingesetzt.

Für die Arbeiten wurde zunächst der im Fraunhofer IIS/EAS entwickelte analoge Fehlersimulator *aFSIM* [4] in Verbindung mit dem Netzwerksimulator *ngSPICE* sowie *Octave* als Rahmenprogramm und Simulationsprogramm auf Verhaltensebene verwendet.

Der analoge Fehlersimulator *aFSIM* ist ein Programm zur analogen Fehlersimulation auf elektrischer Netzwerkebene, mit dem auf der Grundlage von symbolischen Fehlerbeschreibungen in Form von Pattern eine Fehlerliste erzeugt, die Fehler der Fehlerliste in die Schaltung injiziert, die Fehlersimulationen angestoßen und die Ergebnisse der Fehlersimulation ausgewertet werden können. Für unsere Zwecke werden nur die Funktionen Fehlerlistengenerierung und Fehlerinjektion verwendet. Die Simulationsaufrufe erfolgen vom *GridWorker* und die Auswertung der Simulationsergebnisse auf Verhaltensebene mit Hilfe von *Octave*, einer Open-Source-Alternative zu *MATLAB*.

Abbildung 3 zeigt den Workflow zur Erzeugung von erweiterten Fehlersignaturen für die Fehlerdiagnose. Um die große Anzahl der Dateien zu verdeutlichen, wird deren Anzahl in Abbildung 3 durch die Variablen x , y und m ausgedrückt. Dabei ist x die Anzahl der Fehler im ADC, y die der Fehler im DAC und m die Anzahl der Monte-Carlo-Simulationen. Für die 5-Bit-Varianten gilt: $(x, y, m) = (9137, 395, 1000)$.

Die Kennlinien für den ADC und für den DAC werden unabhängig voneinander berechnet. Ausgangspunkt ist jeweils eine elektrische Netzliste und eine symbolische Beschreibung der Fehler. Im **Schritt 1** wird daraus jeweils eine Fehlerliste erzeugt. Mit Hilfe eines Skripts werden die Fehler separiert, und es entsteht für jeden Fehler eine Einzelfehlerliste. Im folgenden **Schritt 2** wird für jeden Fehler aus elektrischer Netzliste und Einzelfehlerliste eine modifizierte Netzliste generiert. Die in dieser modifizierten Netzliste enthaltenen Widerstände und Kapazitäten werden per Skript einer Parametrisierung und damit der Monte-Carlo-Simulation zugänglich gemacht. Mit Hilfe des *GridWorkers* wird in **Schritt 3** jede der modifizierten Netzlisten mit *ngSPICE* einer Monte-Carlo-Simulation unterworfen. Die Dateien mit den SPICE-Simulationsergebnissen enthalten sehr viele Daten, während die Kennlinien selbst nur aus der Zuordnung analoger Eingangswert – digitaler Ausgangswert beim ADC und digitaler Eingangswert – analoger Ausgangswert beim DAC bestehen. In **Schritt 4** extrahiert deshalb ein Octave-Programm aus den SPICE-Simulationsergebnissen die Kennlinien.

Im Ergebnis erhält man für den fehlerfreien und jeden fehlerhaften Wandler ein Kennlinienbündel, bestehend aus der Nominalkennlinie und den Kennlinien der Monte-Carlo-Simulation. Die ersten vier Schritte beziehen sich überwiegend auf die elektrische Netzwerkebene und sind deshalb sehr zeitintensiv.

Sie liefern die Daten für **Schritt 5**, die Loop-Back-Simulation. Dabei kommt es darauf an, einen DAC und einen ADC, jeweils repräsentiert durch ein Kennlinienbündel, in der Loop-Back-Struktur zu simulieren. Aus Effektivitätsgründen wird die Simulation auf der Verhaltensebene durchgeführt. Die Kennlinien wurden schon entsprechend konvertiert.

Zunächst werden die Kenngrößen für den fehlerfreien Fall berechnet. Deren Nominalwerte ergeben sich bei Simulation der Nominalkennlinien der fehlerfreien Wandler. Durch Kombination von je einer Monte-

Carlo-Kennlinie des einen mit einer Monte-Carlo-Kennlinie des anderen Wandlers erhält man eine entsprechende große Anzahl von variierten Werten für jede Kenngröße. Die große Anzahl von Werten wird in **Schritt 6** dadurch reduziert, dass nur die Nominalwerte der Kenngrößen aufgehoben und alle anderen in 11 Bins eingeordnet werden. So entsteht ein Histogramm für jede Kenngröße.

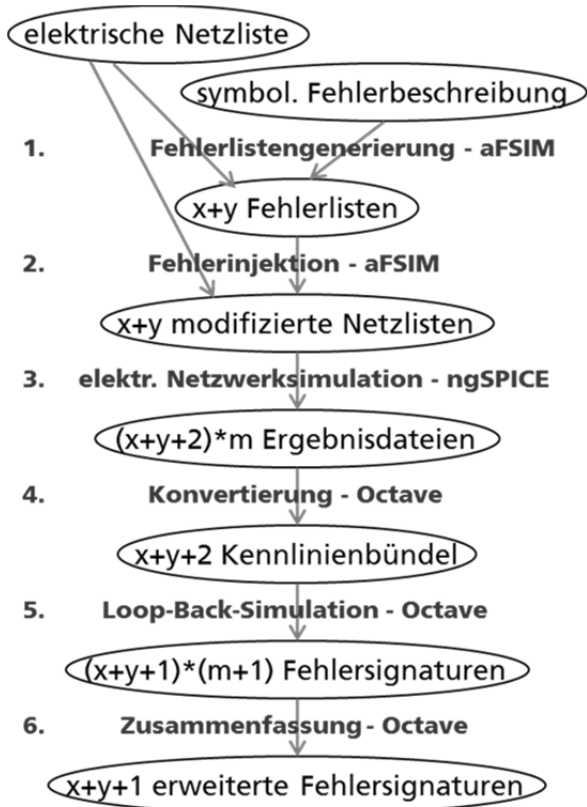


Abbildung 3. Workflow zur Erzeugung von erweiterten Fehlersignaturen. x Anzahl der ADC-, y Anzahl der DAC-Fehler, m Anzahl der Monte-Carlo-Simulationen

Die Fehlerdiagnose geht von einem Einzelfehlermodell aus. Das bedeutet, dass nur einer der beiden Wandler fehlerbehaftet und nur einer der Fehler wirksam ist. Bei der Fehlersimulation in der Loop-Back-Struktur muss deshalb jeder fehlerhafte Wandler mit dem fehlerfreien anderen Wandler kombiniert werden. Zur Berechnung der Nominalwerte der Kenngrößen eines Fehlers werden die Nominalkennlinien ausgewählt und in der Loop-Back-Struktur simuliert. Danach wird jede Monte-Carlo-Kennlinie des Fehlers mit einer zufällig ausgewählten Monte-Carlo-Kennlinie des fehlerfreien anderen Wandlers kombiniert. In Schritt 6 werden wie im fehlerfreien Fall die

berechneten Kenngrößen der Monte-Carlo-Kennlinien in jeweils 11 Bins zu Histogrammen zusammengefasst. Auf diese Weise erhält man eine erweiterte Signatur für die fehlerfreien Wandler und erweiterte Fehlersignaturen für jeden Fehler des ADCs und jeden Fehler des DACs.

Abschließend erfolgt die in Abschnitt 2 beschriebene eigentliche Fehlerdiagnose, d. h., der Vergleich einer gemessenen Fehlersignatur mit den vorausberechneten Fehlersignaturen. Im beschriebenen Ansatz wird dieser Schritt lokal mit der Rechentechnik des Nutzers realisiert, ist also nicht an die Verfügbarkeit einer Cloud-Infrastruktur gebunden.

4 Adaption *GridWorker*

GridWorker ist ein am Fraunhofer IIS/EAS entwickeltes Framework [5], das auf Basis des Map/Reduce-Paradigmas [6] parallel ausführbare Bearbeitungseinheiten (Map-Phase) über angepasste Adapter auf Multi-Core-, Cluster-, Grid- und Cloud-Infrastrukturen verteilen und nach Beendigung die Ergebnisse wieder zusammenführen kann (Reduce-Phase). Ziel ist es dabei, die wachsenden Computere Ressourcen insbesondere im Grid- und Cloud-Umfeld für Endanwender im Systementwurf zu erschließen. Diese haben in der Regel keine Möglichkeit, die vielfältigen Low-Level-Middleware-Schnittstellen wie gLite, UNICORE, Globus Toolkit, Grid Engine, PBS (Portable Batch System), OpenStack, OCCI (Open Cloud Computing Interface) etc. direkt aus ihren Applikationen heraus anzusprechen. *GridWorker* schließt die Lücke zwischen diesen Middleware-Diensten und den Entwurfsprogrammen und ermöglicht dem Systementwerfer das Konfigurieren von Simulations-, Optimierungs- oder Analyseexperimenten und deren Ausführung auf jeweils verfügbaren Backend-Ressourcen, wie sie beispielsweise beim Cloud-Computing von Providern zur Verfügung gestellt werden.

Bei der Fehlerdiagnose besteht die Aufgabe, für alle Fehler jeweils eine Monte-Carlo-Simulation durchzuführen. Da die Einzelsimulationen untereinander entkoppelt und damit unabhängig voneinander ausgeführt werden können, kann der Gesamtzeitbedarf drastisch reduziert werden, wenn Cluster-, Grid- oder Cloud-Ressourcen mit Hilfe von *GridWorker* eingesetzt und ein maximal möglicher Parallelisierungsgrad ausgenutzt werden.

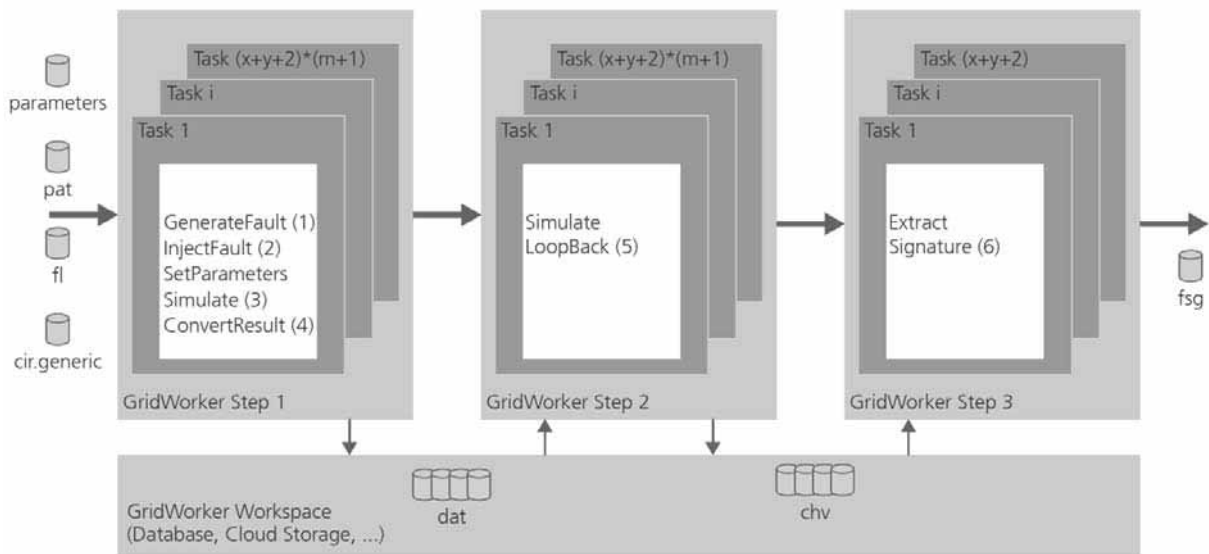


Abbildung 4. Gesamtablauf des Workflows aus Gridworker-Perspektive

Die Realisierung der Fehlerdiagnose mit *GridWorker* erfolgte in fünf Schritten:

1. Der gesamte Fehlerdiagnose-Workflow wurde in drei Phasen (Step 1 - 3) zerlegt, da es im gesamten Ablauf zwei Synchronisationspunkte gibt: Die Loop-Back-Simulationen (5) können erst ausgeführt werden, wenn die Ergebnisse aller Schaltungssimulationen vorliegen. Und die Fehlersignaturen können erst nach Vorliegen der einzelnen Signaturen zusammengefasst werden (6).
2. Alle pro Einzelsimulation erforderlichen Programmaufrufe wurden in einem Shellskript *map.sh* zusammengefasst. *GridWorker* ruft dieses Skript während der Map-Phase für einen konkreten Fehlerfall mit den jeweils durch den Monte-Carlo-Algorithmus bestimmten Parameterwerten auf, führt die Simulation durch und generiert als Ergebnis eine Kennliniendatei (Step 1).
3. Auch die für die Loop-Back-Simulation notwendigen Programmaufrufe (*MATLAB* oder *Octave*) wurden in einem Shellskript *map.sh* beschrieben und *GridWorker* für den Step 2 zur Verfügung gestellt. Analog wurde für Step 3 vorgegangen.
4. Da zwischen den einzelnen Phasen sehr viele Dateien (Tausend bis Millionen) übergeben werden müssen, wurde *GridWorker* um das Konzept der zentralisierten Speicherung erweitert. Dabei kann der Anwender bei Bedarf zwei Referenzen **WORKSPACE** und **STORAGE** konfigurieren und jeder *GridWorker*-Task die Möglichkeit geben, über diese Referenzen Daten zentralisiert abzule-

gen. Das Konzept erlaubt es, über die Standard-URL-Syntax Pfade in Dateisystemen, Datenbanken oder Referenzen innerhalb von Cloud ObjectStorages zu spezifizieren.

5. Schließlich wurden konkrete Anwendungsbeispiele für die Verwendung mit *GridWorker* aufbereitet. Dazu wird jeweils das Simulationsmodell für den fehlerfreien Fall und die zu untersuchenden Fehlermodelle in einem Verzeichnis *inputs/* bereitgestellt. Außerdem müssen die bei der Monte-Carlo-Simulation zu variierenden Parameter in einer Datei *parameters* spezifiziert werden. Die *GridWorker*-bezogenen Konfigurationen (Cloud-Adapter etc.) werden in einer Datei *configuration* zusammengefasst.

5 Anwendungsbeispiele und Experimente

Der Nachweis der Wirksamkeit des Workflows wurde zunächst anhand der 3-Bit-Varianten von Flash-ADC und R2R-DAC geführt (Blockschaltbilder s. Abbildung 5).

Ausgangspunkt für die Fehlerdiagnose ist die elektrische Netzliste, die den Aufbau jedes Wandlers aus Widerständen, Kondensatoren und Transistoren beschreibt. Die Netzliste des 3-Bit-Flash-ADCs enthält 85 Widerstände, sieben Kondensatoren und 161 Transistoren, die des 3-Bit-R2R-DACs 17 Widerstände, einen Kondensator und 21 Transistoren. Die symbolische Fehlerbeschreibung gibt an, wie sich diese Bauelemente im Fehlerfall verändern können. Bei den

Widerständen wird deren Wert von Kurzschluss (sehr kleiner Wert) bis Unterbrechung (sehr großer Wert) variiert, wobei hier mit insgesamt acht verschiedenen Widerstandswerten gearbeitet wird. Das betrifft insbesondere die funktionsbestimmenden Widerstände, beim Flash-ADC die der Widerstandskette und beim R2R-DAC die des R2R-Netzwerkes. Bei den Kondensatoren und bei den Transistoren werden nur Kurzschluss- und Unterbrechungsfehler betrachtet.

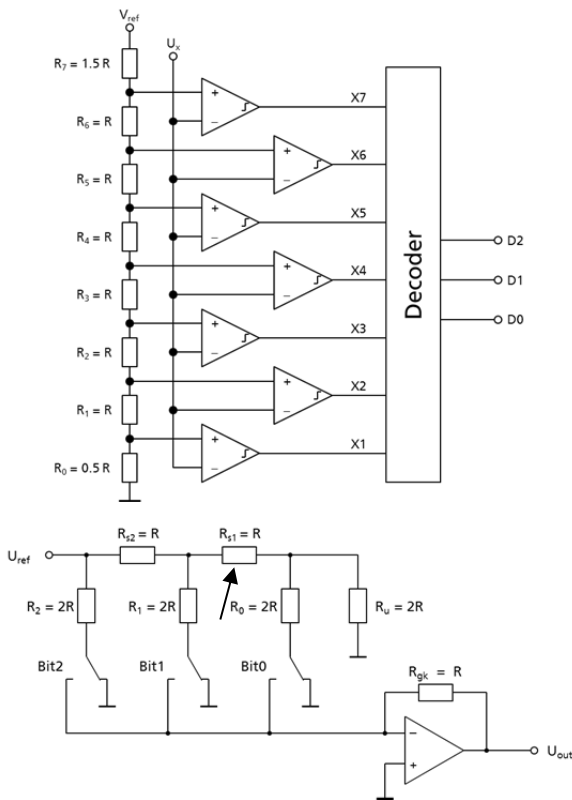


Abbildung 5. 3-Bit-Flash-ADC (oben) und 3-Bit-R2R-DAC (unten)

Bei der Fehlerlistengenerierung werden die Fehlerdefinitionen der symbolischen Fehlerbeschreibung auf die elektrische Netzliste angewendet und auf diese Weise eine Fehlerliste erzeugt, auf die sich dann die Fehlerdiagnose bezieht. Da der Flash-ADC für jede Stelle im Thermometer-Code einen Widerstand und einen Komparator benötigt, nimmt die Anzahl der Fehler exponentiell mit der Stellenzahl zu. So hat der 3-Bit-Flash-ADC 1.197 Fehler, die 5-Bit-Variante hingegen 9.137 Fehler. Demgegenüber nimmt die Anzahl der Fehler beim R2R-DAC nur linear zu: 223 Fehlern bei der 3-Bit-Variante stehen 395 Fehler bei der 5-Bit-Variante gegenüber.

Da alle Fehler der Fehlerliste in gleicher Weise behandelt werden, bietet sich eine Parallelisierung der Bearbeitung mit Cloud Computing geradezu an. Nach der Bildung von Einzelfehlerlisten werden modifizierte Netzlisten gebildet, in die einheitlich Parameter für die Monte-Carlo-Simulation eingefügt werden. Die Simulation mit ngSPICE und die Konvertierung in Kennlinienbündel werden in der Cloud durchgeführt.

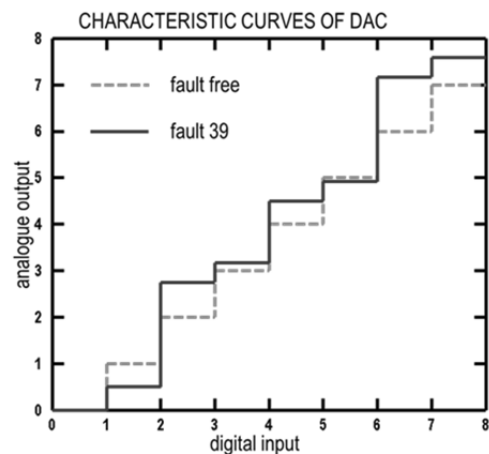


Abbildung 6. Kennlinie des DACs im fehlerfreien Fall (gestrichelt) und bei Fehler 39 (volle Linie)

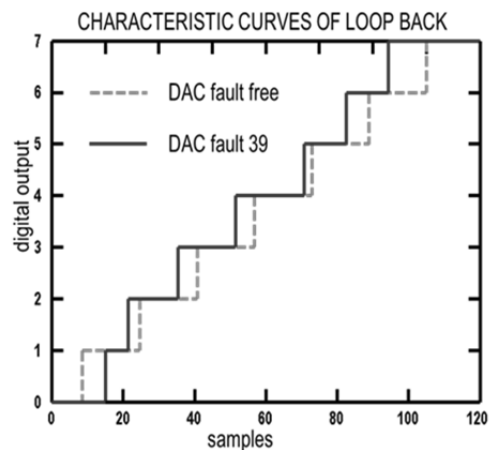


Abbildung 7. Ergebnis der Loop-Back-Simulation für DAC-Fehler 39

Für die Loop-Back-Simulation müssen alle Kennlinienbündel berechnet sein. Da die Loop-Back-Simulation besonders rechenintensiv ist, wird sie in der Cloud durchgeführt. Dabei können die Schritte 5 und 6 aus Abbildung 3 für jeden Fehler unmittelbar hintereinander ausgeführt werden. Die Abbildungen 6 und 7 sowie die Tabellen 1-3 veranschaulichen die

entstehenden Kennlinien, Kenngrößen und Fehlersignaturen.

In Abbildung 6 werden die mit ngSPICE simulierten Kennlinien für den fehlerfreien Fall und DAC-Fehler 39 gegenübergestellt. Bei Fehler 39 erhöht sich der in Abbildung 5 durch den Pfeil markierte Widerstand R_{s1} auf den fünffachen Wert. Abbildung 7 zeigt das Ergebnis nach der Loop-Back-Simulation.

Daraus werden Kenngrößen, z.B. die in Tabelle 1 gezeigten berechnet. Diese Kenngrößen werden bei jeder Monte-Carlo-Simulation ermittelt. Zur Datenreduktion werden nur die Kenngrößen der Nominalparameter in der Fehlersignatur gespeichert. Alle anderen berechneten Kenngrößen werden zu Histogrammen zusammengefasst. Die Fehlersignatur für DAC-Fehler 39 wurde aus Platzgründen auf die Tabellen 2 und 3 aufgeteilt.

fnr	Δ gain	off-	tue	dnl+	dnl-	inl+	inl-
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
39	0.05	0.03	0.69	0.67	-0.58	0.67	-0.61

Tabelle 1. Kenngrößen nach der Loop-Back-Simulation für den fehlerfreien Fall und für DAC-Fehler 39

fnr	char value	nom	min	max
39	Δ gain	0.05	-0.05	0.04
	offset	0.03	-0.25	0.20
	tue	0.69	-0.36	0.90
	dnlmax	0.67	-0.50	0.20
	dnlmin	-0.58	-0.01	0.12
	inlmax	0.67	-0.20	0.51
	inlmin	-0.61	-0.55	0.91

Tabelle 2. Fehlersignatur für den DAC-Fehler 39 Teil 1, bestehend aus Nominal-, Maximal- und Minimalwert

fnr	char value	bin width	bins															
39	Δ gain	0.01	1	2	5	11	18	25	19	12	6	2	0					
	offset	0.05	1	3	7	14	20	22	17	10	5	1	0					
	tue	0.18	0	0	0	1	8	61	15	5	4	4	2					
	dnlmax	0.10	5	13	17	18	16	26	4	1	0	0	0					
	dnlmin	0.02	0	0	0	0	0	74	0	0	18	0	8					
	inlmax	0.10	0	0	0	6	20	27	31	12	2	1	1					
	inlmin	0.19	0	0	1	3	10	42	9	11	13	9	2					

Tabelle 3. Fehlersignatur für den DAC-Fehler 39 Teil 2, bestehend aus den Histogrammen

Zur Validierung der Zeiteffektivität wurden die Rechnungen zunächst auf einem Cluster mit einem, fünf, 10 und 20 parallelen Jobs durchgeführt und dabei Zeitmessungen gemacht. Insgesamt wurden neben dem fehlerfreien Fall 1.197 Fehler im ADC und 223 Fehler im DAC berücksichtigt. Für jeden Fehler wurden 10 Monte-Carlo-Simulationen durchgeführt. Zusammen mit den Simulationen der Nominalwerte ergaben sich damit 15.642 Einzelsimulationen. In Abbildung 8 sind die Bearbeitungszeiten für verschiedene Zahlen von parallelen Jobs dargestellt. Für Step 1 und Step 2 ist die Trennung von Map- und Reduce-Phase markiert. Die Map-Phase (jeweils unterer Balkenabschnitt) ist nahezu ideal parallelisierbar. Bei der Reduce-Phase (jeweils oberer Balkenabschnitt) ist keine Parallelisierung möglich.

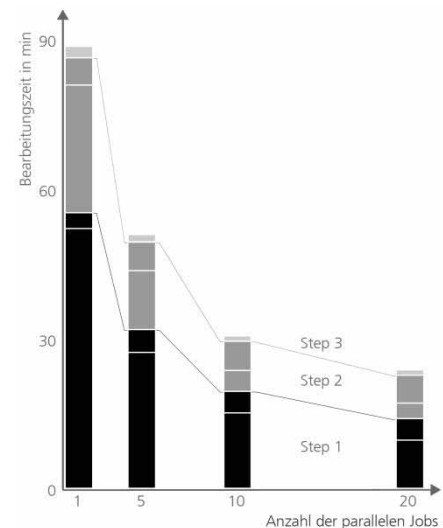


Abbildung 8. Dauer der Berechnungen für die verschiedenen Schritte: Step 1 (schwarz), Step 2 (grau) und Step 3 (hellgrau).

6 Zusammenfassung und Ausblick

Der im Beitrag vorgestellte Ansatz nutzt Cloud-Technologien für daten- und berechnungsintensive Aufgaben bei Entwurf und Test integrierter elektronischer Systeme. Für die Fehlerdiagnose von Mixed-Signal-Komponenten DAC und ADC konnte gezeigt werden, dass signifikante Verbesserungen der Diagnoseeffizienz in Form von Zeitgewinn, höherer Auflösung der Fehler (Netzwerk- anstelle Verhaltensebene) und Skalierbarkeit auf größere Verarbeitungsbreite der Komponenten erreichbar sind. Der Workflow zur Fehlerdiagnose basiert auf dem Ansatz, DAC und ADC in einer Loop-Back-Struktur zu analysieren und partitioniert die Diagnoseaufgabe in Teilschritte, die

für eine Bearbeitung in der Cloud prädestiniert sind. Für die Organisation dieser Bearbeitung wurde das bei Fraunhofer entwickelte Werkzeug *GridWorker* weiterentwickelt. Als Software-Werkzeuge für die Bearbeitung der algorithmischen Teilaufgaben des eigentlichen Diagnoseprozesses kommen Public-Domain-Simulatoren und der bei Fraunhofer entwickelte analoge Fehlersimulator *aFSIM* zum Einsatz.

Ziel weiterführender Arbeiten ist es einerseits, die Skalierbarkeit unseres Ansatzes weiter zu verbessern und Voraussagen zur Laufzeit zu ermöglichen. Andererseits wird an einem nutzerfreundlichen Zugang gearbeitet, um bei Fraunhofer verfügbare Simulations-, Test- und Diagnoseverfahren (bzw. die darauf aufbauenden Werkzeuge) auch in der Cloud als Entwurfsdienstleistungen für externe Nutzer anbieten zu können.

Diese Arbeit wurde im Rahmen der BMBF-Projekte DIANA - Durchgängige Diagnosefähigkeit für Elektroniksysteme im Automobil (Kennzeichen 01M3188B) und OptiNum-Grid - Optimierung technischer Systeme und naturwissenschaftlicher Modelle mit Hilfe numerischer Simulationen im Grid (Kennzeichen 01IG0901D) gefördert.

7 Literatur

- [1] *ADC and DAC Glossary*, MAXIM Application notes, Jul 22, 2012.
<http://www.maximintegrated.com/appnotes/index.mvp/id/641>
- [2] M. Gulbins, W. Vermeiren und S. Redlich. *Fehlerdiagnose für AD- und DA-Wandler in einer Loop-Back-Struktur*. Dresdner Arbeitstagung Schaltungs- und Systementwurf (DASS 2012), Dresden, 3./4. Mai 2012
- [3] M. Gulbins, W. Vermeiren und S. Redlich. *Fehlerdiagnose für ADCs und DACs in einer Loop-Back-Struktur unter Einbeziehung von Parametervariationen*. Testmethoden und Zuverlässigkeit von Schaltungen und Systemen 25. GI/GMM/ITG-Workshop Dresden, 24. - 26. Februar 2013
- [4] B. Straube, B. Müller, W. Vermeiren, C. Hoffmann und S. Sattler. *Analogue fault simulation by aFSIM*. Design, Automation and Test in Europe Conference and Exhibition, DATE 2000 – User Forum, Paris, March 27-30, 2000, pp. 205-210.
- [5] A. Schneider. *Variantensimulation mit GridWorker*. ASIM-Workshop „Simulation technischer Systeme und Grundlagen und Methoden in Modellbildung und Simulation“, Krefeld, 24.-25. Februar 2011
- [6] J. Dean and S. Ghemawat. *MapReduce: Simplified Data Processing on Large Clusters*. Sixth Symposium on Operating Systems Design and Implementation (OSDI '04), San Francisco, California, USA, December 6-8, 2004
- [7] A. Schneider. *Automatisierte Ressourcenbedarfsschätzung für Simulationsexperimente in der Cloud*. Grid4Sys-Workshop „Grid-, Cloud- und Big-Data-Technologien für Systementwurf und -analyse“, Dresden, 27.-28. November 2013

Gekoppelte vs. integrierte Simulation der Steuerungs-, Regelungstechnik und Strukturmechanik mechatronischer Systeme am Beispiel von Werkzeugmaschinen

Gerhard Kehl¹, Peter Wagner²

¹Hochschule Esslingen

²Gebr. Heller Maschinenfabrik GmbH, Nürtingen

gerhard.kehl@hs-esslingen.de

Zusammenfassung

Komplexe mechatronische Systeme wie Werkzeugmaschinen zeichnen sich durch die enge Verzahnung physikalischer Domänen (elektrische und mechanische Hardware) mit logischen Domänen (Informations- und signalverarbeitende Prozesse) aus. Ein neuer Simulationsansatz, die ANSYS-Control-Toolbox, führt die notwendigerweise eng gekoppelte physikalisch-mechanische Ebene und die Signalebene mechatronischer Systeme innerhalb des FE-Paketes ANSYS zusammen und erweist sich als geeigneter Ansatz für die Entwicklungsprozesskette mittelständisch geprägter Werkzeugmaschinenhersteller mit begrenzten Ressourcen für Modellierung und Simulation.

1 Einleitung

Bislang wird bei der Entwicklung mechatronischer Systeme (und insbesondere in traditionell geprägten Branchen wie der Werkzeugmaschinenindustrie) üblicherweise in unterschiedlichen, domänenspezifischen Simulationswerkzeugen gearbeitet. Während die Simulation des mechanischen Teils der Werkzeugmaschine von FE-Paketen dominiert wird, konnten sich für die Abbildung der logischen Ebene vorwiegend Systemsimulatoren wie Matlab-Simulink® oder Caspoc® etc. etablieren.

Der Wunsch nach einer zeitlich parallelisierten, d.h. gleichzeitigen Simulation beider Bereiche unter Berücksichtigung der gegenseitigen Wechselwirkungen macht einen Brückenschlag zwischen diesen Modellierungsansätzen notwendig [1].

In der Vergangenheit konnten sich hierfür bei der Entwicklung von Werkzeugmaschinen im Wesentlichen zwei Vorgehensweisen etablieren: Ein gängiger Ansatz bildet die mechanische Struktur über ein linearisiertes Ersatzmodell (z.B. über eine Transferfunktion oder ein modales Ersatzmodell) ab und bindet dies als diskreten Baustein in einen Systemsimulator ein. Ein hierzu alternativer Ansatz verfolgt die Co-Simulation zwischen FE-Paket und Systemsimulator. Hier werden zwei quasi-parallel ablaufende Berechnungsprozesse im Zeitbereich durchgeführt

und in jedem Zeitschritt durch einen Informationsaustausch über eine geeignete Schnittstelle gekoppelt [2].

2 Neuer Ansatz: Integrierte Simulation

Bei einem neuartigen Ansatz soll die physikalisch-mechanische als auch die logische Domäne komplett im FE-Paket ANSYS modelliert und simuliert werden können, da dies in vielen Fällen am einfachsten durch die Entwicklungsprozesskette mittelständisch geprägter Werkzeugmaschinenhersteller abbildbar ist.

Die Grundidee dieses Ansatzes besteht in der Abbildung von regelungstechnischen Elementen über quasi-mechanische Analogien, die es ermöglichen, Bausteine zur Signalverarbeitung direkt in einem ANSYS-Strukturmodell zu integrieren. Damit können Strukturverhalten, Regelungstechnik und deren Kopplung gleichzeitig innerhalb einer einheitlichen Simulationssumgebung beschrieben werden.

Die eigentliche Modellierung der Reglerbausteine erfolgt durch die Formulierung spezieller Dämpfungs- und Steifigkeitsmatrizen mittels des von ANSYS bereitgestellten MATRIX27-Elements. Die Matrixeinträge werden dabei so gewählt, dass die daraus resultierenden Bewegungsdifferentialgleichungen direkte Analogien zu den beschreibenden Differentialgleichungen des jeweiligen Signalgliedes darstellen.

Dabei fungieren die translatorischen Freiheitsgrade der beteiligten FE-Knoten als Signalgrößen, d.h. die übliche geometrische Interpretation eines Freiheitsgrades als Verschiebung wird nun durch eine Interpretation als Signallevel ersetzt. Basierend auf diesem Prinzip ist mit der ANSYS-Control-Toolbox eine Sammlung von APDL-Makros entstanden, mit denen die gebräuchlichen Bausteine zur Signalverarbeitung einfach in ein Modell integriert werden können. Die Toolbox verfolgt dabei das Prinzip eines „Baukastensystems“, bei dem der Anwender komplexe Regelkreise mit beliebiger Topologie aus elementaren Bausteinen aufbauen kann.

Die Makros übernehmen dabei die direkte Umwandlung von regelungstechnischen Parametern (wie Verstärkungsfaktor, Nachstellzeit, etc.) in korrespondierende Einträge auf Matrix-Ebene.

Die Bausteinbibliothek umfasst momentan die gängigen, linearzeitinvarianten Reglerbausteine (P-, I-, D-, PT1-, PT2-Glieder, und deren Kombinationen), Glieder zur Verschaltung der Bausteine durch Verbinder und Knoten-/Summationselemente sowie verschiedene Signalgeneratoren. Darüber hinaus werden spezielle Sensoren und Aktoren bereitgestellt, die als Schnittstelle zur Anbindung des Regelkreises an das mechanische Strukturmodell dienen. Sensor-Bausteine übertragen dabei einen der sechs mechanischen Freiheitsgrade eines Strukturknotens als Signallevel in den Regelkreis. Aktoren wandeln den Signallevel eines Verbindungspunktes im Regelkreis in Kräfte bzw. Momente um, mit denen wiederum auf Knoten des mechanischen Modells eingewirkt wird.

Die mit der Control-Toolbox erstellten Modelle stellen aus ANSYS-Sicht reine Strukturmodelle dar, mit denen alle üblichen Simulationen sowohl im Zeit- als auch im Frequenzbereich durchgeführt werden können.

Im Folgenden sollen anhand exemplarischer Anwendungen einige Simulations-Szenarien, die mit der ANSYS-Control-Toolbox modelliert wurden, dargestellt werden. Die Simulationen stellen dabei typische Fragestellungen aus dem Bereich des Werkzeugmaschinenbaus dar und wurden bei der Gebr. Heller Maschinenfabrik GmbH, Nürtingen durchgeführt. Parallel zur ANSYS-internen Modellierung mit der Control-Toolbox wurden jeweils Verifikationen mit alternativen Ansätzen (z.B. ANSYS-Excel-Kopplung, ANSYS-Matlab-Simulink-Kopplung) durchgeführt. Vor einem breiten Einsatz in der Entwicklung neuer

Werkzeugmaschinen diente als Referenzmodell zunächst eine dreiaxsigte Werkzeugmaschine, bei der für jede Vorschubachse die in Abbildung 1 dargestellte industrietübliche P-Lage-PI-Drehzahl-Kaskadenregelung mit Geschwindigkeitsvorsteuerung integriert wurde.

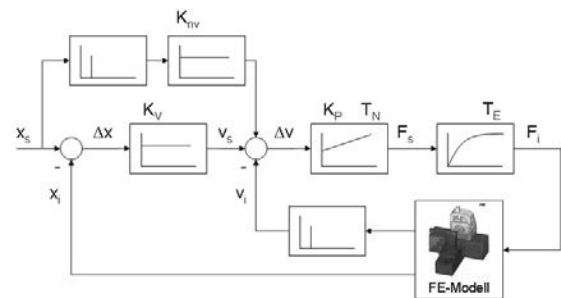


Abbildung 1. Regelungsstruktur

Auf dem mechanischen Modell werden dabei über Sensoren die Relativverschiebung zwischen einem Glasmaßstab am Maschinenbett und einem Tastkopf am Schlitten sowie die Drehzahl am Vorschubmotor gemessen und der Regelung zurückgeführt. Das Ausgangssignal der Regelung wird als Aktorkraft am Kugelgewindetrieb der Werkzeugmaschine eingebracht [3]. Abbildung 2 zeigt ein in ANSYS erstelltes Blockschaltbild für einen Teil des geregelten Systems.

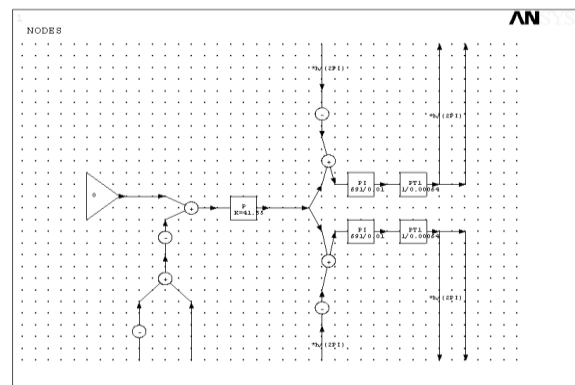


Abbildung 2. Regelung in ANSYS integriert

Hiermit sind z.B. folgende Simulationsszenarien denkbar:

2.1 Führungsverhalten im Frequenzbereich

Das Führungsverhalten beschreibt einen Frequenzgang, bei dem die Sollgröße harmonisch angeregt und der Istgröße gegenübergestellt wird. Die Darstellung erfolgt typischerweise in einem Bode-Diagramm, in

dem Amplituden- und Phasengang über der logarithmierten Frequenz dargestellt sind.

Der Drehzahlregler-Führungsfrequenzgang ermöglicht Aussagen über die Stabilität des Drehzahlregelkreises und wird typischerweise zur Auslegung der Geschwindigkeitsreglerparameter K_P und T_N herangezogen.

In analoger Weise wird auch bei der äußeren Regelschleife, dem Lageregelkreis, zur Ermittlung der Lagereglerversärkung K_V vorgegangen.

2.2 Nachgiebigkeitsfrequenzgang

Beim Nachgiebigkeitsfrequenzgang wird ein konstanter Lagesollwert vorgegeben (= Position halten) und eine zwischen Werkzeug und Werkstück wirkende Prozesskraft harmonisch angeregt. Hieraus können Aussagen über die dynamische Steifigkeit des geregelten Systems in Abhängigkeit der Anregungsfrequenz gewonnen werden.

2.3 Positionierverhalten im Zeitbereich

Der Schleppfehler spielt typischerweise bei Anfahrvorgängen eine Rolle und misst das „Nacheilen“ des Vorschubschlittens im Vergleich zu einer als Sollwert vorgegebenen Anfahrrampe. Vergleiche mit den parallel durchgeführten Simulationen in Matlab-Simulink® konnten die Ergebnisse verifizieren und den mit der ANSYS-Control-Toolbox verfolgten Ansatz bestätigen. Abbildung 3 zeigt dies beispielhaft anhand eine Lagesollwertsprungs.

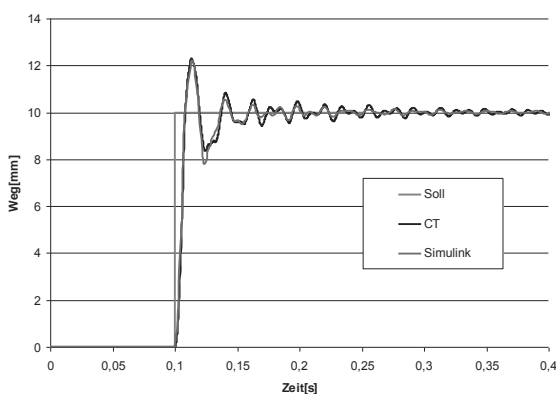


Abbildung 3. Positionierverhalten

Mit Erfahrungen in der Anwendung bei der Werkzeugmaschinenentwicklung hat sich der hohe Rechenaufwand bei komplexen Modellen als Nachteil erwiesen, der die Schritte *Modellreduktion - Berechnung - Expansion* meistens erforderlich macht. Ins-

besondere die gute 3D-Ergebnisvisualisierung am Volumenmodell der Werkzeugmaschine zeigt sich jedoch als deutlicher Vorteil gegenüber den anderen beschriebenen Simulationmethoden [3,4].

3 Fazit

Die in diesem Anwendungsfeld gewonnenen Erfahrungen haben gezeigt, dass durch die Integration der Signalverarbeitung in das Modellierungsportfolio von ANSYS zahlreiche neue Simulationsszenarien denkbar sind, mit denen wichtige Aussagen über einen konstruktiven Entwurf schon in der Entwicklungsphase gewonnen werden können. Hierbei können insbesondere die Wirkzusammenhänge zwischen physikalisch-mechanischer und logischer Domäne mit der notwendigen Sorgfalt untersucht werden. Dies ist sicherlich auch für andere Anwendungen aus dem Bereich der Mechatronik sinnvoll und verspricht Potential für zukünftige Applikationen und weiterentwickelte Softwarewerkzeuge [5].

4 Referenzen

- [1] Brandenberger, M.; Kehl, G.: *Coupled Simulation using FEM, MBS and Control Simulation Tools using the Example of a Machine Tool*. In: Proceedings of the NAFEMS Seminar Mechatronics in Structural Analysis, Wiesbaden, 5.5.-6.5.2004.
- [2] Keller, K.; Kehl, G.: *Simulation geregelter, hochdynamischer Werkzeugmaschinen am Beispiel des Bearbeitungszentrums HELLER MCH 250*. In: Proceedings des 1. Karlsruher Simulationssymposiums, Karlsruhe, 30.11.2005.
- [3] Alber, R.; Kehl, G.: *Systemsimulation in ANSYS integriert*. In: CADFEM-Infoplaner (2006) Nr.2, S.22-23.
- [4] Kehl, G.: *Meilensteine des FEM-Einsatzes im Entwicklungsprozess spanender Werkzeugmaschinen – Eine Zwischenbilanz*. In: Proceedings of the 31st CADFEM Users Meeting 2013, Mannheim, 19.6.-21.6.2013.
- [5] Rudnyi, E.; Rauch, R.: *System Level Simulation with Simplorer and ANSYS: Through Simulation to Stable Productional Process*. In: Tagungsband des IWF-Symposiums: Simulation von Werkzeugmaschinen, Zürich, 12.1.2010.

Theoretische und experimentelle Untersuchungen der Vertikaldynamik eines elektrischen Rollstuhls mit dem Ziel der Optimierung des Fahrverhaltens durch semiaktive Dämpfer

Sönke Lück¹, Rolf Naumann¹

¹ Fachhochschule Bielefeld, Am Stadtholz 24, 33609 Bielefeld

ISyM – Institut für Systemdynamik und Mechatronik

soenke.lueck@fh-bielefeld.de

Im Rahmen des vom BMWi geförderten ZIM Projekts *LoCoMS* (Low-Cost Mechatronic Systems) werden basierend auf einer mechatronischen Entwurfsmethodik Analysewerkzeuge entwickelt, die einen kostengünstiger Entwurf von mechatronischen Systemen ermöglichen. Eine Anwendung für das Projekt ist die Reduktion der Vertikalschwingungen (AVC – Active Vibration Control) an einem elektrischen Rollstuhl. Dabei sollen durch Austausch der passiven Dämpferelemente durch semiaktive Dämpfer die Humanschwingungen während der Fahrt reduziert werden. Für die experimentelle Analyse des Rollstuhls im Zeitbereich und Frequenzbereich wurde der Rollstuhl mit Beschleunigungsaufnehmern versehen. Weiterhin wurde ein Prüfstand konstruiert und realisiert, mit dem über einen Exzenter eine quasi-harmonische Anregung in Vertikalrichtung (0 Hz - 30 Hz, $\pm 5\text{mm}$) an einem Rad erfolgen kann. Für die Durchführung von realistischen und reproduzierbaren Messungen wurde ein Dummy nach DIN 45676 zur Abbildung der mechanischen Eingangsimpedanz und Übertragungsfunktion des menschlichen Körpers konstruiert. Die Bewertung der Schwingungen erfolgt nach DIN EN ISO 8041 (Humanschwingungen). Für den mechatronischen Entwurf wurde ein Mehrkörpermodell in SIMPACK aufgebaut, wobei die Ermittlung der Steifigkeiten und Dämpfungen der Räder experimentell durchgeführt wurde. In dem Beitrag werden die ersten Ergebnisse von Messungen am Rollstuhlprüfstand und den Simulationsrechnungen im Frequenzbereich vorgestellt. Das Modell wird dazu genutzt, die Parameter eines optimierten passiven Feder-Dämpfer-Elements für verschiedene Fahr- und Betriebszustände zu bestimmen. In weiteren Arbeiten werden der reale Rollstuhl und das MKS-Modell um den Einbau von steuerbaren elektro-rheologischen Dämpferelementen und den Aufbau eines mechatronischen Systems zur Verbesserung des Fahrkomforts erweitert.

1 Einleitung

An der Fachhochschule Bielefeld erfolgen in dem *Institut für Systemdynamik und Mechatronik (ISyM)* Untersuchungen von akustischen und strukturdynamischen Fragestellungen sowie Entwicklungen von mechatronischen Lösungen. Im Rahmen des vom BMWi geförderten ZIM Projekts *LoCoMS* (Low-Cost Mechatronic Systems) werden basierend auf einer mechatronischen Entwurfsmethodik Analysewerkzeuge entwickelt, die einen kostengünstiger Entwurf von mechatronischen Systemen ermöglichen.

Die Entwicklung von mechatronischen Lösungen für eine Anwendung erfordert erhebliches Know-How und Wissen auf den Gebieten der Systemanalyse, des Systementwurfs und der Realisierung von Controllern. Hinzu kommen entsprechende Kenntnisse in Sensorik und Aktuatorik. Dieses Systemwissen kann üblicherweise von KMU's nicht bereitgestellt werden.

Der Entwurf und die Realisierung eines kostengünstigen mechatronischen Systems, das für sehr viele Anwendungen mit geringer Anpassung der Hard-

ware- und Software geeignet ist, wird aufgrund der sehr unterschiedlichen Fragestellungen von technischen Systemen nicht möglich sein. Die Idee ist daher, einen mechatronischen Baukasten zu entwickeln, der kostengünstige und robuste Elemente eines mechatronischen Systems enthält, die anwendungsspezifisch miteinander kombiniert werden können.

Ein wesentlicher Kostenfaktor ist der Entwurf des mechatronischen Systems, bei dem eine möglichst optimale mechatronische Lösung erarbeitet wird. Deshalb werden für bestimmte *Anwendungsklassen* bereits vorkonfektionierte Systeme zusammengestellt, die dann noch an die spezielle Problemstellung angepasst werden müssen.

In dem Projekt *LoCoMS* (*Low-Cost Mechatronic Systems*) sollen diese Anwendungsklassen identifiziert und geeignete (günstige, einfache, robuste) Elemente für ein Low-Cost mechatronisches System spezifiziert werden. Die **Abbildung 1** zeigt die hierfür notwendigen Phasen für den Entwurf und die Realisierung an einem technischen System.

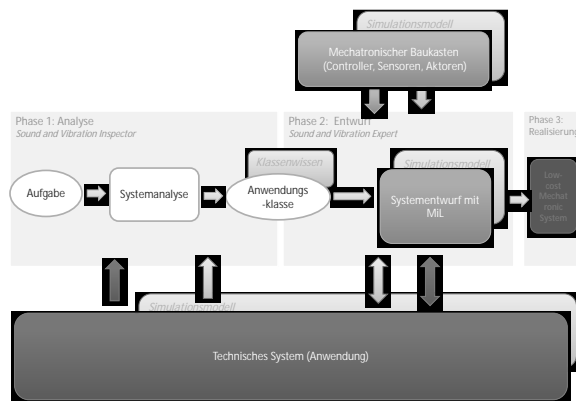


Abbildung 1: LoCoMS Projektphasen

Die Projektpartner sind die FH Bielefeld (Prof. Waßmuth, Prof. Naumann) und die Fa. CAE Software-Systems GmbH.

1.1 Anwendungs-kategorie Kabinenschall

Eine Anwendung wird im Rahmen des Projekts am Beispiel einer Fahrerkabine erarbeitet. Hier steht eine Entlastung des Fahrzeugführers im Vordergrund, der aufgrund eines Luft- oder Körperschallproblems einer erhöhten Störschallbelastung ausgesetzt ist. Am Demonstrator werden die Möglichkeiten der in dem Projekt entwickelten Analysesoftware (*Sound and Vibration Inspector*) entwickelt und genutzt. In Folge werden dann Untersuchungen bzgl. der Art und des Einsatzes von Verfahren zur Reduktion bzw. Kompensation des Störschalls an der Fahrerposition durchgeführt (*ANC - Active Noise Control*). Die hieraus gewonnenen Erkenntnisse, Modelle und Verfahren fließen anschließend in das Expertenwerkzeug (*Sound and Vibration Expert*) ein. Aufgrund des Interesses an der Thematik hat sich die Firma Claas, Harsewinkel bereit erklärt, eine entsprechende Kabine zur Verfügung zu stellen.

1.2 Anwendungs-kategorie Humanschwingungen

Eine weitere Anwendungs-kategorie ist die Schwingungsreduktion am elektrischen Rollstuhl. Dabei steht die Beurteilung des Schwingungsverhaltens hinsichtlich des Fahrkomforts (Vertikaldynamik) im Vordergrund. Über den *Sound and Vibration Inspector* sollen im ersten Schritt die notwendigen Daten aufgenommen und analysiert werden. Die Bewertung der Ergebnisse hinsichtlich des Fahrkomforts sowie die Ableitung der Randbedingungen und Eigenschaften einer zu identifizierenden Anwendungs-kategorie erfolgen dann durch den *Sound and Vibration Expert*. In einem zweiten Schritt sollen prinzipielle Lösungsmöglichkeiten zur Verbesserung des Fahrkomforts entwickelt werden. Aufgrund der Randbedingungen des elektrischen Rollstuhls (Energie, Kosten, Raumbedarf, Gewicht) werden semi-aktive Lösungen betrachtet. Die

Firma *invacare* aus Porta-Westfalica und Hersteller von elektrischen Rollstühlen stellt einen elektrischen Rollstuhl sowie den notwendigen Support für das Projekt zur Verfügung. Es handelt es sich um einen elektrischen Rollstuhl der Oberklasse vom Typ *Invacare Storm X-plore* mit einem gefederter Fahrwerk mit Einzerradaufhängung und folgenden technischen Daten:

- Gewicht $m = 175$ kg, max. Zuladung 150 kg
- max. Geschwindigkeit $v = 10$ km/h,
- Reichweite 42 km (Batterieleistung 70 Ah)

2 Experimentelle Untersuchungen des Rollstuhls

Für die experimentellen Untersuchungen der Vertikaldynamik wurde der Rollstuhl mit 1-axialen Beschleunigungssensoren an der Schwinge, am Chassis und Sitz (3-axial) ausgerüstet. Als Messsystem wird ein NI cDAQ 9184 System, das hinter dem Sitz am Rollstuhl als autonomes Messsystem aufgebaut wurde, verwendet. Für die spätere Realisierung der semi-aktiven Dämpfung ist ein NI cRio System 9012 vorgesehen. Für die Messungen im Feldversuch kann das Messsystem über WLAN konfiguriert und die Messdaten an einen Rechner versendet werden. Problematisch bei Fahrversuchen auf einer realen Strecke ist die geringe Reproduzierbarkeit der Versuche und Versuchsparameter. Hier sind zwei Aspekte entscheidend:

- Durch die Fahrbahn ergibt sich eine stochastische Anregung aller Räder in lateraler, longitudinaler und vertikaler Richtung
- Erheblicher Einfluss des Fahrergewichts (bis 150 kg) auf das Schwingverhalten (Frequenzen, Dämpfungen)

Somit ist mit diesen Messungen aufgrund der Komplexität ein Abgleich und eine Parameteridentifikation mit einem Mehrkörpermodell nur begrenzt möglich. Daher wurde ein Schwingprüfstand für den Rollstuhl und ein Dummy gemäß DIN 45676 [2] zur Berücksichtigung des vertikalen Schwingverhaltens eines Körpers (55kg – 98kg) entwickelt und gebaut.

2.1 Rollstuhlprüfstand

Der im Labor konstruierte und gebaute Prüfstand ermöglicht es, ein Rad eines Rollstuhls mittels eines quasi-harmonischen Signals anzuregen (siehe **Abbildung 2**). Der Prüfstand besteht aus einem Antrieb, der über eine speziell entwickelte Exzenterkonstruktion den Rollstuhl vertikal am Radaufstandspunkt anregt. Die Exzenterkonstruktion umfasst zwei zueinander verdrehbare Exzenter Scheiben, die über einen Spannsatz miteinander verbunden und von einem Lager umschlossen sind. Die Amplitude der harmoni-

schen Anregung kann durch die Verdrehung der Exzenter zueinander von 0 mm bis ± 5 mm stufenlos eingestellt werden. Der gesamte Rollstuhl kann über Nivellierelemente in der Höhe verstellt werden. Neben der Auslegung und Umsetzung der mechanischen Komponenten des Prüfstands wurde ein elektrischer Schaltschrank gefertigt.

Für die Ansteuerung wurde in der Softwareumgebung LabVIEW ein Prüfstandprogramm programmiert, das aus der Position der Motorwelle den Winkel und die Amplitude der resultierenden Exzentrizität errechnet.

Nach der Installation der ersten Version von dem Projektpartner CAE-Software & Systems erstellten Software *Sound and Vibration Inspector* wurden Schwingungsuntersuchungen des Rollstuhls auf dem Prüfstand durchgeführt. In dem Programm sind entsprechend der Anwendungsklasse Bewertungsfunktionen zur Messung der Schwingungseinwirkung auf den Menschen nach DIN EN ISO 8041 [1] implementiert.

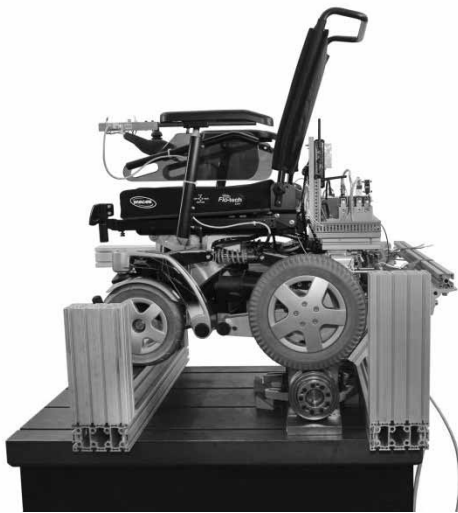


Abbildung 2: Rollstuhlprüfstand

2.2 Fahrerdummy

Parallel zur Prüfstandentwicklung wurde für die Schwingungsuntersuchungen der Vertikalschwingungen am Sitz ein Dummy **Abbildung 3** gemäß DIN 45676 [2] entwickelt, der die mechanische Eingangsimpedanz und die Übertragungsfunktionen eines Menschen besitzt. Das Schwingverhalten des Dummies entspricht einem gedämpften 2-Masse-Schwinger ($m_0=10\text{kg}-18\text{kg}$, $m_1=40\text{kg} - 59\text{kg}$). Aufgaben bei der Konstruktion waren die Auslegung der Feder- und Dämpferparameter für unterschiedliche Körpermassen und die Nachbildung der korrekten Schwerpunktlage nahe an der Rückenlehne. Somit

können reproduzierbare Ergebnisse bei Messungen auf dem Prüfstand unter Berücksichtigung des Einflusses des Fahrers durchgeführt werden.



Abbildung 3: Dummy zur Nachbildung der Eingangsimpedanz eines menschlichen Körpers (98 kg) bei Vertikalschwingungen

3 Aufbau eines MKS-Modells

Ziel ist die Optimierung des Fahrverhaltens insbesondere des Fahrkomforts für den Rollstuhl. Die Auslegung von verbesserten Feder- und Dämpferparametern soll am virtuellen Prototypen erfolgen, somit ist eine Abbildung des dynamischen Verhaltens in einem geeigneten Modell notwendig. Eine erste Auslegung der Fahrwerksparameter erfolgt durch lineare Modelle (2-Massen-Modelle).

3.1 Gesamtmodell

Für die optimale Auslegung ist es notwendig, die nichtlineare Kinematik und Dynamik des Fahrwerks (Lenkeranbindung, Reifenmodelle, Dämpferkennlinien) zu berücksichtigen. Daher wird vom Rollstuhl und Prüfstand ein Mehrkörpermodell in SIMPACK Version 9.5 aufgebaut (siehe **Abbildung 4**).

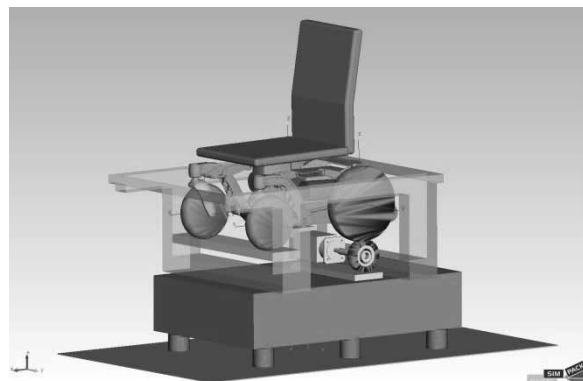


Abbildung 4: MKS-Modell des Rollstuhls und des Prüfstands

Je Fahrwerksseite sind die vordere und hintere Schwingen über ein Feder-Dämpfer-Element miteinander gekoppelt und können sich kinematisch wie

eine Viergelenkkette bewegen. Die hintere Schwinge stützt sich über ein weiteres Feder-Dämpfer-Element an dem Chassis ab. Hierdurch werden alle Starrkörperfreiheitsgrade gefesselt. Für den Aufbau eines Mehrkörpermodells in SIMPACK wurden die relevanten Massen, Trägheiten und Geometrien aus den CAD-Zeichnungen bestimmt.

Derzeit besteht das Modell aus folgenden Elementen:

- 11 Massen: Reifen, Schwingen (mit Motoren), Chassis (Batterie, Sitz, Sitzsäule). Gesamtgewicht 176kg.
- 4 Feder-Dämpfer Elementen (linear). Parameter stammen aus Datenblättern.

3.2 Reifenmodell

Für die Abbildung des Fahrverhaltens ist eine hinreichend genaue Modellierung des Reifens notwendig. Da für die in SIMPACK zur Verfügung stehenden Reifenmodelle die notwendigen Parameter nicht bekannt sind, wird der Aufbau eines eigenen Reifenmodells vorgenommen. Hierfür werden die Standardelemente von SIMPACK genutzt. Für die Modellierung des Kontaktes zwischen Fahrbahn und Reifen wird der 3D-Kontakt (Polygonal Contact Element-Nr. 199) verwendet. Mit diesem Element kann ein 3-dimensionaler Multikontakt zwischen Polygonflächen abgebildet werden. Für die Beschreibung der Reifen-geometrie wurde ein CAD-Modell erstellt und die Fläche durch dreieckige Polygone vernetzt (siehe **Abbildung 5**).

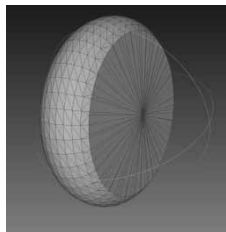


Abbildung 5: CAD-Modell des Reifens mit Polygonelementen für die Modellierung in SIMPACK

Die Fahrbahn mit den Unebenheiten wird ebenfalls durch eine Polygonfläche modelliert. Für die Materialeigenschaften des Reifens wurden ein E-Modul von 20 N/mm² und eine Querkontraktionszahl von 0,4 angenommen.

Die Federeigenschaften des Reifens werden durch ein lineares Feder-Dämpfer-Element abgebildet und zur Bestimmung der Steifigkeit und Dämpfung des Reifens wurden einfache Versuche durchgeführt. Für die Parameteridentifikation wurde ein einfaches MKS-Modell aufgebaut, mit dem die Versuche nachgebildet

wurden und die Werte durch Abgleich zwischen Messungen und Simulationsergebnissen ermittelt:

- Steifigkeit: Messung der Einfederung des Reifens bei statischer Last (dynamische Steifigkeiten werden zunächst nicht berücksichtigt). Steifigkeit Hinterrad = 200 N/mm Vorderrad = 210 N/mm
- Dämpfung: Fallversuch des Reifens und Messung der Sprunghöhe. Dämpfung Hinterrad = 250 Ns/m Vorderrad = 150 Ns/m

Mit diesem Reifenmodell können einfache Fahrmanöver (Beschleunigen/Bremsen, Lenken, Hindernisüberfahrten) hinreichend genau simuliert werden.

3.3 Lineare Analyse

Für eine erste Beurteilung des Modells wurden die Eigenfrequenzen und Eigenschwingformen um die Ruhelage bestimmt. Es ergeben sich folgende Eigenformen:

- 4,87 Hz Nickbewegung des Aufbaus
- 6,15 Hz Wanken des Aufbaus
- 6,94 Hz Huben des Aufbaus

Man erkennt in den Mess- und die Simulationsergebnissen eine entsprechende Anregung der Eigenfrequenzen.

4 Ergebnisse

Die Komfortbewertung der Schwingungen wird nach DIN EN ISO 8041 [1] durchgeführt. In dieser Norm sind Filter für die Nachbildung der menschlichen Empfindlichkeit auf Beschleunigungen angegeben. Im dem Frequenzbereich von 4 Hz – 12 Hz sind die größten Empfindlichkeiten zu finden.

In einer ersten Messung wurde das Spektrum der Übertragungsfunktion zwischen dem Eingangssignal am Exzenter und dem Ausgangssignal Chassis-Dämpferanbindung gemessen (siehe **Abbildung 6**).

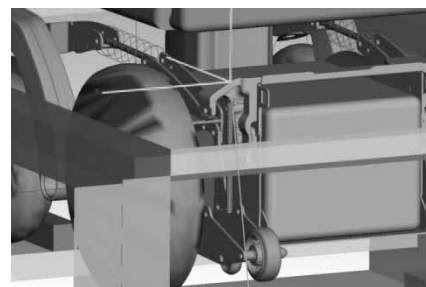


Abbildung 6: Position (grünes Koordinatensystem) des Beschleunigungssensors am Chassis auf der hinteren Schwinge

Als Anregungssignale wurde ein Gleitsinus von 0,5 Hz – 20 Hz und einer Amplitude von $\pm 1,3$ mm bei einer Hochlaufdauer von 40 Sekunden verwendet. Ein Vergleich der Übertragungsfunktion der Wege ist in **Abbildung 7** dargestellt.

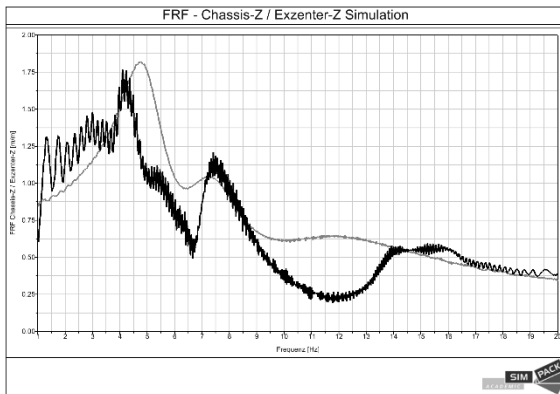


Abbildung 7: Vergleich FRF (Weg) zwischen Messung und Simulationsrechnung

In dem Messsignal sind Resonanzfrequenzen bei 4,2 Hz und 7,4 Hz zu erkennen. Die zugeordneten Eigenfrequenzen finden sich auch in dem MKS-Modell und können dort den Eigenbewegungen zugeordnet werden. Dies sind erste Ergebnisse für die Messungen und Simulationsrechnungen und voraussichtlich wird sich eine noch bessere Übereinstimmung der Ergebnisse bei genauerer Modellierung des Rollstuhls und Durchführung eines Parameterabgleichs ergeben.

5 Konzept zur Optimierung des Fahrverhaltens

In [4] sind grundlegende Konzepte und Überlegungen zur Verbesserung der Fahrsicherheit und des Fahrkomforts im Automobilbereich beschrieben. Daran angelehnt soll für den Rollstuhl die Verbesserung in drei Phasen erfolgen:

1. Phase: Auslegung von optimalen Feder- und Dämpfungsparametern bei unterschiedlichen Fahrergewichten. Daraus abgeleitet können optimale passive Feder-Dämpferelemente je nach Fahrergewicht verwendet werden.
2. Phase: Entwicklung eines semi-aktiven Feder- und Dämpfungsconzeptes zur automatischen Anpassung der Parameter an unterschiedliche Fahrergewichte.
3. Phase: Entwicklung eines mechatronischen Systems mit semi-aktiven Dämpfern für die Regelung des Fahrkomforts. Regelgröße ist die bewertete Beschleunigung am Fahrersitz und Stellgröße ist die Dämpfungskonstante des Dämpfers (Regelung der angelegten Spannung).

Ein aktives Federungskonzept ist aufgrund der begrenzt zur Verfügung stehenden Energie nicht realisierbar. Für die Realisierung eines semi-aktiven Federungskonzeptes wird ein steuerbarer elektromechanischer Dämpfer (Fludicon RheDamp RD 32 [6]) angeschafft [5]. In **Abbildung 8** ist die Dämpferkraft bei unterschiedlichen Spannungen und Geschwindigkeiten dargestellt.

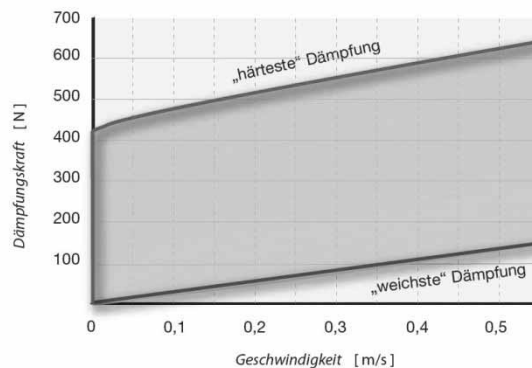


Abbildung 8: Kennlinie Fludicon RheDamp RD 32 aus [6]

In einer aktuellen Arbeit erfolgt die konstruktive Anpassung des Rollstuhls für den Einbau des steuerbaren Dämpfers.

Für die Auslegung der Parameter wird das MKS-Modell des Rollstuhls durch den Einbau von steuerbaren Dämpferelementen und Regelungen erweitert. Die Entwicklung des notwendigen mechatronischen Systems wird modellbasiert (SIMPACK, MATLAB, LabVIEW) durchgeführt. Dabei soll durch Co-Simulation mit Matlab der Entwurf der Federungsparameter (Steifigkeiten, Dämpfungen) und Regelparameter erfolgen. Anschliessend erfolgt die Realisierung des Systems und Überprüfung der Ergebnisse.

6 Literatur

- [1] DIN EN ISO 8041:2005 Schwingungseinwirkung auf den Menschen – Messeinrichtung, Deutsche Fassung, Stand Juni 2006.
- [2] DIN 45676 Mechanische Eingansimpedanzen und Übertragungsfunktionen des menschlichen Körpers, Stand Juni 2003.
- [3] M. Hauck: Geregelter Dämpfung für Traktor-Fahrersitze, Dissertation TU Berlin, FB Maschinenbau, Berlin 2001.
- [4] R. Isermann: Fahrdynamik-Regelung - Modellbildung, Fahrerassistenzsysteme, Mechatronik, Berlin 2001.

ronik, ATZ/MTZ Fachbuch, Vieweg Verlag, ISBN-10 3-8348-0109-7, 2006.

- [5] H. Janocha: Adaptronics and Smart Structures, Springer Berlin Heidelberg New York, ISBN 978-3-540-71965-6, 2007.
- [6] Technische Beschreibung des Fuldion Dämpfer RheDamp RD 32 von Fa. Fludicon, Landwehrstraße 55, 64293 Darmstadt, www.fludicon.de

Konzeption eines hochdynamischen Systems mit sphärischem Elektroantrieb

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

¹Ostfalia Hochschule für angewandte Wissenschaften,
Salzdahlumer Str. 46/48, 38302 Wolfenbüttel, Deutschland

^a*mar.goellner@ostfalia.de*, ^b*x.liu-henke@ostfalia.de*

Im Rahmen eines Forschungsvorhabens soll ein hochdynamisches System mit sphärischem Elektroantrieb namens S(phere)- Mobile mechatronisch entwickelt werden. Im Zuge dessen wird in diesem Aufsatz die mechatronische Komposition des Gesamtsystems vorgestellt. Zudem soll ein physikalisches und mathematisches Modell eines ersten Entwicklungsschrittes des S-Mobiles symbolisch hergeleitet werden. Dieses Simulationsmodell ist so aufgebaut, dass seine grundsätzlichen Annahmen auf die weiteren Entwicklungsschritte des S- Mobile übertragen werden können. Zu diesem wird eine hierarchische Regelstruktur modellbasiert entworfen, welche das grundsätzlich instabile Streckenverhalten des S-Mobile stabilisiert. Die Simulationsergebnisse verschiedener Anregungsarten werden dargelegt und diskutiert.

1 Einleitung

Im Rahmen eines Forschungsvorhabens soll ein S(phere)- Mobile, welches ein hochdynamisches System mit sphärischem Elektroantrieb darstellt, mechatronisch entwickelt werden. Als Entwicklungsmethode wird die durch mechatronische Komposition unterstützte Entwicklungssystematik verwendet. Beim Kernmodul des S-Mobile (Abbildung 1) handelt es sich hauptsächlich um ein typisches mechatronisches System, dass aus einer Antriebskugel und einem darauf balancierten Aufbau als mechanische Tragstruktur, den Aktuatoren, Sensoren und der hierarchischen Reglerstruktur besteht. In diesem vorliegenden Beitrag soll eine modellbasierte Konzeption des Kernmoduls des S-Mobile vorgestellt werden.

Das S-Mobile ist aus physikalischen Gegebenheiten instabil. Durch den Einsatz eines Reglers kann ein stabiles Verhalten auf dieses instabile System aufgebracht werden. In [1] wurde ein entsprechendes System mit Quer- und Längsdynamikregelung vorgestellt. Dieser von Honda gebaute und unter dem Namen UniCub käuflich zu erwerbende Ein-Mann-Transporter verfügt über ein des S-Mobile gleichen Bewegungsspektrums, jedoch mit einer geringeren Dynamik.

Des Weiteren ist der aus [2] bekannte Segway ein solches System, welches allerdings in seiner Querachse stabil und nur in seiner Längsachse quasi-stabil ausgeführt ist. Sein Bewegungsspektrum, einschließ-

lich Freiheitsgrade, ist dementsprechend geringer als das des S-Mobile. Folglich ist auch der Aufwand der mechatronischen Entwicklung geringer.

Der Kugelantrieb ist derweilen nur in der Forschung, zum Bau von Robotersystemen für die Interaktion in menschlichen Bewegungsräumen, eingesetzt worden. Entsprechende Roboter wurden in [3], [4], [5], [6], [7] und [8] beschrieben und befinden sich in der Entwicklungs- oder Prototypenphase.

2 Konzept

2.1 S-Mobile als mechatronisches System

Die folgende Abbildung 1 zeigt das Konzept des S-Mobile, bestehend aus der Antriebskugel und dem Aufbau. Der Aufbau ist untergliedert in die mechanische Tragstruktur (vgl. Kapitel 2.2) sowie den Aktoren, der Sensorik und der Informationsverarbeitung, welche mit der Antriebskugel das Kernmodul bilden.

Als Aktoren werden dabei Elektromotoren eingesetzt, da die geforderte hohe Systemdynamik mit anderen Übertragungsmedien nicht gewährleistet ist. Zudem lässt sich der elektrische Antriebsstrang, inklusive der elektrischen Energiespeichereinrichtung, im geforderten kompakten System leicht integrieren. Hinsichtlich der Anforderungen an die Schnelligkeit und Bandbreite der Aktoren, aus Sicht des zu entwerfenden Regelkreises, kommen hier nur elektrische Motoren in Frage.

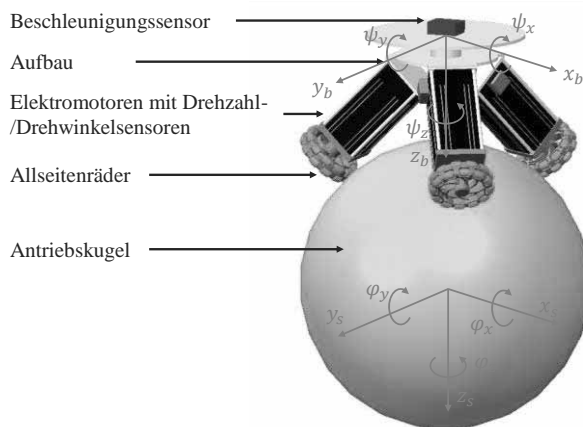


Abbildung 1. Konzept des S-Mobile

Zur Messwerterfassung müssen zudem diverse Sensoren in das System integriert werden. Zur Erfassung der Aufbaubewegung eignen sich hier ein linearer Beschleunigungssensor (Accelerometer) sowie Gyroskope zur Winkelgeschwindigkeitserfassung. Die Bewegung der Antriebskugel lässt sich über Odometrie erfassen. Das heißt, dass über eine Drehzahlerfassung der Antriebsmotoren, sowie eine Umrechnung dieser in das Ursprungskoordinatensystem, auf die Bewegung der Antriebskugel geschlossen wird.

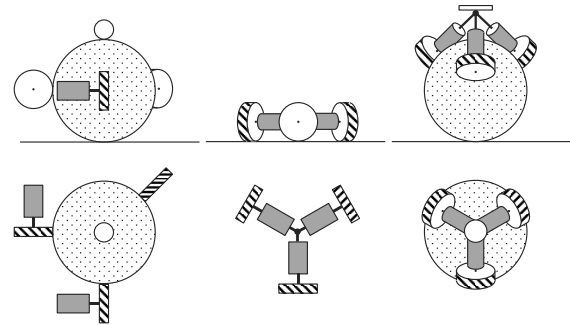
Dadurch ermöglicht das intelligente Kernmodul eine kontrollierte Bewegung des S-Mobile in fünf vorgesehenen Freiheitsgraden, der Längsbewegung in x- und y-Ebene, der Drehbewegung um die Hochachse (gieren), sowie wanken und nicken. Die Aufbaubewegung wird entsprechend der Regelung nach dem Prinzip eines mehrachsigen, inversen Pendels kontrolliert.

2.2 Konfiguration der mechanischen Tragsstruktur

Da die Freiheitsgrade des Aufbaus sehr stark miteinander verkoppelt sind, sollen drei verschiedene Konfigurationen konzipiert werden (Abbildung 2), um die gewünschten kinematischen und dynamischen Funktionen des Systems sukzessiv zu realisieren. Diese drei Konfigurationen sind so gewählt, dass aufgrund ihrer mechanischen Struktur verschiedene Modellierungstiefen für die Realisierung nötig sind.

Zunächst wird ein maßstäblich verkleinertes System, mit geringerer Detaillierungsstufe, als erstes Basis-konzept ausgelegt. Das System besteht aus einem äquatorialen Antrieb mit zwei um 90° versetzten Antriebsrädern, einem dazu jeweils um 135° gegenüberstehendem Stützrad und einem einzelnen, zentra-

len Kugellager zur Aufnahme der Vertikalkräfte (Abbildung 2: Konfiguration 1). Durch die vorteilhafte Anordnung der Antriebe im 90° Winkel können die beiden Längsachsen entkoppelt werden, wodurch nur ein zwei-dimensionales kinematisches Modell abgebildet werden muss [3].



