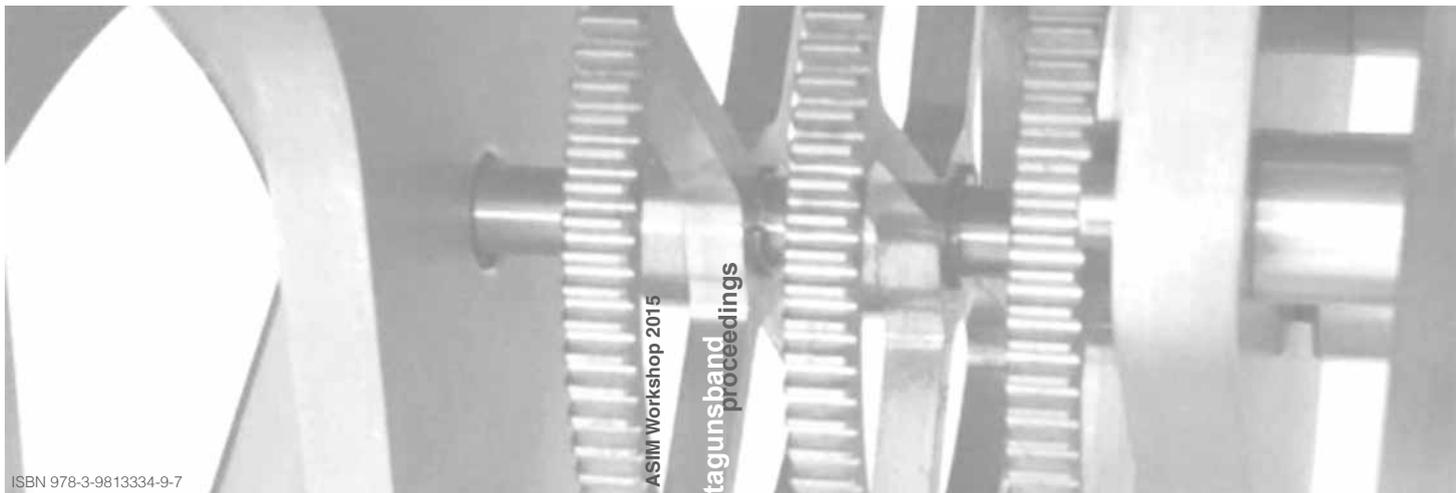


Organisiert durch:

GI (Gesellschaft für Informatik)
ASIM (Arbeitsgemeinschaft Simulation)
FG STS (Simulation technischer Systeme)
FG GMMS (Grundlagen und Methoden
in Modellbildung und Simulation)

Fachhochschule Stralsund
Fachbereich Maschinenbau



Fachhochschule Stralsund
Zur Schwedenschanze 15
D-18435 Stralsund
Zentrale
Fon +49 3831 455
Fax +49 3831 456690
www.fh-stralsund.de
info@fh-stralsund.de

ISBN 978-3-9813334-9-7

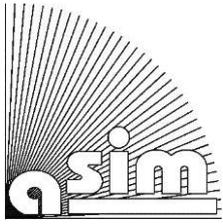
Workshop der ASIM/GI-Fachgruppen

Simulation technischer Systeme

Grundlagen und Methoden in
Modellbildung und Simulation

18./19.06.2015

praxis verstehen — chancen erkennen — zukunft gestalten
understanding reality — facing challenges — creating future



Tagungsband
Argesim Report AR 50

C. Wahmkow (Hrsg.)
P. Roßmanek (Hrsg.)
R. Wendorf (Hrsg.)

Workshop der ASIM/GI-Fachgruppen

Simulation technischer Systeme
Grundlagen und Methoden in Modellbildung und
Simulation

18. & 19. Juni 2015
Stralsund

organisiert durch:



Fachhochschule Stralsund
ASIM FG Simulation technischer Systeme
ASIM FG Grundlagen und Methoden in Modellbildung und Simulation

Zusammenfassung der Vorträge

Fachhochschule Stralsund, Stralsund, 2015
ISBN 978-3-9813334-9-7

Tagungsband zum Workshop der ASIM/GI-Fachgruppen

Published by **University of Applied Sciences Stralsund** and **ASIM**, Arbeitsgemeinschaft Simulation

Editing:

Claudia Rahn, Franz Jendersie, Arnold Lange (Fachhochschule Stralsund)
Zur Schwedenschanze 15, 18435 Stralsund
Tel: +49 3831 456528, Fax: +49 3831 456640
Email: Pressestelle@fh-stralsund.de

ARGESIM

ASIM – Mitteilungen AM 154

Titel: Workshop der ASIM/GI-Fachgruppen
Simulation technischer Systeme
Grundlagen und Methoden in Modellbildung und Simulation
Fachhochschule Stralsund 2015

Herausgeber: C. Wahmkow
P. Roßmanek
R. Wendorf
Email: Pressestelle@fh-stralsund.de

ISBN 978-3-9813334-9-7

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.

© Fachhochschule Stralsund, Stralsund 2015 / ARGESIM / ASIM, Wien, 2015

Fachhochschule Stralsund
Technologie- und Informationstransferstelle
Zur Schwedenschanze 15
18435 Stralsund
Tel: +49 3831 45628, Fax: +49 3831 456640,
Email: Pressestelle@fh-stralsund.de, Homepage: www.fh-stralsund.de

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

Tagungsprogramm und Inhaltsübersicht

Donnerstag 18.06.2015

10:00 - 11:15 Uhr

Tutorium I:

Einführung zu Modelica und Dymola

Leo Gall

LTX Simulation GmbH

Raum 215

Tutorium II:

Systemsimulation in der Leistungselektronik

- SiL-Regelung und MiL-Komponenten, FMI

- Umrichter Kühlung und EMV-Effekte (Kabel, Verschienung, Packages, PCB, Filter)

- Drahtloses Batterieladen

Olaf Hädrich

ANSYS Germany GmbH

Raum 219

11:15 - 11:30 Uhr

Ausstellung und Kaffeepause

➤ *Seite 1*

Aussteller

Foyer

11:30 - 11:45 Uhr

Eröffnung und Begrüßung

Raum 111

11:45 - 12:30 Uhr

Plenarvortrag I

➤ *Seite 3*

Computersimulation der Ausbreitung von Metastasen und ihrer Behandlung

Gero Wedemann

Fachhochschule Stralsund

Raum 111

12:30 - 13:30 Uhr

Ausstellung und Mittagspause

➤ *Seite 1*

Aussteller

Foyer

Modellbildung / Regelung I

Raum 104

Sitzungsleitung: Christine Wahmkow

- *Seite 17*
Modellbasierte Code-Generierung: Verwendung von Floating- Point-Code im Fixed-Point-Umfeld
Uwe Creutzburg
Fachhochschule Stralsund

- *Seite 23*
Entwicklung eines Programmsystems zur Modellierung von Biogasanlagen mittels neuronaler Netze
Maximilian Knappe
Fachhochschule Stralsund

- *Seite 31*
Modellierung und Parametrierung des respiratorischen Systems eines Patienten in der Atemtherapie
Mathias Scheel
Hochschule Wismar

- *Seite 37*
Modellbasierte Reglerauslegung eines sphärischen Elektroantriebs
Marian Göllner
Hochschule Ostfalia

Simulation von Steuerungen / Robotik

Raum 105

Sitzungsleitung: Thorsten Pawletta

- *Seite 49*
Objektorientierte Entwicklung einer hybriden Materialflusssimulation eines NE-Schmelz- und Druckgussbetriebes
Dominik Jeckle
Hochschule Ansbach

- *Seite 57*
MATLAB/Simulink Based Rapid Control Prototyping for Multivendor Robot Applications
Christina Deatcu
Hochschule Wismar

- *Seite 67*
Multi-Robotersteuerungen mit variablen Interaktionsprinzipien auf Basis des Simulation Based Control Frameworks und dem Discrete Event System Spezifikation Formalismus
Birger Freymann
Hochschule Wismar

- *Seite 79*
Anwendung einer iterativ lernenden Regelung auf einen Service Roboter mit Mehrkörperkinematik
Sandra Baßler
Hochschule Wismar

Maritime Anwendungen

Raum 116

Sitzungsleitung: Franz Jendersie

- *Seite 91*
Robuste Regelung zur Dynamischen Positionierung von Schiffen mit Voith Schneider Propellern
Charlotte Siebert
Universität Rostock

- *Seite 97*
Aufbereitung von AIS-Daten der WSV-N für die Simulation von Seeverkehrssystemen
Carsten Hilgenfeld
Hochschule Wismar

- *Seite 105*
Einsatz von Strömungssimulationen bei der Optimierung der Aerodynamik moderner Kreuzfahrtschiffe
Holm Stückrad
MET GmbH

Blended Learning

Raum 111

Sitzungsleitung: Andreas Körner

- *Seite 115*
Individuelle Einrichtung einer Online-Lernumgebung auf Basis eines Einstufungstests - Ein Werkstattbericht
Karin Landefeld
HAW Hamburg

- *Seite 117*
Mathematik-Kompetenzen überprüfen und fördern - Automatisiert Lehren und Lernen mit STACK
Michael Kallweit
Ruhr Universität Bochum

- *Seite 121*
Mathematiktests zwischen Schule und Hochschule
Thomas Schramm
Hafen Uni Hamburg

- *Seite 123*
Test ist nicht gleich online Test – Anforderungen und Umsetzungen von elektronischen Tests
Stefanie Winkler
TU Wien

15:30 - 16:00 Uhr

Ausstellung und Kaffeepause

Foyer

- *Seite 1*
Aussteller

Vortragsblock 2

16:00 - 18:00 Uhr

Physical Modeling / Modelica

Raum 104

Sitzungsleitung: Heinz-Theo Mammen

- *Seite 127*
Automatisierte, parallele Simulation von Modelica-Modellen am Beispiel des OpenModelica-Compilers
Volker Waurich
TU Dresden

- *Seite 135*
Strukturvariabilität für Modelica-Modelle
Daniel Gomez Esperon
TU Berlin

- *Seite 143*
Simulating a simple pneumatics network using the Modelica Fluid library
Peter Junglas
PHWT Vechta

- *Seite 151*
Modellierung elektrischer Antriebe mit Modelica
Leo Gall
LTX Simulation GmbH

Grundlagen und Methoden I
Sitzungsleitung: Christina Deatcu

Raum 105

- *Seite 163*
Zur Struktur von dynamischen Modellen für die Regelung von MIMO Prozessen
Bernhard Lampe
Universität Rostock
- *Seite 165*
Towards a Newer Toolbox for Computer Aided, Polynomial Design of
Sampled-Data Systems
Rudy Cepeda Gomez
Universität Rostock
- *Seite 171*
Hybrid State Estimation of nonlinear index-1 DAE-Systems
Dirk Weidemann
Fachhochschule Bielefeld
- *Seite 183*
Implementation of Hybrid Systems Described by DEV&DESS in the QSS
Based Simulator PowerDEVS
Franz Preyser
TU Wien

Adaptive Modellierung / Simulation
Sitzungsleitung: Christine Wahmkow

Raum 116

- *Seite 193*
Richtlinien zum Erstellen von strukturvariablen Modellen
Alexandra Mehlhase
TU Berlin
- *Seite 195*
Multi-Resolution Car2X-Simulation zur virtuellen Erprobung von
Fahrerassistenzsystemen
Manuel Schiller
TU München

- *Seite 197*
Potential of Dynamically Adaptable Simulation Models for Virtual Commissioning
Philipp Puntel Schmidt
TU Hamburg

- *Seite 205*
2Simulate Modeling Language –
Wiederbelebung und Modernisierung von alter Simulationssoftware
Jürgen Gotschlich
DLR

Engineering Anwendungen

Raum 111

Sitzungsleitung: Leander Marquardt

- *Seite 209*
Kopplung von Prozess- und CFD-Simulation zur Untersuchung des Wärmeverlusts beim Flüssigaluminiumtransport
Jörg Schmidt
Hochschule Ansbach

- *Seite 219*
Simulation eines Kontaktschalters mit COMSOL Multiphysics
Sonja Weinbrecht
COMSOL

- *Seite 225*
Spritzgussimulation mit SOLIDWORKS Plastics
Tobias Kamp
Solidline

- *Seite 233*
Vorausberechnung des Betriebs- und Emissionsverhaltens von kleinen luftgekühlten Industrie-Dieselmotoren
Leander Marquardt
Fachhochschule Stralsund

18:00 - 18:30 Uhr

Neues aus der ASIM

Raum 111

Felix Breitenecker
Heinz-Theo Mammen

Ab 19:30 Uhr

Abendveranstaltung

Eröffnung Abendessen ab 20:00 Uhr

Hotel Hafenresidenz Stralsund
Restaurant Fürst Wizlaw I. & Orangerie
Seestraße 10-13
18439 Stralsund

Freitag 19.06.2015

09:00 - 09:45 Uhr

Plenarvortrag II

Raum 111

➤ *Seite 243*

Simulationsgestützter Entwurf von ActiveNoiseControl-Systemen für
Kabinenanwendungen

Joachim Waßmuth

Fachhochschule Bielefeld

09:45 - 10:15 Uhr

Ausstellung und Kaffeepause

Foyer

➤ *Seite 1*

Aussteller

Vortragsblock 3

10:15 - 12:15 Uhr

Modellbildung / Regelung II

Raum 105

Sitzungsleitung: Arnold Lange

➤ *Seite 257*

Simulationsgestützte Bewertung kritischer Einflussgrößen auf das
Vorderradblockieren bei Elektrofahrrädern

Oliver Maier

Hochschule Pforzheim

➤ *Seite 269*

Ein Ansatz zur datenbasierten Reglerparametrierung

Alexander Schöley

Universität Rostock

- *Seite 275*
Anwendung Subspace basierter Identifikationsverfahren auf gesteuerte Mehrkörpersysteme
Mathias Marquardt
Hochschule Wismar

Grundlagen und Methoden II

Raum 116

Sitzungsleitung: Christine Wahmkow

- *Seite 285*
Domain-Specific Languages for Flexibly Experimenting With Models
Tom Warnke
Universität Rostock
- *Seite 287*
Towards a Common Description of Interdisciplinary Aspects Relevant for Holistic Energy Analysis of Production Facilities
Irene Hafner
dwh GmbH
- *Seite 289*
Design and Optimization of an Energy Manager for an Office Building
Kristin Majetta
Fraunhofer-Institut, Dresden

Elektrotechnische Systeme

Raum 111

Sitzungsleitung: Peter Roßmanek

- *Seite 299*
Modellbasierte Entwicklung des elektrischen Energiemanagements für Elektrofahrzeuge mit einem Zweispannungsbordnetz
Sören Scherler
Hochschule Ostfalia
- *Seite 309*
Effiziente Modellierung heterogener Batteriemagementsysteme (BMS)
Thomas Markwirth
Fraunhofer-Institut, Dresden
- *Seite 315*
Determination of Radiation Models Based on Measured Electromagnetic Field Distribution
Zongyi Chen
TU Dortmund

➤ *Seite 319*

Optimierung des Regelalgorithmus zur aktiven Fahrzeugfederung für
Elektrofahrzeuge

Matthias Fritsch

Hochschule Ostfalia

14:30 - 16:30 Uhr

Führung durch das Ozeaneum

OZEANEUM Stralsund

Hafenstraße 11

18439 Stralsund

Aussteller

ANSYS Germany GmbH
Birkenweg 14a
64295 Darmstadt

www.ansys.com/de_de



CENIT AG
Industriestraße 52-54
D-70565 Stuttgart

www.cenit.com/de_DE.html



Comsol Multiphysics GmbH
Robert-Gernhardt-Platz 1
37073 Göttingen

www.comsol.de/



LTX Simulation GmbH
Wohlfahrtstraße 21b
80939 München

www.ltx.de/



Plenarvortrag 1

Raum 111

Sitzungsleitung: Christine Wahmkow

Computersimulation der Ausbreitung von Metastasen und ihrer Behandlung

Gero Wedemann

Fachhochschule Stralsund

Computersimulation der Ausbreitung von Metastasen und ihrer Behandlung

Bertin Hoffmann¹, Anja Bethge¹, Udo Schumacher², Gero Wedemann¹

¹Competence Center Bioinformatics, Fachhochschule Stralsund

²Institut für Anatomie und Experimentelle Morphologie, Universitätsklinikum Hamburg-Eppendorf
gero.wedemann@fh-stralsund.de

Bei Krebserkrankungen kann der primäre Tumor Tochtergeschwulste, sog. Metastasen, im Körper streuen. In den Fällen, bei denen die Krebserkrankung zum Tod führt, sind meistens nicht der Primärtumor, sondern diese Metastasen die Todesursache. Trotz intensiver Forschung sind viele Details des Prozesses der Metastasierung nicht verstanden. An der Fachhochschule Stralsund wurde ein Computermodell entwickelt, mit dem der Prozess der Metastasierung quantitativ untersucht werden kann. Es wurden verschiedene Modelle des Metastasierungsprozesses mit klinischen und experimentellen Daten verglichen, um neue Erkenntnisse zu gewinnen. Das Modell basiert auf dem Prinzip der diskreten Ereignissimulation. Das Wachstum von Primärtumor und Metastasen wird durch analytische Funktionen beschrieben, während das Abwandern von Tumorzellen aus dem Tumor in die Blutbahn durch sogenannte Intravasations-Ereignisse simuliert wird. Weitere Ereignisse simulieren das Verhalten dieser Zellen, bis sie entweder zugrunde gehen oder neue Metastasen bilden. Anhand von klinischen und experimentellen Daten wurde mit Hilfe des Computermodells untersucht, ob Metastasen früh oder spät im Verlauf der Tumorerkrankung gestreut werden, ob Metastasen selbst auch metastasieren können und welchen Einfluss das Immunsystem auf den Prozess der Metastasierung hat. Weiterhin lassen sich verschiedene Therapien wie die Resektion des Primärtumors, Chemotherapie oder Bestrahlung und deren Einfluss auf die Anzahl der Metastasen und die gesamte Tumorlast simulieren.

1 Einleitung

Im Jahr 2013 erlagen in Deutschland 223842 Personen einem Krebsleiden. Somit ist Krebs die zweithäufigste Todesursache in Deutschland [1]. Jedoch ist nicht der Primärtumor, sondern die Metastasen in über 90% der Fälle die Todesursache bei Krebserkrankungen. Für die Entwicklung neuer Behandlungsmethoden ist es daher von großer Wichtigkeit, den Prozess der Metastasierung zu verstehen.

Der komplexe Prozess der Metastasierung umfasst mehrere aufeinanderfolgende Schritte und wird z.B. durch das umliegende Gewebe oder das Immunsystem beeinflusst. Nachdem der primäre Tumor eine gewisse Größe erreicht hat, muss er die Neubildung von Blutgefäßen mit Hilfe sog. angiogener Signale anregen, um die ausreichende Versorgung mit Nährstoffen weiterhin sicher zu stellen. Anschließend beginnen die Zellen mit der Absiedlung aus dem primären Tumorgeschwulst in die Blutbahnen (Intravasation). Über die Blutgefäße gelangen die malignen Zellen in den Blutkreislauf und somit in andere Organe, wie z.B. Lunge oder Leber. Bei dem Transport über die Blutbahn werden die Zellen durch immunologische Abwehrreaktionen angegriffen und zu

99,9% vernichtet [2]. Die Zellen verlassen die Blutgefäße und treten am Zielort in das umliegende Gewebe ein (Extravasation). Diese Zellen werden zu diesem Zeitpunkt als disseminierte Tumorzellen bezeichnet und bilden ab einer bestimmten Größe eine Mikrometastase. Überlebt diese Mikrometastase, kann sie selbst angiogene Signale induzieren, um zu einer Makrometastase anzuwachsen [2,3]. Da jeder vorherige Schritt abgeschlossen sein muss, bevor der nächste eingeleitet werden kann, wird der Prozess der Metastasierung auch als metastatische Kaskade bezeichnet.

Der Prozess der Metastasierung ist Gegenstand einer Vielzahl experimenteller und klinischer Forschungsarbeiten. Um die Ergebnisse besser zu analysieren, wurden mathematische Modelle entwickelt, die im Vergleich mit klinischen und experimentellen Daten helfen sollen, den Prozess der Metastasierung quantitativ und qualitativ besser zu verstehen. Je nach Fragestellung konzentrieren sich diese Modelle auf bestimmte Details, wie z.B. die molekularen Eigenschaften einer Tumorzelle oder aber den Prozess des Tumorwachstums. Der Einsatz dieser Modelle ermöglicht beispielsweise die Ermittlung der optimalen

Dosis für Bestrahlungstherapien [4] oder die Entwicklung besserer Behandlungsstrategien [5–7].

Der Einsatzbereich solcher mathematischer Modelle ist allerdings beschränkt [8]. Als Ergänzung können Computersimulationen eingesetzt werden, die es im Gegensatz zu mathematischen Modellen ermöglichen, Varianz auf Basis von Zufallsvariablen zu simulieren.

In diesem Artikel wird ein Computermodell vorgestellt, welches an der FH Stralsund zur Analyse von Daten und der daraus folgenden Simulation von alternativen Behandlungen eingesetzt wird. Die bisherigen Ergebnisse [8–12] werden zusammengefasst.

2 Methoden

2.1 Kompartimente und Ereignisse

Ein zu simulierendes System besteht aus einer Menge von Objekten. Diese Objekte besitzen verschiedene Eigenschaften und können untereinander in Beziehung stehen [13]. So steht z.B. das Objekt Primärtumor mit dem Objekt Blutbahn in Beziehung, da die Zellen aus dem Primärtumor in die Blutbahn intravasieren können. Beide Objekte besitzen in diesem Fall die Eigenschaft „Anzahl der Zellen“. Zustandsänderungen am System, z.B. Änderungen der Eigenschaften oder Beziehungen von Objekten, werden als Ereignisse bewirkt. Das entwickelte Computermodell setzt sich hauptsächlich aus den Bausteinen Kompartimenten sowie Ereignissen zusammen (siehe Abbildung 1).

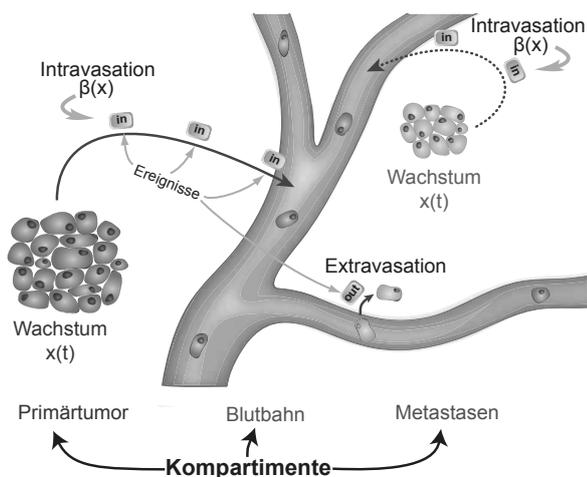


Abbildung 1: Schematische Darstellung von Kompartimenten und Ereignissen. Abbildung modifiziert nach [10].

Die Kompartimente stellen die Objekte im Computermodell dar. Dazu zählen der Primärtumor, die Blutbahn wie auch die Metastasen. Kompartimente werden entweder kontinuierlich oder diskret modelliert. Bei der kontinuierlichen Modellierung werden die Vorgänge mittels mathematischer Funktionen modelliert: Eine Wachstumsfunktion beschreibt das Wachstum des Kompartiments während eine Kolonisationsrate das Streuen von malignen Zellen in das nachfolgende Kompartiment repräsentiert. In diskreten Kompartimenten werden die Vorgänge durch Ereignisse modelliert. Dazu kann für jedes diskrete Kompartiment eine Liste mit Progressionsereignissen konfiguriert werden, die in dem betreffenden Kompartiment auftreten können. Für jedes Ereignis in dieser Liste wird ebenfalls eine Wahrscheinlichkeit definiert, mit der dieses im Vergleich zu den anderen Ereignissen in der Liste in diesem Kompartiment auftritt. Wenn ein neues Ereignis für das Kompartiment erzeugt wird, so wird aus dieser Liste entsprechend der konfigurierten Wahrscheinlichkeiten ein Ereignis ausgewählt.

Ereignisse beschreiben, welches Verhalten in den Kompartimenten zu einem bestimmten Zeitpunkt ausgelöst wird [9,10]. Diese werden in einer Ereignisliste gespeichert und der Reihe nach entsprechend dem Zeitpunkt, an dem ein Ereignis eintritt, ausgeführt. Ein oder mehrere Kompartimente werden durch die Ausführung eines Ereignisses verändert. Es werden auch neue Ereignisse durch das Ausführen eines Ereignisses generiert. Dies geschieht zum Beispiel nach dem Ereignis „Intravasation“, in dem ein neues Ereignis erzeugt wird, welches beschreibt, was als nächstes mit dieser Tumorzelle in der Blutbahn passiert.

Bei den Ereignissen wird zwischen lokalen und globalen Ereignissen unterschieden. Lokale Ereignisse können nur den Zustand genau eines Kompartimentes verändern, während globale Ereignisse den Zustand ein oder mehrerer Kompartimente verändern können.

Mit Hilfe der diskreten Ereignissimulation kann der Zustand des Systems an bestimmten Zeitpunkten modifiziert werden. So kann die Anzahl der Tumorzellen im Primärtumor durch Zellteilung, Zelltod (Apoptose) oder Intravasation einer Tumorzelle in die Blutbahn zu einem bestimmten Zeitpunkt geändert werden, indem die Anzahl der Zellen im Primärtumor entsprechend erhöht oder verringert wird. So erhöht z.B. das Ereignis „Zellteilung“ die Anzahl der Zellen, im entsprechenden Kompartiment um eins. Anschlie-

End werden zwei neue Ereignisse generiert, welche beschreiben, was als nächstes mit der Mutterzelle und den beiden Tochterzellen passiert, z.B. „Apoptose“ und „Intravasation“. Das Ereignis „Apoptose“ verringert die Anzahl der Zellen im Kompartiment ebenfalls um eins. Das Ereignis „Intravasation“ verringert die Anzahl der Zellen im betreffenden Kompartiment um eins und erhöht anschließend die Anzahl der Zellen im Folgekompartiment (hier die Blutbahn) um eins. Anschließend wird wieder ein neues Ereignis erzeugt, das beschreibt, was als nächstes mit der Tumorzelle in der Blutbahn passiert, z.B. ein Apoptose-Ereignis, welches die Anzahl der Zellen im Blutbahn-Kompartiment um eins verringert. Um den genauen Zeitpunkt, an dem das Ereignis ausgeführt wird, zu berechnen wird eine Verteilung sowie eine Funktion genutzt. Die Berechnung erfolgt für jedes Ereignis separat.

Bei dem entwickelten Computermodell kommt zurzeit für den Primärtumor und die Metastasen die kontinuierliche Modellierung zum Einsatz. Bei der Blutbahn hingegen wird die diskrete Modellierung eingesetzt.

2.2 Modellierung des Tumorwachstums in kontinuierlichen Kompartimenten

Die Anzahl der Zellen x in einem kontinuierlichen Kompartiment verändert sich zeitlich um die Rate $g(x)$:

$$(1) \quad \frac{dx}{dt} = g(x), \quad x(0) = N_0$$

Bei einem Patienten und dessen Krankheitsverlauf beginnt das Tumorwachstum mit $N_0 = 1$, also einer einzelnen entarteten Tumorzelle [14]. Die Anzahl der Zellen zum Zeitpunkt $t = 0$ kann jedoch bei Experimenten mit Modellorganismen auch größer sein.

Für die Wachstumsrate $g(x)$ können verschiedene Funktionen eingesetzt werden. Dazu zählen z.B. lineare, exponentielle, logarithmische oder Potenzfunktionen. In den meisten Fällen tritt jedoch eine Gompertz-Wachstumsrate auf [14]. Diese ist definiert als $g(x) = a x \ln(b/x)$ und beschreibt einen s-förmigen Wachstumsverlauf. Wobei der Parameter a für die Wachstumskonstante und der Parameter b für die maximale Tumorgroße steht [8,10].

2.3 Modellierung der Intravasation in kontinuierlichen Kompartimenten

Im entwickelten Computermodell umfasst das Ereignis „Intravasation“ alle Schritte, die für das Abwandern der Zelle vom Tumor in die Blutbahn durchlaufen werden. Die Intravasation beginnt zunächst damit, dass die Tumorzellen die Basalmembran durchbrechen und anschließend in das umliegende Bindegewebe eindringen. Vereinzelt Zellen können sich aus dem Gewebe lösen und dringen in die Blutbahnen ein. Intravasations-Ereignisse werden mit Hilfe der Kolonisationsrate $\beta(x)$ erzeugt, welche als $\beta(x) = m x^\delta$ definiert ist, wobei m die Kolonisationskonstante und δ die fraktale Dimension, also der Grad an Durchblutung im Tumor, beschreibt [14].

Der nächste Zeitpunkt, an dem ein neues Intravasations-Ereignis ausgeführt wird, wird mit Hilfe einer numerischen Integration der Kolonisationsrate beginnend vom Zeitpunkt des letzten Intravasations-Ereignisses berechnet. Es wird dazu eine Zufallszahl zwischen 0 und 2 bestimmt. Erreicht die Berechnung des Flächeninhalts unter der Kurve diesen Wert, wird dieser Zeitpunkt für das neue Intravasations-Ereignis festgelegt [9].

2.4 Modellierung von therapeutischen Behandlungen

Das Computermodell bietet die Möglichkeit verschiedene Therapien bzw. Behandlungen zu simulieren. Dazu gehören die Resektion des Primärtumors, Chemotherapie, Bestrahlung, Radioimmuntherapie sowie Radioembolisation. Am Beispiel der Resektion des Primärtumors wird die Modellierung genauer erläutert. Die anderen Behandlungsoptionen sind in [11] detailliert beschrieben.

Bei der Simulation einer Resektion des Primärtumors wird in diskreten Kompartimenten die Anzahl der Zellen auf 0 gesetzt. In der Ereignisliste noch vorhandene Ereignisse, welche dem Primärtumor zuzuordnen sind, werden gelöscht. In kontinuierlichen Kompartimenten wird die Wachstumsfunktion des Tumors durch die Funktion $x(t) = 0$ ersetzt. Somit ergibt auch die Kolonisationsrate $\beta(x) = 0$, so dass keine neuen Intravasations-Ereignisse ausgehend vom behandelten Primärtumor mehr erzeugt werden [11].

2.5 Modellierung von Ruhephasen

Nach der Extravasation der Tumorzelle aus der Blutbahn in das umliegende Gewebe, können die Zellen zunächst in einer Ruhephase (engl. dormancy) ver-

harren. In dieser Phase ändert sich die Anzahl der Zellen im betreffenden Kompartiment nicht. Eine Ruhephase kann aber auch erst nach einigen Zellteilungszyklen eintreten, nachdem aus einer einzelnen disseminierten Tumorzelle ein mehrzelliges Cluster geworden ist. Ursachen, für diese spätere Ruhephase könnte z.B. der Beginn von Angiogenese oder die Umstellung der Genexpression der Zellen der Metastase sein. In diesem Fall wachsen die Zellen somit zunächst bis zu einem gewissen Zeitpunkt und verfallen erst dann für eine gewisse Zeit in die Ruhephase.

Durch spezifische Ereignisse werden die entsprechend diskreten Kompartimente in eine Ruhephase versetzt und generieren in diesem Zeitraum keine weiteren Ereignisse was das Tumorwachstum betrifft. Nach Beendigung der Ruhephase wird ein Ereignis ausgelöst, welches den Zustand der Ruhephase rückgängig macht. Anschließend wächst dieses entsprechend seiner Parametrisierung weiter.

Bei kontinuierlichen Kompartimenten wird die aktuell gültige Wachstumsfunktion für den Zeitraum der Ruhephase durch eine konstante Funktion ersetzt. Nach Beendigung der Ruhephase beginnt das Wachstum. Bei einer späteren Ruhephase wird nach Beendigung der Ruhephase der Startwert N_0 der verwendeten Wachstumsfunktion auf die Anzahl der Zellen vor der Ruhephase gesetzt.

2.6 Simulation

Die Konfiguration der Simulation wird über XML (Extensible Markup Language [16]) Dateien vorgenommen. Das zu simulierende Szenario lässt sich bausteinartig zusammensetzen um die Objekte (Primärtumor, Blutbahn und Metastasen) und deren Eigenschaften (z.B. Wachstumsrate) sowie anzuwendende Behandlungen zu definieren. Die gesamte Struktur dieser XML-Datei ist durch ein XML Schema vorgegeben. Die Beschreibung des Schemas sowie verschiedene Beispiele stehen unter [15] zur Verfügung.

Beim Start der Simulation wird die XML-Datei von der Simulationssoftware eingelesen und alle benötigten Kompartimente sowie bereits bekannte Ereignisse (z.B. Behandlungen am Tag x) erstellt. Der Zustand des simulierenden Systems wird während der Simulation in definierten Zeitabständen (XML-Datei) zwischengespeichert und am Ende der Simulation in eine EXCEL-Datei geschrieben. In der exportierten Datei werden unter anderem die aktuelle Zeit, die Größe des Primärtumors sowie die Anzahl der Metastasen

und die Anzahl der Zellen in allen Metastasen gespeichert [11].

Da die Generierung der Ereignisse auf Zufallszahlen basiert (siehe Abschnitt 2.3) wird bei einzelnen Simulationsdurchläufen nie das gleiche Endergebnis ermittelt. Daher wird ein Szenario typischerweise 100-mal simuliert, um eine ausreichende Varianz der Daten zu erhalten. Anschließend wird der Mittelwert und die Standardabweichung aller Simulationsdurchläufe berechnet [15].

2.7 Konfiguration der Simulation

Im Folgenden wird die Konfiguration eines zu simulierenden Szenarios anhand eines gekürzten Beispiels vorgestellt: die Resektion des Primärtumors und die Chemotherapie (siehe Listing 1).

```

1 <functions>
2   <gompertzFunction functionID="growthrate">
3     <growthRate>0.00286</growthRate>
4     <tumorSize>7.3e10</tumorSize>
5     <startSize>1</startSize>
6   </gompertzFunction>
7   [...]
8   <colonizationRate functionID="colrate">
9     <colonizationCoefficient>5.3e-8
10    </colonizationCoefficient>
11    <fractalDim>0.663</fractalDim>
12  </colonizationRate>
13 </functions>
14 [...]
15 <hostOrgans>
16   <hostOrgan organID="Liver">
17     <name>Liver</name>
18   </hostOrgan>
19   <hostOrgan organID="blood_vessels">
20     <name>blood</name>
21   </hostOrgan>
22 </hostOrgans>
23 [...]
24 <compartments>
25   <continuousCompartment compartmentID="p_tumor">
26     <hostOrgan organRef="Liver"/>
27     <successor compartmentRef="blood"/>
28     <isPrimaryTumor>true</isPrimaryTumor>
29     <resectionTime>1100</resectionTime>
30     <name>primary_tumor</name>
31     <growthFunction functionRef="growthrate"/>
32     <colonizationRate functionRef="colrate"/>
33   </continuousCompartment>
34   <discreteCompartment compartmentID="blood">
35     <hostOrgan organRef="blood_vessels"/>
36     <discreteCompartmentType typeRef="blood_Type"/>
37   </discreteCompartment>
38 </compartments>
39 <globalTreatments>
40   <chemotherapy>
41     <time>850</time>
42     <quantity>3</quantity>
43     <interval>50</interval>
44     <cellingCellFraction>0.5</cellingCellFraction>
45     <drugSensitivity>1</drugSensitivity>
46     <drugDecay>1</drugDecay>
47   </chemotherapy>
48 </globalTreatments>

```

Listing 1: Ausschnitt der verwendeten XML-Konfigurationsdatei. Im oberen Abschnitt wird die Wachstumsfunktion des Tumors (Zeile 2: *growthrate*) sowie dessen Streurrate (Zeile 8: *colrate*) angegeben. Die Chemotherapie wird in 3 Zyklen im Abstand von 50 Tagen ab Tag 850 durchgeführt. Am Tag 1100 findet eine Resektion des Primärtumors statt.

Wie in Listing 1 dargestellt, wird die Chemotherapie am Tag 850 gestartet (Zeile: 42 bis 49). Dabei wird die Therapie in drei Zyklen im Abstand von 50 Tagen durchgeführt. An Tag 1100 wird der Primärtumor durch eine Resektion entfernt (Zeile: 29). Der Verlauf der Größe des Primärtumors und Anzahl der Zellen in allen Metastasen im Verlauf der Simulation wird in Abbildung 2 bzw. Abbildung 3 dargestellt.

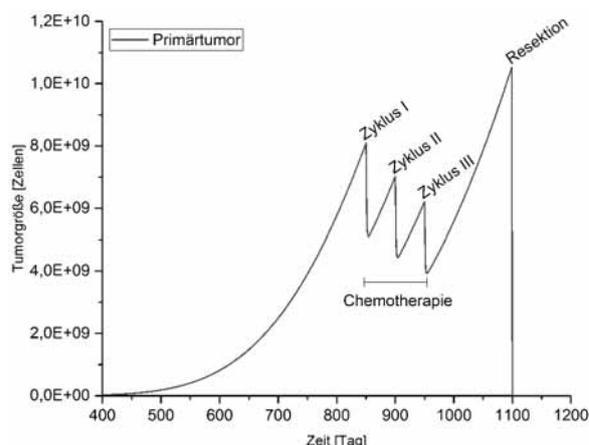


Abbildung 2: Auswirkung von Chemotherapie und Resektion des Primärtumors auf die Primärtumorgroße. Am Tag 850, 900 und 950 wurde jeweils die Chemotherapie angewendet. Die Resektion des Primärtumors erfolgte an Tag 1100.

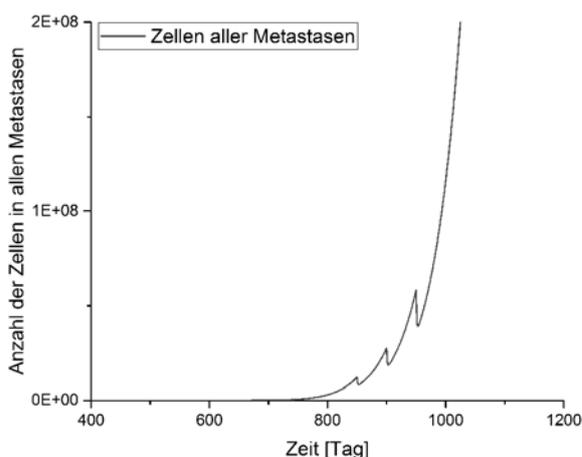


Abbildung 3: Auswirkung von Chemotherapie auf die gesamte Metastasenlast. In der Abbildung ist gut zu erkennen, wie die Chemotherapie an den Tagen 850, 900 und 950 die Anzahl der Zellen in den Metastasen reduziert. Jedoch ist die Chemotherapie nicht in der Lage alle Metastasen zu eliminieren, so dass die Anzahl der Zellen nach Beendigung der Behandlung weiter steigt.

Wie in Abbildung 2 zu erkennen ist, hat die Chemotherapie nur kurzzeitige Auswirkungen auf die Größe des Primärtumors. Erst die Resektion des Primärtumors verringert die Größe des Tumors auf den Wert 0. Auf die Anzahl der Metastasen hat die Chemotherapie ebenfalls nur einen kurzfristigen Einfluss (siehe Abbildung 3). Weitere Informationen zu der XML-Konfiguration sind in [15] zu finden.

3 Klinische Relevanz von spät gestreuten Metastasen

3.1 Fragestellung

Anhand von klinischen Daten eines unbehandelten Patienten mit hepatozellulärem Karzinom (HCC) und multiplen Metastasen in der Leber [14] wurden, unter anderem, folgende Fragestellungen untersucht:

1. Sind Metastasen selbst in der Lage Metastasen zu streuen?
2. Sind Zellen, die von sehr großen Tumoren gestreut wurden noch in der Lage Metastasen zu bilden?

3.2 Simulierte Szenarien

Um die oben genannten Fragestellungen zu untersuchen, wurden vier verschiedene Szenarien simuliert (siehe Tabelle 1). In Szenario A können sowohl der Primärtumor, wie auch die Metastasen neue Metastasen streuen. In Szenario B ist nur der Primärtumor dazu in der Lage. Szenario C und D untersuchen hingegen, ob spät gestreute Tumorzellen großer Tumore oder Metastasen noch in der Lage sind, Metastasen zu bilden. Für die Simulation wurden drei Werte in Bezug auf die Maximalgröße b (siehe Abschnitt 2.2) für die Größe des Tumors, ab dem die gestreuten Zellen keine Metastasen mehr bilden, gewählt: 10^8 , 10^9 und 10^{10} Zellen. In Szenario D sind, wie auch in Szenario B, Metastasen nicht in der Lage selbst Metastasen zu streuen.

Tabelle 1: Simulierte Szenarien [10].

Spät gestreute Tumorzellen...	Metastasen können metastasieren	Metastasen können nicht metastasieren
.. können Metastasen bilden	A	B
... können keine Metastasen bilden	C	D

3.3 Ergebnisse

Die Ergebnisse sind in Abbildung 4 dargestellt. Die Abbildung zeigt die aufsummierte Anzahl der Metastasen im Verhältnis zur Größe der Metastasen. In Abbildung 4 ist gut zu erkennen, dass sich die Szenarien A und B im Bereich der klinischen Daten (Kreise, Quadrate und Dreiecke), welche an den Tagen 1110, 1237 und 1310 nach der Diagnose des Primärtumors mittels CT-Scans ermittelt wurden, nicht unterscheiden. Erst im Bereich der kleinen Metastasen ($< 10^6$ Zellen) ist ein deutlicher Unterschied bei Tag 1310 erkennbar. Dies liegt daran, dass die Metastasen erst eine gewisse Größe erreichen müssen, um selbst Metastasen bilden zu können. Somit sind in den CT-Scans anscheinend nur die Metastasen des Primärtumors erkennbar gewesen. Erst im späteren Verlauf (Tag 1310) könnten Metastasen von Metastasen gestreut worden sein.

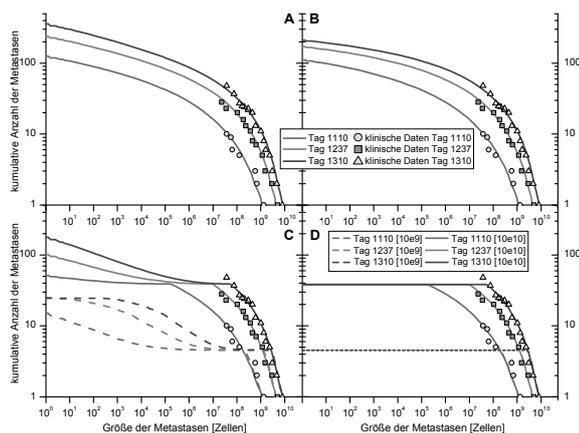


Abbildung 4: Ergebnisse der Simulationen für die Szenarien A-D [10]. Auf der vertikalen Achse ist die kumulative Anzahl der Metastasen und auf der horizontalen die Größe der Metastasen aufgetragen. Beide Achsen sind logarithmisch dargestellt.

Anhand der vorliegenden Daten kann somit nicht mit Sicherheit eine Aussage getroffen werden, ob Metastasen selbst Metastasen bilden können. Für die Beantwortung der Frage sind Daten aus detaillierteren Untersuchungen notwendig. Es zeigt sich allerdings, dass für diese Szenarios die Fragestellung nicht von klinischer Relevanz ist, da bereits die vom Primärtumor gestreuten Metastasen zur tödlichen Tumorlast beitragen.

Um die Frage, ob große Tumore oder Metastasen noch Zellen streuen, welche in der Lage sind Metastasen zu bilden, zu beantworten, wurden die Szenarien C und D untersucht. Dabei wurden verschiedene Größen als Grenzwert gewählt, ab dem die vom Pri-

märtumor oder Metastasen gestreuten Zellen nicht mehr in der Lage sind Metastasen zu bilden. Bei einem Grenzwert von 10^9 Zellen ist zu erkennen, dass die durchgeführten Simulationen nicht mit den klinischen Daten (Kreise, Quadrate, Dreiecke) übereinstimmen (gestrichelte Linien in C und D). Dementsprechend sind Zellen, die von einem Primärtumor oder einer Metastase größer als 10^9 Zellen gestreut wurden, dazu befähigt Metastasen zu bilden. Bei 10^{10} Zellen liegen die klinischen Daten mit den Daten der Simulationsergebnisse nah beieinander. Somit sind Zellen, welche vom Primärtumor oder Metastasen größer 10^{10} gestreut wurden, nicht mehr in der Lage dazu, Metastasen zu bilden. Die klinischen Daten decken jedoch nicht den gesamten Größenbereich der Metastasen ab. Daher kann auch diese Fragestellung nicht eindeutig beantwortet werden [8,10]. Es zeigt sich allerdings auch hier, dass für diese Szenarios die Fragestellung nicht von klinischer Relevanz ist.

4 Progression der Metastasierung unter Einfluss von therapeutischen Behandlungen

4.1 Fragestellung

Mit Hilfe des Computermodells wurde der Einfluss von verschiedenen Behandlungen auf die Progression der Metastasierung untersucht. Dabei wurde ebenfalls untersucht, welche Relevanz eine frühzeitige Diagnose auf das Therapieergebnis hat.

4.2 Simulierte Szenarien

Die Untersuchungen wurden anhand der bereits in Abschnitt 5 vorgestellten klinischen Daten eines Patienten mit HCC durchgeführt. In [15] wurden folgende Behandlungen simuliert: Die Resektion des Primärtumors, phasenspezifische und phasenunabhängige Chemotherapie, Bestrahlung des Primärtumors, Radioimmuntherapie, Radioembolisation und eine Kombination aus Resektion des Primärtumors sowie einer phasenunabhängigen Chemotherapie. Die komplette Parametrisierung der untersuchten Szenarien ist in Tabelle 1 in [15] zu finden. In allen Szenarien wurde der Behandlungsbeginn für Tag 700 (frühzeitige Diagnose) und für Tag 1500 (späte Diagnose) simuliert. Zusätzlich wurden die Fälle betrachtet, ob Metastasen die Fähigkeit besitzen, selbst Metastasen bilden zu können oder nicht.

4.3 Ergebnisse

Auszugsweise wird am Beispiel der Resektion des Primärtumors der Einfluss auf die Metastasierung erläutert.

Bei einer frühzeitigen Diagnose des Primärtumors und der damit verbundenen frühzeitigen Behandlung (Resektion), zeigt sich eine zeitweise Stagnation in der Entstehung neuer Metastasen (Abbildung 5A) bzw. eine komplette Stagnation (Abbildung 5B).

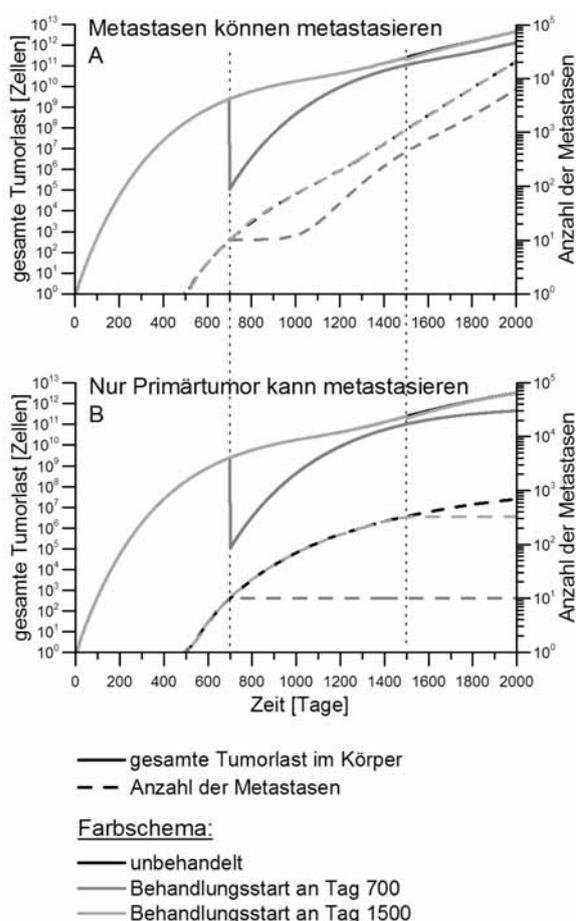


Abbildung 5: Ergebnisse der Resektion des Primärtumors am Tag 700. In Szenario A streuen der Primärtumor sowie die Metastasen. In Szenario B streut nur der Primärtumor Metastasen. Abbildung modifiziert nach: [8].

In dem Fall, dass Metastasen metastasieren können, hat eine späte Behandlung des Tumors (Tag 1500) kaum noch Einfluss auf die Anzahl der Metastasen. Die Anzahl der Metastasen stagniert hingegen in dem Fall, dass nur der Primärtumor Metastasen streuen kann. Die Behandlung zu einem späten Zeitpunkt hat dennoch keinen relevanten Einfluss mehr in Bezug

auf die gesamte Tumorklast (siehe Abbildung 5B) [8,11].

Die Ergebnisse verdeutlichen die Wichtigkeit einer frühzeitigen Diagnose und dem damit zusammenhängenden frühzeitigen Therapiestart. Von großer Wichtigkeit ist ebenfalls eine weiterführende Behandlung, da ansonsten der Vorteil einer vorzeitigen Behandlung verloren geht [11].

5 Einfluss von natürlichen Killerzellen auf den Prozess der Metastasierung

5.1 Fragestellung

Das Immunsystem besitzt verschiedene Abwehrmechanismen, um Tumorzellen im Körper frühzeitig zu erkennen und zu eliminieren. Zu den wichtigsten Bestandteilen im Kampf gegen Tumorzellen zählen die sogenannten natürlichen Killerzellen (NK-Zellen). Diese sind in der Lage Tumorzellen zu eliminieren [17–19]. Wird eine NK-Zelle aktiv, setzt diese Granula frei, welche verschiedene toxische Proteine beinhalten und somit zum Absterben (Apoptose) der Tumorzelle führen. Eine Vielzahl an Studien zeigte, dass eine verringerte Aktivität der NK-Zellen den Metastasierungsprozess erleichtern [20–23], während bei einer hohen Aktivität der Prozess verlangsamt wurde [24–26].

Mit Hilfe des Computermodells sollte ermittelt werden, auf welchem Abschnitt des Metastasierungsprozesses die NK-Zellen den größten Einfluss haben. Dazu wurde dessen Einfluss auf das Wachstum des Primärtumors haben, das Überleben der Tumorzellen im Blut sowie auf das Entstehen von Metastasen untersucht [8,12].

5.2 Simulierte Szenarien

Zu diesem Zweck wurden Daten aus Mausmodellen analysiert. Mäusen der Gruppe rag2 fehlt das „recombination-activating gene 2“ (rag2) was zur Fehlfunktion der B- und T-Lymphozyten führt. Dies bedeutet, dass keine Abwehrreaktion durch diese Zellen stattfinden. Mäuse der Gruppe pfp/rag2 besaßen weder rag2 noch das Perforin-Gen (pfp), was zu einer Abschaltung wichtiger NK-Zellen Aktivität führt. Beiden Mäusegruppen wurden als Primärtumor 10^6 Zellen einer Darmkrebs-Zelllinie (HT29) injiziert. Nach Beendigung des Experiments wurde der Primärtumor vermessen und die Anzahl der Metastasen in der Lunge ermittelt. Die ermittelten Werte wurden mit Simulationen nachgestellt und analysiert. Bei beiden

Mausgruppen wurde eine Ruhephase und bei der Gruppe pfp/rag2 zusätzlich eine späte Ruhephase in der Lunge simuliert. Die Ruhephase wird direkt nach der Extravation der Tumorzellen ausgeführt und führt somit zum Stillstand des Wachstums. Erst nach der Beendigung der Ruhephase tritt das Wachstum dieser Zellen ein. Die späte Ruhephase tritt ein, wenn eine Metastase in der Lunge eine zufällige Größe zwischen 10 und 100 Zellen erreicht hat, verbleibt dann für eine gewisse Zeit in der Ruhephase und wächst anschließend gemäß ihrer parametrisierten Eigenschaften weiter.

5.3 Ergebnisse

Die Ergebnisse der Computersimulationen zeigten, dass die NK-Zellen das Wachstum des Primärtumors verlangsamen. Zusätzlich wurden 80% der Zellen in der Blutbahn eliminiert, welche ohne NK-Zellen-Aktivität in der Lage gewesen wären Metastasen zu bilden. NK-Zellen erschweren zusätzlich den disseminierten Tumorzellen die Ansiedlung in der Lunge für mindestens 30 Tage. Diese Zellen befinden sich in diesem Zeitraum in einer Ruhephase [8,12]. Die experimentellen Daten der Gruppe pfp/rag2 konnten mit zwei unterschiedlich langen Ruhephasen direkt nach der Extravasation nicht nachgebildet werden. Nur mit Hilfe einer späten Ruhephase konnten sie nachgebildet werden. Eine mögliche Erklärung für das Auftreten einer späten Ruhephase könnte sein, dass die Metastasen nach einigen Zellteilungszyklen an Nährstoffmangel leiden und somit zunächst angiogene Signale auslösen müssen. Weitere Untersuchungen sind an dieser Stelle notwendig, um die beobachtete späte Ruhephase zu erklären [8,12].

6 Diskussion

Als Ergänzung zu experimenteller Forschung und klinischen Studien stellen Computersimulationen eine wertvolle Möglichkeit dar, den Prozess der Metastasierung quantitativ zu untersuchen. Das entwickelte Computermodell bietet mit seinem bausteinartigen Aufbau die Möglichkeit, durch Ereignisse und Kompartimente verschiedenste Szenarien abzubilden. Dazu zählen unter anderem verschiedene Beschreibungen zum Tumorwachstum, Streuraten von Metastasen sowie der Einfluss von Behandlungen. Das Modell bietet die Möglichkeit, klinische oder experimentelle Daten noch detaillierter zu analysieren und verschiedene Modellannahmen quantitativ zu überprüfen. Auf Grundlage dieser Modelle können dann

Vorhersagen über Wachstumsverläufe getroffen werden.