Konfiguration 1 Konfiguration 2 Konfiguration 3

Abbildung 2. Die drei Konzeptphasen im Überblick

Als zweiter Schritt wird ein System mit drei Antrieben, welche jeweils mit Allseitenrädern (Mecanum- oder Omnidirektionale Räder) versehen und in einem planaren Winkel von 120° zueinander angeordnet sind, ausgestattet (Abbildung 2: Konfiguration 2). Diese Konfiguration ist nun durch die Drehzahl und Drehrichtung der Antriebsmotoren lenk- und rotierbar [9]. Durch die (grenz-)stabile Regelstrecke ist hier nur ein Querdynamikmodell des Systems erforderlich. Das Antriebskonzept wäre dem gewünschten Kugelantrieb insofern ähnlich, als dass man den Kugeldurchmesser sehr groß wählt. Er wäre dann dem Durchmesser der Erde gleich. Nun würde sich diese Maschine mit dem planaren Antrieb invers zu dem Kugelantrieb in Bezug auf das Erdkoordinatensystem bewegen.

In dritter Phase wird eine Konfiguration mit weitergehender Komplexität konzipiert werden. Es verfügt über einen sphärisch angeordneten Antrieb mit drei um 120° Sphärengrad versetzten Antriebsrädern. Durch diese geeignete Anordnung können jegliche weitere Stützstrukturen, wie Räder oder Lager, eingespart werden. Zudem ermöglicht dieses Antriebskonzept ein aufrechtes Drehen des Aufbaus um seine Hochachse, was mit der Konzeptphase 1 nicht möglich ist [4]. Weiterhin sind für die Realisierung Omniwheels oder Allseiten-Räder als Antriebsräder nötig (Abbildung 2: Konfiguration 3). Zur dieser

Konzeptphase wird ein drei-dimensionales Streckenmodell benötigt [2], [10].

Basierend auf diesen Konzepten kann der mechatronischen Entwurf durchgeführt werden.

3 Modellbildung

Die Entwicklung der gezeigten Konzepte erfordert eine systematische und interdisziplinäre Vorgehensweise, die eine optimale Auslegung des Gesamtsystems erlaubt. Diese mechatronische Entwicklungsmethode erfolgt in einem iterativen Prozess bestehend aus Modellbildung, Analyse und der modellbasierten Auslegung. Das Gesamtsystem wird dabei in Einzelsysteme unterteilt und anschließend mit Hilfe der mechatronischen Komposition zu einem Gesamtmodell zusammengeführt. Durch die modellbasierte Auslegung können z.B. benötigte Stellkräfte gefunden und zum Auslegen von Aktoren genutzt werden [11], [12]. Dieser Entwicklungsprozess wird im Folgenden anhand des Basiskonzeptes (Konfiguration 1) durchgeführt.

Die folgende Abbildung 3 zeigt ein vereinfachtes physikalisches Ersatzmodell, welches die Kinematik und Dynamik des Systems repräsentiert

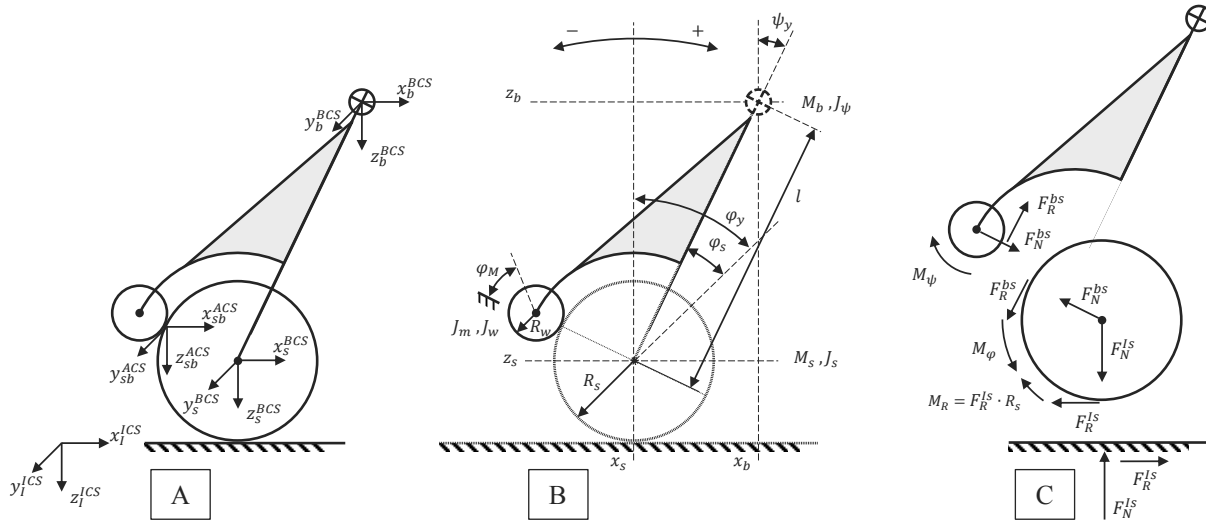


Abbildung 3. Physikalisches Streckenmodell

Um das Verhalten eines solchen dynamischen Systems zu untersuchen und anschließend eine hierarchische Regelstruktur zu entwerfen, müssen sowohl die Kinematik-, als auch die Dynamikfunktionen ausgelegt werden. Die Kinematik befasst sich mit den Bewegungsmöglichkeiten der Glieder eines gegebenen Mechanismus relativ zueinander, sowie auftretenden Geschwindigkeiten und Beschleunigungen innerhalb

des Mechanismus. Das kinematische Verhalten wird aus den Freiheitsgraden und der Geometrie des Systems bestimmt. Die Dynamikfunktionen beschreiben das dynamische Verhalten unter Wirkung von Kräften.

Aus den kinematischen und dynamischen Zusammenhängen des realen Systems soll, unter der Verwendung geeigneter Vereinfachungen, ein Simulationsmodell zur Auslegung eines Regelkonzeptes realisiert werden. Dieses Streckenmodell der xz -Ebene ist dann ebenfalls für die Beschreibung der yz -Ebene verwendbar. Einzig eine Winkelkonvertierung muss vorgenommen werden. Vergleicht man die angetragenen Koordinatensysteme mit dem angetragenen Aufbauwinkel ψ in der xz -Ebene in Abbildung 3 ist zu erkennen, dass sowohl Kugel- als auch Aufbauwinkel bei einer Projektion auf die yz -Ebene in entgegengesetzte Richtung als positiv angenommen werden müssen.

Im Folgenden werden Kinematik und Dynamik im Detail dargestellt.

3.1 Kinematikfunktion

Die kinematischen Zusammenhänge werden aus den Freiheitsgraden und der Geometrie des S-Mobile bestimmt. Dies geschieht durch die Belegung der einzelnen Teilkomponenten mit geeigneten Koordinatensystemen nach Abbildung 3A.

Das Inertialkoordinatensystem (ICS) liegt auf der Fahrbahn, die das S-Mobile benutzt. Die jeweiligen Körperkoordinatensysteme (BCS) liegen in jedem Starrkörper und Verbindungskoordinatensysteme (ACS) in den Koppelpunkten zwischen diesen. Die Kinematik des Gesamtsystems wird durch Koordinatentransformation mittels Transformationsmatrizen symbolisch hergeleitet und später in einen globalen Regelkreis eingebettet. Relevant für die Herleitung des Streckenmodells sind im Speziellen das Inertialkoordinatensystem mit den Komponenten (x_I, y_I, z_I) sowie die Körperkoordinatensysteme des Aufbaus BCS^b (x_b, y_b, z_b) und der Kugel BCS^s (x_s, y_s, z_s).

So entsteht exemplarisch, durch die Drehung des BCS^b um die y-Achse und anschließend um die x-Achse des ICS, die folgende Transformationsmatrix R_{Ib} [11]:

$$R_{Ib} = \begin{bmatrix} \cos \psi_y & \sin \psi_y \sin \psi_x & \sin \psi_y \cos \psi_x \\ 0 & \cos \psi_x & -\sin \psi_x \\ \sin \psi_y & \cos \psi_y \sin \psi_x & \cos \psi_y \cos \psi_x \end{bmatrix} \quad (1)$$

Eine Drehung um die z-Achse ist nicht notwendig, da der Mechanismus der Tragstruktur des Basiskonzeptes eine Rotation um diese Achse aus eigener Kraft dem Körper nicht erlaubt.

3.2 Dynamikfunktion

Die Dynamikfunktionen beschreiben das dynamische Verhalten unter Wirkung von Kräften. Unter Zuhilfenahme der physikalischen Gesetze kann eine mathematisch formulierte Aussage über das Systemverhalten gemacht werden [11]. Die Dynamik des S-Mobile setzt sich aus der Dynamik der Tragstruktur, der Aktoren, Sensoren und der digitalen Signalverarbeitung zusammen. Mit Hilfe geeigneter Vereinfachungen wird ein räumliches Modell des S-Mobile erstellt.

1) Dynamikfunktion der Tragstruktur

Die vereinfachte Abbildung 3B zeigt neben festen geometrischen Eigenschaften, wie den Radien der Antriebskugel (r_s) und des Antriebsrades (r_w), auch physikalische Größen, wie die Massen (M_s, M_b) und Massenträgheitsmomente (J_s, J_b, J_m, J_w), der einzelnen Komponenten. Des Weiteren sind die Systemzustände dargestellt, der Aufbauwinkel (ψ_y), der Rotationswinkel der Kugel (φ_y) und des Antriebsrades bzw. Motors (φ_m), sowie die daraus ableitbaren Koordinaten der Teilkomponenten in der Ebene (x_s, z_s und x_b, z_b). Für die Beschreibung des Systems sind der Aufbauwinkel und der Rotationswinkel der Kugel als Zustandsgrößen definiert.

Nach diesem Schaubild gelten folgende Zusammenhänge für die Bewegung der Kugel (Index s).

$$\vec{q}_s = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} r_s \cdot \varphi_y \\ r_s \cdot \varphi_x \\ r_s \end{bmatrix} \quad (2)$$

$$\dot{\vec{q}}_s = \begin{bmatrix} r_s \cdot \dot{\varphi}_y \\ r_s \cdot \dot{\varphi}_x \\ 0 \end{bmatrix} \quad (3)$$

Mit Hilfe der Gleichungen (1) und (2) ergibt sich für die Bewegung des Aufbaus (Index b).

$$\vec{q}_b = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \vec{q}_s + R_{Ib} \cdot \begin{bmatrix} 0 \\ 0 \\ l \end{bmatrix} = \begin{bmatrix} x_s + l \sin \psi_y \cos \psi_x \\ y_s - l \sin \psi_x \\ z_s + l \cos \psi_x \cos \psi_y \end{bmatrix} \quad (4)$$

$$\dot{\vec{q}}_b = \begin{bmatrix} r_s \dot{\varphi}_y + l \dot{\psi}_y \cos \psi_x \cos \psi_y - l \dot{\psi}_x \sin \psi_y \sin \psi_x \\ r_s \dot{\varphi}_x - l \dot{\psi}_x \cos \psi_x \\ -l \dot{\psi}_y \sin \psi_y \cos \psi_x - l \dot{\psi}_x \sin \psi_x \cos \psi_y \end{bmatrix} \quad (5)$$

Bedingt durch die Konstruktion der Basiskonfiguration können einige weitere Vereinfachungen durchgeführt werden. Unter der Annahme kleiner Auslenkungswinkel kann das planare Modell der xz-Ebene als entkoppelt von der yz-Ebene betrachtet werden und vice versa. Die Funktionen können entsprechend vereinfacht werden, indem der jeweilige Winkel zu Null gesetzt wird. Aus gleichen Gründen können Kopplungskräfte vereinfacht werden.

Die geometrischen Größen der Antriebsräder ergeben sich durch die feste Verbindung mit dem Aufbau. Die Räder haben dabei nur einen Freiheitsgrad, da sie nur um ihre Motorachse drehen können. Exemplarisch für das in Abbildung 3 dargestellte Rad (Index w) mit dem zugehörigen Motor (Index m) gilt.

$$\varphi_m = (\varphi_y - \psi_y) \frac{r_s}{r_w} \quad (6)$$

$$\varphi_m = (\varphi_y - \psi_y) \frac{r_s}{r_w} \quad (7)$$

Das Aufstellen der Differentialgleichungen erfolgt nach dem Lagrange Formalismus.

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \left(\frac{\partial L}{\partial q} \right) = \vec{Q} \quad (8)$$

Die Gesamtenergiebilanz (L) ergibt sich aus der Differenz der kinetischen (T) und potenziellen (V) Energien.

$$L = T - V \quad (9)$$

Die kinetischen Energien ergeben sich jeweils zu.

$$T = \sum_{i=1}^n \left(\frac{1}{2} m \cdot \dot{q}^T \cdot \dot{q} + \frac{1}{2} J \cdot \dot{\vec{\varphi}}^T \cdot \dot{\vec{\varphi}} \right)_i \quad (10)$$

Die potenziellen Energien aus.

$$V = \sum_{i=1}^n (m \cdot \vec{q} \cdot \vec{g})_i \quad \text{mit} \quad g = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (11)$$

Aufgrund der auf beide Starkkörper ausgeübten Kräfte und Momente wird die Q-Matrix hergeleitet. Die Abbildung 3C zeigt Drehmomente und Kräfte am System. Die Drehmomente M_ψ und M_ϕ stellen die Eingangsgrößen des Systems dar. M_ψ wird dabei durch den Elektromotor des Systems generiert. Über die kinematische Beziehung des Antriebsrades zur Antriebskugel entsteht dann M_ϕ . Die Reibkraft F_R^{bs} zwischen dem Rad und der Kugel soll hier vereinfacht als Ideal angenommen werden. Durch die Aufteilung in zwei separate Momente kann später ein entsprechendes Reibungsmodell in die Simulation eingebracht werden. Somit ist Q definiert als.

$$\vec{Q} = \begin{bmatrix} M_\phi \\ M_\psi \end{bmatrix} \quad (12)$$

Nach Vereinfachung entstehen zwei nicht-lineare gekoppelte Differentialgleichungen zweiter Ordnung für die Beschreibung der Bewegung in der xz-Ebene.

$$M_\phi = f(\ddot{\varphi}_y, \ddot{\psi}_y, \dot{\psi}_y, \psi_y, m_s, m_b, r_s, r_w, J_w, J_m, J_s, l) \quad (13)$$

$$M_\psi = f(\ddot{\varphi}_y, \ddot{\psi}_y, \dot{\psi}_y, \psi_y, m_b, r_s, r_w, J_w, J_m, J_\psi, l, g) \quad (14)$$

Unter der Annahme kleiner Winkelbewegungen und Geschwindigkeiten sollen diese nach folgendem linearisiert werden.

$$\Delta M = \left(\frac{\delta f}{\delta \ddot{\varphi}_y} \right)_0 \Delta \ddot{\varphi}_y + \left(\frac{\delta f}{\delta \ddot{\psi}_y} \right)_0 \Delta \ddot{\psi}_y + \left(\frac{\delta f}{\delta \dot{\psi}_y} \right)_0 \Delta \dot{\psi}_y + \left(\frac{\delta f}{\delta \psi_y} \right)_0 \Delta \psi_y \quad (15)$$

Es entstehen zwei lineare gekoppelte Differentialgleichungen zweiter Ordnung. Dargestellt mithilfe der Massen- (E) und Gravitationsmatrizen (G).

$$\begin{bmatrix} E_{1,1} & E_{1,2} \\ E_{2,1} & E_{2,2} \end{bmatrix} \cdot \begin{bmatrix} \ddot{\varphi} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} G_{1,1} & G_{1,2} \\ G_{2,1} & G_{2,2} \end{bmatrix} \cdot \begin{bmatrix} \varphi \\ \psi \end{bmatrix} = \begin{bmatrix} M_\phi \\ M_\psi \end{bmatrix} \quad (16)$$

Aus diesen lässt sich die Zustandsdarstellung des Systems ableiten.

$$\dot{\underline{x}} = \underline{A} \cdot \underline{x} + \underline{B} \cdot \underline{u} \quad \text{mit} \quad \underline{x} = \begin{bmatrix} \varphi_y \\ \psi_y \\ \dot{\varphi}_y \\ \dot{\psi}_y \end{bmatrix} \quad \text{und} \quad \underline{u} = \begin{bmatrix} M_\phi \\ M_\psi \end{bmatrix} \quad (17)$$

$$\dot{\underline{y}} = \underline{C} \cdot \underline{x} + \underline{D} \cdot \underline{u} \quad \text{mit} \quad \underline{y} = \begin{bmatrix} \varphi_y \\ \psi_y \\ \dot{\varphi}_y \\ \dot{\psi}_y \end{bmatrix} \quad (18)$$

Die entsprechenden Matrizen in den Gleichungen (17) und (18) lauten:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & G_{2,2} \frac{E_{1,2}}{\det(E)} & 0 & 0 \\ 0 & -G_{2,2} \frac{E_{1,1}}{\det(E)} & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{E_{2,2}}{\det(E)} & -\frac{E_{1,2}}{\det(E)} \\ -\frac{E_{2,1}}{\det(E)} & \frac{E_{1,1}}{\det(E)} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Damit ist das Streckenmodell definiert.

2) Dynamikfunktion des Aktors

Die Aktorik besteht aus einem elektrischen Motor und einer Leistungselektronik, die einen 4-Quadrantenbetrieb ermöglicht. Während die Leistungselektronik im Modell durch eine Totzeit Berücksichtigung findet, wird das Verhalten des Antriebs durch eine DGL 1. Ordnung modelliert [12]. Die Funktion für das Motormoment lautet:

$$M_\psi = \frac{c_{mech}}{R_m} \cdot \left(U(t) - c_{elekt} \cdot \frac{r_w}{r_s} \cdot (\dot{\varphi}_y - \dot{\psi}_y) \right) \quad (19)$$

Die Eingangsgröße hier ist die Betriebsspannung $U(t)$, welche über motorspezifische Kenngrößen, wie die mechanische (c_{mech}) und elektrische Motorkonstante (c_{elekt}), sowie den Spulenwiderstand R_m zu einem Drehmoment in Abhängigkeit der Induktionsspannung umgerechnet werden kann.

3) Dynamikfunktion der Sensorik & digitalen Signalverarbeitung

Ähnlich verhält es sich mit der Sensorik zur Erfassung der Winkelgeschwindigkeiten, der Wegbeschleunigungen und der Aktordrehzahl. Diese Sensoren werden über ein PT1-Systemverhalten in der Simulation berücksichtigt. Durch die niedrige Zeitkonstante eines Sensors soll diese Näherung auch im weiteren Entwicklungsverlauf des Simulationsmodells bestehen bleiben.

Die digitale Realisierung eines Reglers führt zu einer Phasenabsenkung. Diese resultiert aus den auftretenden Totzeiten, die durch den Abtast-Haltevorgang und die zur Berechnung des Regelalgorithmus notwendige Bearbeitungszeit verursacht werden. Dies kann

durch ein System mit endlich vielen Zustandsgleichungen approximiert werden. Zur Nachbildung dieser digitalen Effekte wird eine Padé-Approximation 2. Ordnung im Modell implementiert [12].

4 Modellbasierte Reglerauslegung

Nach der Modellbildung wird der Regler modellbasiert ausgelegt, denn durch die instabile Strecke führen Anregungen des Systems immer zu einem instabilen umkippen aus der Ruhelage, die nicht wieder kompensiert werden kann. Der Regler soll dementsprechend das System durch ständige Nachregelung stabil halten.

Zu diesem Zwecke wurde eine hierarchisch angeordnete Regelstruktur nach Abbildung 4 entworfen. Die über Sensoren erfassbaren Messgrößen werden über die Vorwärtskinematik des Systems auf die zu regelnden Systemgrößen des Aufbaus transformiert und dem globalen Modalregler zur Verfügung gestellt. Dieser generiert die Modalkräfte und -momente zur Stabilisierung des Systems.

Anschließend wird über die Rückwärtskinematik des Systems und die Sollwerte für die unterlagerte lokale Regelung, die Dynamik jedes einzelnen Motors berechnet [11]. Diese Sollwerte werden der lokalen Kaskadenregelung, bestehend aus Drehzahl und Motorstromregler, übergeben [13].

Um das Verbesserungspotential zu zeigen, wird eine Simulation durchgeführt. Hier soll zunächst das Verhalten der Aufbauwinkelregelung gezeigt werden. Eine Anregung (Abbildung 5) erfolgt durch die Aufschaltung eines Drehmomentes auf den Aufbau und, als Reaktionsmoment, auf die Antriebskugel im Simulationsmodell der Strecke. Nachfolgend wird eine Sprunganregung auf den Regelkreis gegeben.

Die Antwort des Systems auf diese Anregung unter der Verwendung des Modalreglers zeigt Abbildung 6. Es ist zu sehen, dass der Aufbauwinkel zunächst ausschlägt, bevor er durch eine entgegengesetzt wirkende Beschleunigung ausgeregelt wird. Des Weiteren kann ein ähnlicher Effekt beobachtet werden, wenn die Sprunganregung zurückgenommen wird. Der negative Winkelausschlag ist bedingt durch das nun

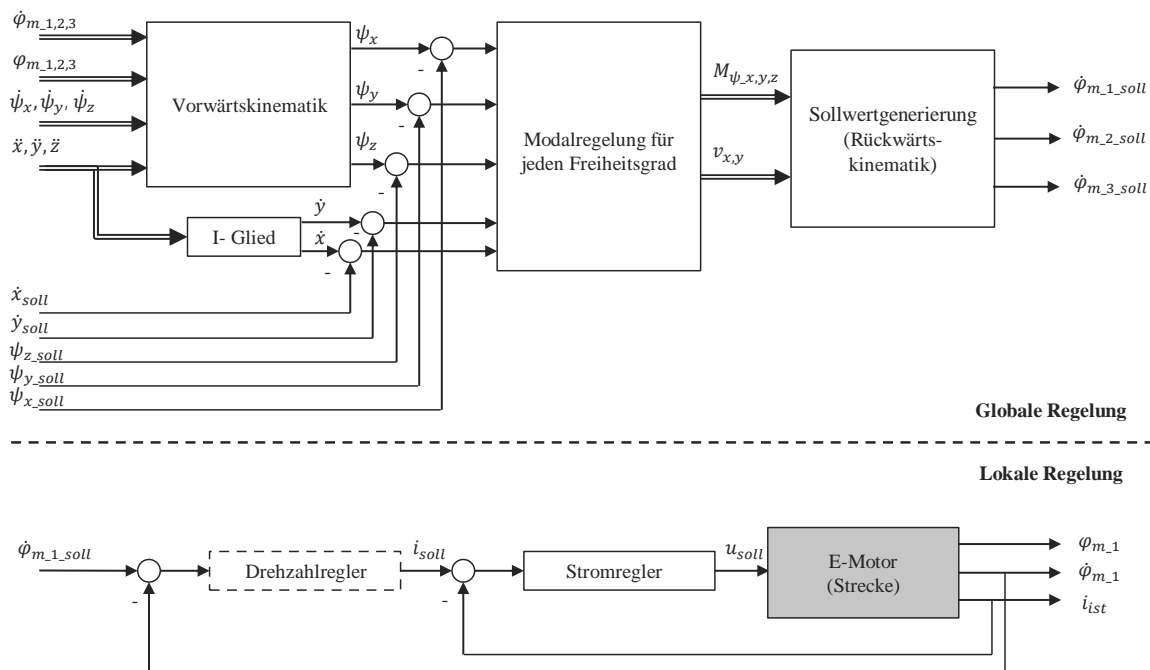


Abbildung 4. Reglerstruktur der Antriebseinheit [13]

zu starke Motormoment. Eine Kompensation des Reglers erfolgt durch einen Abbremsvorgang.

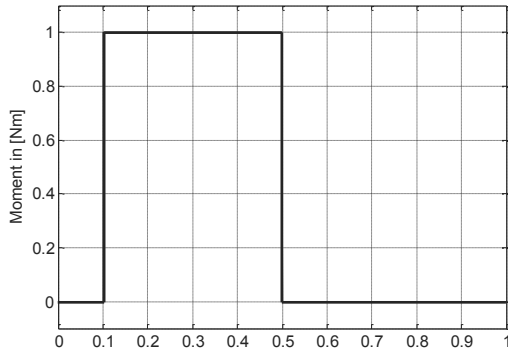


Abbildung 5. Sprunganregung des Gesamtsystems

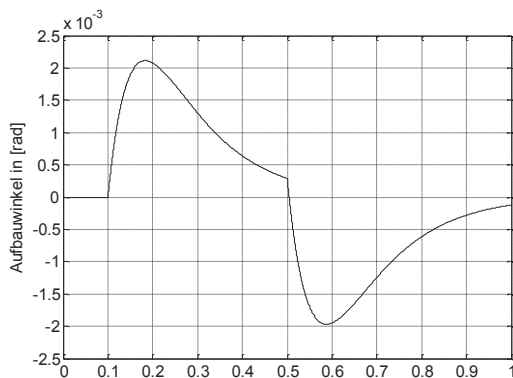


Abbildung 6. Sprungantwort des Gesamtsystems

5 Zusammenfassung und Ausblick

In dem vorliegenden Beitrag wurden Konzepte zur Realisierung des S-Mobile vorgestellt. Anhang der ersten Basiskonfiguration wurde der mechatronische Entwurf beispielhaft durchgeführt, indem zunächst ein Streckenmodell generiert und anschließend analysiert wurde. Zudem wurde eine hierarchische Reglerstruktur, bestehend aus einem globalen Regler und einer untergeordneten, kaskadierten lokalen Regelung, entworfen. In der weiteren Forschungsarbeit soll der mechatronische Entwurf auf die Konfiguration mit höchster Komplexität angewandt, und die mechatronischen Funktionen des S-Mobile modellbasiert ausgelegt werden.

6 Referenzen

- [1] Honda Motor Co: *Honda Announces New Personal Mobility Device, UNI-CUB β Enhancing Compatibility with People*. URL:

<http://world.honda.com/news/2013/c131114Personal-Mobility-UNI-CUB/index.html>, Zugriff am 25.01.2014

- [2] Urban Mobility Germany Authorized Segway Distributor GmbH: *Segway – Das Original! - Die wissenschaftliche Basis hinter der Technologie*. URL: <http://www.segway.de/technologie> Zugriff am 25.01.2014
- [3] Lauwers, T. B.; Kantor, G. A.; und Hollis, R. L.: *A Dynamically Stable Single-Wheeled Mobile Robot with Inverse Mouse-Ball Drive*. Proc. IEEE Int'l. Conf. on Robotics and Automation, Carnegie Mellon University, USA, 2006
- [4] Nagarajan, U.; Mampetta, A.; Kantor, G. A. und Hollis, R. L.: *State Transition, Balancing, Station Keeping, and Yaw Control for a Dynamically Stable Single Spherical Wheel Mobile Robot*. IEEE International Conference on Robotics and Automation, in Kobe, Japan, S. 998-1003, 2009
- [5] Frankhauser, P. und Gwerder, C.: *Modeling and Control of a Ballbot*. Bachelor Thesis, ETH Zürich, Schweiz, 2010.
- [6] Kumagai, M. und Ochiai, T.: *Development of a Robot Balanced on a Ball*. Journal of Robotics and Mechatronics Vol.22 No.3, S. 348-349, 2010
- [7] Fong, J. und Uppill, S.: *Design and Build a Ballbot*. Report, University of Adelaide, Australien, 2009
- [8] Endo, T. and Nakamura, Y.: *An Omnidirectional Vehicle on a Basketball*. Report, University of Tokyo, Japan, 2005
- [9] Ahlers, M. T.: *OmniVehicle: Plattform mit omnidirektionalem Antrieb*. URL: <http://www.mtahlers.de/index.php/robotik/omnivehicle>, Zugriff am 25.01.2014
- [10] Pellegrini, E.; Diepold, K. J.; Dessort, R. und Panzer, H.: *3D-Modelling of a Robot Balancing on a Ball*. Technical Reports on Automatic Control Vol. TRAC-6, S. 1- 28, 2011
- [11] Liu-Henke, X.; Lückel, J. und Jäker, K.-P.: *Ganzheitlicher mechatronischer Entwurf eines aktiven Feder-/Neigemoduls*. VDI-Tagung

Mechatronik – Mechanisch/Elektrische Antriebstechnik, in Wiesloch, Deutschland, 2000

- [12] Liu-Henke, X.; Buchta, R. und Quantmeyer, F.: *Simulation eines mechatronischen Lenkungsmoduls für ein Elektrofahrzeug mit dezentralen Direktantrieben*. ASIM, in Krefeld, Deutschland, 2011
- [13] Liu-Henke, X.; Nachtigal, V.: *Ein integriertes Hardware-in-the-Loop-System zur Funktionsabsicherung von vernetzten Fahrwerksregelsystemen*. AUTOREG Fachtagung, in Wiesloch, Deutschland, 2006

Nutzung einer Parametererregung in MEMS

Till Kniffka¹, Johannes Welte², Horst Ecker¹

¹ Technische Universität Wien, Österreich

² Universität für Bodenkultur, Wien, Österreich

horst.ecker@tuwien.ac.at

Mikroelektromechanische Systeme (MEMS) werden heute schon millionenfach eingesetzt, bewähren sich in vielerlei Anwendungen und tragen zur fortschreitenden Miniaturisierung von Geräten und Maschinen bei. Trotzdem sind die technischen Möglichkeiten noch bei weitem nicht ausgereizt und können durch z.B. komplexere MEMS-Strukturen neue Nutzungsmöglichkeiten erschlossen werden. Während es in konventionellen mechanischen Systemen eher schwierig ist, eine Parametererregung gezielt zu realisieren, bieten im MEMS die sogenannten Kammantriebe eine sehr einfache Möglichkeit, eine gesteuerte auslenkungsproportionale Kraft zu erzeugen. Wird diese Quasi-Steifigkeit zeitlich periodisch verändert, dann spricht man von einer Parametererregung und damit erschließen sich für MEMS neue Möglichkeiten. In dem Beitrag werden die Grundlagen und die Möglichkeiten der gezielten Nutzung von „Parameterresonanzen“ (eigentlich Instabilitäten) in MEMS gezeigt. Insbesondere wird auch auf das Phänomen der Parameter-Antiresonanz eingegangen und Simulationsstudien zu diesem neuen Effekt anhand eines realitätsnahen Modells vorgestellt. Im Falle der Parameterresonanz wird gezeigt, dass je nach Ausprägung der Instabilität, die Amplituden wesentlich rascher aufklingen als bei einer klassischen Resonanz durch Fremderregung. Der Effekt der Antiresonanz kann dazu benützt werden, um vorhandene Schwingungen in einem MEMS rascher und effizienter zum Abklingen zu bringen.

1 Einleitung

Mikroelektromechanische Systeme, kurz MEMS bezeichnet, sind bereits millionenfach in industriellen Produkten in Verwendung. Sie finden sich unter anderem als Sensoren in Airbag-Systemen, in den Aktuatoren von Druckern aber auch zur Signalfilterung in mobilen Kommunikationsgeräten. Wegen des aufwändigen und technologisch sehr anspruchsvollen Herstellungsprozesses von MEMS sind die mechanischen Strukturen gegenwärtig noch sehr einfach ausgeführt. Vielfach handelt es sich um mikroskopisch kleine elastische Balken und Stäbe, welche sich unter der Einwirkung von elektrostatischen und/oder Trägheitskräften bewegen und schwingen. Diese mechanischen Strukturen können daher mit den gleichen mechanischen Ansätzen wie auch makroskopische Strukturen beschrieben werden, wenn die Materialeigenschaften bekannt sind.

Allerdings ist zu beachten, dass die nichtlinearen Eigenschaften von MEMS wesentlich stärker hervortreten als bei ähnlichen Strukturen der Makro-Welt. So sind z.B. geometrische Nichtlinearitäten zu beachten, aber auch die häufig auf Basis von elektrostatischen Kräften wirkenden Antriebe in solchen MEMS sind inherent nichtlinear. Deshalb sind lineare Modelle nur beschränkt verwendbar und müssen damit

erzielte Ergebnisse stets anhand von aufwändigeren nicht-linearen Modellen überprüft werden.

Eine weitere Besonderheit von MEMS besteht darin, dass diese häufig im Vakuum operieren und daher die Dämpfung von Schwingungen primär auf der Materialdämpfung basiert und daher sehr schwach ist. Darauf ist bei der Auslegung solcher Systeme zu achten und das Abklingverhalten von beabsichtigten wie auch von ungewollten Strukturschwingungen ist stets im Auge zu behalten.

2 Parametererregung

Die unmittelbare Kombination von elektrischen und mechanischen Strukturen in MEMS erlaubt es aber auch, Effekte zu realisieren und auch zu nutzen, die in der makroskopischen Welt nur sehr beschränkt oder nur mit hohem Aufwand erzielbar sind. Ein Beispiel dafür stellt die sogenannte Parametererregung dar. Damit wird eine Art der Erregung bezeichnet, bei der ein Systemparameter eine zeitperiodische Größe darstellt. Ist z.B. ein Steifigkeitsparameter eine harmonische Funktion der Zeit, dann spricht man von einer Steifigkeits-Parametererregung (PE).

Wie schon die hinlänglich bekannte Mathieu-Gleichung zeigt, besitzen Systeme mit zeitperiodischen Parametern wesentlich andere Eigenschaften

als konventionelle Systeme. In [1] findet man eine umfangreiche Einführung zu diesem Thema. Für das Verständnis der folgenden Abschnitte ist es wesentlich zu wissen, dass parametererregte Systeme instabil werden können, wenn die Frequenz der Parametererregung in der Nähe einer sogenannten „Parameterresonanz“ liegt. Ein solcherart betriebenes System erfährt progressiv ansteigende Schwingungsamplituden, welche erst durch nichtlineare Effekte beschränkt werden. In vielen Fällen sind solche Resonanzen eine unerwünschte, ggf. auch gefährliche Eigenschaft von PE-Systemen.

In besonderen Fällen ist man aber auch am raschen Anwachsen von Schwingungen interessiert, z.B. dann wenn MEMS zur Signalfilterung verwendet werden und man ein rasch einschwingendes, scharf begrenztes Bandpassfilter realisieren möchte. Literaturzitat [2] verweist auf eine der jüngeren Arbeiten in der vorgeschlagen wird, ein MEMS mit einem mechanischen Freiheitsgrad in einer Parameter-Instabilität zu betreiben, anstatt wie üblich bei einer konventionellen Resonanz. Der Vorteil dieser Anordnung besteht im exponentiellen Anwachsen der Schwingungen bei Parametererregung, statt des vergleichsweise trägen linearen Amplitudenanstieges bei Fremderregung.

In MEMS lässt sich eine Parametererregung relativ einfach durch eine periodische Spannung am Eingang eines elektrostatischen Aktuators realisieren. Allerdings ist zu beachten, dass die elektrostatischen Kräfte auch nichtlineare Komponenten enthalten und daher eine lineare Modellbildung nur eingeschränkt verwendbar ist. Im folgenden Abschnitt werden die Modellgleichungen eines solchen Systems vorgestellt.

3 Modellbildung

In [2] und verwandten Arbeiten wurden MEMS mit einem Freiheitsgrad untersucht, also hinsichtlich der Komplexität ein Standardsystem. Nun sind aber bestimmte Phänomene nur mit Mehr-Freiheitsgrad-Systemen erzielbar. Deshalb wird hier von einem 2-FG-System ausgegangen. Die Frage nach der produktionstechnischen Realisierung stellt sich insofern nicht, als es schon jetzt möglich ist, komplexere Strukturen herzustellen als derzeit üblich. Sobald daher Bedarf und Nutzen solcher Strukturen evident ist, ist es nur eine Frage der Zeit, bis sie auch realisiert werden.

In der Arbeit von Welte [3] wurde ein technisch realisierbares 2FG-MEMS vorgestellt und seine dynamischen Eigenschaften untersucht und optimiert. Die Abbildung 1 zeigt eine schematische Skizze eines 2-FG-MEMS mit Bezeichnung der Systemkomponenten. Abbildung 2 stellt das mechanische Ersatzmodell dar.

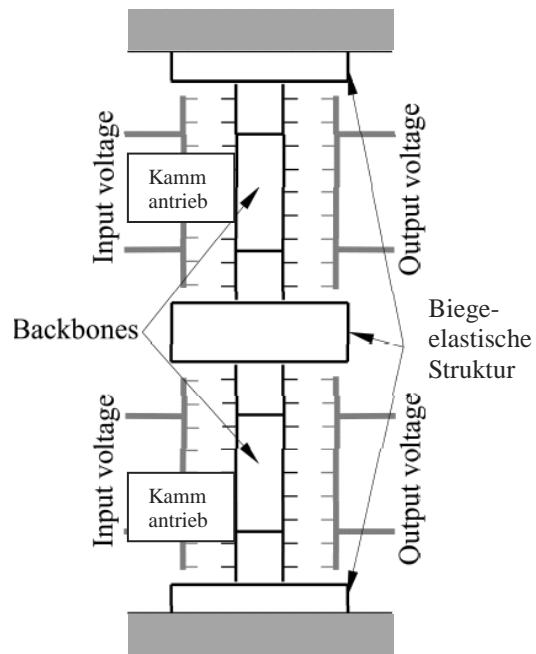


Abbildung 1. Schema eines 2-FG-MEMS, s.a. [4].

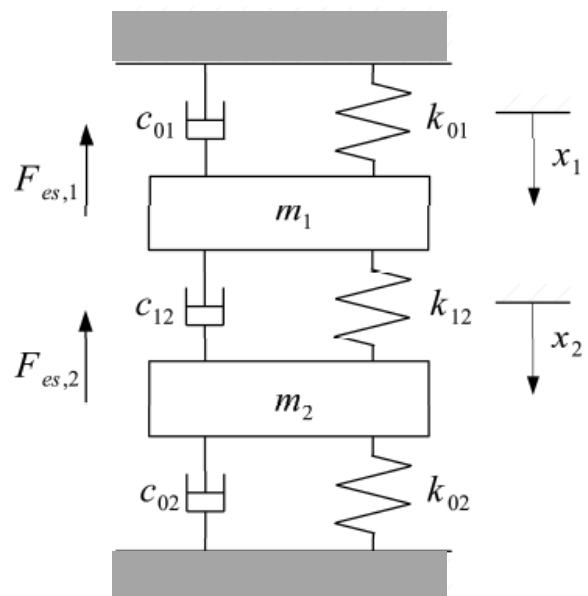


Abbildung 2. Mechanisches Ersatzmodell eines 2-FG-MEMS, siehe auch [3] und [4].

Die systembeschreibenden Modellgleichungen sind nichtlineare Matrizen-Differentialgleichungen

$$x'' + C x' + F_{es}(x, \tau) + K(x^2)x = 0 \quad (1)$$

mit dem dimensionslosen Lagevektor $x = [x_1, x_2]^T$, der Ableitung nach einer dimensionslosen Zeit $(\tau)' = d/d\tau$, der (2x2)-Dämpfungsmatrix C und der nichtlinearen (2x2)-Steifigkeitsmatrix $K(x^2)$. Die vollständige Angabe der beiden Matrizen findet man in [3] und [4].

Mit F_{es} werden die nichtlinearen elektrostatischen Kräfte bezeichnet, welche über die eingeprägte und zeitperiodische elektrische Spannung V_A erzeugt werden:

$$F_{es} = V_A^2 (1 + \cos(\eta\tau)) \times \begin{bmatrix} \lambda_1 + \lambda_3 x_1^2 & 0 \\ 0 & \alpha(\lambda_1 + \lambda_3 x_2^2) \end{bmatrix} x \quad (2)$$

Die Parametererregung tritt hier als Cosinus-Funktion in Erscheinung. Die PE-Frequenz ist mit η bezeichnet. Die Amplitude ist durch die Spannung V_A gegeben. Hinsichtlich der physikalischen Bedeutung der einzelnen Parameter sowie für realistische Wertebereiche wird auf die Literatur [4] verwiesen.

4 Stabilität des parametererregten 2-FG - MEMS

Systeme mit zeit-periodischen Koeffizienten können instabiles Verhalten aufweisen. Gemäß [1] treten primäre Parameterresonanzen auf, wenn die PE-Frequenz η

$$\eta = \frac{2\Omega_i}{n} \quad i, n = 1, 2, 3, \dots \quad (3)$$

beträgt, wobei Ω_i die i-te Eigenkreisfrequenz des ungedämpften konventionellen Systems (also ohne PE) bezeichnet. Zusätzlich können auch sogenannte Kombinationsresonanzen

$$\eta = \frac{\Omega_i + \Omega_j}{n} \quad i, j, n = 1, 2, 3, \dots \quad (4)$$

existieren, also lineare, additive Kombinationen von Eigenkreisfrequenzen $\Omega_{i,j}$ des Systems, bei denen das System ebenfalls instabil ist.

In der wegweisenden Arbeit [5] hat Tondl erstmals gezeigt, dass die sogenannten „nicht-resonanten“ PE-Frequenzen

$$\eta = \left| \frac{\Omega_i - \Omega_j}{n} \right| \quad i, j, n = 1, 2, 3, \dots \quad (5)$$

tatsächlich zur Systemstabilisierung beitragen können. In der Nähe dieser Frequenzen, ist eine scheinbare Erhöhung der Systemdämpfung festzustellen. In [6] und [7] wurden dazu umfangreiche Studien vorgelegt, welche zeigen, dass es möglich ist, auch das Eigenschwingungsverhalten insofern positiv zu beeinflussen, als freie Schwingungen schneller abklingen, wenn eine PE mit der richtigen Frequenz im System wirksam ist.

Der physikalische Hintergrund zu diesem Effekt ist der Energietransfer, welcher durch die Parametererregung bewirkt wird, und der Schwingungsenergie zwischen schwächer und stärker gedämpften Moden periodisch verschiebt. Dadurch wird eine bessere Ausnützung des (i.d.R.) höheren Dämpfungsgrades einer höheren Mode erreicht, siehe [7], [8] und [9].

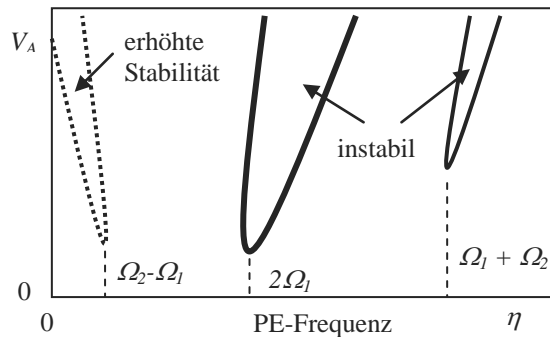


Abbildung 3. Stabilitätskarte eines 2-FG-MEMS

Die Abb. 3 zeigt exemplarisch eine Stabilitätskarte eines 2-FG-Systems, samt den Gebieten mit instabilem Verhalten (Parameterresonanz) und dem von Tondl gefundenen Bereich mit erhöhter Stabilität nahe der „Kombinationsresonanz“ $\eta = \Omega_2 - \Omega_1$.

Die Berechnung solcher Stabilitätskarten erfolgt im Allgemeinen mit dem linearisierten PE-Systemmodell auf Basis der Floquet-Theorie. Eine Überprüfung der Ergebnisse kann mittels numerischer Simulation erfolgen. Dazu müssen allerdings umfangreiche Zeitreihen berechnet werden, um verlässliche Aussagen treffen zu können.

5 Simulationsstudien und –ergebnisse

Mittels numerischer Simulation im Zeitbereich kann das nichtlineare Modell des 2-FG-MEMS ohne Einschränkungen untersucht werden. Abb. 4 zeigt die Zeitreihen der Auslenkungen x_1 und x_2 bei der Parameterresonanz $\eta = 2\Omega_1$. Neben starken Modulationen ist das Anwachsen der Ausschläge gut zu erkennen. In Abb. 5 wurde die Kombinationsresonanzfrequenz bei $\eta = \Omega_1 + \Omega_2$ ausgewählt. Es zeigt sich ebenfalls eine Vergrößerung der Ausschläge, jedoch ohne der starken Modulationen wie in Abb.4.

Schließlich wird in Abb.6 der positive Effekt der Parametererregung bei einer sog. „Antiresonanz“ gezeigt. Bei der PE-Frequenz von $\eta = \Omega_2 - \Omega_1$ ist eine starke Abnahme der Schwingungen über der Zeit festzustellen. Verglichen mit einem System ohne Parametererregung ist ein deutlicher Gewinn zufolge PE im Hinblick auf einen schnelleren Schwingungsabbau festzustellen, siehe dazu auch Literatur [10].

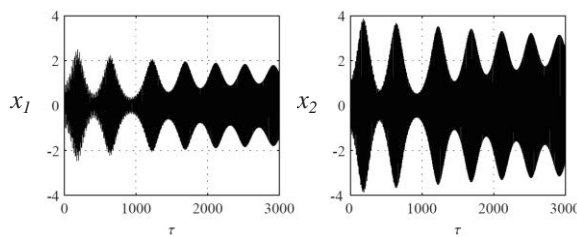


Abbildung 4. Zeitverlauf x_1, x_2 bei $\eta = 2\Omega_1$, siehe [10] für vollständige Systemdaten.

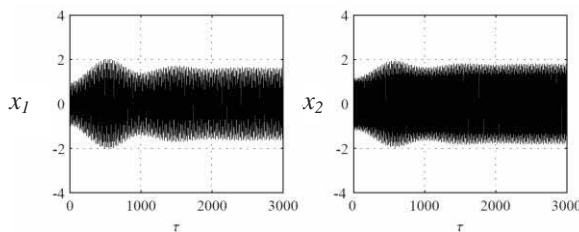


Abbildung 5. Zeitverlauf x_1, x_2 bei $\eta = \Omega_1 + \Omega_2$, siehe [10] für vollständige Systemdaten.

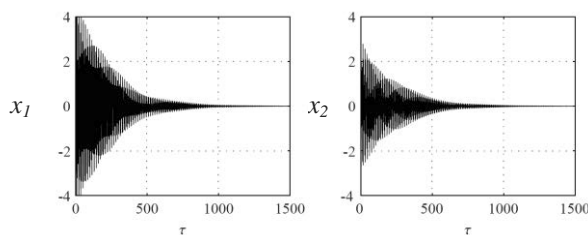


Abbildung 6. Zeitverlauf x_1, x_2 bei $\eta = \Omega_2 - \Omega_1$, siehe [10] für vollständige Systemdaten.

6 References

- [1] M. Cartmell. *Introduction to Linear, Parametric and Nonlinear Vibrations*. Chapman and Hall, London, 1990.
- [2] J. Rhoads, S.W. Shaw, K. Turner and R. Baskaran. *Generalized parametric resonance in electrostatically actuated microelectromechanical oscillators*. J. Sound and Vibration (296), S. 797-829, 2006.
- [3] J. Welte. *Parametric Excitation in Micro-Electro-Mechanical Systems*. Diplomarbeit TU-Wien, 2012.
- [4] T. Kniffka, J. Welte und H. Ecker. *Stability Analysis of a Time-periodic 2-dof MEMS Structure*. Tagungsband 9th Int. Conf. on Mathematical Problems in Eng., Aerospace and Sciences, Vienna, S. 559-566, 2012.
- [5] A. Tondl. *To the problem of quenching self-excited vibrations*. Acta Technica CSAV (43), S. 109 – 116, 1998.
- [6] H. Ecker. *Suppression of Self-Excited Vibrations in Mechanical Systems by Parametric Stiffness Excitation*. ASIM Fortschrittberichte Simulation, Vol. 11, 2005.
- [7] F. Dohnal. *Damping of mechanical vibrations by parametric excitation*. Dissertation, TU-Wien, 2005.
- [8] H. Ecker und T. Pumhössel. *Vibration Suppression and Energy Transfer by Parametric Excitation in Drive Systems*. Proc. IMechE Part C, J. Mechanical Engineering Science (226/8) S. 2000 – 2014, 2012.
- [9] T. Kniffka. *Studien zur Anti-Resonanz bei parametererregten Mehrfreiheitsgrad-Dreh-schwingern*. Diplomarbeit TU-Wien, 2012.
- [10] J. Welte, T. Kniffka und H. Ecker. *Parametric excitation in a two degree of freedom MEMS system*. Shock and Vibration (20), S. 1113–1124, 2013.

Model Based Optimization of Building Control Systems

Christoph Clauss¹ Jürgen Haufe¹ Torsten Blochwitz² Edgar Liebold³
Ullrich Hintzen⁴ Volker Klostermann⁵

¹ Fraunhofer IIS EAS, Zeunerstraße 38, D-01069 Dresden, GERMANY

² ITI GmbH, Schweriner Straße 1, D-01067 Dresden, GERMANY

³ NSC GmbH, Äußere Zwickauer Straße 8, D-08064 Zwickau, GERMANY

⁴ FASA AG, Marianne-Brandt-Straße 4, D-09112 Chemnitz, GERMANY

⁵ Provedo GmbH, Schweriner Strasse 1, Mottelerstraße 8, D-04155 Leipzig, GERMANY
{christoph.clauss, juergen.haufe}@eas.iis.fraunhofer.de

Abstract

The goal of the research project enerMAT is the reduction of energy consumption and CO₂ emissions of buildings by means of new tools for the development and planning of highly energy efficient Building Energy Management Systems (BEMS). EnerMAT will create a novel approach for simulation, optimization, and verification that will aim to design a new generation of energy-aware optimized BEMS which will allow an overall cross-trade automatic control of energy flows to maintain user comfort whilst minimizing energy consumption and CO₂ emission. The energy-aware optimization will be based on a model-based approach with an overall building system model enabling the assessment of the energy performance for different design and operation alternatives of the building automation system in interaction with the building. This system model will allow a simulation-based, energy-aware, global, dynamic, multi-criterial optimization of BEMS. In this paper, the idea, the approach, and the actual state of project research is presented.

Keywords: Building, Energy Management, Green Building Library, FMI, PSO Optimization

1 Introduction

As yet buildings still require about 40% of the primary energy consumption [1]. Apart from industrial buildings the residential buildings with their main components heating, cooling, water heating, lighting as well as home and personal electronic equipment are major consumers.

The reduction of the primary energy consumption by a further promotion of passive measures such as the insulation of passive houses yields no positive effect on the overall energy balance in case the re-

duction is compared to the expenditure in manufacturing.

There is a strong need for novel flexible approaches for the new generation of Building Energy Management Systems (BEMS) that will consider the building construction as well as building usage profiles as a whole for a specific installation [2]. Future concepts for energy optimized buildings have to incorporate a holistic and integral energy management of all installed energy sources, storages, and consumers which are specially designed for their future application. For a wide building user acceptance not only the required energy has to be minimized but also the quality of life, comfort and safety have to be increased.

The design of such a concept will demand a very high degree of automation. Market analysis [3] indicates that major obstacles on the implementation of this principle are the lack of cross-trade device communication, missing or confusing options for configuring the overall system as well as a lack of centralized, cross trade optimization- and control strategies and of the interface management. Furthermore, there are considerable deficits on the availability of design-, test- and commissioning tools for the optimization of the energy flow which are required for a breakthrough in the field of building automation design.

For future concepts of energy minimized buildings an energy-aware optimization of the overall automation solution is required which encompasses all installed components including also physical building parameters and the general climate situation. Thus an individual and optimized overall solution is developed which utilizes all energy saving options in all deployed components.

In this paper, the enerMAT project goals are presented, as well as the way of investigation.

2 Targets

The overall goal of BEMS optimisation with respect to energy consumption is a complex task covering the following goals:

Research and development goals:

- Research on innovative methodologies for the assessment of the building energy performance for different design, planning and energy management alternatives before the building will be built or for future operational modes of existing buildings
- Optimisation of the buildings' performance in terms of energy efficiency, resource efficiency, comfort level, functionality, economic return, and lifecycle value
- Increase BEMS control program safety and reliability
- Development, validation and demonstration of a methodology suitable both for retrofitting and new constructions
- Dissemination and exploitation of the methodology

Scientific and technical objectives:

- Research on a modelling technology for a wide spectrum of building types and energy systems and subsystems in terms of energy sources, energy storages and energy consumers and their control systems
- Research on a modelling technology for building operation scenarios and profiles, desired user comfort levels, building environmental conditions and government directives to describe the long term building operation modes for all four seasons of a year
- Research on model-based assessment of the building energy performance based on the modelling methodologies for the building, its installations and control systems and its long term operation modes
- Research on a methodology for a model-based, global, cross-trade and energy-aware multi-criterial optimisation for dynamic energy flows of buildings
- Research on a verification methodology to increase the BEMS functional correctness, reliability, safety and to reduce commissioning costs
- Development of the automatic generation of energy-aware optimised and verified control program code for the BEMS

- Integration of the energy-aware optimisation environment in an existing development tool chain for building automation systems
- Demonstration of the developed solution
- Validation of the developed methodology as well as the demonstration objects
- Analysis of the economic, environment and social impact of the project

3 Approach

The aims will be reached step by step with example building systems (demonstrators) with an increasing degree of complexity.

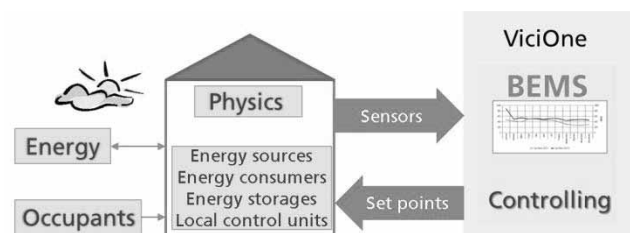


Figure 1: Holistic system model

For each system a holistic simulation model is necessary which covers all energy related aspects. Furthermore, the BEMS has to be modeled in such a way that an optimized BEMS can easily be transformed into an existing BEMS. Models of the human comfort perception are needed as well as models of the user's requirements. Completed by weather models the overall model supplies the energy consumption for different application scenarios, e.g. over a year. Using the holistic building system model an optimizer is employed to find best suitable BEMS parameters. The optimal BEMS code is afterwards checked in real BEMS of the demonstrators. In more detail the sketched approach covers the following aspects:

- Executable system model
By the creation of an executable virtual Modelica based system model (figure 1) including BEMS control models, physical building models and their interaction the system behavior is analyzed by means of a simulation at a very early design stage in terms of energy consumption and in terms of correct functionality. Thus, the building behavior for all four seasons of a year will be considered before the building or its BEMS will be implemented.

- **Operation scenarios**
For the simulation typical operation scenarios are derived from the specification. These defined scenarios are a generalization of commonly used test vectors. Operation scenarios incorporate user interactions, government energy directives, desired comfort levels, building usage profiles and environmental conditions (e.g. weather conditions).
- **Full operating-cycle energy-aware optimization**
Through the support of simulation a valuable optimization potential in concept studies will be unlocked. The executable system model and the pre-defined operating scenarios allow a simulation-based energy aware global optimization of both the BEMS and also physical building parameters.
- **Verification and virtual pre-commissioning**
The system's behavior will be validated before the installation of the automation system by a joint simulation of control model and building model in terms of a virtual pre-commissioning. For discrete components the use of formal methods (model checking) is planned. Erroneous functions that can be found with these methods can afterwards be reprogrammed and fixed early on and thus expensive reprogramming during the commissioning phase of the real system can be avoided. The robustness and design maturity of the building's control system will thus be enlarged.
- **Automatic code generation for function blocks**
To avoid manual intervention during the code generation and to ensure the consistency of control model and control program of the target BEMS program code is generated directly from the verified and optimized control model (open-loop or closed-loop controller).
- **Behavioral test in error and exceptional cases**
The behavior of the automation system cannot be tested in the real building for every possible error or exceptional case. This is because the environment conditions cannot simply be set to a certain situation (e.g. special weather conditions) or because the building or its components (lighting protection, water protection) could be damaged. Such important functions of the control system can now be tested on the building model connected to the real BEMS.
- **Flexible development and planning environment**

The developed tools will be integrated into a common design flow which consists of model libraries (buildings and BEMS models), interfaces for integration of new models, tools, and control systems.

4 Demonstrators

Three demonstrators will be used to validate and evaluate the benefits of the technologies developed both in the commercial and in the residential sectors. In the commercial sector, energy consumption is high and, as a consequence, an efficient management of resources will result in savings in energy use and CO₂ emissions. On the other hand, energy consumption in residential buildings may be low but the impact is high because the benefits of an efficient energy management are spread to the whole population. The enerMAT project incorporates the following three pilot rooms/buildings for validation:

- A single **office room** (figure 2) within an office building in Dresden. The room is only sometimes used for meetings. This promises energy savings due to changing requirements.
- A modern **office building** in Chemnitz. It is a reconstructed building with two floors (figure 3).
- A **residential building** in Zwickau. The building includes 3 floors for living, and one underground floor for various ways of utilization (figure 4).



Figure 2: Single office room in Dresden



Figure 3: FASA office building in Chemnitz



Figure 4: Residential building in Zwickau

5 Actual State

The enerMAT project started in summer 2012. The first project year dealt with the single office room, followed by the FASA building.

5.1 Single Office Room

The single office room is a conference room for up to 20 persons. It is placed within a building and has one wall with windows. Its volume is 169 m³, and its technical devices are cooling units, radiators for heating, roller blinds, as well as halogen lamps, and neon tubes. Sensors measure the temperature at several points, and the automation system ViciOne both receives sensor values and calculates actor commands using a BEMS control algorithm, which is to be developed in an optimal way.

Modeling

Figure 5 shows an overall model view.

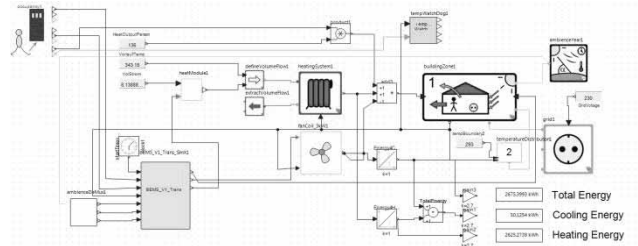


Figure 5: Office room overall model

The room as well as the radiator model are taken from the Green Building Library [4], cooling device models were developed in Modelica directly. Weather data are provided from the TRY data base for typical year runs with different characteristics (dry summer, wet summer ...). By using MSL components the energy consumption is calculated.

The occupancy is read from a text file which can be generated stochastically, or it is filled in according to known occupancies. The text file contains coming and leaving time, and the number of people. After evaluating the text file the occupancy model calculates the actual as well as the next occupancy related event.

To evaluate the user's feeling the Human Comfort Library [5] can be applied.

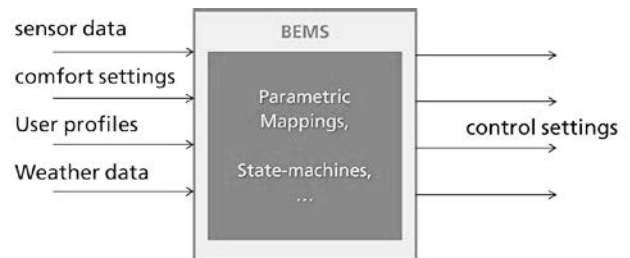


Figure 6: General BEMS model view

The Building Energy Management System (figure 6) analyzes several kinds of input data:

- Sensor data which are generated from the room temperature sensor as well as from a weather station outside the room. The sensors themselves are not part of the model since the temperature values can be taken from the room model as well as the weather model directly.
- Comfort settings are temperature tolerances which depend on the occupancy states. The temperature in an empty room can vary more than in an occupied room. In an advanced model state more sophisticated comfort settings are planned.
- User profiles characterize the numbers of present users due to the occupancy model. The activity

level is “usual” office work. In the example the room is differently occupied like in a “normal” week and repeated weekly.

- Weather data provide a huge number of inputs. The actual model uses the outside temperature only. Advanced models will use radiation data as well as weather forecast information.

Outputs of the BEMS model are generally control settings. In the actual single office room model the heating level (no heating or level one to five) as well as the cooling level (no cooling or level one to three) are calculated. The core of the BEMS model is as usual a state chart which was initially modeled using the corresponding Modelica 3.3 extensions. Since the necessary clock caused performance difficulties the state chart was translated into an “if-then-else” construction which reacts on relevant events instead of the clock.

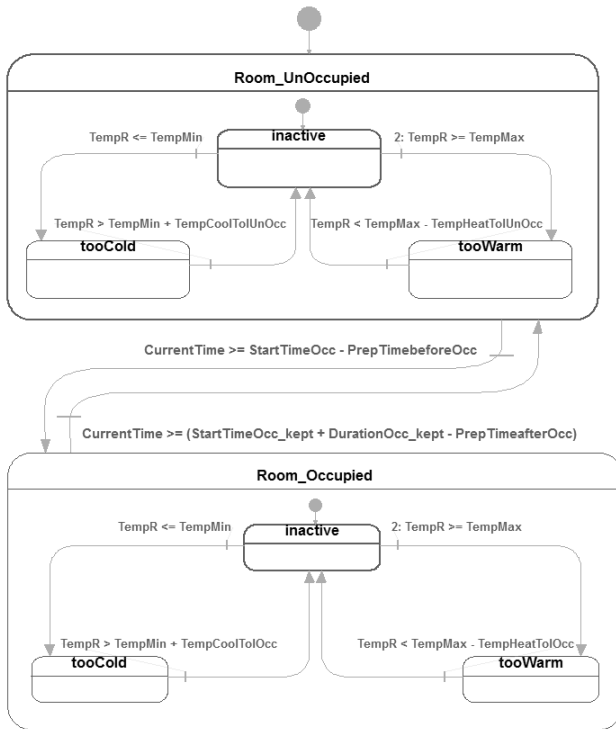


Figure 7: State Chart of the single office room BEMS

There are differently graduated BEMS models possible. The actual model with the fixed statechart structure according to figure 7 includes the following adjustable parameters which are subject of optimization:

- the duration of preparation time of the empty room before occupancy
- the heating level in the occupied room
- the heating level in the empty room
- the cooling level in the occupied room

- the cooling level in the empty room

In advanced BEMS models more parameters can be varied for optimization.

Figure 8 shows as simulation result (using SimulationX) of the complete model, the room temperature of the single office room over one year. Figure 9 shows the summarized energy consumption which increases in summer less than in winter due to the reduction in heating. A simulation of one year takes around 25 seconds CPU time using standard PC hardware.

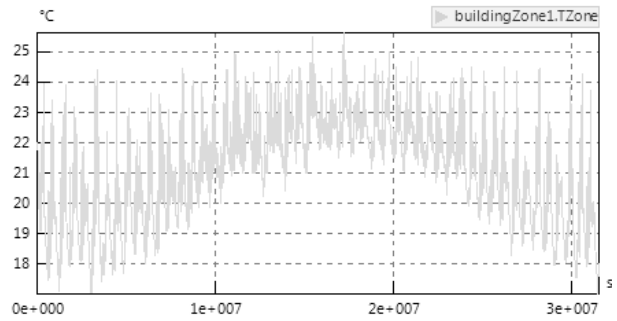


Figure 8: Office room temperature over one year

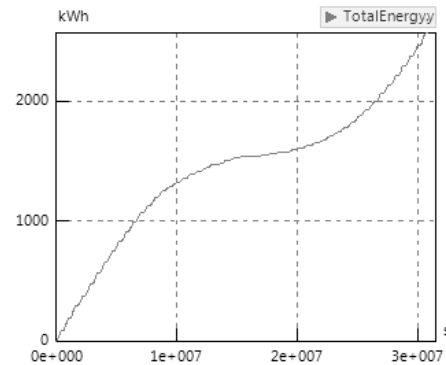


Figure 9: Office room energy consumption over one year

Optimization

The complete model of the single office room is exported as FMU by the SimulationX tool and connected to the optimization framework (figure 10). The FMU output is the total energy consumption, and inputs are the above mentioned parameters.

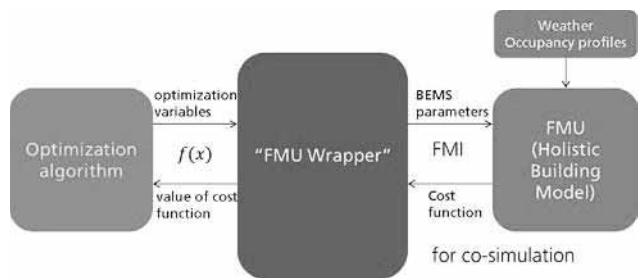


Figure 10: Optimization framework

The optimization framework is supplied with the tolerances of the parameters to be tuned as well as the simulation stop time. Out of several optimization algorithms two multi-criterial optimization algorithms (particle swarm optimization – PSO [6], multitom optimization [7]) are implemented which can operate with more than one target function. Thus the optimization of the energy consumption by tuning the above mentioned BEMS parameters is possible.

As a result the optimization of the energy consumption over one month in winter with 145 simulation runs is shown in figure 11. Starting with 240 kWh by tuning the five parameters 226 kWh are reached. This is a small saving only but it proves that the chosen method can be applied.

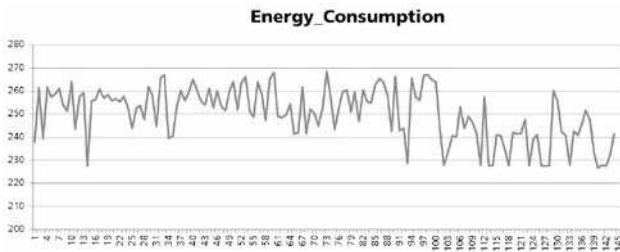


Figure 11: Optimization progress over 145 one-month-simulation runs

A more advanced BEMS with twelve parameters to be tuned was optimized using 3000 simulation over one year (figure 12) which leads to a more significant energy consumption. Such an optimization run takes up to 18 hours.

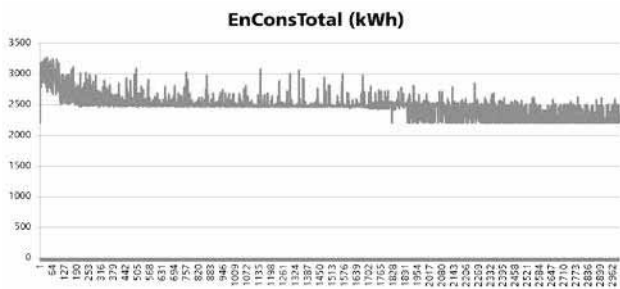


Figure 12: Optimization progress over 3000 one-year-simulation runs

These results demonstrate that the chosen way is a solid base for further investigation. A more advanced BEMS is under development which takes into account solar radiation as well as lighting. The actual optimization examples still suffer from not sufficiently modeled constraints, e.g. the compliance with temperature tolerances.

ViciOne [8] is planned to be the BEMS framework operating in all three demonstrators. It is installed at

each demonstrator. The way of transforming BEMS control code to the real BEMS ViciOne is part of the research project. As an example figure 13 shows the user communication screen of ViciOne.



Figure 13: Example screen of ViciOne

5.2 FASA Office Building

The FASA office building obtains its heating energy from solar radiation via solar collectors. The energy is stored in a water tank within the building. Floor heating transfers energy from the storage tank to the offices in a controlled way. The solar radiation covers the energy demand by more than 90 percent. The remaining energy is supplied by a stove which burns wood on demand. A simplified schematic of the demonstrator is shown in figure 14.

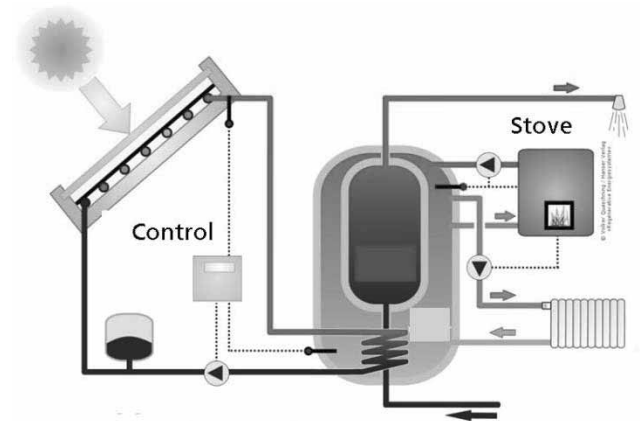


Figure 14: Office building schematic [9]

Currently various main components of the demonstrator are modeled using the Green Building Library: the solar collector, the water tank, and the supplying loop. The heating loop, the stove, the whole building models as well as the controller are still in a rudimentary stage.

The optimization target is a controller which maximizes the solar coverage. Besides control optimization the presented approach allows the optimization of

design parameters of both building and devices. Using the basic FASA demonstrator model the following parameters were changed during optimization:

- area of the collector
- direction of the collector
- volume of the heat storage

The aims of optimization are

- maximum solar coverage
- minimal costs

Figure 15 shows the costs over ten years depending on the area of the solar collector as well as on the volume of the heat storage tank.

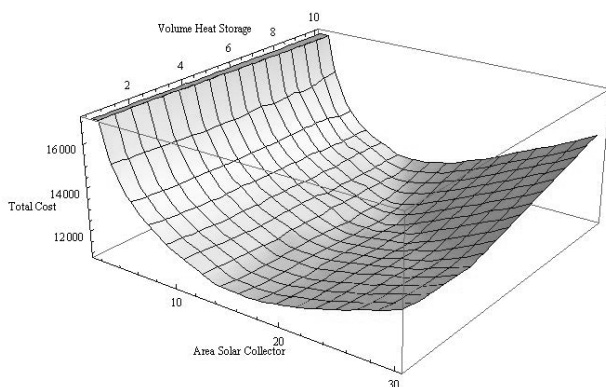


Figure 15: Costs over ten years depending [10]

The presented result is a first principal test result. The method will be applied to the FASA office building. Since the result shown is based on a simplified model results have to be verified with a more realistic model. However, the results show that the chosen approach is a promising one.

6 Conclusion


In this paper the idea of saving energy by optimization of building energy management systems is presented. The research project enerMAT is founded on that idea. Based on an overall simulation model of the building including building architecture, building services, requirements, and a BEMS model the energy consumption is simulated using different weather courses of years. This allows the optimization of BEMS which can be afterwards virtually commissioned. Real commissioning and BEMS operation will be implemented and tested on three different demonstrators.

The project's aims are described in detail, and the actual state of the project is characterized.

Challenges of further investigation are:

- The design of BEMS which are suitable for a class of buildings. The BEMS parameters to be optimized should be invariant with regard to different scenarios, and different application cases.
- The development of dedicated models. The models must be as detailed as required to cover the necessary phenomena, but also rough and fast to allow high-performance simulations.
- The method has to be proven at more complex objects.

Acknowledgement

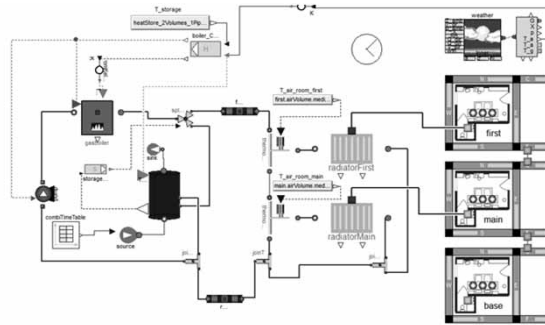
<p>The research is funded by</p>	<p>Gefördert durch:</p>  <p>Bundesministerium für Wirtschaft und Technologie</p> <p>aufgrund eines Beschlusses des Deutschen Bundestages</p> <p>Förderkennzeichen: 03ET1084</p>
----------------------------------	--

References

- [1] EU Energy and transport in figures, statistical pocket book 2007/2008
- [2] SIEMENS: Der Einfluss der Gebäudeautomationsfunktionen auf die Energieeffizienz von Gebäuden. Siemens Building Technologies GmbH&Co. oHG 2009.
- [3] Frost&Sullivan, 2009: Building Automation Systems: The Future Role of Open Protocols in Europe, M0F7-19
- [4] Schwan, T.; Unger, R.; Bäker, B.; Mikoleit, B.; Kehrner, C.; Rodemann, T.: „Green Building“–Modelling renewable building energy systems and electric mobility concepts using Modelica. 9th International Modelica Conference, Munich, September 2012.
- [5] Michaelsen, B.; Eiden, J.: Human Comfort Modelica Library, Thermal Comfort in Buildings and Mobile Applications. 7th International Modelica Conference, Como, Italy, Sept. 20-22, 2009.

- [6] Kennedy J., E. R.: Particle swarm optimization. IEEE International Conference on Neural Networks, (S. 1942 - 1948). Bur. of Labor Stat., Washington, DC, USA, 1995.
- [7] Rose, H.: Evolutionäre Strategien und multitome Optimierung. Diss. HU Berlin, 1998.
- [8] ViciOne – die Revolution in der Gebäudeautomation.
<http://www.acx-gmbh.de/de/home-building-automation/index.html>
- [9] V.Quaschnig: "Regenerative Energiesysteme". Hanser Verlag, 2013
- [10] http://tu-dresden.de/die_tu_dresden/fakultaeten/fakultaet_architektur/ibk/institute/events/2013_entool/601_BEMSOptimization-MatthiasFranke.pdf

Dynamische Simulation von Gebäuden und Anlagen für mehr Komfort und mehr Effizienz



Sven Rutkowski

21.02.2014

ASIM 2014 - Reutlingen

1

Übersicht

- Motivation für dynamische Simulation
- Gebäudemodellierung mit Modelica
- Anlagenmodellierung mit Modelica
- Detaillierte Modellierung von Komponenten

21.02.2014

ASIM 2014 - Reutlingen

2

Motivation

- > Ein Gebäude und die Heiz- oder Klimaanlage ist ein dynamisches System, weil sich die Randbedingungen und inneren Lasten ändern

Aus Sicht des Komponentenherstellers:

- > Komponenten von Heiz- und Klimasystemen werden daher auch immer unter dynamischen Randbedingungen getestet, um z.B. Regler zu prüfen
- > Dieser Versuchsaufwand kann durch den Einsatz von dynamischen Simulationswerkzeugen reduziert werden

Aus Sicht des Planers/Betreibers:

- > Die Funktion der Gesamtanlage ist am jeweiligen Objekt zu prüfen
- > Die Energiekosten hängen maßgeblich vom Teillastverhalten ab, und damit nicht vom besten Wirkungsgrad im stationären Betrieb sondern vom Nutzungsgrad
- > Eine Optimierung der Energiekosten sollte Komfortansprüche berücksichtigen

Möglichkeiten der dynamischen Simulation

Aus Sicht des Komponentenherstellers:

- > Modellierung der Trägheit(en) einer Komponente und Untersuchung von Regelungsstrategien oder -einstellung (HIL) im Systemzusammenhang
- > Interne Modellierung des Systems zur Verbesserung der Performance unter dynamischen Randbedingungen

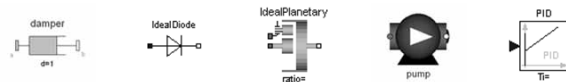
Aus Sicht des Planers/Betreibers:

- > Untersuchung von Lastszenarien, Ganz-Jahressimulationen zur Vorhersage des Energiebedarfs (Auswahl von Komponenten)
- > Optimierung der Anlagenfahrweise, um die Energiekosten und den Komfort zu verbessern
- > Bekämpfung von dynamischen Problemstellungen, die sich aus der Anlagensteuerung ergeben

Was bietet der Modelica-Ansatz?