Die aktuelle Version des Computermodells modelliert einen idealen Verlauf der Behandlungen und berücksichtigt dabei keinerlei Resistenzen gegen die eingesetzten Behandlungsmethoden. Weiterhin wird die Auswirkung von einer Strahlentherapie auf das umliegende gesunde Gewebe nicht beachtet. Die Tumore bzw. Metastasen können aktuell nicht in einem bestimmten Bereich eines Organs angegeben werden. Beispielsweise kann somit nicht nur ein Teil eines Organs bestrahlt werden. Auch wurden die Behandlungsmethoden, Chemo- und Strahlentherapie, noch nicht mit klinischen Daten validiert [11]. In aktuellen Arbeiten werden die Behandlungsmethoden mit einer Bandbreite an klinischen und experimentellen Daten weiterentwickelt.

Im Vergleich zu anderen existierenden Modellen (z.B. [14] und [27]), erlaubt unser Computermodell die Anwendung von unterschiedlichem Wachstums- und Streuverhalten von Primärtumor und Metastasen. Ebenfalls lassen sich Metastasen mit unterschiedlichen Eigenschaften (z.B. Metastasen in Lunge oder Leber) simulieren. Zusätzlich stehen die Behandlungen Resektion des Primärtumors, Chemotherapie, Bestrahlung, Radioimmuntherapie sowie Radioembolisation zur Verfügung. Kein existierendes Modell bietet einen derartigen Funktionsumfang und kann eine Kombination von Resektion, Chemotherapie [28–32] und Bestrahlung [6,33–35] modellieren [11].

7 Literaturverzeichnis

- [1] Statistisches Bundesamt. Fachserie 12 Reihe 4: Todesursachen in Deutschland 2013. Wiesbaden: Statistisches Bundesamt; 2014.
- [2] Talmadge JE, Fidler IJ. AACR centennial series: the biology of cancer metastasis: historical perspective. *Cancer Res* 2010;70:5649–69. doi:10.1158/0008-5472.CAN-10-1040.
- [3] Weinberg RA. *The Biology of Cancer*. Garland Science; 2006.
- [4] Bernhardt P, Ahlman H, Forssell-Aronsson E. Model of metastatic growth valuable for radionuclide therapy. *Med Phys* 2003;30:3227. doi:10.1118/1.1628851.
- [5] Leder K, Ramakrishnan J. Minimizing metastatic risk in radiotherapy fractionation schedules. *ArXiv13127337 Phys Q-Bio* 2013.
- [6] Leder K, Pitter K, LaPlant Q, Hambarzumyan D, Ross BD, Chan TA, et al. Mathematical modeling of PDGF-driven glioblastoma reveals

- optimized radiation dosing schedules. *Cell* 2014;156:603–16. doi:10.1016/j.cell.2013.12.029.
- [7] Benzekry S, Barbolosi D, Benabdallah A, Hubert F, Hahnfeldt P. Quantitative Analysis of the Tumor/Metastasis System and its Optimal Therapeutic Control 2013.
- [8] Bethge A. Computersimulation der metastatischen Progression und therapeutischer Interventionen. Dissertation. Universität Rostock, in Vorbereitung.
- [9] Wedemann G, Bethge A, Haustein V, Schumacher U. Computer simulation of the metastatic progression. *Metastasis Res. Protoc.* 2nd ed., New York: Springer; 2014, p. 107–16.
- [10] Bethge A, Schumacher U, Wree A, Wedemann G. Are metastases from metastases clinical relevant? Computer modelling of cancer spread in a case of hepatocellular carcinoma. *PLoS ONE* 2012;7:e35689. doi:10.1371/journal.pone.0035689.
- [11] Bethge A, Schumacher U, Wedemann G. Simulation of Metastatic Progression using a Computer Model including Chemotherapy and Radiation Therapy, in Revision.
- [12] Brodbeck T, Nehmann N, Bethge A, Wedemann G, Schumacher U. Perforin-dependent direct cytotoxicity in natural killer cells induces considerable knockdown of spontaneous lung metastases and computer modelling-proven tumor cell dormancy in a HT29 human colon cancer xenograft mouse model. *Mol Cancer* 2014;13:244. doi:10.1186/1476-4598-13-244.
- [13] Ulrich Hedtstück. Simulation diskreter Prozesse - Methoden und Anwendungen. Auflage 1. Berlin, Heidelberg: Springer; 2013.
- [14] Iwata K, Kawasaki K, Shigesada N. A dynamical model for the growth and size distribution of multiple metastatic tumors. *J Theor Biol* 2000;203:177–86. doi:10.1006/jtbi.2000.1075.
- [15] Bethge A, Wedemann G. CaTSiT - Computer simulation of metastatic progression and treatments. *CaTSiT - Comput Simul Metastatic Progress Treat* n.d. <http://bioinformatics.fh-stralsund.de/catsit/> (accessed May 11, 2015).
- [16] World Wide Web Consortium. Extensible Markup Language (XML) 1.0 (Fifth Edition) 2008.
- [17] Biron CA. Activation and function of natural killer cell responses during viral infections. *Curr Opin Immunol* 1997;9:24–34.
- [18] Trinchieri G. Biology of natural killer cells. *Adv Immunol* 1989;47:187–376.
- [19] Arina A, Murillo O, Dubrot J, Azpilikueta A, Alfaro C, Pérez-Gracia JL, et al. Cellular liaisons of natural killer lymphocytes in immunology and immunotherapy of cancer. *Expert Opin Biol Ther* 2007;7:599–615. doi:10.1517/14712598.7.5.599.
- [20] Ben-Eliyahu S, Yirmiya R, Liebeskind JC, Taylor AN, Gale RP. Stress increases metastatic spread of a mammary tumor in rats: evidence for mediation by the immune system. *Brain Behav Immun* 1991;5:193–205.
- [21] Gorelik E, Wiltrout RH, Okumura K, Habu S, Herberman RB. Role of NK cells in the control of metastatic spread and growth of tumor cells in mice. *Int J Cancer J Int Cancer* 1982;30:107–12.
- [22] Nüssler NC, Stange BJ, Petzold M, Nüssler AK, Glanemann M, Guckelberger O. Reduced NK-Cell Activity in Patients with Metastatic Colon Cancer. *EXCLI J* 2007;6:1–9.
- [23] Saijo N, Ozaki A, Beppu Y, Takahashi K, Fujita J, Sasaki Y, et al. Analysis of metastatic spread and growth of tumor cells in mice with depressed natural killer activity by anti-asialo GM1 antibody or anticancer agents. *J Cancer Res Clin Oncol* 1984;107:157–63.
- [24] Coca S, Perez-Piqueras J, Martinez D, Colmenarejo A, Saez MA, Vallejo C, et al. The prognostic significance of intratumoral natural killer cells in patients with colorectal carcinoma. *Cancer* 1997;79:2320–8.
- [25] Liljefors M, Nilsson B, Hjelm Skog A-L, Ragnhammar P, Mellstedt H, Frödin J-E. Natural killer (NK) cell function is a strong prognostic factor in colorectal carcinoma patients treated with the monoclonal antibody 17-1A. *Int J Cancer J Int Cancer* 2003;105:717–23. doi:10.1002/ijc.11139.
- [26] Ménard C, Blay J-Y, Borg C, Michiels S, Ghiringhelli F, Robert C, et al. Natural killer cell IFN-gamma levels predict long-term survival with imatinib mesylate therapy in gastrointestinal stromal tumor-bearing patients. *Cancer Res* 2009;69:3563–9. doi:10.1158/0008-5472.CAN-08-3807.
- [27] Benzekry S, Gandolfi A, Hahnfeldt P. Global dormancy of metastases due to systemic inhibition of angiogenesis. *PLoS ONE* 2014;9:e84249. doi:10.1371/journal.pone.0084249.
- [28] Birkhead BG, Rankin EM, Gallivan S, Dones L, Rubens RD. A mathematical model of the development of drug resistance to cancer chemotherapy. *Eur J Cancer Clin Oncol* 1987;23:1421–7. doi:10.1016/0277-5379(87)90133-7.
- [29] Panetta JC. A mathematical model of periodically pulsed chemotherapy: Tumor recurrence and metastasis in a competitive environment. *Bull Math Biol* 1996;58:425–47. doi:10.1007/BF02460591.
- [30] Panetta JC, Adam J. A mathematical model of cycle-specific chemotherapy. *Math Comput Model* 1995;22:67–82. doi:10.1016/0895-7177(95)00112-F.
- [31] Dua P, Dua V, Pistikopoulos EN. Optimal delivery of chemotherapeutic agents in cancer. *Comput Chem Eng* 2008;32:99–107. doi:10.1016/j.compchemeng.2007.07.001.
- [32] Benzekry S, Chapuisat G, Ciccolini J, Er-

linger A, Hubert F. A new mathematical model for optimizing the combination between antiangiogenic and cytotoxic drugs in oncology. *Comptes Rendus Math* 2012;350:23–8.

doi:10.1016/j.crma.2011.11.019.

[33] Wang P, Feng Y. A mathematical model of tumor volume changes during radiotherapy. *ScientificWorldJournal* 2013;2013:181070.

doi:10.1155/2013/181070.

[34] Wheldon TE. Radiotherapy: treatment strategies. *Math. Models Cancer Res.*, Bristol: Adam Hilger; 1988, p. 134–56.

[35] Bernhardt P, Speer TW. Modeling of the systemic cure with targeted radionuclide therapy. In: Speer TW, editor. *Target. Radionucl. Ther.*, Lippincott Williams and Wilkins; 2010, p. 263–81.

Sitzung 1

Raum 104

Modellbildung / Regelung I

(Sitzungsleitung: Christine Wahmkow)

*Modellbasierte Code-Generierung: Verwendung von Floating- Point-Code im
Fixed-Point Umfeld*

Uwe Creutzburg

Fachhochschule Stralsund

*Entwicklung eines Programmsystems zur Modellierung von
Biogasanlagen mittels neuronaler Netze*

Maximilian Knappe

Fachhochschule Stralsund

*Modellierung und Parametrierung des respiratorischen
Systems eines Patienten in der Atemtherapie*

Mathias Scheel

Hochschule Wismar

Modellbasierte Reglerauslegung eines sphärischen Elektroantriebs

Marian Göllner

Hochschule Ostfalia

Modellbasierte Code-Generierung: Verwendung von Floating-Point-Code im Fixed-Point-Umfeld

Dr. Ing. U. Creutzburg
Fachhochschule Stralsund
uwe.creutzburg@fh-stralsund.de

Die Entwicklung von Software für die Automobilindustrie in Form von Fixed-Point-Code (FXP) wird durch gängige Production-Code-Generatoren zur modellbasierten Entwicklung seit einigen Jahren in ausgereifter Weise unterstützt. Als Grund für den Einsatz der FXP-Arithmetik wird dabei allgemein angeführt, dass die Ausführungszeit gegenüber Floating-Point-Code (FLP) deutlich geringer ausfällt.

Inzwischen steigt die Komplexität der verwendeten Algorithmen. Diese neuen Algorithmen lassen sich oft nur schwer mittels gängiger Techniken als FXP-Code bei ausreichender Rechengenauigkeit abbilden. Gleichzeitig nimmt die Verbreitung preiswerter Controller mit integrierter Floating-Point-Unit zu.

Auf Basis eines gängigen Filter-Algorithmus werden in der hier präsentierten Untersuchung unterschiedliche Szenarien zur Verwendung von FLP-Code in FXP-Umgebungen untersucht. Es wird anhand typischer Implementierungs-Varianten dargestellt, dass in der Praxis der automatisch generierte FLP-Code hinsichtlich Laufzeit, Ressourcenverbrauch, Testbarkeit, Wartbarkeit und Entwicklungsaufwand bessere Eigenschaften aufweist, als derzeit mit FXP-Code zu erreichen ist.

1 Ausgangssituation

Mikrocontroller mit integrierter Floating-Point-Unit (FPU) finden sich im Portfolio gängiger Halbleiterhersteller. War die Verwendung der FPU in der Vergangenheit exklusiven Steuergeräten vorbehalten, migrieren diese Bausteine inzwischen in die Breite.

Nimmt man an, dass der überwiegende Anteil der Software auf einem Steuergerät auch in Zukunft als FXP-Code abgebildet wird (z. B. Zustandsmaschinen, Diagnose, Datenhandling auf Bussystemen), stellt sich die Frage, inwieweit sich der Einsatz von FLP-Code in einem FXP-Umfeld lohnt. Dieser Fragestellung wird im Folgenden nachgegangen.

1.1 Testumgebung

Die Testumgebung besteht aus einer Werkzeugkette zur modellbasierten Software-Entwicklung (MATLAB/Simulink®; R2007B+) mit integriertem Code-Generator (TargetLink®; Version 3.1). Das untersuchte Target ist ein Renesas® V850E2/PX4, der mittels TargetLink-TSM (Green Hills, MULTI®; 5.1.6C) in einer Processor-In-the-Loop (PIL) Konfiguration betrieben wurde.

1.2 Testobjekt

Als Testobjekt wird ein IIR-Filter zweiter Ordnung verwendet. Dieses Filter ist als Second-Order-Section (SOS) implementiert. Es enthält fünf abstimmbare Filterkoeffizienten, die üblicherweise im ROM abgelegt sind.

Das Interface zum FXP-Code bildet jeweils das mit 16-Bit quantisierte Ein-/Ausgangssignal des Filters.

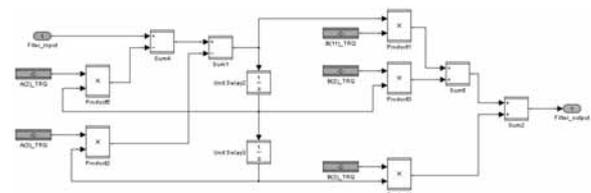


Abbildung 1. Testobjekt SOS-Filter

Im vorliegenden Fall wird das Testobjekt vom Code-Generator als gekapselte Funktion generiert.

Startpunkt für die weitere Analyse ist die FXP-Variante des Filters. Die internen Größen sind so skaliert (Wertigkeit des Least-Significant-Bit), dass mit den per Lastenheft vereinbarten Eingangssignalen keine internen/externen Übersteuerungen auftreten und die erreichbare Dynamik möglichst groß ist.

2 Implementierungs-Varianten

2.1 FXP-Variante

Bei der FXP-Modellierung wurde beachtet, dass nach Möglichkeit keine laufzeitintensiven Umskalierungen (Shifts) im automatisch generierten C-Code der Filterfunktion stattfinden. Die internen Zwischengrößen (Zustände, temporäre Variablen) des Filters sind daher durchgängig als 32-Bit-FXP-Datentypen mit einheitlicher Skalierung implementiert.

Der Entwicklungsaufwand zur Bestimmung der optimalen Skalierung ist erheblich, da Auswirkungen von Übersteuerungen/Überläufen dramatisch sein können. Die erreichbare Dynamik des Filters, insbesondere das Verhalten bei kleinen Signalpegeln, steht dabei in Kontrast zum Verhalten bei großen Signalpegeln. In diesem Zusammenhang sind auch die zulässigen Wertebereiche der ggf. abstimmbaren Parameter zu berücksichtigen.

Um die Auswirkungen von in der Praxis nicht abschließbaren Übersteuerungen (Over-/Underflow) in akzeptable Regionen zu bringen, können die mathematischen Operationen zusätzlich per Schalter in der Modellierungsphase pro Block „saturiert“ werden.

```

1 Sa2_Product4 =
  C_I32SHRI64C6_LT32_SATb( ... );
2 Sa2_Sum2 = Sa2_Sum5 + Sa2_Product4;
3 if ((Sa2_Sum5 >= 0) && ... && ... )
4 {
5   Sa2_Sum2 = 2147483647;
6 }
7 else
8 ...
9 Sa1_OutPort = (sint16) (Sa2_Sum2 >> 6);

```

Code 1. FXP-Code-Fragment

Der Code-Generator setzt diese Saturierungen über zwei Methoden um:

- 1) Einsatz von speziellen Makros, die 64-Bit-Operationen saturiert abbilden: in Zeile 1 führt „C_I32SHRI64C6_LT32_SATb“ eine 64-Bit-Operation mit anschließender Sättigung des 32-Bit-Ergebnisses aus [2].
- 2) Compute-Through-Overflow (Zeile 2) mit anschließender Plausibilisierung der Vorzeichen von Operanden und Ergebnis (Zeile 3) und ggf. Korrektur durch Sättigung (Zeile 5).

In der Regel wird der Entwickler „sicherheitshalber“ für jeden der Blöcke die Saturierung einschalten:

dadurch wird der Code weiter aufgebläht, die Komplexität steigt.

Optimierungen des Code-Generators sorgen zwar optional dafür, dass überflüssige Saturierungen „optimiert“ und damit aus dem Code entfernt werden. Erfahrungsgemäß bleiben trotzdem einige konditionale Zweige bestehen, die in Modultests nicht erreichbar sind oder auch laut optionaler PolySpace®-Analyse tatsächlich „Dead Code“ darstellen.

Durch die komplexe Code-Struktur steigt insgesamt der Aufwand zur Erstellung von Testfällen, insbesondere wenn ein hoher Wert für die Code-Coverage erforderlich ist.

Die Verwendung der zu TargetLink mitgelieferten Bibliothek zur Unterstützung von 64-Bit-Arithmetik und der Einsatz der oben genannten Saturierungen, führen darüber hinaus dazu, dass die Ausführungszeit des Systems nicht mehr konstant ist (s. **Abbildung 2**), die Laufzeit schwankt im konkreten Fall zwischen ca. 4.4µs und 4.8µs.

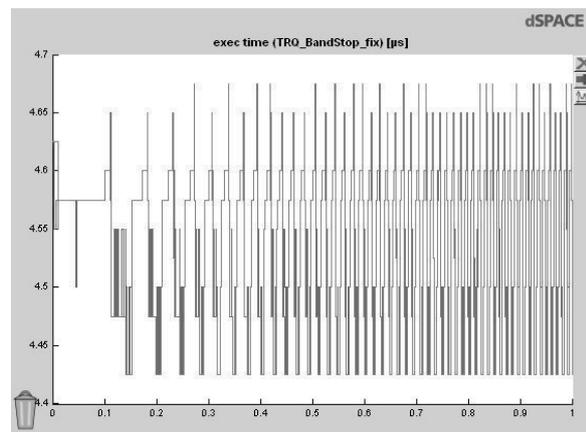


Abbildung 2. Laufzeitverhalten der FXP-Variante

Für die weiteren Diskussionen wird hier angenommen, dass der ermittelte Wert von 4.8µs tatsächlich die „worst-case execution time“ (WCET) ist.

Insgesamt ergeben sich für die hier diskutierte FXP-Variante die in der **Tabelle 1** zusammengefassten Ergebnisse: Die WCET des Filters liegt unter 5µs, der Code belegt auf dem Target 562 Byte im ROM und 60 Byte im RAM (Stack).

Variante	CPU [µs]	ROM [Byte]	RAM [Byte]	Maint
FXP	< 5	562	60	--

Tabelle 1. Eigenschaften der FXP-Variante

Die Maintenance-Eigenschaften (Maint) dieser Variante werden aufgrund der aufwändigen, fehleranfälligen Modellierung, Code-Größe und der Komplexität als sehr schlecht bewertet.

Hinweis: Aktuelle Varianten des verwendeten Green Hills-Compilers enthalten ein spezielles „Idiom-Recognition-Modul“, das die direkte Unterstützung saturierter FXP-Arithmetik des Targets ermöglicht. Die generierten Code-Fragmente der o. g. TargetLink Makro-Bibliothek werden von diesem Modul analysiert und, soweit möglich, durch Aufrufe der entsprechenden saturierten Arithmetik-Befehle des V850E2-Kerns (SATADD, ...) ersetzt. Dieses Feature wird hier nicht verwendet [3].

2.2 FLP-Variante Single

Diese Variante verwendet das 32-Bit Single-Precision-Format nach ANSI/IEEE Standard 754-1985 [4]. Der Filter-Algorithmus nutzt die FPU; die Umgebung des Filters bleibt im FXP-Format. Die Abstimmparameter (Filterkonstanten) sind weiterhin als 16-Bit Signed-Integer codiert.

Der generierte Code enthält keine Makro-Aufrufe zur Sättigung, ebenso sind konditionale Code-Anteile zur Umsetzung von Saturierungen nicht notwendig.

Der dargestellte Ausschnitt des generierten Codes zeigt aber, dass die als Single-Precision-Daten abgelegten Abstimmparameter pro Zeitschritt von FXP nach FLP gewandelt und die Skalierung von Stored-Integer nach Real-World Value konvertiert wird ($6.10E-5 = 2^{-14}$).

```
1 Sc1_OutPort = (sint16) (((Sc2_Sum1 *
  ((float32) VEH_Struct.B11_AI16)
  * 6.103515625e-005F) +
  ((float32) VEH_Struct.B2_AI16)
  * 6.103515625e-005F ...));
```

Code 2. Ausschnitt FLP-Code Single

Da keine konditionalen Anteile im Code vorhanden sind, ist die angegebene Ausführungszeit tatsächlich die WCET. Der Code läuft etwa um den Faktor 4 schneller als die FXP-Variante.

Variante	CPU [µs]	ROM [Byte]	RAM [Byte]	Maint
FLP-Single	1.1	124	16	++

Tabelle 2. Eigenschaften der FLP-Variante Single

Der C-Code ist sehr viel besser lesbar und erheblich kürzer als der FXP-Code. Im Ergebnis ergibt sich ein deutlich kleinerer „Footprint“ im ROM und RAM.

Die Wartbarkeit des Modells ist besser als bei der FXP-Variante. Der Umfang des „Hidden-Layer“ aus verborgenen, nicht visuellen Block-Parametern (Skalierung, Sättigung, Bit-Anzahl) ist stark reduziert.

Die Testbarkeit des Codes wird verbessert und die Code-Coverage wird nicht durch implizit generierte Fragmente reduziert, sie beträgt bei diesem Filter für jeden beliebigen Testvektor 100%.

2.3 FLP-Variante Single Optimiert

Eine weitere Verbesserung kann erreicht werden, wenn zusätzlich die Filterkoeffizienten im Single-Precision-Format vorliegen. Es entfallen dann die im vorigen Code-Fragment notwendigen Typ-Konvertierungen und Umskalierungen.

```
1 Se1_OutPort = (sint16) (((Se2_Sum1 *
  VEH_Struct.B11_f32) +
  (VEH_Struct.B2_f32 * Se2_Unit_Delay2) + ...
```

Code 3. Ausschnitt FLP-Code Single Optimiert

Gegenüber der nicht optimierten Variante ergibt sich eine weitere Reduktion der benötigten Ressourcen Laufzeit (-38%) und ROM-Bedarf (-31%).

Variante	CPU [µs]	ROM [Byte]	RAM [Byte]	Maint
FLP-Single	0.68	86	16	++

Tabelle 3. Eigenschaften FLP-Variante Single Opt.

Die Wartbarkeit ist weiter verbessert, da keine expliziten Vorgaben für die Skalierungen der Filter-Parameter notwendig sind. Die Stabilität des Filters wird durch die größere Genauigkeit der FLP-Parameter erhöht.

2.4 FLP-Variante Double

Diese Variante verwendet das 64-Bit Double-Precision-Format nach ANSI/IEEE Standard 754-1985. Die hier aufgeführten Ergebnisse sind analog zur Diskussion in Abschnitt 2.2 zu bewerten. Interessanterweise schwankt bei dieser Variante aber die ermittelte Laufzeit um einige Prozentpunkte.

Variante	CPU [µs]	ROM [Byte]	RAM [Byte]	Maint
FLP-Single	<1.48	166	32	++

Tabelle 4. Eigenschaften der FLP-Variante Double

2.5 FLP-Varianten Emulation

Um einen Eindruck zu erhalten, ob sich der Einsatz von FLP-Code lohnt, der als *FLP-Emulation* auf dem FXP-Kern läuft, sind hier die beiden zu Abschnitt 2.2 und 2.3 analogen Versionen aufgeführt.

Variante	CPU [µs]	ROM [Byte]	RAM [Byte]	Maint
FLP-Single Emulation	< 18µs	234 Lib.	16 Lib.	++ -
FLP-Single Emulation Optimiert	< 11µs	154 Lib.	16 Lib.	++ -

Tabelle 5. Eigenschaften der FLP-Variante Emulation

Vorteile dieser Varianten sind die guten Eigenschaften hinsichtlich Maintenance und eventuell, dass die FPU für andere Tasks exklusiv zur Verfügung steht.

Der ROM-Bedarf ist gegenüber den Varianten mit Verwendung der FPU deutlich erhöht, da eine entsprechende Emulations-Bibliothek verwendet wird.

Die Laufzeit ist immerhin, zumindest für die optimierte Variante, nur gut doppelt so hoch, wie die der FXP-Variante. Es könnte also durchaus sinnvoll sein, einige FLP-Anteile emulieren zu lassen.

3 Fazit

Die in der **Tabelle 6** zusammengefassten, auf die FXP-Variante normierten, Ergebnisse zeigen deutlich das Potential der vorgeschlagenen optimierten Modellierung als FLP-Code. In der optimierten Variante ist der Single-Precision-Code gut sieben Mal schneller als die FXP-Variante.

Vom Entwicklungsaufwand her ist die FLP-Modellierung zudem deutlich schlanker, robuster und vermeidet an kritischen Stellen den negativen Einfluss des „Hidden-Layer“. Die Fehleranfälligkeit der Modellierung sinkt, die Komplexität des generierten Codes nimmt ab, entsprechend verbessert sich die Testbarkeit.

Aus diesen Gründen könnte selbst die FLP-Emulation eine Daseinsberechtigung haben.

Variante	CPU [%]	ROM [%]	RAM [%]	Maint
FXP	100	100	100	- -
FLP-Single Emulation	360	42 Lib.	27 Lib.	++ -
FLP-Single Emulation Optimiert	220	28 Lib.	27 Lib.	++ -
FLP-Double	30	30	54	++
FLP-Single	22	22	27	++
FLP-Single opt.	14	16	27	++

Tabelle 6. Vergleich benötigter Ressourcen

4 Ausblick

Die präsentierten Ergebnisse hängen sicherlich stark vom verwendeten Target ab, sind in ihrer Tendenz aber aussagekräftig.

Die konkrete Verwendung der FPU auf dem Steuergerät kann als Ressource exklusiv einem bestimmten Teil der Applikation vorbehalten sein oder der gesamten Software zur Verfügung stehen. Störungen durch Task-Wechsel sind in der vorliegenden Untersuchung nicht berücksichtigt, das gilt auch für Störungen der Pipeline [5]. Die hier aufgeführte „Worst-Case Execution Time“ könnte in der Praxis also überschritten werden.

Die Eigenschaften des verwendeten Compilers könnten in die konkreten Untersuchungen einbezogen werden. Features wie das genannte „Idiom-Recognition“ der Saturierungs-Makros könnten sich auf die Ergebnisse auswirken.

Ebenso könnte der Einsatz spezieller TargetLink Target-Optimization-Modules (TOM) die ermittelten WCET beeinflussen.

Anmerkungen zum Schluss: Die angegebenen Ausführungszeiten ändern sich bei Verwendung mehrerer SOS-Stufen annähernd proportional zur Stufenanzahl. Die erforderliche Signaldynamik erzwang im Beispiel die Verwendung von 64-Bit-Operationen, diese sind zwangsläufig aufwändig im Hinblick auf die Ausführungszeit.