- > Technologisch, fortschrittliche Modellierungskonzepte, die mit einer übersichtlichen Sprachsyntax auskommen
- > Multi-Domain-Ansatz (Thermodynamik, Elektrotechnik, Mechanik, Regelung)
- > Viele freie Bibliotheken als Startpunkt (Modelica Standard Library, u. v. m. siehe <http://www.modelica.org>)
- > Kommerzielle Lösungen von verschiedenen Anbietern
- > Schnittstellen zu anderen Simulationswerkzeugen (Functional-Mockup-Interface)



21.02.2014

ASIM 2014 - Reutlingen

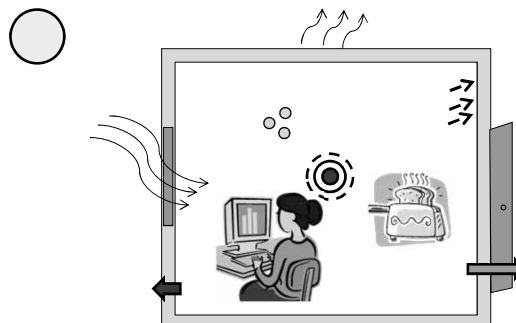
5

Physikalische Gebäudemodellierung

Effekte, die durch das HumanComfort Gebäudemodell berücksichtigt werden:



- > Solare Einstrahlung
- > Langwellige Strahlung
- > Konvektiver Wärmeübergang
- > Trägheit von Materialien und Luft
- > Wärmeleitung durch Materialien
- > Innere Lasten (Personen, Abwärme von Geräten)
- > Kompressible Luftströmung (mit CO₂)
- > Kondensation an kalten Wänden



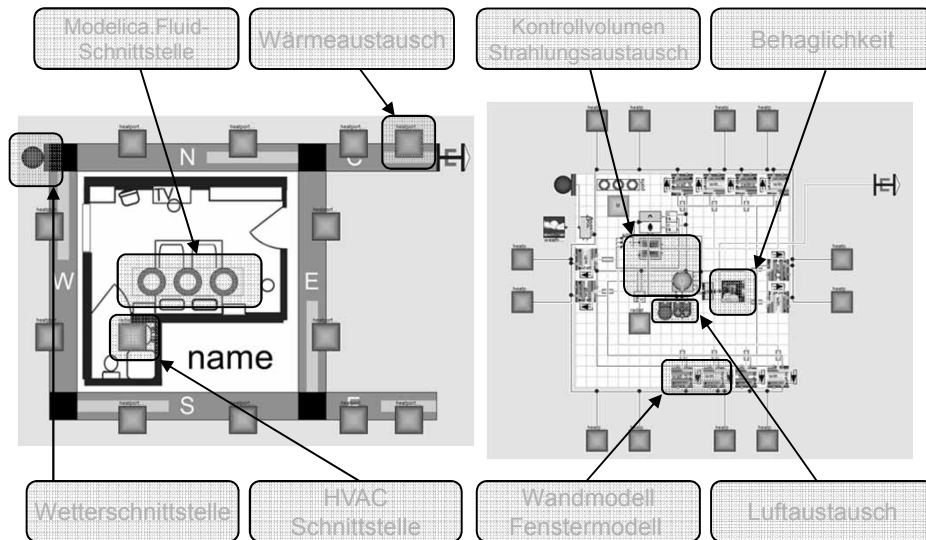
Verfügbar für Dymola & MSL 3.2

21.02.2014

ASIM 2014 - Reutlingen

6

HumanComfort Gebäudemodell



21.02.2014

ASIM 2014 - Reutlingen

7

PMV (Predicted Mean Vote)

Vorausgesetztes mittleres Votum PMV nach DIN EN ISO 7730

Das PMV ist ein Index, der den Durchschnittswert für die Klimabeurteilung durch eine große Personengruppe anhand folgender **7-stufiger Klimabeurteilungsskala** vorhersagt. Die sieben Einflussgrößen des PMVs sind in thermische und persönliche Faktoren eingeteilt.

Im Idealfall sollte der der PMV zwischen -0.5 und +0.5 liegen.

Thermische Faktoren

Mittlere Strahlungstemperatur
Lufttemperatur
Luftfeuchtigkeit
Luftgeschwindigkeit



Persönliche Faktoren

Bekleidung
Aktivität
Wirksame Leistung



+ 3	heiß
+ 2	warm
+ 1	etwas warm
0	neutral
- 1	etwas kühl
- 2	kühl
- 3	kalt

21.02.2014

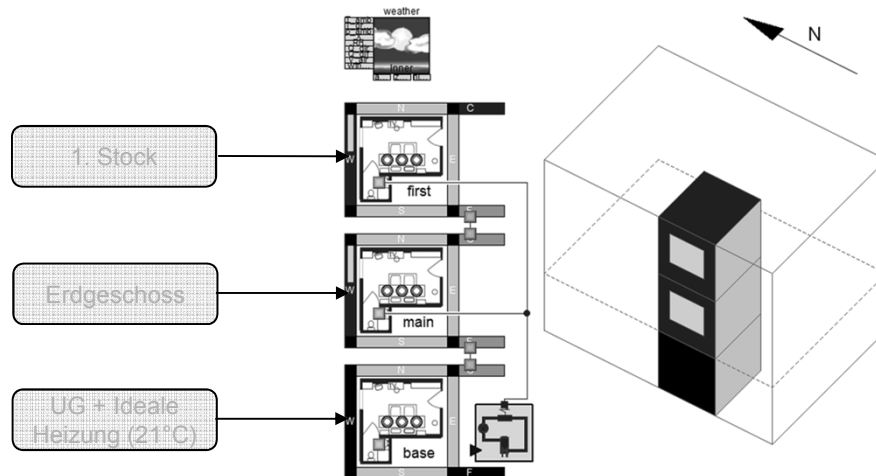
ASIM 2014 - Reutlingen

8

Simulation der Heizlast von Bürogebäuden



Simulation time	CPUTime	
19.612.800 sec to 2.0217.600 sec 16.08.2010 to 22.08.2010 (week)	~30 sec	1 hour = 3600 sec 1 day = 86400 sec 1 week = 604.800 sec 30 days = 2.592.000 sec 1 year = 31.536.000 sec

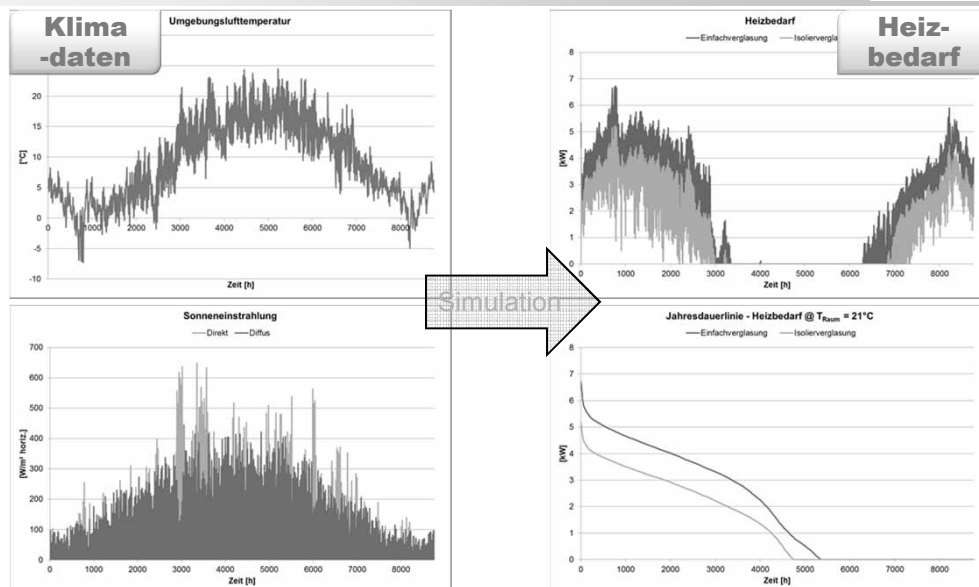


21.02.2014

ASIM 2014 - Reutlingen

9

Simulation der Heizlast von Bürogebäuden



21.02.2014

ASIM 2014 - Reutlingen

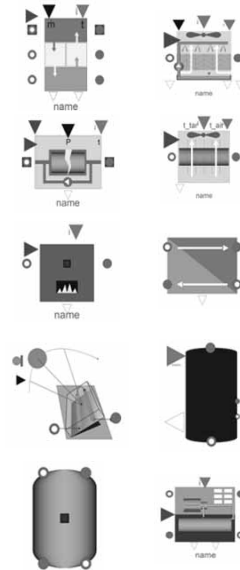
10

Modellierung der Gebäudetechnik

- > Absorptions- und Kaltdampfkomppressionskältemaschinen
- > Kühltürme (Trocken- oder Feuchtbetrieb)
- > Heizkessel und BHKW
- > (Gas-)Wärmepumpen
- > Wärmetauscher
- > Radiatoren, Fußbodenheizung (kompatibel mit HumanComfort Library)
- > Wärme- und Kälteverbraucher
- > Solarkollektoren
- > Puffer- und Hygienespeicher
- > Rohrleitungen, Ventile, Verteiler
- > Pumpen und Gebläse
- > Regler
- > Automatische energetische Bilanzierung der Gesamtanlage
- > Unterschiedliche flüssige Medien



Verfügbar für
Dymola & MSL 3.2



21.02.2014

ASIM 2014 - Reutlingen

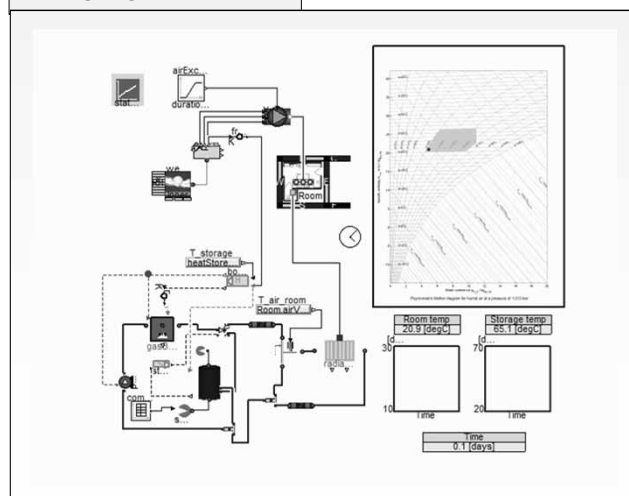
11

HVAC Anlagenmodelle



- > Schnellste dynamische Anlagensimulation (1 Jahr in wenigen Minuten)
- > Ansatz erlaubt systematische Parameteroptimierung durch mathematische Algorithmen (z. B. XRG's ModelOpt Tool)
- > Kompatibel mit HumanComfort Library zur gekoppelten Simulation mit einem Gebäudemodell

HVAC MODEL

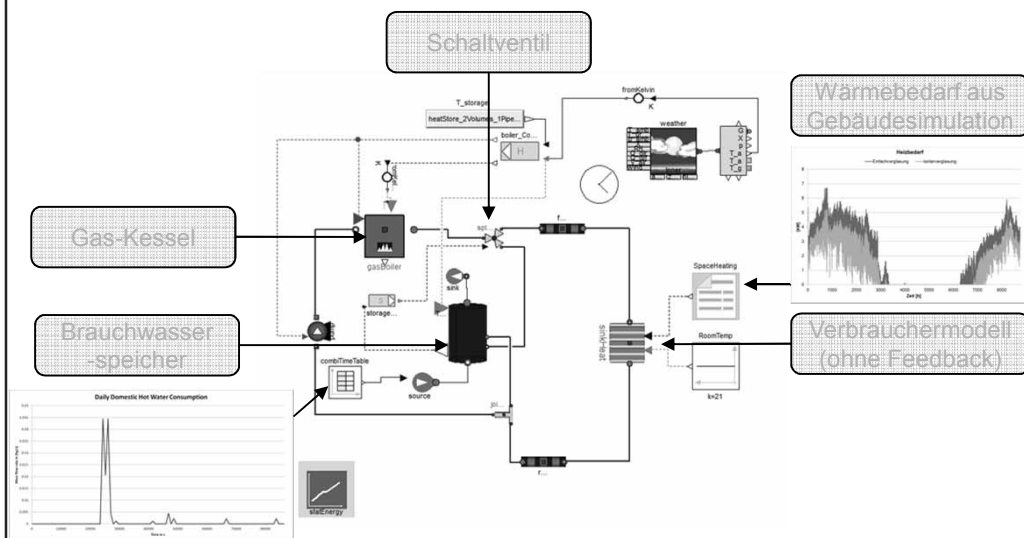


21.02.2014

ASIM 2014 - Reutlingen

12

Standalone Anlagenmodell

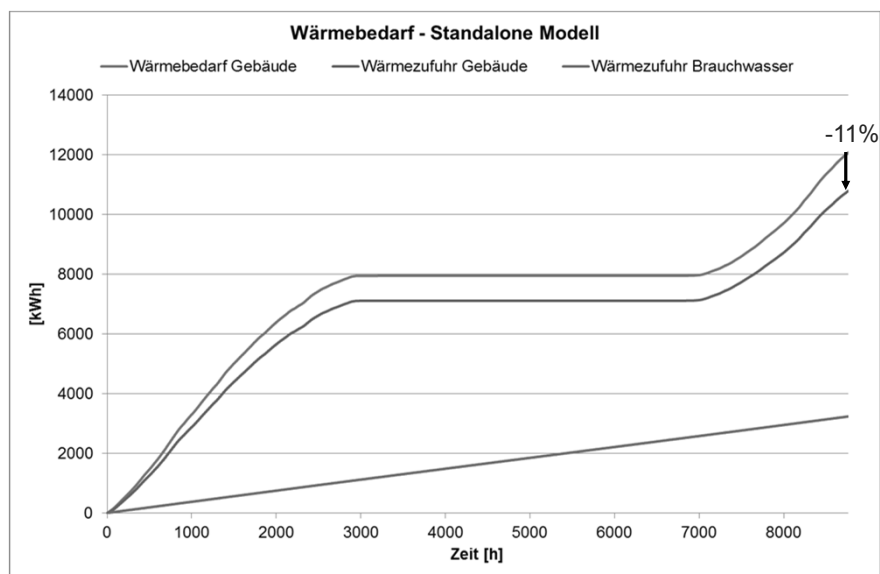


21.02.2014

ASIM 2014 - Reutlingen

13

Ergebnis Standalone Modell

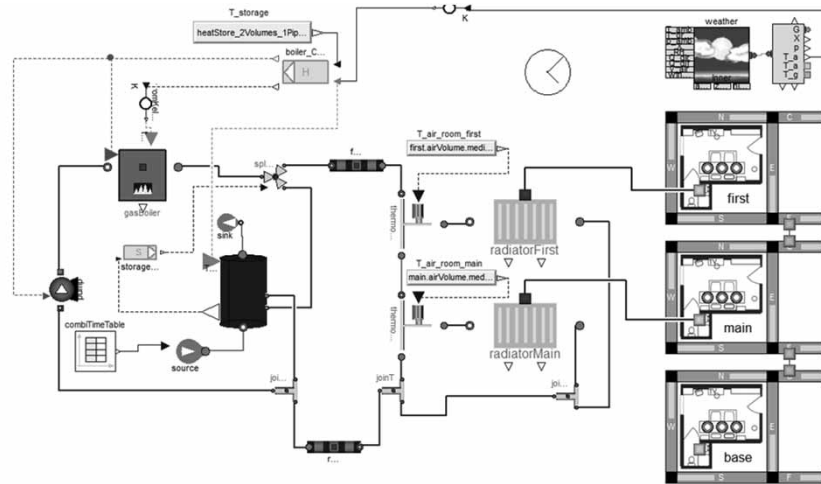


21.02.2014

ASIM 2014 - Reutlingen

14

Gesamtmodell

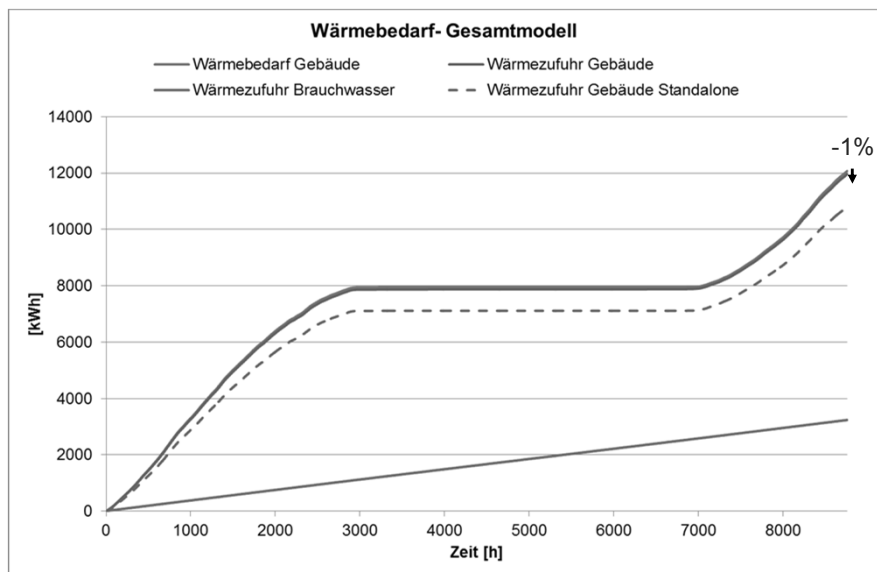


21.02.2014

ASIM 2014 - Reutlingen

15

Ergebnisse Gesamtmodell

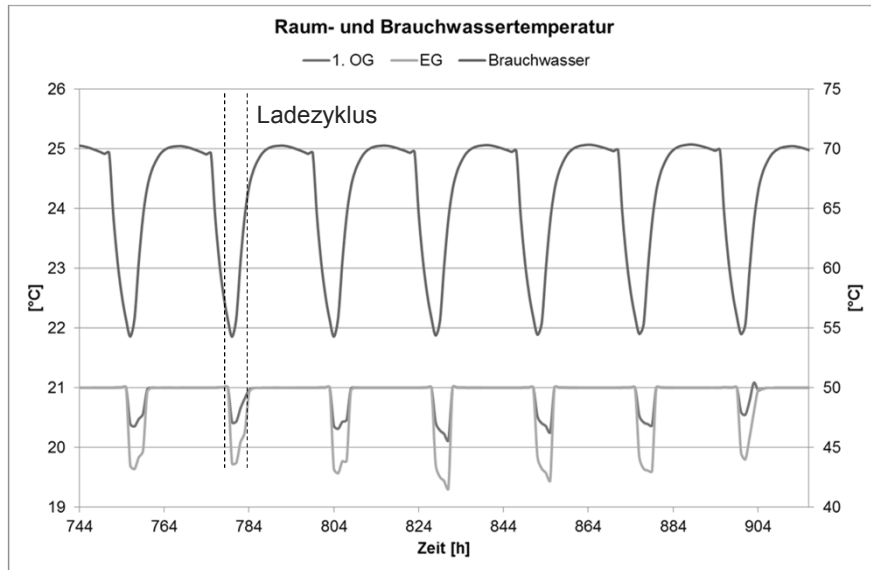


21.02.2014

ASIM 2014 - Reutlingen

16

Ergebnisse Gesamtmodell

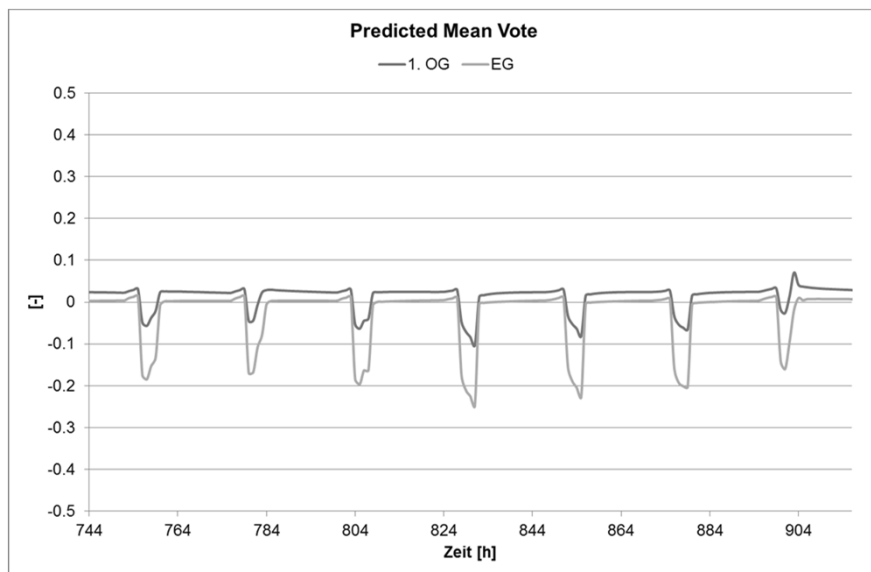


21.02.2014

ASIM 2014 - Reutlingen

17

Ergebnisse Gesamtmodell

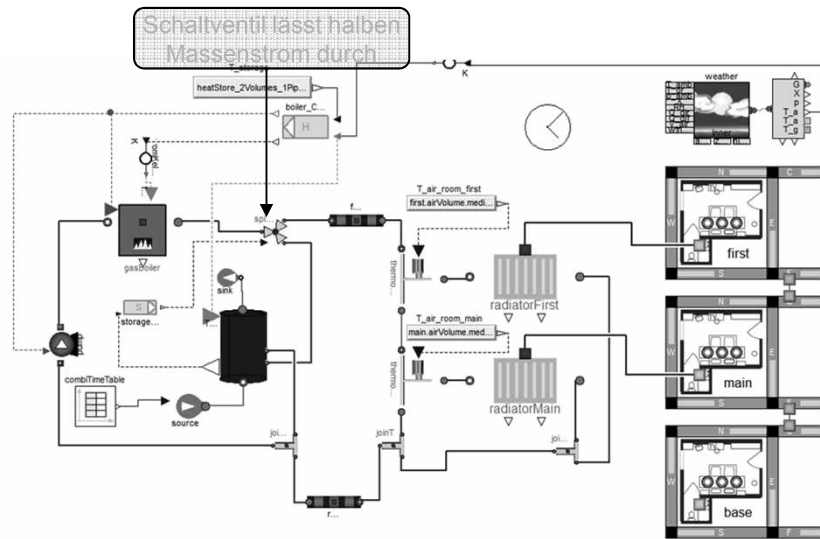


21.02.2014

ASIM 2014 - Reutlingen

18

Gesamtmodell - modifiziert

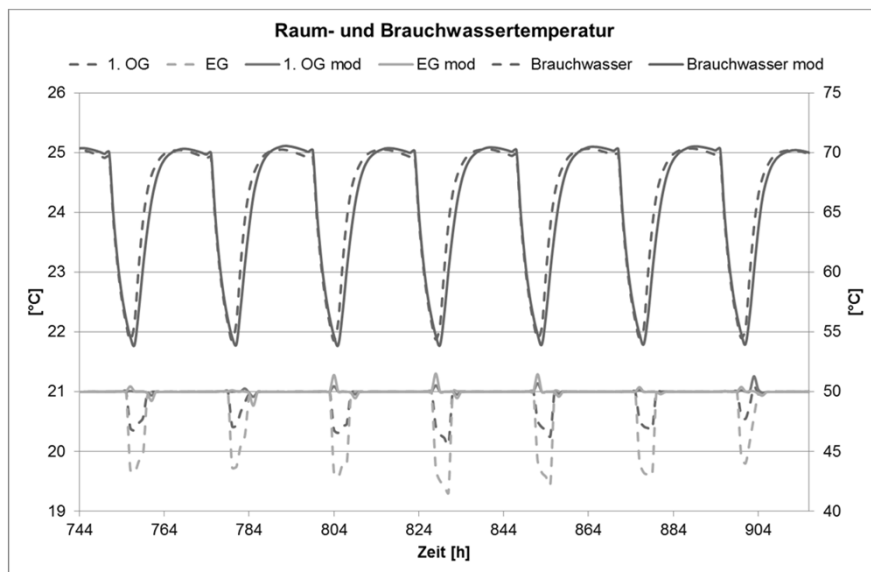


21.02.2014

ASIM 2014 - Reutlingen

19

Ergebnisse modifiziertes Gesamtmodell

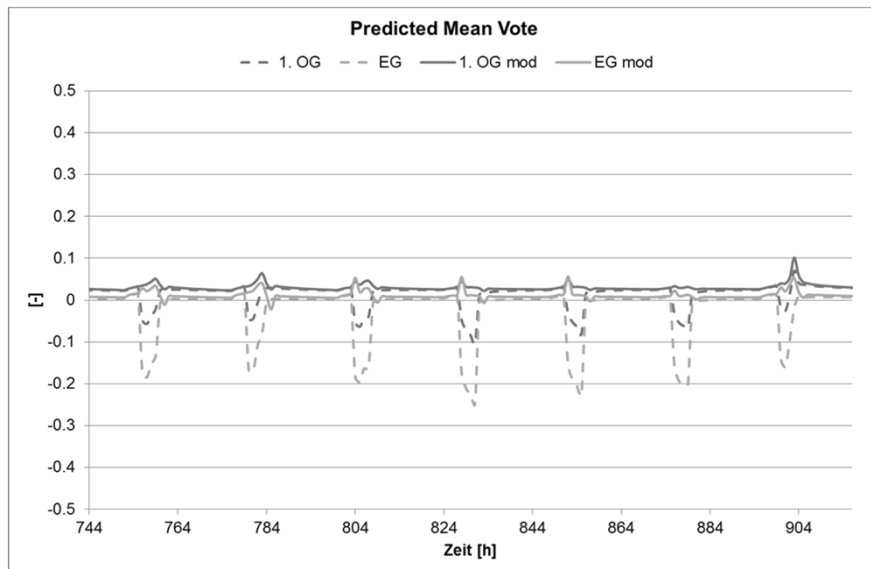


21.02.2014

ASIM 2014 - Reutlingen

20

Ergebnisse modifiziertes Gesamtmodell



21.02.2014

ASIM 2014 - Reutlingen

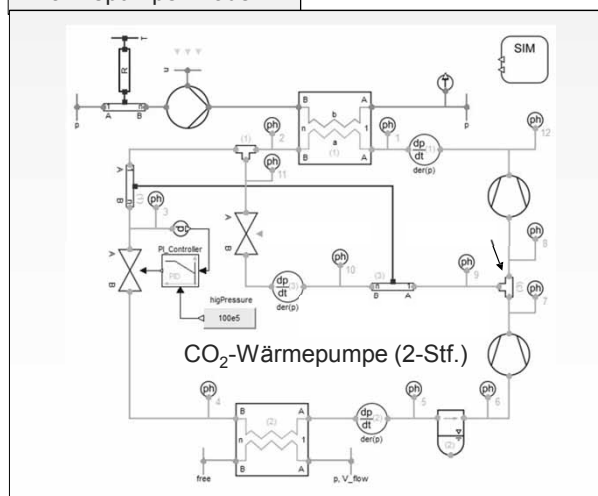
21

Detaillierte Komponentenmodelle

> Detaillierte Wärmepumpenmodellierung unter Verwendung der AirConditioning oder TIL Library (siehe Bild).

> Aber: detaillierte, physikalische Modelle sind nicht zur Durchführung von Ganzjahressimulationen geeignet. Diese Modelle können gleichwohl dazu dienen, um Kennfelder zu generieren oder das kurzzeitige Verhalten der Komponenten zu verbessern. Auch für (HIL / SIL) geeignet.

Wärmepumpenmodell



21.02.2014

ASIM 2014 - Reutlingen

22

Für weitere Informationen stehen wir gerne zur Verfügung.

XRG Simulation GmbH
Otto-Lilienthal-Str. 36
71034 Böblingen
Germany
www.xrg-simulation.de

Wir freuen uns auf Sie.

Mechatronische Entwicklung eines HiL-Prüfstands zur Erprobung einer aktiven Federung

Michael Scheele^{1,a}, Sven Jacobitz^{1,b}, Xiaobo Liu-Henke^{1,c}

¹Ostfalia Hochschule für angewandte Wissenschaften, Fakultät Maschinenbau,

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

^am.scheele@ostfalia.de, ^bs.jacobitz@ostfalia.de, ^cx.liu-henke@ostfalia.de

In diesem Beitrag wird die modellbasierte Auslegung der Funktionen eines Hardware-in-the-Loop-Prüfstands dargestellt. Der Prüfstand ermöglicht die Erprobung einer aktiven Fahrzeugfederung unter realitätsnahen Bedingungen und setzt sich aus dem *Prüfstandsmodul*, der *Echtzeithardware* und *Softwaremodulen* zusammen. Das Prüfstandsmodul besteht hauptsächlich aus einer *aktiven Fahrzeugfederung* zur Schwingungskompensation und einer Anregungseinheit zur Nachbildung der *Fahrbahnunebenheit*. Für die aktive Fahrzeugfederung wird die Regelstruktur hierarchisch in globale und lokal arbeitende Controllergruppen strukturiert. Zur Auslegung des Globalreglers wird ein Skyhook-Algorithmus angewendet. Als lokale Informationsverarbeitung dient eine 3-fache Kaskadenregelung, welche die Dynamik des einzelnen Aktors hinreichend schnell ausregeln kann. Im vorliegenden Beitrag werden die Funktionen der beiden Kernmodule *Anregungseinheit* und *aktive Fahrzeugfederung* anhand des mechatronischen Entwicklungskreislaufs sukzessive von der Modellierung über die Synthese bis hin zur Echtzeit-Realisierung mit Inbetriebnahme des Prüfstands vorgestellt. Ebenso wird die verwendete Aktorik, Sensorik und digitale Signalverarbeitung dargestellt. Der Beitrag schließt mit Zusammenfassung und Ausblick.

1 Einleitung

Im Rahmen eines Forschungsprojekts ist es das Ziel ein domänenübergreifendes Konzept zur Funktionsintegration des Fahrwerks und des Antriebsstrangs von Kraftfahrzeugen zu entwickeln. Untersucht wird dabei unter anderem die Interaktion verschiedener Aktoren im dezentralen, verteilten Fahrzeugregelsystem. Anhand dessen wird eine Strategie konzipiert, um das Potenzial zur Rekuperation durch Kooperation verteilter Aktoren höchstmöglich auszunutzen.

Als begleitende Entwicklungsmethode wird dabei die durchgängige modellbasierte Vorgehensweise aus Model-in-the-Loop (MiL)-, Software-in-the-Loop (SiL)- und Hardware-in-the-Loop (HiL)-Simulation eingesetzt.

Zur Erreichung der Ziele des Forschungsprojekts wird als Verbesserung des Fahrwerks eine *aktive Federung* fokussiert. Aus diesem Grund wird im vorliegenden Beitrag die modellbasierte Entwicklung eines HiL-Prüfstands zur Systemerprobung aufgezeigt.

2 Lösungsansatz und Entwicklungsmethodik

Im Kraftfahrzeugbereich wird konventionell zwischen Fahrzeugaufbau und Radmodul ein passives Feder-Dämpfer-System für eine definierte Relativbewegung zwischen diesen Baugruppen eingesetzt. Für die Systemauslegung einer passiven Fahrzeugfederung ist stets ein Kompromiss, mit einem Zielkonflikt zwischen Fahrsicherheit und -komfort, zu wählen, denn einerseits soll das Rad dem Straßenprofil folgen, aber gleichzeitig soll der Fahrzeugaufbau gegenüber Straßenunebenheiten isoliert werden.

Im Gegensatz zu passiven Systemen ist der wesentliche Unterschied von aktiven Systemen eine externe Energiezufuhr mit der ein verbessertes Führungs- und Störverhalten erzielt werden soll. Dementsprechend weißt die Konfiguration von aktiven Systemen, neben passiven Feder-/Dämpferelementen, Aktoren, Sensoren und eine Informationsverarbeitung mit Regelalgorithmen auf.

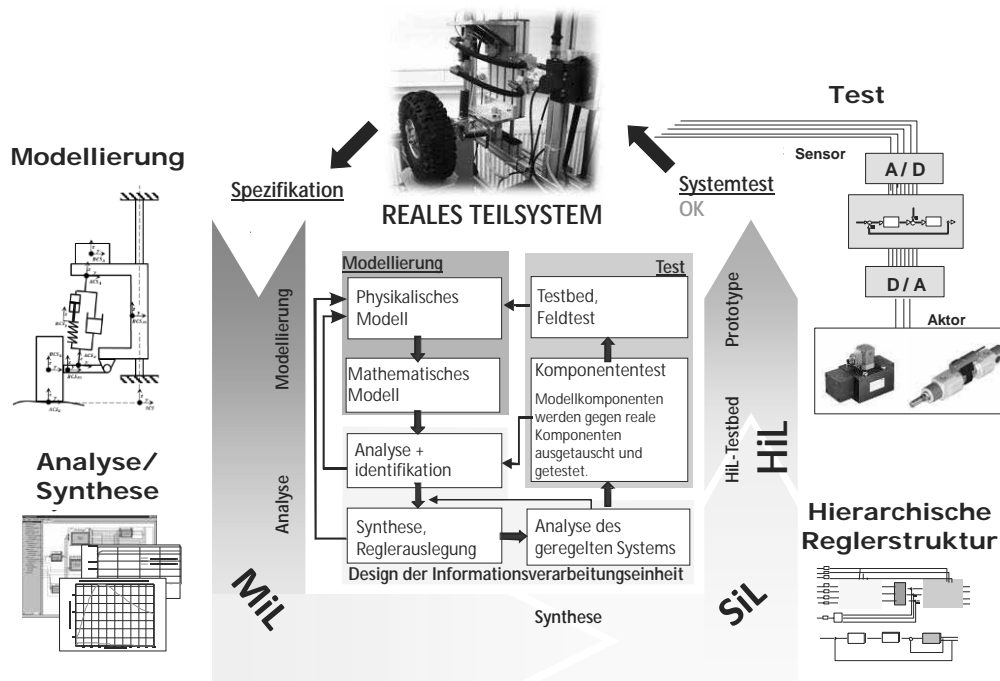


Abbildung 1. Mechatronischer Entwicklungskreislauf am Beispiel einer aktiven Federung

Zur Entwicklung eines solchen mechatronischen Systems ist eine erweiterte Entwicklungsmethodik gegenüber dem klassischen Konstruktionsprozess notwendig, um in frühen Entwurfsphasen die notwendige Dynamik der mechatronischen Funktion, neben der gestaltorientierten Konstruktion, berücksichtigen zu können.

Abbildung 1 zeigt den verwendeten Entwicklungskreislauf mittels Model-in-the-Loop (MiL)-, Software-in-the-Loop (SiL)- und Hardware-in-the-Loop (HiL)-Simulation bezogen auf den Prüfstand und die mechatronische Funktion *aktive Federung* (vgl. [1] u. [2]). Ein zu entwickelndes, reales Teilsystem wird dazu nach einer festgelegten Spezifikation, z. B. Modellierungstiefe, modelliert. Eine Analyse des passiven Systemverhaltens mit identifizierten und validierten Modellen ermöglicht anschließend die Synthese der Regleralgorithmen. Die modellbasiert ausgelegte Regelung ist anschließend im SiL-Prozess hinsichtlich der verwendeten Algorithmen und Rechenarchitektur zu optimieren. Der Ersatz von Teilkomponentenmodellen durch reale Bauteilen ermöglicht mittels HiL-Simulation eine weitergehende Optimierung der Algorithmen unter Echtzeitbedingungen.

Wie dargestellt, sind damit für die Entwicklungsmethodik validierte und identifizierte Modelle schon in frühen Entwurfsphasen notwendig. Daher dient der

Prüfstand auch zur Modellidentifikation und -validierung von theoretischen Modellen, die das Verhalten von Teilkomponenten wie Aktorik, Sensorik und Informationsverarbeitung des Funktionsmoduls beschreiben.

Anhand der dargestellten Vorgehensweise wird der HiL-Prüfstand entwickelt, welcher abschließend die Erprobung des mechatronischen Funktionsmoduls *aktive Fahrzeugfederung* unter realitätsnahen Bedingungen ermöglicht.

Die Prüfstandsentwicklung gliedert sich dazu im vorliegenden Beitrag in: Konzeption, mechatronische Entwicklung – bestehend aus Modellbildung, Analyse und Synthese – sowie konstruktiver Entwurf und Inbetriebnahme.

3 Konzeption des HiL-Prüfstands für eine aktive Fahrzeugfederung

Die Konzeption lässt sich in drei Bereiche unterteilen: *Gesamtprüfstandskonzept*, *aktives Federungskonzept*, *Anregungs- und Energieversorgungskonzept*.

3.1 Gesamtprüfstandskonzept

Das Prüfstandskonzept fokussiert zunächst die Beeinflussung des vertikalen Freiheitsgrads eines Fahrzeugs durch die *aktive Federung*. Zwischen der Federung an

den vier Rädern eines konventionellen Fahrzeugs besteht in der Regel kein wesentlicher Unterschied, so dass die Abbildung lediglich eines Federbeins, bei entsprechend reduzierter Aufbaumasse, für den vertikalen Aufbaufreiheitsgrad zweckmäßig ist (s. [3]). Mit der Betrachtung dieses isolierten Freiheitsgrads wird die Strategie verfolgt die restlichen Freiheitsgrade simulativ zu erfassen und diese mit einer übergeordneten Regelung unabhängig voneinander zu beeinflussen. Als reales System wird dabei ein Kleinfahrzeug (Quad) mit McPherson-Radaufhängung verwendet.

Das Konzept des Prüfstands umfasst außerdem flexible Konfigurations- und Erprobungsmöglichkeiten, die sich wie folgt darstellen:

- Erprobung verschiedener passiver und aktiver Fahrzeugfederungssysteme mit unterschiedlichen Fahrwerkeinstellungen wie Sturz oder Aufbaumasse
- Erprobung und Optimierung verschiedener Regleralgorithmen unter Echtzeitbedingungen
- Validierung und Identifikation von theoretischen Modellen

Als mechatronisches Gesamtsystem lässt sich der Prüfstand in *Prüfstandsmodul*, *Echtzeithardware* und *Softwaremodul* gliedern (Abbildung 2).

Das **Prüfstandsmodul** besteht aus einer *Anregungseinheit* zur Nachbildung der Fahrbahnnunebenheit und einer *aktiven Fahrzeugfederung* zur Schwingungskompensation. Der als starre Masse vereinfachte Fahrzeugaufbau ist dabei in vertikaler Richtung über eine Kugelhülsenführung frei beweglich und über einen Querlenker und die konventionelle oder aktive Federeinheit mit der Radmasse verbunden. Die Reibung der Führung stellt eine ungewollte Dämpfung des Aufbaus dar, die am realen Gesamtfahrzeug nicht auftritt. Diese prüfstandsbedingte Reibung lässt sich in der Informationsverarbeitung durch einen Offset berücksichtigen.

Die **Echtzeithardware** umfasst, neben digitalen Input/Output-Schnittstellen, diverse A/D- und D/A-Kanäle. Diese ermöglichen das Einlesen der Messsignale der Sensorik und die Ausgabe von Sollwerten für die Aktorik. Mit Hilfe von Softwaremodulen, die auf dem digitalen Signalprozessor (DSP) der Echtzeithardware ablaufen, kann der Prüfstand betrieben werden.

Die **Softwaremodule** umfassen unter anderem die Regalgorithmen, die Skalierung der Signale, Bewertungsfiler und diverse Anregungsprofile. Programmiert werden diese Module mit Hilfe von Matlab/Simulink, die anschließend über eine automatische C-Code-Generierung auf dem DSP unter Echtzeitbedingungen laufen.

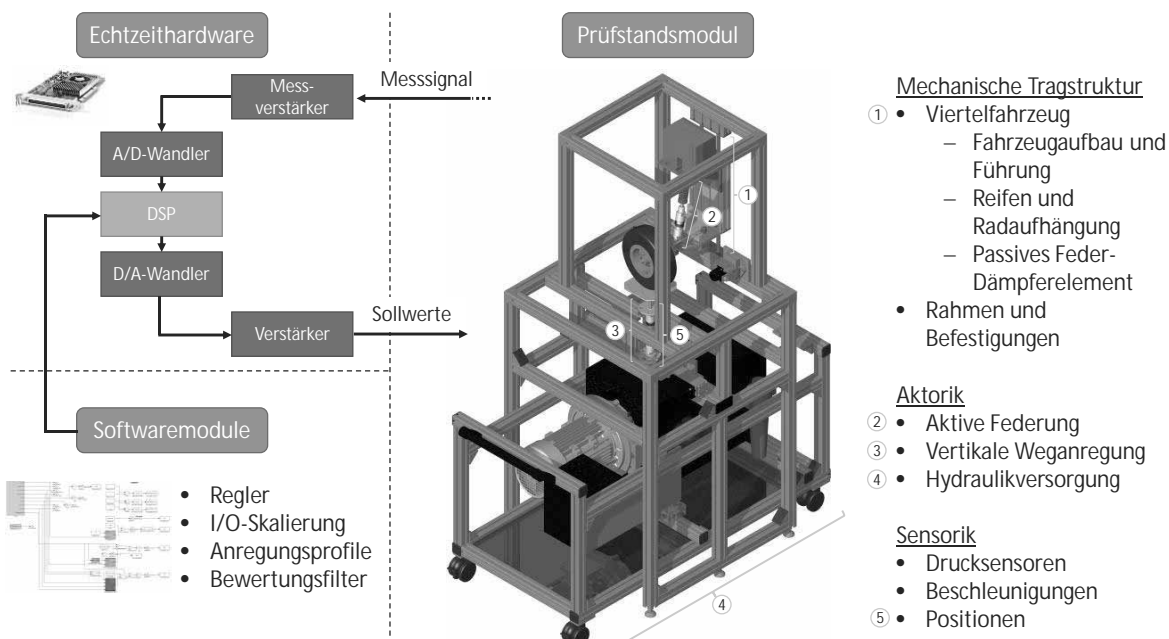


Abbildung 2. Prüfstandskonzept zur Erprobung einer konventionellen und aktiven Aufbaufederung

3.2 Konzeption der aktiven Federung

Die Konzeption der mechatronischen Funktion *aktive Federung* unterliegt der wesentlichen Anforderung, dass der Zielkonflikt zwischen Fahrkomfort und Fahr-sicherheit entschärft werden soll.

Als Maß zur Bewertung des Fahrkomforts dient die Aufbaubeschleunigung, denn die Beschleunigungs-kräfte werden von den Insassen als Störung empfunden. Je kleiner die Beschleunigung ist, desto besser der Fahrkomfort. Der Mensch nimmt dabei, auf Grund der Tatsache, dass er selber ein schwingungsfähiges Ge-bilde darstellt, Schwingungen gleicher Stärke, aber verschiedener Frequenz unterschiedlich stark wahr (Abbildung 3 u. vgl. [3]).

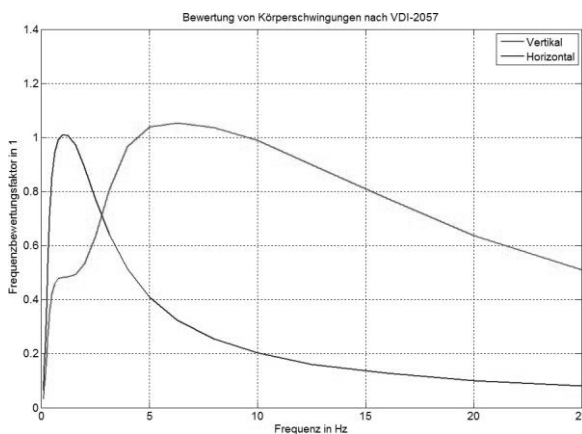


Abbildung 3. Frequenzbewertungsfilter für Körperschwingungen nach VDI-2057

Eine niedrige Aufbaubeschleunigung kann durch die schwingungstechnische Entkopplung des Aufbaus vom Fahrwerk mittels einer weichen Aufhängung in der betrachteten Bewegungsrichtung erreicht werden.

Die Erreichung einer hohen Fahrsicherheit kann mittels starker Dämpfung des Fahrzeugaufbaus zur Stabilisierung aller Aufbaubewegungen erreicht werden. Allerdings übertragen Dämpfer die hochfrequenten Anteile der Fahrwerksschwingungen (Abbildung 4). Die Abstützung der Dämpfungskräfte erzeugt Beschleunigungskräfte im Aufbau. 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 [4].

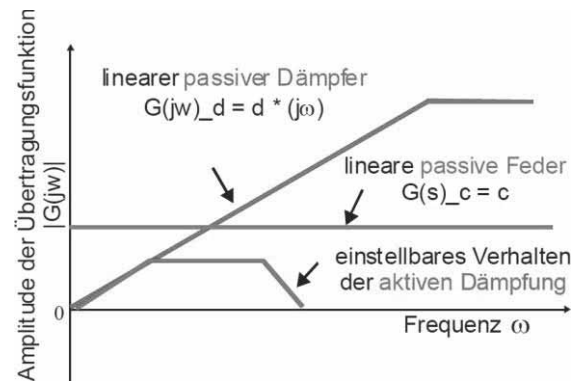


Abbildung 4. Übertragungsverhalten von Feder- und Dämpfungssystemen nach [1]

Im Gegensatz dazu kann durch eine 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 (s. [2]).

Für den Prüfstand wird dazu folgendes Funktionsprinzip der *aktiven Federung* angewendet (s. [1]). Mit Hilfe eines hydraulischen Aktorsystems, bestehend aus einer Hydraulikversorgung, einem Ventil und einem Zylinder wird die Lage des Federkopfpunktes aktiv verstellt (Abbildung 5). 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.

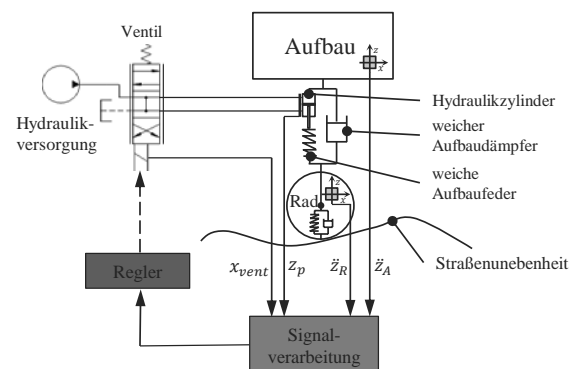


Abbildung 5. Prinzip der aktiven Federung

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 Kopfpunktverstellung der Federung bedämpft werden. Damit wird die gewünschte aktive Dämpfung im unteren Aufbaufrequenzbereich realisiert (s. [1]).

Der Vorteil dieses langsam-aktiven Systems besteht darin, dass mit der geringeren Anforderung an die Bandbreite der Regelung auch weniger Stellenergie benötigt wird, wobei gleichzeitig durch die passive Aufbaufeder eine gute Isolationswirkung für höhere Frequenzen erreicht werden kann. Ziel dieses Systems ist ein mit vollaktiven Federungen vergleichbares Ergebnis bei geringerem Energieaufwand.

3.3 Anregungs- und Energieversorgungskonzept

Wie beim realen Fahrzeug ermöglicht die Konfiguration des *Prüfstandsmoduls* ein Abheben des Reifens, was bei der modellbasierten Reglerauslegung mit zu berücksichtigen ist (s. Abschnitt 4). Das Straßenprofil wird über die Reifenaufstandsfläche mit Hilfe eines positionsgeregelten Hydraulikzylinders in das System geleitet. Die dabei eingesetzte Regelung entspricht der Lokalregelung der *aktiven Federung* mit an die Regelstrecke angepassten Parametern (s. Abschnitt 4.4).

Die hydraulische Leistung für die Aktorik wird mit Hilfe einer selbst konzipierten, mobilen Hydraulikversorgung aus dem Drehstromnetz der Hochschule erzeugt (Abbildung 6).

Ein über die Echtzeithardware steuerbarer Frequenzumrichter ermöglicht einen variablen Volumenstrom der Anlage (U/f-Steuerung). Die am Asynchronmotor direkt angeflanschte Außenzahnradpumpe wird dabei mit unterschiedlichen Drehzahlen betrieben. Die Pumpendrehzahl und damit der Volumenstrom ergeben sich aus der Außenleiternetzfreqenz des Motors, die der Frequenzumrichter erzeugt. Eine unterlagerte Druckregelung ermöglicht so mit Hilfe einer dynamischen Volumenstromversorgung einen energieeffizienten Betrieb der Anlage unter Laborbedingungen. Der Systemdruck ist durch ein Druckbegrenzungsventil abgesichert. Die Hydraulikversorgung verfügt aus Sicherheitsgründen über zusätzliche Sensorik für den Systemdruck, die Verschmutzung und die Temperatur des Druckmediums. Zwei Membranspeicher dienen der Glättung von Druckpulsationen und decken zusätzliche Spitzenverbräuche im Volumenstrom.

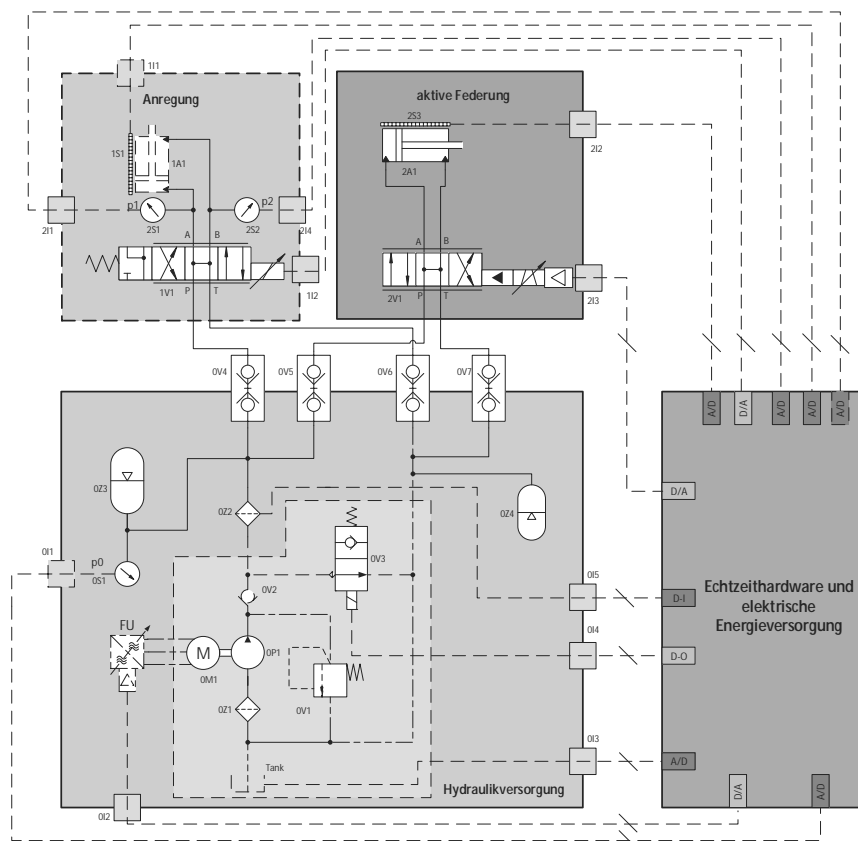


Abbildung 6. Hydraulischer und elektrischer Signalflussplan des Prüfstands

4 Modellbildung und Analyse

Der mechatronische Entwicklungsprozess unterstützt die gestalterorientierte Konstruktion durch die modellbasierte Auslegung unter Berücksichtigung der Dynamik der Übertragungsglieder. Zu Beginn des Entwicklungsprozess sind dazu vereinfachte Modelle in der Regel ausreichend. Im vorliegenden Abschnitt wird die Entwicklung der beiden Kernmodule *aktive Fahrzeugfederung* und *Anregungseinheit* anhand des mechatronischen Entwicklungskreislaufs sukzessive von der Modellierung über die Synthese bis hin zur Echtzeit-Realisierung dargestellt.

Im Rahmen der Modellbildung wird ein reales System, z. B. ein Fahrzeug, auf ein physikalisches und anschließend mathematisches Modell mit einer für das Problem ausreichenden Modellierungstiefe reduziert.

4.1 Dynamik der Tragstruktur

Zur Beschreibung der Vertikaldynamik eines Fahrzeugs ist, wie zuvor dargestellt, eine Reduktion auf ein zwei Massensystem als Modellierungstiefe zunächst ausreichend. Dem Feder-Dämpferbein eines realen Fahrzeugs können eine lineare Aufbaufedersteifigkeit c_A und eine viskose Aufbaudämpfung d_A zugeordnet werden. Die Aufbaumasse m_A , die auch als gefederte Masse bezeichnet wird, entspricht auf Grund der Symmetrie einem Viertel der Karosseriemasse samt Insassen. In der ungefederten Masse m_R werden die Bauteile des Radmoduls (Felge, Bremsscheibe, etc.) zusammengefasst, wobei die Radmasse den prozentual größten Anteil in der Regel darstellt. Die Dämpfung des Reifens ist vernachlässigbar gering, sodass lediglich die Reifensteifigkeit c_R im Modell betrachtet wird (Abbildung 7).

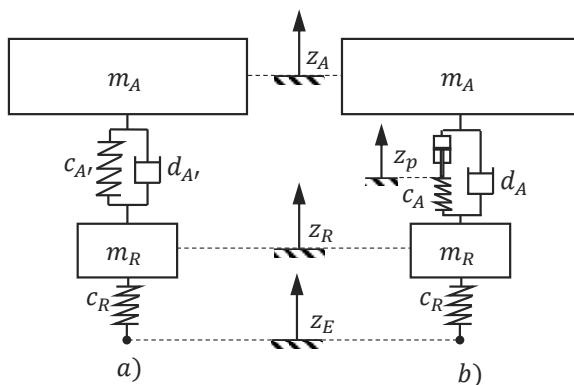


Abbildung 7. Lineares Viertelfahrzeugmodell für eine konventionelle (a) und langsam-aktive (b) Fahrzeugfederung

Für die langsam-aktive Fahrzeugfederung ist die zusätzliche Zustandsgröße der Zylinderposition z_p im Modell zu ergänzen. Mit dem Newton-Euler-Ansatz können daraus die Zustandsgleichungen für die vertikale Richtung z hergeleitet werden:

$$\begin{aligned} \dot{\underline{x}} &= \underline{A} \underline{x} + \underline{B} \underline{u} & \underline{y} &= \underline{C} \underline{x} + \underline{D} \underline{u} \\ \underline{x} &= [z_A \quad \dot{z}_A \quad z_R \quad \dot{z}_R \quad z_p]^T & \underline{u} &= [z_E] \\ \underline{y} &= [z_A \quad \ddot{z}_A \quad z_p]^T \end{aligned} \quad (1)$$

4.2 Dynamik des Stellsystems

Das Stellsystem der aktiven Federung besteht aus einem doppeltwirkenden Differentialzylinder und einem Servo-Wegeventil. Für die Anregungseinheit wird ein Regel-Wegeventil verwendet. Die Ventile wandeln das elektrische Steuersignal u_v über die Ventilschieberbewegung x_v dynamisch in einen Druckölstrom (Abbildung 8). Diese Dynamik kann durch ein System zweiter Ordnung modelliert werden (s. [5]).

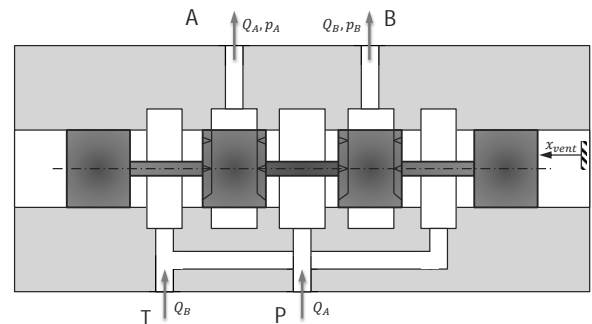


Abbildung 8. Physikalisches Ersatzmodell des stationären Verhaltens mit Nullüberdeckung (vgl. [1])

Die statischen Eigenschaften der Ventile, die den Bauteilen Nichtlinearitäten einprägen, werden durch die Durchflusskennlinien beschrieben. Die dynamische Abhängigkeit des Druckaufbaus in den beiden Kammern des Zylinders (p_A und p_B) in Abhängigkeit des Volumenstroms Q und der Kompressibilität des Druckmediums sowie Leckverlusten kann über eine Differentialgleichung 1. Ordnung modelliert werden (s. [6] u. [7]). Für den nichtlinearen Zustandsraum kann damit das Stellsystem wie folgt modelliert werden:

$$\begin{aligned} \dot{\underline{x}} &= f(\underline{x}, \underline{u}) & \underline{x} &= [x_v \quad \dot{x}_v \quad x_z \quad \dot{x}_z \quad p_A \quad p_B]^T \\ \underline{u} &= [u_v \quad F_z]^T & \underline{y} &= \underline{x} \end{aligned} \quad (2)$$

4.3 Dynamik der Sensorik und der digitalen Signalverarbeitung

Für die Reglerauslegung muss das dynamische Verhalten der eingesetzten Sensoren und die mit der digitalen Realisierung des Reglers auftretenden Totzeiten in der Gesamtdynamik berücksichtigt werden.

Als Positionssensoren für die Hydraulikzylinder wird hier ein induktiver Wegaufnehmer direkt am Zylinder gewählt. Die Messung der Stellung jedes Ventilschiebers erfolgt über die im Ventil integrierte Elektronik. Die Aufbau- und Radbeschleunigung wird mit Hilfe von zweidimensionalen Beschleunigungssensoren mit einer Bandbreite von 2 kHz bestimmt. Zur Bestimmung der Radlastschwankung wird je ein Drucksensor (p_1 und p_2 , Abbildung 6) an den Kammern des Anregungszylinders eingesetzt, die mit Hilfe der hydraulisch wirksamen Flächen und der identifizierten Reibung eine Umrechnung auf die am Zylinder wirkenden Kräfte ermöglichen. Die Sensoren verringern die Bandbreite des Systems, was durch ein PT2-Glied berücksichtigt wird.

4.4 Reglersynthese und Analyse des Systems

Mit Hilfe der erstellten Modelle wird die Regelstruktur modellbasiert ausgelegt. Folglich wird am Prüfstand eine hierarchisch angeordnete Regelstruktur konzipiert, die aus einer Global- und einer Lokalregelung

besteht (Abbildung 10). Der Vorteil dieser Modularisierung ist, dass bei einem Austausch des Aktorsystems nur die Lokalregelung angepasst werden muss.

Die Globalregelung ermittelt aus den Sensorsignalen nach einem Skyhook-Algorithmus mit den Reglerparameter k_c und k_d unter Berücksichtigung der Kinematikfunktionen die Referenzkraft zwischen Aufbau und Fahrwerk als Vorgabe für die Lokalregelung. Mit Hilfe der Federsteifigkeit c_A wird die Sollauslenkung der passiven Feder bestimmt (vgl. [8]). Diese Sollauslenkung wird durch den Hydraulikzylinder geregelt. Die dazu notwendige Bandbreite der Ventil-Zylinder-Einheit wird durch eine dreifache Kaskade erreicht.

Bei einem konventionellen Aufbaufederungssystem hängt die erzeugte Kraft zwischen Aufbau und Fahrwerk von Relativweg und -geschwindigkeit ab (s. Gl. 3 und Abbildung 7). Bei der Auslegung derartiger Systeme muss immer ein Kompromiss zwischen der Isolation des Fahrzeugs gegen hochfrequente Störungen und der Dämpfung der Resonanzantwort des Systems gefunden werden (s. [1]).

$$F_{konventionell} = c_A \cdot (z_A - z_R) + d_A \cdot (\dot{z}_A - \dot{z}_R) \quad (3)$$

$$F_{skyhook} = k_c \cdot (z_A - z_R) + k_d \cdot \dot{z}_A \quad (4)$$

Für das aktive System wird deswegen nach dem Stand der Technik ein Regelgesetz abgeleitet, dass strukturell

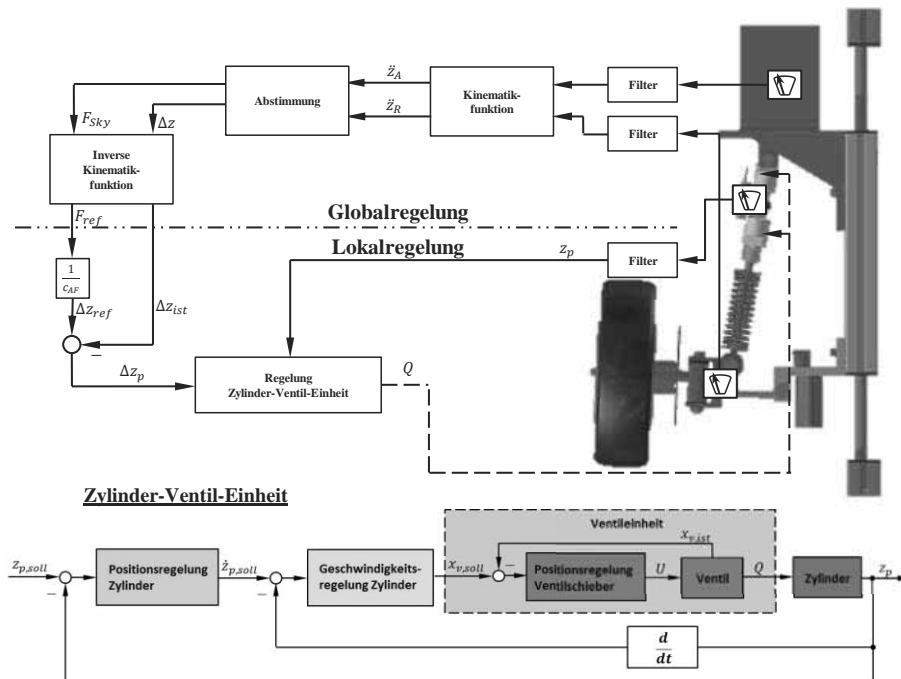


Abbildung 10. Hierarchische Reglerstruktur der Aufbauregelung am Prüfstand

wie ein Dämpfer wirkt, der sich gegen das Inertialsystem abstützt (s. Gl. 4). Das System führt einerseits zu einer effektiven Isolation der hochfrequenten Schwingungen und andererseits zu guter Dämpfung der Resonanzüberhöhung. Ein solcher Dämpfer kann allerdings nicht durch passive Elemente bei Fahrzeugen realisiert werden, da es keine festen Bezugspunkte im Inertialsystem gibt. Nur mit dem Einsatz der Aktoren und der Regelung kann die Wirkung eines Skyhook-Dämpfers im Fahrzeug aufgebracht werden (s. [1]).

Abbildung 9 stellt das Schwingungsverhalten dieser Gesetzmäßigkeiten in der Simulation für das Kleinfahrzeug unter Berücksichtigung des Maßstabes für das passive und aktive System bei einer Sprunganregung am Systemeingang z_E dar. Das Simulationsergebnis zeigt, dass das Schwingungsverhalten des aktiven Systems eine wesentlich bessere Dämpfung als das passive System aufweist. Daraus lässt sich auf ein gutes Führungsverhalten der Aufbaubewegung z_A schließen.

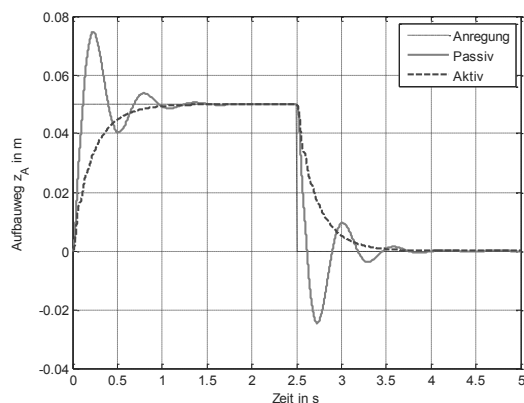


Abbildung 9. Zeitverhalten des linearen Viertelfahrzeugmodells bei Sprunganregung.

5 Realisierung und Inbetriebnahme des HiL-Prüfstands

Mit Hilfe des modellgestützten Entwicklungsprozesses wird in der Entwurfsphase neben der gestaltorientierten Konstruktion auch die Dynamik der dazugehörigen Aktorik, Sensorik und Informationsverarbeitung berücksichtigt. Im vorliegenden Abschnitt sind die Realisierung der *aktiven Federung* und die Inbetriebnahme des Prüfstands aufgeführt.

5.1 Konstruktive Realisierung der aktiven Federung

Zur Funktionserprobung des Systems wird das vorgestellte Konzept der aktiven Federung konstruktiv

durch eine Reihenschaltung einer passiven Spiralfeder und einem doppelwirkenden Differentialzylinder mit einem Hub von 32 mm, der aus der zusätzlichen Anforderung des verfügbaren Bauraums am Versuchsfahrzeug resultiert, umgesetzt. Die Spiralfeder benötigt konstruktiv eine Querkraftstabilisierung, die durch eine reibungsarme Gleitlagerführung erreicht wird (Abbildung 11). Im Modell kann diese Reibung mit Hilfe einer viskosen Dämpfung zusätzlich berücksichtigt werden, ebenso ist eine Stellgrößenbeschränkung des Federwegs vorzusehen.

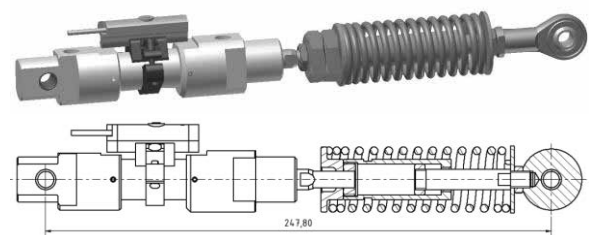


Abbildung 11. Realisiertes, aktives Federelement

5.2 Inbetriebnahme des HiL-Prüfstands

Die Inbetriebnahme des Prüfstands erfolgt modulweise von der untersten Hierarchie zur obersten, um eine fehlerfreie Funktionsweise sicherzustellen. Abbildung 12 zeigt den Aufbau des Prüfstands im Labor. Mit Hilfe einer grafischen Benutzeroberfläche (GUI), die mit Hilfe eines Rapid-Control-Prototyping-Systems erstellt wurde, ist eine komfortable Bedienung der Echtzeithardware möglich. Über die Oberfläche kann neben der Erprobung auch die Überwachung der Messsignale und Betriebszustände gewährleistet werden. Darüber hinaus kann auf wichtige System- und Reglerparameter online zugegriffen werden.

Das Prüfstandsmodul ermöglicht Referenzmessungen an konventionellen Aufbaufederungssystemen sowie die Erprobung von aktiven Systemen. Die Kopplung mit der Echtzeithardware erfolgt über definierte Schnittstellen der Aktorik und Sensorik.

Neben den diversen Anregungsprofilen für die Identifikation und Validierung der Modelle und der Regelung, ermöglicht der Prüfstand das Abbilden von realitätsnahen Fahrbahnprofilen nach ISO-Norm 8609.

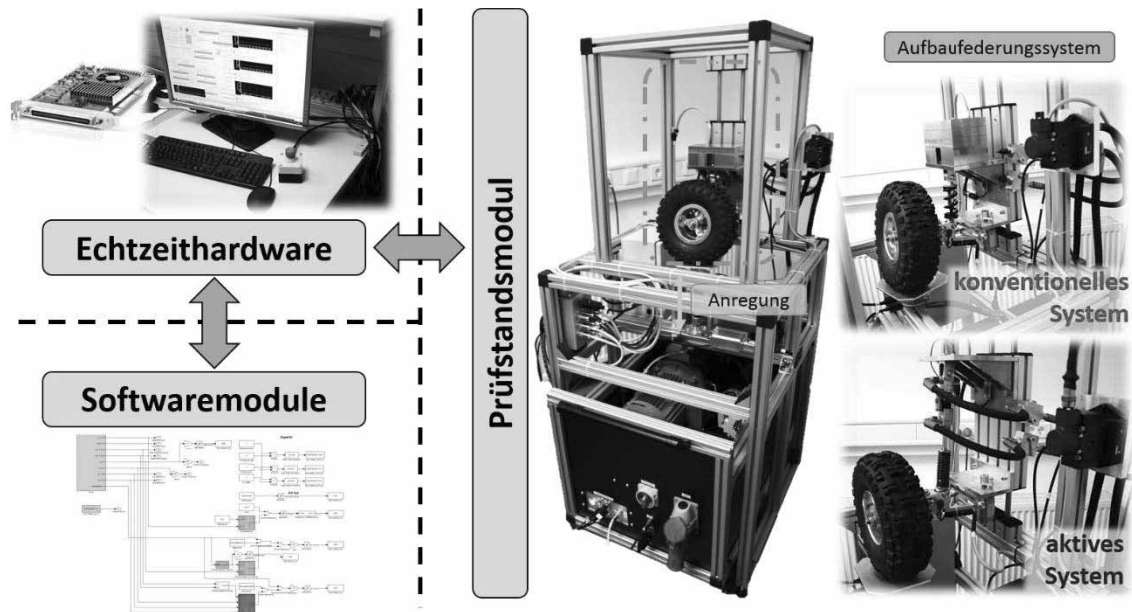


Abbildung 12. Realisierung des HiL-Prüfstands zur Erprobung einer aktiven Federung

Die Norm klassifiziert reale Straßen anhand von Kennwerten in Klassen von A bis E in Abhängigkeit der subjektiven Ebenheit von *sehr gut* bis *sehr schlecht*. Am Prüfstand werden die gewünschten Profile mit einem weißen Rauschen, das in seiner Intensität nach der Methodik aus [9] skaliert ist, realisiert (Abbildung 13).

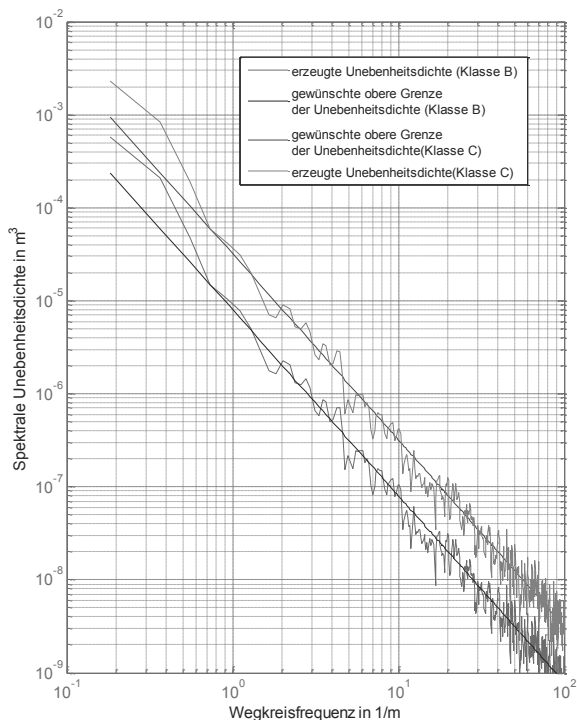


Abbildung 13. Realitätsnahes Fahrbahnprofil nach ISO-8609

Darüber hinaus ist es möglich an den Signalschnittstellen Frequenzgangmessung mit einem Signalanalysator und -generator durchzuführen.

Abbildung 14 zeigt das Ergebnis der Messung einer konventionellen Aufbaufederung aus einem Fahrrad. Dargestellt ist die Aufbaubeschleunigung \ddot{z}_A im Verhältnis zum Anregungssignal z_E , welches als weißes Rauschen in das System eingebracht wurde. Auffällig daran ist die hohe Eigenfrequenz des Rads von etwa 35 Hz. Ursache ist die hohe Aufbaufedersteifigkeit, die aus der unterschiedlichen Kinematik eines Fahrrads im Vergleich zu einem Kraftfahrzeug resultiert.

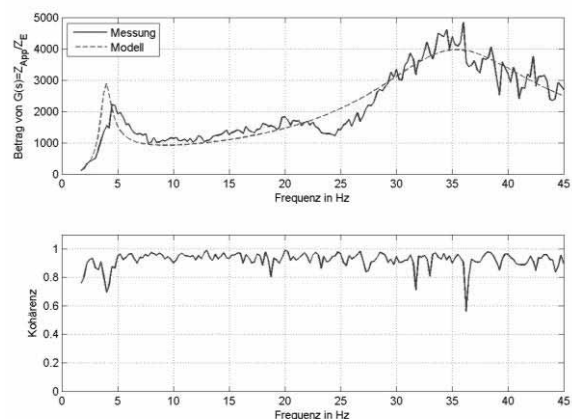


Abbildung 14. Messung am realisierten Prüfstand eines konventionellen Federungssystems

6 Zusammenfassung

Der Beitrag stellt die mechatronische Entwicklung eines HiL-Prüfstands zur Erprobung einer *aktiven Fahrzeugfederung* dar. Dazu wird die verwendete Entwicklungsmethodik vorgestellt, anhand derer die Auslegung der mechatronischen Funktionen modellbasiert erfolgt.

Als Übergang vom theoretischen Rechnermodell zum realen System erfüllt der HiL-Prüfstand die Aufgaben der Erprobung und Optimierung der Regleralgorithmen unter realitätsnahen Bedingungen. Die Inbetriebnahme des Prüfstands bestätigt anhand von Messungen die modellbasierte Auslegung der Funktionen.

Der Prüfstand ist damit ein unterstützendes Werkzeug zur Realisierung von Forschungsprojekten und gleichermaßen ein Hilfsmittel zur Veranschaulichung anwendungsorientierter Probleme im Rahmen von Lehrveranstaltungen.

Danksagung

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

Gefördert durch:



**Niedersächsisches Ministerium
für Wissenschaft und Kultur**

7 Referenzen

- [1] Liu-Henke, X.: *Mechatronische Entwicklung der aktiven Feder-/Neigetechnik für das Schienenfahrzeug RailCab*, VDI- Fortschritt-Berichte, Reihe 12, Nr. 589, VDI-Verlage, Düsseldorf, 2004.
- [2] Liu-Henke, X.; Buchta, R.; Scheele, M.: *Systemkonzept einer aktiven Fahrzeugfederung für Elektrofahrzeuge*, ASIM STS/GMMS Workshop 2013, Düsseldorf, 28.02. - 01.03., 2013.
- [3] Mitschke, M.: *Dynamik der Kraftfahrzeuge*. Springer Verlag, Deutschland, 2004.
- [4] Darenberg, W.; Gall, H.; Acker, B.: *Chancen und Probleme aktiver Kraftfahrzeugfederung*, VDI-Berichte, Nr. 515, VDI-Verlag, 1984, S. 187-194.
- [5] Panther, M.: *Identifikation physikalischer Systemparameter mechanisch-hydraulischer Mehrgrößensysteme*, Fortschr.-Ber. VDI-Z, Reihe 8, Nr. 76, VDI-Verlag, Düsseldorf, 1984.
- [6] Backé, W.: *Umdruck zur Vorlesung Servohydraulik*. Institut für hydraulische und pneumatische Antriebe und Steuerungen der Rheinisch-Westfälischen Technischen Hochschule Aachen, 1986.
- [7] Liu-Henke, X.; Jücker, J.; Jäker, K.-P.: *Ganzheitlicher mechatronischer Entwurf eines aktiven Feder-/Neigemoduls*. VDI-Tagung: Mechatronik - Mechanisch/Elektrische Antriebstechnik, Wiesloch, 29./30. März 2000.
- [8] Koch, T.: *Integration von Konstruktion und mechatronischer Komposition während des Entwurfs mechatronischer Systeme am Beispiel eines integrierten Radmoduls*. Dissertation, Universität Paderborn, Deutschland, 2005.
- [9] Benz, R.: *Fahrzeugsimulation zur Zuverlässigkeitsabsicherung von karosseriefesten Kfz-Komponenten*, Dissertation, Universität Karlsruhe, Deutschland, 2007.