5 References

- [1] dSPACE, *TargetLink Production Code Generation Guide*, TargetLink 3.1 dSPACE GmbH, Deutschland, 2009
- [2] dSPACE, *TargetLink Bibliothek dsfxp.h*, TargetLink 3.1 dSPACE GmbH, Deutschland, 2009
- [3] Green Hills, Diskussion zu Compiler-Optimierungen, 2013.
- [4] A. S. Tanenbaum. *Computerarchitektur*. Pearson Studium, Deutschland, 2006, S. 732ff.
- [5] J. J. Labrosse. *μ C/OS-III*. Micrium Press, USA, 2010, S. 167ff.

Entwicklung eines Programmsystems zur Modellierung von Biogasanlagen mittels neuronaler Netze

Christine Wahmkow¹, Franz Jendersie¹, Maximilian Knappe¹

¹FH-Stralsund, Fachbereich Maschinenbau

Christine.Wahmkow@fh-stralsund.de

Betrachtet man die Entwicklung des EEG über die Jahre, so ist der stetig wachsende Druck zur Flexibilisierung von Biogasanlagen unübersehbar. Diesem Druck sind Betreiber neuer Anlagen unausweichlich ausgesetzt, aber auch für bestehende Anlagen ist das subventionierte „Geldverdienen“ eine Frage der Zeit. Neben rein technischen Möglichkeiten, wie dem Errichten zusätzlicher Gasspeicher, die Aufbereitung von Biogas in Erdgasqualität oder die verfahrenstechnische Beeinflussung des Gärprozesses sind softwaretechnische Lösungen die preislich interessantesten Alternativen. Will man ein System informationstechnisch modellieren, benötigt man im besten Fall die funktionale Beschreibung der ablaufenden Prozesse. Die Mikrobiologie in einer Biogasanlage ist aber noch zu einem großen Teil ungeklärt bzw. nur durch sehr aufwändige Messtechnik erfassbar. Gut ausgebildete Anlagenbediener mit einem reichen Erfahrungsschatz verstehen es aber trotzdem, eine Anlage behutsam und dynamisch zu fahren. Die Theorie des Einsatzes Künstlicher Neuronaler Netze (KNN) geht davon aus, dass sich das Verhalten eines Systems in den Messdaten widerspiegelt. Dieses Verhalten wird in einem Netz vieler einzelner Knoten mit gewichteten Verbindungen beschrieben. Hat man ein solches Netz gefunden, kann es auf neue Situationen reagieren, sich dynamisch anpassen und auf gelernte Bedingungen mit sinnvollen Aktionen reagieren. Ein Programmsystem zum selbständigen Finden eines solchen Netzes wurde entwickelt. Ein darauf aufbauender Regler, der ausgehend von einem bestimmten Gasbedarf in der Zukunft über die aktuelle Fütterung den Gasertrag prognostiziert, befindet sich im Versuchsstadium auf einer industriellen Anlage.

1 Der Weg zur Flexibilisierung

Das EEG trat im Jahr 2000 in Kraft und hat Novellierungen in 2004, 2009, 2012 und 2014 erhalten. Aufgabe des EEG's am Anfang war es, den Markteintritt für Strom aus erneuerbaren Energien durch Förderung mit festen Vergütungssätzen (Einspeisevergütung) zu schaffen. Mit dem sogenannten Maisdeckel von 60% hat man mit dem EEG 2012 der Gefahr der Monokulturen Einhalt geboten. Die Direktvermarktung wurde als Alternative zur Einspeisevergütung aufgenommen und mit Einführung der Marktprämie ein Anreiz für dieses Modell geschaffen.

Sieht man sich das EEG 2014 an, so wird die Direktvermarktung von Strom aus Biogasanlagen der Regelfall sein. Man kann erkennen, dass die Einspeisevergütung nur noch für kleine Anlagen und Ausnahmefälle gewährt wird und damit ungewollt ist. Sieht man den gesamten Mix erneuerbarer Energien, so liegt der Grund dafür klar auf der Hand mit der Möglichkeit der Bereitstellung von Regelenergie aus Biogasanlagen. Bedarfsorientiert zu liefern heißt aber auch, von den Spotmarktpreisen der Strombörsen abhängig zu sein und technische Veränderungen in der Fahrweise einer Anlage zu realisieren. Damit

werden „das Geldverdienen“ sowie die Effizienz einer Anlage nicht mehr kalkulierbar.

Seit dem EEG 2012 haben ca. 10% aller Anlagen in Deutschland das Modell der Direktvermarktung gewählt, neue Anlagen werden ab 2015 kaum in Betrieb gehen [1].

2 Möglichkeiten der Flexibilisierung

Um die Flexibilisierung dennoch interessant zu machen, müssen Lösungen geschaffen werden. Dazu zählen Betreiber, deren subventionierte Einspeisevergütung in ein paar Jahren ausläuft oder mit dem Gedanken spielen, auf Flexibilität zu erweitern oder auch eine neue Anlage auf der Grundlage der Fütterung verschiedener Substrate zu installieren.

Die angestrebte flexible Fahrweise und die Vorgabe von verschiedensten Futterstoffen lassen nur folgende Lösungsansätze einer bedarfsgesteuerten Gasbereitstellung zu:

- Speicherung von Gasmengen zur Pufferung des schwankenden Bedarfs
- Verfahrenstechnische Beeinflussung der Gärung z.B. durch die Hydrolyse von Futterstoffen
- Einsatz intelligenter Steuerungen

Die 2 erst genannten Alternativen sind mit technischen Erweiterungen und damit hohem Kostenaufwand verbunden.

Dieser Beitrag beschreibt eine softwaretechnische Lösung, deren Installations- und Betriebskosten gegen Null gehen.

3 Kurzbeschreibung zur Neuheit

Grundlage der Idee des Projektes ist es, das Verhalten eines Systems softwaretechnisch abzubilden.

3.1 Die Entwicklung des künstlichen neuronalen Netzes

Im ersten Schritt entstand ein Programmsystem, welches aus Messdaten einer gut geführten Biogasanlage ein künstliches neuronales Netz erstellt. Dieses Netz ist dazu geeignet, den Fermenter der Biogasanlage anhand von Vergangenheitsdaten zu simulieren und dessen „Verhaltensweise“ abzubilden. Als Eingangsdaten dienen dabei lediglich die Fütterungsdaten, die in Qualität und Quantität anpassbar sind und damit das System mehr oder weniger sensibilisieren. Im Punkt 4 wird das Programm näher beschrieben.

3.2 Die Prognose

Im zweiten Schritt wird durch einen Auswahlalgorithmus eine Fütterungsempfehlung auf Grundlage des künstlichen neuronalen Netzes erstellt. Dabei werden auch die schwankenden Marktpreise sowie verschiedene Vermarktungsstrategien zeitnah durch Beschreibung des Bedarfs mit berücksichtigt.

3.3 Die Adaption

Im letzten Schritt ist die Software in der Lage, das künstliche neuronale Netz aktiv zu verändern, so dass es sich dem aktuellen biologischen Verhalten des Fermenters immer wieder neu anpasst. Prognose und Adaption werden im Punkt 5 als Regelung zusammengefasst.

4 Der Neuronale Netzwerk Entwickler (NNEntwickler)

Der „Neuronale Netze Entwickler“ ist eine auf MATLAB basierende Software, die auf der Grundlage von Messdaten die Struktur eines künstlich neuronalen Netzes entwickelt und belehrt. Somit kann für eine

Modellierungsaufgabe das Verhalten eines Prozesses mittels Softcomputing aus einem Daten-Set extrahiert werden. Bei der Suche nach einem geeigneten künstlichen neuronalen Netz für eine bestimmte Problemstellung können im Allgemeinen keine festen Richtlinien gegeben werden. Die Generalisierungsfähigkeit, die es zu maximieren gilt, hängt von folgenden Netzparametern ab:

- Die Anzahl der Neuronen in den inneren Schichten
- Die Anzahl und Stärke der Verbindungsgewichte
- Das Lernverfahren, dessen Parameter und der Trainingsablauf

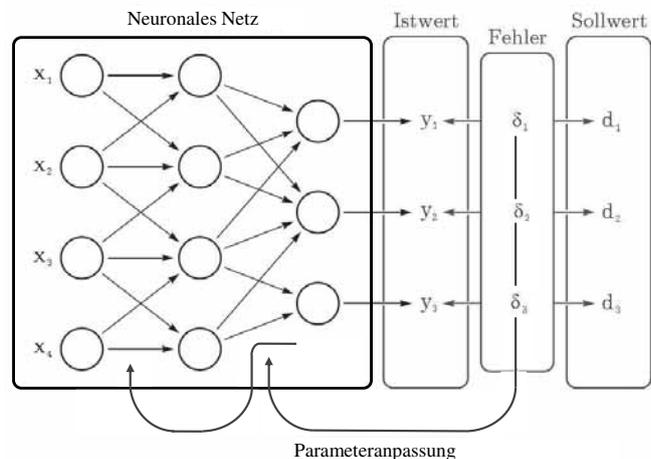


Abbildung 1. Ein einfaches künstliches Neuronales Netz

Abbildung 1 zeigt einen Netzaufbau und das Prinzip des Belohnens. Die grundlegenden Eigenschaften, wie die Netztopologie, können meist schon anhand der vorliegenden Problemstellung auf bestimmte Netzstrukturen reduziert werden. Die Wahl des Lernverfahrens und der Anzahl der Neuronen wird jedoch in der Regel empirisch ermittelt. Bei dieser Vorgehensweise wird zum Beispiel die Größe eines Netzes, also die Anzahl und Breite der inneren Schichten verändert und das Netz anschließend erneut trainiert. Danach wird das neue Netzwerk mit Hilfe eines für das Netz unbekanntes Datensatzes auf seine Generalisierungsfähigkeit geprüft. Ist der Generalisierungsfehler bei der neuen Netzarchitektur geringer, gewinnt diese und ersetzt die alte Struktur.

Auch die Parameter des Lernverfahrens, wie zum Beispiel die Lernrate, beeinflussen das Endergebnis des Trainings enorm. Dadurch muss eine neue Netztopologie mehrmals mit verschiedenen Parametern trainiert werden, um das globale Minimum der

wodurch zu jedem Zeitpunkt die Fütterungen der letzten Tage mit in die Berechnung des Gasertrages zum aktuellen Zeitpunkt einfließen. Die Elman-Netzstruktur, welche die interne Rückkopplung der Neuronen durch Kontextneuronen dar-

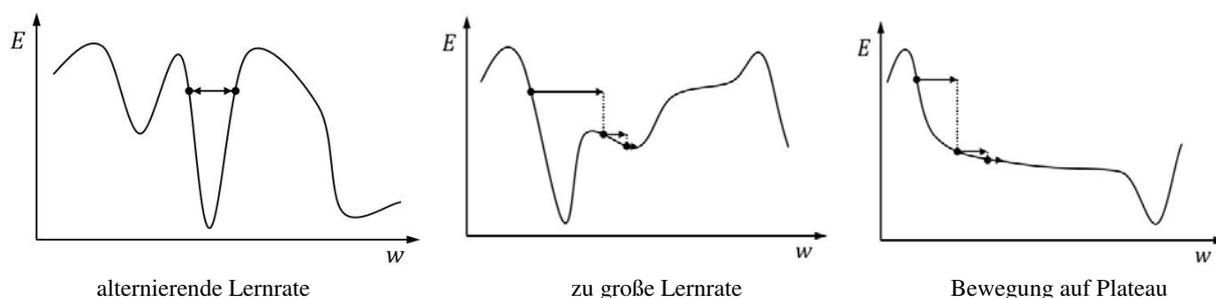


Abbildung 2. Lernratenproblem [3]

Fehlerfunktion zu erreichen. Durch die zufällige Festlegung der Verbindungsgewichte am Anfang des Trainings ergeben sich bei einer wiederholten Durchführung, selbst bei unveränderten Suchparametern, immer andere Endergebnisse. Abbildung 2 zeigt die Probleme der Anpassung der Lernrate.

Der NNEntwickler automatisiert diese immer wiederkehrenden Schritte, um die Suche nach einer geeigneten Netzstruktur und dessen Gewichte zu vereinfachen und mit Hilfe einer strukturierten Vorgehensweise eine Vielzahl von Möglichkeiten in Betracht zu ziehen.

Durch die weitest gehende Automatisierung der Suche nach einem neuronalen Netz ist der NNEntwickler auf spezielle Netzarchitekturen festgelegt, damit der Rahmen der zu testenden Möglichkeiten nicht gesprengt wird. Deshalb können mit dem NNEntwickler nur Netze für Modellierungsaufgaben bearbeitet werden, die

- mehrere Eingangsparameter,
- einen Ausgangsparameter und
- einen Zeitbezug

aufweisen können. Diese Beschränkungen entstanden durch die Erfahrungen am Fermenter einer Biogasanlage, dessen Simulation die Hauptaufgabe des NNEntwicklers darstellt. Deshalb repräsentieren die Eingangsparameter die in den Fermenter gefütterten Massen und der eine Ausgangsparameter den Gasertrag des Fermenters. Der zeitliche Versatz, der sich durch die mechanische und auch chemische Aufspaltung der gefütterten Stoffe bis zu deren Vergasung ergibt, wird durch eine Elman-Netzstruktur beachtet,

stellt, hat sich im Vergleich zu vorwärts gerichteten Netzstrukturen und Jordan-Netzstrukturen am erfolgreichsten für die Prognose von Zeitreihen herauskristallisiert. In den Kontextneuronen werden beliebig viele Vergangenheitswerte gespeichert, wobei deren Verbindungsgewichte konstant bei 1 und damit unveränderbar bleiben. Siehe Abb. 3

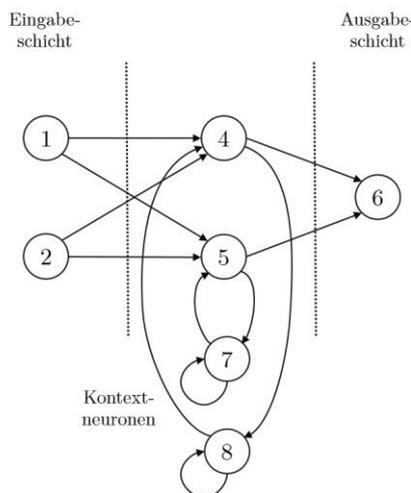


Abbildung 3. Die Elman-Netzstruktur [3]

Die Belehrung der künstlichen neuronalen Netzwerke kann über das „Backpropagation through time“- oder das „QuickProp“-Verfahren geschehen. Dabei stellt das „QuickProp“-Verfahren, das die Fehlerkurve E der Verbindungsgewichte w als parabelähnliche Kurve annimmt, die schnellere und vorteilhaftere Variante dar und wird im NNEntwickler vorrangig genutzt. Siehe Abb.4

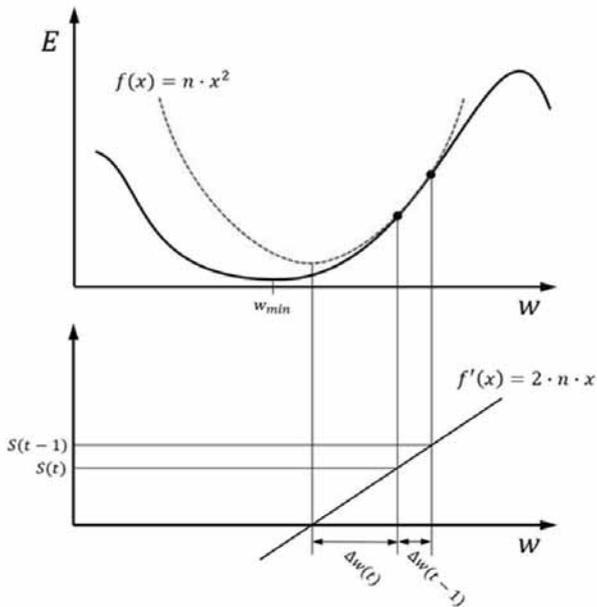


Abbildung 4. QuickProp- Lernverfahren [3]

Damit trotz sinkendem Fehler ein Belehrungsvorgang durch das Erreichen der maximalen Zyklenzahl nicht abgebrochen wird, enthält der NNEntwickler eine „Variable Zyklenzahl“-Funktion, mit der an Hand des Fehlerkurvenverlaufes die „richtige“ Zyklenzahl ermittelt wird. Auch die Lernrate, also die Schrittweite der Gewichtsanzpassung, wird in jedem Lernzyklus an die Fehlerkurve angepasst, um die Bewegung auf größeren Plateaus oder in steilen Schluchten der Fehlerkurve zu optimieren. Siehe Abb.2

Eine der grundlegenden Algorithmen zur Optimierung der Netzfindung stellt jedoch der Evolutionsalgorithmus dar, welcher kontinuierlich das künstliche neuronale Netz aufbaut, neue Verbindungen und Schichten schafft, sowie den Vergangenheitseinfluss und die dazugehörigen Kontextneuronen und Zurückkopplungen variiert. Dadurch kann weitestgehend sichergestellt werden, dass für ein bestimmtes Daten-Set auch die richtige Netzstruktur mit dem idealen Vergangenheitseinfluss gefunden werden kann. Die in Versuchsreihen mit generierten und realen Musterdaten entstandenen neuronalen Netze erreichten bei der Prognose des Gasertrages eines Fermenters gute Fehlerwerte. Dabei konnte auf Grundlage generierter Daten der Gasertrag mit einem durchschnittlichen Fehler von 0,35% und auf Grundlage realer Daten auf durchschnittlich 4% vorhersagt werden. Dadurch wird ersichtlich, dass das Verhalten einer Biogasanlage durch künstlich neuronale Netze sehr gut abgebildet werden kann. Die Güte der Messdaten bestimmt dabei die Präzision der Prognose.

Bei der Suche nach einem neuronalen Netz für ein bestimmtes Daten-Set können durch den NNEntwickler mehrere Lösungen einer Netzstruktur mit gleichen Fehlerwerten gefunden werden, die sich in ihrer Charakteristik jedoch unterscheiden. Es kann dabei in Netze mit einem kriechenden, überschwingenden und pendelnden Verhalten unterschieden werden. Dieses Phänomen ist durch den Netzaufbau mit Vergangenheitseinfluss, also einer Rückkopplung, zu erklären und sollte bei Regelprozessen mit dieser Netzstruktur beachtet werden.

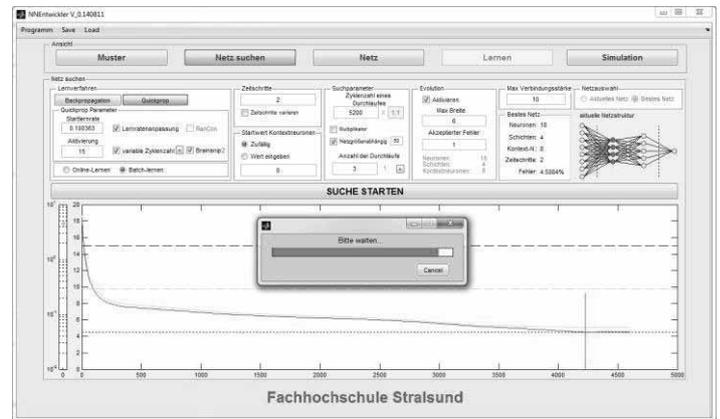


Abbildung 5. Die Oberfläche von NNEntwickler [3]

Abbildung 5 zeigt die Oberfläche des Programmes und Abbildung 6 die Ergebnisse einer Simulation von 50 realen Datensätzen als Vergleich des Gasertrages als Sollwert und berechnetem Gasertrag vom Neuronalen Netz. Tests mit völlig neuen Messdaten lassen auf die Generalisierungsfähigkeit der gefundenen Netze schließen.

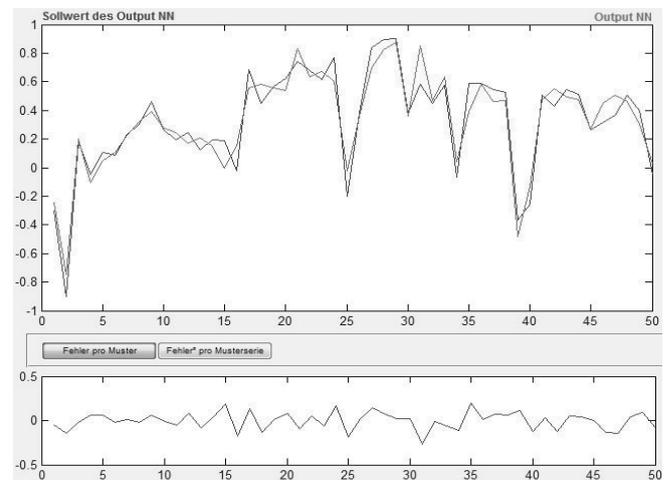


Abbildung 6. Simulation eines realen Datenmusters der BGA Rampe [3]

5 Die Regelung (NNRegler)

Das durch den NNEntwickler erstellte künstliche neuronale Netz wird in einer Arraystruktur gespeichert und kann in anderen Anwendungen verwendet werden. Das Softwaretool NNRegler ist eine ebenfalls auf MATLAB basierende Anwendung. Das Prin-

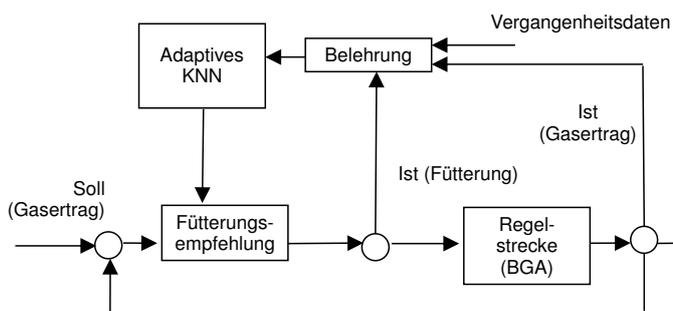


Abbildung 7. Prinzip der Regelung
zip des Reglers ist in Abb. 7 dargestellt.

Im NNRegler wird das gefundene Netz dazu genutzt, um eine Zusammenstellung von Inputgrößen zu erstellen, welche einen vorgegebenen Outputwert erreichen.

Die Prognose

Im konkreten Fall soll prognostiziert werden, welche Futterstoffe für einen benötigten Gasertrag zum Einsatz kommen sollten. Diese Futterstoffe werden anschließend dem Anwender als Fütterungsempfehlung, unter Beachtung von Art und Menge, für den aktuellen Tag ausgegeben.

Neben dem erfolgreich erstellten neuronalen Netz ist die Kenntnis über den Gasbedarf der nächsten 4 Tage erforderlich. Im Laufe des Projektes hat sich ein Tag als ein Zeitschritt am praktikabelsten erwiesen, da der Biogasprozess einerseits träge und zweitens ein Gasspeicher für mehrere Stunden immer verfügbar ist. Aber auch hier ist der Regler variabel.

Zunächst werden die zur Verfügung stehenden Futterstoffe tabellarisch erfasst und es wird festgelegt, welche Höchst- oder Mindestmengen der einzelnen Stoffe definiert sind. Auch kann entschieden werden, ob mindestens 30% der empfohlenen Menge aus Gülle oder anderen tierischen Exkrementen besteht.

Nach dem Start des Berechnungsvorgangs werden alle Möglichkeiten der Fütterung der verfügbaren Futterstoffe nacheinander durch das Netz berechnet und das Ergebnis des Gasertrages mit dem Sollwert verglichen. Die Stoffkombination mit der kleinsten

Abweichung zwischen Soll- und Istwert wird gespeichert und als Fütterungsempfehlung ausgegeben. Dieser Vorgang läuft wie folgt ab:

Die Variation der Inputstoffe n wird für die nächsten vier Zeitschritte k oder besser Prognoseschritte (PS) erstellt. Das bedeutet, jede Variation eines Zeitschritts wird mit jeder Variation der anderen Zeitschritte variiert. Hierbei handelt es sich um eine Variation mit Wiederholung mit der allgemeinen Form zur Berechnung der Anzahl der Variationen (AV).

$$AV_n^k = n^k \tag{1}$$

Die Anlage des Projektpartners arbeitet mit 8 verschiedenen Futterstoffen, so dass für jeden Zeitschritt 8 Inputneuronen am Netz anliegen. Jedes dieser Neuronen kann einen Zustand zwischen 0t und der maximalen Futtermenge annehmen, zum Beispiel realistisch 30t. Beachtet man die zusätzlichen Variationsmöglichkeiten aller vier Prognoseschritte ergibt sich eine Variationsanzahl von:

$$AV_{8 \times 4}^{30} = 32^{30} \tag{2}$$

Um die Prognose für eine Variation zu erstellen, werden zusätzlich sieben Zeitschritte aus der Vergangenheit durch das neuronale Netz berechnet. Die Vergangenheitsdaten (VD) werden benötigt, um die Information der bereits in den letzten Zeitschritten gefütterten Stoffe in das Netz zu übertragen. Notwendig wird dies, da sich der Biogasprozess träge verhält und eine Veränderung des Gasertrages in den nächsten Tagen vom Zustand der vergangenen Fütterungen abhängig ist. So ergibt sich ein Ergebnisset mit 11 Zwischenergebnissen. Siehe Abb.8

Zeitpunkt der Berechnung (heute)										
	Vergangenheit in [d]					Prognose in [d]				
	←					→				
Zeitschritte	7	6	...	2	1	1	...	4		
Inputstoff 1	Vergangenheitsdaten (nicht veränderlich)					Variation der Fütterung				
Inputstoff 2										
Inputstoff 3										
.....										
Inputstoff n										

Abbildung 8. Verwendung der Daten (prinzipiell)

Auch die Einbeziehung der übernächsten Zeitschritte ist notwendig, um die Tendenz des Sollwerts in die Prognose der besten Futterkombination für den nächsten Zeitschritt zu erstellen.

Zusammengerechnet müssten vom Neuronalen Netz die folgende Anzahl von Einzelberechnungen durchgeführt werden.

$$\text{Anzahl der Einzelberechnungen} = AV \times (VD + PS)$$

im Beispiel: $= 32^{30} \times (7 + 4)$ (3)

Diese Anzahl an Berechnungen ist zu hoch, um sie innerhalb einer noch akzeptablen Zeit durchführen zu können. Daher sind im Softwaretool NNRegler einige Annahmen verankert.

Es werden nie alle Futterstoffe gleichzeitig auf der Anlage zur Verfügung stehen. Folglich können vor Beginn der Berechnung die nicht vorhandenen Inputstoffe blockiert werden, so dass sich n aus (1) um die Anzahl dieser verringert. Ein weiterer Weg zur Minimierung des Berechnungsaufwandes ist die Vergrößerung der Schrittweite zwischen der oberen und der unteren Variationsgrenze der Futterstoffmasse. Dadurch wird eine Verringerung

	Futterstoff	Menge[t]
1	Maissilage	40
2	GPS_Roggen	17
3	Festmist	0
4	Grassilage	0
5	Futtergetreide	0
6	Kartoffeln	0
7	frei	0
8	Guelle	20
9	Schaumann	0

Abbildung 9. Fütterungsempfehlung für einen Tag

der Variationsmöglichkeiten erreicht. Die Schrittweite kann beliebig vergrößert oder verkleinert werden. Je mehr Zeit zur Berechnung zur Verfügung steht, also je kleiner eine Schrittweite gewählt werden kann, desto präziser wird die Empfehlung für die bedarfsgerechte Fütterung ausfallen. Versuche mit dem NNRegler haben gezeigt, dass innerhalb eines praxistauglichen Zeitrahmens ausreichend genaue Prognosen und damit eine Empfehlung erstellt werden kann. Von den 4 prognostizierten Zeitschritten in der Zukunft werden die Ergebnisse genutzt, um sie mit den zugehörigen Sollwerten zu vergleichen. Die Ergebnisse stellen Fütterungskombinationen dar. Die Sollwerte sind die Gasbedarfe der nächsten Zeitschritte. Die Kombination, deren Output durch das neuronale Netz die kleinste Abweichung vom Sollwert hat, stellt die beste Kombination von Futterstoffen für die nächsten 4 Zeitschritte dar. Der Anwender erhält jedoch nur die Kombination des ersten Zeitschritts

(Abb.9), da nur dieser zur aktuell anstehenden Fütterung genutzt wird.