Ein skalierbares Echtzeitsystem zur Erprobung des Batteriemanagements in Elektrofahrzeugen

Florian Quantmeyer, Matthias Roch, Waldemar Diehl, Xiaobo Liu-Henke

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

Flo.Quantmeyer@Ostfalia.de

An der Ostfalia Hochschule wurde ein skalierbares Echtzeitsystem konzipiert und realisiert, das zur Entwicklung, Erprobung und Optimierung von Softwarefunktionen für das Management von Batterien in Elektrofahrzeugen dient. Es ermöglicht bereits in frühen Entwicklungsphasen die Erprobung von Softwarefunktionen am realen System unter Echtzeitbedingungen. Neben dem übergeordneten Controller verfügt es über dezentrale Zellmodule. Dieser Aufbau ermöglicht flexible Skalierung für verschiedene Anwendungen. Neben der Darstellung der hardwareseitigen Realisierung gibt dieser Beitrag einen Einblick in den effizienten Softwareentwicklungsprozess, welcher durch das Echtzeitsystem ermöglicht wird.

1 Einleitung

Derzeit stellen Batteriesysteme ein wesentliches Hindernis für die flächendeckende Verbreitung von Elektrofahrzeugen dar. Die im Vergleich zu fossilen Energieträgern wesentlich geringere Energiedichte führt trotz hohen Wirkungsgrades bei der Energiewandlung zu einer erheblich niedrigeren Reichweite. Ein weiterer Nachteil liegt in den hohen Investitionskosten in Verbindung mit der geringen Lebensdauer. Auch die Sicherheit einiger Batterietypen stellt derzeit noch ein Risiko dar.

Als Batterietypen höherer Energiedichte steigt im Besonderen die Bedeutung von Lithium-Ionen-Zellen in mobilen Geräten sowie in Kraftfahrzeugen stetig an. Um die benötigte Spannung bereitstellen zu können, werden mehrere Zellen in Reihenschaltung betrieben. Auch Batteriezellen identischer Bauart variieren dabei in den maßgebenden Eigenschaften (Kapazität, Innenwiderstand, etc.), was unumgänglich zu unterschiedlichen Ladeständen im Betrieb führt. Da Lithium(Li)-Ionen-Zellen sowohl durch zu hohe als auch zu niedrige Spannung beschädigt werden, können die unterschiedlichen Ladestände kritische Folgen haben.

Um zu gewährleisten, dass die Zellen sicher arbeiten und ihr Potenzial vollständig zu nutzen wird ein Batteriemanagementsystem, kurz BMS, eingesetzt.

2 Batteriemanagement

Das BMS für Li-Ionen-Batterien muss eine Vielzahl von Aufgaben erfüllen, damit ein störungsfreier Betrieb des Speichers gewährleistet werden kann.

Als Basis zur Erfüllung dieser Funktionen werden mithilfe der Sensorik Spannung, Temperatur und Strom der Batteriezellen erfasst und durch eine Analog-Digital-Wandlung dem Mikrocontroller, auf dem die Softwarefunktionen ausgeführt werden, bereitgestellt.

Das BMS selbst hat nicht die Möglichkeit, die aktuelle Leistung, die der Batterie entnommen wird, zu beeinflussen. Diese Aufgabe übernimmt das Energiemanagement. Dazu ist es erforderlich, dass das BMS die aktuell entnehmbare Leistung ermittelt und diesen s.g. State-of-Function (SoF) dem Energiemanagement kommuniziert.

Von besonderer Bedeutung für die Zuverlässigkeit des Gesamtsystems „Elektrofahrzeug“ ist eine zuverlässige Ermittlung der Restreichweite. Hierfür ist die Ladezustandsermittlung der Lithium-Batterie grundlegende Voraussetzung.

Der Alterungsprozess der Batterie, mit dem die Abnahme der verfügbaren Kapazität sowie die Zunahme des Innenwiderstands einhergehen, wird durch das BMS überwacht.

In der Regel unterliegen die Parameter der Batteriezellen durch den Produktionsprozess einer geringen Streuung. Dies führt dazu, dass die individuellen Zellen während des Betriebs der Batterie verschiedene Ladezustände annehmen. Mit jedem Lade- und Entladezyklus nimmt diese Ungleichmäßigkeit in der Folge zu. Es ist daher notwendig die Ladestände der jeweiligen Zellen während des Ladens anzugleichen.

Die Wandlung chemischer in elektrische Energie und umgekehrt findet nicht verlustfrei statt. Während des Betriebs der Batterie kommt es zur Erwärmung. Um die Betriebsgrenzen auch bei hohen Leistungen einzuhalten ist eine Kühlung erforderlich. Außerdem kann es sinnvoll sein, die Batterie bei geringer Temperatur zu wärmen, um ihre Leistungsfähigkeit zu steigern. Es ist demnach ein Temperaturmanagement der Batterie erforderlich, das durch das Batteriemanagement gewährleistet wird.

Weiterhin kommen dem Batteriemanagement verschiedene Sicherheitsfunktionen zu. So muss das Batteriemanagement die Einhaltung der Betriebsgrenzen der Batterie gewährleisten, um Sicherheitsrisiken und eine irreversible Schädigung der Batterie zu vermeiden. Für diesen Fall sind entsprechende Abschaltstrategien zu entwickeln, die dem Fahrer ein sicheres Halten ermöglichen und nicht zu einem plötzlichen Verlust der wesentlichen Fahrzeugfunktionen führen. Ebenso kann auf Verlangen anderer Fahrzeugsysteme eine Abschaltung des Batteriepakets notwendig sein, z.B. bei einem Unfall.

Tabelle 1 fasst die wesentlichen Funktionen des Batteriemanagements zusammen.

Funktion	Beschreibung
Sicherheit	Betriebsgrenzen einhalten
Ladezustands-ermittlung	Ermittelt den Ladezustand auf Basis der verfügbaren Information
Alterungszustand ermitteln	Berechnung eines Kennwerts zur Charakterisierung der Zellalterung
Leistungsfähigkeit ermitteln	Ermittelt die maximale Leistung
Messung	Verarbeitung der Messdaten zu Spannung, Strom und Temperatur
Temperaturregelung	Wärmung oder Kühlung der Lithium-Batterie
Ladeausgleich	Angleichung des Ladestands der Zellen eines Batteriestrangs
Fahrzeugintegration	Einbindung in den Steuergeräteverbund
Fail-Safe	Erkennung von Fehlern, Notlauf- / Abschaltstrategie

Tabelle 1. Grundlegende Funktionen des BMS

Das Batteriesystem verfügt somit neben dem eigentlichen technischen System über verschiedene Sensoren, ein Computersystem und Aktoren. Es handelt sich demnach um ein typisches eingebettetes System.

Das Batteriesystem als eingebettetes System ist in Abbildung 1 dargestellt.

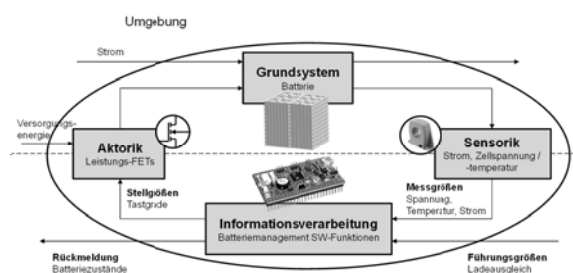


Abbildung 1. Batteriemanagement als Embedded Control System

3 Softwareentwicklungsprozess

Zur Beherrschung der Softwarekomplexität in der Entwicklung wird der durchgängig modellbasierte, verifikationsorientierte Entwicklungsprozess nach Abbildung 2 angewendet.

Nach der Modellbildung der Batterie erfolgt der Softwareentwurf und die Erprobung mittels Modell-in-the-Loop (MiL)- und Software-in-the-Loop (SiL)-Simulation. Dabei werden die Regelfunktionen offline gegen ein Modell der Batterie getestet.

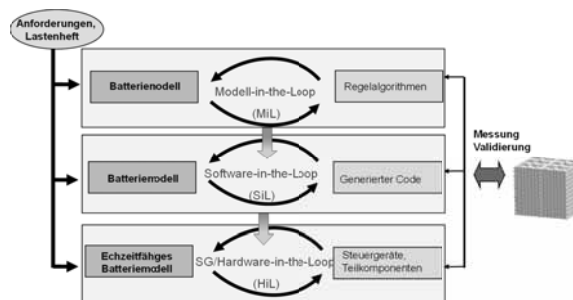


Abbildung 2. Softwareentwicklungs- und Absicherungsprozess nach [1]

Ist die geforderte Güte der Regelfunktionen in der Offline Simulation nachgewiesen, wird die Software auf die Zielhardware übertragen. Anschließend erfolgt die Absicherung dieses Steuergeräts mittels Hardware-in-the-Loop (HiL)-Simulation. Dabei wird ein echtzeitfähiges Streckenmodell auf einer leistungsstarken Hardware berechnet und dem Steuergerät das reale Systemverhalten über die vorhandenen Schnittstellen vorgespielt.

Neben dem dargestellten Entwicklungs- und Absicherungsprozess ist es besonders bei Batteriesystemen sinnvoll, die Softwarefunktionen möglichst früh am Realsystem zu erproben. Dadurch kann festgestellt werden, ob das Streckenmodell eine ausreichende Güte aufweist, ob Regleroptimierungen sinnvoll sind und ob das Entwurfsziel überhaupt erreicht wird [2]. Besonders das komplexe Verhalten der Lithium-Batterie führt dazu, dass Modelle die das Verhalten genau abbilden nur unter großem Aufwand gewonnen werden können (Alterung, etc.).

Die Grundvoraussetzung für Erprobung der Softwarefunktionen gegen die reale Strecke ist eine durchgängige Entwicklungstoolkette, um aus den Regelfunktionen, die üblicherweise graphisch programmiert vorliegen, automatisch Code zu erzeugen, der auf eine Experimentalhardware geladen werden kann, die über geeignete Peripherie verfügt, um über Sensorik und Aktorik mit der Batterie gekoppelt zu werden. Während der Tests muss eine Möglichkeit zur Steuerung und zum Anzeigen systemrelevanter Größen für den Benutzer vorhanden sein.

Da sich die Batterie meist aus einer hohen Anzahl von Zellen zusammensetzt und jede dieser Zellen gemessen und geregelt (Ladeausgleich) werden muss, muss die Experimental-Hardware über eine Vielzahl von Schnittstellen verfügen. Hierdurch hebt sich das Batteriesystem vom Einsatzgebiet anderer RCP-Systeme ab.

4 Konzeption des skalierbaren Echtzeitsystems zur Erprobung der BMS-Software

Zur Unterstützung des modellbasierten Entwicklungsprozesses des Batteriemanagements wurde ein skalierbares Echtzeitsystem konzipiert, das speziell auf diesen Einsatzzweck zugeschnitten ist.

Mithilfe dieses Echtzeitsystems sollen alle SW-Funktionen des realen Steuergeräts erprobt werden können. D.h. es muss über alle Schnittstellen verfügen, die auch das spätere Steuergerät aufweist.

Besonders die Realisierung des Ladeausgleichs kann auf unterschiedliche Wege erfolgen. Eine umfangreiche Übersicht der Verfahren ist beispielsweise in [3] und [4] gegeben. Beim Shunting Resistor Verfahren wird während der Ladung bei Zellen hohen Ladestands ein Bypass mit einem Widerstand freigegeben, an dem elektrische Energie dissipiert wird. Das Boost Shunting ermöglicht eine Verteilung überschüssiger

elektrischer Energie an das Batteriepack. Das Switched Converter Verfahren erlaubt hingegen den direkten Transport elektrischer Energie von einer Zelle mit hohem Ladestand zu einer mit geringem Ladestand.

Es deutet sich an, dass die Auswahl des Ladeausgleichsverfahrens also einen erheblichen Einfluss auf die Gestaltung der Experimental-Hardware haben muss.

Weiterhin muss eine Kommunikation der Echtzeitsystems mit anderen Steuergeräten des Fahrzeugs ermöglicht werden. Dies erfolgt typischerweise mittels CAN-Bus.

Generell sind nach [5] verschiedene Hardwaretopologien des Batteriemanagementsystems möglich (Abbildung 3).

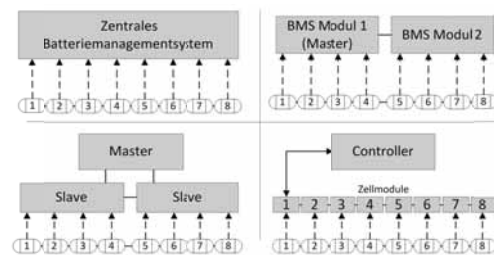


Abbildung 3. Hardwarestrukturen von Batteriemanagementsystemen [5]

Beim zentralen Batteriemanagementsystem sind alle Aktoren und Sensoren der einzelnen Zellen mit dem einzigen Steuergerät verbunden. Nachteilig ist hierbei, dass jede Batterie mit der unterschiedlichen Anzahl von Zellen eine entsprechende Anzahl an I/Os am Steuergerät erfordert. Zur Steigerung der Flexibilität wurde daher eine modulare Bauweise eingeführt (Abbildung 3, oben rechts). Je nach Anzahl der Zellen im Batteriepack lässt sich eine unterschiedliche Anzahl von BMS-Modulen einsetzen, die über einen Kommunikationskanal miteinander gekoppelt sind. Eines der Module agiert hierbei als Master.

Auch die dritte Variante verfügt über eine Master-Slave-Struktur. Mehrere identische Slave-Module überwachen jeweils mehrere Zellen und kommunizieren mit einem Master-Modul, das weitere Funktionen des BMS ausführt. Die Flexibilität des BMS kann durch diese Struktur noch einmal gesteigert werden.

Die vierte BMS-Struktur (unten rechts) ist vollständig dezentralisiert. Hierbei wird auf jeder Batteriezelle ein Zellmodul implementiert, das die lokalen Funktionen, wie z.B. Messdatenaufbereitung, Schalten übernimmt. Die Zellmodule kommunizieren mit dem

BMS-Controller über ein Bussystem. Hierdurch kann ein hohes Maß an Flexibilität erreicht werden. Die wesentlichen Softwarefunktionen sind jedoch auf dem BMS-Controller zentralisiert.

5 Hardwarerealisierung

Für die Experimental-Hardware wurde die vollständig dezentrale Struktur gewählt (Abbildung 4). Hierbei werden die Softwarefunktionen auf den BMS-Controller und die Zellmodule verteilt. Während die Zellmodule Basisfunktionen, wie die Messdatenaufbereitung und die Ansteuerung des Ladeausgleichs ausführen, werden auf dem BMS-Controller intelligente Funktionen wie die Ermittlung der Batteriezustände implementiert.

Es können verschieden viele Zellmodule über das interne Bussystem mit dem BMS-Controller gekoppelt werden. Dadurch lässt sich das Batteriesystem beliebig skalieren.

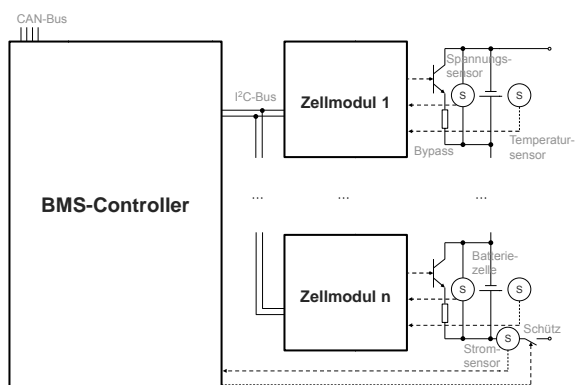


Abbildung 4. Konzept des skalierbaren Echtzeitsystems

Kern des BMS-Controllers (Abbildung 5) ist ein Mikrocontroller vom Typ STM32. Die Programmierung und Kommunikation zum Host-PC erfolgt mittels USB (vgl. Kapitel 6). Verschiedene Anschlüsse zur Bedienung und ein Display ermöglichen verschiedene Einstellungen im Betrieb ohne Host-PC.

Über den CAN-Bus erfolgt eine Kommunikation mit anderen Steuergeräten im Fahrzeug. Weiterhin sind eine Schnittstelle zum Schalten des Batterieschützes, sowie eine Schnittstelle zur Kopplung des Lüfters vorgesehen. Zur Auswertung der Stromsensoren sind passende Messwiderstände und analoge Filter vorgesehen.

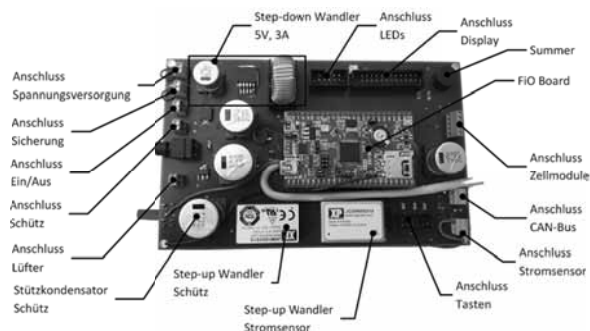


Abbildung 5. BMS-Controller

Die Zellmodule verfügen über einen 8-bit-Mikrocontroller, auf dem die Messdatenverarbeitung durchgeführt wird und der den Ladeausgleich lokal realisiert. Die Kommunikation mit dem BMS-Controller erfolgt mittels I²C-Bus. Die unterschiedlichen Potentiale von BMS-Controller und Zellmodulen erfordern eine galvanische Trennung der Busverbindung.

Die Simulation der Batterie mit individuellen Zellen vom gleichen Typ (Kapazitätsverteilung wurde durch Stichproben an Lithium-Zellen ermittelt) ergab einen geringen energetischen Vorteil der Ladeausgleichsverfahren mit Energieumverteilung von kleiner 1.5% (angenommener Wirkungsgrad $\eta = 80\%$). Aus diesem Grund wurde das *Shunting Resistor* – Verfahren gewählt. Dazu verfügt das Zellmodul über einen Bypass mit Widerstand, der mittels eines MosFET freigegeben werden kann.

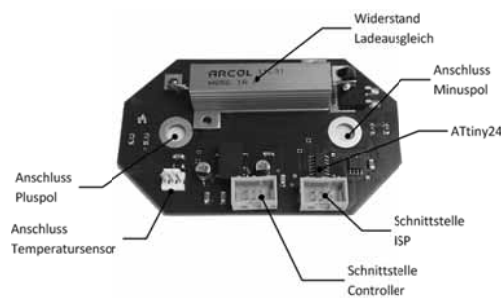


Abbildung 6. Zellmodul

6 Anwendung im Entwicklungsprozess

Wie in Kap. 3 beschrieben erfolgt die Entwicklung und Absicherung der BMS-Software zunächst offline mittels MiL-Simulation. Bereits nach diesem Entwicklungsschritt lässt sich die entworfene Software auf das Echtzeithardwaresystem übertragen und gegen die reale Strecke testen.

Die effiziente Vorgehensweise wird dabei durch die Möglichkeit der graphischen Programmierung des

BMS-Controllers in Simulink® mithilfe des *RapidSTM32*-Blocksets (Fa. Aimagin) geschaffen (vgl. Abbildung 7). Mithilfe des Real-Time Workshops / Embedded Coders wird automatisch C-Code dieses Programms erstellt. Ein geeigneter Compiler erzeugt aus dem C-Code einen vom Mikrocontroller ausführbaren Code, der schließlich auf den Controller übertragen wird.

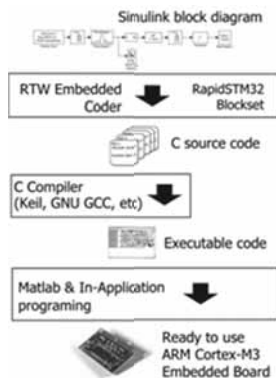


Abbildung 7. Programmierung des FiO-Board [6]

Abbildung 8 zeigt die typische Struktur eines Programms für den BMS-Controller. Gegenüber dem Simulationsprogramm aus der MiL/SiL-Phase wurde das Streckenmodell entfernt. Stattdessen werden die Schnittstellen der Softwarefunktionen mit den Ein- und Ausgängen des Controllers verknüpft. Über zusätzliche Konfigurationsblöcke lassen sich Einstellungen zum Compiler sowie zur Taktfrequenz des Controllers vornehmen.

Um eine Beobachtung und Steuerung durch den Nutzer zu ermöglichen, wird eine Verbindung mit dem Host-PC hergestellt. Dies geschieht über zusätzliche Sende- und Empfangsblöcke, die mit den zu beobachtenden Signalen verknüpft werden. Die eigentliche Beobachtung und Steuerung erfolgt über ein weiteres Simulink-Programm das auf dem Host-PC parallel zum BMS-Controller ausgeführt wird.

Das skalierbare Echtzeitsystem für die Entwicklung der Software des Batteriemanagements erlaubt es somit innerhalb von Sekunden Änderungen der Software auf den BMS-Controller zu übertragen und ermöglicht somit effiziente Tests und Optimierungen unter Echtzeitbedingungen.

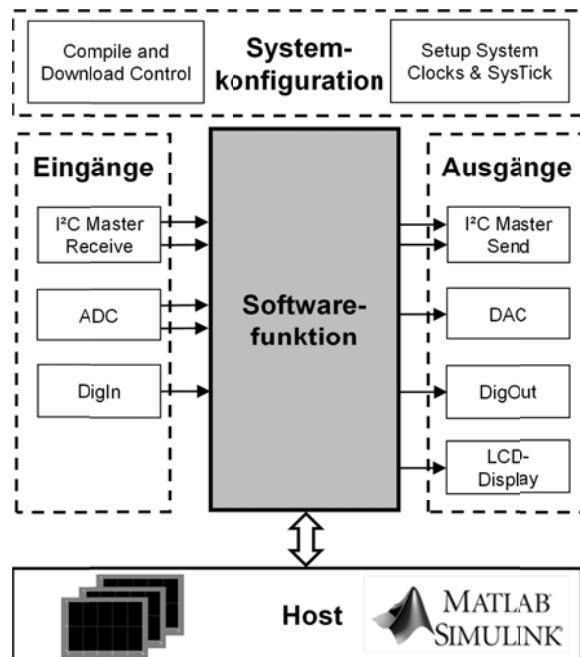


Abbildung 8. Struktur des Programms für den BMS-Controller

7 Zusammenfassung

Um das Potenzial dieser Energiespeicher vollständig zu nutzen, ihre Lebensdauer zu erhöhen und vor Schäden im Betrieb zu schützen, ist der Einsatz eines Batteriemanagementsystems notwendig. Dabei handelt es sich um ein typisches eingebettetes System mit hochgradig komplexer Software. Zur Beherrschung der hohen Systemkomplexität wird bei der Funktionsauslegung zunehmend ein durchgängiger Entwicklungsprozess von Model-in-the-Loop (MiL) über Software-in-the-Loop (SiL) bis hin zu Hardware-in-the-Loop (HiL) eingesetzt.

Speziell für die Entwicklung, Erprobung und Optimierung der Softwarefunktionen des Batteriemanagements unter Echtzeitbedingungen wurde ein kostengünstiges Echtzeitsystem konzipiert und realisiert. Dieses RCP-System ermöglicht es noch vor dem Vorliegen des Steuergeräts die Software gegen die reale Strecke (Lithium-Batterie) zu testen.

Das System besteht aus einer modularen und skalierbaren Hardwareplattform mit einer zentralen Steuereinheit – dem BMS-Controller – und mehreren Zellmodulen, die mit dem BMS-Controller kommunizieren. Die Integration in den Steuergeräteverbund des Fahrzeugs erfolgt mittels CAN-Bus.

Den Kern der Steuereinheit bildet ein FiO-Board der Fa. Aimagin, auf dem ein Mikrocontroller vom Typ

STM32 arbeitet. Der Einsatz eines Autocodegenerators sowie eines speziellen Simulink-Blocksets (*RapidSTM32*-Blockset, Fa. Aimagin) zur Unterstützung des Controllers erlauben die vollständige graphische Programmierung. Ferner lassen sich über eine Schnittstelle die Zustände des Batteriemanagements in die Entwicklungssoftware auf dem Host-PC übertragen und online betrachten.

Das dargestellte skalierbare Echtzeitsystem wird derzeit zur Entwicklung verschiedener Funktionen des Batteriemanagements an der Ostfalia Hochschule eingesetzt.

8 Danksagung

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 17N0911 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

9 References

- [1] X. Liu-Henke, S. Duym: *Modellgestützte Funktionsabsicherung des vernetzten mechatronischen Kraftfahrzeugs*. VDI-Tagung Mechatronik, Wiesloch, 01.-02.06.2005.
- [2] H. Hanselmann. *Beschleunigte Mechatronik-Entwicklung durch Rapid Control Prototyping und Hardware-in-the-Loop-Simulation*. At – Automatisierungstechnik, Nr. 3, S. 113-119, 1998.
- [3] M. Daowd, N. Omar, P. Van den Bossche und J. Van Mierlo. *Passive and Active Battery Balancing comparison based on MATLAB Simulation*. Proc. of the IEEE Vehicle Power and Propulsion Conference (VPPC), p. 1-7, September 6-9, Chicago, 2011.
- [4] J. Cao, N. Schofield and A. Emadi. *Battery Balancing Methods: A Comprehensive Review*. Proc. of the IEEE Vehicle Power and Propulsion Conference (VPPC), p. 1-6, September 3-5, Harbin, 2008.
- [5] W. Diehl. *Entwicklung eines Batteriemanagementsystems für Lithium-Batterien*. Master Thesis, Ostfalia Hochschule, Wolfenbüttel, 2013.
- [6] Aimagin Co., Ltd. *The RapidSTM32 Way*. https://www.aimagin.com/learn/images/2/29/The_rapid_stm32_way.png

Powertrain Cosimulation in a MiL and SiL Setup using the AUTOSAR and FMI Standards

Dr. Christoph Störmer, Dr. Christoph Malz, Dr. Corina Mitrohin

ETAS GmbH Stuttgart, Germany

christoph.stoermer@etas.com

Developing control software for heterogeneous physical domains challenges the embedded systems engineers in at least two directions. First, domain barriers, like between mechanical and electrical domains, have to be overcome such that a heterogeneous system can be built, either like a real system, or like a virtual representation of it. Secondly, breaks between development phases should be avoided and a seamless transition along them should be struggled at. With our work, we do contribute a concrete example on how emerging standards for model exchange (FMI) and automotive software architecture (AUTOSAR) help bringing the gap between domains and development phases.

1 Challenges of Powertrain Co-Simulation

The virtualization of development processes in the automotive area, i.e., loading activities from late development phases to the front, thus increasingly working with virtual representations of engineering artifacts, and having seamless transitions between different abstraction layers (e.g., between Model, Software and Hardware in the Loop - MiL, SiL, HiL), impacts also the way in which engineers currently design new functions in the powertrain domain.

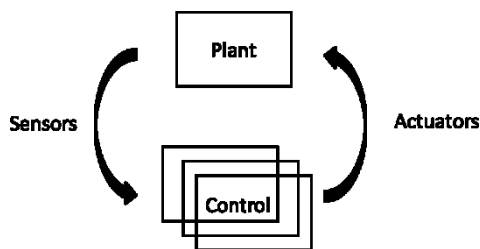


Figure 1. Generic picture of an embedded system, consisting of plant and control. The plant is typically an heterogeneous multi-domain system.

Development of e.g. injection and ignition control for fuel consumption optimization strongly relies on high fidelity models described in specific authoring tools (e.g., ASCET [1] or Dymola [2]). Provided the heterogeneity of modeling tools with specific strengths and following different modeling paradigms, it is often a challenge to bring together a detailed engine model with a sophisticated control algorithm, while fulfilling tight synchronization requirements. Once the models are available and their interaction is well-described, it is not a trivial step to refine them from MiL to the SiL setup (see Figure 2). In MiL, the focus

is on the control algorithm, on its functionality. In SiL, the functionality of the model is extended to the complete software which will be running on the ECU. Additionally to it, an abstraction of the target platform is considered. In SiL, the functionality and processing of data can be tested. Depending on the quality of the abstraction, timing analysis can be conducted with certain computational costs.

With this work, we present a sustainable and scalable solution on how to bring together artifacts coming from multi-domain simulation tools (scalability within the system under investigation), providing a seamless transition from MiL to SiL in the same environment (sustainable and flexible switch between development phases).

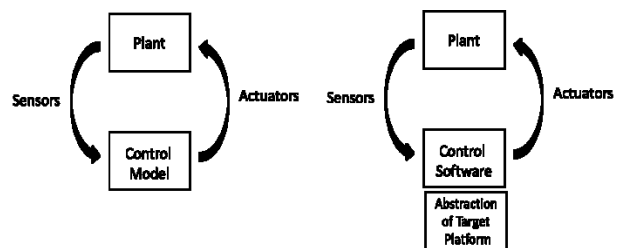


Figure 2. Generic picture of two development phases: Model in the Loop – MiL (left), and Software in the Loop – SiL (right). For the MiL case, only the functionality of the control model together with the plant model is in focus, while in the SiL phase the complete software of the control like it will be running on the platform is considered, again in composition with the plant model. Despite the Hardware in the Loop case, where the control software is running on the target platform, in SiL an abstraction of the target platform is considered, e.g. an instruction set simulator (ISS).

2 AUTOSAR and FMI: Enablers for Virtual ECU Development and standard-based Integration, Simulation and Analysis

There are two main pillars on which we built our solution, namely the AUTomotive Open System ARchitecture (AUTOSAR, [4]) and Functional Mockup Interface (FMI, [3]) standards, for virtual ECU development and standard-based integration, simulation, and analysis, respectively.

The experience over the past decades has shown that proprietary implementations for interoperability and exchange of models bare huge disadvantages: lacking scalability, strong binding to authoring tools, prohibitive effort to co-simulate models from multi-domain tools. The increasing complexity of systems requires dedicated concepts for integration, with a clear distinction between modeling and integration activities. The FMI standard aims at overcoming those barriers across domains, while AUTOSAR was specifically designed to facilitate the simple exchange of individual software elements and to separate the hardware dependent and independent software parts. The Multi-Domain Mockup (MDM) tool, a product developed within the Innovation Centre at ETAS, opens the perspective of a system approach to the embedded system engineer, built upon the FMI standard. It might be either the control model or the virtual ECU which can be integrated with plant models yielding a system, which can be simulated, validated, tested, or parameterized.

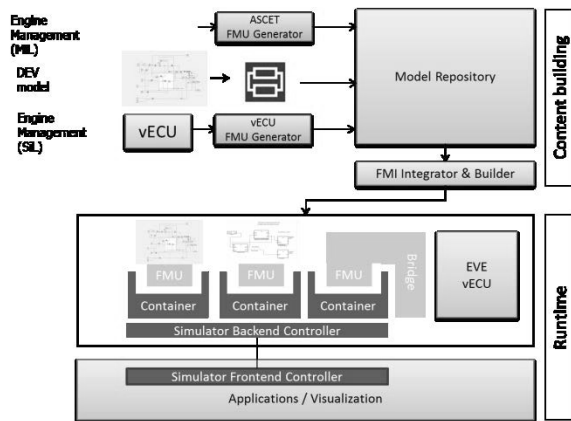


Figure 3. ETAS solution to support the MiL and SiL use cases within same integration, simulation and analysis environment.

With Isolator-EVE, a new product in the ETAS portfolio, the standardized development of ECU software is made possible on everyone's desktop, with all its

advantages, from scalable computational power to debugging features. Figure 3 gives a representation of generic workflows in MiL and SiL: control and plant (Driver, Vehicle, Environment) models, together with virtual ECUs are loaded like FMUs (Functional Mockup Unit, an entity implementing the Functional Mockup Interface) to a collaborative content platform (extended notion of a model repository). The embedded systems engineer is able to search, select and integrate models from the content platform. The integrated system can be simulated. Further analyses can be conducted, e.g. debugging, parameterization, or simulation data post-processing.

3 FMU Generation for ASCET and ETAS Virtual ECU

Each artifact which is integrated within MDM should be FMI compliant. ETAS develops FMU generators for ASCET models and for ETAS Virtual ECU (Isolar-EVE [5]). A highlight of the ETAS FMU generators is the preservation of task scheduling within the FMU. Periodic and crank-angle synchronous tasks can be triggered using the interface functions. Sophisticated non-standard features such as time-discrete ports or interrupts are already implemented in this solution. ETAS aims at extending the FMI standard to include these features in upcoming versions of the standard.

4 FMI-based Integration, Master Algorithm, and Suitable Solvers

Each simulation environment based on the FMI standard should provide a master algorithm which orchestrates the entire simulation, e.g. the order of calculation, synchronizes the simulation step sizes, or iterates discrete events. MDM provides a solution for the powertrain cosimulation, comprising of variable-step solvers with the capability to identify and react to zero-crossing events, and a master algorithm which enables time and crank-angle computational grids. In this way, we succeeded splitting the system in FMUs, where the sensor and the tasks consuming sensor information are in different components.

5 Conclusion

The solution provided by Isolator-EVE and MDM offer already a good solution for the seamless transition from MiL to SiL, even under consideration of the

stringent synchronization and timing constraints, which are then indispensable for the further development of a HiL system enabling to run a real engine in a dynamic mode. While the activities in MiL and SiL are not evolving under strict real-time requirements, it will be interesting to investigate if our approach can be propagated in an equivalent manner to the HiL setup, based again solely on the FMI and AUTOSAR standards.

6 References

- [1] ASCET, Model-based Development of Application Software and Automatic Code Generation for Models,
http://www.etas.com/en/products/ascet_software_products.php
- [2] Dymola, System Simulation Tool based on Modelica, <http://www.3ds.com/products-services/catia/capabilities/systems-engineering/modelica-systems-simulation/dymola/>
- [3] Blochwitz, T., Otter, M., Akesson, J., Arnold, M., Clauß, C., Elmqvist, H., Friedrich, M., Junghanns, A., Mauss, J., Neumerkel, D., Olsson, H., Viel, A. (2012). Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. In M. Otter D. Zimmer (eds.), Proceedings of the 9th International Modelica Conference, 173-184. Linköping University Electronic Press, Linköping, Sweden.
- [4] Kirschke-Biller, F., Fürst, S., Lupp, S., Bunzel, S., Schmerler, S., Rimkus, R., Gilberg, A., Nishikawa, K., Titze, A. (2011). AUTOSAR - A worldwide standard current developments, roll-out and outlook. In 15th International VDI Congress Electronic Systems for Vehicles. Baden Baden, Germany.
- [5] IsolarEVE,
http://www.etas.com/de/products/isolar_eve.php.

A model for airline personnel schedule simulation

Patrick Kuckertz¹, Hubert Randerath²

¹Institut für Informatik, Universität zu Köln

²Institut für Nachrichtentechnik, Fachhochschule Köln

kuckertz@informatik.uni-koeln.de

The personnel of an airline is one of its most important resources, contributing significantly to the business success. During the personnel planning process every step from corporate deployment structures to individual task schedules is determined, ensuring the quality and reliability of flight plan implementations. Although there are simulation models supporting this process, none of them is specialized in describing individual personnel members.

In this paper a stochastic event-based model for the simulation of individual airline personnel schedules is presented. Implemented as a simulation application it allows the robustness assessment of individual personnel schedules under the influence of disruptions and recovery procedures. It is compared with three of the most sophisticated air traffic simulation approaches.

1 Introduction

Personnel planning is a complex task for airlines. Many constraints have to be met and multiple demands and requirements have to be considered. Airlines, labor unions, and federal aviation authorities are the major stakeholders in this process. Personnel schedule planning is a part of a greater planning process conducted after flight schedule design, fleet and tail assignment, and before operational flight plan management and recovery. Typically personnel schedule design consists of crew pairing and crew assignment, meaning that first single tasks are aggregated into work packages, which are then assigned to crews (see [3]). The exploration of optimization procedures to solve crew schedule planning and especially the crew pairing problem is very well advanced. As in other steps of the airline planning process simulation methods are applied to evaluate and improve the robustness of optimization results. However, no specific airline personnel simulation models have been developed, but existing general models were applied. This paper presents a model which was developed for the simulation of individual airline personnel schedules.

A summary of the necessary airline personnel planning background is given in section 2, followed by a presentation and a comparison of three of the most sophisticated air traffic simulation approaches in section 3. Section 4 comprises a detailed description of the newly developed personnel schedule simulation model and a short insight into other elements of

the DOGS (Dynamic Optimization of Group Schedules) project. A summary of this paper and a prospect of future work are given in section 5.

2 Background

In this paper the term airline personnel refers only to the flight personnel used to implement the airline's offered flights on board of airplanes. Usually a plane's crew is divided into a flight deck crew and a cabin crew. While flight deck crews can consist of captains (CPT), senior first officers (SFO), and first officers (FO), cabin crews comprise flight attendants (FA) and senior cabin crew members called pursers (P1, P2). The particular compositions of the crew complements depend on different factors, e.g. aircraft type, flight time, and number of passengers. An international flight from Frankfurt to Sao Paulo for example may require one CPT, one SFO, one FO, one P2, one P1 and twelve FAs. In addition to the positions of the crew complement a flight may necessitate further qualifications of the flight personnel, like speaking a certain language, being trained to serve the increased requirements of first class passengers or accommodate guests with respect of their foreign cultural background.

Within the crew pairing planning step the tasks implicated by the flight plan are allocated to daily duties and those again to pairings. Together airlines, labor unions, and federal aviation authorities determine basic labor conditions which have to be respected in this context. This regards work breaks and rest peri-

ods, work load limits, e.g. in terms of maximum working and flying times, and maximum numbers of take-offs. Besides the tasks on board of airplanes a schedule comprises reserve and standby duties at airports. Standby duties are performed by reserve personnel who can be deployed in cases of disruptions within the daily flight operations. Depending on their response times different types of standby services can be distinguished. A normal standby service usually takes place at the end of a pairing and lasts about two hours in which the personnel can respond to disruptions. While on ‘Reserve 12’ or ‘Reserve 24’ service the personnel is bound to arrive at their home base within 12 or 24 hours respectively upon notice.

3 Existing approaches

Within the field of air traffic simulation a broad range of applications facilitating a variety of purposes can be found. Depending on the detail levels of their underlying air traffic models they can be divided into three classes: microscopic, mesoscopic, and macroscopic simulation applications. Due to their richness of detail microscopic models can reproduce individual movements of an airplane in the air and on the ground. Macroscopic models on the other hand have a low detail level and are for example used to evaluate the performance of an entire air transportation system neglecting individual flights along the way. Mesoscopic models lie in between and have an appropriate level of detail for the purpose of simulating an airline’s planning decisions. Three of the most notable mesoscopic air traffic simulation approaches are presented below, followed by a comparison and a discussion regarding their applicability for airline personnel schedule simulation.

3.1 SIMAIR

SIMAIR (Simulation of Airline Operations) is a research tool developed within a cooperation between the Georgia Institute of Technology and the National University of Singapore (see [5] and [6]). SIMAIR is written in object oriented C++ and allows the simula-

tion of airline operations following a discrete event approach. It includes unexpected disruptions during daily operations leading to a more realistic behavior of the underlying air traffic model. The tool was designed to facilitate the research of airline recovery policies. It serves as a means for quantifying and evaluating the effectiveness of various recovery mechanisms to handle disruptions. It provides an environment in which the performance of new strategies can be tested before being implemented in the field. SIMAIR can also be applied to analyze the robustness of given flight plans and crew schedules.

The SIMAIR simulation tool has a modular architecture (see figure 1). The simulation module is the centerpiece of the application and contains a stochastic model describing an airline’s operations in terms of states and transitions. It defines entities with their characteristics, events and their probabilities, and processes which are needed to simulate operational schedules. For each occurring event the controller module is prompted to check for current and future disruptions, e.g. caused by weather, technical failures or staff shortages. If a disruption has been detected, it is passed on to the recovery module which tries to compensate by implementing recovery measures like rerouting passengers or deploying reserve crews. After reviewing those schedule changes and thereby guaranteeing the flow balance, the accordance of arrival and departure locations of planes, crews, and passengers, the controller module transfers them to the simulation module.

The model describes the entities needed to simulate the operations of a particular airline. It contains all legs of the flight schedule with their scheduled departure and arrival times and all origin and destination airports. An airport is mapped by considering its runways, taxiways, and gates. While the runways are modeled as queues with infinite capacity, taxiways and gates are modeled as time consuming processes and location values for planes respectively. Since airports represent resources shared by other airlines, the other airlines’ influence on airport congestion in



Figure 1. Modular simulation architecture of SIMAIR

daily operations is indirectly expressed by reduced service rates of runway queues at peak times, implemented by time-dependent semi-Markov processes. Airplanes are modeled as means for crews and passengers to move from one airport to the other, following a rotation of flight legs. Although no further data about plane types have been published, priority was set on the need for maintenance in certain time intervals. While some crews complete their pairings on board of airplanes including rest intervals and layovers, others serve as reserve personnel at airports. Passengers are divided into groups according to their itineraries. As for the influence of other airlines the model accounts for weather effects by service rate changes at airports.

In the air transport modelling basically two complementary processes can be distinguished: the off-blocks process, including everything between a plane leaving a gate of an airport and arriving at a gate of another airport, and the ground (or turnaround) process, including everything between a plane arriving at a gate and leaving it. The SIMAIR model places the emphasis on the off-blocks process, whereas it greatly simplifies the ground process. Because the model does not consider the many sources of ground time delays explicitly, an aggregated distribution is used to determine the length of the ground process. The off-blocks process is decomposed into six events and five processes pictured by the event-driven process chain (EPC) in figure 2. The process starts with a departure event and the plane leaves the gate and begins to taxi to the runway. It enters the runway queue and waits until it is allowed to take off. At the end of the flight phase the plane enters the airspace queue of the arrival airport. When it reaches the head of the queue it is allowed to land and to taxi to the gate. The process is concluded by arriving at the gate. Upon the arrival of the plane, the simulation determines if the plane needs maintenance. According to certain probabilities unscheduled major and minor maintenance service events occur, with the duration of the processes depending on the aircraft.

As mentioned above, SIMAIR is applied to determine the performance of operational schedules in combination with different recovery strategies. For this purpose it accumulates a plurality of data. Regarding the evaluation of flight schedules, information about the on-time performance of flight legs divided into different levels of lateness, the number of cancelled flights, and the percentage of delayed or cancelled passenger itineraries is gathered. The crew schedule is assessed on the basis of operational personnel cost, total block time, number of deployed reserve crews, number of deadheads, and the number and degree of planning rule violations. Numbers of departures and arrivals and queue waiting times add to the possibility of single airport analyses.

3.2 MEANS

In 2001 the Massachusetts Institute of Technology started a project to investigate the effects of congestion at hub airports on airline networks (see [1]). During the project, the field of research has expanded to the task of analyzing the combined effects of air traffic control, traffic flow management, and airline operations control on the U.S. national airspace system. The focus lies on the development and evaluation of air traffic management strategies taking airline scheduling and recovery actions into account. To support this research a software application with the name MIT Extensible Air Network Simulation (MEANS) has been developed. Programmed in C++ it provides a discrete event simulation framework which is designed to be easily extendable and adjustable to desired levels of detail. It implements a model of the U.S. air transportation system including airplane, crew, and passenger movements in the air and on the ground.

The MEANS software is based on a modular architecture comprising information, decision making, and state modules (see figure 3). In addition to a single computer's internal communication the modules can exchange data via TCP/IP allowing the MEANS model to be distributed among different hardware and software platforms. The mutual exchange of infor-

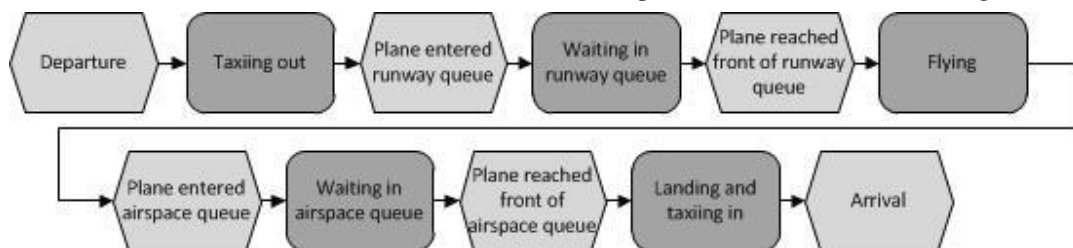


Figure 2. EPC of the SIMAIR off-blocks process

mation is carried out by updating the event queues of the different modules. For some modules up to four implementations were realized, which differ in their levels of detail. This enables users to adapt the simulation network to their individual needs and available data sets.

The weather module is purely an information module, only providing others with current and predicted data and receiving none in return. Depending on the selected implementation it accesses archived weather conditions such as ceiling, visibility, and wind speed and direction, or applies probability density functions to create weather forecasts. The module provides weather information at airports, but no en-route weather.

The air traffic control system command center (ATCSCC) module is one of two decision making modules. It monitors the traffic throughput within the considered airspace. Based on the respective arrival rates it determines the ratios of demand to capacity of all airports. Upon the occurrence or prediction of air side congestions it initiates and manages ground delay programs and ground stops. Essentially, this means the reassignment of airport slots to the different airlines.

The other decision making module is the airline module, which represents an airline's operations control and initiates recovery actions after an airspace disruption. Depending on the impact on passenger itinerar-

ies it can decide to swap, delay, and cancel flight legs, accounting for minimum passenger connection times and aircraft seat capacities. A passenger disruption metric is used to determine the airline recovery decision quality. In the most advanced version of the module crew and maintenance constraints are incorporated into the decision process.

There are four modules managing the movements of aircrafts, and implicitly thus of crews and passengers, by updating their states: The en-route module is responsible for movements in the air, while the tower, taxi, and gate modules manage the ground movements at airports. Basically they follow the process shown in figure 2. The airports within the regarded flight network are modeled as sources and sinks of air traffic. Airports outside the system are viewed as one additional airport. The gate module handles a plane's turnaround when it arrives at an airport. The tasks of boarding and deboarding passengers, loading and unloading baggage, cleaning, fueling and catering the plane are considered as a single ground process, its duration depending on the weight class of the plane, the operating airline, and the respective airport. Additional time is added in case of an unscheduled maintenance event. The occurrence of such an event depends on the aircraft type and the time passed since the last check up, its length depending on aircraft and failure type. During the taxi-in and taxi-out processes a plane is subject to the taxi module. Multiple intersecting taxi ways between the gates and the runways

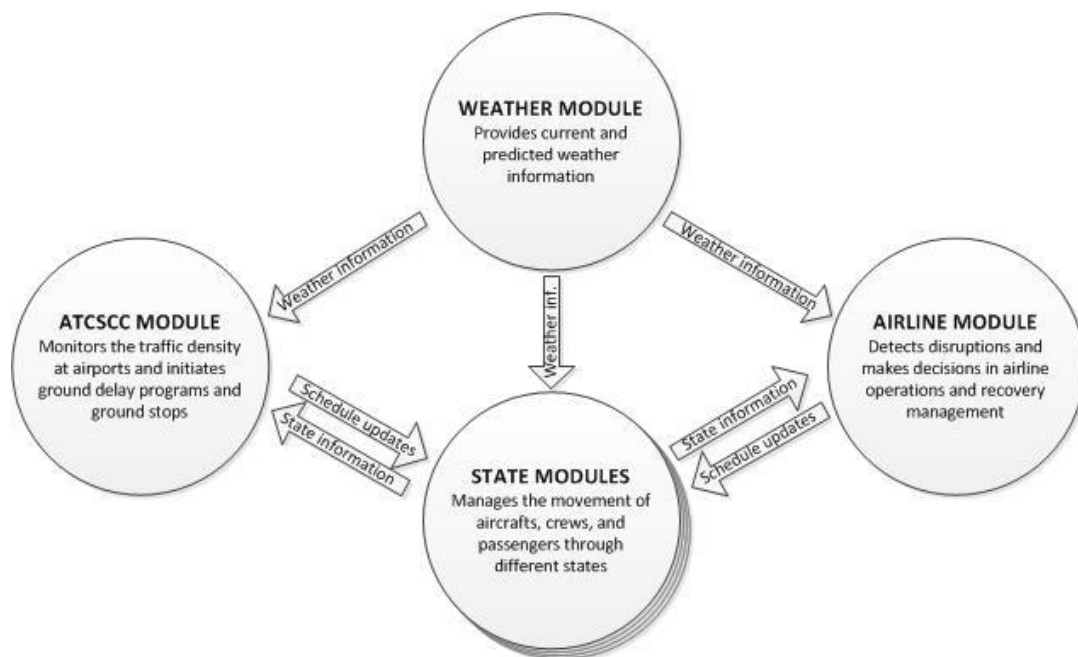


Figure 3. Modular simulation architecture of MEANS

can be modeled. The traffic on taxi ways is accounted for by giving planes the possibility to pass others or to stop for them at intersections, leading to significantly uncertain process durations. Various module implementations provide fix taxi times, or those based on historical or estimated data. Each regarded airport of the system gets an instance of the tower module. Mainly it is responsible for determining the airport's departure and arrival rates, and derived from that the times aircrafts have to wait in the runway landing or take-off queues. Completing the cycle the en-route module describes the off-blocks process of an airplane's flight leg. The flight time is determined by randomly picking a time from a flight time distribution corresponding to the pair of departure and arrival airport.

Considering the user's configuration of the simulation framework, it is clear that MEANS is capable of producing a large amount of data. With regard to developing and evaluating air traffic management strategies the impacts of taken decisions during a simulation run can be analyzed. Recorded state and transition times as well as flight leg and passenger itinerary delays and disruptions provide a sound means to determine the quality of the applied strategies.

3.3 SimOps

As part of a dissertation about robust flight scheduling with stochastic models published in 2007 at the Technical University Clausthal, the event based simulation tool SimOps was developed (see [2]). It implements a stochastic event based flight plan model which is based on detailed information provided by the Deutsche Lufthansa AG. The tool supports the determination of planning times and the allocation and dimensioning of time buffers in order to meet targeted flight schedule punctualities while requiring a minimum of resources. Today SimOps is applied by the Deutsche Lufthansa AG to gauge the effects of various minimum ground times, to dimension the size of reserve fleets, and to evaluate and compare personnel schedule scenarios.

The flight plan simulation model can be implemented in several levels of detail, depending on whether the influences of crew schedules or passenger itineraries are taken into account. In any case, the off-blocks process is of minor interest and therefore greatly simplified (see figure 4). For the taxi, waiting, and flying times an aggregated distribution is applied,

incorporating possible traffic congestions by modeling on and off peak times at airports. The focus of the model lies on the ground process. In order to relate the activities of this process to the respective flight leg, it is divided into the sub-processes flight preparation and flight post-processing (see figure 5). The flight preparation process depicted in figure 6 starts at a specified point of time prior to the planned departure of the plane. Four processes are executed in parallel: fueling, cleaning, catering, and baggage loading. The duration of the fueling depends on the aircraft's type and the scheduled flight time, the duration of the cleaning on the last leg's passenger number, and the catering on the passenger number of the next leg. Although baggage loading and unloading processes have been considered in the model, they have not been implemented in the simulation software. The author explains that those two processes are not part of the critical path and therefore do not have to be taken into account. After fueling, cleaning, and catering are completed the boarding of the crew takes place. The following passenger boarding is either finished when all passengers are in the plane or a certain waiting time has expired. In both cases the preparation process comes to conclusion and when the baggage loading is also finished the plane is ready for departure. The post-processing of a flight leg starts with the plane's arrival at the gate (see figure 7). The passenger deboarding and the baggage unloading begin right away. The crew leaves the plane after the passengers. With completion of these tasks the post-processing is finished. Within a plane's rotation (pictured in figure 8) its idle times between the post-processing of a leg and the preparation of the next is called operational gap.



Figure 4. EPC of the SimOps off-blocks process

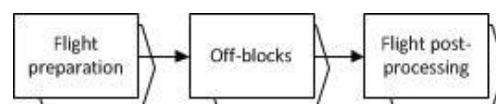


Figure 5. EPC of the SimOps 'plane flies leg' process

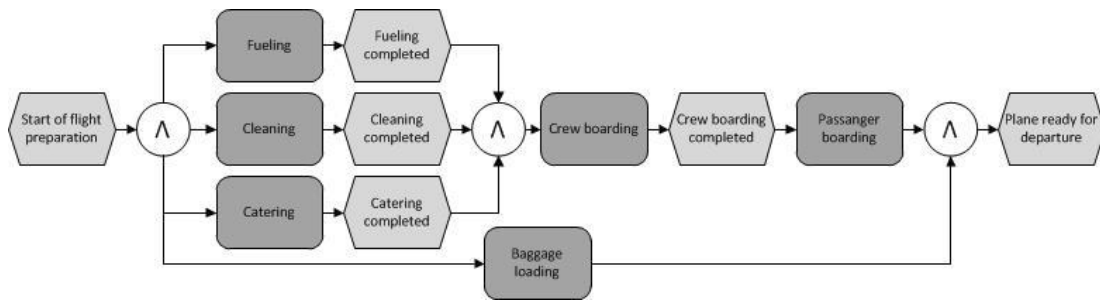


Figure 6. EPC of the SimOps flight preparation process

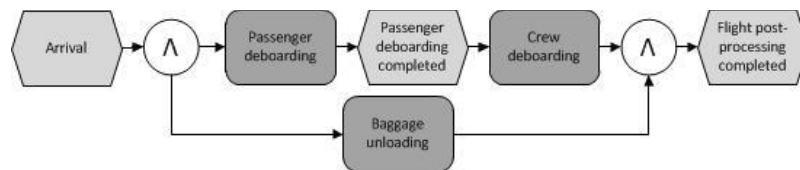


Figure 7. EPC of the SimOps flight post-processing

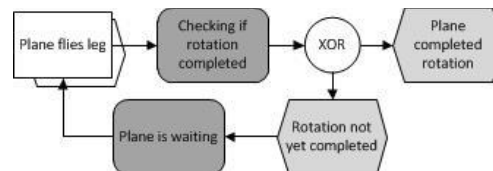


Figure 8. EPC of the SimOps 'plane flies rotation' process



Figure 9. EPC of the SimOps 'crew flies leg' process

The outlined processes are described from the perspectives of aircrafts flying the legs of their rotations. To illustrate the dependencies between the flight legs the simulation model accounts for, a few more processes from the perspective of crews and passengers are presented. The process of a crew completing a flight leg is depicted in figure 9. At the beginning of a crew's rotation (see figure 10) or for plane changes a certain transfer time has to be scheduled. Since a plane is not allowed to start with an incomplete crew, the crew transfer connections between flight legs have significant impact on the spreading of disruptions through the flight network. SimOps possesses a detailed model of airports so that individual distances between arrival and departure positions at gates and on aprons can be considered in this context. The process of passengers flying a leg works in the same manner, with the differences that the model accounts for an average passenger running speed and the pos-

sibility to miss a connecting flight (see figures 11 and 12).

The primary purpose of the SimOps application is to examine the robustness of flight schedules. Over the course of a simulation run it collects data to flight disruptions and creates statistics dividing them into five minute delay intervals. The use of human and aircraft resources is recorded minutely. To further describe a flight plan the tool determines its density and crew synchronism. The density is a metric for how much buffer time was allocated between succeeding flight legs. The crew synchronism metric indicates to what extend aircraft rotations are in unison with crew pairings. The higher the number of plane changes within pairings, the lower a flight plan's crew synchronism. To gather information about the degrees of influence crews and passengers have on a flight plan, simulation runs with and without the respective dependencies can be compared.

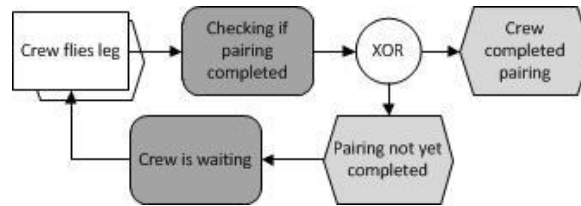


Figure 10. EPC of the SimOps 'crew flies pairing' process

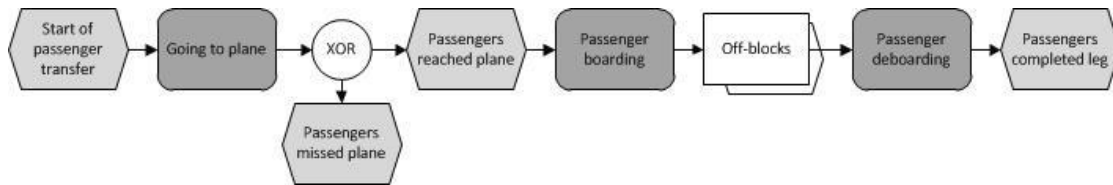


Figure 11. EPC of the SimOps 'passengers fly leg' process

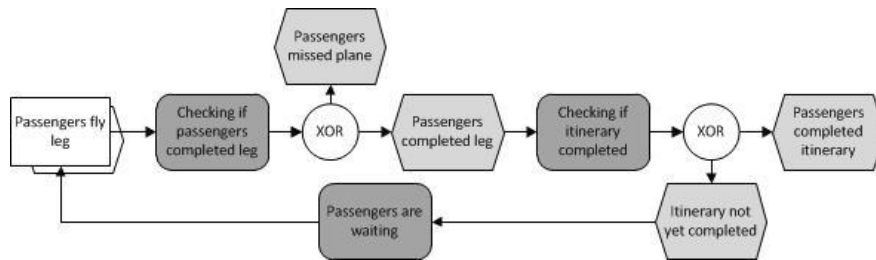


Figure 12. EPC of the SimOps 'passengers fly itinerary' process

3.4 Comparison and discussion

In this section the described models are compared and their suitability for personnel schedule simulation is discussed. Although all three models follow a stochastic event-based approach and are concerned with air traffic simulation, and therefore have much in common, they were designed for different purposes. Five criteria that represent important characteristics for the personnel schedule simulation were selected and applied for the comparison and the suitability evaluation. An overview of the comparison results is given in table 1.

The activities of flight personnel are not limited to the tasks during flight legs, but also include tasks at airports. And although a crew member's flight time can be derived from the off-blocks processes alone, also the ground processes have to be considered to capture total working times, including breaks and rest periods. Additionally the consideration of the ground process offers the possibility to adequately account for personnel dependencies between flight legs. While SIMAIR and MEANS describe the off-blocks

process in detail, defining the same events and processes, they aggregate the ground process to a single task. SimOps provides the complement by focusing on the ground and reducing the off-blocks process (compare figures 2, 4, 6, and 7).

The execution of a personnel schedule is a highly stochastic process and a multitude of causes, e.g. weather phenomena, technical failures, or staff insufficiencies, can lead to disruptions. An airline's recovery management responds to these disruptions and tries to bring the schedule back to feasibility as fast as possible. Recovery procedures directly concerning crew schedules include swapping, deadheading, and rerouting crews as well as deploying those in standby. Due to these facts recovery management must be a part of personnel schedule simulation. MEANS considers recovery actions within the operations control of its airline module, although changes of crew schedules are by-products of flight schedule updates. While SIMAIR provides a wide range of procedures in its recovery module, SimOps forgoes an implementation.

To determine to what extent airline crews were in the focus of the modeling process, the respective project objectives have to be considered. SIMAIR focusses on airline recovery strategies and their influence on the robustness of flight and crew schedules. And though the flight schedules seem to be in the foreground crews are considered to be a separate factor of influence in air traffic operations. That seems also to be the view of SimOps. Primarily investigating the robustness of flight schedules, crews and the implications of intersecting pairings are considered in detail. Only in the MEANS model crews are handled as annexes of flight legs. Since its area of research includes a whole national airspace system focusing on air traffic management strategies, crews seem to be an internal part of airlines with limited influence on air side congestions and overstrained airport capacities.

Beyond the consideration of crews, none of the three approaches goes as far as to model individual flight personnel members. Hence no individual flight, rest, and working times can be recorded, although over the course of a personnel schedule individuals can be part of different crews. Against the background of non-linearly increasing overtime wages and penalties for rest period violations this leads to inaccuracies in personnel cost calculations. On the aggregation level of crews no individual process can be displayed correctly, e.g. the falling ill or late arrival of a crew member. Individual personnel characteristics, e.g. qualifications or seniority, are also ignored.

	Ground process	Off-blocks process	Recovery procedures	Independent crews	Individual personnel
SIMAIR	No	Yes	Yes	Yes	No
MEANS	No	Yes	Yes	No	No
SimOps	Yes	No	No	Yes	No

Table 1. Overview of the model characteristics

4 A model for airline schedule simulation

The stochastic event-based simulation model presented in this section was developed as part of the DOGS project (see [4]). The project started in 2011 as a cooperation between the University of Cologne and the Cologne University of Applied Sciences. It aims for designing methods and implementing software tools for the construction, evaluation, and comparison

of robust airline personnel schedules in connection with staff deployment and recovery strategies.

The planned simulation application will employ a modular architecture resembling the MEANS and the SIMAIR designs, facilitating its maintainability and extensibility (see figure 13). Its core will be the state module, an implementation of the simulation model, performing and extending the tasks of MEANS' en-route, tower, taxi, and gate modules. Like SIMAIR's simulation module it will execute an airline's operational schedules and stochastically generate delays and disruption events. Since weather is an important factor of influence when it comes to simulating operational air traffic schedules, particularly when robustness is considered, a separate weather module will be implemented. It is not intended to simulate weather phenomena themselves but rather to facilitate other modules to simulate their consequences on air traffic. In a first approach it will provide simple weather information in categories like "good weather", "bad weather", and "ice and snow" with each weather event being local and independent from another. The information generated by the module will not only affect airports but also the routes connecting them. It furthermore will not only influence the duration of processes but the process compositions themselves, e.g. whether the de-icing of a plane before a take-off is necessary, or whether a take-off is possible at all. In the future the module might serve as interface to sophisticated weather simulation applications providing as much data as deemed reasonable for this simulation context. The architecture is completed by a recovery module comprising the tasks of SIMAIR's controller and recovery modules, complementing the Airline Module of MEANS. It will detect current and future disruptions and simulate an airline's decisions on how to bring the plans back to feasibility, following an implemented recovery strategy.

The simulation model describes an airline's operations on the basis of states and processes. A state is specified by property values characterizing a set of interrelated physical and logical entities. While physical entities can be touched and seen, logical entities capture abstract concepts. The property values are updated during the execution of processes. Figure 14 gives an overview of the model's entities and their relations, only some of which are described below in more detail to indicate distinct modeling approaches.

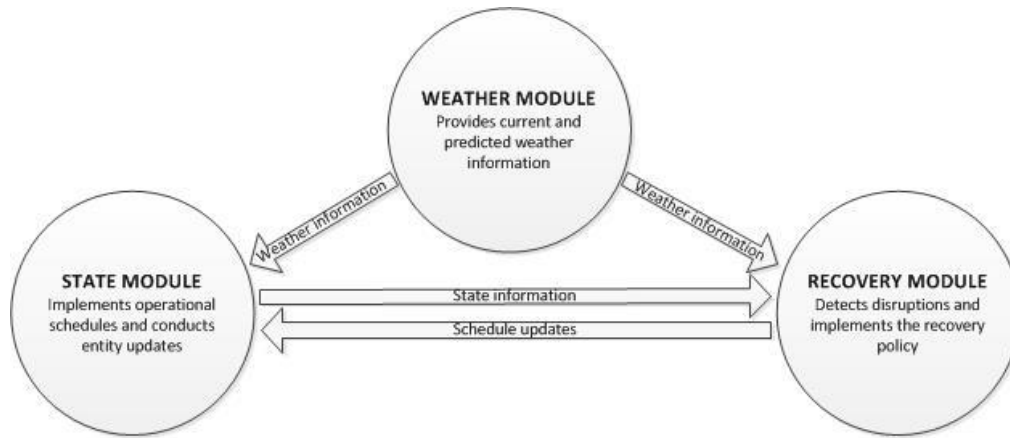


Figure 13. Modular simulation architecture of DOGS application

Following the SIMAIR approach an airport is modeled by a connection of runways, taxiways, and gates. However, congestion is not implemented by the reduction of the service rates of airport resources, but rather by a simplified consideration of foreign airlines' aircrafts within airport boundaries. The detail level of the MEANS and the SimOps models, e.g. the consideration of aircraft behavior on taxiways or individual distances between departure and arrival gates, is not pursued. To incorporate en-route weather and thereby extending the MEANS model, routes get a parameter reflecting current weather conditions. In turn, this parameter affects the durations of respective flight legs or even prohibits their implementations.

Deviating from all presented approaches, an airline's personnel is not modeled as a set of crews, but rather as a set of individual employees, accommodating the fact that employees have individual sets of professional skills. This view allows a more detailed recording of individual flight, work, and rest times as well as a better compliance with work rules and labor legislation. Furthermore, individual capabilities to occupy different positions aboard aircrafts and additional skills in the accommodation of passengers are incorporated into the model as qualifications and ascribed to personnel members. Those qualifications are put in relation to job requirements, which in turn allow a more precise view on the tasks to be per-

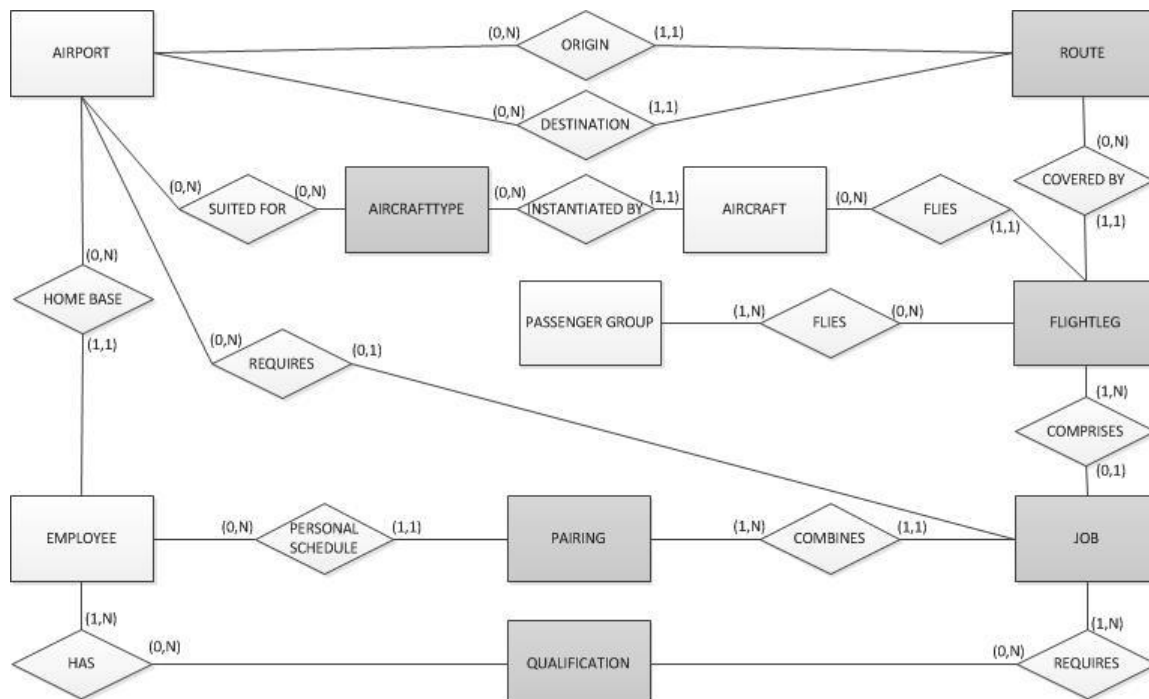


Figure 14. Model entities and their relationships (white: physical entities, grey: logical entities)

formed. The tasks considered in the model, do not only include duties on board of airplanes, but also reserve duties at airports. The differentiated view on reserve duties, e.g. stand-by, ‘Reserve 12’, and ‘Reserve 24’, allows the dynamic build up and reduction of reserve capacities as part of the recovery management.

With regard to the processes, the model includes both the off-blocks process as well as the ground process. Therefore, process models based on a combination of the SIMAIR and MEANS off-blocks process with the SimOps ground process are designed. To allow the weather to influence process compositions, requesting of take-off permissions and de-icing are added to the off-blocks process. The SimOps crew processes can be adapted to consider individual employees without great effort.

5 Summary and future work

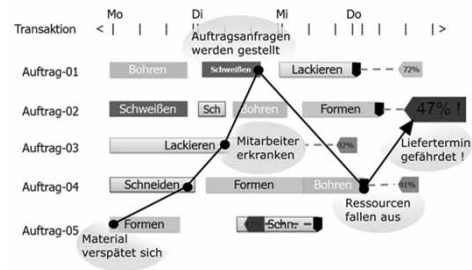
After a short introduction to airline personnel planning as part of a larger planning process, a summary of the necessary technical background was given, including the description of personnel structures, on-board duties and reserve duties, and their implicated requirements. A presentation of the sophisticated air traffic simulation models of the SIMAIR, MEANS, and SimOps applications was followed by a comparison highlighting differences and similarities, and a discussion of their applicability for airline personnel simulation. Based on this information, a stochastic event-based simulation model was introduced, especially suited for the construction, evaluation, and comparison of individual robust airline personnel schedules.

The next steps in the DOGS project will include the implementation of a simulation tool based on the presented model and its application to a variety of test data. Results will then be incorporated into optimization approaches specifically designed for robust personnel planning.

6 References

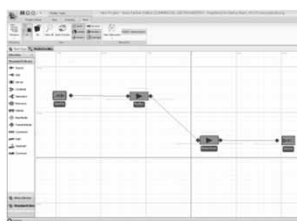
- [1] Clarke, J.-P.; Melconian, T.; Bly, E.; Rabbani, F.: MEANS MIT Extensible Air Network Simulation. In: Sage’s Simulation, Vol. 83, Issue 5, 2007, pp. 385-400.
- [2] Fuhr, B.: Robuste Flugplanung mittels stochastischer Modellierung. Dissertation. Technische Universität Clausthal, 2007.
- [3] Gopalakrishnan, B.; Johnson, E.: Airline Crew Scheduling: State-of-the-Art. In: Annals of Operations Research, Vol. 140, Springer 2006, pp. 305-337.
- [4] Kuckertz, P.; Ullrich, O.; Randerath, H.: A simulation based approach on robust airline job pairing. In: Simulation Notes Europe (SNE), Volume 22, Number 2, August 2012, pp. 77-82.
- [5] Lee, L. H.; Huang, H. C.; Lee, C.; Chew, E. P. et al.: Discrete event simulation model for airline operations: SIMAIR. In: Proceedings of the 2003 Winter Simulation Conference, pp. 1656-1662.
- [6] Rosenberger, J. M.; Schaefer, A. J.; Goldsman, D.; Johnson, E. L. et al.: A stochastic model of airline operations. In: Transportation Science, Volume 36, Number 4, November 2002, pp. 357-377.

3D-Simulation und Scheduling mit Simio



Simulation statt Spekulation!

Multiparadigmen-Software



Modellverfeinerung im Prozess-Flussdiagramm:

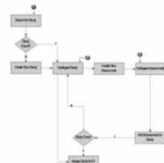
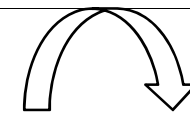


Abbildung: Simio-Objekt-Bibliothek/2D-Modell



Modell direkt in 3D verfügbar

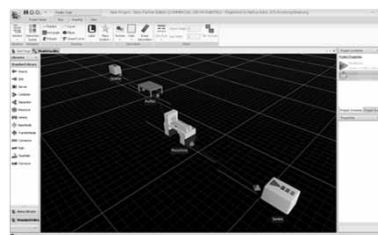


Abbildung: 3D Modell

Woher kommt Simio / Vertrieb

Hersteller ist Simio LLC in Pittsburgh, USA

CEO: Dr. Claude Dennis Pegden

Dieser und sein Team haben Simulationserfahrung von mehreren Jahrzehnten!

Weltweiter Vertrieb über Repräsentanten-Netzwerk

Im deutschsprachigen Raum:

ATS Prozessoptimierung und 4 Partner



Simio: D, A, CH



Mönchengladbach Nähe Düsseldorf

GF, Inhaber: Dipl.-Ing. (FH) Markus Bans

Berufserfahrung auf dem Gebiet der Simulation seit 1996

Früher mit 3 anderen Tools gearbeitet

Master-Repräsentant D, A, CH für Simio:

=> Vertrieb, Dienstleistung, Support, Schulung



Simio-Team Deutschland



Michal Stec, Beratung/ Hotline sowie Vertrieb
Ostdeutschland, Bayern und Polen

Dietmar Böttner, Vertrieb, Hessen, Nord-
deutschland



Hans-Georg Konert, Vertrieb/Beratung, NRW Süd, RP,
Saarland, Schwerpunkt Logistik und SCM

Dan Diulescu, Vertrieb, Baden-Württ.,
Schweiz, Rumänien



Simulations-Referenzen

SIMIO wird in unterschiedlichsten
Branchen und bei Unternehmen
unterschiedlichster Größe eingesetzt:

S. www.simio-sim-aps.de/simio-simulation-fuer-produktion-logistik-und-scm/fallstudienreferenzen

sowie

<http://www.simio.com/applications/>

Simio-Optimierungsmodul

Zusatz-Optimierungsmodul 'OptQuest' kann günstig erworben werden

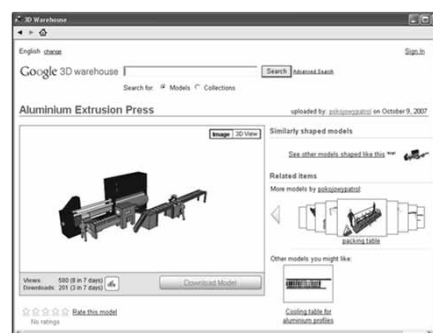
Zweck: Semiautomatische Mini- bzw. Maximierung einer Zielfunktion (Durchlaufzeiten, Gewinn ...) mittels 4 mathematischer Algorithmen, auf Wunsch unter Berücksichtigung von Einschränkungen (Investitionslimit u.a.).

Bsp.: *Maximierung des Gewinns eines Fertigungsbetriebs* durch Verändern der Anzahlen verschiedener Ressourcen (Maschinen, Mitarbeiter, Pufferkapazitäten etc.) bei einem Investitionslimit von X EUR und einem maximalen Absatz von Y Fertigteilen pro Woche.

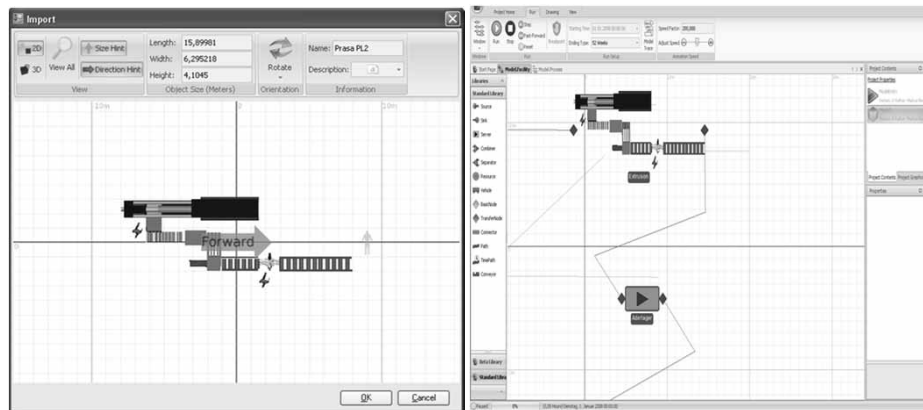


Trimble 3D Warehouse

Nutzung kostenfrei!



Trimble 3D Warehouse



Warum Scheduling?

- > Auftrags**FEIN**planung gegen begrenzte Kapazitäten (*keine* Grobplanung)
- > Visualisierung und Transparenz
- > Spontanes und schnelles Reagieren auf

*Auftragsanfragen
Maschinenausfälle
Erkrankung von Mitarbeitern
Lieferverzögerungen von Rohmaterial ...*



Ihr Nutzen durch Scheduling

- > Weniger Lager- und Umlaufbestand
- > Höhere Liefertreue
- > *Neu*: Risikobasiertes Planen

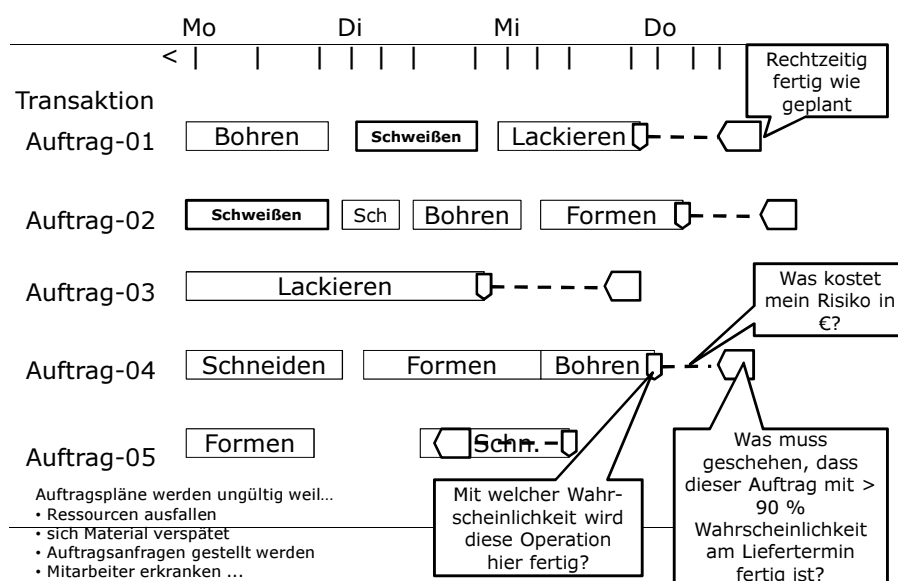


Was kostet mein Risiko?

*Was muss für > 90 % Liefertreue
geschehen?*

Target Liefertermin Risk		Target Kosten Plan		Target Kosten Risk	
Average	Rechtzeitig Probability	Value (Default Currency)	Status	Average (Default Currency)	InBudget Probability
11.10.2011 12:46:48	93,34%	106.674,3392	InBudget	105.580,7687	93,34%
11.10.2011 23:48:27	93,34%	99.922,7025	InBudget	100.559,4272	93,34%
12.10.2011 19:09:20	89,87%	115.034,5469	InBudget	114.549,8604	93,34%
13.10.2011 06:52:17	93,34%	90.802,9469	InBudget	92.384,9407	93,34%
19.10.2011 16:54:35	82,94%	102.014,1803	InBudget	101.791,6140	93,34%

Traditionelle Auftragsplanungssysteme können Auftragsrisiken nicht vorhersagen



Planungsrisiken früher erkennen, um hohe Kosten zu meiden

Mo Di Mi Do

Transaktion

Auftrag-01 Bohren Schweißen Lackieren 72%

Auftrag-02 Schweißen Sch Bohren Formen 47%

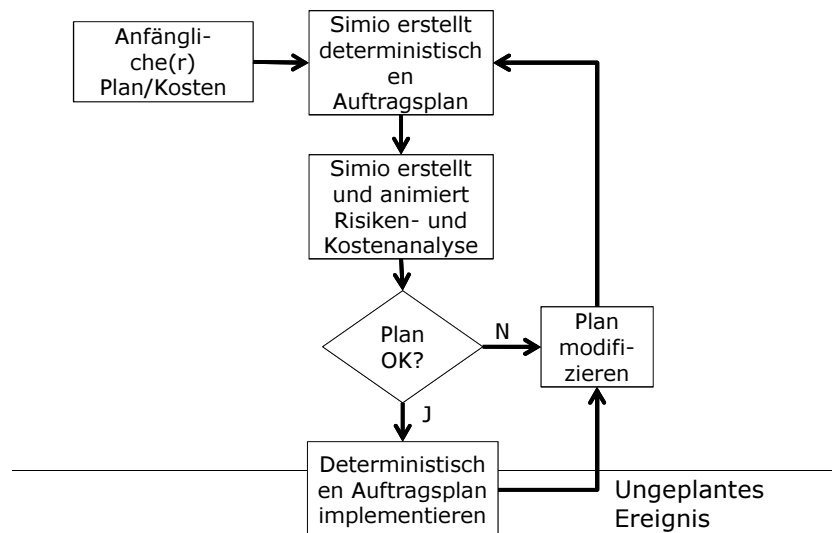
Auftrag-03 Lackieren 92%

Auftrag-04 Schneiden Formen Bohren 91%

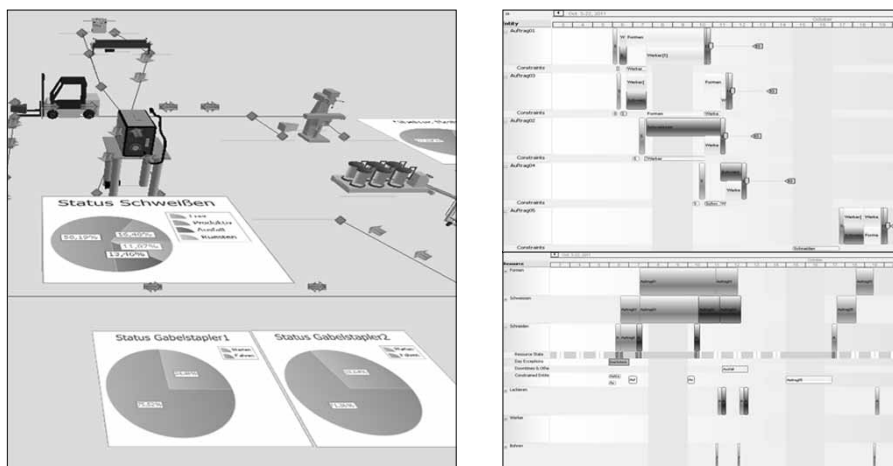
Auftrag-05 Formen 12% Schn.



Riskobasierte Auftragsplanung & -terminierung (RPS) - Flussdiagramm



Nun folgt eine Simio-Demo



Multi-depot multi-vehicle-type vehicle scheduling for Cologne's tram network

Daniel Lückcrath¹, Oliver Ullrich², Aleksander Kupicha³, Ewald Speckenmeyer¹

¹Institut für Informatik, Universität zu Köln

²National Science Foundation's Industry-University Cooperative Research Center for Advanced
Knowledge Enablement, Florida International University

³Prof. Roll & Pastuch – Management Consultants

lueckerath@informatik.uni-koeln.de

To be a feasible base for simulation studies of Cologne's tram network, a valid vehicle schedule has to consider several requirements, like multiple vehicle depots and multiple types of vehicles. The local transport provider utilizes both low-floor and high-floor vehicles, with high-floor vehicles being qualified to serve both high-floor and low-floor platforms. Therefore mixed vehicle rotations are acceptable, but generally not desired. This paper presents a set of models which adhere to these requirements, while also considering several possible optimization goals, like minimum number of deployed vehicles, and minimum combined length of maintenance trips.

1 Introduction

In recent work, some of the authors conducted simulation studies on the influence of robust time tables on punctuality in tram networks, especially in the tram networks of the cities of Montpellier (see [15]) and Cologne (see [13]). A combination of heuristic and exact optimization methods was applied to generate robust time tables, which then were simulated with a microscopic simulation model (see [11]). It has been shown that a tram network has to fulfill a set of structure constraints for robust time tables to have an effect on overall punctuality (see [14]). Up until now, the vehicle schedules, consisting of the assignment of the scheduled trips to a fleet of vehicles, were generated by simple heuristic methods. The resulting vehicle schedules were usually feasible, but were not considering optimization goals like cost minimization or maximizing robustness. They were thus not very realistic and restricting the accuracy of the simulation results.

To address this issue, this paper presents a network flow model and its accompanying integer linear model based on the model introduced in [8], which adhere to the requirements for a feasible vehicle schedule for Cologne's tram network while considering several optimization goals like minimizing the number of deployed vehicles, minimizing the combined lengths of maintenance trips, or minimizing overall cost. A CPLEX implementation of this model is then utilized

to generate such schedules for Cologne's tram network.

This paper continues with sharing some background on vehicle scheduling and recent research on the subject (section 2). Following that, an optimization model for multi-depot, multi-vehicle-type vehicle scheduling for Cologne's tram network is presented (section 3). Several experiments are conducted, demonstrating the adaptivity of the model for different optimization goals (section 4). The paper closes with a short summary of lessons learned and some thoughts on further research (section 5).

2 Background

2.1 Vehicle scheduling

A vehicle schedule R consists of an assignment of scheduled trips $f \in F$ to one of a fleet of vehicles, with $F = F_s \cup F_m$, and F_s the set of planned service trips, and F_m the set of maintenance trips. A rotation $r = (f_{i_1}, f_{i_2}, \dots, f_{i_n})$ for a given vehicle usually starts with a maintenance trip from the depot, where the vehicle is stored, to the start platform of the first service trip. After this service trip the vehicle may continue with a maintenance trip to the start platform of the next service trip, etc. The rotation ends with a return trip to the depot. A vehicle schedule $R = (r_1, \dots, r_k)$ consists of a set of rotations covering all planned trips of an operational day. The vehicle

scheduling optimization problem consists of finding the optimal vehicle schedule R^* , usually regarding minimal cost.

Vehicle scheduling problems are frequently solved using network flow models. **Figure 1** depicts an example of such a model for a simple single-depot vehicle scheduling problem with one vehicle type. The graph can be transformed into an integer linear problem which can then be solved by a software solver like CPLEX.

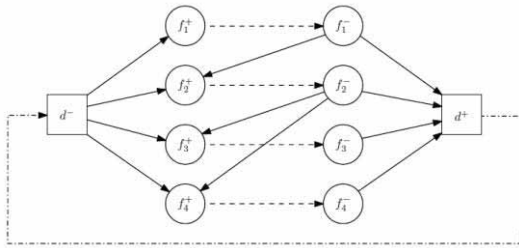


Figure 1. Simple network flow model for vehicle scheduling. Dashed lines are service trips, solid lines represent maintenance trips

As a first step, a given instance is represented by a graph $G(V, F)$ with vertices $v, w \in V$ representing start and end platforms of trips and edges $e = (v, w) \in F$ representing the trips. A cost function $c(e), c: F \rightarrow \mathbb{R}$ maps the cost of each trip, which is usually proportional to the length of the corresponding trip. A depot $d \in D$ is a marked vertex; its capacity λ_d represents the number of vehicles which can be stored in d . There is only a single depot in this example.

Each trip $f \in F$ starts at a platform f^+ and ends at a platform f^- . F is defined as $F = F_{d,f} \cup F_s \cup F_m \cup F_{f,d}$ and thus consists of the maintenance trips $F_{d,f} \subset F_m$ from the depot d to the start of each service trip $f \in F_s$, the regular service trips F_s , the maintenance trips $(f_i, f_j) \in F_m, i \neq j$ from the end of a trip f_i to the start platform of each trip f_j with $f_i < f_j$, and the return trips from the last platform of a rotation to the depot $F_{f,d} \subset F_m$. We define an order of time compatibility on the set of trips: $f_i < f_j$ means f_j can be served after f_i . This order considers the transfer time of a vehicle to get in time for the scheduled departure from the last platform of trip f_i to the first platform of trip f_j . If $f_i \not< f_j$, trips f_i and f_j cannot be served by the same vehicle.

The network flow model is then transformed into an integer linear model (see [16]), as shown in **Table 1**.

Minimize	$\sum_{f \in F} c_f x_f$		(OF)
Subject to	$x_f = 1$	$\forall f \in F_s$	(C1)
	$x_f \in \{0,1\}$	$\forall f \in F_m$	(C2)
	$\sum_{f \in \delta^+(v)} x_f = \sum_{f \in \delta^-(v)} x_f$	$\forall v \in V$	(C3)
	$x_{(d^-, d^+)} \leq \lambda_d$		(C4)

Table 1. Integer linear program for a simple vehicle schedule

The elements x_f of the solution vector x are interpreted as decisions whether a potential trip is covered by a vehicle. Obviously all service trips have to be covered, therefore $x_f = 1$ for all $f \in F_s$ (see (C1)). For each maintenance trip $f \in F_m$ the value of x_f can either be 1, if this trip is covered by a vehicle, or 0, if it is not (see (C2)). Constraint (C3) denotes that the number of outgoing edges $\delta^+(v)$ of a node v which are covered by a vehicle has to be equal to the number of covered incoming edges $\delta^-(v)$. This means that each vehicle which enters a platform has to leave it subsequently. The edge between d^- and d^+ denotes a virtual trip and can be interpreted as a counter of deployed vehicles. Because of (C3) all vehicles leaving the depot at d^+ have to return to it eventually via d^- . Constraint (C4) therefore sets an upper bound to the number of deployed vehicles.

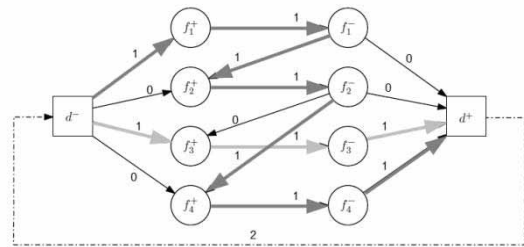


Figure 2. Vehicle 1 executes (f_1, f_2, f_4) , vehicle 2 executes (f_3)

A valid solution to the example is shown in **Figure 2**. The value $x_{(d^+, d^-)} = 2$ denotes that two vehicles are employed: the first vehicle leaves the depot, executes the trips f_1, f_2 , and f_4 , and then returns to the depot;

the second vehicle leaves the depot, executes trip f_3 and then also returns to the depot.

The optimizer finds the vector x which minimizes the objective function (OF) value, and thus yields the combination of rotations with the minimum cost. Because all service trips have to be covered under any such vehicle schedule, this yields a schedule R^* with minimum cost for maintenance trips.

For each instance of this simple vehicle scheduling problem, a network flow model and a corresponding integer linear model can be generated which allows for an optimal solution to be found by the well known methods of integer linear programming (see [16]).

2.2 Related Work

The vehicle scheduling problem has been extensively covered in the past 50 years and several different formulations and approaches exist (see e.g. [1, 2, 3, 5, 6, 7, 8, 9, 10] or, for an overview, see [4]). While the general vehicle scheduling optimization problem is known to be NP-hard (see [1]) some special cases are known to be in P and can be solved efficiently. Gavish and Shlifer in [7] for example use a (quasi-) assignment model to minimize cost resulting from fleet size and maintenance trips for solving the single-depot vehicle scheduling problem with only one vehicle type. Similarly, Bodin et al. in [2] use the network flow approach to convert the single-depot vehicle scheduling problem into a minimum cost flow problem.

More realistic (albeit NP-hard) instances arise when multiple depots and vehicle types are considered. In those cases the problem is often solved using multi-commodity models (as in [1, 8, 10]) or set partitioning formulations (as in [9]). Kliewer, Mellouli and Suhl in [10] for example apply a two-stage aggregation process to reduce the number of decision variables before solving the multi-depot multi-vehicle-type vehicle scheduling problem using a multi-commodity approach. Grötschel, Schöbel and Völker in [8] on the other hand first solve the corresponding single-depot problem before applying heuristic methods to repair invalid rotations, i.e. rotations including service trips not compatible with the respective depot. Hadjar, Marcotte and Soumis in [9] in turn develop a branch-and-bound algorithm combining column generation, variable fixing and cutting planes to solve the problem with the set partitioning formulation.

3 Vehicle scheduling for Cologne's tram network

Cologne's local transport provider utilizes both low-floor and high-floor vehicles based in several depots, with high-floor vehicles being qualified to serve both high-floor and low-floor platforms. Therefore mixed vehicle rotations (i.e. rotations containing both low-floor and high-floor service trips) are acceptable to some extent, but generally not desired. As a result a feasible vehicle schedule for Cologne's tram network has to consider several requirements, like multiple vehicle depots and multiple types of vehicles. A feasible model should also enable several optimization goals: a minimum number of employed vehicles (as acquisition and maintenance of vehicles is expensive), a minimum combined length of connecting trips (as too many non-service trips congest the network), minimum overall cost, or a balance of those.

The simple model shown in section 2.1 does obviously not accommodate those requirements, but it can be utilized as a starting point to build a more complex model.

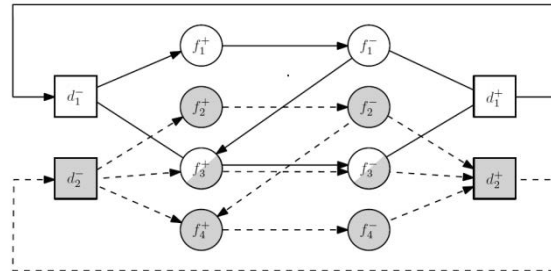


Figure 3. Multi-depot vehicle scheduling

To accommodate for multiple depots, we use a multi-commodity model based on the one presented in [8] (see **Figure 3**), which allows for several depots $d \in D$, with each depot storing only one type of vehicles. The set F^d denotes the service trips a depot d can serve, thus considering multiple vehicle types (indicated by node color in **Figure 3**). Set $D^f \subseteq D$ denotes the subset of all depots from which a service trip f can be served. There exist trips which can be served by several depots and vehicle types (trip f_3 in **Figure 3**), therefore $F^{d_1} \cap F^{d_2}$ will typically not be empty.

Service trips have to be covered under each valid vehicle schedule. Thus, the combined cost of service trips can be considered constant, it can therefore be parametrized. The resulting compressed model (see

Figure 4) manages on less decision variables and can thus be computed faster.

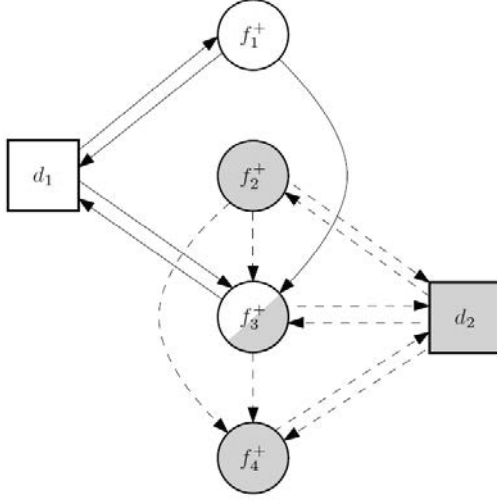


Figure 4. Multi-depot vehicle scheduling with compression

Table 2 shows an integer linear program for the multi-depot multi-vehicle-type vehicle schedule problem. Here, the elements x_{f_1, f_2} of the solution vector x are interpreted as decisions whether a potential maintenance trip between the end platform of trip f_1 and the start platform of trip f_2 should be covered by a vehicle.

Min.	$\sum_{d \in D} [\sum_{f \in F_s^d} ((v_{(d,f)} + c) * x_{(d,f)})$	(O1, O2)
	$+ \sum_{f_1 \in F_s} \sum_{f_2 < f_1} (v_{(f_1, f_2)} * x_{(f_1, f_2)}^d)$	(O3)
	$+ \sum_{f \in F_s^d} (v_{(f,d)} * x_{(f,d)})]$	(O4)
S.t.	$\sum_{d \in D} (x_{(f_1,d)} + \sum_{f_2 < f_1} x_{(f_1, f_2)}^d) = 1 \quad \forall f_1 \in F_s^d$	(C1)
	$[x_{(d,f_1)} + \sum_{f_2 < f_1} x_{(f_2, f_1)}^d - \sum_{f_1 < f_2} x_{(f_1, f_2)}^d - x_{(f_1, d)}] = 0 \quad \forall f_1 \in F_s^d$	(C2)
	$0 \leq \sum_{f \in F_s^d} x_{(d,f)} \leq \lambda_d \quad \forall d \in D$	(C3)
	$x_{(i,j)} \in \{0,1\} \quad \forall i,j \in D \cup F_s$	(C4)

Table 2. Integer program for a multi-depot multi-vehicle-type vehicle schedule

The objective function considers (O1) the fixed cost c of a vehicle's deployment, (O2) the cost of the first

maintenance trip from the depot to the first platform of its first trip, (O3) the combined cost of the maintenance trips connecting service trips, and (O4) the return trip to the depot from the last platform of the last service trip.

Constraint (C1) guarantees that for every service trip f only one of the possible succeeding trips is selected. Together with the network flow conservation constraint (C2) this guarantees that each trip is covered by at most one vehicle and has only one preceding trip. Constraint (C3) guarantees for each depot a number of deployed vehicles that is within this depot's capacity, while (C4) guarantees that each potential maintenance trip is either covered by a vehicle or not.

Several optimization goals can be reached by varying the fixed cost c : If c is set to a value greater than the maximum length of maintenance trips v_{max} , then executing maintenance trips is generally preferred to deploying another vehicle, thus minimizing the total number of deployed vehicles. If $0 < c < v_{min}$, then the model prefers deploying another vehicle to executing any maintenance trips with length greater than zero, thus minimizing the combined lengths of maintenance trips. By sweeping fixed cost c between v_{min} and v_{max} a trade-off between number of deployed vehicles and lengths of maintenance trips may be observed.

4 Experiments

4.1 Modeling Cologne's tram network

We apply the developed model to our hometown Cologne's tram network based on the time table data of 2001 (see **Figure 5**). It consists of 528 platforms and 58 track switches connected via 584 tracks. These tracks cover a total length of 407.4 kilometers, resulting in an average track length of 697.6 meters. 15 lines with 182 line routes are served by 178 vehicles which execute 2,814 trips per operational day. The vehicles are stored in three maintenance depots, two of them store high-floor vehicles (near stations Aachener Straße/Gürtel (ASG) and Niehler Straße/Gürtel (NSG)), and one stores low-floor vehicles (near station Kalk Kapelle (KKP)).

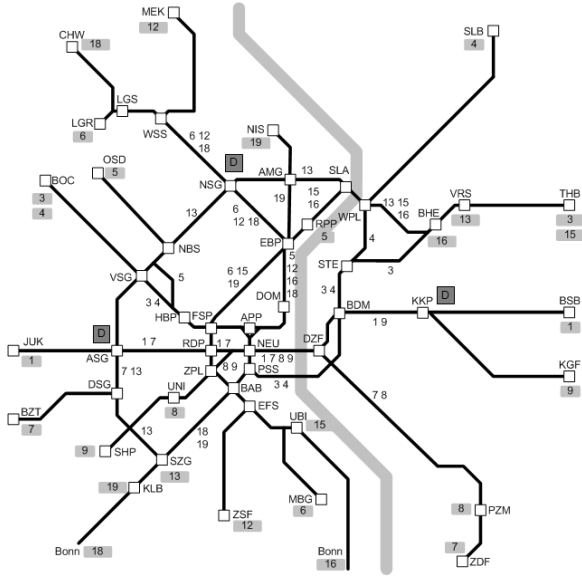


Figure 5. Cologne's tram network

As test scenario we chose the tram schedule of 2001 from 3 am to 12 pm, which covers most of the planned service trips of a typical operational day. The described instance is solved via the CPLEX software package. We conduct five experiments:

(E1) The fixed cost is set to a value $c > v_{max}$ to minimize the number of deployed vehicles. For Cologne's tram network v_{max} is established as $v_{max} = 41.907$ kilometers, describing the distance from station Chorweiler (CHW) to station Bad Godesberg Stadthalle in the neighboring town of Bonn. The fixed cost are accordingly set to $c = 41.908$.

(E2) The fixed cost is set to a value $c < v_{min}$ to minimize the combined length of maintenance trips. The value of v_{min} is established as $v_{min} = 1.342$ kilometers, occurring between stations Zollstock Südfriedhof (ZSF) and Klettenbergpark (KLB). For this experiment the fixed cost is set to $c = 1.341$.

(E3) A sweep over $v_{min} \leq c \leq v_{max}$ is conducted to explore the trade-off between the number of deployed vehicles and the lengths of maintenance trips.

(E4) Up until now, low-floor platforms could be served by both high-floor and low-floor vehicles. For this experiment we explicitly forbid mixed vehicle rotations, which results in two separated problem instances.

(E5) This experiment allows mixed vehicle rotations, but sets a penalty by doubling the cost of low-floor

trips served by high-floor vehicles. For (E4) and (E5) c is again set to 41.908.

4.2 Results and discussion

Table 3 shows the results of experiments (E1) and (E2). Setting the fixed cost to a value greater than v_{max} results in a vehicle schedule with 109 vehicles serving 18.72 service trips on average. In comparison, a fixed cost value less than v_{min} raises the number of utilized vehicles by 4.4 percent (or 5 vehicles) to 114 vehicles in total, which serve 17.91 service trips on average. By utilizing more vehicles variable cost can be lowered by 1.05 percent.

	(E1): Minimizing number of vehi- cles	(E2): Minimizing length of mainte- nance trips
Run time	2,470 s	1,037 s
Overall cost	6,007	1,577
Fix cost	4,567.86	152,87
Variable cost	1,439.14	1,424.13
Vehicles	109	114
μ_L	18.72	17.91
σ_L	8.79	8.15
\min_L	6	6
\max_L	50	46

Table 3. Results of experiments (E1) and (E2). μ_L and σ_L denote average rotation length and standard deviation. \min_L and \max_L denote minimum and maximum rotation length

A general trade-off between the number of vehicles and the length of the maintenance trips is highlighted by the results of experiment (E3) (see Figure 6). For $0 \leq c \leq 5.4$ a reduction of vehicles is compensated by longer maintenance trips. For fixed cost values of $c \geq 5.4$ both the number of utilized vehicles and the length of the maintenance trips stagnate, indicating that it is not possible to serve all planned service trips with fewer vehicles.

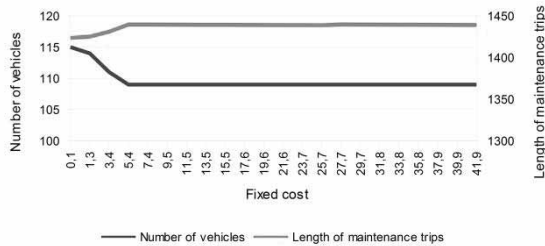


Figure 6. Results of experiment (E3)

The results of the last experiments (E4) and (E5) are shown in **Table 4**. As expected, banning mixed rotations reduces the set of valid solutions and subsequently results in a less efficient vehicle schedule compared to the solutions from experiments (E1) and (E2). On the other hand, penalizing mixed rotations in (E5) does not result in significant changes compared to (E1).

	(E4): No mixed rotations	(E5): Penalty for mixed rotations
Run time	436.92 s	7,650.91 s
Overall cost	6,592	6,251
Fix cost	4,651.68	4,567.86
Variable cost	1,940.32	1,683.14
Vehicles	111	109
μ_L	18.10	18.66
σ_L	9.27	8.08
\min_L	4	6
\max_L	51	52

Table 4. Results of experiments (E4) and (E5)

5 Summary and further research

In this paper, we shared an optimization model to generate multi-depot, multi-vehicle-type vehicle schedules for Cologne's tram network. This model can be tuned to consider optimization goals like minimizing the number of deployed vehicles, minimizing the combined lengths of maintenance trips, or minimizing overall cost. Several series of experiments showed the applicability of the model while exploring its tuning capabilities.

In a further step, the described model will be applied to generate vehicle schedules for given time tables,

which in turn are generated by the optimization tools described in [12] and [13]. These combined schedules will then be simulated with the simulation engine described in [11] to further validate their applicability.

6 Acknowledgements

This material is based in part upon work supported by the National Science Foundation under Grant Nos. I/UCRC IIP-1338922, AIR IIP-1237818, SBIR IIP-1330943, III-Large IIS-1213026, MRI CNS-0821345, MRI CNS-1126619, CREST HRD-0833093, I/UCRC IIP-0829576, MRI CNS-0959985, FRP IIP-1230661 and U.S. Department Transportation under a TIGER grant.

7 References

- [1] Bertossi, A. A., Carraraesi, P., Gallo, G.: On Some Matching Problems Arising in Vehicle Scheduling Models. In: Networks, Volume 17, Number 3, 1987, pp. 271-281.
- [2] Bodin, L., Golden, B., Assad, A., Ball, M.: Routing and scheduling of vehicles and crews: the state of the art. In: Computers & Operations Research, Volume 2, Number 10, 1983, pp. 63-211.
- [3] Borndörfer, R., Löbel, A., Weider, S.: A bundle method for integrated multi-depot vehicle and duty scheduling in public transport. In: Computer-aided systems in public transport. Springer, 2008, pp. 3-24.
- [4] Bunte, S., Kliwer, N.: An overview on vehicle scheduling models. In: Public Transport, Volume 1, Number 4, 2010, pp. 299-317.
- [5] Ceder, A.: Public-transport vehicle scheduling with multi vehicle type. In: Transportation Research Part C, Volume 19, 2011, pp. 485-497.
- [6] Daduna, J. R., Branco, I., Paixao, J. M. P.: Vehicle Scheduling Problem with Multiple Types of Vehicles and a Single Depot. In: Computer-Aided Transit Scheduling. Springer, 1995, pp. 115-129.
- [7] Gavish, B., Shlifer, E.: An approach for solving a class of transportation scheduling problems. In: European Journal of Operational Research, Volume 3, 1978, pp. 122-134.
- [8] Grötschel, M., Löbel, A., Völker, M.: Optimierung des Fahrzeugumlaufs im öffentlichen

- Nahverkehr. In: Mathematik Schlüsseltechnologie für die Zukunft. Springer, 1997, pp. 609-624.
- [9] Hadjar, A., Marcotte, O., Soumis, F.: A Branch-and-Cut Algorithm for the Multiple Depot Vehicle Scheduling Problem. In: Operations Research, Volume 54, Number 1, 2006, pp. 130-149.
- [10] Kliwer, N., Mellouli, T., Suhl, L.: A new solution model for multi-depot multi-vehicle-type vehicle scheduling in (sub)urban public transport. In: Proceedings of the 13th Mini-EURO Conference and the 9th meeting of the EURO working group on transportation, 2002.
- [11] Lückcrath, D., Ullrich, O., Speckenmeyer, E.: Modeling time table based tram traffic. In: Simulation Notes Europe (SNE), ARGESIM/ASIM Pub., TU Vienna, Volume 22, Number 2, August 2012, pp. 61-68.
- [12] Ullrich, O.: Modellbasierte Parallelisierung von Anwendungen zur Verkehrssimulation - Ein dynamischer und adaptiver Ansatz. Dissertationsschrift, Univ. Köln, 2014.
- [13] Ullrich, O., Lückcrath, D., Franz, S., Speckenmeyer, E.: Simulation and optimization of Cologne's tram schedule. In: Simulation Notes Europe (SNE), ARGESIM/ASIM Pub., TU Vienna, Volume 22, Number 2, August 2012, pp. 69-76.
- [14] Ullrich, O.; Lückcrath, D.; Speckenmeyer, E.: Reduzieren robuste Fahrpläne Verspätungen in Stadtbahnnetzen? - Es kommt drauf an! In: Proceedings of HEUREKA 2014, to appear.
- [15] Ullrich, O.; Lückcrath, D.; Speckenmeyer, E.: A robust schedule for Montpellier's Tramway network. Technical Report, Univ. Köln, 2013, 17 pg.
- [16] Vanderbei, R. J.: Linear Programming - Foundations and Extensions. Third Edition. Springer, 2008.

Modellbibliothek für die Interaktion von Robotern in der MATLAB/DEVS-Umgebung auf Basis des SBC-Frameworks

Birger Freymann, Thorsten Pawletta, Tobias Schwatinski, Sven Pawletta

Hochschule Wismar – University of Applied Sciences: Technology, Business and Design

Research Group Computational Engineering and Automation

{birger.freymann, thorsten.pawletta, tobias.schwatinski, sven.pawletta}@hs-wismar.de

Ausgehend von einer Analyse möglicher Interaktionsprinzipien für Knickarmroboter wird die geometrische Kopplung als ein wesentliches Interaktionsprinzip identifiziert. Nachfolgend wird das *Simulation Based Control* (SBC) Framework zur durchgängigen Entwicklung von ereignisorientierten Steuerungen nach dem *Rapid Control Prototyping* (RCP) Ansatz vorgestellt und auf den *Parallel Discrete Event System* (PDEVS) Formalismus eingegangen. Der PDEVS-Formalismus definiert eine systemtheoretische Spezifikation und Ausführungsalgorithmen für dynamische Systeme. Dieser wird mit dem PDEVS-RCP-Ansatz um RCP-Fähigkeit erweitert. Basierend auf dem SBC-Framework und dem PDEVS-RCP wird die Entwicklung einer Modellbibliothek für geometrisch gekoppelte Roboter in der MATLAB-Umgebung vorgestellt. Abschließend wird diese an einer Applikation demonstriert.

1 Einleitung

Industrieroboter finden in unterschiedlichen Bereichen, von der Automobilindustrie bis hin zur Medizintechnik, Anwendung. Auf dem Markt sind unterschiedliche Hersteller vertreten, die individuelle Roboter und Software zur Programmierung der Roboter anbieten. Zumeist sind die Softwarelösungen herstellerspezifisch und lassen sich nicht für herstellereigene Roboter verwenden. Langjährige Normierungsbemühungen für die explizite Roboterprogrammierung, wie die IRL (Industrial Robot Language) und deren Nachfolger die PLR (Programming Language for Robots), wurden durch die Roboterhersteller bis heute zumeist ignoriert. Diese Gegebenheit führt zu einer Einschränkung bei einer gemeinsamen Nutzung von Robotern unterschiedlicher Hersteller beziehungsweise macht sie unmöglich. Applikationen, bei denen mehrere Roboter interagieren sollen, sind somit zumeist auf die Softwarelösung eines Herstellers begrenzt.

Aufbauend auf dem *Simulation Based Control* (SBC) Framework für die ereignisorientierte Steuerungsentwicklung nach [7] wird ein Ansatz zur Lösung dieses Problems vorgestellt. Der SBC-Ansatz baut auf dem *Rapid Control Prototyping* (RCP) nach [1] auf. Demgemäß unterstützt der SBC eine schrittweise und durchgängig modellbasierte Steuerungsentwicklung von der frühen Planungsphase bis zum operativen Betrieb. Steuerungsentwürfe können sukzessive simulativ getestet und um weitere Anforderungen ergänzt werden. Das SBC-Framework wurde in [5,9] bereits erfolgreich bei der Steuerungsentwicklung für nicht in-

teragierende Roboter unterschiedlicher Hersteller angewendet. Die Integration konkreter Robotertypen erfolgt über eine spezielle Schnittstelle (Interface). Die softwaretechnische Umsetzung des SBC-Frameworks erfolgte in beiden Fällen mit der MATLAB-Umgebung. Die Arbeiten in [5] zeigten, dass die Umsetzung ereignisorientierter Steuerungen mit den Standard-MATLAB-Werkzeugen (Simulink, Stateflow, SimEvents) aufgrund nicht theoriekonformer Implementierungen der Werkzeuge aufwendig ist. Aus diesem Grund wurde basierend auf dem *Discrete Event System* (DEVS) Formalismus nach [10] eine DEVS-Toolbox für MATLAB [2] entwickelt. Diese unterstützt eine systemtheoretischbasierte Modellierung und Simulation ereignisorientierter Systeme. Hinsichtlich eines durchgängigen Einsatzes der Modelle bis zum operativen Betrieb unter Echtzeitbedingungen wurde in [8,10] der DEVS konforme PDEVS-RCP-Formalismus definiert. Im Beitrag werden unterschiedliche Interaktionsprinzipien für Industrieroboter analysiert und aufgrund ihrer praktischen Relevanz die geometrische Kopplung als zu untersuchendes Interaktionsprinzip ausgewählt. Ein typischer Anwendungsfall der geometrischen Kopplung ist das Lastteilungsverfahren (load sharing). Hierbei wird ein schweres Bauteil von mehreren Robotern bewegt. Dies spart im Vergleich zu einem größeren Roboter, der die gleiche Last tragen könnte, Kosten und Energie, bringt aber auch weitere Herausforderungen mit sich. So muss jeder am load sharing beteiligte Roboter individuell bewegt und eingelernt (goteached) werden. Dabei stellt die zeitliche

Synchronisierung eine große Herausforderung dar. Viele der Roboterhersteller haben für solche Problemstellungen individuelle Lösungen erarbeitet. Diese sind aber, wie zuvor erwähnt, nicht uneingeschränkt auf Roboter anderer Hersteller übertragbar. Nach einer Einführung in das SBC-Framework und den PDEVS-RCP-Formalismus werden nachfolgend robotertypunabhängige Modellkomponenten zur Umsetzung von Robotersteuerungen mit geometrischer Kopplung beschrieben und anhand einer prototypischen Problemstellung erläutert.

2 Interaktionsprinzipien von Robotern

Interaktionsprinzipien von Knickarmrobotern, lassen sich basierend auf der allgemeinen Klassifizierung in [4] wie folgt einteilen:

Kommunikation

Bei der Kommunikation besitzen die Roboter getrennte Arbeitsräume und Steuerungen. Die Kommunikation beinhaltet eine gerichtete (uni- oder bidirektionale) nicht abgestimmte Übertragung von Objekten oder Informationen zwischen Robotern. Die Übertragung von Objekten bedingt aufgrund der getrennten Arbeitsräume eine zusätzliche Instanz.

Kooperation

Bei der Kooperation wird eine Überlappung der Arbeitsräume der Roboter vorausgesetzt. Kooperierende Roboter nehmen Veränderungen an Objekten in gemeinsamen Arbeitsräumen vor oder die Steuerungen benutzen gemeinsame Informationen. Die Operationen sind zeitlich nicht abgestimmt.

Koordination

Unter Koordination wird eine abgestimmte Kommunikation oder Kooperation verstanden. Sie ermöglicht die Ausführung von Operationen (Objektveränderungen oder Informationsverarbeitung) durch Roboter in einer bestimmten Reihenfolge.

Mobilität

Allgemein wird unter Mobilität die Änderung von Arbeitsräumen einer Instanz oder von deren Steuerung verstanden. Knickarmroboter können verschiedene Arbeitsräume besitzen, wenn sie auf einem Portal verfahrbar sind. Eine informationstechnische Mobilität ist zum Beispiel ein objektabhängiger Austausch von Steuerungsprogrammen.

Kopplung

Eine Kopplung ist eine Kooperation mit abgestimmten informations- oder prozesstechnischen Operationen (vgl. Abbildung 1). Ein typisches Beispiel in der Robotik ist die gemeinsame Bewegung eines Objektes (z.B. beim Lastteilungsverfahren). Hierbei bedingt die Bewegung eines Roboters (Master) die Nachführung der anderen beteiligten Roboter (Slaves). Die Realisierung derartiger Kopplungen basiert zumeist auf kinematischen Ketten und wird als geometrische Kopplung bezeichnet.

Dynamische Kopplung:

Bei der dynamischen Kopplung können sich die Interaktionen dynamisch ändern. Eine dynamische Kopplung besteht aus einer Folge zeitlich begrenzter unveränderbarer Kopplungen. In der Robotik folgen dynamische Kopplungen zum Beispiel aus veränderten kinematischen Ketten aufgrund von Werkzeugwechseln oder durch sich ändernde Master-Slave-Beziehungen beim gemeinsamen Handling von Objekten.

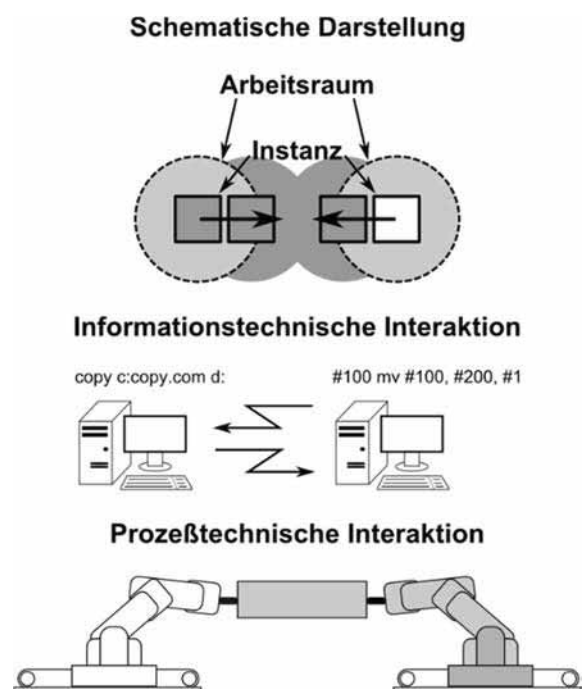
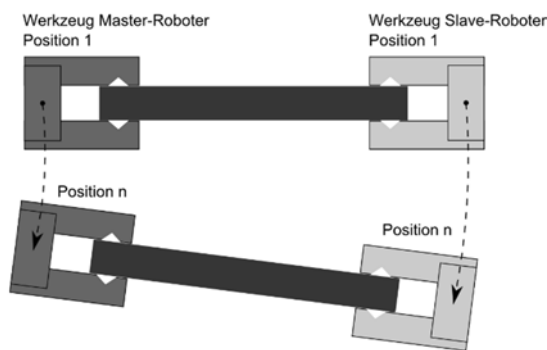


Abbildung 1: Kopplung mit abgestimmten Operationen
nach [4]

In Abbildung 2 sind zwei unterschiedliche Varianten der geometrischen Kopplung dargestellt. Diese stellen Kopplungen mit abgestimmten Operationen dar und sollen in dieser Arbeit näher betrachtet werden. Teil-

bild (a) zeigt schematisch eine *geometrische Kopplung*, bei der die Position des Master-Roboter-Tool-Center-Points (TCP) zur Position des Slave-Roboter-TCP während einer Bewegung konstant bleibt. Der Slave folgt dem Master direkt. Auch bei der *positionsabhängigen geometrischen Kopplung* folgt der Slave dem Master. Der Slave verfügt hierbei zusätzlich über eine eigene Kinematik, die diese Bewegung überlagert.

a) Geometrische Kopplung



e.g. Lastteilungsverfahren

b) Positionsabhängige geometrische Kopplung

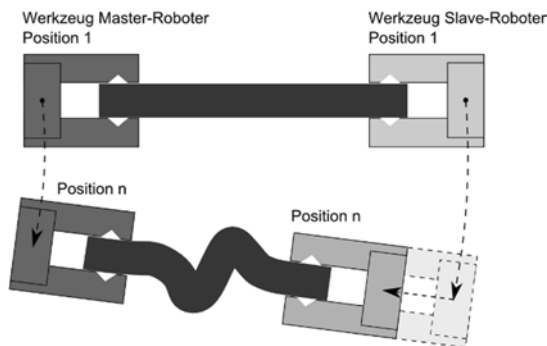


Abbildung 2: Varianten der geometrischen Kopplung

3 Softwaretechnische Grundlagen

Nachfolgend wird das SBC-Framework und der PDEVs-RCP-Formalismus vorgestellt. Sowohl das SBC-Framework, als auch PDEVs bilden die Basis für die Entwicklung der Modellbibliothek.

3.1 Das SBC-Framework

Dieser Abschnitt basiert auf [3,5] und beschreibt das Simulation Based Control (SBC)-Framework. Das SBC-Framework ist eine rechnergestützte Entwurfsmethodik zur Regelungs- und Steuerungsentwicklung.

Der grundlegende Gedanke besteht in der durchgängigen Nutzung von Simulationsmodellen während der gesamten Steuerungsentwicklung (vgl. Abbildung 3).

Das SBC-Framework baut auf dem Rapid Control Prototyping (RCP) Ansatz nach Abel in [1] auf und stellt eine spezielle Form des *Software in the Loop* (SiL)-Prinzips dar. Ein in der Entwurfsphase entwickeltes Simulationsmodell wird schrittweise bis zu einem operativen Steuerungsprogramm erweitert und simulativ getestet. Unter Verwendung der impliziten Code-Generierung können Simulationsmodelle nahezu unverändert für eine reale Steuerung genutzt werden. Damit entfällt die Reimplementierung von Simulationsmodellen in Steuerungscode, wodurch Fehler vermieden werden, Entwicklungszeit eingespart wird und insgesamt die Entwicklungskosten reduziert werden. Um diese Form der durchgängigen Softwareentwicklung zu ermöglichen, bedarf es einer durchgängigen Softwarekette, welche auch als Toolkette bezeichnet wird. Der durchgängige Einsatz von Simulationsmodellen ermöglicht es, Fehler in den Algorithmen frühzeitig zu erkennen und zu beheben. Dabei ist es notwendig, möglichst früh während der Entwicklung, konsequent zwischen dem Control-Modell (CM) mit der Steuerungslogik und dem Process-Modell (PM) mit dem Abbild des realen Prozesses zu unterscheiden. Für die Betriebsphase auf Basis des *Software In the Loop* (SiL)-Prinzips wird ein Interface zum realen Prozess benötigt, welches Sensorwerte des realen Prozesses aufbereitet und umgekehrt Aktorbefehle an die Komponenten des realen Prozesses sendet. Hierfür sollte ein Interface-Modell (IM) entwickelt werden, welches ein simulatives Testen unterstützt und als Schnittstelle zum realen Prozess agieren kann. Nur am Rande erwähnt werden soll, dass die Integration des Process-Modells in die operative Steuerungssoftware die Berechnung nicht oder schlecht messbarer Prozessgrößen ermöglicht und die Realisierung von Beobachterkonzepten unterstützt.

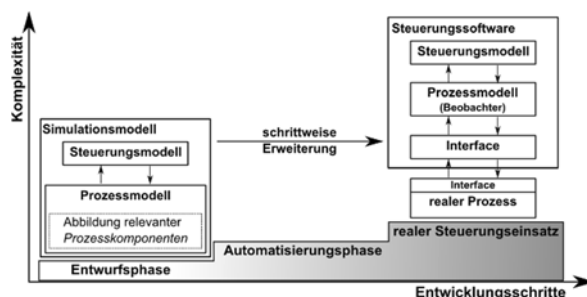


Abbildung 3: SBC-Framework

3.2 PDEVS-RCP und die “DEVS-Toolbox for Matlab”

Die konkrete softwaretechnische Umsetzung der Modellbibliothek erfolgt in der MATLAB-Umgebung auf Basis des Discrete Event System (DEVS) Formalismus. Basierend auf [9] werden die grundlegenden Prinzipien und die darauf aufbauenden Erweiterungen, die in dieser Arbeit verwendet werden, eingeführt.

DEVS wurde 1976 von Zeigler [10] eingeführt und ist eine ereignisorientierte und systemtheoretisch-basierte Modellierungs- und Simulationsmethodik. DEVS basiert auf modular hierarchischen Modellspezifikationen und zugehörigen Simulatoralgorithmen. Der DEVS-Formalismus geht von einer eindeutigen Trennung zwischen Modellspezifikation und Modellbearbeitung aus. Bei der Modellspezifikation wird zwischen zwei DEVS-Systemtypen unterschieden. Das dynamische Verhalten wird mit atomaren (atomic) DEVS-Systemen abgebildet. Daneben gibt es gekoppelte (coupled) DEVS-Systeme, welche eine Komposition aus atomic beziehungsweise coupled DEVS-Systemen beschreiben. Jedes coupled DEVS kann wiederum Bestandteil eines anderen coupled DEVS sein. Beide Systemtypen, atomic und coupled DEVS, definieren kompatible Ein- und Ausgänge. Die Spezifikation der Systemdynamik in atomic DEVS erfolgt mit einer Menge festgelegter Funktionen, ähnlich einem endlichen Zustandsautomaten.

Der *Parallel DEVS* (PDEVS) Formalismus ist eine Weiterentwicklung des ursprünglich von Zeigler eingeführten *classic DEVS*-Formalismus. Er definiert neue Mechanismen zur Verarbeitung zeitgleicher Ereignisse und behebt damit eine Schwachstelle von *classic DEVS*. DEVS definiert zwei Arten von Ereignissen, externe und interne Ereignisse, die grundsätzlich unabhängig voneinander auftreten können. Der Zeitpunkt eines jeden internen Ereignisses und auch der Zeitpunkt für das Senden eines Ausgangsereignisses wird von jedem atomic DEVS selbst bestimmt. Entsprechend den Kopplungsbeziehungen, wird ein Ausgangsereignis zu einem Eingangsereignis einer anderen Komponente, welche dieses externe Ereignis verarbeiten muss. Folglich kann dies bei einem atomic DEVS zum zeitgleichen Auftreten von internen und externen Ereignissen führen. Der PDEVS-Ansatz löst diesen Konflikt intern im atomic PDEVS. Hierfür wird die Modellbeschreibung um eine neue Dynamikfunktion erweitert, welche beim parallelen Auftreten von externen und internen Ereignissen ausgeführt wird.

Der Konflikt wird somit ausschließlich auf atomic PDEVS-Ebene gelöst.

Ein atomic PDEVS ist formal definiert als:

$$PDEV S = (X, Y, S, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta) \quad (1)$$

X ist die Menge der Eingangs-, S ist die Menge der Zustands- und Y die Menge der Ausgangswerte. δ_{int} ist die interne und δ_{ext} die externe Zustandsüberföhrungsfunktion. δ_{con} ist die confluent Funktion welche beim zeitgleichen Auftreten von internen und externen Ereignissen ausgeführt wird. λ ist die Ausgabefunktion und ta ist die Zeitfortschrittsfunktion zur Einplanung des nächsten internen Ereignisses.

Ein coupled PDEVS oder auch DEVS-Network (DEVN) ist formal definiert als:

$$PDEV N = (X, Y, D, \{M_d\}, \{I_d\}, \{Z_{i,d}\}) \quad (2)$$

X, Y sind analog PDEVS definiert. D ist die Indexmenge der Subkomponenten. $\{M_d\}$ ist die Menge der atomaren oder gekoppelten Subsysteme mit $d \in D$. I_d ist die Indexmenge (Namen) der auf $d \in D$ einwirkenden Systeme, mit $I_d \subseteq D \cup \{N\}$, wobei $\{N\}$ für den Namen des coupled PDEVS steht. $Z_{i,d}$ ist die Menge der i -zu- d -Output-Beziehungen (Kopplungsbeziehungen) mit $i \in I_d$.

Der PDEVS-RCP-Formalismus ist ebenfalls eine Erweiterung des *Classic DEVS*-Formalismus und wird in [9] detailliert beschrieben. PDEVS-RCP basiert auf PDEVS und erweitert diese um die Möglichkeit der Interaktion mit einer Umgebung unter Echtzeitanforderungen. Unter der Umgebung werden externe Softwarekomponenten oder Hardware verstanden. Da PDEVS-RCP unmittelbar auf PDEVS aufbaut, können PDEVS-RCP-Modelle direkt mit einer PDEVS-Simulationsumgebung ausgeführt werden, wobei diese aufgrund der eingeföhrten Erweiterungen auch als Echtzeitumgebung genutzt werden kann. Dadurch ist es möglich, PDEVS-RCP-Modelle schrittweise von der Entwurfsphase bis zum operativen Steuerungsbetrieb zu erweitern und mit einer einzigen Simulationsumgebung auszuführen. Diese Eigenschaft ist eine notwendige Voraussetzung zur softwaretechnischen Umsetzung des SBC-Frameworks. Aufgrund der Nichterfüllung dieser Eigenschaft ist die Real-Time-DEVS-Erweiterung nach [1] nicht zur Umsetzung des SBC-Frameworks geeignet. Die Namensgebung PDEVS-RCP weist auf die enge Verwandtschaft zu PDEVS hin

und auf die durchgängige Nutzbarkeit der Modelle gemäß dem Vorgehensmodell des Rapid Control Prototypings (RCP) nach [1].

Ein PDEVS-RCP ist formal definiert als:

$$PDEVS - RCP = (X, Y, S, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta, A) \quad (3)$$

Die Definitionen von $Y, S, \delta_{int}, \delta_{ext}, \delta_{con}, ta$ sind analog PDEVS. Die neu eingeführte Menge A spezifiziert die Interaktionen mit der Umgebung. Für jede Interaktion ist ein zulässiges Zeitintervall $[t_{min}, t_{max}]$ anzugeben, welches die Echtzeitanforderungen festlegt. Aufgrund des Ausführungsintervalls wird eine Interaktion als Aktivität $a \in A$ bezeichnet. Die Menge der Eingangsergebnisse setzt sich aus den Mengen X_{model} und X_{clock} zusammen. Dabei umfasst X_{model} die Eingangsergebnisse gewöhnlicher PDEVS-Modelle sowie von der Umgebung. Die Eingangs Menge X_{clock} wird von einer Echtzeituhr generiert. Die Ausgabefunktion λ definiert analog zu PDEVS die Berechnung von Ausgangsergebnissen. Diese können an Modellkomponenten oder die Umgebung versendet werden. Weiterhin erfolgt durch die Ausgabefunktion λ die Bindung der aktuellen Aktivität $a \in A$ an den aktuellen Zustand $s \in S$. Analog dazu wird auch das zulässige Zeitintervall der aktuellen Aktivität a im aktuellen Zustand s abgebildet. Das dynamische Verhalten eines atomic PDEVS-RCP-Modells zeigt Abbildung 4. Aus Übersichtsgründen ist die Behandlung zeitgleicher Ereignisse mit der δ_{con} Funktion nicht dargestellt. Wie in [9] dargestellt, legt diese in der Regel fest, ob für den aktuellen Zustandsübergang zuerst δ_{int} oder δ_{ext} auszuführen ist.

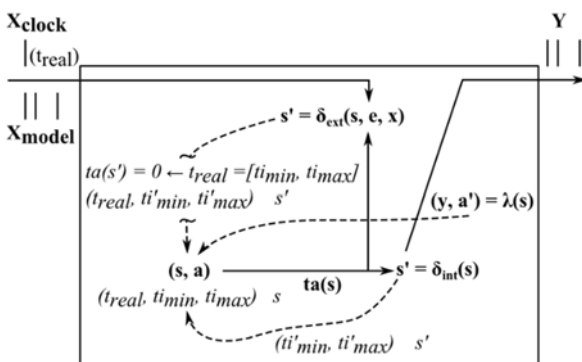


Abbildung 4: Dynamisches Verhalten eines PDEVS-RCP nach [9].

Das grundlegende dynamische Verhalten eines atomic PDEVS-RCP ist identisch mit dem von PDEVS. Unterschiede ergeben sich nur in der konkreten Modellspezifikation. Wie bereits erwähnt, ordnet ein atomic PDEVS-RCP jedem inneren Zustand $s \in S$ explizit eine

Aktivität $a \in A$ mit einem zulässigen Ausführungsintervall zu. Gemäß Abbildung 4 muss die als externes Ereignis ("clock", v) $\in X_{clock}$ empfangene Realzeit durch δ_{ext} als Zustand gespeichert werden. Auf Basis der so gespeicherten Realzeit kann das zulässige Zeitintervall der aktuellen Aktivität $a \in A$ in Realzeit umgerechnet werden, welches ebenfalls im aktuellen Zustand $s \in S$ abgespeichert wird. Weiterhin zeigt Abbildung 4, dass eine Aktivität $a \in A$ durch die Ausgabefunktion λ beendet wird und diese eine neue Aktivität $a' \in A$ startet. Das zugehörige Zeitintervall wird in Echtzeitwerten durch δ_{int} definiert. Alle externen Ereignisse $x \in X$ werden wie bei PDEVS durch die externe Zustandsüberföhrungsfunktion $\delta_{ext}(s, e, x)$ ausgewertet. Diese bestimmt auf Grundlage des aktuellen Zustands s , der verstrichenen Zeit e seit dem letzten Ereignis und den aktuellen externen Ereignissen x den nachfolgenden Zustand s' . Darüber hinaus wird innerhalb der externen Zustandsüberföhrungsfunktion geprüft, ob die aktuellen Ereignisse innerhalb des Zeitintervalls $[t_{min}, t_{max}]$ eintreten. Dabei werden nach [9] zwei Fälle unterschieden.

1. Falls externe Ereignisse innerhalb des Zeitintervalls eintreten, muss der Folgezustand s' zum virtuellen Zeitfortschritt $ta(s)=0$ führen und somit unmittelbar ein internes Ereignis auslösen.
2. Falls externe Ereignisse außerhalb des Zeitintervalls eintreten, werden der Folgezustand s' berechnet und somit auch die Zeitwerte $[t'_{min}, t'_{max}]$ aktualisiert. Anschließend wird der virtuelle Zeitfortschritt $ta(s)$ neu berechnet.

4 MATLAB/DEVS Modellbibliothek

Aufbauend auf den zuvor angeführten Grundlagen, wird in diesem Kapitel eine generische DEVS-Modellbibliothek entwickelt. Zur Implementierung der Komponenten wird die MATLAB/DEVS-Toolbox [2] der Forschungsgruppe CEA verwendet. Bevor auf die Komponenten der Modellbibliothek im Einzelnen eingegangen wird, werden kurz die allgemeinen Anforderungen an den Komponentenentwurf und wesentliche Aspekte zur geometrischen Kopplung dargestellt.

4.1 Anforderungen und Strukturierung

Die Anforderungen ergeben sich aus der Verwendung des SBC-Frameworks, durch DEVS und durch Echtzeitbedingungen. Weiterhin sollen die Komponenten der Modellbibliothek modular und wiederverwendbar sein. Darüber hinaus sollen die Komponenten fehler-tolerant umgesetzt werden. Dafür ist die Kenntnis der Echtzeit zwingend erforderlich. Um diesen Aspekt zu

berücksichtigen, wird die allgemeine SBC-basierte Modellstruktur um eine Komponente RTC (Real Time Clock) ergänzt. Die erweiterte Struktur ist in Abbildung 5 dargestellt. Die RTC-Komponente modelliert eine Echtzeituhr, die über einen Ausgangsport periodisch die Echtzeit an das Prozess- und Interface-Modell als Ereignisse sendet.

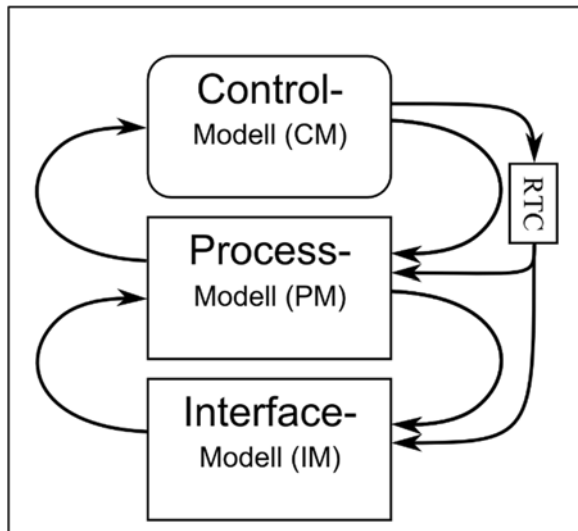


Abbildung 5: SBC-Modell einer Robotersteuerung

4.2 Basismethoden zur geometrischen Kopplung

Zur Umsetzung der geometrischen Kopplung werden unterschiedliche Basismethoden (Hilfsfunktionen) benötigt. Die hier vorgestellte Realisierung der geometrischen Kopplung basiert auf der Koordinatentransformation mit Frames. Ein Frame ist eine spezielle 4x4 Matrix, welche sich aus einer 3x3 Rotationsmatrix und einem Ortsvektor zusammensetzt. Die Rotationsmatrix beschreibt die Verdrehung zweier Koordinatensysteme zueinander. Der Ortsvektor gibt die Position des Koordinatenursprungs des einen Koordinatensystems im anderen Koordinatensystem an.

Für das Aufstellen einer kinematischen Kette zur Koordinatentransformation werden drei verschiedene Frames benötigt. Ein Frame ergibt sich aus der Relation der Basiskoordinatensysteme der geometrisch zu koppeln den Roboter und kann durch einen Einlernprozess gewonnen werden. Ein weiteres Frame kann manuell vorgegeben werden und beschreibt die Relation der TCP's zwischen zwei geometrisch zu koppeln den Robotern. Weiterhin folgt ein Frame aus der aktuell anzufahrenden Roboterkoordinate. Hierbei muss allerdings unterschieden werden, ob die Koordinate eine

relative oder absolute Bewegung des Roboters beschreibt. Absolut bedeutet in diesen Zusammenhang, dass durch die Koordinate die neue Position des TCP im Basiskoordinatensystem des Roboters beschrieben ist. Bei einer relativen Koordinate ergibt sich die neue Position des TCP aus der aktuellen Position des TCP. Die neue Position wird hierbei im Koordinatensystem des TCP beschrieben.

Beide Varianten müssen im Algorithmus der Koordinatentransformation separat betrachtet werden und führen zu unterschiedlichen kinematischen Ketten.

Aus dem Zielframe werden anschliessend die Eulerwinkel, nach der für den Roboter gültigen Konvention, und der Ortsvektor zur Beschreibung der neuen errechneten Koordinate für den Slave-Roboter genutzt. Zur einfachen Umsetzung der sich wiederholenden Berechnungen wurden entsprechend verallgemeinerte MATLAB-Funktionen implementiert, welche durch die Komponenten der Modellbibliothek genutzt werden. Eine detaillierte Beschreibung der Basismethoden kann [3] entnommen werden.

4.3 Komponenten der Modellbibliothek

Dieser Abschnitt beschreibt die Entwicklung der modularen Komponenten für eine Modellbibliothek. Die Modellbibliothek soll eine Grundlage für die Entwicklung von Robotersteuerungen darstellen. Besonders wichtig ist hierbei die Entwicklung der atomic PDEVs für einen Roboter, für ein Interface und die Implementierung einer Echtzeituhr (RTC), da diese Komponenten in jeder geometrisch gekoppelten Robotersteuerung Anwendung finden.

Wie in Abbildung 6 dargestellt, kann ein Roboter dabei als eine Blackbox mit Eingangs-, Ausgangsport und interner Dynamik verstanden werden. Des Weiteren verfügt ein Roboter über Zustände $s \in S$ und Parameter. Die Parameter stellen in diesem Zusammenhang zum Beispiel Roboterkoordinaten, Korrekturwerte oder Informationen zur Position anderer Roboter dar.

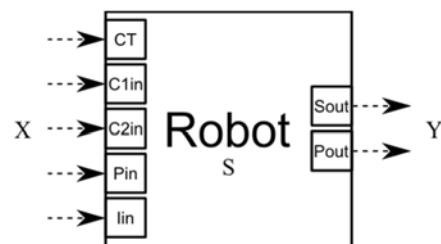


Abbildung 6: Atomic PDEVs-RCP-Roboter

In DEVs kommunizieren Komponenten über Ereignisse miteinander. Ereignisse besitzen einen Ereigniszeitpunkt und einen Wert. Der Roboter verfügt über keine eigene Steuerungslogik, sondern erhält Steuerungseignisse vom Steuerungsmodell. In der Abbildung 6 sind die entsprechenden Ports mit C1in und C2in gekennzeichnet. Über den Port C1in (Control in) sendet die Steuerung dem Roboter die aktuelle Aufgabe (Task). Zum gleichen Zeitpunkt wird von der Steuerung über den Port C2in ein Ereignis gesendet, dass die im Robotermodell integrierte geometrische Kopplung aktiviert bzw. deaktiviert. Basierend auf diesen zwei parallelen Ereignissen bestimmt der Roboter aus seinen Parametern und dem Zustand die Roboterkoordinaten und sendet diese als Ereignis über den Ausgangsport Pout an ein nachgeschaltetes Interface weiter. Für den Fall, dass eine geometrische Kopplung von der Steuerung gefordert wird, muss der Roboter basierend auf den Parametern bestimmen, ob er sich als Slave oder Master verhalten soll. Als Master verfügt der Roboter über Roboterkoordinaten für die von der Steuerung befohlene Task. Sind keine Koordinaten vorhanden, so muss der Roboter aus seinen Parametern die benötigten Kopplungsbeziehungen und die Position seines Masters bestimmen. Der Roboter verhält sich dann als Slave und führt eine Koordinatentransformation auf Basis der Master-Roboterkoordinaten durch. Hierfür werden die Roboterkoordinaten des Masters über den Port Pin (Points in) erfasst. Sendet der Roboter Koordinaten zu seinem korrespondierenden Interface, so muss dieses ihm antworten. Hierfür verfügt das Roboter-PDEVs-RCP über einen weiteren Port Iin (Interface in). Aufgrund der Anforderung, dass ein fehlertolerantes System realisiert werden soll, verfügt der Roboter über einen weiteren Eingangsport CT. Über diesen Port wird die Echtzeit (current Time) empfangen. Der Roboter speichert den Zeitpunkt, an dem ein Befehl von der Steuerung übermittelt wird. Dadurch kann geprüft werden, ob das Interface innerhalb eines vorgegebenen Zeitintervalls antwortet. Ist dies nicht gegeben, so wird ein Fehler erkannt. Der Roboter muss außerdem mit der übergeordneten Steuerung kommunizieren können. Dafür existiert ein weiterer Ausgangsport Sout (Status out). Über diesen Port erhält die Steuerung aktuelle Zustandswerte des Roboters. Die konkreten Algorithmen der Roboterkomponente sind in PDEVs-Notation und als MATLAB/DEVs-Implementierung in [3] zu finden.

Ein Interface ist eine PDEVs-RCP-Komponente im Interface-Modell des SBC-Frameworks. Die Interface-

Komponente bildet die Schnittstelle zu einer realen oder virtuellen Prozesskomponente. Ein Roboter des Process-Modells ist immer mit einem Interface gekoppelt, welches mit einem realen oder virtuellen Roboter kommuniziert. Damit stellt das Interface das Bindeglied zwischen der DEVs-Modellbeschreibung eines Roboters im Process-Modell und einem virtuellen oder realen Roboter dar.

Das mit PDEVs-RCP zu modellierende Interface kann wiederum als eine Blackbox mit Eingangs- und Ausgangsporten verstanden werden und ist in Abbildung 7 schematisch dargestellt. Dem Interface werden vom zugehörigen Roboter des Process-Modells Roboterkoordinaten gesendet. Diese werden ausgeführt, indem sie an den virtuellen oder realen Roboter übermittelt werden. Ob die Aufgabe durch den realen/virtuellen Roboter ausgeführt worden ist, wird durch regelmäßiges Polling getestet. Hierfür verfügt das PDEVs-RCP des Interfaces über einen weiteren Eingang, über den die Echtzeit empfangen wird (Port CT). Das Interface verfügt über Grenzwerte, die bestimmen, wann frühestens oder spätestens mit einer Antwort des realen oder virtuellen Roboters zu rechnen ist. Wird dieses Intervall nicht eingehalten, tritt ein Fehler auf. Antwortet ein Roboter im vorgegebenen Zeitintervall (Normalfall), wird über den Port Iout (Interface out) des Interfaces ein Ereignis gesendet, welches dem PDEVs-Roboter Modell bestätigt, dass sein Bewegungskommando ausgeführt wurde.



Abbildung 7: Atomic PDEVs Interface

Die Real-Time-Clock (RTC) ist ebenfalls ein atomic PDEVs-Modell. Die RTC sendet periodisch über den Port CT die Echtzeit. Die RTC ist immer aktiv ($\sigma=0$ - vgl. DEVs Abschnitt 3.2) und wird nur im Falle eines Fehlers von der Steuerung beendet.



Abbildung 8: Atomic PDEVs RTC

Zum Erstellen einer Robotersteuerungsapplikation werden weitere problemspezifische Modellkomponenten auf der Process-Modell-Ebene und ein konkretes Control-Modell benötigt.

5 Anwendungsbeispiel

Nachfolgend wird ein Anwendungsbeispiel zur Verifikation der Modellbibliothek vorgestellt. Das Anwendungsbeispiel benutzt für die konkrete Prozessanbindung die MatlabKK-Robotic [6] und MatlabKK-Robotic-Visualization [6] Toolboxes.

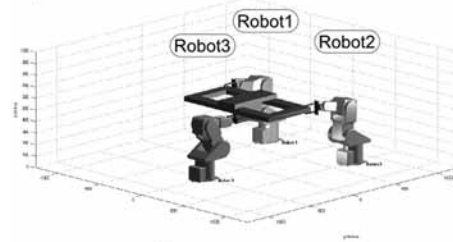
Abbildung 9 zeigt den grundlegenden Aufbau der Applikation sowie vier unterschiedliche Prozessphasen in Form visualisierter Prozesskomponenten. Die Applikation besteht aus drei Robotern, einer Tischplatte und einem Gleitstein. Roboter 3 soll den Gleitstein bewegen. Die beiden Roboter 1 und 2 sollen gemeinsam die Tischplatte bewegen. Es soll nur die Bahn des Gleitsteines beschrieben werden, die einer gestauchten Parabel folgen soll. Hierbei wird die Tischplatte von den Robotern 1 und 2 so nachgeführt, dass diese ständig Kontakt zum Gleitstein hat. Dabei soll lediglich der Roboter 3, welcher den Gleitstein bewegt, „angelernt“ werden. Der vordere Roboter 3 ist hierbei der Master für die beiden hinteren Roboter (Slaves). Die Slaves folgen den Bewegungen des Masters mittels geometrischer Kopplung.

Die Ausgangssituation ist in Teilbild (a) abgebildet. Teilbild (b) zeigt die Phase job1, wobei die Tischplatte und der Gleitstein zuerst nach rechts bewegt und gleichzeitig geneigt werden. Am höchsten Punkt bewegen sich die Roboter zurück zur Mittellage, um beide Objekte in die entgegengesetzte Richtung zu neigen. In der Phase job2 (Teilbild (c)) bewegen sich die beiden hinteren Roboter unabhängig vom Master Roboter 3. Dieser verharrt in seiner Position. Einer der beiden hinteren Slave-Roboter übernimmt dabei die Führung und wird für diese Phase zum temporären Master. Der andere Slave folgt diesem mittels geometrischer Kopplung. Wieder in der Nulllage angekommen (Teilbild (a)), erfolgt der Beginn der Phase job3 (Teilbild (d)). Hierbei bewegt sich nur der vordere Roboter und die beiden hinteren Roboter verharran in ihrer Position.

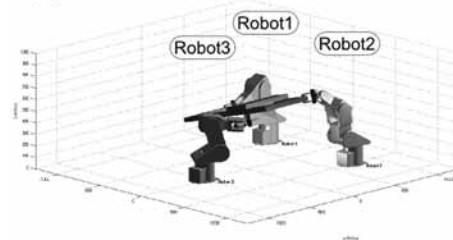
Abbildung 10 zeigt die für die Applikation folgende DEVS-Modellstruktur gemäß dem SBC-Framework. Der Informationsaustausch zwischen den DEVS-Komponenten findet über Ereignisse statt. In der Abbildung ist dies durch gestrichelte Linien zwischen den einzelnen Komponenten dargestellt. Roboter 3 agiert als Master für den Roboter 1 und Roboter 1 als Master für den Roboter 2. Die Master-Slave Beziehung der Roboter folgt gemäß Unterabschnitt 4.3 aus den Ro-

bot.Pout zu Robot.Pin Kopplungen, über die anzufahrende Roboterkoordinaten vorgegeben werden. Der Master bestimmt seine Koordinaten auf Basis der von der Steuerung vorgegebenen Aufgabe (Port C1in).

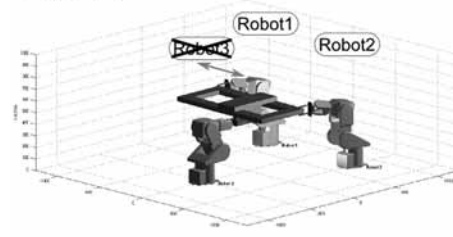
a) initial / finish



b) job1



c) job2



d) job3

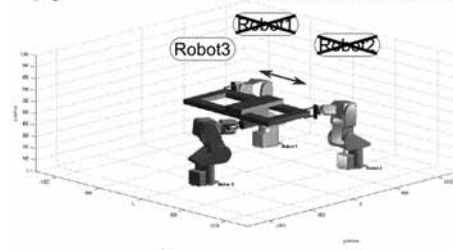


Abbildung 9: Prozessphasen der Applikation

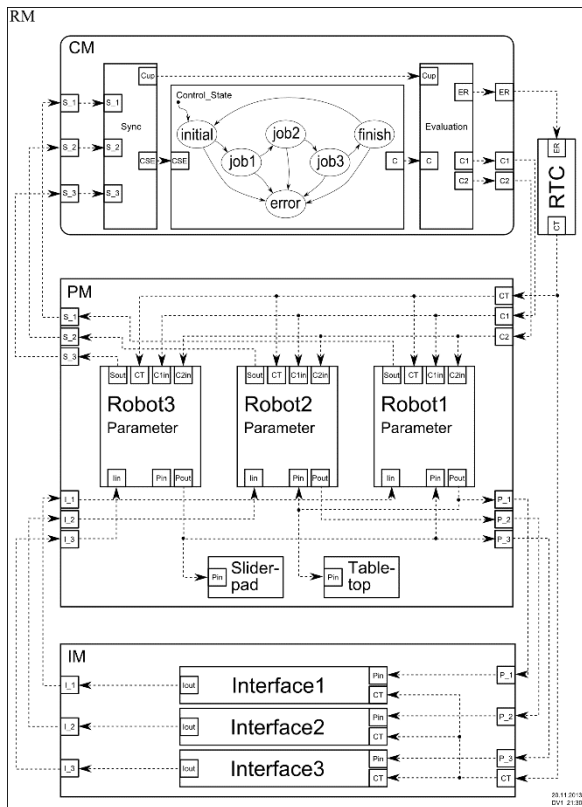


Abbildung 10: Modellstruktur des Anwendungsbeispiels

Die grundlegende Modellabarbeitung, d.h. die Simulation beziehungsweise die operative Steuerungsausführung ist in Abbildung 11 schematisch auf Basis der coupled DEVS Systeme dargestellt. Zum Simulationsstartzeitpunkt $t=0$ sind nur das Control-Modell (CM) und die Echtzeituhr aktiv. Das Control-Modell sendet über die Kopplungsbeziehungen ein aus Aufgabe und Kopplungsbefehl bestehendes Ereignis an das Process-Modell (PM) (1). Aufgrund des Eingangsereignisses, wird das Process-Modell aktiv. Es verarbeitet den Ereigniswert und berechnet Roboterkoordinaten, welche an das Interface-Modell (IM) durch ein Ereignis übertragen werden (2). Das Interface-Modell empfängt das Ereignis und sendet die Roboterkoordinaten an seinen realen oder virtuellen Roboter. Das Interface-Modell deaktiviert sich. Zu diesem Simulationszeitpunkt ist nur noch die Echtzeituhr aktiv. Die Echtzeituhr ist in der Abbildung durch einen Blitz schematisch dargestellt. Die Echtzeituhr versendet die Echtzeit als Ereigniswert. Das Interface-Modell erhält die Echtzeit als Eingangsereignis und überprüft, ob die Roboter den zuvor übertragenen Bewegungsbefehl vollständig abgearbeitet haben. Ist dies der Fall, so wird ein finish-Ereignis an das Process-Modell gesendet (3). Dieses wertet die Information aus und generiert ein Statusereignis, welches an das Control-Modell übertragen wird

(4). Daraufhin wird ein neuer Befehl von der Steuerung an das Process-Modell gesendet und die Prozedur wiederholt sich.