Fortlaufend wird in jedem Zeitschritt eine empfohlene Kombination für den nächsten ausgegeben.

Die Adaption

Bevor der Benutzer die Berechnung zur Empfehlung für den nächsten Tag starten kann, müssen die Fütterungsdaten des aktuellen Tages sowie der aktuelle Gasertrag ins Betriebstagebuch eingetragen werden. Diese Werte sind wichtig für die Prognose einerseits und für die Weiterentwicklung des Netzes andererseits. Die Eigenschaften der Futterstoffe unterliegen Schwankungen, die saisonal bedingt, abhängig vom Anbauggebiet oder der Lagerung sind und damit verschiedene Energiegehalte aufweisen. Daher besteht die Notwendigkeit einer permanenten Adaption des neuronalen Netzes auf die aktuellen Eigenschaften der Inputstoffe. Diese Justierung ist dem Berechnungsprozess der Empfehlung vorgeschaltet. So passt sich das Netz jeden Tag auf die aktuell im Silo liegenden Stoffe an. Das Netz wird dabei durch den bereits beschriebenen Trainingsalgorithmus belehrt und dessen Eigenschaften leicht geändert. Die leichte Veränderung passiert ausschließlich durch eine Gewichtsanzpassung der Verbindungen zwischen den Neuronen. Eine Änderung der Netzstruktur erfolgt nicht.

Dadurch ist das Programmsystem in der Lage, über einen großen Zeitraum für eine Anlage zu arbeiten und sich an die verändernden Eigenschaften der Futterstoffe oder anderer Parameter anzupassen und das Systemverhalten der Anlage abzubilden.

6 Die Spotpreise regeln den Gasbedarf

Wie oben genannt, werden in Zukunft die Marktpreise an den Strombörsen über die Einkünfte einer Biogasanlage entscheiden. Jeder Betreiber will bzw. muss mit seiner Anlage in erster Linie rentabel sein. Das heißt, es wird dann Strom verkauft, wenn es sich lohnt und die Preise an den Börsen entsprechend hoch sind. Genau das ist ja gerade der Vorteil der Bioenergie im Vergleich zu den anderen regenerativen Energien.

Ziel der weiteren Arbeit im Projekt ist es demnach, den Gasbedarf nicht mehr nur manuell einzugeben. Die Spotmarktpreise sind programmtechnisch zu erfassen, die bekannten tages- und saisonabhängigen

Höckerkurven des Strombedarfs in die Analysen einzubeziehen und dem Betreiber daraufhin einen angepassten Gasbedarf vorzuschlagen.

7 Danksagung

Das Projekt wird vom Bundesministerium für Bildung und Forschung im Rahmen von FHprofUnt - Forschung an Fachhochschulen - gefördert.

Als langjähriger Kooperationspartner steht uns die Firma Rotaria Energie und Umwelttechnik GmbH aus Rerik mit Rat und Tat zur Seite. Ohne die Erfahrung, die wir selbst auf einer firmeneigenen Industrieanlage von 1MW Leistung in Rampe (M/V) sammeln konnten, wären wir nicht zu den jetzigen Ergebnissen gekommen. Sämtliches Datenmaterial, das Betriebs-tagebuch, das Validieren unserer Tests und die tägliche Kommunikation mit den Anlagenbetreibern waren notwendige Bausteine zum Gelingen.

8 Literatur

- [1] Mattes Scheftelowitz u.a. *Vorhaben IIa Stromerzeugung aus Biomasse*-Wissenschaftlicher Bericht, DBFZ Leipzig, Juli 2014
- [2] EEG 2014 Ausfertigungsdatum: 21.07.2014
- [3] M. Knappe. *Entwicklung eines Programmsystems für das Modellieren von Biogasanlagen mittels optimierter Neuronaler Netze*. Bachelorarbeit, FH Stralsund, 2014.
- [4] Ch. Wahmkow und M. Knappe und E. Konnerth. *Entwicklung einer intelligenten Steuerung für Biogasanlagen*. Abschlussbericht zum Förderprojekt des Ministeriums für Wirtschaft, Arbeit und Tourismus MV, FH Stralsund, 2013.

Modellierung und Parametrierung des respiratorischen Systems eines Patienten in der Atemtherapie

M. Scheel^{1,2}, A. Berndt¹, O. Simanski²

¹HOFFRICHTER GmbH Schwerin, Mettenheimerstraße 12, 19061 Schwerin

²Computational Engineering and Automation Group - Hochschule Wismar-
University of Applied Sciences Technology, Business and Design,
Philipp-Müller-Straße 14, 23966 Wismar
scheel@hoffrichter.de

Die CPAP-(Continuous Positive Airway Pressure) Therapie ist das Standardverfahren, um das obstruktive Schlafapnoesyndrom zu behandeln. Während des Schlafes kann es zu Atemstillständen (Apnoe) kommen, wenn die oberen Atemwege kollabieren. Das CPAP-Gerät generiert einen positiven Luftdruck, um die oberen Atemwege pneumatisch zu schienen. Dieser Überdruck darf bestimmte Grenzen nicht über- bzw. unterschreiten, damit ein angenehmes Atmen garantiert werden kann. Durch die Atmung des Patienten wird dieser Druck erheblich gestört. Um ein besseres Therapiegefühl sicherstellen zu können, sollen adaptierte Patientenmodelle in modernen Regelungsverfahren Anwendung finden. Dafür ist die Identifikation des respiratorischen System essentiell.

1 Einleitung

Das obstruktive Schlafapnoesyndrom (OSAS) ist eine weitverbreitete Schlafkrankheit. Das OSAS kann bei Nichtbehandlung zu schwerwiegenden und irreparablen Folgeerkrankungen (Herzinsuffizienz, Diabetes, Schlaganfall) führen. Durch die kontinuierliche Überdrucktherapie (CPAP - Continuous Positive Airway Pressure) kann das Schlafapnoesyndrom weitestgehend behandelt und das Risiko der Erkrankungen minimiert werden. Dennoch zeigen einige Studien, dass die Akzeptanz der CPAP-Therapie bei lediglich ca. 50% liegt [1]. Der Grund dafür liegt in falsch eingestellten Therapiedrücken und in der Nichteinhaltung des vorgegebenen Solldruckes. Ein zu hoher Druck kann die Lunge schädigen - ein zu niedriger kann eine auftretende Apnoe nicht verhindern.

Um die Risiken zu mindern, müssen patientenoptimierte Geräteeinstellungen gefunden werden. Es werden mathematische Modelle des respiratorischen Systems als Basis für einen späteren modellbasierten Regelungsentwurf benötigt [2]. Daher ist eine zuverlässige Identifikation des Atemsystems notwendig.

2 Patientenmodelle

Der Identifikation des respiratorischen Systems liegen verschiedene Modellstrukturen zu Grunde. Dabei können die Modelle in lineare und nichtlineare unterteilt werden. Viele wissenschaftliche Arbeiten haben sich mit der Identifikation verschiedener Patientenmodelle befasst [3, 4, 5, 6, 7]. Die in dieser Arbeit untersuchten Modelle sollen möglichst einfach sein, dennoch eine ausreichende Modellqualität aufweisen. Sie sollen die Basis für einen späteren Regelungsentwurf bilden.

2.1 Lineare Patientenmodelle

Ein sehr einfaches Modell eines Patienten zeigt Bild 1. In Analogie zur Elektrotechnik kann die Übertragungsfunktion des linearen RC-Modells hergeleitet werden. Die Größen des respiratorischen Systems sind der Maskendruck p , der Patientenfluss \dot{V} , der Widerstand (Resistance) der oberen Atemwege R , die Lungendehnbarkeit (Compliance) C und der Druckan-

teil der Atemmuskulatur p_{mus} . Im Bildbereich ergibt

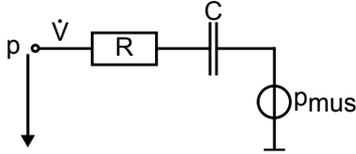


Abbildung 1: Elektrisches Ersatzschaltbild RC-Modell

sich für Druck und Patientenfluss der folgende Zusammenhang.

$$p = R \dot{V} + \frac{1}{C} \int \dot{V} dt + p_{mus} \quad (1)$$

$$= \left(R + \frac{1}{sC}\right) \dot{V} + p_{mus} \quad (2)$$

$$= \frac{1 + sRC}{sC} \dot{V} + p_{mus} \quad (3)$$

Weitere bekannte Patientenmodelle sind in Bild 2 dargestellt [4, 5].

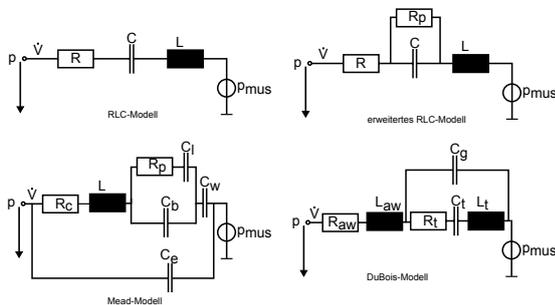


Abbildung 2: Elektrisches Ersatzschaltbild verschiedener Modelle

RLC-Modell: Der Widerstand der oberen Atemwege R , die Lungenträgheit (Inertance) L und die Lungendehnbarkeit C sind als Reihenschwingkreis modelliert.

erweitertes RLC-Modell: Das erweiterte RLC-Modell ist eine Verbesserung des RLC-Modells. Der hinzugefügte periphere Widerstand R_p erlaubt eine frequenzabhängige Beobachtung der typischen Impedanz.

Mead-Modell: Das Modell nach Mead simuliert verschiedene Vorgänge in der Lunge und in der Brustwand. Die Parameter können wie folgt interpretiert werden: Lungenträgheit L , zentraler und peripherer

Widerstand R_c und R_p und die Dehnbarkeit von Lunge C_l , Brustwand C_w , Bronchien C_b und extrathorakale Dehnbarkeit C_e .

DuBois-Modell: Das DuBois-Modell unterteilt die Luftwege, Gewebe und Lungenbläschen in verschiedene Sektoren. Die Parameter sind die Atemwegs- und Gewebewiderstände (R_{aw}, R_t), Atemwegs- und Gewebeträgheit (L_{aw}, L_t) und die Gewebe- und Lungendehnbarkeit (C_t, C_g),

2.2 Nichtlineare Patientenmodelle

Grundlage der nichtlinearen Patientenmodelle ist die Bewegungsgleichung des respiratorischen Systems, bestehend aus Atemwegswiderstand R , Lungendehnbarkeit C und Lungenträgheit L (RLC-Modell). Schüttler beschreibt den zeitlichen Verlauf des Druckes $p(t)$ als Funktion von Volumen V , Volumenstrom \dot{V} und Volumenbeschleunigung \ddot{V} - p_0 ist der Ausgangsdruck [8].

$$p(t) = L \ddot{V}(t) + p_{aw}(V, \dot{V}, t) + p_{el}(V, \dot{V}, t) + p_0 \quad (4)$$

In Anlehnung an die realen Verhältnisse können die Druckkomponenten p_{aw} und p_{el} durch eine Taylorreihe approximiert werden:

$$p(t) = \sum_{i=1}^{N_C} \frac{1}{C_i} V^i(t) + \sum_{j=1}^{N_R} R_j \dot{V}^j(t) + \sum_{k=1}^{N_L} L_k \ddot{V}^k(t) \quad (5)$$

Durch Wahl der oberen Summengrenzen $p_{\{N_C, N_R, N_L\}}$ können lineare und nichtlineare Modelle beschrieben werden. Die linearen RC- und RLC-Modelle können durch Wahl von $p_{\{1,1,0\}}$ und $p_{\{1,1,1\}}$ erstellt werden. Ein nichtlineare Modell, welches untersucht wurde, ist das Modell $p_{\{2,2,1\}}$. Die vollständige Bewegungsgleichung des respiratorischen System lautet dann:

$$p(t) = \frac{1}{C_1} V(t) + \frac{1}{C_2} V^2(t) + R_1 \dot{V}(t) + R_2 \dot{V}^2(t) + L \ddot{V}(t) \quad (6)$$

3 Parameteridentifikation

3.1 Grundidee der Forced Oscillation Technique

Die Atmung der Patienten hat Einfluss auf den Verlauf des Maskendruckes (s. Gl. 3). Da der Anteil der

Atemmuskulatur p_{mus} nicht gemessen wird, muss er für eine zuverlässige Identifikation der Lungenparameter eliminiert werden. Die Anfänge der *Forced Oscillation Technique* (FOT) gehen bis in die 50er/60er Jahre zurück und wurden durch DuBois geprägt [9]. Sie ist heutzutage ein Standardverfahren zur Bestimmung eines respiratorischen Atemmodells. Die Idee der FOT besteht in der Erzeugung eines überlagerten Sinussignals zum Therapiedruck. Die Frequenzen der Sinusschwingung liegen typischerweise zwischen $f_0 = 4 - 30 Hz$. Ein großer und zwingend notwendiger Vorteil der FOT in der Atemtherapie ist, dass die Therapie während der FOT-Analyse nicht unterbrochen bzw. auch kein spezielles Atemmanöver durchgeführt werden muss. Da die Frequenzen des Atemanteils im Bereich von $0,1 - 1 Hz$ liegt und damit deutlich geringer als die FOT-Frequenzen, kann durch Bandpassfilterung die Atmung der Patienten eliminiert werden.

Durch Aufnahme des Maskendruckes und des Patientenvolumenstroms können die Ein- und Ausgangsgrößen des respiratorischen Modells festgelegt werden. Ein beispielhafter Verlauf der Messwerte ist in Bild 3 dargestellt. Es ist deutlich die überlagerte Sinusschwingung in den Signalen zu erkennen.

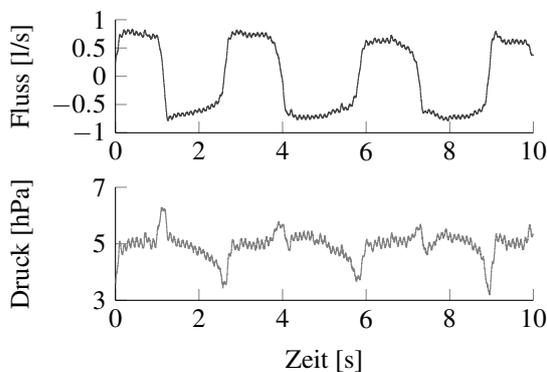


Abbildung 3: Verlauf der Messgrößen Patientenfluss (blau) und Maskendruck (rot)

Durch Bandpassfilterung der Druck- und Volumensstromsignale enthält jedes Signal nur einen Frequenzanteil an der Stelle f_0 . Für die Offline-Identifikation der Parameter wird der Filter sowohl vorwärts, als auch rückwärts angewendet, um eine Phasenverschiebung von 0 zu erhalten. Mit diesen gefilterten Signalen kann die Identifikation der Lungenparameter durchgeführt werden.

3.2 Parametrierung verschiedener Lungenmodelle

Für die Identifizierung der respiratorischen Modelle wurde der Simplex-Algorithmus nach Nelder-Mead benutzt. Dieser ist in der Matlab-Funktion *fminsearch* implementiert. Durch Vorgabe der Startparameter werden die Systemparameter nach der Methode der kleinsten Fehlerquadrate iterativ bestimmt.

Es wurden beispielhaft die Druck-Volumenstromkurven von 4 Probanden mit 2 verschiedenen überlagerten Sinusschwingungen ($f_0 = 5 Hz$ und $f_0 = 10 Hz$) aufgenommen und anschließend ihre Parameter identifiziert. Die Modellqualität des RLC-Modells über 1 Sekunde ist beispielhaft in Bild 4 dargestellt.

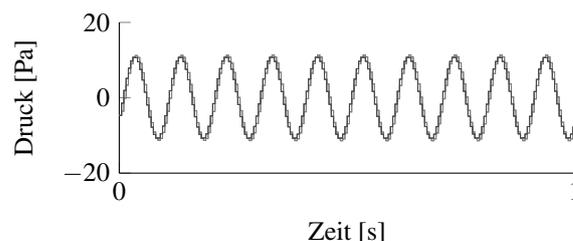


Abbildung 4: Modellqualität nach Parameteroptimierung - gemessen (blau) - modelliert (rot)

In den folgenden Tabellen sind die identifizierten Lungenparameter der 4 Probanden für unterschiedliche Modellstrukturen dargestellt. Zudem ist die Summe der Fehlerquadrate (über 10 Sekunde) dargestellt, um den Einfluss der Modellstruktur/Modellkomplexität auf die Modellqualität aufzuzeigen. In Tabelle 1 wurde zunächst der Einfluss der Sinusfrequenz auf die Modellqualität des RC-Modells für 4 Probanden untersucht. Fehler_x bezeichnet die Summe der kleinsten Fehlerquadrate bei der Frequenz f_0 .

Tabelle 1: Fehler des RC-Modells

	Fehler ₅	Fehler ₁₀
Proband 1	38400	11445
Proband 2	69570	11085
Proband 3	57730	3649
Proband 4	29460	3109

Tabelle 1 zeigt einen geringeren Fehler des RC-Modells bei der Frequenz $f_0 = 10\text{Hz}$. Das kann damit begründet werden, dass der Abstand der Oszillationsfrequenz f_0 zu den Atemfrequenzanteilen größer ist und die Bandpassfilterung dort eine höher Dämpfung aufweist.

In den folgenden 4 Tabellen 2-5 wurden die Parameter der verschiedenen Lungenmodelle aus den überlagerten 10 Hz Signalen bestimmt. R, L und C sind in Pa/ml/s , Pa/ml/s^2 und ml/Pa gegeben.

Tabelle 2: Lungenparameter Proband 1

Modell	Parameter		
RC	R = 0,359	C = 0,083	
RLC	R = 0,352	C = 0,067	L = 0,002
eRLC	R = 0,373	C = 0,058	L = <0,001
	$R_p = 1270$		
DuBois	$R_{aw} = 0,367$ $R_t = 0,043$	$L_{aw} = <0,001$ $L_t = 0,026$	$C_t = 0,009$ $C_g = 0,070$
Mead	$R_c = 0,261$ $R_p = 0,065$ $C_t = 356$	L = 0,001 $C_b = <0,001$	$C_w = 0,068$ $C_e = <0,001$
$p_{2,2,1}$	$R_1 = 0,358$ $C_1 = -0,111$	$R_2 = <-0,001$ $C_2 = -0,698$	L = -0,021
$p_{3,3,1}$	$R_1 = 0,359$ L = -0,027 $C_3 = 2,236$	$R_2 = <-0,001$ $C_1 = -0,106$	$R_3 = <-0,001$ $C_2 = -0,411$

Tabelle 3: Lungenparameter Proband 2

Modell	Parameter		
RC	R = 0,266	C = 0,105	
RLC	R = 0,262	C = 0,086	L = 0,002
eRLC	R = 0,280	C = 0,080	L = <0,001
	$R_p = >9000$		
DuBois	$R_{aw} = 0,258$ $R_t = 1,552$	$L_{aw} = <0,001$ $L_t = 0,033$	$C_t = 0,004$ $C_g = 0,097$
Mead	$R_c = 0,263$ $R_p = <-0,001$ $C_t = 95,57$	L = 0,002 $C_b = 0,019$	$C_w = 0,085$ $C_e = <0,001$
$p_{2,2,1}$	$R_1 = 0,266$ $C_1 = 0,171$	$R_2 = <-0,001$ $C_2 = -4,497$	L = -0,004
$p_{3,3,1}$	$R_1 = 0,269$ L = 0,009 $C_3 = -3,118$	$R_2 = <-0,001$ $C_1 = 0,198$	$R_3 = <-0,001$ $C_2 = 1,178$

Tabelle 4: Lungenparameter Proband 3

Modell	Parameter		
RC	R = 0,276	C = 0,169	
RLC	R = 0,280	C = 0,161	L = <0,001
eRLC	R = 0,280	C = 0,153	L = <0,001
	$R_p = >9000$		
DuBois	$R_{aw} = 0,272$ $R_t = 0,149$	$L_{aw} = <0,001$ $L_t = 0,036$	$C_t = 0,006$ $C_g = 0,156$
Mead	$R_c = 0,158$ $R_p = 0,112$ $C_t = 108,65$	L = <0,001 $C_b = 0,009$	$C_w = 0,161$ $C_e = <0,001$
$p_{2,2,1}$	$R_1 = 0,276$ $C_1 = 0,184$	$R_2 = <-0,001$ $C_2 = -7,646$	L = 0,005
$p_{3,3,1}$	$R_1 = 0,304$ L = 0,115 $C_3 = -3,118$	$R_2 = <-0,001$ $C_1 = -0,475$	$R_3 = <-0,001$ $C_2 = -0,182$

Tabelle 5: Lungenparameter Proband 4

Modell	Parameter		
RC	$R = 0,274$	$C = 0,059$	
RLC	$R = 0,284$	$C = 0,055$	$L = 0,001$
eRLC	$R = 0,312$	$C = 0,051$	$L = <0,001$
	$R_p = 869$		
DuBois	$R_{aw} = 0,266$ $R_t = 5,586$	$L_{aw} = <0,001$ $L_t = 0,041$	$C_t = 0,005$ $C_g = 0,057$
Mead	$R_c = 0,234$ $R_p = -0,002$ $C_l = 145,89$	$L = 0,001$ $C_b = 0,005$	$C_w = 0,054$ $C_e = <0,001$
$p_{2,2,1}$	$R_1 = 0,277$ $C_1 = -0,302$	$R_2 = 0,002$ $C_2 = 0,116$	$L = 0,062$
$p_{3,3,1}$	$R_1 = 0,278$ $L = 0,064$ $C_3 = -2,256$	$R_2 = 0,002$ $C_1 = -0,247$	$R_3 = <0,001$ $C_2 = 0,104$

Ein Vergleich zu verschiedenen Literaturreferenzen [4, 10] zeigt eine gute Übereinstimmung mit den identifizierten Widerständen (R, R_1, R_c, R_{aw}) und der Lungendehnbarkeit (C, C_t, C_l). Lediglich die Inertance (L, L_{aw}) wird teilweise unter- oder überschätzt und nimmt sogar negative Werte an (Tabelle 3 $p_{2,2,1}$). Die teilweise negativen Werte beim Mead- und DuBois-Modell können mit den Literaturangaben nicht bestätigt werden. Bei den nichtlinearen Modellen $p_{2,2,1}$ und $p_{3,3,1}$ kann kein direkter Zusammenhang zur den realen Zuständen der Lunge hergeleitet werden. Trotzdem wurden sie hier erwähnt, da sie in Summe den kleinsten quadratischen Fehler aufweisen.

Die folgende Grafik 5 soll den Einfluss der Modellkomplexität auf die Summe der Fehlerquadrate darstellen.

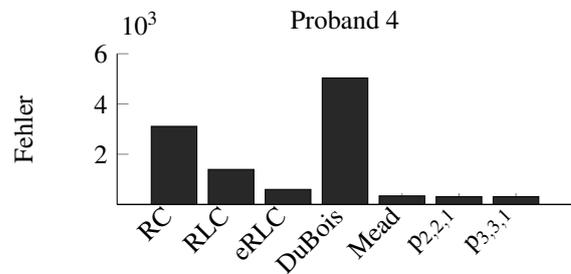
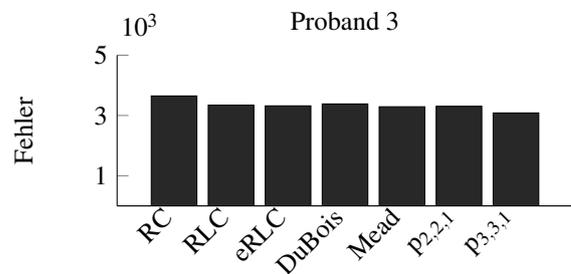
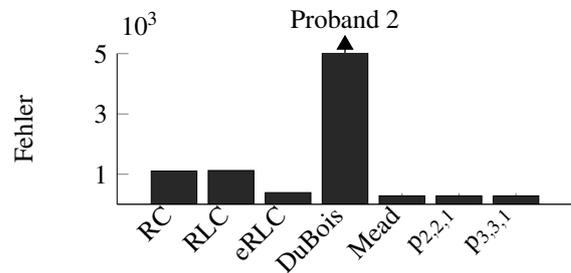
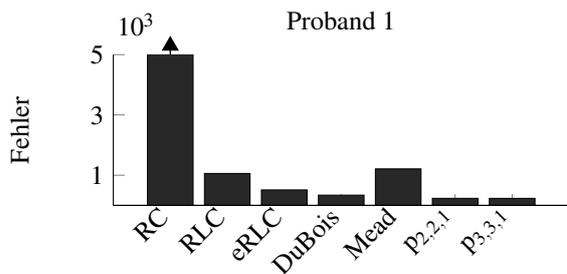


Abbildung 5: Summe der kleinsten Fehlerquadrate für verschiedene Modelle

Es ist zu erkennen, dass im Allgemeinen mit steigender Modellkomplexität die Summe der Fehlerquadrate abnimmt, da der Verlauf der Ausgangsgröße durch mehrere Parameter angepasst werden kann. Die großen Unterschiede beim Mead- und DuBois-Modell können auf eine zu große Abtastzeit und den dadurch nicht erfassten Dynamiken zurückgeführt werden. Bei allen Probanden ergeben sich für das nichtlineare Modell $p_{3,3,1}$ die geringsten Fehlerwerte.

4 Schlussfolgerung

Sowohl in der künstlichen Beatmung, als auch in der Atemtherapie bekommen Patientenmodelle eine immer wichtigere Bedeutung zugesprochen, um patienten-individuelle Geräteeinstellungen zu finden. Die hier durchgeführten Untersuchungen an verschiedenen Probanden hat gezeigt, dass mit der FOT-Methode eine Parameteridentifikation verschiedener Lungenmodelle vollzogen werden kann.

Die Qualität des Modells hängt zumeist von der Modellkomplexität ab. Komplexere Modelle erzielten in dieser Arbeit bessere Ergebnisse als einfache RC- bzw. RLC-Modelle, benötigten aber wesentlich mehr Rechenleistung für Identifikation und weiterführende Berechnungen. Welches Modell benutzt wird, hängt vom Anwendungsfall und den verfügbaren Ressourcen (Hardware, Rechenleistung) ab.

In der Atemtherapie sind die Ressourcen aufgrund des günstigen Gerätepreises begrenzt. Deshalb sollte hier ein einfaches, aber dennoch gutes Modell verwendet werden. Das RLC-Modell lieferte bei allen Probanden gute Ergebnisse und ist für viele Berechnungen ausreichend.