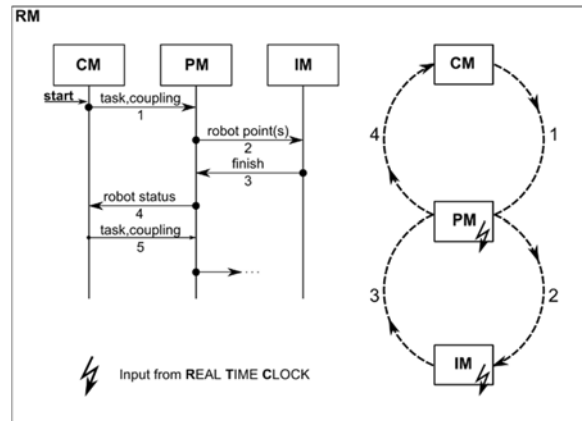


Abbildung 11: Prinzipielle Abarbeitungsreihenfolge beim Anwendungsbeispiel

6 Zusammenfassung

Roboter sind flexible Maschinen und in der Produktion sowie anderen Anwendungen weit verbreitet. Auf dem Markt sind verschiedenste Robotertypen und -hersteller vertreten. Dementsprechend groß ist auch die Anzahl der verfügbaren Softwarelösungen. Viele Anwendungen erfordern einen Einsatz von kooperierenden Robotern. Diese Arbeit analysiert die unterschiedlichen elementaren Interaktionsprinzipien von Knickarmrobotern. Als ein wesentliches Interaktionsprinzip wurde die geometrische Kopplung identifiziert. Dazu wurden die theoretischen und mathematischen Grundlagen erläutert und diese robotertypenabhängig softwaretechnisch in einer Modellbibliothek umgesetzt.

Bei der softwaretechnischen Umsetzung wurde auf dem Simulation Based Control (SBC) Framework aufgebaut, welches auf den Grundideen des Rapid Control Prototypings (RCP) der Regelungs- und Steuerungstechnik basiert. Die Implementierung der Modellbibliothek erfolgte in der MATLAB/DEVS-Umgebung auf Basis des PDEVS-RCP-Formalismus, einer Weiterentwicklung des Discrete Event Structure (DEVS) Formalismus von Zeigler. Die Umsetzung des SBC-Frameworks in einer PDEVS basierten Simulationsumgebung wurde ausführlich erläutert. Anschließend wurde die Benutzung der Modellbibliothek an einem ausgewählten Anwendungsbeispiel aufgezeigt, welche die einzelnen Schritte der Steuerungsentwicklung detailliert beschreibt.

Das Anwendungsbeispiel zeigt deutlich die noch bestehenden Grenzen und Probleme hinsichtlich der Interaktion von Robotern auf. Die Roboteraktionen werden über die Steuerung synchronisiert. Wenn die Steuerung ein Kommando an die Roboter sendet, müssen Roboterkoordinaten neu berechnet und diese über eine Schnittstelle (Interface) an die realen oder virtuellen Roboter gesendet werden. Die Übertragung erfolgt dabei sequenziell und nicht parallel an die Roboter. Dadurch kann nicht gewährleistet werden, dass die unterschiedlichen Roboter zum selben Zeitpunkt mit ihrer Bewegung beginnen. Um dieses Problem zu minimieren, müssen Roboterbewegungen diskretisiert werden. Bei einer gekoppelten Bewegung mehrerer Roboter sind nur kleine Teilbewegungen möglich. Die zeitliche Synchronisation stellt somit ein Problem dar.

Ein weiteres Problem stellen die noch festen Koppelungsbeziehungen innerhalb der übergeordneten Robotersteuerung dar. Diese bestimmen, welcher Roboter als Master für einen anderen Roboter agiert.

Einen Ansatz zur Lösung dieses Problems könnte der in [9] dargestellte reaktive Steuerungsansatz bieten. Dieser basiert auf einer deklarativen Beschreibung von einer Menge unterschiedlicher Steuerungsstrategien mit sukzessiver Generierung temporärer Steuerungen. Diese Art der automatischen Anpassung während des operativen Betriebs soll fortführend für die Interaktion von Robotern untersucht werden.

7 References

- [1] Abel, D.; Bollig, A.: Rapid Control Prototyping, Methoden und Anwendungen, Springer, 2006 ISBN 978-3-540-29524-2
- [2] Deatcu, C.; Schwatinski, T.; Pawletta, T.: DEVS Toolbox for Matlab®, Website 2013, http://www.mb.hs-wismar.de/cea/DEVS_Tbx/MatlabDEVS_Tbx.html, Vers.1.3, Hochschule Wismar, Stand November 2013
- [3] Freymann B.: Entwicklung einer Modellbibliothek für die Interaktion von Robotern in der MATLAB/DEVS Umgebung. (Entwurf zur Master-Thesis, Dezember 2013)
- [4] Lüth, T.: Technische Multi-Agenten-Systeme: verteilte autonome Roboter- und Fertigungssysteme. München, Wien; Hanser, 1998 ISBN 3-446-19468-1
- [5] Maletzki, G.: Rapid Control Prototyping komplexer und flexibler Robotersteuerungen auf Basis des SBC-Ansatzes. HS-Wismar, FG-CEA Dissertation (eingereicht, Univ Rostock, Juni 2013)
- [6] Otto, J.; Christern, M.; Schmidt, A.; Schwatinski, T.; Pawletta T.: KUKA-KAWASAKI-Robotic Toolbox for Matlab® & KUKA-KAWASAKI-Visualization Toolbox for Matlab®, Websites, http://www.mb.hs-wismar.de/cea/KK_Robotic_Tbx/KK_Robotic_Tbx.html http://www.mb.hs-wismar.de/cea/MatlabKK_Robotic-and-Visualization_Tbx/MatlabKK_Robotic_Visualization_Tbx.html, Hochschule Wismar, Stand November 2013
- [7] Pawletta, T.; Pawletta, S.; Maletzki, G.: Integrated Modeling, Simulation and Operation of High Flexible Discrete Event Controls. In: Proc. of Mathematical Modelling MATHMOD 2009, Argesim Report No. 35, Ed. I. Troch & F. Breitenecker, Vienna, Austria, Feb. 11-13, 2009, 13 pages, ISBN 978-3-901608-35-3
- [8] Schwatinski, T., Pawletta T.: An Advanced Simulation Approach for Parallel DEVS with Ports. In: Proc. of Spring Simulation Multi-conference 2010, Book 4 - Symposium on Theory of Modeling & Simulation - DEVS, Orlando/Florida, USA, April 11-15, 2010, 132-139, ISBN 1-56555-342-X
- [9] Schwatinski, T.: Reaktive und aufgabenorientierte Robotersteuerungen mit dem SES/MB-Framework und dem SBC-Vorgehensmodell. HS-Wismar, FG-CEA (Entwurf zur Dissertation, Oktober 2013)
- [10] Schwatinski, T.; Pawletta, T.; Pawletta, S.; Kaiser, C.: Simulation-based development and operation of controls on the basis of the DEVS formalism. In: Proc. of The 7th EUROSIM 2010 Congress, Vol.2: Full Papers, Prag, Czech Republic, 2010, 8 pages, ISBN 978-80-01-04589-3
- [11] Zeigler, B.P.; Praehofer, H.; Kim, T.G.: Theory of Modeling and Simulation. Second Edition, Elsevier Academic Press, 2000 ISBN 0-12-778455-1

2Simulate: A Distributed Real-Time Simulation Framework

Jürgen Gotschlich, Torsten Gerlach, Umut Durak

German Aerospace Center (DLR)

Institute of Flight Systems

{ juergen.gotschlich, torsten.gerlach, umut.durak }@dlr.de

Simulating large scale complex real-time systems requires enabling infrastructure for real-time co-simulation of various subsystems with complex behaviors and interfaces. AVES (Air Vehicle Simulator) is a reconfigurable flight simulator of the Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR) for research into rotorcraft and fixed-wing aircraft behavior. Complex models of aircraft subsystems in AVES required a distributed real-time simulation framework. This paper presents 2Simulate, the enabling simulation infrastructure of the AVES facility that facilitates integrating a wide range of models and simulation hardware and software components. 2 Simulate is a unique simulation infrastructure being domain independent and methodology neutral. Its three components, 2SimCC, 2SimRT and 2SimMC, provide various capabilities including simulation control, task scheduling, model integration and hardware/software interfacing.

1 Introduction

Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR) Institute of Flight Systems has a reconfigurable flight simulator for research into rotorcraft and fixed-wing aircraft behavior. It is called Air Vehicle Simulator (AVES). AVES features a common motion platform and interchangeable roll-on/roll-off (RoRo) cockpits, enabling rapid turnaround of research activities [1].



Figure 1 DLR AVES

While developing system simulations, composing various models of subsystems and integrating them with the diverse tools and components that are required for the operation of simulation has always been a major technical challenge. The complex nature of the modelled subsystems and evolving requirements of the user community made this challenge heavier for flight simulators.

AVES has been developed based upon the idea to utilize reusable, flexible, standardized and properly validated software modules. It has a distributed architecture that enables each module to run either on a single computer connected via Ethernet or run together on the same hardware as distinct processes. Critical processes, e.g. the flight loop, are run in hard real-time conditions on real-time operating systems.

In this paper, 2Simulate, the enabling simulation infrastructure of the AVES facility is presented. 2Simulate is an overall simulation framework to facilitate integrating a wide range of models and simulation components like data recorders or image generators. The next section will provide a background about simulation frameworks. 2Simulate will then be introduced with a quadrotor simulation example that demonstrates its capabilities.

2 Simulation Frameworks

Simulating large-scale complex real-time systems requires specific attention on the infrastructure that enables the real-time co-simulation of various subsystems with complex behaviors and interfaces. The simulation community has long been working on tools and infrastructures that make reliable, maintainable and extensible complex systems simulations possible.

Huang and Sarjoughian [2] state that a separate effort to develop a methodology for simulation of complex real-time systems is required. They advocate utilizing the system-theoretic modelling approach Discrete Event System Specification (DEVS) [3] for simula-

tion modelling just as the Unified Modeling Language (UML) is used for software design. With RTDEVS/CORBA, Cho et al. introduced a real-time distributed simulation infrastructure [4]. This infrastructure provides services for time synchronization, message delivery, interfacing external systems and implementing real-time computations.

About ten years before the RTDEVS/CORBA effort, Lee and his colleagues had already proposed one of the first simulation frameworks, Ptolemy, for simulating heterogeneous systems [5]. Ptolemy aimed at making use of object-oriented software technology to model subsystems. It was a framework which provides a set of object-oriented class definitions with standard interfaces. Thus, with generic objects more specialized interoperable domain-specific objects can be implemented.

In 1998, NASA published a domain-specific framework for simulation of aircraft [6]. They introduced LaSRS++, an object-oriented framework for developing real-time flight simulators. In this framework, a set of abstract base classes is provided to interface the modelled aircraft with the framework services. These base classes include e.g. *FlightSim* that defines the initialization and execution of the vehicle model, *World* that provides a world to fly around, *HardwareControl* that abstracts the hardware used in the simulation and *Supervisor* that cares about the real-time clock. Via LaSRS++ framework services one can achieve real-time framing, simulation models management like trim, hold, reset, and interfacing with the I/O hardware.

These three important approaches (DEVS, Ptolemy, LaSRS++) each provide an infrastructure for simulation of complex real-time systems. While the first one utilizes a domain-independent simulation formalization approach and expects its user to develop DEVS models, the second one, Ptolemy, provides an object-oriented approach for systems modelling. The third one, LaSRS++, provides a domain-specific solution. However, there are two important issues about these approaches. First, simulation developers require flexible frameworks so that they can utilize various methodologies for systems modeling. There is no single methodology that satisfies all user requirements for modeling large and complex systems. While power system modelers find bond graphs more useful, flight systems modelers may like state flow diagrams better. Second, Simulation developers require frameworks to be flexible also in creating spe-

cific architectures for their particular problems. Domain-specific frameworks always possess the developer's abstraction of the domain which may not fit all of their users' needs.

The simulation framework that is presented in this paper, 2Simulate, neither enforces a modelling methodology nor enforces a domain architecture. It provides various real-time simulation services via Application Programming Interfaces (APIs), which do not depend on the domain architecture or the modelling methodology.

3 2Simulate

2Simulate is a C++ real-time distributed simulation framework. It is composed of three components, namely 2Simulate Real-Time Framework (2SimRT), 2Simulate Control Center (2SimCC) and 2Simulate Model Control (2SimMC). Figure 2 presents a simple UML Component Diagram of 2Simulate.



Figure 2 Components of 2Simulate

2SimRT is the core simulation framework of 2Simulate that provides deterministic scheduling and controlling of real-time tasks. It comes as Windows or QNX images (Libraries) and API header files. Any simulation application that is based on 2SimRT is called a Target. Each Target runs various real-time tasks that are implemented utilizing the 2SimRT API. These real-time tasks include model control tasks as well as a wide range of data connections to external devices or components. 2SimRT also provides a Common Database to manage the data that flow through the internal and external interfaces.

2SimMC is the component that abstracts model interfaces for 2SimRT. It works with MATLAB/Simulink [7], Advantage Framework [8] or native C++ models. Targets may have more than one model that they co-simulate over 2SimMC. It supports the real-time operating system QNX and Windows. For native C++ model development, users can employ 2SimMC via developing their models using its API. For MATLAB/Simulink and Advantage Framework, 2SimMC is integrated automatically into the models during the code generation process.

2SimCC is the component to configure the Control Center for specific needs. It is a Windows executable

which can be customized via configuration files called 2SimCC project files. Control Center can run, pause or stop various Targets. Besides, it accesses the Target Data Dictionaries which can be defined as the data access mechanisms and enables presenting or editing Target data at runtime. It can also enable user management to define and enforce user access rights.



Figure 3 2Simulate Simulation Architecture

As presented in Figure 3, the simulation architecture of 2Simulate has a Control Center that can control a number of Targets which may co-simulate various Models and interacts with various external systems.

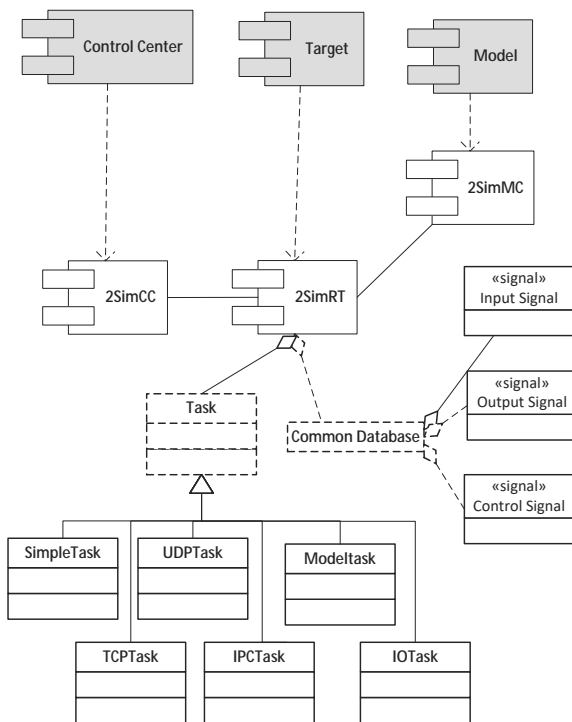


Figure 4 2Simulate Component Architecture

The component architecture of 2Simulate is presented in Figure 4. 2SimRT provides a number of schedulable task templates and a common database. Some major tasks are depicted in the figure. They can be programmed using their pre- and post-initialization and pre- and post-process callbacks with extra functionality depending on their types. SimpleTask is the simplest task type which has no extra functionality. The user can modify it for his/her needs. With a UDPTask, one can schedule a UDP communication and

with TCPTask, a TCP communication. IPCTask is used for inter-process communication with other applications on the same machine. With IOTask, a 2SimRT user can connect to I/O interfaces like switches or onboard computers and lastly the ModelTask enables to run the models that are built to be integrated into 2Simulate. There are more task types whose properties and functions are mostly inherited from these major tasks (see Fig. 5). As an example, 2SimRT has an ARINCTask and a CANTask derived from IOTask for these widely used communication protocols. As another example ConTask, that is used to connect the developed 2SimRT application to 2SimCC, is derived from TCPTask. One can integrate Simulink models and C++ models using SimulinkTask and CppModelTask that are derived from ModelTask. Last to mention, users can also extend these tasks to create their own special tasks. A nice example for that is WclsTask that is derived from UDPTask to enable communication with the control loading systems from Wittenstein GmbH. It implements a particular protocol for this subsystem in AVES.

There are also simulation utilities like adding displays or command line monitors and data injectors in 2SimRT.

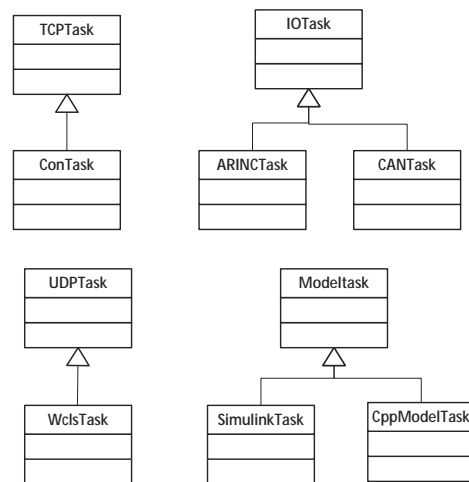


Figure 5 Examples of Task Hierarchies

Common Database is a 2SimRT add-on. It allows its users to define interfaces of the Target by Input and Output Signal specifications. And it provides an internal data interface to Control Signals. These signals are defined in text files, which are then used for automatic code generation that produces a Common Database source code with an API to access and mod-

ify these data items. Users can also design and develop their indigenous data management routines.

4 Simulating a Quadrotor Using 2Simulate

Further details of 2Simulate will be presented by an example implementation that simulates a quadrotor. This Quadrotor Simulator consists of an Operator Node that has a joystick for getting operator inputs, a virtual instrument panel that provides the user with a primary flight display, a Visualisation Node that has an out-of-the window image generator for simulating the camera on the quadrotor. It has a Simulation Node that runs the flight dynamics and control model of the quadrotor and an Instructor Node that controls and monitors the simulator execution. The architecture of this Quadrotor Simulator is depicted below.

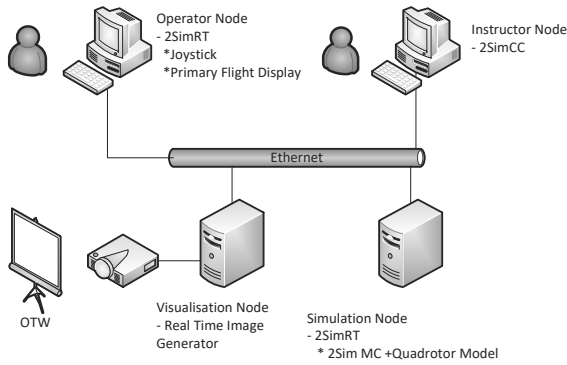


Figure 6 Quadrotor Simulator Architecture

The Simulation Node uses an open source Simulink implementation of a quadrotor model [9] from the Mathworks File Exchange site. The Simulink model implements flight dynamics and control algorithms from Bouabdallah's work [10].

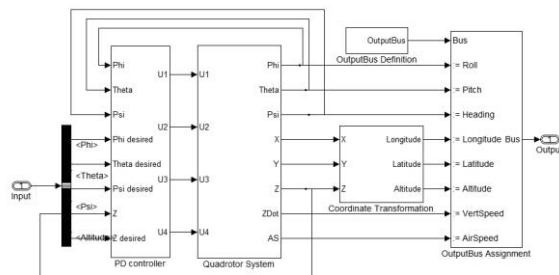


Figure 7 Simulink Model of Quadrotor Flight Dynamics and Control

The Simulink model can be used with 2Simulate after it has been converted into C++ code using Mathworks Simulink Coder [11]. A part of the Simulink Coder is the Target Language Compiler. It is a specification of the code generation [12] utilizing so called system target files, which can be customized for specific needs. 2Simulate has such a set of customized system target files. They embed 2SimMC into the model code during the code generation. Thus, an auto-generated model is readily available for SimulinkTask.

While developing the Simulation Node, a Simulink-ModelTask is generated from ModelTask to interface the quadrotor model with 2Simulate. Below is a code extract that adds the SimulinkModelTask to the Simulation Node. The scheduling schema of the task is specified as Round Robin (*TASK_SCHED_RR*), the task priority is set to 30 (0 is the highest and 50 is the lowest) and the frame time is set to 10 milliseconds.

```
quadSimTask *pQuadST = new quadSimTask ( pTSim, "QUAD",
    TASK_SCHED_RR, 30, 10*IMSECTONSEC);
quadSimTask ->setDesc( "Quadrotor Simulink Task" );
quadSimTask ->setPreProcCB ((void (*)(TSim *, TSimRtTask*))
    &pQuadrotorTSimSimulinkTask_CB );
```

Code 1. SimulinkModelTask

Using a UDPTask Simulation Node gets the user inputs from Operator Node and sets them in the Common Database. As presented in the third line of Code 1, there is a pre-process callback function pointer specified for the SimulinkModelTask. In this callback function the inputs of the Simulink model are set using the values in the Common Database.

Here is a code extract from Simulation Node, the *com* is an instance of Common Database.

```
void pQuadSimTask_CB( TSim *pAppl, TSimRtTask *pRtTask ) {
    com->o.r.quad.Input.Phi = com->i.r.acctrl.ksiCmd;
    com->o.r.quad.Input.Theta = com->i.r.acctrl.etaCmd;
    com->o.r.quad.Input.Psi = com->i.r.acctrl.zetaCmd;
    com->o.r.quad.Input.Altitude = com->i.r.acctrl.plaLCmd;
}
```

Code 2. SimulinkModelTaskCallback Function

It has input (i) and output (o) signals. As an example, *Phi* is sent to the quadrotor model as an input and it comes from the aircraft command *acctrl ksiCmd*.

The Common Database code is auto-generated using the signal specifications. Signal specifications are well formed text files wherein the user defines the identifier, the type and the length of signals. An extract from the signal specifications of Simulation Node is given below.

```

#####
#//
#// aircraft control from joystick
#//
#//
#####
q
21000 00000 acctrl1 simNodeCmd 1 D D D RAW [-] simNodeCmd
21001 00000 acctrl1 etaCmd 1 F F F RAW [-] etaCmd
21002 00000 acctrl1 ksiCmd 1 F F F RAW [-] ksiCmd
21003 00000 acctrl1 zetaCmd 1 F F F RAW [-] zetaCmd
21004 00000 acctrl1 plaCmd 1 F F F RAW [-] plaCmd

```

Code 3. Simulation Node Signal Specification

2Simulate provides a Display utility to add a 2Indicate [13] display to its 2SimRT framework presenting the signals that are specified in Common Database. VisualisationTask is a kind of UDPTask to picture the state of the simulated entity in a virtual environment. It requires the spatial state of the entity, i.e. latitude, longitude and altitude and sends this state to an OpenSceneGraph-based RealTimeImageGenerator [14] over UDP.

In Operator Node SimpleTask is used to collect joystick inputs from the user and Display is used to present a Primary Flight Display. Simulation Node employs a VisualizationTask to drive the Visualization Node.

The last component to mention is the Instructor Node which utilizes 2SimCC to control the whole simulation (Fig. 8).

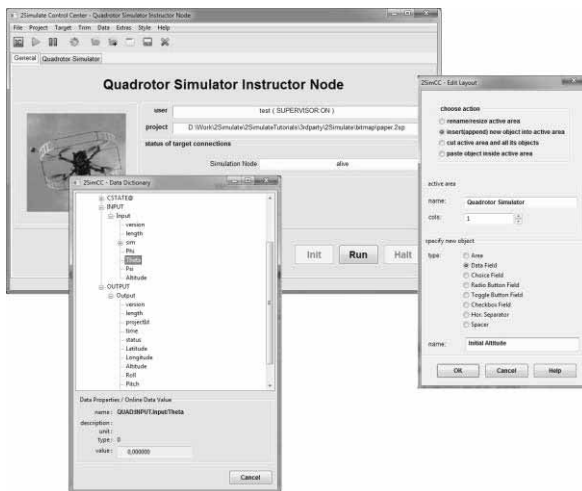


Figure 8 Instructor Node with 2SimCC

It can be presented as a reconfigurable front end for 2SimRT. In its General tab, one can track the status of 2SimRT Targets and Init, Run or Halt the simulations of these targets. It also provides utilities to add new tabs to visualize and edit values that are already defined in the signal specifications of the connected targets. It is also possible to access the values in the Common Database of the connected target and modi-

fy it during runtime via the Target Data Dictionary functionality.

5 Conclusion

This paper presents 2Simulate, a distributed real-time simulation framework. With its components 2SimCC, 2SimRT and 2SimMC it furnishes its users with tools and services to simulate complex real-time systems in a distributed fashion. As we identified that the basic pitfall of various other simulation frameworks has been their dependencies either on a modeling methodology or a domain architecture, our objective while developing 2Simulate has been to create a simulation framework that is independent of the domain architecture and the modeling methodology.

2Simulate is being employed as the underlying simulation framework of AVES rotorcraft and fixed-wing simulators with great success. The authors plan to extend the services and facilities of this infrastructure by supporting the commonly used distributed simulation standard IEEE 1516 High Level Architecture [15,16,17] and the emerging independent model interfacing standard Functional Mockup Interface [18].

6 References

- [1] H. Duda, T. Gerlach, S. Advani and M. Potter, *Design of the DLR AVES Research Flight Simulator*, in AIAA Modeling and Simulation Technologies (MST) Conference, Boston, MA, 2013.
- [2] D. Huang and H. Sarjoughian, *Software and Simulation Modeling for Real-Time Software-Intensive Systems*, in Eighth IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT'04), 2004.
- [3] B. Zeigler, T. Kim and H. Praehofer, *Theory of Modeling and Simulation*, New York: Academic Press, 2000.
- [4] Y. K. Cho, X. Hu and B. Zeigler, *The RTDEVS/CORBA Environment for Simulation-Based Design of Distributed Real-Time Systems*, Simulation: Transactions of The Society for Modeling and Simulation, Vol. 79, Nr. 4, pp. 197-210, 2003.
- [5] J. Buck, S. Ha, E. Lee and D. Messerschmitt, *Ptolemy: A Framework for Simulating and Prototyping Heterogeneous Systems*, Int. Journal of Computer Simulation, Vol. 4, pp. 155-182, April 1994.

- [6] R. Leslie, D. Geyer, K. Cunningham, P. Glaab, P. Kenney and M. Madden, *LaSRS++ An Object-Oriented Framework for Real-Time Simulation of Aircraft*, in AIAA Modeling and Simulation Technologies Conference, 1998.
- [7] Mathworks, *Simulink: Simulation and Model-Based Design*, 05 Dec 2013. [Online]. Available: <http://www.mathworks.com/products/simulink>
- [8] Applied Dynamics International, *ADvantage Framework*, 05 Dec 2013. [Online]. Available: <http://www.adl.com/products/advantage/>.
- [9] Mathworks File Exchange, *PD Control Quadrotor - Simulink*, 05 Dec 2013. [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/41149-pd-control-quadrotor-simulink>.
- [10] A. Samir, *Design and Control of Quadrotors with Application to Autonomous Flying*, Ph.D. Thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2007.
- [11] Mathworks, *Simulink® Coder™ Target Language Compiler*, 05 Dec 2013. [Online]. Available: http://www.mathworks.com/help/pdf_doc/rtw/rtw_tlc.pdf.
- [12] Mathworks, *Simulink Coder: Generate C and C++ code from Simulink and Stateflow Models*, 5 Dec 2013. [Online]. Available: <http://www.mathworks.com/products/datasheets/pdf/simulink-coder.pdf>.
- [13] DLR, *2Indicate - Flexible Visualisierung technischer Prozesse*, 05 Dec 2013. [Online]. Available: http://www.dlr.de/tm/desktopdefault.aspx/tabid-3015/7941_read-6822.
- [14] V. Kuehne and P. Nartz, *OpenSceneGraph Reference Manual v2.2*, Ann Arbor, MI: Blue Newt Software, 2007.
- [15] IEEE, *IEEE Standard for Modeling and Simulation High Level Architecture (HLA)– Object Model Template (OMT) Specification*, 2010.
- [16] IEEE, *IEEE Standard for Modeling and Simulation High Level Architecture (HLA)– Framework and Rules*, 2010.
- [17] IEEE, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)– Federate Interface Specification*, 2010.
- [18] MODELISAR Consortium, *Functional Mock-up Interface for Co-Simulation Version 1.0*, 2010, [Online] Available <https://fmi-standard.org/>.

Henon - eine alternative numerische Methode zur Ereignislokalisierung

Franz Preyser¹, Bernhard Heinzl², Felix Breiteneker¹

¹Institut für Analysis und Scientific Computing, TU Wien

²Institut für Rechnergestützte Automation, TU Wien

Franz.Preyser@tuwien.ac.at

In den folgenden Zeilen wird das Funktionsprinzip einer nicht-iterativen Methode zur Lokalisierung von Ereignissen, während des numerischen Lösens eines ODE-Systems, beschrieben. Weiters werden Ergebnisse eines Vergleichs, hinsichtlich Laufzeit, einer Matlab-Implementierung mit dem sonst üblichen Regula-Falsi-Verfahren präsentiert.

1 Einleitung

Beim numerischen Lösen von gewöhnlichen Differentialgleichungssystemen liefert das numerische Lösungsverfahren (ODE-Solver) Lösungswerte zu konkreten Werten der unabhängigen Variablen(meist die Zeit). Diese Werte haben entweder festen Abstand zueinander, oder die Abstände sind variabel(fix-step-/variable-step-solver). In manchen Fällen ist es jedoch notwendig, Lösungswerte bei ganz bestimmten Systemzuständen zu berechnen, um z.B. anschließend Parameterwerte oder Gleichungen zu ändern.

Die übliche Herangehensweise ist, eine sogenannte Ereignisfunktion $S: \Omega \rightarrow \mathbb{R}$ zu definieren. Diese Funktion bildet den Zustandsraum Ω stetig in die reellen Zahlen ab und ihre Nulldurchgänge kennzeichnen die Ereigniszeitpunkte. Das numerische Lösungsverfahren überprüft nun nach jeder Lösungswertberechnung, ob sich das Vorzeichen der Funktion S geändert hat. Wird ein Vorzeichenwechsel detektiert, so wird ein Algorithmus zur Nullstellenlokalisierung angeworfen. Iterative Verfahren sind das Newton-, das Sekanten-, das Bisektions- oder das Regula-Falsi-Verfahren. M. Henon stellt in [1] ein alternatives Verfahren vor, das über einen Wechsel der Integrationsvariablen versucht ohne Iterationen auszukommen.

2 Henon-Verfahren

Das Henon-Verfahren kann als eine Erweiterung des Newton-Verfahrens gesehen werden und stellt deshalb auch dieselben Ansprüche an die Funktion,

deren Nullstelle lokalisiert werden soll - hier die Ereignisfunktion. Vernachlässigt man den numerischen Fehler und geht also davon aus, dass die berechneten Lösungswerte x_n gleich den Werten der analytischen Lösung $x(t_n)$ sind, so kann die Ereignisfunktion auch als Funktion der Zeit gesehen werden: $\tau(t) = S(x(t))$. Ist nun τ in dem Intervall $(t_n, t_e + (t_e - t_n))$ stetig und monoton, wobei $\text{sign}(\tau(t_n)) \neq \text{sign}(\tau(t_{n+1}))$ und $\tau(t_e) = 0$ gelte und ist zusätzlich die Ableitung $d\tau/dt$ in dem Intervall bekannt, so darf Newton angewandt werden:

$$t_T = t_n - \left(\frac{d\tau}{dt}(t_n) \right)^{-1} \cdot \tau(t_n) \quad (1)$$

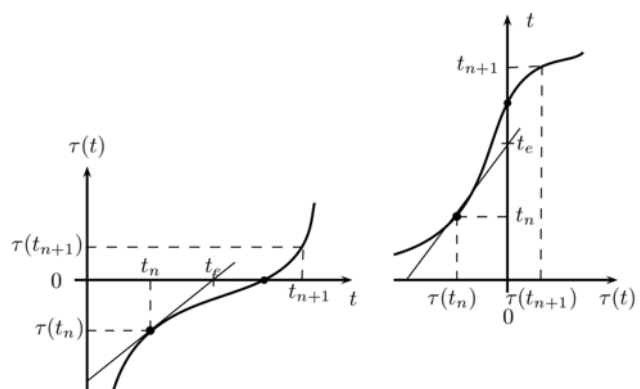


Abbildung 1. Eine Iteration im Newton-Verfahren bzw. im Henon-Verfahren unter Verwendung des expliziten Euler. li.: t als Integrationsvariable; re.: τ als Integrationsvariable

Gleichung (1) kann jedoch auch anders interpretiert werden. Da mit den obigen Voraussetzungen $\tau(t)$ im Intervall $(t_n, t_e + (t_e - t_n))$ invertierbar ist und die Inverse

im Punkt $\tau(t_n)$ die Steigung $(d\tau/dt(t_n))^{-1}$ hat, entspricht (1) genau einem Schritt des expliziten Eulerverfahrens mit Schrittweite $-\tau(t_n)$, angewandt auf das AWP:

$$\frac{dt}{d\tau}(\tau) = \left(\frac{d\tau}{dt}(t) \right)^{-1}, \quad t(\tau_n) = t_n \quad (2)$$

Es liegt daher nahe, anstelle der iterativen Anwendung des Newton-/Expliziten Euler-Verfahrens, einmalig ein Verfahren höherer Ordnung anzuwenden. Beim Henon-Verfahren passiert genau das. Für die Dauer der Ereignislokalisierung nimmt τ den Platz der unabhängigen Variable ein, während die Zeit t zu einer Funktion von τ wird. Anschaulich spiegelt man um die 1. Mediane und integriert dann im Ereigniswertebereich von $\tau=\tau(n)$ bis $\tau=0$ (siehe Abb. 1). Eine wesentliche Voraussetzung für Newton wie auch für Henon ist also, im Unterschied zu allen anderen aufgezählten Methoden, dass die Ableitung der Ereignisfunktion analytisch bekannt und die Ereignisfunktion in einem kleinen Bereich um die zu lokalisierende Nullstelle invertierbar sein muss.

3 Implementierung in Matlab

Um die Brauchbarkeit der Henon-Methode zu überprüfen, haben wir sie in Matlab implementiert. Konkret wurde eine Wrapper-Funktion für Matlab ODE-Solver geschrieben. Diese kann somit völlig universell auf jedes Beispiel angewandt werden, wo üblicherweise ein Matlab-Solver zum Einsatz kommt und die Voraussetzungen für das Newton-Verfahren erfüllt sind. Also AWP's der Bauart:

$$\frac{dx}{dt} = f(t, x), \quad x(t_0) = x_0 \quad (3)$$

Es ist möglich, AWP (2) und (3) so zusammenzufassen, dass lediglich der Wert einer Variablen K entscheidet, ob beim Lösen des resultierenden AWP's (4), gerade das eigentliche Problem (3) integriert, oder eine Nullstelle der Ereignisfunktion berechnet wird.

$$\begin{aligned} \frac{dy}{d\tau} &= f(t(\tau), y) \cdot K(\tau, y) \\ y(\tau_0) &= y_0 \\ \frac{dt}{d\tau}(\tau) &= K(\tau, y), \quad t(\tau_0) = t_0 \end{aligned} \quad (4)$$

Ist die Funktion K die konstante 1-Funktion, so gilt $\tau=t$ und somit auch $y(\tau)=x(t)$. Setzt man K gleich $(d\tau/dt)^{-1}$ und die Anfangswerte entsprechend, so

wechselt die Rolle der unabhängigen Variablen τ von Zeit auf Ereignisfunktion. Man integriert (4) dann bis $\tau=0$ und erhält somit den Systemzustand $x(t_e)=y(0)$ zum Ereigniszeitpunkt t_e , sowie t_e selbst.

4 Vergleiche

Verglichen wurde die Henon-Implementierung mit dem Regula-Falsi-Verfahren. Um den Vergleich fair zu gestalten, wurde nicht der ausgereifte Matlab-Lokalisierungsmechanismus verwendet, sondern eine eigene Implementierung analog zur Henon-Implementierung verwendet. Am Beispiel des Bouncing Ball mit unterschiedlichen Ereignisdefinitionen, wurde die Performance hinsichtlich gemachtem Fehler und der Laufzeit verglichen. Lokalisiert wurden die Nullstellen der Ball-Flugbahn sowie deren Extremstellen.

Beispiel	Solver	t_{RF}/t_H	t_{ML}/t_H
BB	ode23	8.75	0.19
BB M	ode23	0.60	0.11
d BB M	ode23	1.06	0.10
BB	ode45	4.76	0.35
BB M	ode45	4.43	0.35
d BB M	ode45	2.98	0.25

Tabelle 1. Ergebnis Laufzeitvergleich

t_H ... Laufzeit der Henon-Implementierung
 t_{RF} ... Laufzeit der Regula-Falsi-Implementierung
 t_{ML} ... Laufzeit des Matlab-Mechanismus
 BB ... Bouncing Ball
 BB M ... Bouncing Ball mit Maximumlokalisierung
 d BB ... gedämpfter Bouncing Ball

5 Fazit

Henon hat sich im Vergleich zu Regula Falsi als mindestens ebenbürtig erwiesen. Vor allem bei verhältnismäßig großer Schrittweite der zugrundeliegenden Integrationsmethode lassen die Ergebnisse auf Laufzeitvorteile hoffen. Es bedarf allerdings sicherlich noch diverser Optimierungen um mit ausgereiften Algorithmen mithalten zu können.

6 Referenzen

1. M. Hénon: *On the numerical Computation of Poincaré Maps*, Physica D: Nonlinear Phenomena, 5(2), S. 412–414, 1982.

Control Strategies for Energy-Optimized Operation of Assembly Systems

Robin Diekmann, Joscha Heinze, Ilja Alkov, Dirk Weidemann

Institute of System Dynamics and Mechatronics,

University of Applied Sciences Bielefeld

{robin.diekmann, joscha.heinze, ilja.alkov, dirk.weidemann}@fh-bielefeld.de

Controller modifications offer possibilities to increase the energy efficiency of industrial plants. These can easily be applied to plants under operation in contrast to constructional adaptations, which are often economically risky after startup. To obtain such modifications, an appropriate model-based approach is used. First the technically relevant units are modeled and the model is parameterized as well as verified using measurement data collected at the real machine. Subsequently, a sensitivity analysis is conducted via simulation to reveal the most relevant controller parameters concerning the energy consumption. In the next step, the energy consumption can be optimized using an algorithmic approach varying these controller parameters. Finally, measurements are performed for validation. This model-based approach is applied to the demonstrator application of an assembly system. By integrating the identified modifications, the energy consumption is lowered by about 40 % and as a further effect the machine cycle time is shortened by up to 8 %.

1 Introduction

Due to the climatic change and declining fossil energy sources, energy efficiency gets a more and more important topic. In industry, an improvement of the energy efficiency is one reaction to deal with the unknown development of the energy costs and supply in the future [1]. Especially for the energy intensive field of production, multiple innovations are currently under development to lower the energy consumption of plants. Many of these approaches, like redesigning of system components or optimization of procedural definitions and the organizational structure, lead to changes of the hardware [2]. In case of existing plants, these required hardware modifications can be, however, very complex and economically risky.

Another way to increase the energy efficiency is to improve the software, i.e., control algorithms of machines. The work presented within this paper follows this approach: adaptations of the process control are conducted to lower the energy consumption of an industrial plant. Therefore, a structured procedure is applied step by step

to the demonstrator machine of a riveting assembly system. First, a physical model of the most relevant system units in terms of energy consumption, especially the hydraulic circuit, is created. Thereby, measurement data is used for the identification of the model parameters utilizing an algorithmic optimization. Furthermore, the part of the PLC code containing the control algorithm concerning the hydraulic valves is modeled. Thus, the system's reaction to changes of the process control can be tested independently of the real plant via simulation, avoiding the risk of damages. With help of a sensitivity analysis, the most influential controller parameters concerning the energy consumption of the process are identified and an optimization procedure can be executed for chosen parameters.

By applying the results to the demonstrator system, it can be shown that the use of the obtained controller parameters leads to a decrease in energy consumption for the real plant, comparable with the results of the simulation. The energy consumption of the plant is lowered by about 40 % by adapting the controller. Furthermore, the cycle time of the riveting process can be shortened by about 8 % using optimized controller parameters.

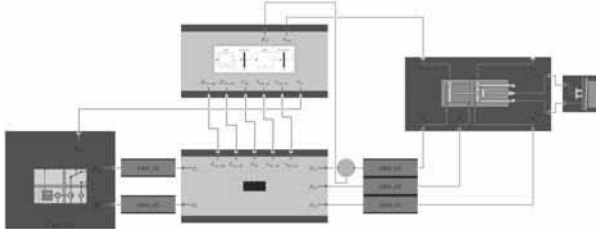


Figure 1: Model of the considered assembly system. A pump (bottom left) supplies hydraulic power to the circuit. The valve block (bottom center) divides the volume flow to drive the actuator (right) depending on the outputs of the sequential controller (top). Hydraulic lines connect the according units.

2 Modeling

The considered industrial plant assembles metal sheets by riveting them, which is a common technique, e.g., in automotive production. The according model is structured in two parts, as shown in Fig. 1: the physical model and the model of the sequential controller. Both are modeled in Modelica language within the software Dymola. The Dymola environment provides graphical and textual modeling and is also used as the simulation tool.

2.1 Modeling of the Sequential Controller

The model of the sequential controller is composed of elements of the Modelica StateGraph package [3], which can be used for the modeling of discrete event systems in Modelica language. It is based on a subset of the JGrafchart language, which in turn contains elements of the Grafset language and the Statechart formalism.

In case of the PLC of the demonstrator system, the code of the sequential controller is written in Structured Text. Hence, this code is translated to the StateGraph language, whereby the graphical representation supports the required analysis process. In addition to this, there are several more arguments that lead to the proceeding to work with a model of the sequential controller instead of running the PLC code on the simulation system. As the identical modeling language is used, the exchange of data between the physical model and the model of the sequential controller does not require the use of interfaces. Furthermore, by working with a model, the parameters of the sequential controller can be easily accessed, which is important for the application of algorithmic methods on the controller, e.g., optimization procedures.

For the testing of the controller model, measurement data collected at the demonstrator machine is used. In order to access the relevant process values, the assembly system is equipped with several sensors. For usage in the simulation, the measured input values of the sequential controller are written to a log file for each considered process cycle. Additionally, this file contains the output values of the sequential controller. By stimulating the controller model with the input values and comparing the simulated output values with the output values of the PLC, the controller model is successfully verified.

2.2 Modeling of Physical Effects

2.2.1 Bond Graph Modeling Formalism

Several different physical domains have to be considered to model the assembly system with reasonable accuracy, especially hydraulics and mechanics. Due to this multi-domain nature of the system, all physical effects are modeled using the Bond Graph (BG) approach. This offers a methodology for the description of complex multi-disciplinary systems closely related to their physical structure, whereby only a few generalized elements need to be used for modeling. By the object-oriented nature, BG possess the according advantages and allows the modular and hierarchical modeling of complex systems.

The BG formalism is based on the idea that the power P is defined for each technically relevant discipline as the product of an effort value e and a flow value f , i.e.,

$$P = e \cdot f. \quad (1)$$

For example, in the field of hydraulics, it is reasonable to assume pressure as the effort variable and the volume flow as the flow variable while in the field of translative mechanics, force is conventionally assumed as the effort variable and velocity as the flow variable. Independent of the discipline, linear standard elements are defined by constitutive equations, e.g., resistive, capacitive, and inductive elements. Each of these can be given a physical interpretation in the corresponding discipline.

Whereas the standard BG elements are used for linear models with constant inputs, a significant extension of the modeling opportunities is given by modulated elements, which are marked by the prefix M (e.g., MSe denotes a modulated effort source). Modulated elements

are generalizations of linear elements, where coefficients are given by an external variable. Considering a combination of linear elements, modulated elements, and sensors observing power variables, it is possible to describe a wide range of nonlinear models by case-related definitions of the external variables.

Nevertheless, the definition of application-specific, and, where appropriate, nonlinear elements may lead to a convenient modeling process and ensures the clarity of the obtained models. As complex hydraulic effects play a significant role in the considered assembly system, hydraulic elements are therefore modeled slightly different to the conventional BG approach. In the standard case, one-port elements are used for resistive, inductive, and capacitive effects. In contrast, hydraulic capacitances, hydraulic inductances, several types of hydraulic resistances, and further elements are modeled as two-port elements. However, these are still compatible in the conventional way with the other BG elements. A continuous description of the BG formalism can be found e.g. in [4].

The graphical system representation given by a BG can be translated automatically to a system of differential-algebraic equations if an appropriate tool is used. Therefore, the library BondGraph was implemented in Modelica to be used for the modeling process. The library is open source and published under the LGPL license at the official Modelica website [5]. A detailed description of the library is given in [6].

2.2.2 Modeling Procedure

At the top level, the model is structured into the main technical units (cf. Fig. 1). Except for the sequential controller, each unit is composed of elements of the BondGraph package in several hierarchical layers. This modular as well as hierarchical approach allows the simple aggregation and – if necessary – refinement of the submodels.

The technical units are aggregated from single effects to complete components. For example, the individual effects of hydraulic resistances, capacitances and inductances are combined by a connection in series to form the model of a hydraulic line as shown in Fig. 2. The joints are considered by *HRT* elements, which represent turbulent resistances, whereas strait pipe friction is considered by *HR* elements. Inductive effects of the working

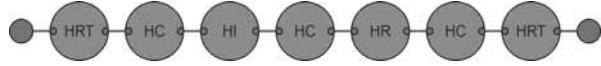


Figure 2: Model of a hydraulic line.

fluid are considered by *HI* elements and volume nodes modeled by *HC* elements are placed between each inductance and resistance. Though the pictured model contains only one *HI* and one *HR* element, it may be refined by adding more elements without considerable additional work. In the model of the assembly system, hydraulic lines with two of these elements each are used. It has turned out that this is of sufficient accuracy at reasonable computational cost. At the top level, relevant model parameters such as the hydraulic diameter or the initial pressures can be given and are propagated to the according submodels.

The remaining technical units are modeled in a similar way, where special attention is paid to computational efficiency and numerical stability. This is especially of importance for switching components, e.g., the valve block. Further explanations concerning the modeling procedure can be found in [6, 7].

3 Optimization

An algorithmic optimization can be executed for both the identification of the model parameters and the determination of optimal parameter values for the sequential controller concerning the energy consumption. The required functions are implemented in Matlab, where also the setting of the optimization parameters and the execution takes place. Therefore, an interface between Matlab and Dymola is installed, which is based on Matlab functions provided by Dymola.

3.1 Model Parameter Identification

The considered machine is equipped with several sensors, thus this data can be used for a model parameter identification. This is realized by stimulating the model with measurement data using modulated source elements. Related to the hydraulic line introduced above, the pressure before and after the line as well as the volume flow rate are measured. As shown in Fig. 3, for the parameter identification, modulated effort sources

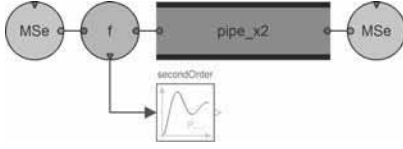


Figure 3: Exemplary model for the parameter identification. The hydraulic line is stimulated by measured pressures using the modulated effort sources. The simulated volume flow is low-pass filtered by the PT2-element to reproduce the dynamic performance of the sensor.

are connected to each end of the line and stimulated in a simulation by the pressure curves that have been detected. This allows the comparison between the flow determined from the simulation run and the measurement. Their deviation is supposed to be minimized in an optimization procedure.

Since the according sensor works on the turbine principle, it reacts rather slowly to changes of the volume flow. For an appropriate comparison, this behavior is transferred to the simulation by using a PT2-element for a low-pass filtering of the simulated volume flow. In addition to the parameters of the considered hydraulic line, the filter parameters are determined via the optimization procedure to reproduce the dynamic performance of the sensor properly.

For the algorithmic optimization, the Matlab functions mentioned above are applied. The objective function δ is defined as

$$\delta = \int (Q_{sim,lp}(t) - Q_{meas}(t))^2 \cdot \gamma(t) dt. \quad (2)$$

By using the squared deviation, both negative and positive deviations are handled the same and small deviations are less factored than great ones. Furthermore, the deviation is weighted with help of the function $\gamma(t)$. In case of the hydraulic line, the temporal progress of the weighting function is defined according to the dynamics of the simulated volume flow and the attributes of the volume flow sensor. Because of its low-pass characteristic, the sensor is not able to make note of quick signal changes. Therefore, intervals in which the simulated volume flow shows high dynamics (before the low-pass filtering) are less weighted than intervals with more static characteristic.

In Fig. 4, the measured volume flow is compared with the simulated, unfiltered volume flow as well as the simulated, filtered volume flow. Before low-pass-filtering, the simulated volume flow shows an offset in time in

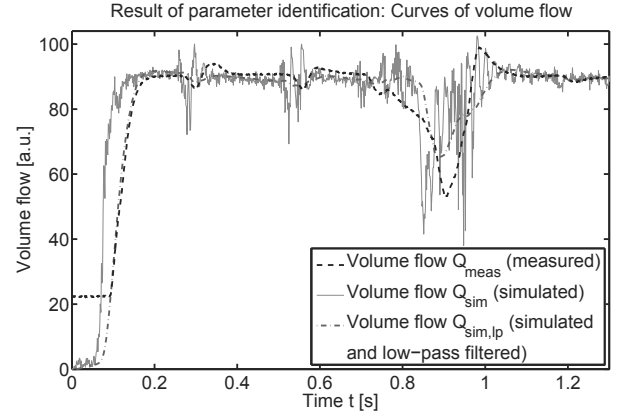


Figure 4: Comparison of measured and simulated curves of volume flow as exemplary results of the parameter identification.

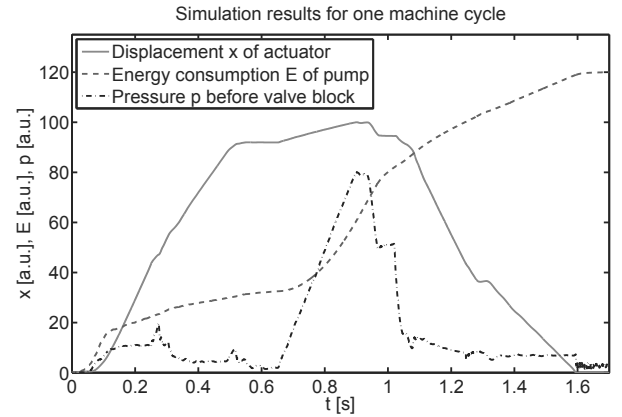


Figure 5: Simulation results for one machine cycle.

comparison to the measured value. After appliance of the low-pass filter, the curves are very similar, especially in intervals with low dynamics. At the beginning of the process, the volume flow is below the minimal output value of the sensor. Thus, the weighting function $\gamma(t)$ equals zero for this interval.

Further submodels are parameterized in a similar way and afterwards combined to obtain the model of the complete system as shown in Fig. 1. Simulation results for one machine cycle are plotted in Fig. 5. From 0 s to 0.5 s, the pump is started and subsequently, the actuator is moved at a relatively low pressure until it reaches the metal sheets and its standstill is detected. Next, a higher pressure is build up to apply the force necessary for pressing the rivet in. In this interval from 0.65 s to 1.0 s, more than 40 % of the complete energy of the machine cycle is consumed. The actuator is moved back to its initial position at a relative low pressure afterwards.

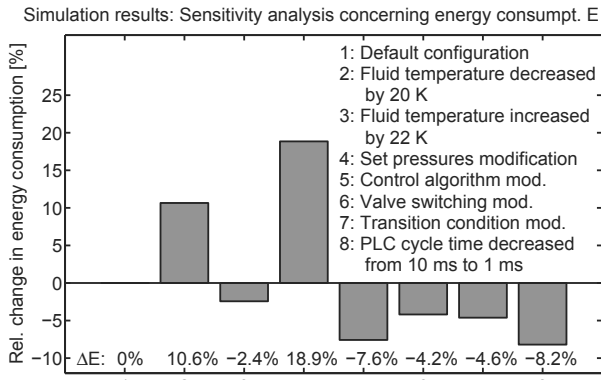


Figure 6: Selected simulation results of the sensitivity analysis concerning the energy consumption.

3.2 Energy Optimization

To ensure the energy-optimized operation of the plant, in principle, it is possible to conduct an optimization procedure as described above with the energy consumption for one machine cycle being the objective for a minimization. In the considered case, the energy consumption is determined via simulation, thus, the gradient is not analytically available and numerical methods have to be used. As multiple parameters of the control algorithm have to be taken into account, the computational effort for an optimization algorithm would be disproportionately high if all parameters that could possibly be modified were used. Therefore, a sensitivity analysis is conducted to reveal parameters with high influence on the energy consumption of the system. It is possible to identify these from the experience gained during the modeling procedure and from the operation of the plant. Some results of the according simulations are shown in Fig. 6.

As can be seen, the fluid temperature significantly influences the energy consumption. Its increase has a positive effect on the energy efficiency which is due to the decrease in viscosity with higher temperatures leading to less hydraulic friction. Thus, this result suggests to set the temperature starting point for active cooling higher than in the default case. While there are also modifications that lead to an increase in energy consumption, many simulations show promising adaptations of the sequential controller.

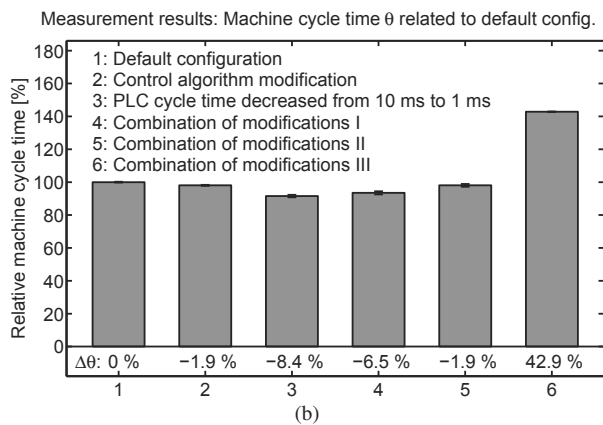
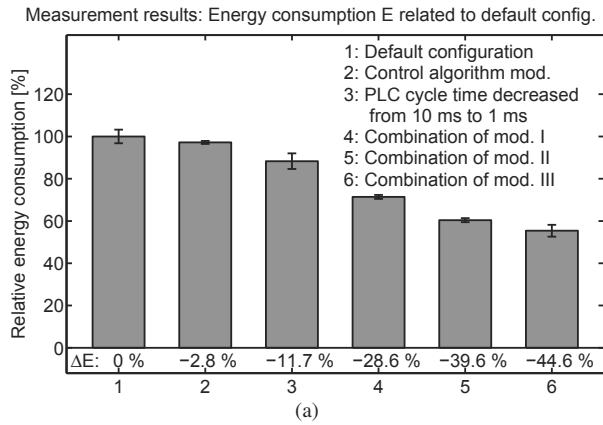


Figure 7: Measurement results for different controller configurations concerning the energy consumption (a) and the machine cycle time (b). Several measurements are performed for each configuration of which the mean values are shown. The error bars show the standard deviations.

4 Results

After an evaluation of the simulation results, reasonable modifications are implemented for the sequential controller of the plant and measurements are conducted. The criterion used to assess the effects of the modifications is the complete consumed electrical energy of the machine for one operation cycle. Selected results are shown in Fig. 7(a).

In case of the control algorithm modification, the simulation shows a decrease by 7.6 % in energy consumption (cf. Fig. 6) though the measurements only demonstrate a decrease by 2.8 %. Still, the energy consumption is lowered by this modification, which concerns an adaptation of the controller code to increase its efficiency, but

does not change the control sequence itself, e.g., transition conditions.

The adaptation of the PLC cycle time from 10 ms to 1 ms is more effective than predicted as it lowers the energy consumption by 11.7 % while the simulation shows a decrease by 8.2 %. By this modification, the control algorithm is executed ten times more often per time interval as in the default case. Thereby, needless energy consumption is avoided. For example, a high pressure is necessary to apply the force for pressing the rivet into the metal sheets and this yields a high power demand. As soon as this subprocess is finished, the according pressure is no more required and may be relieved, resulting in a significantly lower power demand. By the adapted PLC cycle time, the controller can detect more precisely when the subprocess is finished and react accordingly, which shortens the interval where the high power is demanded. In many cases, the soonest possible reaction to changes in the process is desired and therefore, it is reasonable to choose the shortest possible PLC cycle time which does not affect the computational performance of the controller.

Further controller modifications are related to different valve switching conditions, the avoidance of unnecessary idle running of the pump as well as pressure build-up, and additional adaptations.

In the next step, several different controller modifications, that have shown to be effective, are combined. For combinations I and II, the energy consumption is lowered by 28.6 % or even 39.6 %, respectively.

However, energy efficiency is not the only factor of relevance for an operator of the assembly machine. Another crucial point is the machine cycle time as this considerably influences the integration into a production line. As Fig. 7(b) shows, almost all presented modifications lead to a decrease in machine cycle time. Thereby, the reduction of the PLC cycle time shows the maximum decrease. Compared to this, all combined modifications yield a longer cycle time. A significant difference can be found from combination I to combination II, where the decrease in cycle time is 4.6 percentage points less. On the other hand, the decrease in energy consumption is 10.3 percentage points higher in the second case. Hence, it seems reasonable to apply combination II because of the significantly higher energy efficiency.

The evaluated combination of modifications III shows a further decrease in the energy consumption by 5.0 per-

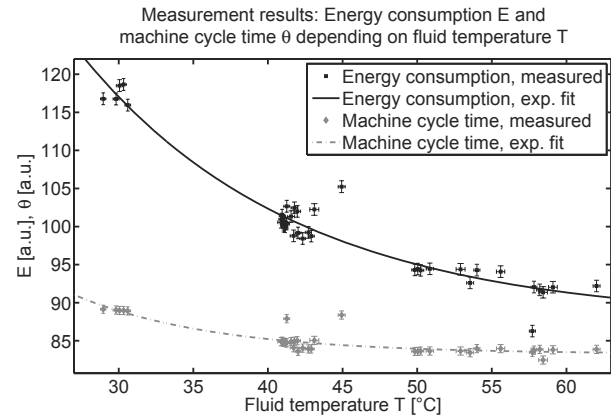


Figure 8: Measurement results for the default sequential controller configuration concerning the energy consumption and the machine cycle time depending on the temperature of the working fluid. The temperature values refer to the mean values of the measured temperatures of the working fluid returning from the machine during the according machine cycles and error bars show the 99 % confidence interval.

centage points, but the cycle time is almost doubled in comparison to the standard configuration. Thus, this modification is mostly not suitable for the standard usage of the assembly system being integrated into a production line. However, it might be used in cases where short cycle times are not of concern, e.g., when parallel working machines take longer to finish their job.

Different temperatures of the working fluid have been evaluated as well and the results are plotted in Fig. 8. Related to the energy consumption, the effect predicted by the model of a decreasing energy consumption with an increasing temperature is validated. An exponential curve fits well to the measurement data which confirms the previously discussed correlation with the viscosity of the fluid. This relation is also integrated into the computation of the fluid properties in the model. Furthermore, the slightly decreasing machine cycle time at an increasing temperature indicates the relationship between fluid temperature and friction, too.

To summarize, the combination of modifications II and a higher temperature of the working fluid than in the standard case have proven to be the most reasonable actions for the evaluated system in terms of energy efficiency and machine performance.

5 Conclusion

Suitable control strategies may significantly influence the energy efficiency of industrial plants. It has emerged that a model-based procedure can lead to controller adaptations which lower the energy consumption by about 40 %. This approach offers the advantage of easily evaluating modifications via simulation and, furthermore, it allows the use of algorithmic methods. Often, multiple physical domains have to be considered to obtain models with sufficient accuracy and thus the Bond Graph formalism is a well suited modeling tool for this task.

The presented work refers to controller modifications of existing plants. This strategy shows promising potential to lower the energy consumption of machines that are already under operation in an easily applicable way.

Acknowledgments

This work was funded by the European Union and the federal state of North Rhine-Westphalia, Germany.

References

- [1] T. Komenda and V. Malisa, “Energy consumption of production lines as a newly established factor of optimisation,” *Proceedings of the 14th International Workshop on Research and Education in Mechatronics*, pp. 212–215, 2013.
- [2] H. Bartusch, A. M. F. Alcalde, M. Fröhling, F. Schultmann, and F. Schwaderer, *Erhöhung der Energie- und Ressourceneffizienz und Reduzierung der Treibhausgasemissionen in der Eisen-, Stahl und Zinkindustrie*. KIT Scientific Publishing, 2013.
- [3] M. Otter, K.-E. Årzén, and I. Dressler, “Stategraph – a modelica library for hierarchical state machines,” *Proceedings of the 4th International Modelica Conference*, pp. 569–578, 2005.
- [4] W. Borutzky, *Bond Graph Methodology*. New York, USA: Springer, 2010.
- [5] I. Alkov and R. Diekmann. (2013) BondGraph library. [Online]. Available: <https://www.modelica.org/libraries>
- [6] I. Alkov, R. Diekmann, and D. Weidemann, “A generalized power-based modelica library with application to an industrial hydraulic plant,” *Proceedings of the 10th Modelica Conference*, 2014, accepted.
- [7] I. Alkov and D. Weidemann, “Nichtlineare Modelle hydraulischer Komponenten zur energieflussbasierten Modellierung,” *Tagungsband ASIM-Treffen STS/GMMS*, pp. 125–133, 2013.

Interdisziplinäre Optimierung der Energieeffizienz in Fertigungsbetrieben mittels Co-Simulation

Bernhard Heinzl^{1,2}, Wolfgang Kastner¹, Matthias Rößler², Friedrich Bleicher³, Fabian Dür³,
Iva Kovacic⁴, Ines Leobner⁵, Niki Popper²

¹Institut für Rechnergestützte Automation, Technische Universität Wien

²dwh GmbH Simulation Services, Wien

³Institut für Fertigungstechnik und Hochleistungslasertechnik, Technische Universität Wien

⁴Institut für Interdisziplinäres Bauprozessmanagement, Technische Universität Wien

⁵Institut für Energietechnik und Thermodynamik, Technische Universität Wien

bernhard.heinzl@tuwien.ac.at

Zunehmend steigende Anforderungen sowohl an die wirtschaftliche Wettbewerbsfähigkeit als auch ökologische Nachhaltigkeit erhöhen die Notwendigkeit für Maßnahmen zur Steigerung der Energieeffizienz in der Produktionsindustrie. Einzelne Bereiche innerhalb von Produktionsbetrieben (Produktionssystem, Energiesystem, Gebäudehülle, etc.) lassen sich mithilfe von simulationsbasierten Methoden individuell analysieren. Um allerdings zusätzliches Optimierungspotential zu erschließen, ist es notwendig die Grenzen von derartigen Simulationen zu erweitern und auch dynamische Wechselwirkungen zwischen einzelnen Optimierungsfeldern zu berücksichtigen. Die vorliegende Arbeit stellt einen Ansatz für eine interdisziplinäre Co-Simulation vor, bei dem für eine integrierte Gesamtsimulation mehrere Simulationsumgebungen gekoppelt werden und quasi-parallel zur Laufzeit periodisch Daten austauschen. Damit können nicht nur verschiedene Modellbeschreibungen, sondern auch speziell angepasste Berechnungsalgorithmen kombiniert werden. Die technische Implementierung anhand einer Fallstudie eines metallverarbeitenden Fertigungsbetriebes zeigt ein Anwendungsbeispiel. Ziel ist es ein Werkzeug zur strategischen Entscheidungsunterstützung in den frühen Planungsphasen von Produktionsbetrieben zur Verfügung zu stellen, mit dem sich qualifizierte Vorhersagen über den Einfluss und finanzielle Auswirkungen von Energiesparmaßnahmen treffen lassen.

1 Einleitung

Bei der Planung von Produktionsbetrieben ergeben sich immer komplexere Herausforderungen durch die stetig steigenden Anforderungen an Energie- und Ressourceneffizienz bei gleichzeitiger Reduktion von Zeit und Kosten zur Sicherung der Wettbewerbsfähigkeit. Die Entscheidungen, die in den frühen Planungsphasen getroffen werden, sind von zentraler Wichtigkeit für die zukünftige Performance einer Industrieanlage, denn in dieser Phase besteht das größte Potential für energetische Optimierung. Je weiter fortgeschritten der Planungsprozess ist, desto weniger Freiheitsgrade und Möglichkeiten für Änderungen bestehen und desto mehr Kapital muss dafür investiert werden. Gleichzeitig sind allerdings die frühen Stadien der Planung in der Regel charakterisiert durch mangelnde Informationen und divergierende Interessen einzelner Stakeholder. Neue simulationsbasierte Werkzeuge können hier zusätzliche

Informationen liefern und damit Entscheidungsprozesse unterstützen.

Für eine umfassende Verbesserung der Energieeffizienz von Produktionsbetrieben müssen einerseits der Produktionsprozess selbst sowie Maschinen und Anlagen, andererseits auch die energetische Infrastruktur (Heizung, Kühlung, Energieversorgung) analysiert werden. In einer mehrstufigen Herangehensweise (siehe Abbildung 1.) wurden zuerst die individuellen Optimierungsfelder Prozess, Maschine, Produktionssystem, Energiesystem und Gebäude in bestehenden Industrieanlagen gesondert auf ihre energetischen Einsparungspotentiale hin analysiert und verschiedene Möglichkeiten und Szenarien mittels Simulationsmodellen quantitativ bewertet. Eine integrierte Gesamtsimulation dieser Teilsysteme erlaubt es schließlich auch Abhängigkeiten und energetische Wechselwirkungen untereinander abzubilden und damit weiteres Optimierungspotential zu erschließen. Dies konnte über eine Kopplung der Teilmodelle in

einem Co-Simulations-Framework realisiert werden. Die großen Unterschiede in der zeitlichen Dynamik der einzelnen Teilsysteme (Zeitkonstanten im Bereich von ms bei Prozess- und Maschinenmodellen gegenüber Stunden bei der Gebäudesimulation) stellt hierbei eine der größten Herausforderungen dar und müssen entsprechend auch bei der Modellierung berücksichtigt werden (z.B. Wahl der Zeitgranularität).

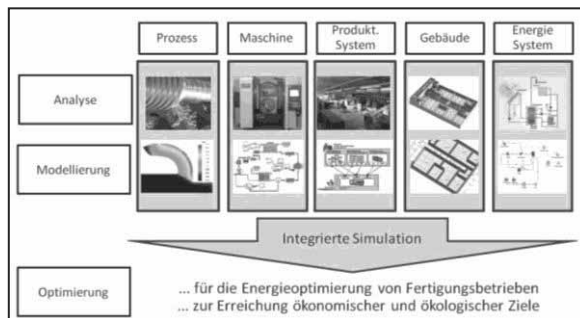


Abbildung 1: Methodischer Zugang mit Analyse und Modellierung der einzelnen Handlungsbereiche, gefolgt von einer integrierten Gesamtsimulation.

Einer Top-down-Sichtweise auf das Gesamtsystem folgend, werden im nächsten Abschnitt der Co-Simulations-Ansatz und das Software-Framework präsentiert. Kapitel 3 beschreibt schließlich Aspekte der einzelnen Teilmodelle.

2 Co-Simulation

Der interdisziplinäre Zugang zur energetischen Analyse von Produktionsbetrieben verlangt eine vorausgehende Formalisierung der Systemstruktur mit ihren Komponenten, Schnittstellen, Abhängigkeiten und charakteristischen Parametern. Die resultierende generische Systembeschreibung wird detailliert präsentiert in [1]. Da derzeit kein vollständiges Simulationstool zur Verfügung steht, das eine Simulation des gesamten Produktionsbetriebes mit ausreichender Komplexität, Detailgrad sowie Rechengeschwindigkeit erlaubt, kann durch die Kombination von mehreren Simulationsumgebungen im Rahmen einer sogenannten Co-Simulation (cooperative simulation) bessere Ergebnisse liefern [2]. Dabei implementiert jeder der beteiligten Simulatoren einen abgegrenzten Teil des Gesamtsystems (z.B.: Fertigungsmaschinen, Energiesystem Gebäude) und tauscht zur Laufzeit periodisch Ein- und Ausgangsvariablen mit den anderen Simulatoren aus. Die zeitliche Synchronisation und Kommunikation wird dabei meist von einer zusätzlichen Software (sog. Middleware) koordiniert.

Dieser Ansatz einer Co-Simulation verspricht mehrere wichtige Vorteile: Vom Standpunkt der Modellierung lässt sich jedes der Teilmodelle mit einer speziell dafür geeigneten Modellbeschreibung (z.B. Differentialgleichungen, datenbasiert, discrete event) in einer dafür ausgelegten Simulationsumgebung implementieren, was nicht nur Zeit bei der Modellumsetzung spart, sondern auch in Bezug auf die numerische Simulation effizientere Modelle liefert. Von Seiten der numerischen Berechnung lassen sich verschiedene Algorithmen (z.B. ODE-Solver, Dateninterpolation), die in den jeweiligen Simulationstools ausgeführt werden, kombinieren und auf die speziellen Bedürfnisse (z.B. Schrittweiten, implizite vs. explizite Berechnung) des jeweiligen Teilmodells bzw. deren Dynamik anpassen. Die einzelnen Experten können außerdem nicht nur ihr bevorzugtes Simulationstool verwenden, sondern eventuell auch auf bestehende Vorarbeiten (z.B. Modellbibliotheken) aufbauen, bzw. wird die Weiterverwertung von entwickelten Modellen erleichtert.

Für die Umsetzung einer Co-Simulation ist u.a. ein prototypisches Software-Framework (Middleware) als Open-Source Software verfügbar, das vom Lawrence Berkeley National Laboratory an der University of California entwickelt wird [3]. Das sog. Building Controls Virtual Test Bed (BCVTB) erlaubt eine Laufzeitkopplung einiger in der Gebäudesimulation gebräuchlicher Simulationsumgebungen, u.a. EnergyPlus, Dymola und MATLAB. Der Datenaustausch für die Co-Simulation erfolgt in BCVTB gemäß einer Client/Server-Struktur zu vorab festgelegten Zeitpunkten. Für einen Überblick über die Software-Architektur sei auf [3] verwiesen.

Mithilfe von BCVTB konnte eine prototypische Implementierung der vorgestellten Gesamtsimulation technisch umgesetzt werden, das folgende Teilmodelle umfasst: (1) MATLAB für datenbasierte Modelle von Maschinen und Produktionssystem, (2) Dymola für Modelle des Energiesystems und (3) EnergyPlus für das Gebäudemodell. Dieses Framework erlaubt es schließlich verschiedene Szenarien zu evaluieren für tiefergehende Analysen verschiedener Energiesparmaßnahmen und deren dynamischer Einfluss auf das Gesamtsystem. Durch die in der generischen Modellbeschreibung klar definierten Schnittstellen lassen sich individuelle Teilmodelle für einzelne Szenarienrechnungen ersetzen ohne die Implementierungen der restlichen Systemteile zu beeinflussen.

3 Optimierungsfelder und Teilmodelle

3.1 Prozess, Maschine und Produktionssystem

Neue objektorientierte Modellierungsansätze für physikalische Systeme (z.B. Modelica, Simscape) erlauben es modulare und einfach erweiterbare Modelle basierend auf wiederverwendbaren Modellklassen zu entwickeln. Die zugrunde liegende akasale Modellbeschreibung mittels Energieflüssen bietet dabei die Möglichkeit Domänenübergreifende Modelle zu beschreiben, die z.B. elektrische, mechanische und thermische Aspekte vereinen [4]. Mithilfe derartiger Simulationsmodelle können z.B. detaillierte Untersuchungen von Produktionsprozessen und Analysen der Energieflüsse innerhalb von Werkzeugmaschinen durchgeführt werden [5].

Um allerdings eine größere Anzahl an Produktionsmaschinen zusammen mit Aspekten von Gebäudehülle und Energiesystem simulieren zu können, ist es notwendig die Komplexität der verwendeten Maschinenmodelle zu reduzieren und sich auf die jeweilige Betrachtung relevanten Energieflüsse zu konzentrieren. Ein Ansatz dafür ist die Verwendung von vereinfachten Parametermodellen mit einer deutlich geringeren zeitlichen Auflösung (z.B. Minuten anstatt von ms) und die Identifikation von energierelevanten Parametern aus höher auflösenden Messdaten bzw. anderen detaillierten Simulationsmodellen. Dazu können beispielsweise der Energieverbrauch einer Maschine in verschiedenen Bearbeitungszuständen gemessen und daraus Durchschnittswerte für Grundlast, dynamischer Energieanteil sowie Bearbeitungsenergie extrahiert werden. Zeitliche Anteile der jeweiligen Energieniveaus lassen sich dabei aus detaillierten Simulationen von Lastprofilen einzelner Produktionsprozesse ermitteln. In einer Kombination dieser beiden Methoden wäre es darüber hinaus außerdem möglich die beschriebene Parametrisierung während der Simulation dynamisch anzupassen, indem ein analytisches Simulationsmodell für einen kurzen Zeitraum außerhalb des Co-Simulations-Frameworks ausgeführt wird (z.B. einmalige Simulation von sich wiederholenden Produktionsprozessen) und aus den Resultaten aktuelle Parameterwerte für die Co-Simulation extrahiert werden.

Die Implementierung der Modelle für Produktionsprozess und Maschinen erfolgte mittels MATLAB und Excel. Dieses Modell berechnet den Energiebedarf, Wärmeeintrag in die Umgebung sowie rückgewinnbare Wärmemenge, welche schließlich über die

Co-Simulation weiter in die Teilmodelle für Gebäude und Energiesystem einfließen.

3.2 Energiesystem

Das Teilmodell für das Energiesystem beschreibt sämtliche Komponenten für die Versorgung der Anlage mit elektrischer sowie thermischer Energie, z.B. Pumpen, Kälteanlagen, Wärmepumpen, etc. Das Modell berechnet den Primär- bzw. Endenergiebedarf aus dem Bedarf der Produktionsanlage sowie des Gebäudes. Die interdisziplinäre Co-Simulation stellt spezielle Anforderungen an die Modellimplementierung. Neben einer Simulationsumgebung, die einen Datenaustausch zur Laufzeit mit anderen Simulationstools zulässt, müssen auch ausreichende Rechengeschwindigkeit und numerische Stabilität adressiert werden.

Die Implementierung der Modelle für das Energiesystem erfolgte in Dymola. Aufgrund der erwähnten Anforderungen wurde allerdings anstatt einer akasalen objektorientierten Beschreibung auf eine klassische signalflossbasierte Modellbeschreibung zurückgegriffen. Die Modelle wurden validiert mittels Daten von Prüfzeugnissen bzw. verfügbarer Literatur.

3.3 Gebäude

Ein weiteres Teilmodell beschreibt die geometrischen und thermischen Aspekte der Gebäudehülle, welche die Produktionsmaschinen und das Energiesystem beheizt, und berücksichtigt neben Wärmeeinträgen von Maschinen, Personen, Beleuchtung und anderer technischer Ausrüstung auch Umwelteinflüsse (Außentemperatur, Wetter, etc.). Das Modell berechnet aus dem Vergleich mit voreingestellten Temperaturwerten (Soll-Werte) den resultierenden Heiz- bzw. Kühlenergiebedarf.

Für die Modellimplementierung wurden Werkzeuge und Methoden des Building Information Modeling (BIM) herangezogen, welche es erlauben im Zuge einer integralen Gebäudeplanung neben den eigentlichen Geometrieinformationen auch sämtliche relevanten Design- und Projektdaten digital zu verwalten.