Literatur

- [1] G. Nilius. *Ist eine Akzeptanzverbesserung der nächtlichen Überdruckbeatmung möglich?* 54. Kongress der Deutschen Gesellschaft für Pneumologie und Beatmungsmedizin e. V., Deutschland, 2013.
- [2] M. Scheel, A. Berndt, R. Kölln, O. Simanski und A. Sievert. *Model-based Control Approach for a CPAP-Device*. Walter de Gruyter, Volume 59 Number S1, ISSN 0013-5585 , e-ISSN 1862-278X, Hannover, 2014.
- [3] F. Dietz, A. Schloßler und D. Abel. *Flow Feedback Control in Home Care Ventilation Considering Spontaneous Breathing*. GMA Kongress 2003, VDI-Berichte 1756, VDI-Verlag, Düsseldorf, 2003.
- [4] B. Diong, H. Nazeran, P. Nava und M. Goldman. *Modeling Human Respiratory Impedance - Comparing the Best Method with the Least Estimation Errors*. IEEE ENGINEERING IN MEDICINE AND BIOLOGY MAGAZINE, Lyon, France, 2007.
- [5] K. Lopez-Navas, E. Rother und U. Wenkebach. *Comparison of six models of the respiratory system based on parametric estimates from three identification models*. Walter de Gruyter, Biomedical Engineering / Biomedizinische Technik. Band 57, Heft SI-1 Track-O, Seiten 48-51, ISSN (Online) 1862-278X, ISSN (Print) 0013-5585, DOI: 10.1515/bmt-2012-4011, Jena, 2012.
- [6] C. Ngo, B. Misgeld, T. Vollmer, S. Winter, S. Leonhardt. *Linear affine lung mechanics model with emphasis on pleural dynamics*. Walter de Gruyter, Volume 59 Number S1, ISSN 0013-5585 , e-ISSN 1862-278X, Hannover, 2014.
- [7] C. Schranz, P. Docherty, Y. S. Chiew, K. Möller und J. G. Chase. *Iterative integral parameter identification of a respiratory mechanics model*. BioMedical Engineering OnLine, 2012.
- [8] F. Schüttler. *Parameterschätzung atmungsmechanischer Modelle zur Vorhersage pharyngealer Obstruktionen bei Patienten mit schlafbezogenen Atmungsstörungen*. Dissertation, Philipps-Universität Marburg, Deutschland, 2004.
- [9] A. DuBois, A. Brody, D. Lewis und B. Burgess. *Oscillation mechanics of lungs and chest in man*. Journal of Applied Physiology, 8, 1956.
- [10] S. Leonhardt, S. Böhm und B. Lachmann. *Optimierung der Beatmung beim akuten Lungenversagen durch Identifikation physiologischer Kenngrößen*. at-Automatisierungstechnik 46(9), 532-539, R. Oldenbourg Verlag GmbH, München, 1998.

Modellbasierte Reglerauslegung eines sphärischen Elektroantriebs

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

¹Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel, Germany

^a*x.liu-henke@ostfalia.de*, ^b*mar.goellner@ostfalia.de*, ^c*ha.tao@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. Die Kinematik und Dynamik des gesamten mechatronischen Systems wird in einem detaillierten dreidimensionalen Modell repräsentiert. Darauf basierend wird die Regelstruktur des Systems ausgelegt und anhand von Anwendungsszenarien validiert.

1 Einleitung

Die höhere Lebenserwartung der Menschen führt direkt zu einer Zunahme multimorbider Erkrankungen, die mit der Alterung im Zusammenhang stehen. Der Anteil der in der Gesellschaft vorhandenen Menschen mit körperlicher Beeinträchtigung steigt aufgrund dessen weiter an. Dementsprechend ergeben sich in diesem Zusammenhang auch Alltagsprobleme, die die Pflege und Autonomie körperlich beeinträchtigter Menschen erschwert. Konventionelle Mobilitätshilfen, wie z.B. ein konventioneller Rollstuhl, sind nur in begrenztem Maße in der Lage den gewohnten Mobilitätswinkel aufrecht zu erhalten. Gerade die fehlende Dynamik und Wendigkeit dieser Systeme stellt ein Problem dar.

Ziel eines Forschungsvorhabens der Fachgruppe Regelungstechnik der Ostfalia HaW ist die modellbasierte, rechnergestützte Entwicklung eines omnidirektionalen, sphärischen Elektroantriebes (S-Mobile) als Stelleinrichtung einer Serviceplattform für den Pflegen- und Gesundheitsbereich [1]. Dieses S-Mobile soll z.B. als Basis eines Serviceroboters für die Interaktion mit Menschen in typischen humanoiden Umgebungen oder als Stellrichtung einer intelligenten Mobilitätshilfe dienen.

Abbildung 1 zeigt das Konzept des S-Mobile, bestehend aus der Antriebskugel und dem Aufbau. Der Aufbau ist untergliedert in die mechanische Tragsstruktur sowie den Aktuatoren, der Sensorik und der Informationsverarbeitung, welche mit der Antriebskugel das Kernmodul bilden.

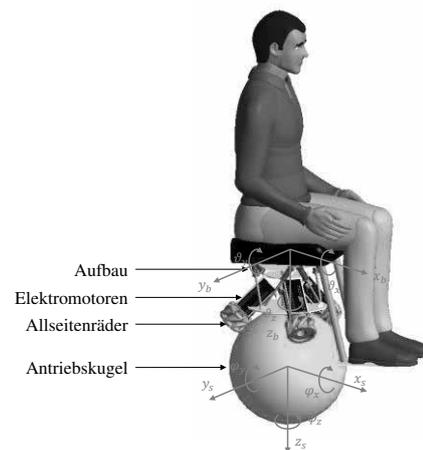


Abbildung 1. Konzept des S-Mobile

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

Mittels einer hierarchischen Regelstruktur wird eine kontrollierte Bewegung des S-Mobile in fünf vorgesehenen Freiheitsgraden, der Längsbewegung in x- und Y-Ebene, Gieren, Wanken und Nicken, ermöglicht.

Als Aktuatoren werden Elektromotoren eingesetzt, da die geforderte hohe Systemdynamik mit anderen Übertragungsmedien nicht gewährleistet ist. Hinsichtlich der Anforderungen an die Schnelligkeit und Bandbreite der Aktoren, aus Sicht des zu entwerfenden Regelkreises, kommen hier nur elektrische Motoren in Frage. Zudem müssen kleine Winkel mit konstanten Drehmomenten ausgeglichen werden, weshalb hier Synchronstrommotoren verwendet werden sollen. Die Anordnung der Aktoren spielt für die Realisierung eine große Rolle. Abbildung 1 zeigt die gewählte Konfiguration mit einem sphärisch angeordneten Antrieb mit drei um 120 Sphärengrad versetzten Antriebsrädern. Durch diese geeignete Anordnung können jegliche weitere Stützstrukturen eingespart werden. Zudem ermöglicht dieses Antriebskonzept ein aufrechtes Drehen des Aufbaus um seine Hochachse.

Basierend auf diesem Konzept kann der mechatronischen Entwurf durchgeführt werden.

2 Modellbildung

Die Entwicklung erfordert eine systematische und interdisziplinäre Vorgehensweise, die eine optimale Auslegung des Gesamtsystems erlaubt. Diese mechatronische Komposition erfolgt in einem iterativen Prozess, bestehend aus Modellbildung, Analyse und der modellbasierten Reglerauslegung. Durch die modellbasierte Auslegung können z.B. benötigte Stellkräfte gefunden und zum Auslegen von Aktoren genutzt werden [2], [3].

Ein vereinfachtes planares Streckenmodell wurde anhand dieser Systematik gefunden [1]. Dieses soll weiterhin zur Auslegung der Regelstruktur herangezogen werden. Um die Koppelkräfte zwischen diesen planaren Modellen zu berücksichtigen, wird hier zudem ein dreidimensionales, nichtlineares Modell hergeleitet. Mit diesem Modell soll untersucht werden, inwiefern die getroffenen Vereinfachungen in den physikalischen Extrembereichen des Systems noch Gültigkeit haben und ob entsprechenden Abweichungen toleriert werden können. Durch dieses Vorgehen kann validiert werden, inwieweit die hier hergeleitete Reglerstruktur für die geforderten Systemcharakteristika ausreichend ist.

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

Im Folgenden werden Kinematik und Dynamik im Detail dargestellt.

2.1 Kinematikfunktionen

Die Kinematikfunktionen dienen als Grundlage der dynamischen Funktionen um Lage, Orientierung, Winkel und Winkelgeschwindigkeiten zu ermitteln. Zudem sind die kinematischen Funktionen Teil der Regelalgorithmen um die stark miteinander verknüpften Regelgrößen zu entkoppeln (vgl. Kap.3).

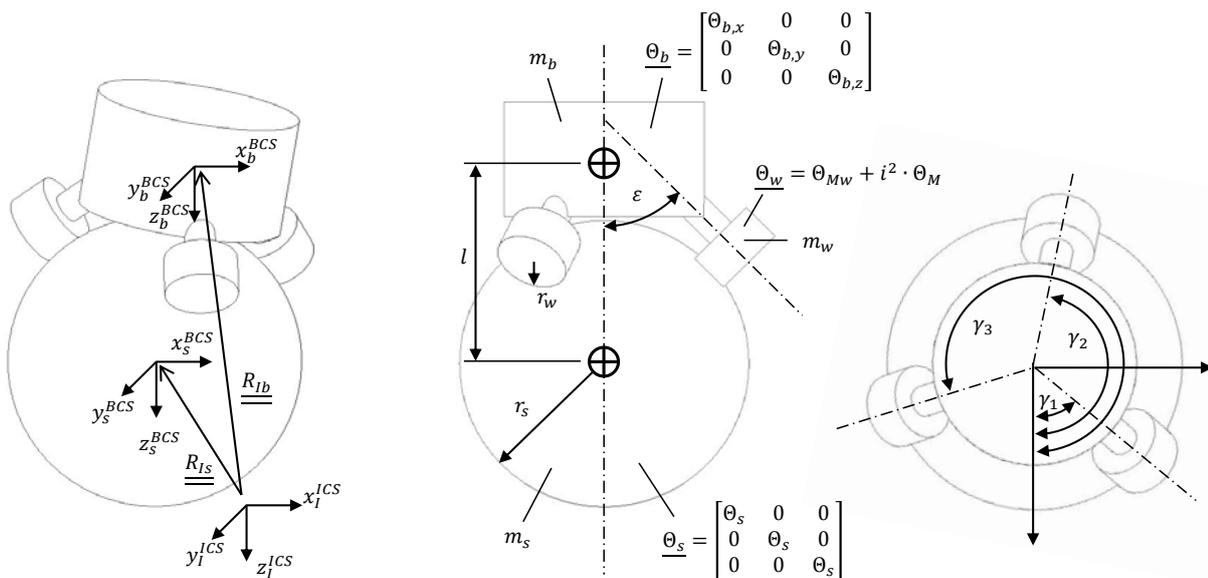


Abbildung 2. Physikalisches Streckenmodell

Die kinematischen Zusammenhänge werden aus den Freiheitsgraden und der Geometrie des S-Mobile bestimmt. Die einzelnen Teilkomponenten werden mit geeigneten Koordinatensystemen nach Abbildung 2 versehen. Das Inertialkoordinatensystem (ICS) mit den Komponenten (x_I, y_I, z_I) liegt auf der Fahrbahn. Die jeweiligen Körperkoordinatensysteme (BCS) liegen in jedem Starrkörper. Die Kinematik des Gesamtsystems wird durch Koordinatentransformation mittels Transformationsmatrizen symbolisch hergeleitet. Für die Rotation um die jeweiligen Achsen x, y, z entstehen die grundlegenden Rotationsmatrizen:

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

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

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

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

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

$$\underline{\underline{s\Omega_s}} = \begin{bmatrix} \dot{\varphi}_x \\ \dot{\varphi}_y \\ 0 \end{bmatrix} \quad (4)$$

Die absoluten Winkelgeschwindigkeiten des Aufbaus ergeben sich durch die Multiplikation der Tait-Bryan-Winkel $\underline{\hat{\theta}}$ mit dem totalen Differential der entsprechenden Transformationsmatrix, dargestellt als Jacobi-Matrix $\underline{\underline{J}}$:

$$\underline{\underline{b\Omega_b}} = \underline{\underline{J}} \cdot \underline{\hat{\theta}} = \begin{bmatrix} \dot{\theta}_x - \sin(\theta_y) \cdot \dot{\theta}_z \\ \cos(\theta_x) \cdot \dot{\theta}_y + \cos(\theta_y) \cdot \sin(\theta_x) \cdot \dot{\theta}_z \\ -\sin(\theta_x) \cdot \dot{\theta}_y + \cos(\theta_x) \cdot \cos(\theta_y) \cdot \dot{\theta}_z \end{bmatrix} \quad (5)$$

Die relativen Geschwindigkeiten der Antriebsräder sind im Koordinatensystem BCS^b direkt mit der Win-

kelgeschwindigkeit $\dot{\psi}_i$ des Motors gekoppelt. Somit ergibt sich für alle Räder i mit $i = 1, 2, 3$:

$$\underline{\underline{b\omega_w}} = \dot{\psi}_i \quad (6)$$

Die absoluten Geschwindigkeiten der Räder können durch Bindungsgleichung in Abhängigkeit der jeweiligen Positionen der einzelnen Koordinatensysteme gefunden werden. Für die relative Geschwindigkeit der Kugel im BCS^b ergibt sich:

$$\underline{\underline{b\omega_s}} = \underline{\underline{R_{bl}}} \cdot \underline{\underline{s\Omega_s}} - \underline{\underline{b\Omega_b}} \quad (7)$$

Nach Abbildung 3b ergibt sich die Bedingung, dass kein Schlupf zwischen Kugel und Rad vorhanden ist, sodass folgende allgemeine Gleichung für alle Räder i mit $i = 1, 2, 3$ gilt:

$$\left(\underline{\underline{b\omega_s}} \times \begin{bmatrix} R_s \cdot \sin(\varepsilon) \cdot \cos(\gamma_i) \\ R_s \cdot \sin(\varepsilon) \cdot \sin(\gamma_i) \\ R_s \cdot \cos(\varepsilon) \end{bmatrix} \right) \cdot \begin{bmatrix} -\sin(\gamma_i) \\ \cos(\gamma_i) \\ 0 \end{bmatrix} = \underline{\underline{b\omega_w}} \cdot R_w \quad (8)$$

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

$$\underline{\underline{b\Omega_w}} = \underline{\underline{b\Omega_b}} + \begin{bmatrix} \sin(\varepsilon) \cdot \cos(\gamma_i) \\ \sin(\varepsilon) \cdot \sin(\gamma_i) \\ -\cos(\varepsilon) \end{bmatrix}^T \cdot \underline{\underline{b\Omega_b}} \quad (9)$$

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

2.2 Dynamikfunktionen

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 [2]. Die Dynamik des S-Mobile setzt sich aus der Dynamik der Tragstruktur (Aufbau, Räder, Antriebskugel), der Aktoren, der Sensoren und der Informationsverarbeitung zusammen. Mit Hilfe geeigneter Vereinfachungen wird ein räumliches Modell des S-Mobile erstellt.

Dynamik der mechanischen Tragstruktur

Hier sollen die Differentialgleichungen, die die dynamischen Vorgänge des Systems beschreiben, mit Hilfe des Lagrange Formalismus hergeleitet werden.

Diese Methode verwendet die Erhaltung der Energie im System zur Herleitung der kraft- und momentabhängigen Bewegungsgleichungen. Es muss folgende Gleichung gelöst werden:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\vec{q}}} \right) - \left(\frac{\partial T}{\partial \vec{q}} \right) + \left(\frac{\partial V}{\partial \vec{q}} \right) - f_{NP} = 0 \quad (10)$$

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

$$\vec{q} = [\vartheta_x \quad \vartheta_y \quad \vartheta_z \quad \varphi_x \quad \varphi_y]^T \quad (11)$$

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

$$T_s = \frac{1}{2} \cdot m_s \cdot \underline{\dot{r}}_s^T \cdot \underline{\dot{r}}_s + \frac{1}{2} \cdot \underline{s\Omega}_s^T \cdot \underline{\Theta}_s \cdot \underline{s\Omega}_s \quad (12)$$

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

$$T_b = \frac{1}{2} \cdot m_b \cdot \underline{\dot{r}}_b^T \cdot \underline{\dot{r}}_b + \frac{1}{2} \cdot \underline{b\Omega}_b^T \cdot \underline{\Theta}_b \cdot \underline{b\Omega}_b + T_{coup} \quad (13)$$

$$T_{coup} = m_b \cdot \left(\underline{R}_{bl} \cdot \underline{\dot{r}}_b \right) \cdot \left(\underline{b\Omega}_b \times \begin{bmatrix} 0 \\ 0 \\ l \end{bmatrix} \right) \quad (14)$$

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

$$T_{w,i} = \frac{1}{2} \cdot \underline{\Theta}_{w,i} \cdot \underline{b\Omega}_w^2 \quad \text{mit } i = 1,2,3 \quad (15)$$

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

$$T = T_s + T_b + T_{w,1} + T_{w,2} + T_{w,3} \quad (16)$$

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

$$V = V_b = m_b \cdot \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \cdot \underline{R}_{bl} \cdot \begin{bmatrix} 0 \\ 0 \\ l \end{bmatrix} \quad (17)$$

Die nicht konservativen Kräfte bestehen aus den durch die Motoren erzeugten Drehmomenten M_1 , M_2 und M_3 und deren reaktiven Gegenmomenten $M_{C,1}$, $M_{C,2}$ und $M_{C,3}$. Da die Minimalkoordinaten \vec{q} des Modells nur Winkel in den BCS^b und BCS^s enthalten, müssen die Drehmomente über Jacobi-Matrizen umgerechnet werden.

$$\underline{f}_{NP} = \sum_{i=1}^3 ({}^i J_T^T \cdot M_i) + J_{TC}^T \cdot \sum_{i=1}^3 (M_{C,i}) \quad (18)$$

Die Gegenmomente sind nach Abbildung 3c definiert nach:

$$M_{C,i} = \begin{bmatrix} \sin(\varepsilon) \cdot \cos(\gamma_i) \\ \sin(\varepsilon) \cdot \sin(\gamma_i) \\ \cos(\varepsilon) \\ 0 \\ 0 \end{bmatrix} \cdot -M_i \quad \text{mit } i = 1,2,3 \quad (19)$$

Die entsprechende Jacobi-Matrix zur Umrechnung der Motordrehmomente lässt sich durch Lösen der folgenden Gleichung durch polynomiale Separation finden:

$$\underline{b\Omega}_w = \underline{J}_T \cdot \underline{\dot{q}} \quad (20)$$

Die Jacobi-Matrix zur Umrechnung der reaktiven Gegenmomente ist der Jacobi-Matrix zur Berechnung der absoluten Winkelgeschwindigkeit des Aufbaus gleich, nur dass sie auf eine 5×5 Matrix erweitert werden muss. Somit ergibt sie sich zu:

$$J_{TC} = \begin{bmatrix} 1 & 0 & -\sin(\theta_y) & 0 & 0 \\ 0 & \cos(\theta_x) & \cos(\theta_y) \cdot \sin(\theta_x) & 0 & 0 \\ 0 & -\sin(\theta_x) & \cos(\theta_x) \cdot \cos(\theta_y) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (21)$$

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

Dynamik des Aktuators

Die Aktuatoren des Systems sind Synchronmotoren. Das maximale Drehmoment liegt bei dieser Motorenbauart an, wenn ein 90° vorauseilendes Drehfeld generiert wird. Zur Auslegung der Regelung wird das Dreiphasensystem mit den Strömen i_1 , i_2 und i_3 in das vom Polradwinkel abhängige Zweiphasensystem mit den Strömen i_d und i_q überführt.

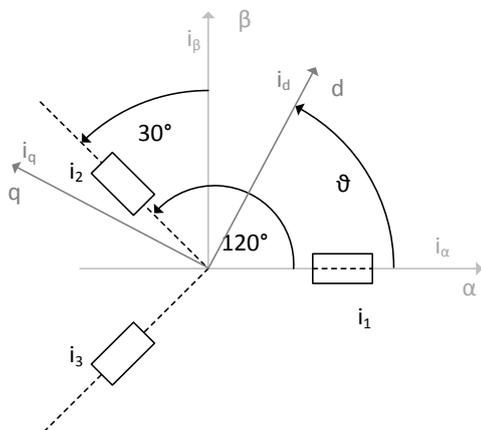


Abbildung 3. Park- und Clarke- Transformation

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

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} \quad (22)$$

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

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) \\ \sin(\vartheta) & \cos(\vartheta) \end{bmatrix} \cdot \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (23)$$

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

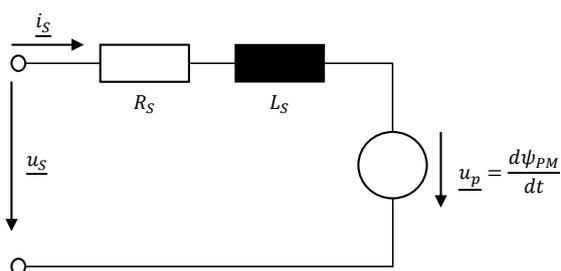


Abbildung 4. Elektrisches Ersatzmodell

Dieses ist im Wesentlichen ein elektrischer Gleichstromkreis bestehend aus der Motorinduktivität L_S und dem Strangwiderstand R_S . Die Strandspannung u_S ist die Spannungsquelle aus der der Strangstrom i_S

resultiert. Die Polradspannung u_p wirkt als Gegeninduktionsspannung.

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

$$u_d = R_s \cdot i_d + L_d \cdot \frac{di_d}{dt} - \psi_y \cdot L_q \cdot i_q \quad (24)$$

$$u_q = R_s \cdot i_q + L_q \cdot \frac{di_q}{dt} + \psi_y \cdot (L_d \cdot i_d + \phi_0) \quad (25)$$

Innerhalb der Regelung kann nun das gewünschte Motormoment durch Vorgabe von i_q eingestellt werden. Zur Bildung des Momentes der Synchronmaschine gilt nach [6] folgender Zusammenhang:

$$M_{1,2,3} = \frac{3}{2} \cdot p \cdot (\phi_0 \cdot i_q + (L_d - L_q) \cdot i_d \cdot i_q) \quad (26)$$

Weiterhin wird eine Entkopplung der beiden elektrischen Teilsysteme vorgenommen. Anschließend werden die Spannungen durch inverse Park- und Clarke-Transformationen in die Spannungen U_1 , U_2 und U_3 zurückgeführt und über die Leistungselektronik gestellt.

Dynamik der Sensoren und der Informationsverarbeitung

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

Die digitale Realisierung eines Reglers führt zu einer Phasenabsenkung. Diese resultiert aus den auftretenden Totzeiten, die durch den Abtast-Haltvorgang und die zur Berechnung des Regelalgorithmus notwendige Bearbeitungszeit verursacht werden. Dies kann durch ein System mit endlich vielen Zustandsgleichungen approximiert werden. Zur Nachbildung dieser digitalen Effekte wird eine Padé-Approximation 2. Ordnung im Modell implementiert [3].

3 Modellbasierte Reglerauslegung

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

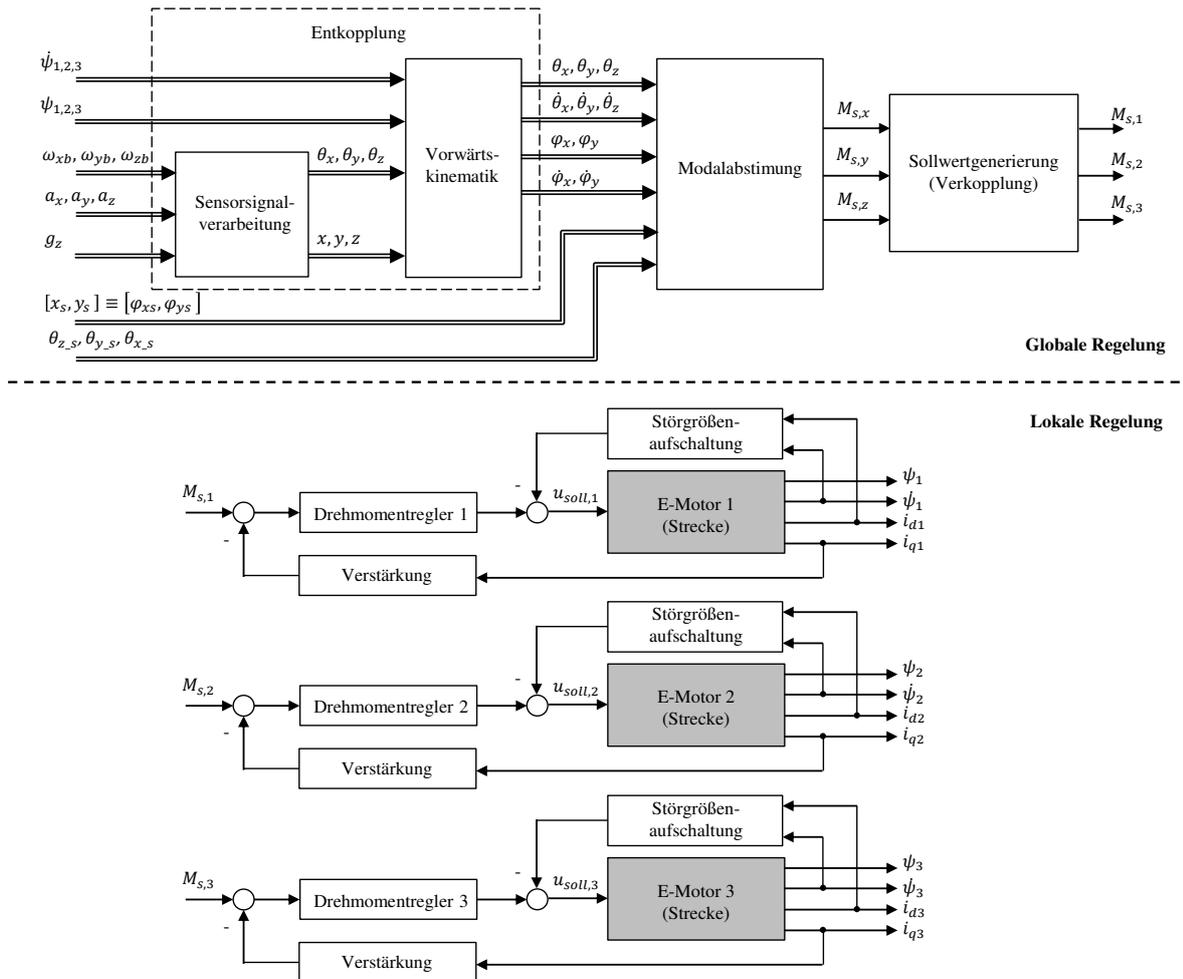


Abbildung 5. Regelstruktur der Antriebseinheit

3.1 Globaler Regler

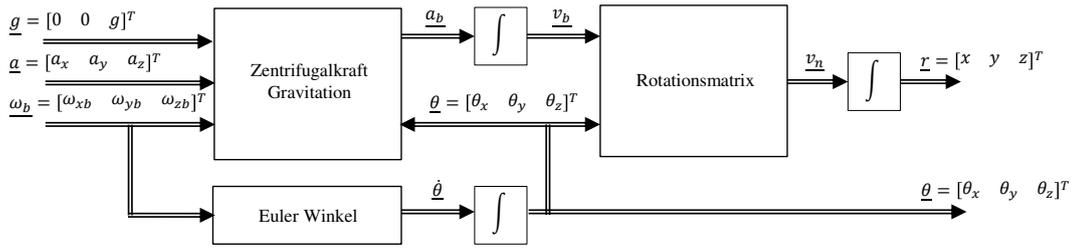
Durch die physikalische Beschaffenheit des Systems sind die Regelgrößen der Freiheitsgrade sehr stark miteinander Verkopfelt. Um nun den Einsatz der angestrebten Aufbauregler [4] zu gewährleisten, muss das vermaschte Gesamtsystem durch *Entkopplung* mittel Vorwärtskinematik in schwach gekoppelte Teilsysteme zerlegt werden. Diesen soll dann nachfolgend durch die *Modalabstimmung* die gewünschte Systemdynamik aufgeprägt werden. Durch abschließende *Verkopplung* basierend auf Rückwärtskinematik werden die Stellsignale für die untergeordneten lokalen Regler aus den gefundenen Modalkräften generiert.

Entkopplung

Die Herleitung der für die Modalregelung notwendigen kinematischen Eingangsgrößen des S-Mobiles

erfolgt durch Koordinatentransformationen im Block *Entkopplung*, die aus zwei weiteren Funktionen besteht: *Sensorsignalverarbeitung* und *Vorwärtskinematik*.

Zur Ermittlung der aktuellen Lageinformationen des S-Mobile ist der Einsatz eines Sensorclusters, bestehend aus Gyroskop, Accelerometer und Magnetometer vorgesehen. Zudem sollen Drehzahl und Drehwinkel der Antriebsmotoren über Encoder ermittelt werden. Die gemessenen Daten der Sensorik benötigen zudem Koordinatentransformationen, da sich die Sensoren nicht im Aufbauschwerpunkt befinden. Folglich müssen so erfassbaren Messgrößen zunächst über eine *Sensorsignalverarbeitung* auf geforderte systemrelevante Größen konvertiert werden. Die Abbildung 6 stellt die Funktion der Sensorsignalverarbeitung im Detail dar.


Abbildung 6. Sensorsignalverarbeitung

Zunächst wird aus den Daten des Gyroskops, Accelerometers und Magnetometers des Sensorclusters ein korrigierter Beschleunigungsvektor errechnet, der die Aufbaubewegung im Sensorkoordinatensystem beschreibt.

$$\underline{a}_b = \begin{bmatrix} a_{xb} \\ a_{yb} \\ a_{zb} \end{bmatrix} = \underline{a} - \underline{\omega}_b \times \underline{v}_b + \underline{g} \cdot \begin{bmatrix} -\sin(\theta_y) \\ \cos(\theta_y) \cdot \sin(\theta_x) \\ \cos(\theta_y) \cdot \cos(\theta_x) \end{bmatrix} \quad (27)$$

Mit den Vektoren:

$$\underline{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}, \underline{g} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \underline{\omega}_b = \begin{bmatrix} \omega_{xb} \\ \omega_{yb} \\ \omega_{zb} \end{bmatrix}, \underline{v}_b = \begin{bmatrix} v_{xb} \\ v_{yb} \\ v_{zb} \end{bmatrix} \quad (28)$$

Der Geschwindigkeitsvektor der Aufbaubewegung wird über kontinuierliche Integration des resultierenden Beschleunigungsvektors im Sensorkoordinatensystem nach folgendem Zusammenhang kalkuliert:

$$\underline{v}_b = \int_0^t \underline{a}_b d\tau + \underline{v}_b|_{t=0} \quad (29)$$

Die zur Transformation des Geschwindigkeitsvektors in das Referenzkoordinatensystem nötigen Euler-Winkel werden durch Integration der gemessenen Rotationsgeschwindigkeiten nach [5] bestimmt:

$$\begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix} = \begin{bmatrix} 1 & \frac{\sin(\theta_y) \cdot \sin(\theta_x)}{\cos(\theta_y)} & \frac{\sin(\theta_y) \cdot \cos(\theta_x)}{\cos(\theta_y)} \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \frac{\sin(\theta_x)}{\cos(\theta_y)} & \frac{\cos(\theta_x)}{\cos(\theta_y)} \end{bmatrix} \cdot \begin{bmatrix} \omega_{xb} \\ \omega_{yb} \\ \omega_{zb} \end{bmatrix} \quad (30)$$

Die Transformation vom Sensor- zum ortsfesten Referenzkoordinatensystem kann somit durch die folgende Transformationsmatrix erfolgen:

$$\begin{bmatrix} v_{xn} \\ v_{yn} \\ v_{zn} \end{bmatrix} = \underline{R}_{bl} \cdot \begin{bmatrix} v_{xb} \\ v_{yb} \\ v_{zb} \end{bmatrix} \quad (31)$$

Durch Integration erhält man somit die zu regelnden Aufbauwinkel, sowie die Position der Maschine im Referenzkoordinatensystem.

Die Winkel der Antriebskugel werden über kinematische Beziehungen aus den Sensordaten der Motorencoder gewonnen. Die Drehzahlen und Winkel der Motoren werden vom radfesten Koordinatensystem BCS^w zum Koordinatensystem der Antriebskugel BCS^s transformiert. Da die Winkel der Kugel direkt

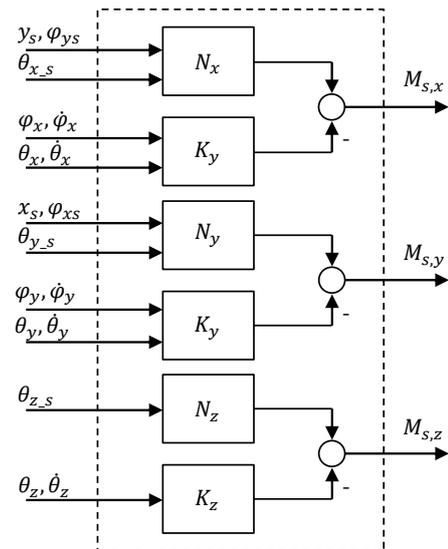
über deren Radius in Positionen umgerechnet werden können, ist somit auch die Lage der Kugel bekannt. Die Lage des Aufbaus ist durch die feste Transformation vom BCS^s auf das BCS^b mit der Lage der Kugel korrelierend.

Die so erhaltenen Daten aus der *Sensorsignalverarbeitung* werden anschließend in dem Funktionsblock *Vorwärtskinematik* genutzt, um die gekoppelten Systemzustände in quasi-modale Freiheitsgrade des Aufbaus zu entkoppeln. Dazu dienen die in Kapitel 2.1 symbolisch hergeleiteten Vorwärtskinematikfunktionen.

Die Ergebnisse der Entkopplung werden der *Modalabstimmung* zugeführt.

Modalabstimmung

Um sowohl Aufbau- und Kugelwinkel als auch deren Position im Raum zu regeln wird ein Mehrgrößenregler genutzt. Dieser soll als linear quadratisch optimaler Regler ausgeführt werden. Zu diesem Zweck wird eine Zustandsrückführung nach Abbildung 7 entworfen.


Abbildung 7. Blockschaltbild des LQ-Reglers

Allgemein gilt es nun das Gütekriterium nach Ricatti zu optimieren. Zu diesem Zweck soll der Regelfehler e , durch Betrachtung des Zustandsvektors x (mit

$y = C \cdot x$ und $e = w - y$ bei $w = 0$), und der Stellaufwand durch den Eingangsvektor u berücksichtigt werden. Es gilt die folgende Gleichung zu minimieren [7]:

$$J_{LQ} = \int_{t_0}^{t_e} \left(\underline{\underline{x}}^T(t) \cdot \underline{\underline{Q}} \cdot \underline{\underline{x}}(t) + \underline{\underline{u}}^T(t) \cdot \underline{\underline{R}} \cdot \underline{\underline{u}}(t) \right) dt \quad (32)$$

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

$$Q_{ii} = \frac{1}{\text{maximal akzeptierbare GröÙe } x_i^2} \text{ mit } i \in \{1, 2, \dots, l\} \quad (33)$$

$$R_{jj} = \frac{1}{\text{maximal akzeptierbare GröÙe } u_j^2} \text{ mit } j \in \{1, 2, \dots, m\} \quad (34)$$

Was dem folgenden Kriterium entspricht:

$$J_{LQR} = \int_0^\infty \left(\sum_{i=1}^l Q_{ii} \cdot x_i(t)^2 + \sum_{j=1}^m R_{jj} \cdot u_j(t)^2 \right) dt \quad (35)$$

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

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

$$\underline{\underline{K}} = \underline{\underline{R}}^{-1} \cdot \underline{\underline{B}}^T \cdot \underline{\underline{P}} \quad (36)$$

Mit der Matrix P als Lösung der Riccati-Gleichung:

$$0 = \underline{\underline{A}} \cdot \underline{\underline{P}} - \underline{\underline{P}} \cdot \underline{\underline{B}} \cdot \underline{\underline{R}}^{-1} \cdot \underline{\underline{B}}^T \cdot \underline{\underline{P}} + \underline{\underline{Q}} + \underline{\underline{A}} \cdot \underline{\underline{P}} \quad (37)$$

Daraus ergibt sich das Stellgesetz:

$$\underline{\underline{u}}^* = -\underline{\underline{K}} \cdot \underline{\underline{x}} \quad (38)$$

Da diese Gleichungen ein Optimierungsproblem darstellen, müssen diese mit Hilfe numerischer Methoden berechnet werden. Die Lösung ist nur in Ausnahmefällen und bei sehr einfachen Modellen algebraisch möglich.

Zur Verbesserung des stationären Verhaltens muss eine Vorsteuerung mit der Verstärkung N ausgelegt werden:

$$\underline{\underline{N}} = -\left(\underline{\underline{C}} \left(\underline{\underline{A}} - \underline{\underline{B}} \cdot \underline{\underline{K}} \right)^{-1} \underline{\underline{B}} \right)^{-1} \quad (39)$$

Da die aufgeschaltete Matrix der Inversen der stationären Verstärkung entspricht, wird durch sie nun der Zustands-Regler stationär genau. [9]

Verkopplung

Die Verkopplung dient zur Generierung der Sollmomente für die unterlagerte Momentenregelung des Aktuators. Daher werden die Sollmomente, die von den zweidimensionalen LQ-Reglern für die drei Raumebenen generiert werden, auf die Positionen der drei Antriebsräder transformiert. Die folgende Abbildung 8 zeigt die allgemeine räumliche Anordnung der Antriebsräder und deren Winkelbeziehungen in der xy -Ebene. Die Drehmomente zur Bewegung der Maschine auf der Ebene werden durch die Größen M_y und M_x bezeichnet. Die Antriebsräder können auf einem festen Radius, vorgegeben durch die Projektion des Antriebsballs in dieser Ebene, in einem bestimmten Winkel β zum Koordinatensystem und einem Winkel γ zueinander stehen.

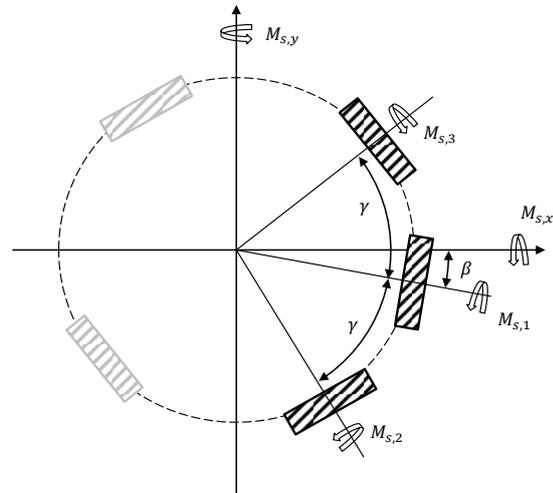


Abbildung 8. Geometrische Momentenverkopplung

Aus dieser Darstellung können zunächst allgemein die Zusammenhänge für die von den Motoren aufzubringenden Drehmomente $M_{s,1}$, $M_{s,2}$, $M_{s,3}$ zur Erzeugung der Momente $M_{s,x}$, $M_{s,y}$ abgeleitet werden:

$$M_{s,1} = \frac{-\cos(\beta) \cdot M_{s,x} + \sin(\beta) \cdot M_{s,y}}{\cos(\gamma) - 1} \quad (40)$$

$$M_{s,2} = \frac{\sin(\beta - \gamma) \cdot M_{s,x} - \sin(\beta) \cdot M_{s,x} + \cos(\beta - \gamma) \cdot M_{s,y} - \cos(\beta) \cdot M_{s,y}}{-\sin(2\gamma) + 2\sin(\gamma)} \quad (41)$$

$$M_{s,3} = \frac{\sin(\beta + \gamma) \cdot M_{s,x} - \sin(\beta) \cdot M_{s,x} + \cos(\beta + \gamma) \cdot M_{s,y} - \cos(\beta) \cdot M_{s,y}}{\sin(2\gamma) - 2\sin(\gamma)} \quad (42)$$

Wird für diese Gleichung die vorgesehene Antriebskonfiguration mit $\gamma = 120^\circ$ eingesetzt und eine

Projektion auf die Antriebskugel um den Winkel ε vorgenommen, so ergibt sich für die Vorwärtskinematik der Zusammenhang:

$$\begin{bmatrix} M_{s,1} \\ M_{s,2} \\ M_{s,3} \end{bmatrix} = \begin{bmatrix} \frac{2 \cdot \cos(\beta)}{3 \cdot \cos(\varepsilon)} & -\frac{2 \cdot \sin(\beta)}{3 \cdot \cos(\varepsilon)} & \frac{1}{3} \\ -\frac{\sqrt{3} \cdot \sin(\beta) - \cos(\beta)}{3 \cdot \cos(\varepsilon)} & \frac{\sin(\beta) - \sqrt{3} \cdot \cos(\beta)}{3 \cdot \cos(\varepsilon)} & \frac{1}{3} \\ \frac{\sqrt{3} \cdot \sin(\beta) + \cos(\beta)}{3 \cdot \cos(\varepsilon)} & \frac{\sin(\beta) + \sqrt{3} \cdot \cos(\beta)}{3 \cdot \cos(\varepsilon)} & \frac{1}{3} \end{bmatrix} \cdot \begin{bmatrix} M_{s,x} \\ M_{s,y} \\ M_{s,z} \end{bmatrix} \quad (43)$$

Mit diesen Transformationen können nun die linearen Regler auf die jeweiligen Raumebenen einwirken.

3.2 Lokaler Regler

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

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

Durch die Induktion ist das Motormodell jedoch nichtlinear, wodurch der Einsatz der vorgesehen linearen SISO-Regler nicht ohne weiteres möglich ist. Die Störgrößen müssen durch eine Störgrößenaufschaltung kompensiert werden. Die Kompensationspannung wird dann zu der Stellgröße des Reglers addiert. Durch diese Maßnahme kann das einsträngige Ersatzmodell nun als PT1 System angesehen werden. Für dieses System sieht die Regeltheorie bei Einsatz eines PI-Reglers ein gutes Führungsverhalten ohne Regelabweichung voraus.

Das Regelgesetz ergibt sich dementsprechend zu:

$$u_d = K_p \cdot (0 - i_d) + \int K_i \cdot (0 - i_d) dt - p \cdot \psi_y \cdot L_d \cdot i_d \quad (44)$$

$$u_q = K_p \cdot (M_{soll} - i_q \cdot c_{mech}) + \int K_i \cdot (M_{soll} - i_q \cdot c_{mech}) dt + p \cdot \psi_y \cdot (L_d \cdot i_d + \phi_0) \quad (45)$$

Mit der Verstärkung:

$$c_{mech} = \frac{3}{2} \cdot p \cdot \phi_0 \quad (46)$$

Die Reglerparameter K_p und K_i wurden modellbasiert eingestellt. Durch Vorgabe eines Toleranzbandes des

tolerierbaren Überschwingens wurde eine Lehrsche Dämpfung der Systemantwort gewählt. Mit dem nach Kompensation anwendbaren PT1-Verhalten wurden die Parameter K_p und K_i ermittelt.

4 Funktionsabsicherung mittels MiL

Zur Funktionsabsicherung wird Modell-in-the-Loop-Simulation herangezogen werden.

Ziel der Aufbauregelung ist primär ein Verfahren des S-Mobile bei gleichzeitiger Stabilisierung des Aufbaus zu ermöglichen. Zu diesem Zweck wurde ein Positionssprung als Sollwert auf den Regler gegeben. Abbildung 9 stellt die Sprungantwort des Systems dar.

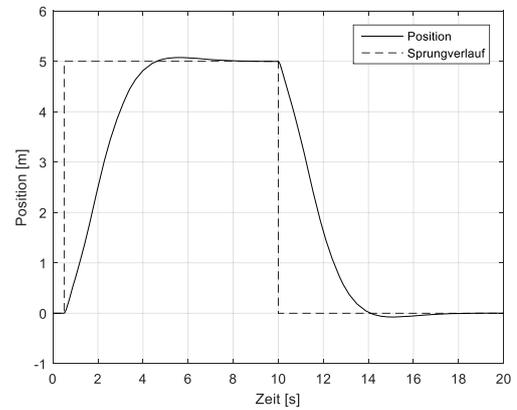


Abbildung 9. Positionssollwertsprung

Der Sprung erfolgt bei 0,5s auf einen Vorgabewert von $x_s = 5\text{m}$. Die Maschine folgt diesem Sprung mit einem kontinuierlichen Anstieg der Position und einem leichten Überschwingen, welches der Ausregelung des Aufbauwinkels dient. Nach 10s springt der Vorgabewert auf $x_s = 0\text{m}$, also auf seinen Ausgangspunkt zurück. Die Maschine folgt dem Sprung zurück auf gleicher Weise.

Der Verlauf der Bewegung des Aufbaus wird in Abbildung 10 dargestellt.

Hier ist der Aufbauwinkel in Relation zum Sollwertsprung aufgetragen. Es ist zu erkennen, dass der Aufbauwinkel zur Stabilisierung jeweils in die entgegengesetzte Fahrtrichtung geneigt wird. Erreicht die Maschine ihre Soll-Position wird der Aufbauwinkel wieder zurück auf seinen festen Sollwert von 0° gestellt.

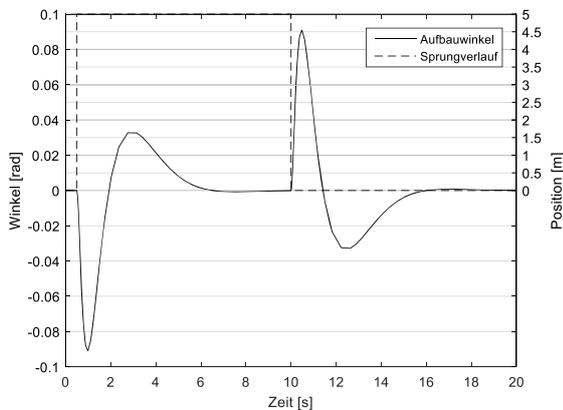


Abbildung 10. Aufbaubewegung als Aufbauwinkel

5 Resümee

In dem vorliegenden Beitrag wurde eine hierarchische Reglerstruktur, bestehend aus einem globalen Regler und einer untergeordneten lokalen Regelung, modellbasiert entworfen. Zu diesem Zweck wurden die kinematischen und dynamischen Funktionen des mechatronischen Systems untersucht. Darauf basierend wurde die Reglerstruktur entwickelt.

In der weiteren Forschungsarbeit sollen die Funktionen mittels SiL und HiL weitergehend validiert und optimiert werden. Darüber hinaus ist ein Demonstrator vorgesehen. Dazu werden auf Basis der durchgeführten MiL-/SiL und HiL-Simulation die Komponenten des Systems ausgelegt. Durch die iterative Anpassung der realen Konstruktion und Nachführen der Parameter des Systems, unterstützend durch eine enge Verflechtung von Simulation und Messung am realen System, sollen so die realen Komponenten einschließlich des Systems sukzessiv realisiert werden. Ein typisches Beispiel hierfür ist die geeigneten Auswahl der Aktoren zu nennen. Darüber hinaus kann aufgrund der ermittelten, auftretenden Drehmomente und Kräfte der Aufbau konstruiert und unter Leichtbauaspekten optimiert werden.

6 Referenzen

- [1] Göllner, M.; Liu-Henke, X.: *Konzeption eines hochdynamischen Systems mit sphärischem Elektroantrieb*, ASIM STS/GMMS Workshop 2014, Reutlingen, 2014
- [2] Liu-Henke, X.; Lückel, J.; Jäker, K.-P.: *Ganzheitlicher mechatronischer Entwurf eines aktiven Feder-/Neigemoduls*. VDI-Tagung Mechatronik – Mechanisch/Elektrische Antriebstechnik, in Wiesloch, Deutschland, 2000
- [3] Liu-Henke, X.; Buchta, R.; Quantmeyer, F.: *Simulation eines mechatronischen Lenkungsmoduls für ein Elektrofahrzeug mit dezentralen Direktantrieben*. ASIM, in Krefeld, Deutschland, 2011
- [4] Liu-Henke, X.: *Mechatronische Entwicklung der aktiven Feder-/Neigetechnik für das Schienenfahrzeug RailCab*. VDI- Fortschritt-Berichte, Reihe 12, Nr. 589, VDI-Verlage, Düsseldorf, 2004
- [5] Dorobantu, R.: *Simulation des Verhaltens eines low-cost Strapdown-IMU unter Laborbedingungen*. TU München, Deutschland, 1999.
- [6] Sahhary, B.: *Elektrische Antriebe mit dauermagnetenerregten Maschinen im dynamischen sensorlosen Betrieb*. Dissertation, Universität der Bundeswehr, Deutschland, 2008.
- [7] Blath, J.: *Zustandsregelungen*. TU Clausthal, Deutschland, 2012
- [8] Hespanha, J.: *LQG/LQR Controller Design*. University of California, USA, 2007
- [9] Holzhütter, T.: *Zustandsregelung*. Fachhochschule Hamburg, Deutschland, 2009
- [10] Göllner, M.; Liu-Henke, X.: *Mathematical derivation of the geometry of a Mecanum-wheel for a size exact roll off on a spherical surface*. MSM. Opole, Polen, 2014

Sitzung 2

Raum 105

Simulation von Steuerungen / Robotik

(Sitzungsleitung: Thorsten Pawletta)

Objektorientierte Entwicklung einer hybriden Materialflusssimulation eines NE-Schmelz- und Druckgussbetriebes

Dominik Jeckle

Hochschule Ansbach

MATLAB/Simulink Based Rapid Control Prototyping for Multivendor

Robot Applications

Christina Deatcu

Hochschule Wismar

Multi-Robotersteuerungen mit variablen Interaktionsprinzipien auf Basis des Simulation Based Control Frameworks und dem Discrete Event System Spezification Formalismus

Birger Freymann

Hochschule Wismar

Anwendung einer iterativ lernenden Regelung auf einen Service Roboter mit Mehrkörperkinematik

Sandra Baßler

Hochschule Wismar

Objektorientierte Entwicklung einer hybriden Materialflusssimulation eines NE-Schmelz- und Druckgussbetriebes

B.Eng Dominik Jeckle, Prof. Dr. Wolfgang Schlüter, Dipl. Ing. Ansgar Ringleb¹

¹HS-Ansbach

d.jeckle@hs-ansbach.de

Die vorliegende Studie zeigt die systematische Entwicklung einer hybriden Materialflusssimulation eines NE-Schmelz- und Druckgussbetriebes mit den Werkzeugen MATLAB, Simulink und Stateflow. Die zentralen Komponenten Schmelzöfen, Transportfahrzeuge und Druckgussmaschinen wurden in einer vorausgehenden Prozessanalyse in einem Referenzbetrieb eingehend untersucht. Dabei wurde deutlich, dass kontinuierliche und ereignisdiskrete Prozesse sowie eine Vielzahl unterschiedlicher Prozessparameter vorliegen. Des Weiteren zeigte sich, dass die Transportfahrzeuge als zentrale Elemente den Materialaustausch sicherstellen.

Die Entwicklung der Simulation beinhaltet die folgenden Punkte:

1. Definition der Komponentenklassen mit Parametern und Funktionalität
2. Entwicklung der Zustandsautomaten für die einzelnen Komponenten
3. Kopplung der Zustandsautomaten mit einem übergeordneten Event-Handling
4. Ausarbeitung einer modular aufgebauten Steuerung

Ein Problem stellt dabei die Kommunikation zwischen der objektorientiert aufgebauten und global definierten Anlagenstruktur in MATLAB und den ereignisorientierten Zustandsautomaten der einzelnen Komponenten in Stateflow dar. Die Übergabe der Daten wird, in Simulink mittels Interpreted MATLAB Functions realisiert.

In der modular aufgebauten Steuerung wird der Einfluss verschiedener Entnahme- und Befüllregeln der Schmelzöfen und Druckgussanlagen unter Berücksichtigung typischer Nachfragesituationen auf den Materialfluss und den Lagerbestand untersucht, um damit den Materialfluss zu optimieren und die Prozesssicherheit zu erhöhen..

1 Einleitung

Einer der größten Endenergieverbraucher in Deutschland ist, der Industriesektor mit 29 % am Gesamtendenergieverbrauch [1]. Zukünftig ist zu erwarten, dass der Energiekonsum sowie die Energiepreise weiterhin ansteigen werden [2], weshalb es für Unternehmen immer wichtiger wird, ihr Energieeinsparpotential zu identifizieren und zu heben. In der energieintensiven (NE)-Schmelz- und Druckgussindustrie wurde in den letzten beiden Jahrzehnten der Energieverbrauch kontinuierlich reduziert [3]. Unter anderem wurde in [4] die effiziente Energienutzung in Metall-Schmelzbetrieben untersucht. Dabei wurden erste

Schwachstellen identifiziert und Maßnahmen zur Reduzierung des Gesamtenergieverbrauchs vorgeschlagen. Auf die Reduzierung des Energieverbrauchs durch Optimierung von Betriebsabläufen und Betriebssteuerung wurde nicht eingegangen. In [5] wurde gezeigt, dass durch die Verwendung von zugeliefertem flüssigem Aluminium der Gesamtenergieverbrauch zum Schmelzen von Aluminium gesenkt werden kann. Allerdings sprechen dagegen wirtschaftliche Gründe, weshalb die meisten Schmelz- und Druckgussbetrieben festes Aluminium verwenden. Ein weiterer Ansatz, um den Energieverbrauch in Schmelz- und Druckgussbetrieben zu senken, ist die

Betrachtung der gesamten Prozesskette von der Metallbereitstellung bis zur mechanischen Bearbeitung des Gussteils. In dem Projekt ProGress wurde dieser Ansatz mittels energieorientierter hybrider Produktionssystemsimulation verfolgt. Aus den vorgestellten Forschungsergebnissen [6] geht hervor, dass es aufgrund der kleinen Betriebsgröße (ein Schmelzofen, drei Aluminiumverbraucher) nicht möglich war, Steuerungsstrategien für einen effizienten Betrieb zu entwickeln. [7,8] zeigen auf, dass durch den Einsatz verschiedener Steuerungskonzepte ein nennenswertes Potential in der Steigerung der Robustheit und Energieeffizienz liegt. Wobei in [7] der Schwerpunkt auf der Implementierung einer holonischen Steuerungsarchitektur liegt und dabei keine Erkenntnisse über das Systemverhalten von Schmelz- und Druckgussbetrieben gewonnen werden. In [8] wird ein Konzept vorgestellt um verbraucherübergreifenden Energieeinsparpotentiale zu erzielen. Die dargelegten Ansätze und Ideen sind allgemein gehalten und auf eine Vielzahl von Betrieben der produzierenden Industrie anwendbar. Eine Anwendbarkeit auf Schmelz- und Druckgussbetriebe ist noch zu untersuchen.