4 Anwendungsbeispiel

Im Rahmen einer realen Fallstudie konnte das entwickelte Simulationswerkzeug im Planungsprozess eines neuen Fabrikgebäudes inkl. Bürobereich für einen metallverarbeitenden Fertigungsbetrieb eingesetzt werden (ca. 500 Beschäftigte, 48 Werkzeugmaschinen, 8 Mio. kWh jährlicher Energieverbrauch).

Ein bereits vorhandener Maschinenpark soll dabei weitgehend übernommen und an einen neuen Standort transferiert werden, welcher neben einem flexiblen Layout auch hohe Anforderungen an Energieeffizienz stellt.

Der vorhandene Maschinenpark bestehend aus Bearbeitungszentren, Härteöfen und Laserschneidanlagen wurde mit Hilfe von umfangreichen Mess- und Produktionsdaten sowie Detailberechnungen einzelner Komponenten als datenbasiertes Modell abgebildet und validiert. Detailsimulationen von neun verschiedenen Szenarien der Gebäudehülle gepaart mit angepassten Luftwechselraten und Beleuchtungs- bzw. Verschattungsstrategien lieferten Aussagen über die hinsichtlich Heiz- und Kühlenergiebedarf beste Variante [6]. Für das Energiesystem wurden drei unterschiedliche Szenarien definiert und mittels Co-Simulation verglichen. Eine anschließende Bewertung der Resultate nach ökonomischen sowie ökologischen Gesichtspunkten lieferte schließlich Anhaltspunkte für entsprechende Investmententscheidungen.

5 Fazit

Die beschriebene Co-Simulation stellt ein anwendbares Werkzeug dar für die ganzheitliche Untersuchung und Vorhersage des Energiebedarfs von Produktionsbetrieben in den frühen Planungsphasen. Das Anwendungsbeispiel zeigt, dass dieser Ansatz hilft die Abhängigkeiten zwischen den einzelnen Teilsystemen hervorzuheben und so zusätzliches Optimierungspotential zu erschließen. Beispielsweise konnte in dem Beispiel gezeigt werden, dass die Menge an rückgewonnener Energie aus der Maschinenabwärme nicht nur durch das zur Verfügung stehende Potential aus der Produktion, sondern vor allem durch die Kapazität des aufbereitenden Energiesystems bestimmt wird. Der Ansatz der gekoppelten Simulation stellt allerdings auch spezifischen Anforderungen und die integrierten Teilmodelle. Parameterbasierte Modelle können hier helfen die Lücke zwischen Genauigkeit und Vereinfachung zu überbrücken. Die Vereinheitlichung der unterschiedlichen Zeitgranularitäten stellt hier eine der größten Herausforderungen dar.

6 Ausblick

Aufbauend auf bisherigen Erkenntnissen versuchen aktuelle Arbeiten in Zusammenarbeit mit Industriepartnern eine ganzheitliche Methodik und eine Software-Tool-Kette zu entwickeln, mit Hilfe derer basierend auf Simulationen und aggregierten Realdaten

Produktionsanlagen im operativen Betrieb energetisch überwacht, gesteuert und deren Energieverbrauch prognostiziert werden kann. Damit soll es möglich werden optimierte Betriebsführungsstrategien abzuleiten unter Berücksichtigung des Zusammenspiels zwischen Produktionsbedarf, Energiebedarf und Verfügbarkeit.

Danksagung

Diese Arbeit wird unterstützt durch den österreichischen Klima- und Energiefonds im Rahmen des Forschungs- und Technologieprogramms e!MISSION.at (FFG Projekt Nr. 3687399). Die Autoren bedanken sich bei allen akademischen und Industriepartnern für ihre Mitarbeit.

Referenzen

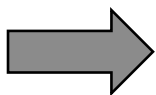
- [1] I. Leobner, G. Neugschwandtner, K. Ponweiser und W. Kastner. *Energy efficient production – a holistic modelling approach*. Tagungsband World Congress on Sustainable Technologies, S. 73-78, 2011.
- [2] C. Brecher, M. Esser und S. Witt. *Interaction of manufacturing process and machine tool*. CIRP Annals – Manufacturing Technology, 58(2):588-607, 2009.
- [3] M. Wetter. *Co-simulation of building energy and control systems with the Building Controls Virtual Test Bed*. Journal of Building Performance Simulation, 4(3):185-203, 2010.
- [4] B. Heinzl, M. Landsiedl, N. Popper, A. Dimitriou, F. Dür, F. Bleicher, C. Reinisch und F. Breitenacker. *Object-oriented multi-domain modelling of machine tools: a case study*. Tagungsband 24th European Modeling and Simulation Symposium, S. 471-476, 2012.
- [5] B. Heinzl. *Objektorientierte Multi-domain Modellierung und Simulation von Werkzeugmaschinen*, Diplomarbeit, TU Wien, Österreich, 2012.
- [6] I. Kovacic, K. Orehounig, A. Mahdavi, F. Bleicher, A. Dimitriou und L. Waltenberger. *Energy efficient production – interdisciplinary systemic approach through integrated simulation*. Strojarski, 55 (1) 17-34, 2013.



A MODELICA BASED LITHIUM ION BATTERY MODEL

MOTIVATION

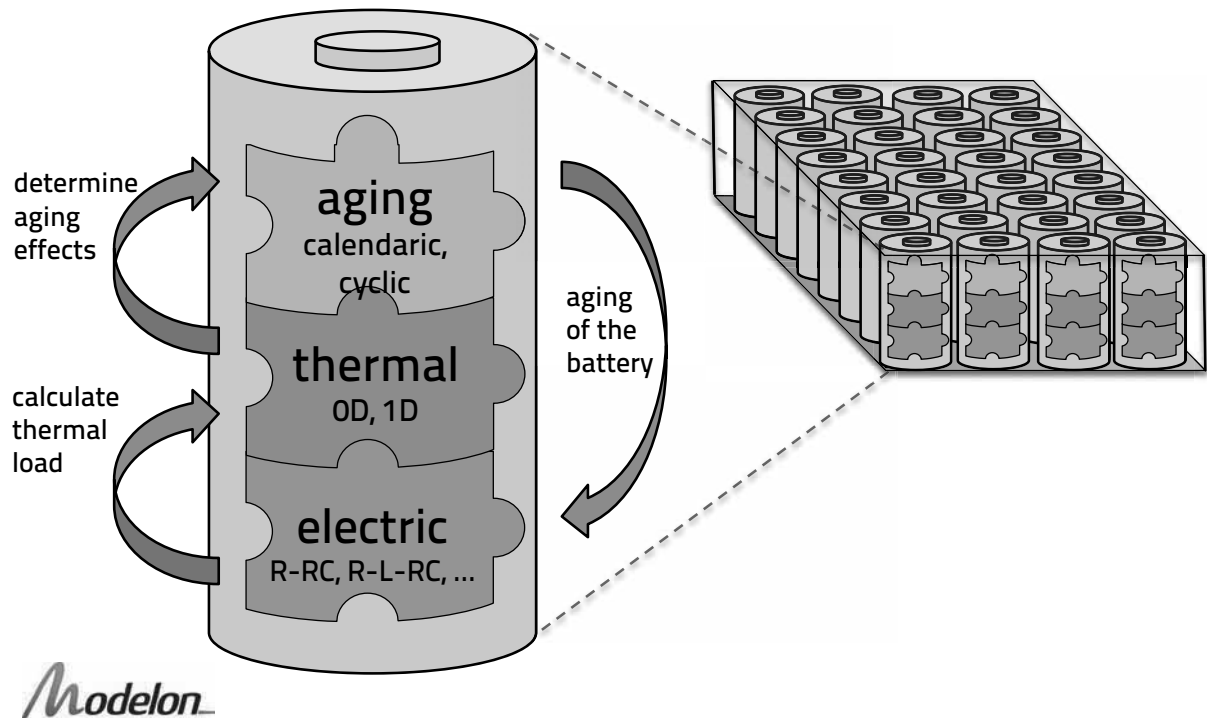
- Lithium ion cell: state of the art for electric power trains
- Which cell type fits best?
- Design of pack structure?
- Need for integrated battery design: electric, thermal and aging performance



Modelica library
for batteries



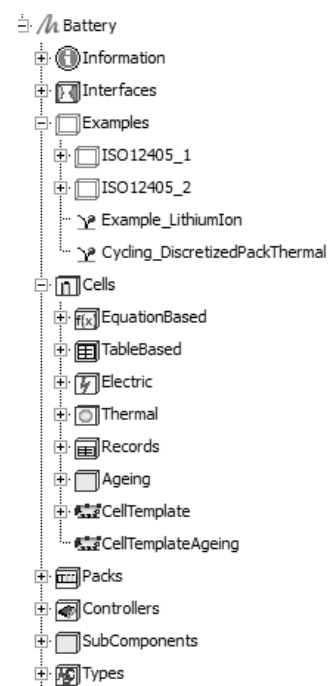
BATTERY LIBRARY



LIBRARY CONTENTS

- Examples (ISO-Norms ready to use)
- Cells
 - table-based
 - equation-based
 - calendaric and cyclic aging
- Packs
 - discretized and scaled
 - thermal models for housing and heat transfer
- Controllers
 - blocks for current, power and temperature control

Modelon



ELECTRIC MODEL

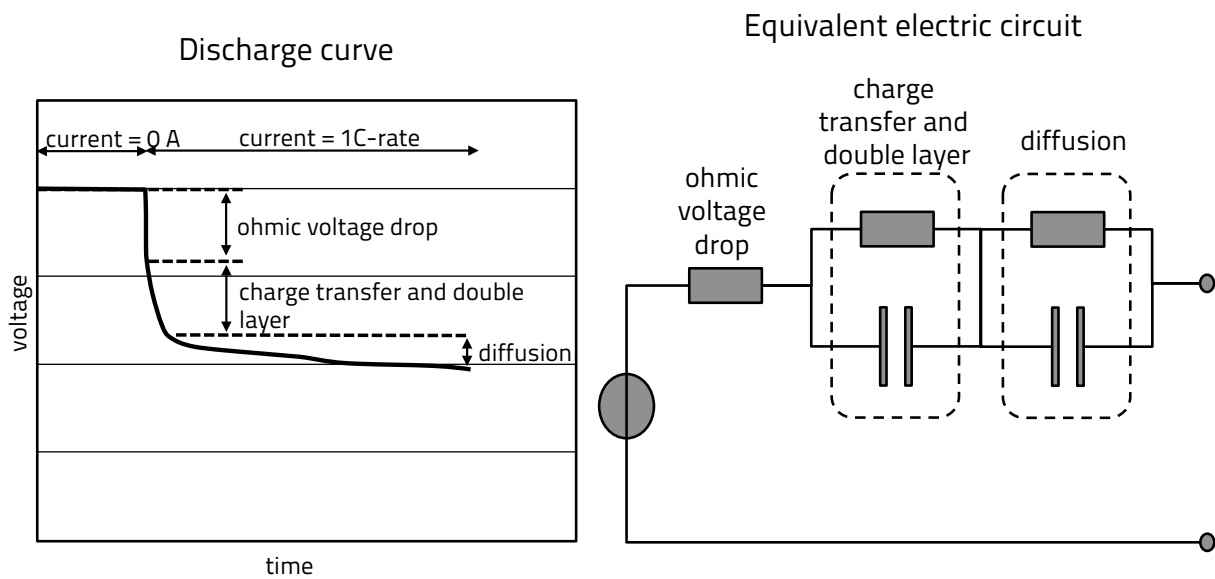
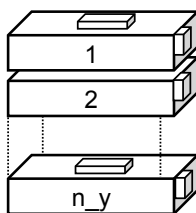
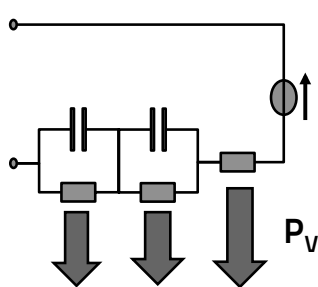


figure adopted from „Moderne Akkumulatoren richtig einsetzen“ by A.Jossen and W.Weydanz

Modelon

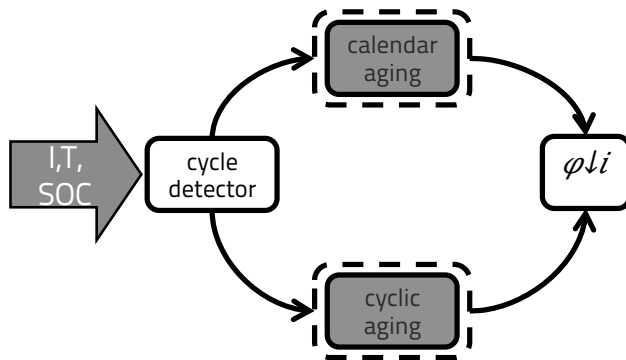
THERMAL MODEL



- Heat loss mainly due to Joule effect
- Heat capacity and conductivity are calculated based on material properties and geometry
- Conditional heatPorts: where is heat transferred?
- Simple heat transfer correlations for convection and radiation

Modelon

AGING MODEL

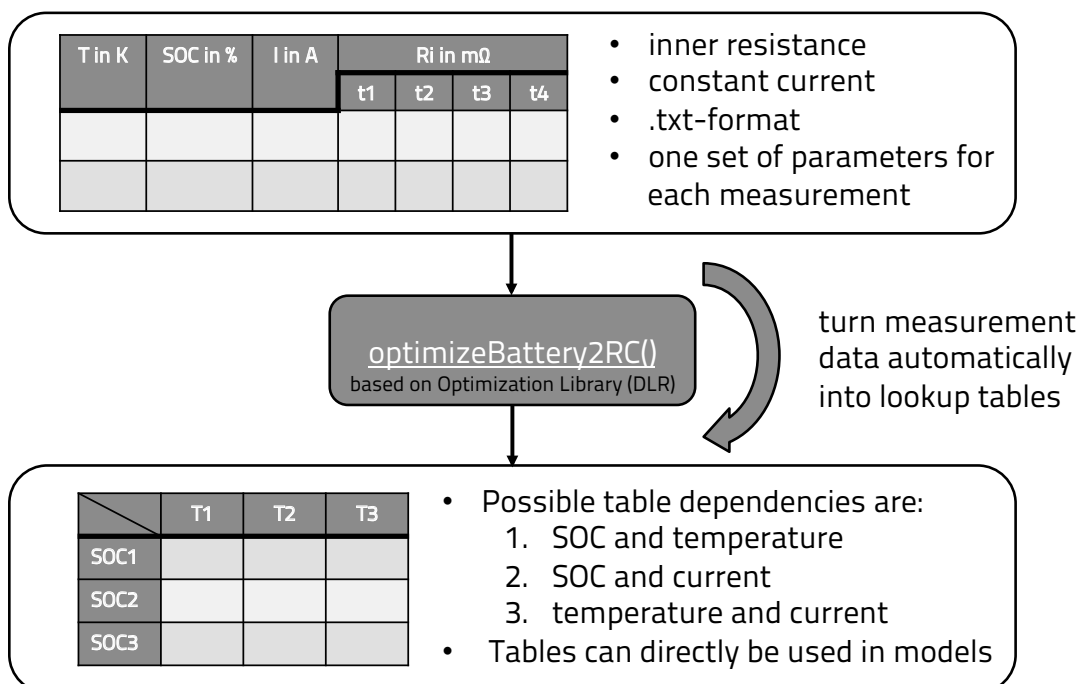


- Replaceable generic aging models
- Parameters based on measurement data
- Detects start/end of cycle
- Calculates aging factors with mean values for DOD, I, T, OCV



Modelon

AUTOMATIC PARAMETRIZATION

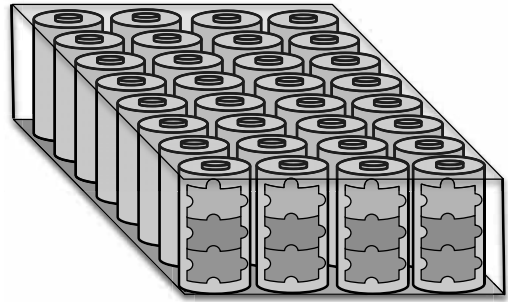


Modelon

BATTERY PACKS

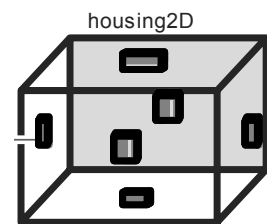
Scaled:

- One instance of cell model = same temperature, SOC, initial values throughout pack
- Scaled outputs
- Fast
- Use case: mileage or lifetime calculation



Discretized

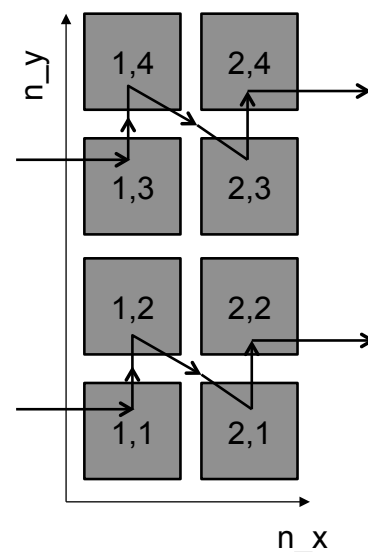
- Array with $[n_x, n_y]$ cells with individual states
- Conditional heat transfer inbetween cells via pins or cell wall
- Use case: design of thermal management system



Modelon

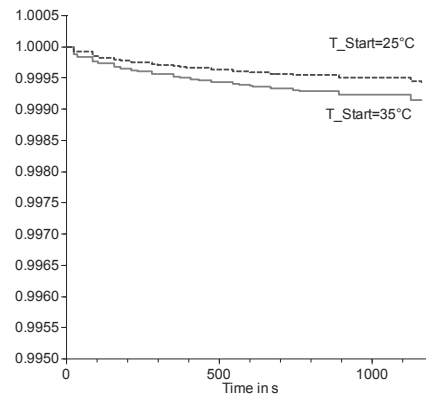
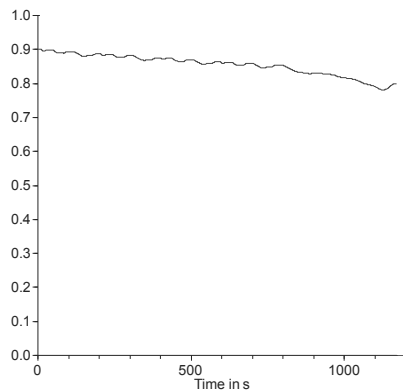
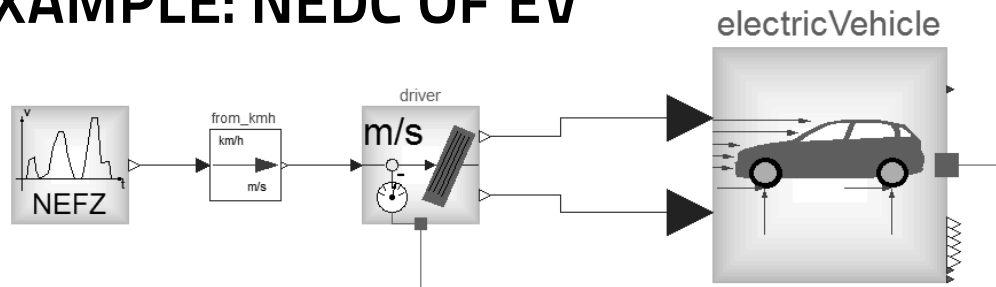
DISCRETIZED PACKS: ELECTRIC CONNECTION

- Geometric layout \neq electric connection
 - Implemented via Matrix M ($n_{\text{parallel}}, n_{\text{serial}}$)
 - Assure: $n \downarrow x \ n \downarrow y = n \downarrow s \ n \downarrow p$
 - Set-up of M difficult for large number of cells
- \Rightarrow Typical layouts can be constructed via given functions



Modelon

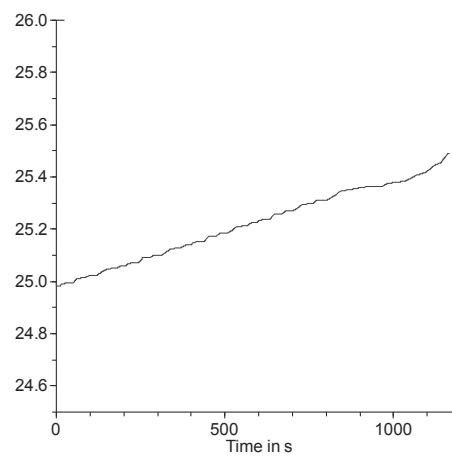
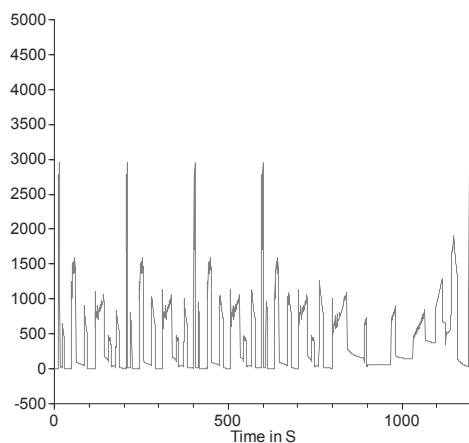
EXAMPLE: NEDC OF EV



Modelon

EXAMPLE: NEDC OF EV

Thermal behaviour of cell



Modelon

CONCLUSION

- Library for multiphysical aspects of battery packs and cells
- High variability of modeled effects and modeling depth
- Customer friendly library structure
- Calculates mileage, SoC, SoH, temperature
- Compatible to other Modelica libraries

Contact:

imke.krueger@modelon.com

sales.germany@modelon.com



Untersuchung eines alternativen Aktorkonzepts für aktive Bedienelemente auf Basis von Open-Source Simulationsprogrammen

Nanno Peters^{1,a}, Gerhard Stebner^{1,b}, Christoph Hartwig^{1,c}

¹Ostfalia Hochschule für angewandte Wissenschaften,
Salzdahlumer Str. 46/48, 38302 Wolfenbüttel, Deutschland

^a*n.peters@ostfalia.de*, ^b*ge.stebner@ostfalia.de*, ^c*ch.hartwig@ostfalia.de*

Komplexer werdende Systeme erfordern immer häufiger den Einsatz von aktiven X-by-Wire Bedienelementen, um den Bedienern Informationen direkt beim Betätigen über den haptischen Sinneskanal zu übermitteln. Diese Art der Informationsübertragung, ohne akustisches und visuelles Feedback an den Nutzer, erfordern eine störungsarme Aktorik sowie ausreichend Stellmoment. Das Paper beschreibt die Analyse einer permanenten Synchronmaschine (PMSM) aus der professionellen Luftbildfotografie hinsichtlich der Tauglichkeit als aktiver Drehschalter. Der High-Torque-Motor in Außenläuferausführung wird mit einer magnetischen Finite-Elemente-Methode (FEM) hinsichtlich des Nutz- und Nutrastmoments untersucht und den psychophysikalischen Schwellwerten gegenübergestellt. Im Anschluss wird kurz auf den aktuellen Stand der Entwicklung eingegangen.

1 Einleitung und Hintergrundwissen

In vielen technischen Bereichen werden heutzutage Systeme, welche früher mechanisch mit dem dazugehörigen Bedienelement gekoppelt waren, für eine Bauraum- und Gewichtsreduzierung entkoppelt und elektronisch angesteuert. Die X-by-Wire Strategie führt dazu, dass Systemzustände nicht mehr direkt an den Nutzer über den taktilen Sinn übertragen werden können. Dieser Verlust von haptischen Informationen, kann ohne Kompensation dazu beitragen, dass der Benutzer eines Systems Zustände falsch interpretiert und falsche Entscheidungen trifft.

Um den direkten Informationsfluss der Systemzustände zum Bediener wiederherzustellen, kommen immer häufiger aktive Bedienelemente zum Einsatz. Diese gewünschte Funktionalität der Betätigungseinrichtungen erfordern mindestens einen Aktor, um elektronisch eingreifen zu können (Abbildung 1).

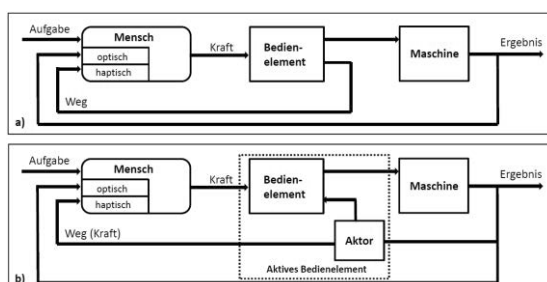


Abbildung 1. Schema eines passiven Schalters (a) und eines aktiven Bedienelements (b)

Diese Aktorik besteht zurzeit primär aus DC-Motoren, die nur kleine Momente aufzubringen können und so den Nutzer auf seine Fehlbedienung hinweisen. Der aktive Momentenbereich liegt hier bei bis zu 0,06Nm [1][2][3]. Diese aktiven Bedienelemente werden heutzutage fast ausschließlich bei der Infotainmentsteuerung im Kfz eingesetzt.

Das Ziel ist es, größere Momente als bisher bereitzustellen, damit auch Primärfunktionsbedienelemente mit einer ‚stärkeren‘, rein aktiven Haptik eingesetzt werden können. Darunter ist zu verstehen, dass keine konventionelle, passive Kombination aus Feder und Nut-Konturscheibe verbaut werden soll, sondern der Momenten-Drehwinkel-Verlauf (Abbildung 2) von einem aktiv geregelten Moment erzeugt wird.

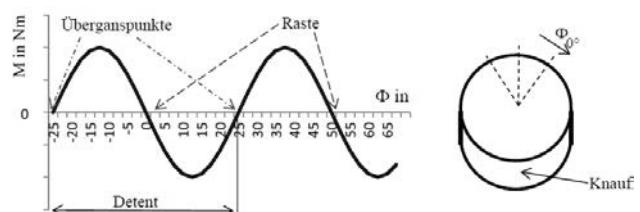


Abbildung 2. Allgemeine Darstellung einer Moment-Drehwinkel-Kennlinie

Der Einsatz größerer DC-Motoren ist praktisch nicht zu vertreten, da diese für den Dauereinsatz im Anlauf nicht ausgelegt sind und extrem überdimensioniert werden müssten.

2 Neuartiger Low-Budget, High-Torque-Motor

Im der professionellen Luftbildfotografie ist ein neuer Trend erkennbar. Um eine Entkopplung zwischen der Roll- und Nickbewegung des Fluggerätes und der Kamera zu erzeugen, werden vermehrt umgewickelte Außenläufermotoren eingesetzt (Abbildung 3).



Abbildung 3. 2-Achs-Kameragimbal mit BLDC [4]

Allgemein handelt es sich um eine permanenterrechte Synchronmaschine in Außenläuferausführung, die umgangssprachlich Outrunner genannt wird. Durch die hohe Anzahl an Polen und den großen Luftspaltdurchmesser entstehen Motoren mit einer hohen Drehmomentdichte. Tabelle 1 zeigt die typischen Herstellerangaben eines solchen Motors.

Gewicht	98g
Abmessungen	45mmx25mm
Widerstand	12.4Ω
Windungszahl	120 turns
Cu-Querschnitt	0,023mm ²
Hersteller	iFlight
Bauweise	24N22P
Kamera-Gewicht	600-1200g

Tabelle 1. Eigenschaften des GBM4108-120T

Die spärlichen technischen Daten der Hersteller ermöglichen keine direkte Vergleichbarkeit. Dies gilt auch für die Vergleichbarkeit innerhalb der Produktpalette eines Herstellers. Einzig über das mögliche Kamera-Gewicht und die Motordimensionen können

vereinfacht Rückschlüsse auf das Moment gezogen werden. Um eine Bewertung des Motors zu ermöglichen, sind diese Informationen jedoch nicht Aussagekräftig genug.

Zusätzlich zu den fehlenden Nenngößen (Nennspannung, Nenndrehzahl, Nennstrom, Nennmoment, Wirkungsgrad, etc.) fehlt die Angabe des Nutrastmoments. Dies ist für die Anwendung im aktiven haptischen Bedienelement von großem Interesse, da der Bediener auch sehr kleine Störungen wahrnehmen kann.

Daher wurden zehn verschiedene Motoren ähnlicher Baugröße von den Autoren und weiteren Testpersonen hinsichtlich der gefühlten, subjektiven Intensität des Nutrastmoments bewertet. Auffällig dabei waren die großen Abweichungen innerhalb der Motorenauswahl.

Es wurde einstimmig das Nutrastmoment des GBM4108-120T der Fa. iFlight als am geringsten wahrnehmbar empfunden. Um das tatsächliche Potential des Motors einschätzen zu können, müssen weitere Parameter ermittelt werden. Die Abbildung 4 zeigt den zu untersuchenden Motor.



Abbildung 4. Ausgewählter Motor

3 Technische Untersuchung des GBM4108-120T

Zur Untersuchung des Motors wird dieser in die FEM-Software FEMM 4.2 eingegeben. Hierzu mussten die Herstellerangaben um weitere Informationen erweitert werden.

3.1 Eigenschaften der Wicklung

Die elektrischen Eigenschaften des Motors werden vom Hersteller kaum beschrieben. Aus der Modellbe-

zeichnung geht lediglich hervor, dass die 24 Pole des Stators mit jeweils 120 Windungen versehen sind. Da es sich um eine dreiphasige Maschine handelt, besitzt jeder Strang 8 Pole.

Für die Ermittlung der Grundverschaltung wurde an zwei der drei Außenleiter Spannung angelegt und der Temperaturanstieg mit Hilfe einer Wärmebildkamera beobachtet. Die Abbildung 5 zeigt die Wärmeentwicklung in den Spulen.

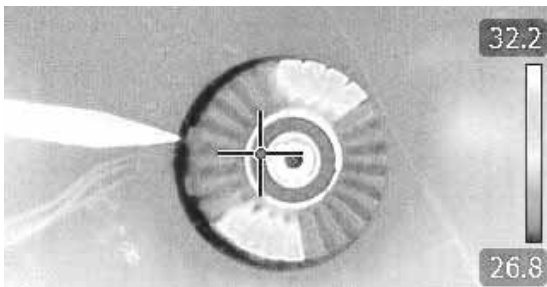


Abbildung 5. Wärmebild des PMSM-Stators bei zwei Phasen Spannungsversorgung

Die Pole des Stators weisen dabei zwei Zonen mit einer Temperatur von ca. 32°C auf. Diese Zonen lassen sich eindeutig auf 8 Pole des Stators zuordnen. Die restlichen 16 Pole weisen eine homogene Temperatur auf, die deutlich niedriger ist aber trotzdem über der Raumtemperatur liegt. Die Aufteilung der Zonen ist charakteristisch für die Dreieckschaltung, da bei der Bestromung von zwei Außenleitern trotzdem Strom durch alle Stränge gelangt. Zum Vergleich würden bei einer Sternschaltung 16 Pole homogen erwärmt werden, während die restlichen 8 Pole keine Eigenerwärmung aufweisen würden.

Der gemessene Widerstand zwischen zwei Außenleitern liegt bei 14,2Ω. So ergibt sich der Strangwiderstand zu 21,3Ω.

Aus Abbildung 5 ist zu erkennen, dass es sich um eine Zahnspulenwicklung handelt. Zahnspulenwicklungen erzeugen im Luftspalt eine Vielzahl von Oberwellen, welche es erlauben den Stator mit unterschiedlichen Polpaarzahlen der Magnete zu betreiben. Mit einem Bewicklungsrechner kann man die sinnvollen Kombinationen für die Nut- und Polzahlen errechnen und dafür ein Wicklungsschema darstellen. Die untersuchten Motoren weisen 24 Nuten und 22 Pole auf. Der Wicklungsfaktor der Hauptwelle beträgt 0,95.

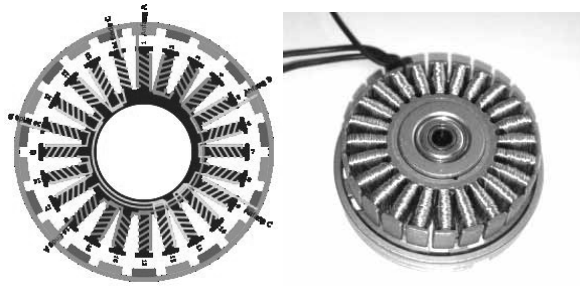


Abbildung 6. Wicklungsschema des Motors [6]

3.2 Eigenschaften des Magnetkreises

Anders als die Eigenschaften der Wicklung sind die magnetischen Eigenschaften des Motors nur schwer zu identifizieren. Da der Hersteller keine Angaben zum verwendeten Stahl macht, wird von einem Elektroblech mittlerer Güte ausgegangen.

Das verwendete Magnetmaterial im Rotor (siehe Abbildung 6) ist ebenfalls nicht bekannt. Da es sich um einen Motor im mittleren Preissegment handelt, wird von Magneten ausgegangen, die eine remanente Flussdichte von 1,0T aufweisen. Dies ist bei vergleichbaren Modellen anderer Hersteller ein üblicher Wert.



Abbildung 7. Aufbau des Rotors

3.3 2D-FEM mittels FEMM 4.2 und Scilab

Die aus den Abschnitten 3.1 und 3.2 gewonnenen Erkenntnisse werden nun in FEMM 4.2 eingegeben. Scilab wird zur Steuerung der Software FEMM 4.2 verwendet. Hierdurch kann auf parametrisierte Geometrien zurückgegriffen werden und das Postprocessing automatisiert werden. Die dazu benötigte Schnittstelle heißt SciFEMM.

FEMM 4.2 ist eine GNU lizenzierte Software des Programmiers David Meeker. Mit FEMM 4.2 lassen sich auf Basis der Maxwell'schen Gleichungen stationäre magnetische Kreise berechnen. Durch die 2D-Berechnung werden jedoch Wickelköpfe bei elektrischen Maschinen vernachlässigt. Ebenso ist es nicht möglich, transiente Berechnungen auszuführen, was die Berechnung von Wirbelstrom- und Hystereseverlusten direkt in FEMM verhindert.

Zur Berechnung des Motors wird auf mehrere stationäre 2D-Berechnungen gesetzt, über welche der Rotor kontinuierlich gedreht wird. Hieraus lässt sich unmittelbar das Nutrastrmoment über dem Drehwinkel ermitteln. Ebenso kann das Nutzmoment über dem Polradwinkel durch Bestromung der Wicklungsflächen ermittelt werden.

Während das Nutzmoment relativ robust gegenüber der Vernetzungsqualität und den Solver-Einstellungen ist, reagiert das Nutrastrmoment relativ stark auf Fehler bei der Diskretisierung. Dies resultiert daraus, dass die Nutrastrmomente aus vielen betragsmäßig hohen Einzelkräften bestehen, die zwischen den einzelnen Magneten und Statorpolen wirken. Dies ist numerisch sehr ungünstig, weshalb ein besonderes Augenmerk auf die Vernetzung gelegt wurde.

Alternativ können die Nutrastrmomente aus der Änderung der im Magnetkreis gespeicherten Energie nach Gleichung (1) berechnet werden.

$$M_{Nutrastr.} = l \frac{dE_{mag}}{d\varphi} \quad (1)$$

Zur Berechnung der induzierten Spannung wird im Postprocessing der verkettete Fluss der Stränge nach dem Winkel abgeleitet und mit einer vorhandenen Drehzahl verrechnet. Hierfür kann in FEMM für je einen Strang eine Schaltung hinterlegt werden, welche den verketteten Fluss für die Stränge automatisch ausgibt.

3.4 Ergebnisse der FEM-Analyse und Validierung

Die einfachste zu validierende Größe aus der FEM-Analyse ist die induzierte Spannung. Hierzu wird der Motor mit etwa 1050 U/min angetrieben und mittels eines Speicher-Oszilloskops die Spannung zwischen zwei Außenleitern aufgezeichnet. Abbildung 8 zeigt den direkten Vergleich zwischen der gemessenen und der errechneten induzierten Spannung.

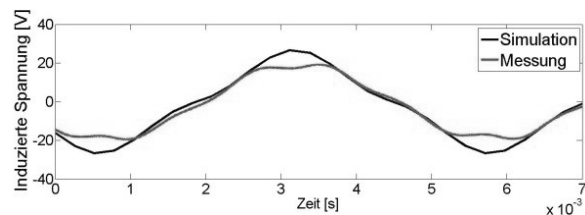


Abbildung 8. Induzierte Spannung über der Zeit

Der deutliche Unterschied der Amplituden der Hauptwellen deutet daraufhin, dass aufgrund der sehr flachen Bauweise eine Vernachlässigung der Wickelköpfe und die Idealisierung des Feldes durch die 2D-Berechnung nur bedingt zulässig sind. Daher wird der Faktor k_{eff} eingeführt. Dieser setzt die effektive und tatsächliche Paketlänge des Stators ins Verhältnis.

$$k_{eff} = \frac{l_{eff}}{l} = \frac{\hat{U}_{mess}}{\hat{U}_{sim}} = \frac{17,9V}{27,V} = 0,663 \quad (2)$$

Der Faktor k_{eff} dürfte sich ebenso stark auf das Nutzmoment auswirken, welches in Ermangelung eines Messaufbaus noch validiert werden muss. Abbildung 9 zeigt das errechnete Moment über dem Polradwinkel bei einem Strangstrom von 1A und der Multiplikation mit dem Faktor k_{eff} .

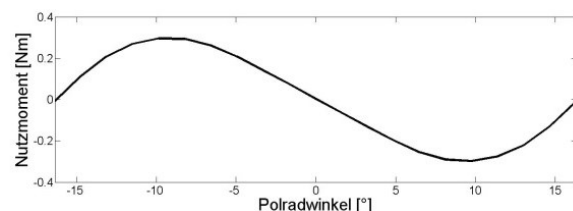


Abbildung 9. Nutzmoment über Polradwinkel

Es ist zu erkennen, dass der Motor im Anlauf so ein theoretisches Moment von 0,32Nm bei einer angelegten Spannung von 14,2V erzeugen kann. Der Eisenkreis befindet sich noch nicht in der Sättigung, aber die herrschende Stromdichte von $50 \frac{A}{mm^2}$ begrenzt eine weitere Steigerung des Moments.

Wie in Abbildung 10 zu sehen ist, weist der Motor ein errechnetes Nutrastrmoment mit einer Amplitude von 0,0001Nm auf. Dies ist ein sehr geringer Wert im Verhältnis zum Nutzmoment.

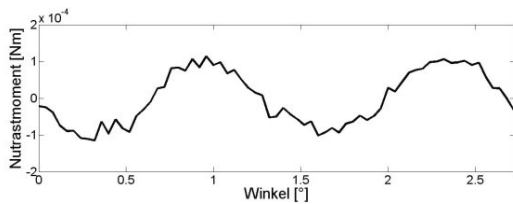


Abbildung 10. Berechnetes Nutrastmoment über dem Winkel

Da der Motor primär nach der gefühlten Intensität des Nutrastmoments ausgewählt wurde, ist es von besonderer Bedeutung, ob das Nutrastmoment noch weiter reduziert werden kann oder ob der Hersteller den Motor nahe am Optimum ausgelegt hat. Eine Variation der Polschuh- und Magnetbreite in FEMM 4.2 hat gezeigt, dass die Auslegung des Herstellers nahe dem theoretischen Optimum liegt. Somit bestätigt sich, dass der Motor hinsichtlich der Nutrastmomente geeignet ist, um als vorläufiger Aktor für ein aktives Bedienelement eingesetzt zu werden. Eine weitere Reduzierung der Nutrastmomente muss, wenn nötig, durch eine geeignete Ansteuerung erfolgen.

4 Vergleich mit haptischen Kenngrößen

Der aktive Arbeitsbereich des GBM4108-120T im Drehsteller wird voraussichtlich bei Momenten zwischen 0,06 bis 0,16 Nm liegen. Die theoretischen psychophysikalischen Schwellwerte (Differenzschwelle) in diesem Bereich liegen bei 12% des Referenzdrehmoments [7]. Laut E.H. Weber kommt noch hinzu, dass auf einen Stimulus ϕ eine Stimulus-Änderung $\Delta\phi$ erst ab einem bestimmten Betrag bezogen auf ϕ wahrgenommen wird. Dieser Effekt ist das sogenannte Webersche Gesetz ($k = \Delta\phi / \phi$). Nur niedrige Stimuli bilden eine Ausnahme in der Höhe des absoluten Schwellwertes (Abbildung 11).

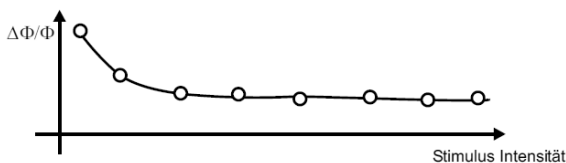


Abbildung 11. Ermittlung der Differenzschwelle $\Delta\phi$ bezogen auf den Stimulus ϕ [7]

Im Betrieb dürfte das Nutrastmoment so gar nicht mehr wahrgenommen werden. Psychophysikalisch problematisch ist einzig die Zunahme der menschlichen, haptischen Auflösung im Frequenzbereich um 100-300 Hz. Hier nimmt die Empfindlichkeit des Menschen stark zu (Abbildung 12).

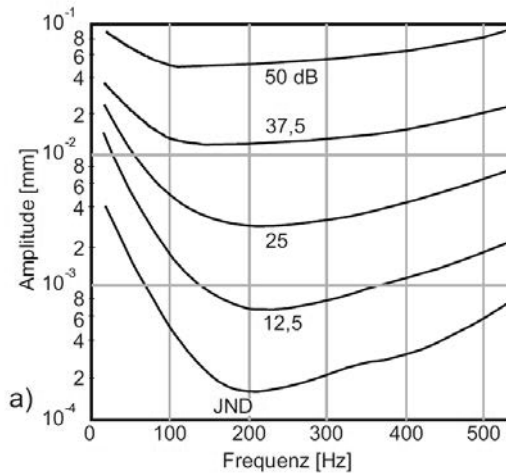


Abbildung 12. Schwellwert der Wahrnehmung von Schwingungen nach HUGONY 1935 [7]

Ob Störungen in diesem Frequenzbereich auftreten, lässt sich theoretisch schwer abschätzen. Auch ob eine dauerhafte Bestromung des Motors in dieser geplanten Größenordnung zulässig ist, muss am realen Aufbau bzw. im Einsatz getestet werden.

5 Aktueller Entwicklungsstand und Ausblick

Für die Validierung der zuvor beschriebenen Simulationsergebnisse und erste haptische Tests ist es nötig, den Motor mit einer Ansteuerungs- und Leistungselektronik auszustatten. Die Kommutierung der PMSM muss, anders als beim Gleichstrommotor, elektronisch erfolgen. Kaufösungen von Ansteuer-elektroniken bieten nicht die gewünschte Funktionsvielfalt, welche im Bereich der Haptik aufgrund der menschlichen sensorischen Fähigkeiten nötig sind. Eine präzise Momentenregelung mit einer genauen Lageerkennung sowie Zugriff auf alle Reglerparameter sind notwendig. Um Bauraum- und Gewichtsvorteile des Gesamtkonzepts aufzuzeigen, wurde bei der Entwicklung der Hardware zusätzlich auf Miniaturisierung gesetzt.

Die Abbildung 13 zeigt das Konzept der entwickelten Elektronik. Überwacht und geregelt wird das System vom einem 32Bit Mikrocontroller. Für die Lageerkennung wird ein magnetischer 14Bit Absolut-Encoder (AS5048A) eingesetzt. Die Leistungsstufe ist ein 3-Phasen-Motortreiber (B6-Brücke) mit integrierter Treiberstufe und Ladungspumpe. Für die spätere Ansteuerung sind CAN- und USART-Anschlüsse herausgeführt.

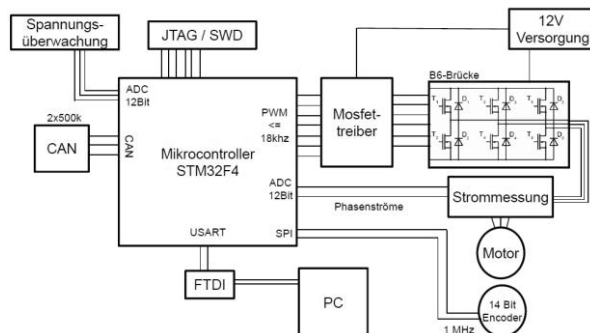


Abbildung 13. Konzipierung der Elektronik

Das Drehfeld des Motors wird über eine feldorientierte Regelung (Field Oriented Control, FOC) in dem geschlossenen Regelkreis geregelt. So läuft der Motor immer bei optimalem Drehmoment und arbeitet energieeffizient. In Abbildung 14 wird ein Überblick über die hierfür verwendeten Transformationen gegeben.

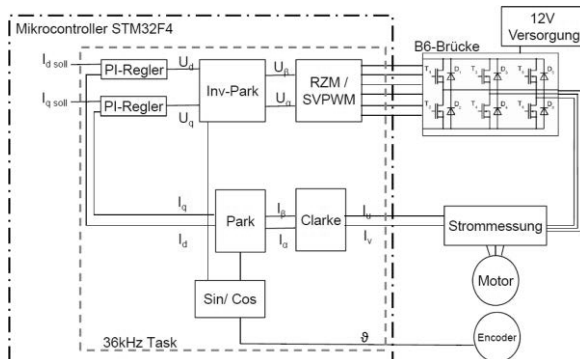


Abbildung 14. Umsetzung der feldorientierten Regelung

Der Stromregeltask läuft Interrupt gesteuert mit 36kHz und liegt somit deutlich außerhalb dem vom Menschen wahrnehmbaren Frequenzbereich.

Die gesamte Elektronik findet auf einer runden Platine mit einem Durchmesser von 45mm Platz und passt so direkt unter den Motor (Abbildung 15). Auf dem Wellenende (nicht sichtbar) ist zur Positionsmessung ein diametral-magnetisierter Magnet befestigt, welcher mittig über einem 14Bit Hall Encoder dreht.

Die Low-Level-Programmierung der Clark-Park-FOC-Regelung ist abgeschlossen. Die Haptikregelung sowie die Validierung der Simulationsergebnisse stehen noch aus. Erste Tests bei der Inbetriebnahme sehen sehr vielversprechend aus.



Abbildung 15. Motor mit verbauter Elektronik im Größenvergleich

Zusammenfassend kann gesagt werden, dass diese Art der Motoren eine Lücke zwischen den klassischen DC-Motoren und den Schrittmotoren schließen. Es ist zu hoffen, dass die Hersteller sich über das Potenzial ihrer Antriebe bewusst und diese auch mit guten, belastbaren technischen Daten hinterlegen werden. Es ist vorstellbar, dass diese Miniatur-High-Torque Motoren für eine Vielzahl anderen Antriebsaufgaben eingesetzt werden können.

6 Referenzen

- [1] Bubb H., Mauter G. Wild J., Reisinger J.: Haptical feeling of rotary switches, Proceedings of Euro haptics2006 (EHC2006), p49-55, 3-6.7., Paris, France, 2006
- [2] Jandura L., Srinivasan M.: Experiments on Human Performance in Torque Institute of Technology, Cambridge, 1994
- [3] Anguluelov N.: Haptische und akustische Kenngrößen zur Optimierung der Wahrnehmen von Schaltern und Bedienfeldern für den Kfz-Innenraum, Dissertation 2009, technische Universität Dresden
- [4] C.Clees Multikopter.cc <http://www.multikopter.cc/teile-zubehoer/sonstiges-1/brushless-gimbal-fuer-kameras-bis-1500g.php> (22.01.2014)
- [5] iFlight Model Limited <http://www.iflight-rc.com/product/iPower-Brushless-Motor-iPower-GBM-Series-iPower-GBM4108-80T.html> (22.01.2014)
- [6] Online Wicklungsrechner <http://i.caendle.de/dev/test2/> (22.01.2014)
- [7] Kern T.A., Matysek M., Merkel O., Rausch J.: Entwicklung haptischer Geräte, Springer Verlag, 2009

Simulation of hybrid Railway Vehicles and Comparison of electrical Energy Storage Systems

Dipl.-Ing. (FH) Sebastian Schmid

atr – Research and Development, Voith Turbo GmbH & Co. KG, Heidenheim

zzsebastian.zzschmid@Voith.com

A python-based software tool is developed to compare different hybrid propulsion systems for railway vehicles. The tool allows for an automated parameterization of multiphysical 1D railway vehicle simulation models. Three different power transmission technologies are available as models (diesel-hydroneumatic, diesel-hydrodynamic and diesel-electric) and can be linked with three type of energy storage systems based on batteries, double-layer capacitors or flywheel technology. Two types of service profiles can be loaded to the simulation model (suburban and regional) and an optimization routine secures that the vehicle simulation models drive according to pre-given timetables. With the help of the software tool, the fuel saving potential of a hybrid two-coach Siemens Desiro Classic on a regional service profile and a hybrid two-coach Bom Sinal Mobile 2 on a suburban service profile is investigated. Detailed LCC analyses show the savings which can be achieved with each type of storage system for the hybrid Bom Sinal Mobile 2 rail vehicle with a hydroneumatic DIWA transmission.

1 Introduction

In the last ten years, starting with the development of the Hitachi/JR East New Energy Hybrid Train [1], the current trend of hybrid and electric vehicles in the automotive sector was expanded to the railway sector and railway suppliers present promising figures of up to 25 % in terms of possible fuel reductions for hybridized diesel-driven railway vehicles [2,3,4]. On the other hand, the automotive sector still has to fight against customer concerns in terms of safety, costs and lifetime of hybrid components.

For this reason, Voith Turbo initiated a collaborative research project with the University of Bradford, England, to investigate on the potential of modern electrical energy storage systems (batteries, double-layer capacitors and flywheels) and how they can be used to hybridize railway vehicle drive systems.

In a first step, in section 2 the main summary of a literature review on the different available electrical energy storage systems (ESS) and their current state of art is portrayed. Section 2 also highlights the simulation parameters used for each ESS. In a second step, in section 3 detailed simulation models of the different subsystems of a train vehicle are described and developed with the help of 1D simulation software. These models offer the possibility to join them to a model of a complete train vehicle whose journey on specific routes can then be simulated with the appropriate timetable and station stops. For the investigation two typical train vehicles, currently used on

routes in Germany and Europe, the Siemens Desiro Classic and Bom Sinal Mobile, both in a 2-car configuration, are equipped with different electrical energy storage systems. With the help of a Python-based simulation tool, described in section 4, the potential in terms of fuel reduction and the resulting life-cycle costs (LCC) are analyzed for both vehicles on different routes. The simulation results and LCC analysis are found in sections 6 and 7. The paper concludes with a discussion of the simulation results and a recommendation for further research in terms of use of electrical energy storage systems for Railway Vehicles (see section 8).

2 State of the Art of Electrical Energy Storage Systems

Lithium-Ion (Li-Ion) Battery

Due to the achievable power and energy densities, Lithium-Ion (Li-Ion) batteries are regarded as the key technology for current and upcoming hybrid and electric vehicles in all market segments. Compared to Ni-MH batteries, almost twice the gravimetric energy density is available, what would bring an electric vehicle twice the range, keeping the battery weight constant. The following Table 1 highlights the main characteristics of modern Li-Ion Cells. For the simulations in section 6, Nano NMC cell data by Akasol [5] is used. A 200s1p circuitry per powerpack of the 46Ah cells is used, which results in an overall energy content of 34 kWh per used powerpack.

Characteristics	Unit	State of the art
Nominal Voltage	[V]	2.3 - 3.8
Gravimetric/ volumetric energy density	[Wh/kg] [Wh/l]	150 250
Gravimetric/ volumetric power density	[W/kg] [W/l]	3000 4200
C-Rate - Continuous / Peak 10s	[-]	5.8 12.2
Cycle Life @ 80 % DOD	[-]	3000
Cycle Efficiency @ 80 % DOD	[%]	96

Table 1. State of the Art Characteristics of Li-Ion Battery Cells [6]

Double-Layer Capacitors (DLC)

Over the last years, double-layer capacitors (DLC), also known under the brand names UltraCap[®], SuperCap[®] or BoostCap[®], are gaining importance in terms of using them as energy storage in hybrid vehicles. Because of the relatively low internal resistance, higher power densities can be realized compared to Li-Ion batteries. For this reason, as well as their extremely high cyclic lifetime of up to 1 million cycles, they are often used for short-time storage of brake energy in train vehicles, where the charging c-rate of a Li-Ion battery would not be sufficient enough. One example is the Sitras[®] MES (mobile energy storage unit) system by Siemens used in LRV applications for a reduction in energy consumption (see Figure 1).

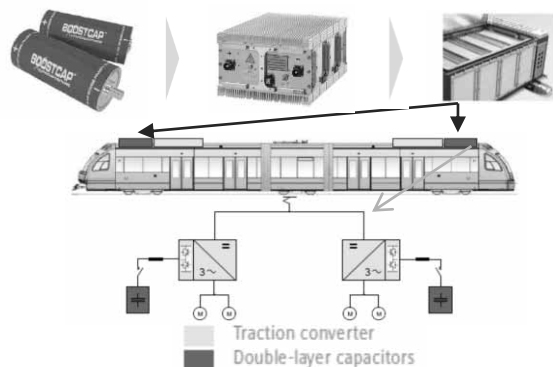


Figure 1. Overview of Siemens Sitras[®] MES (mobile energy storage unit) system for light rail vehicles [7]

Figure 2 is showing a Ragone diagram for different energy storage technologies. The gravimetric energy density and the gravimetric power density are plotted

against each other with logarithmic scales. By dividing both values, one obtains straight lines showing the times needed for a complete discharge of the energy storage. It shows the huge advantage as well as disadvantage of DLCs. On the one hand a high power output can be achieved, but on the other hand only for a short period of time in the range of 10 – 15 seconds.

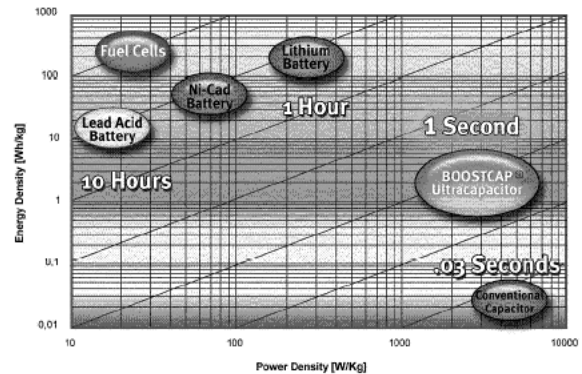


Figure 2. Ragone Diagram for different Energy Storage Technologies [8]

One of the best known manufacturers of DLCs under the brand name BoostCap[®] for “Heavy Duty” applications (e.g. buses and train vehicles) is Maxwell Technologies. A specially designed DLC module BMOD0063 with a rated capacitance of 63 F and a usable energy content of 137 Wh is available (see Figure 3).

PRODUCT SPECIFICATIONS

ELECTRICAL	BMOD0063 P125 B04/B08
Rated Capacitance ¹	63 F
Minimum Capacitance, initial ¹	63 F
Maximum ESR ω_c initial ¹	18 mΩ
Rated Voltage	125 V
Absolute Maximum Voltage ^{1,5}	136 V
Maximum Continuous Current ($\Delta T = 15^\circ C$) ²	140 A _{RMS}
Maximum Continuous Current ($\Delta T = 40^\circ C$) ²	240 A _{RMS}
Maximum Peak Current, 1 second (non repetitive) ³	1,800 A
Leakage Current, maximum (VMS 2.0) ⁴	10 mA
Maximum Series Voltage	1,500 V
POWER & ENERGY	
Usable Specific Power, P_d ⁵	1,700 W/kg
Impedance Match Specific Power, P_{max} ⁶	3,600 W/kg
Specific Energy, E_{max} ⁷	2.3 Wh/kg
Stored Energy ⁸	136.7 Wh

Figure 3. Characteristics of Maxwell Technologies BoostCap[®] Module BMOD0063 [10]

The nominal voltage is 125 V_{DC}. With a very high temperature range of - 40 up to + 65 °C and a cycle life of 1,000,000 cycles it is ideally suited for mobile applications and 6 of them in series will therefore be used per powerpack for the hybrid railway vehicle

simulations in this paper. One example for its use is the hybrid bus Solaris Urbino 18 with a DIWAHybrid gearbox by Voith [9].

Flywheels

In a flywheel energy is stored by accelerating a rotor to a high rotational speed of up to 60.000 rpm [11]. By decelerating the rotor with the help of an electric generator or a hydraulic pump, the rotational kinetic energy can be transformed to either electric energy or hydraulic pressure. Modern flywheel systems use rotors made up of carbon fibres, two exemplary flywheels by Porsche and Ricardo are shown in Figure 4. In order to increase the efficiency of flywheel energy storage systems, modern developments try to reduce friction losses as far as possible by evacuating the chamber in which the rotor is spinning. Another means is using almost frictionless magnetic bearings which require a sophisticated control system.



Figure 4. Examples for Flywheels used by Porsche (left) and Ricardo (right) [11,12]

In the year 2001 an Alstom Coradia LIREX[®] test carrier diesel-electric multiple unit was equipped with a flywheel energy recuperation system in a cooperation project between the German vehicle operator DB (Deutsche Bahn) and Alstom LHB [13]. The carbon fibre and epoxy resin flywheel was designed by the Scientific Technical Centre in Rosslau, Germany. It had a diameter of 700 mm and a maximum speed of 25.000 rpm. An overview of the technical data of the flywheel is shown in Table 2. A scaled down version of this system with 1.5 kWh energy content will be used in the simulations in section 6.

Manufacturer	Scientific Technical Centre, Rosslau
Energy Content	6 kWh
Maximum Power:	350 kW
Efficiency including Frequency Converter	> 90 %
Idling Losses	2.5 - 7 kW

Voltage [V]	550 V – 750 V
Rotor Material	Carbon Fibre / Epoxy Resin
Rotor Diameter	700 mm
Maximum Speed	25000 r/min
Minimum Speed	12500 r/min
Type of Motor	Synchronous motor
Life	20 years
Working Temperature Range	-40°C to 60°C
ESS Dimensions	1900 x 1625 x 1080 mm
Mass including carrying frame	1300 kg

Table 2. Technical Data of the Alstom Coradia LIREX[®] Storage Flywheel [13]

3 Train Vehicle Simulation Models

Figure 5 shows the schematic representation of a train vehicle and the different subsystems which have to be available as submodels within a simulation tool in order to perform a drive cycle simulation. The different subsystems are modelled with the multi-domain 1D simulation tool LMS Imagine.Lab AMESim.

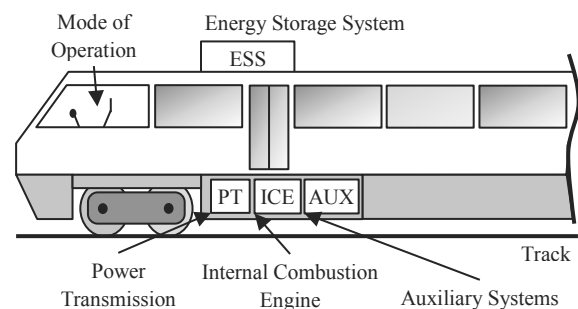


Figure 5. Overview of the different Train Vehicle Simulation Subsystems

Because of disclosure agreements, only the submodel used for the hydromechanic Voith DIWA transmission will be explained hereafter. For information on the other submodels, please contact the author of this paper.

DIWA Transmission

DIWA is an abbreviation for the German word “Differentialwandler” and is a transmission by Voith Turbo and used in busses and train vehicles. Figure 6 shows the composition of the hydromechanic trans-

mission VOITH DIWA D 864.5 used in railway vehicles. The AMESim simulation model of the DIWA transmission is a representation of the real system except for the counter-rotating torque converter which has been modeled with the help of components from the signal library (see Figure 7). The transmission has four speeds. In the first gear a power split principle is used, at low speeds all the required torque is supplied from the converter. When the vehicle speed increases the torque supplied from the mechanical gears is increased and the torque output from the converter is reduced..

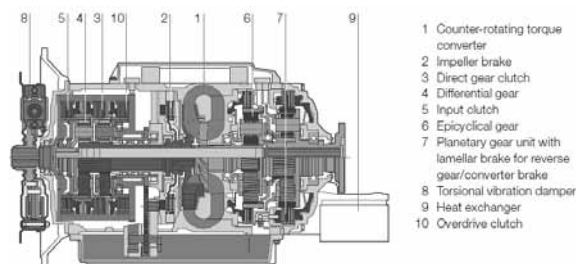


Figure 6. Voith DIWA D 864.5 [14]

In order to parameterize the simulation model of the DIWA transmission the following global parameters were defined and can be adjusted to the given railway vehicle (see Table 3).

Parameter	Unit	Description
maxM1	[Nm]	Max. torque first gear
maxM2	[Nm]	Max. torque second gear
maxM3	[Nm]	Max. torque third gear
maxM4	[Nm]	Max. torque fourth gear
maxP1	[kW]	Max. power first gear
maxP2	[kW]	Max. power second gear
maxP3	[kW]	Max. power third gear
maxP4	[kW]	Max. power fourth gear
SG12	[km/h]	Shifting speed 1-2 gear
SG23	[km/h]	Shifting speed 2-3 gear
SG34	[km/h]	Shifting speed 3-4 gear
lambda_nue	[-,-]	converter λ vs. v table
Mue_nue	[-,-]	converter μ vs. v table

Table 3. DIWA Transmission Parameters

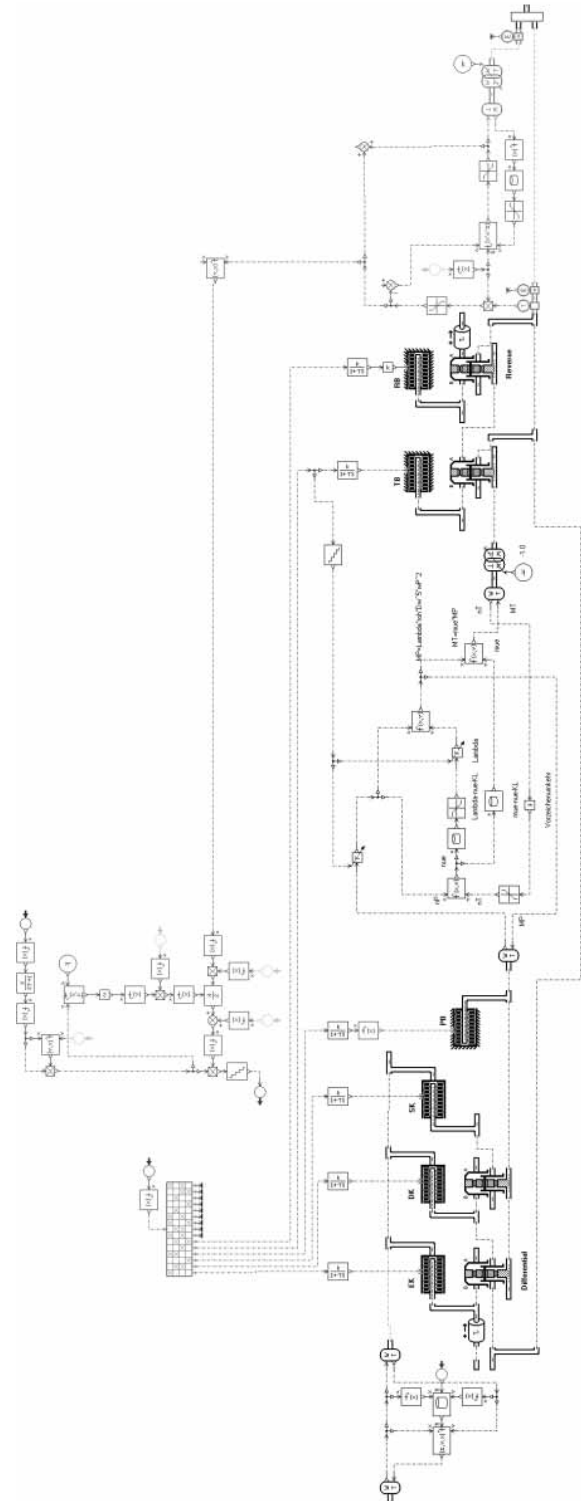
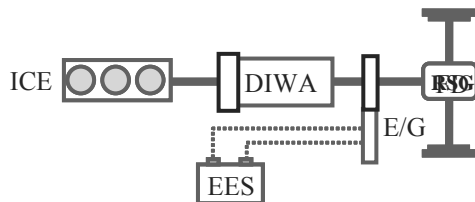


Figure 7. AMESim Simulation Model of Voith DIWA Transmission

As already mentioned above, in the first gear the DIWA transmission uses a torque converter in combination with a mechanical gear to transmit power. In

terms of hybridization of the gearbox, this means that the necessary electric motor is connected to the secondary side of the gearbox. Otherwise, no brake energy could be recuperated in the first gear because the torque converter can only transmit power in one direction from primary to secondary side of the gearbox. The following Figure 8 highlights the type of parallel hybrid configuration used for railway vehicles with a DIWA transmission in this work. This has the advantage that during energy supply and brake energy recuperation the power of the ESS doesn't have to be transmitted via the gearbox, on the other hand high torques are required by the electric motor because the ratio of the transmission is not used.



ICE = Internal Combustion Engine
EES = Electrical Energy Storage
FD = Final Drive
E/G = E-Motor / Generator Unit

Figure 8. Type of parallel hybrid Configuration used for DIWA Transmission

4 Voith Turbo Rail Simulation Tool

For a user-friendly pre- and post-processing and an evaluation of the AMESim train vehicle simulation models, as well as a comparison of different energy storage systems according to fuel savings and LCC costs, a software tool has been programmed. It is based on the object-oriented programming language Python and can be run separately from the LMS Imagine.Lab AMESim simulation software.

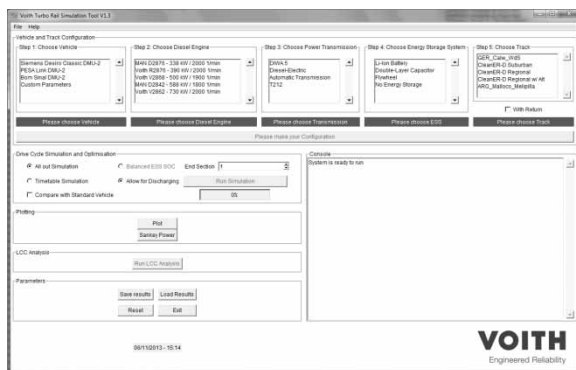


Figure 9. Main GUI of Voith Turbo Rail Simulation Tool

Figure 9 shows a screenshot of the main GUI of the Python based simulation tool named “Voith Turbo Rail Simulation Tool” and the different functions of the user interface. They can be divided into the main sections “vehicle and track configuration”, “simulation options”, “LCC analysis”, “plotting options”, “console” and “save/load results” of which the first three will be explained in the following sub sections 4.1 to 4.3 The “console” is showing the most important results during a simulation run and the current progress. In the “save/load results” the user can store the simulation results after a run or load results from old runs. “Plotting Options” let the user display simulation results.

4.1 Vehicle and Track Configuration

In order to start a new simulation analysis the user has to make a vehicle and track selection in the top area of the simulation tool GUI. Figure 10 shows the different choices available in terms of vehicle, diesel engine, power transmission, energy storage system and track.



Figure 10. Vehicle, Diesel Engine, Power Transmission, ESS and Track Selection

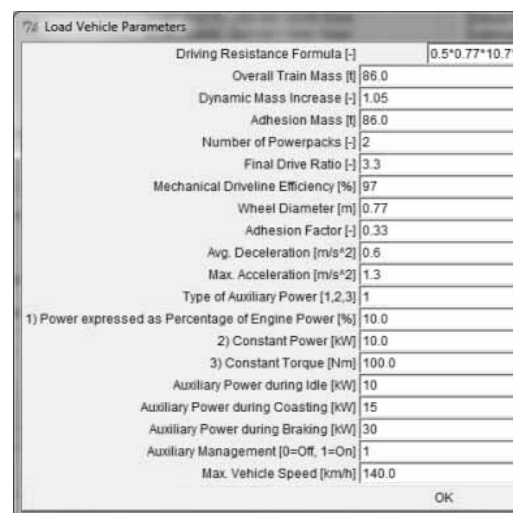


Figure 11. Vehicle Parameters Pop-Up Window

In a first step, the railway vehicle has to be chosen. Therefore three different predefined vehicles with the corresponding parameters are given, when selecting one of the vehicles a pop-up window appears which gives the user the opportunity to adapt the vehicle

parameterization. The user can also define the parameters for a new train vehicle when selecting the option “custom parameters”. Figure 11 depicts the vehicle parameters pop up window.

In a second step a diesel engine has to be chosen, 3 engine sizes from 6 cylinders in line up to a V12 engine with 5 different power outputs ranging from 338 kW up to 730 kW are available. The third step is the power transmission selection. Three different types are available: DIWA transmission, automatic transmission and diesel-electric transmission. Upon choosing one type, a pop-up appears where the different parameters (e.g. maximum input torques and shifting speeds) can be adapted to the given vehicle. In terms of ESS, the technologies li-ion battery, double-layer capacitor and flywheel as described in section 2 are available. Similar to the engine and vehicle selection, a pop up appears after clicking on one of the ESS's (see Figure 12 for li-ion battery example). Here all the ESS specific characteristics can be adapted to the given use case.

State of Charge at Start [%]	50.0
Number of parallel Branches [-]	1
Number of Cells in series [-]	200
Capacity of one Cell [Ah]	46.0
Specific Energy of ESS [Wh/kg]	100
Cell maximum continuous charge current [A]	230.0
Cell maximum continuous discharge current [A]	265.0
Cell maximum pulse charge current [A]	367.1
Cell maximum pulse discharge current [A]	561
Pulse duration [sec]	10
Cell minimum voltage [V]	3.4926
Cell maximum voltage [V]	4.1102
Boost Factor [0-1]	0.15
Open Circuit Voltage Table [V(%)]	Select OCV[SOC] File
Internal Resistance Table [Ri(%)]	Select Ri[SOC] File
Efficiency of Converter [%]	99.0
OK	

Figure 12. Li-Ion Battery Parameters Pop-Up Window

In a final step, the track on which the train vehicle is going to operate has to be chosen. Three different choices can be made. The track “CleanER-D Suburban” corresponds to a typical suburban service profile with a track length of 40 km, 12 station stops and a maximum speed of 120 km/h, the track “CleanER-D Regional” corresponds to a regional service profile with a track length of 70 km, 14 station stops and a maximum speed of 140 km/h.

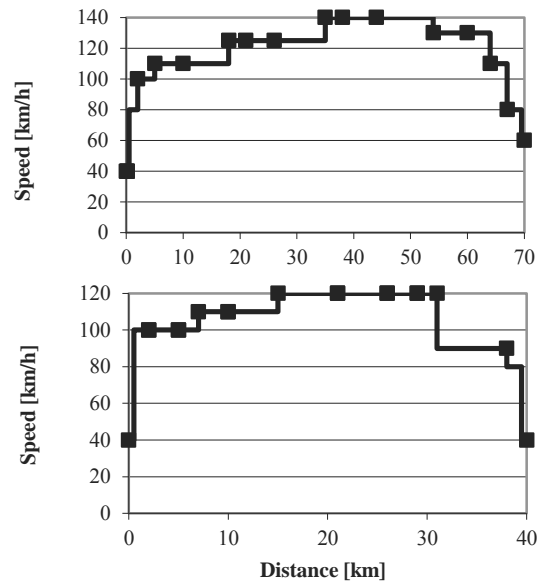


Figure 13. Speed limits and Station Stops for CleanER-D Regional and Suburban Service Profile [15]

Both mission profiles are on a flat track without any gradients. The third profile is named “CleanER-D Regional w/ Alt” and is the same as the “CleanER-D Regional” but has an additional altitude profile. All the tracks were defined within the CleanER-D EU project and their details, including the travel times between the station stops, can be found in [15] (see Figure 13). Figure 14 shows the altitude profile for the CleanER-D regional service profile. The final station is elevated 70 meters above the start station and the maximum height of the track is 140 meters.

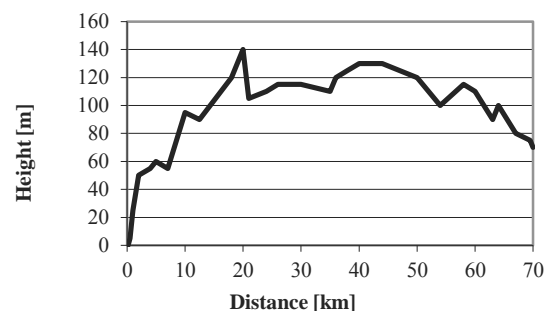


Figure 14. Altitude Profile for the CleanER-D Track Regional Service Profile [15]

The user of the tool can choose if the return journey on the selected track shall be considered in the simulation run. If this is the case, the vehicle makes a 20 minute station stop in the final station before returning to the start station. The time corresponds to a typical train vehicle operator’s break based on experiences by Voith Turbo.

4.2 Simulation Options

After the vehicle and track selection, the options for the simulation run have to be chosen. Therefore different options are available (see Figure 15).

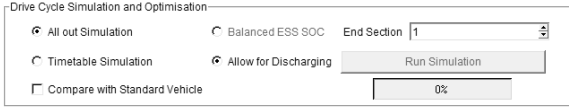


Figure 15. Drive Cycle Simulation and Optimization Options

The first two options are concerning the driving style of the vehicle: All out or timetable. All out is meaning that the vehicle operator travels on the track as fast as possible, in this mode no coasting phases are used and the fuel consumption is highest. In the other mode timetable simulation, the virtual driver sticks to the given timetable of the selected service profile. By changing the length of the coasting phases in front of reference speed changes or before braking to a station stop, the vehicle time is adopted. Therefore, as a first step a simulation is conducted where the vehicle drives as fast as possible on the track and the driving times for the different track sections (incl. return journey if selected) are stored in a vector called *timeZeroVect*. After that, the driving times for the track sections are optimized one after another. Once the optimization reaches a time deviation of 1 second compared to the timetable time, which is stored in the vector *Time*, the next section is optimized and the timetable time of the section is adopted in regard to the last time deviation.

For the optimization a linear interpolation approach is used. Therefore the driving time for the fastest driving style, where *coasting_1* and *coasting_2* are set to 1, is compared with the driving time where *coasting_1* and *coasting_2* are set to 5. After every linear interpolation the new factors are stored in a vector *coast1vect*. The new factors *coasting_1* and *coasting_2* for the optimization step $n+1$ are then calculated with the following formulae 1 and 2.

$$coasting_1 = \frac{coast1vect[n-1] \cdot (timeZeroVect[n] - Time[k]) - coast1vect[n] \cdot (timeZeroVect[n-1] - Time[k])}{timeZeroVect[n] - self.timeZeroVect[n-1]} \quad (1)$$

$$coasting_1 = coasting_2 \quad (2)$$

With the other two radio buttons named “balanced ESS SOC” and “Allow for Discharging” the behavior of the Energy Management is determined. When the option “balanced ESS SOC” is chosen, the battery ESS is optimized in order to achieve the same SOC at

the end as at the start station. This means that no external charging is necessary at the final station. Therefore the same linear interpolation as described before for the optimization of the coasting factors is used, with SOC at the end of the simulation being the target variable instead of the driving time. But this is only valid for a battery ESS, for the DLC and fly-wheel it is assumed that the ESS is empty at the start and therefore a balanced ESS SOC is not necessary. When “Allow for Discharging” is chosen, the factor boost remains unchanged and the battery can have a lower SOC at the final station stop compared to the start station. Another option is the check button “Compare with Standard Vehicle”. Upon selection the hybrid vehicle will be compared with the non-hybrid version. In order to execute a LCC analysis after the run, this option has to be selected, because otherwise no information about the fuel benefits is available. In order to check if the vehicle configuration delivers satisfactory results without having to simulate a whole service profile, the end section of the track can also be defined before a simulation run. When e.g. *end section* is set to 1, only the route from the start station to the first station stop will be simulated. The user can then have a look at the simulation results and see if the vehicle behaves as expected in terms of e.g. acceleration, fuel consumption or top speed.

4.3 LCC Analysis

A LCC (life cycle costs) analysis is a very important feature to investigate the return on investment time of a hybridisation of a railway vehicle. In order to conduct the analysis, in a first step a simulation run with a comparison of the standard vehicle has to be done.

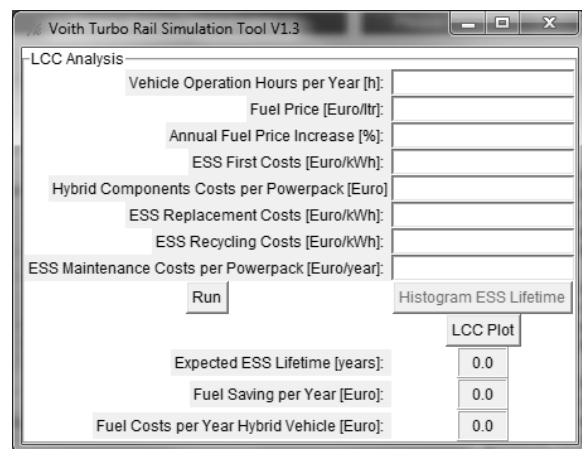


Figure 16. LCC Analysis User Input Pop-Up Window

After the simulation run, a pop-up window for the LCC analysis can be opened. The user has to provide several input data which will be described hereafter:

- *Vehicle operation hours per year [h]*: The cumulated time the vehicle operates in a whole year. For railway vehicles the value is usually in the range of 2500 to 3500 hours.
- *Fuel Price [€/ltr]*: The current diesel fuel price.
- *Annual fuel price increase [%]*: Because of the limitation of oil resources, the fuel price will increase in the coming years. This factor describes the annual increase.
- *ESS first costs [€/kWh]*: The price of the energy storage system without the other hybrid components (e.g. electric motor and power electronics)
- *Hybrid components costs per PowerPack [€]*: The price of the hybrid components per PowerPack (e.g. for electric motors and power electronics)
- *ESS replacement costs [€/kWh]*: The LCC analysis is calculating the expected lifetime of the energy storage system. Since the price for ESS is expected to drop in the coming years, the user can insert a new price for the replacement.
- *ESS recycling costs [€/kWh]*: In case a country is asking for a charge for recycling of an ESS, this value can be entered here.
- *ESS maintenance costs per PowerPack [€/year]*: The price for the replacement of parts or other system checks can be entered.

After the user has entered the above mentioned data, the LCC analysis is conducted by activating the “Run” button and the field for the expected lifetime of the ESS, the fuel savings per year in [€] and the overall fuel costs per year for the hybrid vehicle in [€] are filled out.

The calculation of the ESS lifetime T_{ESS} is depending on the type of technology used:

DLC

One of the main advantages of DLCs are their extremely high cyclic lifetime. For the ESS lifetime, a total number of 1,000,000 cycles is used according to [10]. The cycles in the simulation are summarized independent on the DOD. With the journey time (sec) for one simulation t_{dc} , the number of cycles during this journey N_{C_used} and the vehicle operation hours per year t_{Op_h} , the DLC ESS lifetime T_{ESS_DLC} in [a] is calculated with the following formula.

$$T_{ESS_DLC} = \frac{1000000}{N_{C_used}} \cdot \frac{t_{dc}}{3600} \cdot \frac{1}{t_{Op_h}} \quad (3)$$

Flywheel

The lifetime of a flywheel is very hard to estimate, since it mainly consists of mechanical parts and depends on the material, bearings and other components used in each individual case. In order to have a first estimation, the same formula as for the DLC is used, with 1,000,000 possible cycles.

$$T_{ESS_FW} = \frac{1000000}{N_{C_used}} \cdot \frac{t_{dc}}{3600} \cdot \frac{1}{t_{Op_h}} \quad (4)$$

Battery

The chemical deterioration processes of a battery have many influences such as temperature, charge and discharge currents and calendaric ageing. After discussions with battery experts, it was decided to use a rainfall-counting algorithm for the ageing of the battery. The origin of this approach lays in fatigue analysis and is defined by Standard ASTM E1049. A python script can be found in [16] and was integrated in the simulation tool. The rainflow-counting algorithm calculates the number of cycle and half cycles depending on pregiven DOD ranges for the battery SOC curve. In the following Figure 17 an example for a SOC curve from the simulation is shown.

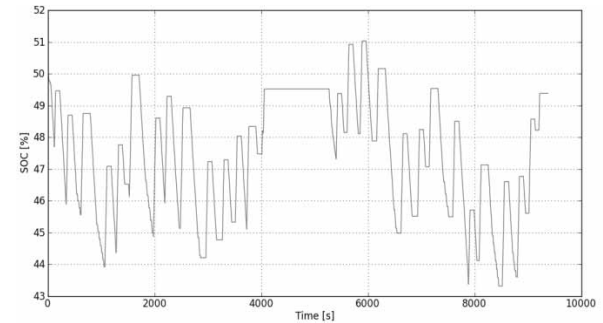


Figure 17. Example for a SOC Curve of a Battery during a Simulation Run

When the algorithm is applied to the curve, the following results are obtained (see Table 4).

Range [DOD]	N_{C_used}	$N_{C_possible}$	$N_{C_used} / N_{C_possible}$
90.00 to 100.00	0	2168	0
80.00 to 90.00	0	2583	0
70.00 to 80.00	0	3146	0
60.00 to 70.00	0	3942	0

50.00 to 60.00	0	5128	0
40.00 to 50.00	0	7034	0
30.00 to 40.00	0	10450	0
20.00 to 30.00	0	17753	0
15.00 to 20.00	0	31135	0
10.00 to 15.00	0	52893	0
5.00 to 10.00	0.5	118252	4.2^{-6}
1.00 to 5.00	26	500684	5.2^{-5}

Table 4. Rainflow Counting Algorithm Results for Battery SOC Curve Example

This means that there are 26 cycles in the DOD range 1 to 5 % and one half cycle in the DOD range 5 to 10 %. This half cycle corresponds to a discharge phase starting at 50 % SOC and can be seen in Figure 17 at around time 6500 seconds. Cycles with DOD in the range 0 – 1 % are not taken into account for the battery lifetime calculation.

The next information needed to calculate the battery lifetime is the number of possible cycles for each of the DOD ranges. Therefore the mean depth of discharge DOD_{mean} for each range j is taken, e.g. 3 % for the range 1 to 5 % and inserted into the following formula, which is used to calculate the maximum number of cycles $N_{C_possible}$ for a given DOD until the EOL criteria of 80 % residual capacity is achieved.

$$N_{C_possible_j} = a \cdot DOD_{mean_j}^b \quad (5)$$

Where a and b are variables which greatly influence the characteristic curve and are usually determined experimentally by battery manufacturers. In this work, after discussions with ESS experts at Voith, a conservative approach has been used for both values with $a = 2000$ and $b = -1.575$ according to [17]. This means that 2000 full charge/discharge cycle are feasible with the battery until reaching the EOL criteria. The number of possible cycles for each DOD_{mean} is also shown in Table 4. With this information, the overall lifetime of the battery T_{ESS_bat} in [years] can be calculated with the following formula.

$$T_{ESSBat} = \left[\sum_{j=1}^n \left(\frac{N_{C_used_j}}{N_{C_possible_j}} \right) \right]^{-1} \cdot \frac{t_{dc}}{3600} \cdot \frac{1}{t_{oph}} \quad (6)$$

In order to give a graphical representation of the LCC costs, the user can generate a bar chart showing the costs of the hybrid components. This gives the user

an idea about the investment costs and amortisation time for each individual use scenario.

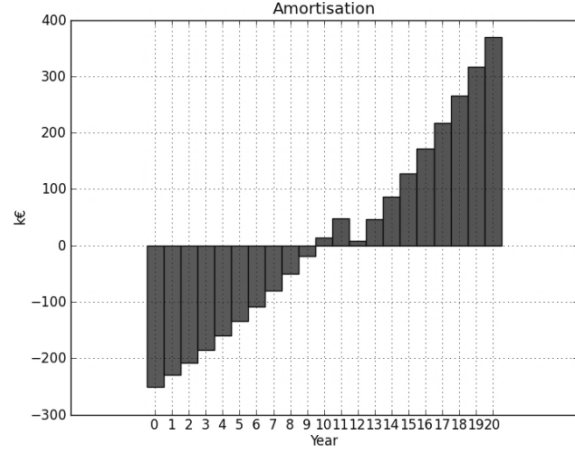


Figure 18. Bar Chart Example to represent LCC Analysis

Figure 18 shows an example of such a bar chart for a battery ESS. The drop after year 12 in the example is due to the replacement of the battery.

5 Railway Vehicles under Investigation

Two different vehicles will be analysed in terms of their potential for hybridization. The first one is the Siemens Desiro Classic, also known as DB class 642 in a two-car configuration and the second one a Bom Sinal Mobile 2 mainly used in South America (e.g. Brazil). The followings sections 5.1 and 5.2 describe the main vehicle parameters used for the simulations.

5.1 Siemens Desiro Classic Vehicle Parameters

Around 320 of these vehicles (see Figure 19) are currently in service in Germany.



Figure 19. Siemens Desiro Classic

The considered configuration will be powered by two PowerPacks, both using a 360 kW diesel engine with a six-speed hydromechanic power transmission by ZF. Table 5 shows the main parameters of the vehicle used for the simulation. The vehicle has a top speed

of around 120 km/h and is mainly used on suburban and regional service profiles.

Parameter	Unit	Value
Train mass	[t]	86.0
Adhesion mass	[t]	57.0
Number of powerpacks	[-}	2
Max. speed	[km/h]	124.8
Max. engine power	[kW]	390 kW

Table 5. Siemens Desiro Classic Vehicle Parameters

5.2 Bom Sinal Mobile 2 Vehicle Parameters

The second train vehicle under investigation is a two-coach Bom Sinal Mobile 2 (see Figure 20). It is a two-coach vehicle typically used in South America for public transport between smaller cities and suburban areas.



Figure 20. Bom Sinal Mobile 2

The vehicle is equipped with two PowerPacks with a DIWA transmission and a 338 kW MAN 6-cylinder inline Diesel engine. Since the vehicle is used on tracks with steep gradients with a maximum passenger capacity of 358 people, a high tractive effort is required. Table 6 shows the main parameters of the vehicle used for the simulation.

Parameter	Unit	Value
Train mass	[t]	98.0
Adhesion mass	[t]	40.0
Number of powerpacks	[-}	2
Max. speed	[km/h]	90.5
Max. engine power	[kW]	338 kW

Table 6. Vehicle Parameters Bom Sinal Mobile 2

6 Simulation Results

The simulation models of the two train vehicles are equipped with the different energy storage technologies (li-ion battery, DLC and flywheel) in order to obtain potential fuel reduction predictions. For the two-coach Siemens Desiro Classic, the CleanER-D regional service profile will be used in the simulative analysis, since it is commonly used on regional tracks in Germany. The simulations are conducted with and without altitude to investigate the effect on fuel savings. The two-coach Bom Sinal Mobile 2 is simulated on the CleanER-D suburban service profile, since with its low diesel engine power it is mainly used for commuter services in South America.

When a battery ESS is used for the hybrid railway vehicle simulation models, for each use case in sections 6.1 and 6.2 the boost factor is optimized to achieve a balanced SOC of the battery at the end of the simulation run.

6.1 Results for two-coach Siemens Desiro

CleanER-D Regional Service Profile

For the 140km long regional service profile (incl. return journey) the 86 ton 2-coach DMU Siemens Desiro Classic with all three different types of ESS technologies, fuel savings are achieved. With a battery ESS, the consumption is reduced by 7.3 % to 136.4 litres. The DLC ESS ends up with a reduction of 5 % and the Flywheel ESS with 7.9 % (see Table 7).

Standard Vehicle	[%]	100.0
Hybrid Vehicle with Battery ESS	[%]	92.7
Hybrid Vehicle with DLC ESS	[%]	95.0
Hybrid Vehicle with Flywheel ESS	[%]	92.1

Table 7. Fuel Consumption of two-coach Hybrid Siemens Desiro Classic DMU-2 on CleanER-D Regional Service Profile

CleanER-D Regional Service Profile with Altitude

The simulation results in this section correspond to the hybrid Siemens Desiro Classic DMU-2 on the regional service profile with an altitude profile ac-

cording to Figure 14. Compared to the standard vehicle, all three ESS technologies achieve fuel savings when being equipped to the vehicle (see Table 8). The hybrid vehicle with battery ESS achieves a reduction in fuel consumption of 8 %, the DLC ESS 4.2 % and the flywheel ESS 7.8 %.

Standard Vehicle	[%]	100.0
Hybrid Vehicle with Battery ESS	[%]	92.0
Hybrid Vehicle with DLC ESS	[%]	95.8
Hybrid Vehicle with Flywheel ESS	[%]	92.2

Table 8. Fuel Consumption of two-coach Hybrid Siemens Desiro Classic DMU-2 on CleanER-D Regional Service Profile with Altitude

6.2 Results for two-coach Bom Sinal Mobile 2

The standard two-coach Bom Sinal DMU-2 without hybrid components and without an on-board electrical storage system needs 68.2 liters (see Table 9) of diesel fuel on the 80 km long suburban service profile (including return journey). This equals an average fuel consumption of 85.25 l/100km.

Standard Vehicle	[l]	68.2	[%]	100.0
Hybrid Vehicle with Battery ESS	[l]	61.0	[%]	89.4
Hybrid Vehicle with DLC ESS	[l]	62.8	[%]	92.1
Hybrid Vehicle with Flywheel ESS	[l]	62.1	[%]	91.1

Table 9. Fuel Consumption of two-coach Hybrid Bom Sinal Mobile 2 on CleanER-D Suburban Service Profile

When being equipped with hybrid components and a battery ESS, the fuel consumption is reduced by 10.6 % to 61.0 litres. The same applies for the DLC and flywheel ESS, which reduce the fuel consumption from 68.2 litres to 61.3 litres (7.9 %) and 62.1 litres (8.9 %), respectively.

7 LCC Results for hybrid Bom Sinal Mobile 2

The following three sections 7.1 to 7.3 describe the LCC results obtained for the hybrid 2-coach Bom Sinal Mobile 2 with the three ESS technologies. The LCC results for the hybrid Siemens Desiro cannot be shown in this public paper due to non-disclosure agreements.

7.1 Battery ESS

The hybrid components and battery ESS for the Bom Sinal DMU-2 cost approximately 202,000 € (see Figure 20). After 10 years of operation the initial expenses for the components are paid off. Since the battery has to be replaced after 12 years for the given use scenario, the LCC bar chart as shown in Figure 21 has a negative value in year 12. This means, that it takes 13 years to make profit with the installed hybrid components.

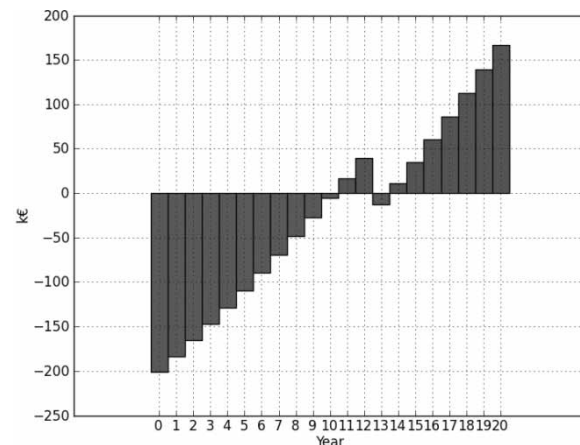


Figure 21. LCC Analysis of Bom Sinal Mobile 2 DMU-2 with Battery ESS on CleanER-D Suburban Service Profile

7.2 DLC ESS

After 8 years of operation, the 124,000 € costs for the DLC ESS and hybrid components are paid off. In the end of the 20-year service life of the train vehicle, around 210,000 € are saved by the vehicle operator as shown in the bar chart in Figure 22.

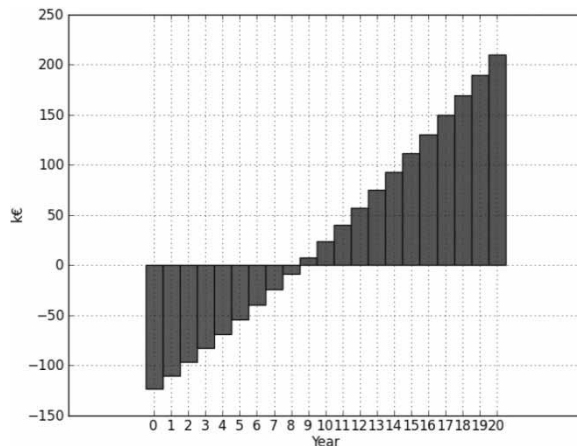


Figure 22. LCC Analysis of Bom Sinal Mobile 2 DMU-2 with DLC ESS on CleanER-D Suburban Service Profile

7.3 Flywheel ESS

For the flywheel ESS, with approximately 190,000 € nearly the same savings are achieved compared to the costs of 180,000 € for the energy storage system and the hybrid components. After 10 years of operation, the vehicle operator realises profit with the hybrid train vehicle (see Figure 23).

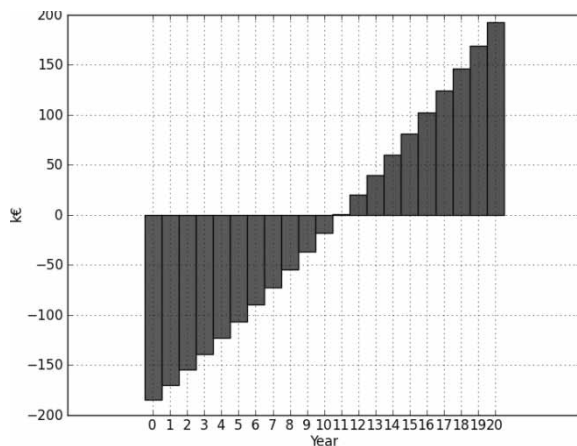


Figure 23. LCC Analysis of Bom Sinal Mobile 2 DMU-2 with Flywheel ESS on CleanER-D Suburban Service Profile

8 Conclusion

With the help of a python-based simulation tool, two railway vehicles were investigated in terms of their hybridization potential for electrical energy storage systems. Depending on the type of service profile, fuel savings in the range of 4.2 to 10.6 % compared to the non-hybrid vehicles are achieved.

Based on realistic data for the different ESS types and prices for the hybrid components and storage systems, a LCC analysis shows that for the hybrid Bom Sinal on the CleanER-D suburban service profile a DLC ESS is the best option. Even though the DLC ESS achieves the lowest fuel savings on this service profile, it ends up with the highest return on investment in a 20 year service life for the railway vehicle.

The investigations show, that it is very important to have a software tool at hand to design the optimum electrical storage type and size for a railway vehicle for each individual use case. By the means of simulation, fuel savings and storage lifetime predictions can be conducted at a very early project phase.

9 References

- [1] Fujii, T., Teraya, N. and Osawa, M. (2004) 'Development of a NE Train', *JR East Technical Review* - No. 4, pp. 62-70.
- [2] Lehmann, I. et al., 2011. *Hybrid-Powerpack für nachhaltigen und umweltfreundlichen Triebwagenantrieb*. Eisenbahntechnische Rundschau No. 9, September 2011, pp. 18-22.
- [3] Shiraki, N., Satou, H. and Arai, S., 2010. A Hybrid System for Diesel Railcar Series Ki-ha E200. *The 2010 International Power Electronics Conference*, Sapporo, Japan, 21-24 June 2010, pp. 2853-2858
- [4] Railway Gazette International, 2008. *Diesel-battery hybrid trial ends* Available at: <http://www.railwaygazette.com/nc/news/single-view/view/diesel-battery-hybrid-trial-ends.html>
- [5] AKASOL GmbH Publication, 2012. *AKA-MODULE. HIGHLY FUNCTIONAL. VALIDATED. UNIQUELY POWERFUL*. Available at: http://www.akasol.com/fileadmin/Kundendaten/pdf/datenblatt/update_dezember_2012/AKASOL_Datenblatt_AKAMODULE_E_12_2012.pdf
- [6] Danzer, M., Kuhn, R. and Döring, H., 2012. *Expert Discussions between Voith Turbo and the Centre for Solar Energy and Hydrogen Research Baden-Württemberg, Ulm*.

- [7] Siemens, 2013. Sitras MES: Mobile energy storage unit for railway vehicles. Available at: <http://w3.usa.siemens.com/mobility/us/Documents/en/rail-solutions/railway-electrification/dc-traction-power-supply/sitras-mes-en.pdf>.
- [8] Dr. Schneuwly, A. -Maxwell Technologies, 2006. *Designing powerful electronic solutions with ultracapacitors*. Available at: http://www.eetimes.com/document.asp?doc_id=1273082&page_number=1.
- [9] Green Car Congress, 2011. New Urbino hybrid bus with Voith parallel hybrid system. Available at: <http://www.greencarcongress.com/2011/03/urbino-20110325.html>
- [10] Maxwell Technologies, 2013. *125 V Heavy Transportation Module BMOD0063 P125 B04/B08 Datasheet*. Available at: http://www.maxwell.com/products/ultracapacitors/docs/datasheet_bmod0063_1014696.pdf.
- [11] Ricardo, 2014. Flywheel Technology. Available at: <http://www.ricardo.com/en-GB/Our-Markets/Clean-Energy-and-Power-Generation/Energy-Storage-Systems/flywheel/>
- [12] Porsche, 2014. GT3 R Hybrid Flywheel Technology. Available at: <http://www.porsche.com/usa/eventsandracing/motorsport/racingcars/911gt3r-hybrid-997/technologyandconcept/> (Accessed: 7 January 2014)
- [13] Deutsche Bahn AG, 2001. Applications for energy storage flywheels in vehicles of Deutsche Bahn AG. World Congress on Railway Research 2001, Cologne, Germany, 25 – 29 November 2001. Available at: <http://www.uic.org/cdrom/2001/wcrr2001/start.htm>.
- [14] Voith Publication, 2013. *Combining Ride Comfort with Economy. DIWA.5*. Available at: http://resource.voith.com/vt/publications/downloads/483_e_483_e_g_1747_e_diwa5_2012-08_screen.pdf
- [15] European Commission, 2013. *Clean European Rail-Diesel, D7.2.1: Detailed Specification: Parameters definition*. Brussels (FP7 – 234338).
- [16] Vibrationdata Python Wiki, 2013. *Rainflow Fatigue*. Available at: http://vibrationdata.com/python-wiki/index.php?title=Rainflow_Fatigue
- [17] Swierczynski, M., Teodorescu, R., Rodriguez, P., 2011. *Lifetime investigations of a lithium iron phosphate (LFP) battery system connected to a wind turbine for forecast improvement and output power gradient reduction*. 15th Battcon Stationary Battery Conference and Trade Show, Battcon/Albercorp.

Hybride Simulation des Materialflusses in Aluminium-Druckgussbetrieben

B. Eng. Sven Hirschberg, Prof. Dr. phil. nat. Wolfgang Schlüter, Dipl.-Ing.(FH) Ansgar Ringleb
Hochschule Ansbach
s.hirschberg@hs-ansbach.de

Das Kompetenzzentrum Industrielle Energieeffizienz (KIEff) der Hochschule Ansbach arbeitet im Rahmen des Forschungsprojektes „Smart-Melting“ gemeinsam mit den Partnern ZF Gusstechnologie GmbH und Bundesverband der Deutschen Giesserei-Industrie an einer Fertigungssimulation für Aluminium-Druckguss- und Schmelzbetriebe. In einem ersten Schritt ist dazu erstmalig eine Simulation des Materialflusses aufgebaut worden. Die besondere Schwierigkeit liegt dabei in der Kombination einer großen Anzahl von kontinuierlichen und ereignisgesteuerten Komponenten zu einem hybriden System. Die Umsetzung erfolgt mit Hilfe von Matlab/Simulink und dem ebenfalls von Mathworks entwickelten Simulationstool für ereignisgesteuerte Systeme Stateflow.

Das zu simulierende System besteht aus den Bereichen Schmelzbetrieb mit den Komponenten Schmelzöfen, Aluminiumdistribution mit den Komponenten Transporter und Druckgussbetrieb mit den Komponenten Druckgussmaschinen. Der komplette Aluminium-Druckgussbetrieb mit parallel arbeitenden Druckgussmaschinen, Transportern und Schmelzöfen wird als Gesamtsystem kontinuierlich betrachtet. Die Produktion der Schmelze im Schmelzofen erfolgt als technischer Prozess ebenfalls kontinuierlich. Bei der Entnahme von Aluminium aus dem Schmelzofen, dem Transport und der Produktion von Druckgussteilen handelt es sich um ereignisgesteuerte Prozesse, die ereignisdiskret zu simulieren sind. Auf Grund der großen Anzahl der Komponenten Schmelzofen, Transporter und Druckgussmaschine wurde das System modular aufgebaut, so dass eine einfache Erweiterung oder auch Neukonfiguration möglich ist. Eine erste einfache Steuerung des Druckgussbetriebes wurde in die Simulation implementiert.

1 Einleitung

Das Forschungsprojekt „Smart-Melting“, das vom Kompetenzzentrum Industrielle Energieeffizienz der Hochschule Ansbach durchgeführt wird, ist Teil des vom Freistaat Bayern geförderten Technologieverbundes „Green Factory Bavaria“. Ziel des Forschungsverbundes ist die „grüne Fabrik“, die möglichst energieeffizient produziert. Im Fokus des Forschungsprojektes „Smart-Melting“ steht die Steigerung der Energieeffizienz in Druckgussbetrieben der Aluminiumindustrie. Durch eine Anbindung der Industrie an ein Smart-Grid, die eine Anpassung der Produktion an die zur Verfügung stehende Strommenge zur Folge hat, steigen dabei die innerbetrieblichen Anforderungen an die Fertigungssteuerung.

Um sich an die neuen veränderlichen Anforderungen anpassen zu können, ist der bestehende Betrieb genau zu analysieren. Hierfür erfolgt im vorliegenden Projekt zunächst eine umfassende Datenaufnahme zu relevanten Komponenten. Zudem wird der Betrieb des Industriepartners (ZF Gusstechnologie GmbH) simulationstechnisch abgebildet und mit den Informa-

tionen aus der Datenaufnahme validiert. Somit ist eine vielseitigere Analyse des Betriebes möglich.

In einem ersten Schritt wird eine Simulation des Materialflusses, der hauptsächlich den Energieverbrauch bestimmt, aufgebaut. Es handelt sich dabei um ein hybrides System aus kontinuierlichen und diskreten Prozessen.

Die Vorgehensweise beim Aufbau der Materialflusssimulation und die dabei angewandte Methodik werden in den folgenden Abschnitten erläutert. In die Materialflusssimulation wurde eine erste einfache Steuerung implementiert.

2 Der Schmelz- und Druckgussbetrieb

Für den Aufbau einer Simulation des Materialflusses ist zunächst das reale System zu betrachten. Es handelt sich dabei um einen in Reihe geschalteten Schmelz- und Druckgussbetrieb. Der Schmelzbetrieb wird von Staplern mit Aluminium in fester Form versorgt. In den Schmelzöfen wird das Aluminium geschmolzen und auf die für den Druckgussbetrieb

benötigte Temperatur überhitzt. Bei Nachfrage vom Druckgussbetrieb wird aus den Schmelzöfen eine definierte Menge flüssigen Aluminiums entnommen und zum jeweiligen Dosierofen der Druckgussmaschine transportiert. Die Druckgussmaschine entnimmt zur Herstellung eines Druckgussteils eine bestimmte Menge Flüssialuminium aus dem Dosierofen. Die entnommene Menge des Flüssialuminiums und die Taktzeit können dabei als konstant angesehen werden. Jedoch sind beide Prozessparameter von Produkt oder Werkzeug abhängig. Im Druckgussbetrieb sind die Komponenten Schmelzofen, Druckgussmaschine und Stapler (Transport) zu modellieren. Der zu untersuchende Betrieb besitzt dabei rund 40 Druckgussmaschinen, 4 Schmelzöfen und 10 Stapler. Die Schmelzöfen sind zentral angeordnet, die Druckgussmaschinen sind auf mehrere Hallen verteilt, dadurch ergeben sich unterschiedlich lange Transportwege.

2.1 Steuerung im realen Betrieb

Die Regelung des Materialflusses im Betrieb besitzt keine zentrale Steuerung. Grundsätzlich bestimmen die Druckgussmaschinen mit ihrer Auslastung den Materialfluss. Ihre Auslastung wird durch den Produktionsplan festgelegt (s. Abbildung 1). Die Dosieröfen der Druckgussmaschinen, die eine bestimmte Menge Aluminium fassen und als Pufferspeicher dienen, geben bei Unterschreiten eines bestimmten Füllstandes ein Signal an den Schmelzbetrieb weiter, das dort für die Mitarbeiter visualisiert wird. Bei Vorhandensein von Aluminiumschmelze und Transportkapazität im Schmelzbetrieb, erfolgt die Versorgung der jeweiligen Druckgussmaschinen. Die Versorgung der Schmelzöfen mit neuem festem Aluminium erfolgt „auf Sicht“. Durch visuelle Kontrolle von Schmelzschacht und Warmhaltebecken des Schmelzofens bestimmt das Personal den Beschickungszeitpunkt. Bei Erreichen des maximalen Füllstandes im Warmhaltebecken des Schmelzofens schaltet dieser ab, die Beschickung wird gestoppt.

Eine intelligente Verknüpfung von Druckguss- und Schmelzproduktion ist nicht vorhanden. Ebenfalls besteht wenig Kenntnis über das mögliche Optimierungspotential in Bezug auf Energieeffizienz und Prozess- bzw. Versorgungssicherheit.

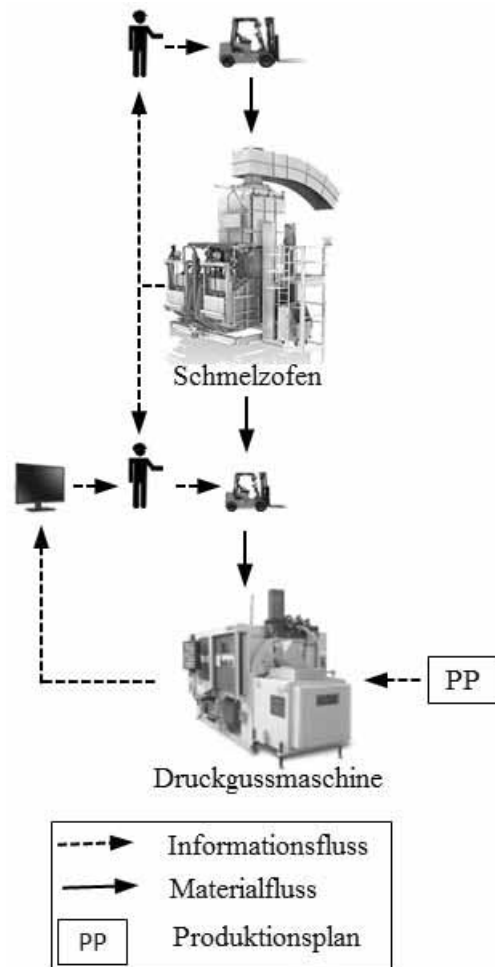


Abbildung 1: Steuerung im realen Betrieb

2.2 Modellierung

Zur Analyse und Optimierung des dargestellten Problems stellt die Simulation, ein wertvolles Hilfsmittel dar. Allerdings stellt die Komplexität des realen Systems besondere Anforderungen an den Aufbau des Modells durch den Zusammenhang von Simulationstiefe mit der Zunahme von Aufwand und Rechenzeit [1].

Im vorliegenden System weisen vor allem die Komponenten Schmelzofen und Druckgussmaschine eine hohe Komplexität auf. Das benötigte Abstraktionsniveau, um die gewünschten Ergebnisse mit der Simulation erreichen zu können, kann dabei erst nach Entwicklung und Validierung der ersten Modelle abgeschätzt werden. Daher empfiehlt sich ein modularer Aufbau der Simulation.

In einem ersten Modell wird eine Materialflusssimulation aufgebaut, die aus den Komponenten Schmelz-

ofen (SO), Transporter (T) und Druckgussmaschine (DGM) inklusive Dosierofen (DO) besteht. Die einzelnen Prozesse teilen sich dabei in kontinuierliche und diskrete, ereignisgesteuerte Prozesse auf (s. Tabelle 1).

Komponente	Prozess	
Schmelzofen	Beschickung	Diskret
	Schmelzen	Kontinuierlich
	Entnahme	Diskret
Transporter	Transport	Diskret
Dosierofen	Befüllung	Diskret
	Entnahme	Diskret
Druckgussmaschine	Produktion	Diskret
Druckgussbetrieb	Produktion	Kontinuierlich

Tabelle 1. Auflistung der Prozesse

Der komplette Druckgussbetrieb (DGB) wird als Gesamtsystem kontinuierlich betrachtet. Die Abhängigkeiten des realen Systems, die auf die Simulationsebene übertragen werden, sind im Folgenden formalisiert dargestellt.

$DGB(S_i, T_j, DGM_k)$
 $S_i(\dot{m}_{s,i}(t), sto_i(t))$
 $T_j(m_T, st_j(t))$
 $DGM_k(t_{DGMk}, m_{DGMk}, sto_{DGMk})$
 mit:

$i=1 \dots n_{SO}$, n_{SO} =Anzahl Schmelzöfen
 $j=1 \dots n_{TP}$, n_{TP} =Anzahl Transporter
 $k=1 \dots n_{DGM}$, n_{DGM} =Anzahl Druckgussmaschinen

Charakteristische Eigenschaften des Schmelzofens (S_i) sind seine Schmelzleistung ($\dot{m}_{s,i}(t)$) und sein Speicherstand ($sto_i(t)$). Ein Transporter j (T_j) wird durch seinen momentanen Zustand ($st_j(t)$) und seine Transportmenge (m_T) charakterisiert. Für die Druckgussmaschine k (DGM_k) sind die Taktzeit (t_{DGMk}), die Masse des zu fertigenden Werkstücks (m_{DGMk}) und der Speicherstand des Dosierofens ($sto_{DGMk}(t)$) bestimmende Parameter.

Durch Kombination von kontinuierlichen und diskreten Prozessen entsteht eine hybride Simulation. Diese wird als dynamische Simulation aufgebaut, um bspw. Unterschiede der Tag-/Nachtschicht sowie Wochenendschicht in der Produktion abbilden zu können.

Der Aufbau des Modells erfolgt mit Hilfe von Matlab/Simulink, für die ereignisdiskret zu simulierenden Prozesse wird das Simulationstool Stateflow verwendet.

3 Aufbau der hybriden Simulation

Methodik

Bei der Entwicklung des Modells wird inkrementell-iterativ vorgegangen. Dabei wird der Anwendungsfall in den Mittelpunkt des Planungsprozesses gestellt. Es wird versucht, sehr früh ein lauffähiges Grundsystem zu erstellen. Durch die inkrementelle Entwicklung wird das System nicht in einem Zug, sondern in einer Reihe von aufeinander aufbauenden, funktional erweiterbaren Ausbaustufen aufgebaut. Durch das iterative Vorgehen können die Erfahrungen aus vorangegangenen Entwicklungsschritten in das Modell einfließen.

Die inkrementell-iterative Simulationsentwicklung eignet sich bei dem vorliegenden Projekt mit innovativem Charakter besonders, da noch nicht alle Anforderungen an die Simulation bekannt sind und der Endzustand noch nicht feststeht [2]. Die Vorstellungen der Industriepartner und das sich verändernde betriebliche Umfeld lassen sich durch diese Vorgehensweise einfacher integrieren.

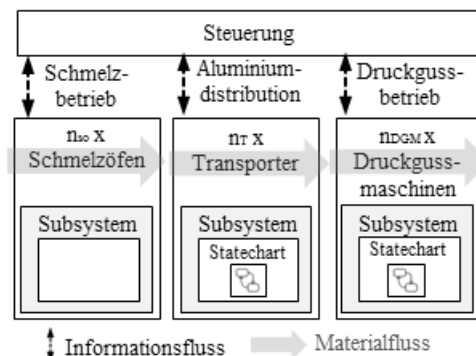


Abbildung 2: Schema des Gesamtmodells

Das Simulationsmodell wird modular mit den drei Komponenten Schmelzofen (SO), Transporter (T) und Druckgussmaschine (DGM) aufgebaut (s. Abbildung 2). Dies ermöglicht eine einfache und schnelle Erweiterung des Modells sowie die Simulation verschiedener Betriebsgrößen.

Im Folgenden wird der Aufbau der Komponenten Schmelzofen, Transporter und Druckgussmaschine erläutert. Aufgrund der Methodik beim Aufbau der Komponenten, werden diese zunächst einfach gehalten.

ten und in weiteren Iterationsschritten an Komplexität zunehmen.

Schmelzofen

Der Schmelzofen schmilzt bei konstanter Schmelzleistung Aluminium. Erhält der Schmelzofen das Signal zum Befüllen des Tiegels eines Transporters, wird dem Speicher des Schmelzofens kontinuierlich über einen festgelegten Zeitraum Aluminium entnommen.

Transporter

Für die Distribution des Aluminiums zu den jeweiligen Druckgussmaschinen stehen in der Simulation mehrere Transporter zur Verfügung. Erhält ein Transporter einen Auftrag von der Steuerung, wird ihm die zu beliefernde Druckgussmaschine mitgeteilt und er fährt los. Der Transporter durchläuft während der Ausführung die in Abbildung 3 dargestellten Zustände.

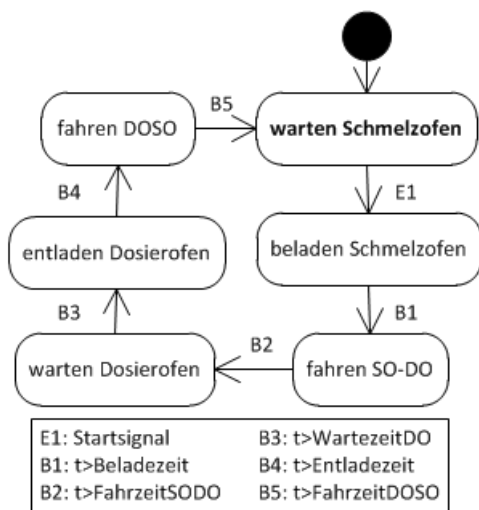


Abbildung 3: Zustandsdiagramm des Transportprozesses

Hat ein Transporter einen Auftrag ausgeführt, nimmt er wieder seinen Grundzustand ein („Warten-Schmelzofen“) und steht zur Ausführung eines neuen Auftrages zur Verfügung. Zwischen den Staplern selbst gibt es keine Unterschiede. Der Faktor Mensch wird zunächst nicht berücksichtigt.

Druckgussmaschine

Die Druckgussmaschine, besteht aus dem Dosierofen und der eigentlichen Druckgussmaschine. Jeder Dosierofen besitzt einen bestimmten Anfangsfüllstand. Bei unterschreiten eines bestimmten Füllstandes wird

ein Signal an die Steuerung gegeben, die dann einen Transporter mit der Befüllung beauftragt. Der Dosierofen wartet bis dieser befüllt wird und geht auf den Anfangszustand zurück (s. Abbildung 4).

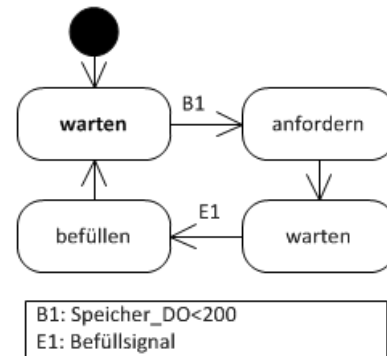


Abbildung 4: Zustandsdiagramm des Dosierofens

Der eigentliche Druckgussprozess der Druckgussmaschine ist in Abbildung 5 dargestellt. Bevor der Druckgussprozess zur Produktion eines neuen Werkstücks erneut durchlaufen wird, wird der Füllstand (sto_{DO}) überprüft und mit der benötigten Menge ($\text{wt}_{\text{WP},k}$) verglichen. Ist der Füllstand im Dosierofen zu gering, um ein weiteres Druckgussteil herzustellen, bleibt die Druckgussmaschine stehen. Für die Herstellung eines Druckgussteils benötigt der Prozess eine bestimmte Taktzeit (ct), diese Taktzeit kann zwischen den einzelnen Druckgussmaschinen variieren (ct_k).

Der Prozessablauf in den einzelnen Druckgussmaschinen ist identisch, sie unterscheiden sich jedoch in der Taktzeit (ct_k) und im Aluminiumverbrauch pro Werkstück ($\text{wt}_{\text{WP},k}$).

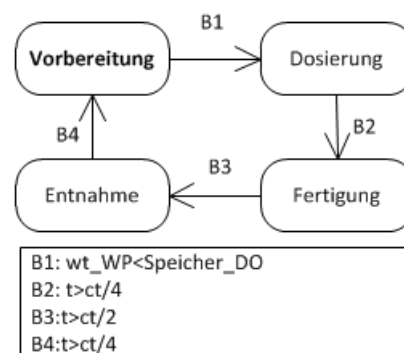


Abbildung 5: Zustandsdiagramm des Druckgussprozesses

Steuerung

Die Steuerung von Schmelzöfen und Transporter im momentan aufgebauten Simulationsmodell erfolgt zentral d.h. alle relevanten Informationen laufen in der Steuerung zusammen. Die Steuerung verarbeitet die eingehenden Informationen und leitet diese oder daraus abgeleitete Handlungsanweisungen an die entsprechenden Stellen weiter (s. Abbildung 6).

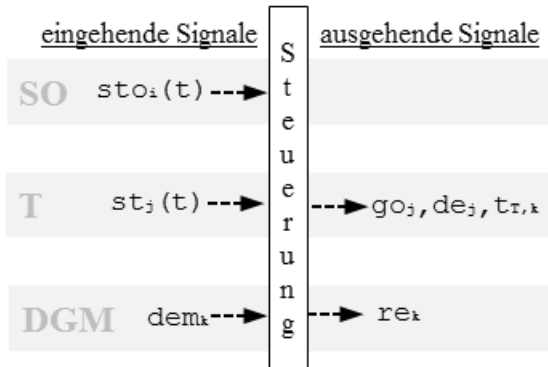


Abbildung 6: Ein- und ausgehende Signale an der Steuerung

Vom Schmelzofen erhält die Steuerung den Speicherstand im Ofen ($sto_i(t)$), vom Transporter den aktuellen Zustand ($st_j(t)$) und von der DGM ein Nachfragesignal (dem_k). An den Transporter gibt die Steuerung das Startsignal (go_j) und das Ziel (de_j) und, nachdem der Transporter das Ziel erreicht hat, an die DGM das Befüllsignal (re_k). Die Steuerung verarbeitet die eingehenden Signale in drei parallel ablaufenden Aktionen (s. Abbildung 7.1-3).

Geht bei der Steuerung ein Nachfragesignal einer DGM ein, so speichert sie diese in einer Warteliste die nach dem FIFO-Prinzip arbeitet (s. Abbildung 7.1). Durch die Warteliste wird verhindert, dass Nachfragesignale von DGMs bei Auslastung aller Transporter unbearbeitet bleiben oder verloren gehen. Die Steuerung überprüft die aktualisierte Warteliste ständig auf Aufträge. Bei vorhandenem Auftrag und einem ausreichendem Speicherstand in einem der Schmelzöfen vergibt die Steuerung den Auftrag an den ersten freien Transporter (s. Abbildung 7.2). Der Auftrag beinhaltet dabei Startpunkt (SO_i), Zielpunkt (DGM_k), und Fahrtzeit ($t_{\tau,k}$). In einer dritten Aktion überprüft die Steuerung den Zustand der Transporter. Befindet sich ein Transporter zur Befüllung am Dosierofen, so wird dieser darüber informiert und der Speicher des jeweiligen Dosierofens wird gefüllt (s. Abbildung 7).

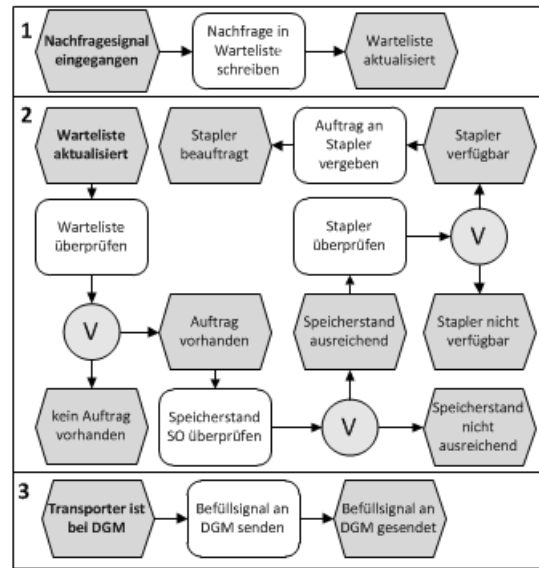


Abbildung 7: Ereignisgesteuerte Prozesskette der Steuerung

Im Betrieb des abzubildenden Industriepartners herrscht bei Volllastbetrieb eine Auslastung aller Transporter, somit ist die Warteliste für die Abbildung dieses Betriebes bereits im Grundmodell ein notwendiger Bestandteil.

Die dargestellte (einfache) Steuerung ist bereits in der Lage, das aktuelle Modell ausfallfrei zu betreiben und zu kontrollieren.

4 Möglichkeiten und Grenzen des Modells

Durch die modular aufgebaute Simulation, können die einzelnen Komponenten schnell vervielfältigt werden und somit verschiedene Betriebsgrößen innerhalb kürzester Zeit aufgebaut und simuliert werden. Durch die variabel gehaltenen Parameter, lassen sich die Komponenten bis zu einem gewissen Grad individualisieren. Es können beispielsweise unterschiedlich stark schwankende Schmelzleistungen, verschiedene Taktzeiten, Transportzeiten oder Werkstückgrößen festgelegt werden.

Das Modell ermöglicht eine Simulation einer Tages- oder Wochenproduktion durch zeitlich veränderliche Variablen.

Die Validierung der Simulation wird in Zusammenarbeit mit dem Industriepartner durchgeführt. Insbesondere im Bereich Steigerung der Prozesssicherheit ist eine genaue Validierung der Simulation notwendig,

um Risikoszenarien simulieren und daraus aussagekräftiges Ergebnisse ermitteln zu können.

Zur Berücksichtigung des Energieverbrauchs in der bestehenden Materialflusssimulation müssen die einzelnen Komponenten und Prozesse der Simulation mit den ermittelten Verbrauchswerten aus dem realen Betrieb erweitert werden.

5 Zusammenfassung und Ausblick

Ziel des vorliegenden Projektes ist die Entwicklung einer Fertigungssimulation für Aluminium-Druckguss- und Schmelzbetriebe. Hierfür wurde zunächst eine Simulation des Materialflusses aufgebaut. Diese hybride Simulation, die aus der Kombination von kontinuierlichen und ereignisgesteuerten Prozessen besteht, ist durch ihren modularen Aufbau bereits in der Lage verschiedene Betriebsgrößen abzubilden und einfache Szenarien mit veränderlichen Parametern zu simulieren. Die Integration der Energieverbräuche der einzelnen Prozesse erfolgt in den nächsten Iterationsschritten.

Einer der wichtigsten Arbeitsschritte die dem Aufbau des Grundmodells folgen, ist die umfangreiche Erfassung von Betriebsdaten und anschließende Validierung des Modells. Erst nach diesem Schritt können aus der Simulation direkt übertragbare Ergebnisse für den Industriepartner erzeugt werden.

Im weiteren Verlauf des Projektes soll die Materialflusssimulation zu einer Fertigungssimulation ausgebaut werden, in der Fertigungsstrategien in einem Aluminium-Druckgussbetrieb im Hinblick auf ihre Prozesssicherheit und Energieeffizienz beurteilt und optimiert werden können. Auswirkungen zukünftiger Entwicklungen auf den Produktionsprozess und die Produktivität werden sich durch die Simulation ebenfalls besser beurteilen lassen.

6 References

- [1] D. Friedrich. *Simulation in der Fertigungssteuerung*. Deutscher Universitäts Verlag, Deutschland, 1998
- [2] C. Engelhardt-Nowitzki, O. Nowitzki, B. Krenn. *Praktische Anwendung der Simulation im Materialflussmanagement*. Gabler Verlag, Deutschland, 2008
- [3] L. März, W. Krug, O. Rose, G. Weigert. *Simulation und Optimierung in Produktion und Logistik*. Springer Verlag, Deutschland, 2011.

Automatisierte Auswertung von DYMOLA-Simulationsergebnissen mit MATLAB

Ivan Windemut¹, Mounir Nasri¹, Holger Dittus¹

¹Deutsches Zentrum für Luft- und Raumfahrt e.V.,

Institut für Fahrzeugkonzepte

ivan.windemut@dlr.de

Der bekannteste Modelica-Simulator DYMOLA von Dassault Systemès bietet diverse Möglichkeiten für die Darstellung von Simulationsergebnissen. Mit DYMOLA ist auch ein gewisser Grad der Automatisierung über die integrierte Befehlssprache und .mos-Skripte erreichbar. Diese Skripte stoßen jedoch an ihre Grenzen, wenn eine kompliziertere Nachbearbeitung der Ergebnisse, umfangreichere Parametervariationen oder spezieller Datenexport benötigt werden.

Die Simulationsergebnisse von DYMOLA liegen in Form von .txt oder .mat-Dateien vor. Daher liegt es nahe, MATLAB für deren Aufbereitung zu nutzen. Eine gute Basis dafür bilden die mit DYMOLA mitgelieferten .m-Skripte wie z.B. *dymbrowse*, *dymload* und *dymget*. Darauf aufbauend wurde am DLR-Institut für Fahrzeugkonzepte ein strukturierter Skriptsatz namens „SARA“ entwickelt. „SARA“ steht für „Simulation Automation & Result Analysis“ und ist wie folgt gegliedert:

- Initialisierung
- Parametrisierung
- Aufruf der Simulation(en)
- Auswertung der Ergebnisse, ggf. Neustart von Simulationen mit veränderten Parametern
- Aufbereitung von Grafiken

Der Skriptsatz wird in erster Linie für Parametervariation, Postprocessing und grafische Darstellung von Ergebnissen verwendet. Generell lässt sich mit „SARA“ die volle Funktionsvielfalt von MATLAB ausnutzen.

Im Beitrag wird die Funktionalität von „SARA“ anhand von Beispielen aus aktuellen Arbeiten des Instituts für Fahrzeugkonzepte im Bereich der Automobil- und Schienenfahrzeugtechnik erläutert.

1 Einleitung

Seit Jahren gilt MATLAB als etabliertes numerisches Berechnungs- und Simulationswerkzeug in Wissenschaft und Industrie [1]. Mit voranschreitender Weiterentwicklung von MATLAB zu einer eigenständigen Programmiersprache wächst der Funktionsumfang und die Benutzerfreundlichkeit verbessert sich. Dabei spielt der Datenaustausch mit anderen Programmen eine zentrale Rolle. Eine bewährte Methode dafür ist der Import/Export der Daten in Standard-Dateiformate wie .txt oder .csv. Auch .mat-Dateien zählen heute zu Standard-Dateiformaten, so dass mehrere Programme solche Dateien erzeugen und bearbeiten können.

Auch der meist verwendete MODELICA-Simulator DYMOLA von Dassault Systemès speichert seine Ergebnisse standardmäßig als .mat - Dateien, wobei für die Zukunft auch andere Formate in der Diskussi-

on sind, deren Standard derzeit ausgearbeitet wird [2].

Es gibt für DYMOLA mehrere Möglichkeiten, die Modelle zu parametrisieren, die Simulationen zu steuern und Ergebnisse darzustellen. Eine davon ist die Verwendung von MATLAB, wobei ein Teil der dafür benötigten Basisskripte mit DYMOLA ausgeliefert wird. Darauf aufbauend wurde am DLR-Institut für Fahrzeugkonzepte ein strukturierter Skriptsatz namens „SARA“ („Simulation Automation & Result Analysis“) entwickelt. Der Skriptsatz erlaubt es, Randbedingungen für Simulationen festzulegen, Simulationen durchzuführen und Ergebnisse auszuwerten. Prinzipiell kann dazu jede MATLAB-Funktion genutzt werden.

In diesem Beitrag wird der Aufbau des Skriptsatzes „SARA“ beschrieben und die Funktionalität anhand von Beispielen erläutert.

2 Strukturierter Skriptsatz „SARA“

2.1 Aufbau

Der Aufbau des Skriptsatzes „SARA“ basiert auf der Ordnerstruktur in Abbildung 1.

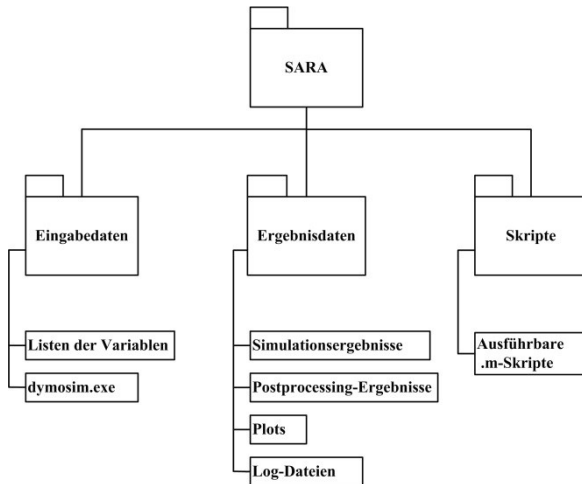


Abbildung 1. SARA - Ordnerstruktur

Die Beziehung einzelner Skripte zueinander und ihre Teilaufgaben sind in der Abbildung 2 schematisch dargestellt.

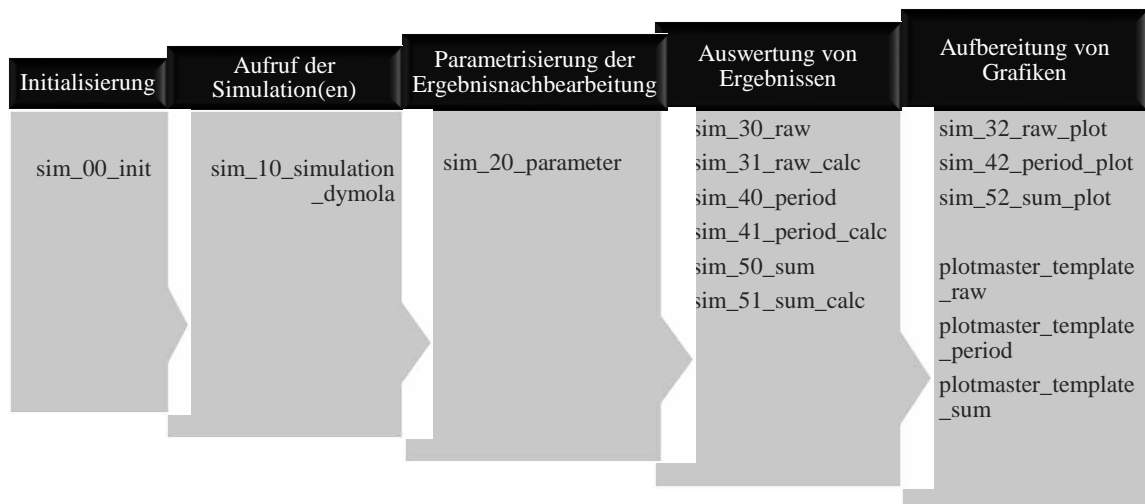


Abbildung 2. Struktur des Skriptsatzes „SARA“

Die Skripte mit den Namen *sim_30_raw.m*, *sim_40_period.m* und *sim_50_sum.m* werden zum Auswerten und Zusammenfassen von Simulationsergebnissen benutzt. Dabei kommen die im Ordner **Eingabedaten** gespeicherten Variablenlisten zum Einsatz. Die in diesen Textdateien angegebenen Variablen werden aus der DYMOLA Ergebnisdatei extra-

Das Initialisierungsskript *sim_00_init.m* ist den Anderen übergeordnet. In diesem wird der Simulationspfad vorgegeben, in dem sich die Basisordnerstruktur befindet (vgl. Abbildung 1) und es wird festgelegt, welche weiteren Teilskripte auszuführen sind.

Das Teilskript *sim_10_simulation_dymola.m* dient zur Steuerung der Simulation. Das Skript basiert auf dem mit DYMOLA gelieferten Skript *dymosim.m* [3]. In *sim_10_simulation_dymola.m* werden Angaben zur Versuchsreihe, dem Anwender und der verwendeten DYMOLA-Version gemacht. Außerdem werden die Simulationsbedingungen wie die Simulationszeit, die Eigenschaften der numerischen Lösungsmethode und der Ausgabedaten gesetzt. Darüber hinaus ermöglicht das Skript Parametervariationen, wobei ein oder mehrere Parameter variiert werden können. Im letzteren Fall müssen für jeden Parameter gleich viele Werte angegeben werden. Das Skript kombiniert dann jedes Element i des Parametervektors $p1(i)$ mit dem Element i des Parametervektors $p2(i)$ etc. Für jeden Wert bzw. jede Wertekombination der Parameter wird ein Simulationslauf durchgeführt.

Das Skript *sim_20_parameter.m* ermöglicht die Eingabe von Parametern, die für das Postprocessing benötigt werden, z.B. Umrechnungsfaktoren.

hiert und gemeinsam mit dem Simulationszeitvektor abgespeichert. Durch die Zusammenfassung und Extrahierung wird die Größe der Ergebnisdateien reduziert.

Die Skripte mit der Endung *_calc* dienen für weiterführende Berechnungen mit einzelnen Verläufen wie z.B. Umrechnen von Einheiten.

Die Skripte mit der Endung `_plot` werden für die grafische Darstellung von einzelnen und / oder zusammengefassten Simulationsergebnissen eingesetzt. Diese Skripte verwenden Funktionen, die im am DLR-Institut für Fahrzeugkonzepte entworfenen Basisskript `plotmaster.m` definiert sind. Das Basisskript erlaubt vielseitige Formatierung, Darstellung und Export von Plots, z.B. in spezielle Formate für ein- oder mehrzeilige Publikationen. Die Eigenschaften einzelner Plots (Layout, Beschriftungen, Farben etc.) können als Vorlagen (`plotmaster_template_`) für mehrmalige Nutzung gespeichert werden.

2.2 Funktionsbeschreibung

Für eine erfolgreiche Verwendung des Skriptsatzes „SARA“ müssen folgende Voraussetzungen erfüllt sein:

- Das zu simulierende Modell ist in DYMOLA zu einem ausführbaren Programm `dymosim.exe` kompiliert und im Ordner **Eingabedaten** abgespeichert
- Das Hilfsskript `generate_dymola_varlist` wurde auf die `dymosim.exe` angewendet und es liegt eine Liste aller zugänglichen Variablen `variables_all.txt` vor.
- Falls Ausführung von `raw-`, `period-` und `sum-`Skripten geplant ist, müssen für diese entsprechende Variablenlisten manuell erstellt werden.
- Die Pfade zu den Basisskripten sind im `MATLAB Search Path` dauerhaft oder für aktuelle MATLAB-Sitzung gespeichert.

Sind die Voraussetzungen erfüllt und alle gewünschten Funktionen im Skript `sim_00_init.m` aktiviert, kann das Skript in MATLAB ausgeführt werden (`run`). Der Prozess läuft vollautomatisch im Hintergrund ab: die `dymosim.exe` wird mit in Skripten definierten Parametern ausgeführt und die erzeugten Ergebnisdateien werden gemäß den in den Teilskripten implementierten Befehlen bearbeitet. Im Ordner **Ergebnisdaten** liegen dann entsprechend den Variablenlisten sortierte und umgerechnete Simulationsergebnisse sowie Plots in gewünschten Dateiformaten vor.

3 Anwendungsbeispiel: Thermisches Wagenkastenmodell

Im Rahmen des DLR-Projektes „Next Generation Train“ wurde ein thermisches Wagenkastenmodell in

der Modellierungssprache MODELICA auf Basis der HumanComfortLibrary [4] entwickelt, welches zur Ermittlung der Heiz- und Kühlleistungsbedarfe in Schienenfahrzeugen eingesetzt wird. Das Modell berücksichtigt neben der Wagenkastengeometrie, den verwendeten Materialien und den Umgebungseinflüssen (Außentemperatur, Sonneneinstrahlung, etc.) auch die bahnspezifischen Normen und Vorschriften. Durch Kopplung des thermischen Modells mit einem längsdynamischen Schienenfahrzeugmodell werden strecken-, geschwindigkeits- und wetterabhängige Simulationen ermöglicht, z.B. Fahrt an einem sonnigen Sommertag, Fahrt in einer eisigen Winternacht sowie Fahrten mit verschiedenen Besetzungsgraden. Darüber hinaus ist es möglich, mit Hilfe des Modells Algorithmen zur energieeffizienten Regelung von Klimaanlage zu entwickeln und zu testen [5].

Zur Untersuchung des Fahrgasteinflusses auf die Klimatisierungsleistung wurde mit Hilfe des „SARA“- Skriptsatzes für das thermische Wagenkastenmodell eine Variation des Parameters *Besetzungsgrad* durchgeführt.

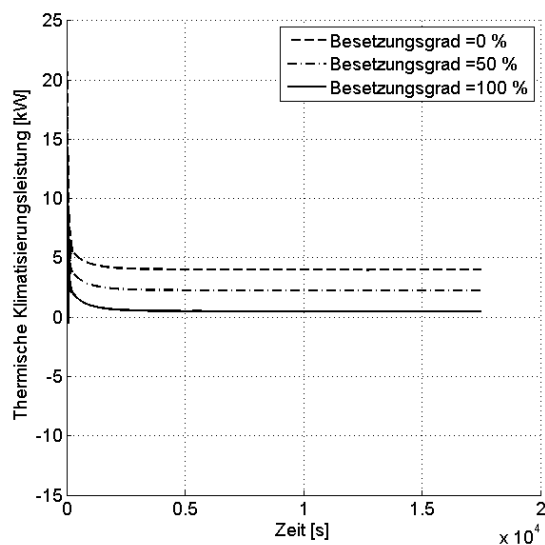


Abbildung 3. Variation des Besetzungsgrades

Für den Parameter wurden drei Werte vorgegeben: 0% (leerer Wagen), 50% (Hälfte der Plätze besetzt) und 100% (vollbesetzt). Es fanden drei Simulationsläufe statt und es wurden drei Ergebnisdateien erstellt. Aus diesen Dateien wurde die Variable `HVAC_Power_input` extrahiert, ihre Einheit von Watt in Kilowatt umgerechnet und über der Zeit aufgetragen. Der fertige Plot wurde anschließend als `.png`-

Bild gespeichert. Das Ergebnis ist in der **Abbildung 3** dargestellt.

4 Erweiterung von SARA

In technischen Anwendungen, insbesondere für Systemoptimierung, spielt neben der Parametervariation auch die Komponentenvariation eine wichtige Rolle. Müssen mehrere Modellkomponenten manuell geändert werden und danach Simulationsläufe gestartet werden, kostet der Prozess nicht nur viel Zeit, sondern man verliert auch schnell den Überblick.

Mit dem oben beschriebenen SARA-Skriptsatz kann eine Simulationsreihe erstellt werden, die mit einer bestimmten Modellkonfiguration eine Parametervariation durchführt. Dabei ist zwischen festen Parametern und Variationsparametern zu unterscheiden: feste Parameter sind für alle Simulationsläufe gleich, während die Variationsparameter einen bestimmten Wertebereich mit einer vorgegebenen Schrittweite durchlaufen.

Mit der im Kapitel 2 vorgestellten Version von SARA kann die Modellkonfiguration nicht verändert werden. Um eine Komponentenvariation zu ermöglichen, wurde der Skriptsatz SARA erweitert.

Die erste Erweiterung betrifft das Skript *sim_10_simulation_dymo.m*, damit Wertebereiche für die Parametervariation und Teilmodelle für Komponentenvariation eingegeben werden können. Dafür wird im Quellcode des MODELICA-Modells (*.mo*-Datei) die Zeile ersetzt, die das abgeleitete Teilmodell (extended model) definiert. Anschließend wird die neue *dymosim.exe* erzeugt und ausgeführt.

Die zweite Erweiterung betrifft die Aufbereitung von Grafiken. Für technische Berichte sowie PowerPoint-Präsentationen wird meist ein vordefiniertes Layout verwendet, wobei insbesondere folgende Punkte festgelegt werden:

- Größe, Anzahl und Position der Abbildungen
- Schriftgröße und Schriftart, vor allem für Achsenbeschriftungen und Diagrammüberschriften
- Abstände, Verhältnis Text zu Bild

Die Erweiterung berücksichtigt diese Vorgaben. Damit erhält man ein einheitliches Layout für alle Grafiken. Außerdem wird ein grafischer Vergleich zwischen mehreren Simulationsdurchläufen sowie zwi-

schen den Signalen eines Simulationsdurchlaufs ermöglicht.

Für die Anwendung des erweiterten Skriptes ist es notwendig, eine Liste der darzustellenden Variablen in Form einer *.txt*-Datei zu erstellen. Die Grundlage dafür bildet die zuvor mit dem Hilfsskript *generate_dymola_varlist.m* erstellte *variables_all.txt* (siehe Kapitel 2.2 **Fehler! Verweisquelle konnte nicht gefunden werden.**). Dabei ist zu beachten, dass die Variablen, die in einem Graphen dargestellt werden sollen, gleiche Einheiten haben. Die Erstellung einer *.txt*-Datei pro physikalische Größe hat sich als sinnvoll erwiesen. Sehr oft werden Dateien wie *pressure.txt*, *temperature.txt*, *electrical.txt*, *control.txt*, *power.txt*, *heatwaste.txt* und *massflow.txt* verwendet (Code 1).

```
1 coolantCircuit_NT.FuelCell.port_a.m_flow
2 coolantCircuit_NT.PTC.port_a.m_flow
3 coolantCircuit_HT.PTC.port_a.m_flow
4 coolantCircuit_HT.FuelCell.port_a.m_flow
5 controlBus.coolantCircuit2.control.NTK_
  mdot_air_kgps
6 controlBus.coolantCircuit1.control.Cabin_
  mdot_KW_kgps
```

Code 1. Inhalt der Datei *massflow.txt*

Die Auswahl von DYMOLA-Simulationsergebnissen (*.mat*-Datei) und Variablenlisten (*.txt*-Datei) erfolgt über MATLAB-GUI-Fenster (siehe **Abbildung 4**). Dabei können mehrere *.mat*-Dateien ausgewählt werden. Dies ermöglicht den Vergleich von mehreren Simulationsdurchläufen. Die *.txt*-Dateien liefern die darzustellenden Variablen.

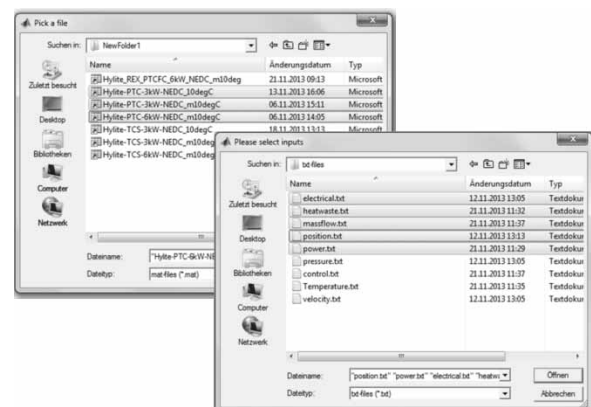


Abbildung 4. GUI-Fenster für die Auswahl der *.mat*- und *.txt*-Dateien

In der aktuellen Version sind zwei Darstellungsvarianten implementiert. Bei der ersten Variante werden

unterschiedliche Signale eines Simulationsdurchlaufs in einem Koordinatensystem dargestellt. Bei der zweiten Variante wird die gleiche Variable aus mehreren Simulationsdurchläufen abgebildet. Anschließend werden die Grafiken als *.fig*- und *.pdf*-Datei gespeichert.

Um die Verständlichkeit der Darstellungen zu erhöhen werden die SI-Einheiten des Modells wie z.B. Watt, m/s, m etc. in die in der Fahrzeugtechnik üblichen Einheiten kW, km/h und km automatisch umgerechnet. Die Achsenbeschriftungen setzen sich gewöhnlich aus der physikalischen Größe und ihrer Einheit zusammen (**Abbildung 5**). Oberhalb des Diagramms ist ein Header vorhanden, der Informationen über das simulierte Modell, die Simulationszeit, die Simulationsvariante und das Erstellungsdatum beinhaltet. Weitere Informationen, wie der Endwert einer Variablen (z.B. der Batterieladezustand (SOC)) können dem Header hinzugefügt werden.

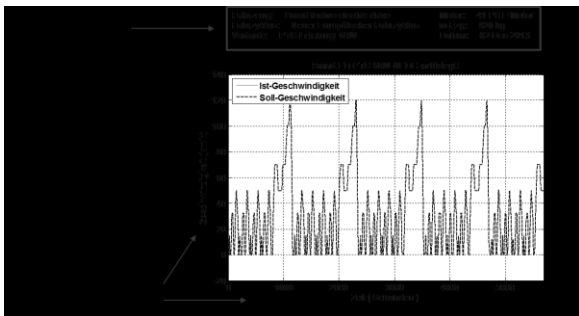


Abbildung 5. Layout der automatisch erstellten Abbildung

5 Anwendungsbeispiel: Automobiltechnik

In der Fahrzeugtechnik muss oft eine bestimmte Fahrzeugkonfiguration mit mehreren Fahrprofilen simuliert werden. Für den Austausch von Fahrprofilen reicht die im Kapitel 2 beschriebene automatisierte Parametervariation mit SARA nicht aus. Die im Kapitel 4 vorgestellte Erweiterung des SARA-Skriptsatzes erlaubt es, den Austausch von als Teilmodell implementierten Fahrzyklen automatisiert durchzuführen. Darüber hinaus wird die automatisierte Aufbereitung von Vergleichsdiagrammen ermöglicht.

Ein solches Vergleichsdiagramm ist in der **Abbildung 6** zu sehen. Hier wird ein Brennstoffzellenfahrzeug mit zwei Fahrprofilen untersucht. Dabei erfolgt der

Austausch der Fahrprofil-Instanz FTP 75 (Federal Test Procedure) durch NEDC (New european drive cycle) völlig automatisiert. Die Ergebnisse zweier Simulationsdurchläufe sind im Diagramm gegenüber gestellt. In diesem Beispiel wurde die Fahrzeuggeschwindigkeit für die Darstellung ausgewählt.

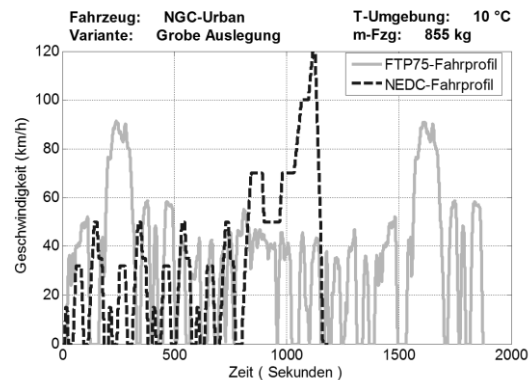


Abbildung 6. Vergleich der Fahrzeuggeschwindigkeit zweier Simulationen mit unterschiedlichen Fahrprofilen

In der Automobiltechnik werden oft Massen-, Leistungs- und Energiebilanzen gebildet. Die Summe aller Massen- und Energieflüsse eines Systems muss unter Berücksichtigung des Vorzeichens gleich null sein, wenn kein Speicher vorhanden ist. Wie aus der **Abbildung 7** ersichtlich, ist die Summe der Brennstoffzellen- und der Batterieleistung gleich der Leistung des Antriebsmotors. Mit dem erweiterten SARA-Skriptsatz können Flächendiagramme erzeugt werden, die der Darstellung solcher Bilanzen dienen.

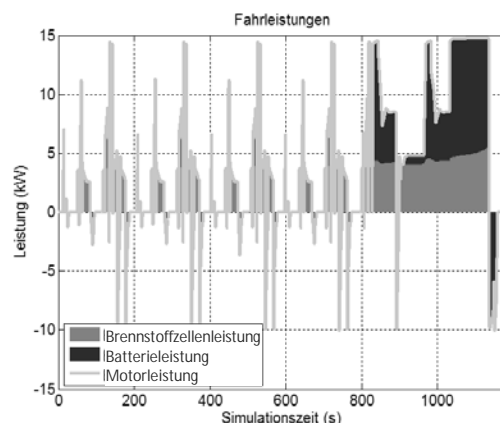


Abbildung 7. Elektrische Leistungen der Batterie, Brennstoffzelle und E-Maschine

6 Weitere Anwendungsmöglichkeiten von SARA

Die obigen Beispiele zeigen die vielseitigen Einsatzmöglichkeiten des SARA-Skriptsatzes. Weitere bereits realisierte Funktionen sind automatisierte Parameteroptimierungen mit dem Ziel, ein zuvor definiertes Ergebnis zu erreichen.

Bekanntlich fahren Schienenfahrzeuge nach einem Fahrplan. Eine Möglichkeit, energiesparend zu fahren ergibt sich, wenn man den Antrieb des Schienenfahrzeugs abschaltet und in den Ausrollzustand übergeht. Dafür ist es notwendig, die Position auf der Strecke zu detektieren, ab welcher man bei gegebener Fahrzeugmasse und -geschwindigkeit mit dem Ausrollen beginnen soll, damit die im Fahrplan definierte Fahrzeit zwischen zwei Bahnhöfen erreicht wird.

Mit dem Skriptsatz SARA lässt sich eine solche Bestimmung der Ausrollpunkte realisieren. Nach einem Simulationsdurchlauf wird überprüft, ob die vordefinierte Fahrzeit erreicht wurde. Ist es nicht der Fall, werden die Parameter automatisch mit dem Bisektionsverfahren angepasst und ein weiterer Simulationsdurchlauf gestartet. Dieses wird wiederholt, bis das Ziel ggf. mit einer vorgegebenen Abweichung erreicht wird. Bei diesem Beispiel basiert die Parameteranpassung auf automatisierter Parametervariation.

7 References

- [1] O. Beucher. *MATLAB und Simulink*. Pearson Studium, Deutschland, 2006.
- [2] A. Pfeiffer, I. Bausch-Gall und M. Otter. *Proposal for a Standard Time Series File Format in HDF5*. Proceedings of the 9th International Modelica Conference, in München, Deutschland, S. 56, 2012.
- [3] Dassault Systèmes AB. *Dymola User Manual*. Volume 2, 2013
- [4] XRG Simulation GmbH. *Human Comfort Library 1.2, User's Guide*, 2012
- [5] I. Windemut und H. Dittus. *Analyse des Leistungsbedarfes der HVAC-Anlagen im doppelstöckigen Wagenkasten des Next Generation Train*. Bahntechnik Aktuell, Band 47, S. 97-113, 2013.