Das Projekt „Smart-Melting“ des Forschungsverbundes Green Factory Bavaria hat das Ziel der Entwicklung eines Prozessmanagementsystems. Mit dessen Hilfen soll ein energieeffizienter Betrieb der Schmelzöfen in der (NE)-Schmelz- und Druckgussindustrie ermöglicht und die Prozesssicherheit des Gesamtbetriebes erhöht werden. Dazu ist es erforderlich, die vorgeschaltete Aluminiumanlieferung (Fest- und Flüssigmetall) ebenso wie die nachgeschalteten Prozesse der Metallaufbereitung der internen Disposition und des Druckgusses zu berücksichtigen. Da eine Entwicklung an Anlagen im laufenden Betrieb nicht möglich ist, wird eine Materialfluss- und Energieflusssimulation notwendig, welche die Prozess- und Anlagenstruktur derartiger Gießereien realitätsgetreu abbildet. Diese Arbeit stellt die Entwicklung der Materialflusssimulation unter Verwendung der Tools Matlab, Simulink und Stateflow vor. Die Simulation ist modular aufgebaut und kann auf andere Betriebe adaptiert werden. Dazu müssen die Steuerung und die Komponenten vor Simulationsbeginn konfiguriert werden. Durch die simulationsgestützte Untersuchungen soll eine geeignete Betriebsführung in den jeweiligen Lastfällen (Überlast, Unterlast, Normalbetrieb) ermittelt werden, um den Materialfluss zu optimieren und die Prozesssicherheit zu erhöhen.

2 Aufbau eines realen NE-Schmelz- und Druckgussbetrieb

Charakteristisch für den Fertigungsprozess von Schmelz –und Druckgussbetrieben, welche der Hybridindustrie zugeordnet werden können, ist, dass hier sowohl kontinuierliche (Schmelzen) als auch diskrete (Gießen) Prozessschritte vorliegen. Um diese komplexen Prozesse in einer Simulation abbilden zu können ist eine umfassende Analyse der betrieblichen Strukturen notwendig. Zum Visualisieren des Material- und Informationsflusses eignet sich dabei die Wertstromanalyse [9].

2.1 Materialfluss

Der betrachtete Referenzbetrieb weist einen Materialkreislauf auf der typisch für NE- Schmelz- und Druckgussbetriebe ist.

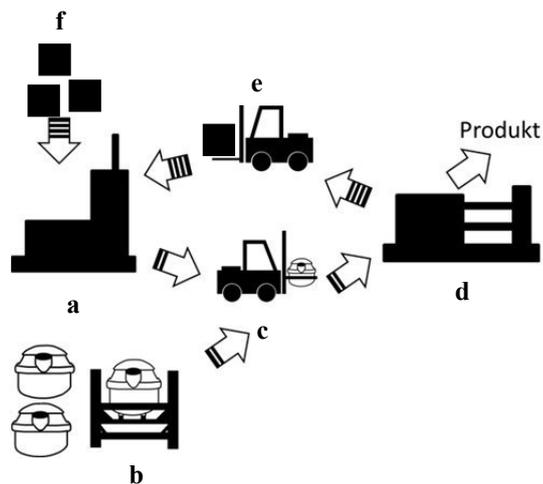


Abbildung 1. Vereinfachte Darstellung des Materialflusses eines NE-Schmelz- und Druckgussbetriebs

Die Transporter (b) fahren flüssiges Aluminium von den Schmelzöfen (a) zu den Druckgussanlagen (d). Dort wird das flüssige Aluminium durch das Druckgussverfahren zum Produkt verarbeitet. Als Nebenprodukt fällt recyclingfähiges festes Aluminium (Angussreste) an, welches an den Druckgussanlagen dezentral in Transportbehältern bereitgestellt wird. Die Stapler (e) beliefern die Schmelzöfen mit Angussresten und Ausschuss. Ist eine direkte Zuführung nicht möglich, wird das Material zentral an den Schmelzöfen zwischengelagert. Zur vollständigen Materialversorgung des Schmelzofens beliefern die Stapler (e) die Schmelzöfen zusätzlich mit festem Aluminium (Masseln)(f). Transporter (c) fahren neben dem Material aus den Schmelzöfen auch zuge-

kaufte Flüssialuminium aus Kippstationen (b). Für Masselmaterial und flüssiges Aluminium existiert jeweils ein zentrales Lager.

2.2 Beschreibung der relevanten Komponenten im Materialfluss

Wesentliche Komponenten des Materialflusses sind die Schmelzöfen, die Transporter, die Stapler und die Druckgussanlagen. Der Schmelzofen, besteht aus einer Beschickungsanlage, Schmelzschacht, mehreren Brennern und dem Schmelzbad. Dabei fungieren die Beschickungsanlage, der Schmelzschacht und der Badspeicher als Materialspeicher. Die Materialübergänge sind in die ereignisdiskreten Prozesse „Beschickung“, „Entnahme“ und dem kontinuierlichen Prozess „Schmelzen“ zu unterteilen. Die Geschwindigkeit des Materialübergangs während des schmelzens ist neben anderen Faktoren abhängig von der Einstellung der Schmelzbrenner.

Die Druckgussanlage bildet zusammen mit dem Dosierofen eine Einheit. Dieser hat die Funktion, das angelieferte flüssige Aluminium für den Produktionsprozess zu speichern und warmzuhalten. Der ereignisdiskrete Produktionsprozess besteht aus zwei parallel ablaufenden Prozessketten (Gießen und Bearbeiten), wobei die Taktzeitbestimmende Prozesskette das Gießen ist.

Neben den technischen Unterschieden des Transporters und des Staplers lassen sich die beiden Fahrzeuge auch in der Art der Aufträge unterscheiden. Die Auftragsstypen des Staplers sind:

- Anlieferung von Masselmaterial
- Anlieferung von Angussrestes
- Anlieferung von Ausschussmaterial
- Rücktransport des leeren Transportmediums
- Beschickung des Schmelzofens

Demgegenüber hat der Transporter, mit der Versorgung der Druckgussanlagen mit flüssigem Aluminium nur einen Auftragstyp. Bei diesem ändern sich lediglich die die Anzahl und der Ort der anzufahrenden Komponenten.

2.3 Produktionssteuerung

Die Produktionssteuerung orientiert sich an zwei Prinzipien – dem Pull- und Push-Prinzip.

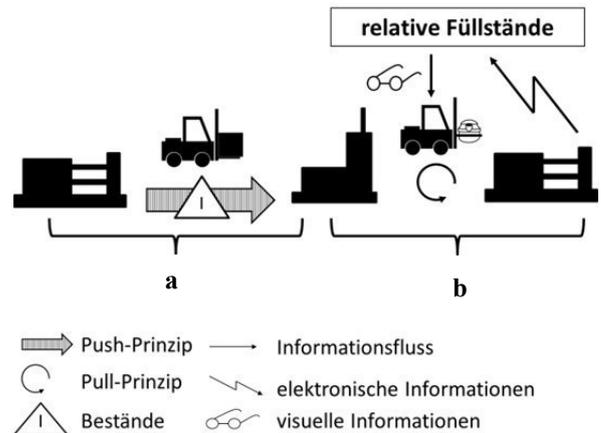


Abbildung 2. Vereinfachte Darstellung des Wertstroms eines NE-Schmelz- und Druckgussbetriebes

Charakteristisch für das Push-Prinzip (a) ist, dass Materialien bzw. Teile zum nächsten Prozess gedrückt werden und dort in Pufferzonen abgestellt werden. Oftmals ist dafür nicht der vorgelagerte Prozess sondern die betriebsinterne Logistik verantwortlich. Die Folge des Push-Prinzips ist, dass die Schmelzöfen bei maximaler Leistung gefahren werden, damit soviel Aluminium wie möglich geschmolzen wird. Eine Anpassung an den aktuellen bzw. zukünftigen Bedarf an flüssigem Aluminium erfolgt nicht. Das bedarfsorientierte Pull-Prinzip (b) ist durch einen Informationsfluss gekennzeichnet, der entgegengerichtet zu dem Produktionsablauf ist. In dem analysierten Schmelz- und Druckgussbetrieb bilden visuelle Informationen, welche sich aus den von Sensoren erfassten relativen Füllständen der Druckgussanlagen ableiten, die Entscheidungsgrundlage für die Aluminium-Distribution.

3 Aufbau der Simulation eines NE-Schmelz- und Druckgussbetrieb

3.1 Grundlegender Aufbau der Simulation

Bei der Analyse des Referenzbetriebes wurden kontinuierliche und ereignisdiskrete Prozesse identifiziert, die mit Matlab, Simulink und Statflow in ein hybrides Simulationsmodell umgesetzt wurden. Dabei übernehmen die drei genannten Werkzeuge die Funktionen:

- Matlab: Simulationssteuerung- und auswertung, Objektinstanzierung- und Verwaltung,
- Simulink: Simulation kontinuierlicher Prozesse

- Stateflow: Simulation ereignisdiskreter Prozesse

Vor Beginn der Simulation werden Datensätze zur Betriebs- und Simulationskonfiguration benötigt. Es handelt sich dabei um anlagenspezifische Parameter und um Daten der Produktionsplanung- und steuerung. Diese beinhalten unter anderem geplante Anlagenstillstände und aus zuvor bestimmten statistischen Kennwerten stochastisch berechnete ungeplante Stillstände.

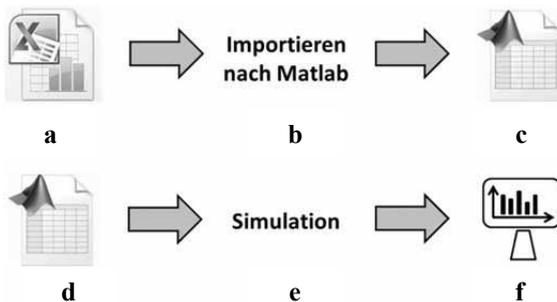


Abbildung 3. Import- und Simulationsprozess

Die relevanten Daten werden in Tabellen (a) bereitgehalten. Durch ein selbst entwickeltes von der Simulation unabhängiges Konvertierungsprogramm (b) werden diese in ein matlabspezifisches Format (c/d) umgewandelt und gespeichert. Bei Simulationsstart (e) erfolgt das Laden der Daten (c/d) in die Simulation. Nach Simulationende erfolgt die Visualisierung und Auswertung (f) der Daten. Der Vorteil der beschriebenen Vorgehensweise ist, dass die Betriebs- und Steuerungskonfiguration bei einmaligem Importieren mehrmals verwendet werden kann und vor der Simulation ausgeführt wird. Dies reduziert die Simulationszeit deutlich.

Der in 2.1 beschriebene Materialfluss eines NE-Schmelz- und Druckgussbetriebs ist in vereinfachter Form in der Simulation abgebildet. Dies betrifft unter anderem den Druckgussprozess, die Lagerung und den Transport der als Nebenprodukte anfallenden Materialien (Angussreste/Ausschuss), sowie die Fahrzeiten der Transporter/Stapler. Die Unterschiede im Detail sind:

- Zusammenfassung des in 2.2 beschriebenen Druckgussverfahrens in einem Prozess „Giesen“, da Fokussierung auf den Materialfluss und nicht auf das Druckgussverfahren
- Zentrale Erfassung und Bereitstellung der Behältern in materialspezifischen Lagern,

anstatt einer dezentrale Bereitstellung der Transportbehälter an den Druckgussanlagen

- Abbildung der Behälter in Form von Lagerbeständen, anstatt der aufwendigen Verwaltung einer großen Anzahl an Objekten
- Verwendung des statistischen Mittelwertes für die Fahrzeiten der Fahrzeuge, anstelle einer aufwendigen Erfassung der Koordinaten aller Komponenten und einer standortabhängigen Berechnung der Fahrzeit
- Keine Erfassung des Materials als eigenständige Komponente, sondern Abbildung des Materials in Form von Füllständen der Komponentenspeicher

Durch das gewählte Abstraktionsniveau konnte die Komplexität der Simulation gering gehalten werden. Es muss keine große Anzahl an speziellen Objekten (Transportbehälter, Materialpakete) verwaltet werden, um den Materialfluss abzubilden. Zudem wäre eine Abbildung des Materials und der Transportbehälter als eigenständige Objekte nur für eine materialpaket-spezifische energetische Betrachtung (z.B. Abkühlverhalten des Materials im Materialfluss) relevant. Innerhalb dieses Projektes stehen allerdings die energetische Optimierung und effiziente Betriebsweise der Schmelzöfen und der Einfluss auf das Gesamtsystem im Mittelpunkt der Untersuchungen, weshalb das gewählte Abstraktionsniveau für die Untersuchung genügt.

Aufgrund der Anforderungen einer einfachen Adaption auf andere NE-Schmelz- und Druckgussbetriebe sowie die unterschiedliche Komplexität der einzelnen Komponenten wird ein objektorientierter modularer Aufbau der Simulation verfolgt. Dies betrifft nicht nur die Komponenten, wie etwa den Schmelzöfen, sondern auch die Steuerung. Eine Voraussetzung, um eine modulübergreifende Kommunikation zu ermöglichen, ist das Definieren von Schnittstellen.

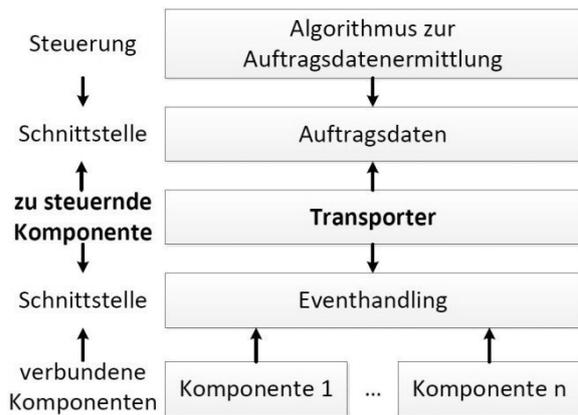


Abbildung 4. Schematischer Aufbau der Simulation

In **Abbildung 4** wird der schematische Simulationsaufbau am Beispiel des Transporters beschrieben. Dieser Aufbau gilt für alle zu steuernden Komponenten. Von der übergeordneten Steuerung wird mittels eines Algorithmus, in dem das spezifische Prozesswissen abgebildet ist, ein Auftrag ermittelt. Dessen Daten werden von der zu steuernden Komponente gelesen. Die Abarbeitung des Auftrags erfolgt autonom in der Komponente. Dazu ist Kommunikation zwischen der gesteuerten Komponente und der mit ihr verbundenen Komponenten notwendig. Die Schnittstelle dafür ist das sog. Event-Handling.

Im folgenden Abschnitt werden die Komponenten, die Schnittstellen und die Steuerung detaillierter beschrieben.

3.2 Definition der Komponentenklassen

Definiert sind die Komponenten in jeweils einer Klasse. Der Schmelzofen ist aufgrund seines komplexeren Aufbaus, bestehend aus dem kontinuierlichen Schmelzprozess und den ereignisdiskreten Prozessen Beschickung und Entnahme, durch mehrere Klassen (Beschickung Schmelzschacht, Brenner, Badspeicher) dargestellt. Die Eigenschaften der Klassen sind in Anlagenparameter und dynamische Prozessparameter zu unterscheiden, deren Wertzuweisung durch das in 3.1 beschriebene Verfahren erfolgt. Die Verwaltung der globalen Objekte erfolgt in Matlab. Modelliert und simuliert werden die ereignisdiskreten Prozesse in Stateflow.

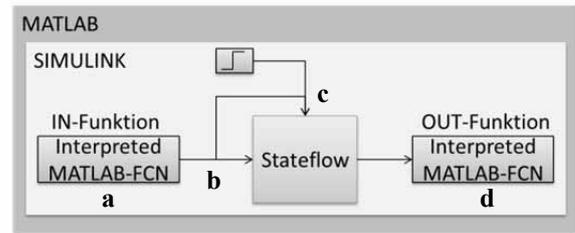


Abbildung 5. Schnittstelle zwischen Matlab, Simulink und Stateflow

Die notwendige Schnittstelle zwischen Matlab, Simulink und Stateflow ist, wie abgebildet, durch die „Interpreted MATLAB Function“ realisiert. Die sog. „IN-Funktion“ (a) übergibt die notwendigen Anlagenparameter (b) und Events (c). Die Wertänderungen des Systems werden zyklisch in der OUT-Funktion (d) durch Abfrage des Status ermittelt. Hier erfolgt in Abhängigkeit des Status die Anpassung der Prozessparameter.

3.3 Entwicklung der Zustandsautomaten und Kopplung mit einem Event-Handling

Das Verhalten der autonom arbeitenden Komponenten wird durch die in Stateflow definierten endlichen Zustandsautomaten bestimmt. Die Komponenten reagieren auf das Verhalten (Zustandsänderungen) weiterer Komponenten.

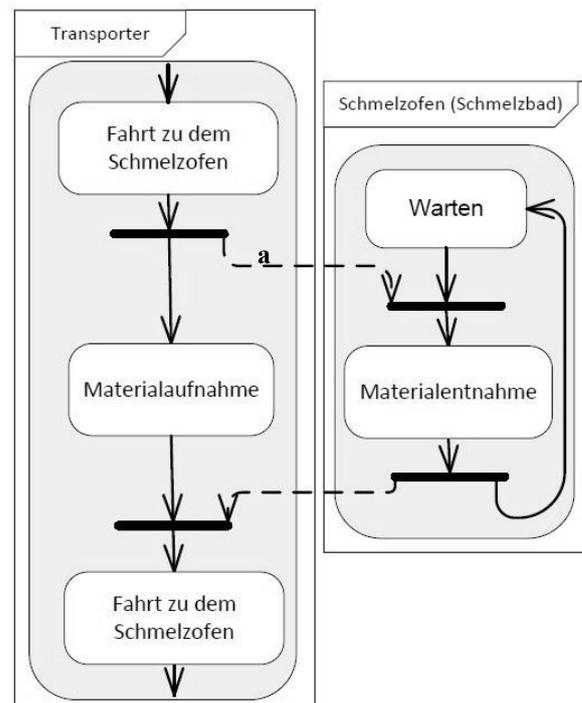


Abbildung 5. Kommunikation der Zustandsautomaten und Kommunikation zweier Komponenten

Als Beispiel dient die Abwicklung eines Auftrages zur Belieferung der Druckgussanlagen mit flüssigem Aluminium. Dabei sind neben dem Transporter auch der Schmelzofen und mehrere Druckgussanlagen beteiligt. Der modulare Aufbau der Simulation verhindert eine Realisierung mit einer direkten Übergabe der Events (a). Daher fungiert als zentrales Kommunikationsmedium zwischen der zu steuernden Komponente und den beteiligten Komponenten, der als Objekt definierte übergeordnete Event-Handler, wobei zwischen den Komponenten sowohl eine unidirektionale als auch bidirektionale Kommunikation ermöglicht wird.

3.4 Beschreibung der modular aufgebauten Steuerung

Die Simulationssteuerung übernimmt neben dem An- und Ausschalten (Simulation von Ausfallzeiten) der Komponenten, sowie der Zuweisung eines Produktionsauftrages an die Druckgussmaschinen auch die Funktion der Auftragsdatenermittlung für die Transporter und der Stapler. Dabei wird der Auftrag "just in time" zusammengestellt. Die Vorteile dieses Vorgehens sind, dass keine Auftragsliste aufwendig verwaltet werden muss und immer der „optimale“ Auftrag ermittelt wird. Der prinzipielle Aufbau der Steuerung ist für beide Fahrzeugtypen gleich, obwohl wie in 2.2 aufgezeigt, sie sich in der Anzahl und Art der Aufträge unterscheidet.

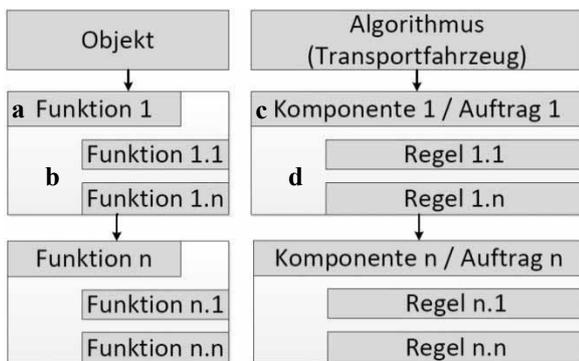


Abbildung 6. Modularer Aufbau der Steuerung

Für jeden Fahrzeugtyp ist der Steuerungsalgorithmus in einem Objekt hinterlegt. Die Steuerungsstruktur und die hinterlegten Regeln bilden dabei die Produktionssteuerung und das Prozesswissen ab. In den übergeordneten Funktionen (a) wird das Produktionssystem durch die Priorisierung der Komponenten/Aufträge (c) abgebildet. Die untergeordneten Funktionen (b) behinhalten das Prozesswissen. Dabei wird in jeder Regel geprüft, ob eine Komponente verfügbar ist (Auftragsermittlung für die Transporter) oder ob ein Auftrag ausgelöst werden kann (Auf-

tragsermittlung für die Stapler). Die Software ist so konzipiert, dass verschiedene Steuerungsalgorithmen getestet werden können, ohne Veränderungen an der Basissimulation (Zustandsautomaten, Komponenten, Event-Handling) vornehmen zu müssen. Im Folgenden wird das resultierende Anwendungspotential beschrieben.

4 Anwendungspotential

Mit dem vorgestellten Modell des Schmelz- und Druckgussbetriebes können verschiedene Zustände des realen Betriebes abgebildet und das Systemverhalten analysiert werden. Durch Parameteränderungen oder die Veränderung von Regeln im Betriebsablauf des Modells können zudem die Möglichkeiten der Betriebsoptimierung in Bezug auf Prozesssicherheit und Energieverbrauch untersucht werden. Ein Anwendungsfeld ist die Untersuchung der Auslegung der Schmelzöfen bzw. des Maschinenparks. Dabei wird der Einfluss der Schmelzleistung und der Speicherkapazität der Öfen auf die Prozesssicherheit untersucht. Ein zweites Untersuchungsfeld ergibt sich in der Betrachtung verschiedener Angebots/Nachfrageszenarien von flüssigem Aluminium und der Ermittlung des optimalen Zeitpunktes der Integration des zugekauften flüssigen Aluminiums in den Materialfluss. Durch die Ausfälle von Druckgussanlagen werden Überlast- bzw. Unterlastsituationen erzeugt. Die Auswirkungen auf die Speicherstände und Stillstandszeiten der Druckgussanlagen können untersucht werden. Das größte Potential wird aber in einer energieoptimierten Schmelzofenführung gesehen. Dabei können verschiedene Beschickungs- und Entnahmeregelungen oder die Auswirkung von vorgewärmtem Beschickungsmaterial auf den spezifischen Energieverbrauch getestet werden. Eine weitere Möglichkeit den Energieverbrauch der Schmelzöfen zu senken ist, diese bei einem Überangebot an flüssigem Aluminium in einen Standby-Betrieb zu versetzen.

5 Zusammenfassung

Der energieeffiziente Betrieb von NE-Schmelz- und Druckgussanlagen kann einen großen Beitrag zur Energiewende leisten. Ein großes Potential wird in dem Schmelzprozess selbst und der Verteilung des flüssigen Aluminiums auf die Druckgussanlagen gesehen. Zur simulationsgestützten Untersuchung dieses Prozesses wurde eine hybride Simulation entwickelt. Bei der Konzipierung der Komponenten, des Event-Handlings und der Steuerung wurde objektori-

entierten vorgegangen und auf den modularen Aufbau geachtet. Die beschriebene Software lässt sich dadurch einfach auf verschiedene NE-Schmelz- und Druckgussbetriebe adaptieren und die Steuerung lässt sich ohne Veränderungen am Sourcecode der Basis-simulation frei konfigurieren. Damit sollen in Zukunft unter Berücksichtigung verschiedener Anlagen-konfigurationen, Angebots- und Nachfragescenarien Regeln zu einer energieeffizienten Betriebsführung des Schmelzofens getestet werden. Das neu gewonnene Prozesswissen soll in die Entwicklung des angestrebten Prozessmanagementsystems „SmartMelting“ einfließen.

6 References

- [1] Arbeitsgemeinschaft Energiebilanzen. *Auswertungstabellen zur Energiebilanz 1990 bis 2013*. Berlin, 2014.
- [2] Hirschvogel, Markus und Kleindienst, Mario. *Auswirkungen der globalen Energiepreisentwicklungen auf ausgewählte Produktionsprozesse eines Automobilzulieferers*. Graz, 2014.
- [3] WVM, Wirtschaftsvereinigung Metalle. *Metallstatistik 2013*. Berlin, 2014
- [4] W. Stephan, et. al. *Effiziente Energieverwendung in der Industrie - Teilprojekt "Metallschmelzbetriebe"- Effiziente Energienutzung in Nicht-Eisen-Metall-Schmelzbetrieben*, Bayer. Landesamt für Umweltschutz, Augsburg, 2005.
- [5] A. Dentel, et. al. *Rationeller Energieeinsatz in der aluminiumverarbeitenden Industrie. Abwärmennutzung und verbesserte Isolierung von Transportbehältern für flüssiges Aluminium*. In: Ergebnisse des Forschungsverbundes Energieeffiziente Technologien und Anwendungen (FO-RETA), Herausgeber: W. Mayer, Attenkofer, Straubing, S. C1-C24, 2013.
- [6] Herrmann et. al. *Energie- und ressourceneffiziente Produktion von Aluminiumdruckguss*. Springer, Berlin, 2013.
- [7] B. Mert und M. Hashemipour. *Virtual factory approach for implementation of holonic control in industrial applications: A case study in die-casting industry*. Robotics and Computer-Integrated Manufacturing: Elsevier Ltd., 2009, S. 570-581.
- [8] R. Neugebauer, et. al. *Energy-Sensitive Production Control in Mixed Model Manufacturing Processes in Leveraging Technology for a Sustainable World*. Springer, Berlin, S 399-404, 2012
- [9] Gram, Markus. *Wertstromanalyse als Potentialanalyse in der Prozessindustrie*, WINGbusiness (02/2011) 39-41.

MATLAB/Simulink Based Rapid Control Prototyping for Multivendor Robot Applications

Christina Deatcu¹, Birger Freymann¹, Artur Schmidt¹, Thorsten Pawletta¹

¹Hochschule Wismar – University of Applied Sciences

Research Group CEA, Wismar, Germany

christina.deatcu@hs-wismar.de

Industrial robots are used in various fields of application and many robot manufacturers are active in the market. In most cases, their software solutions are proprietary and, consequently, they cannot be used for third party robots. Moreover, the integration of external hard- or software is highly restricted. Long term standardization efforts for robot programming languages, such as the Industrial Robot Language (IRL) and its successor, the Programming Language for Robots (PLR), have been mostly ignored by robot manufacturers. This fact leads to a restriction on the combined usage of robots. Multi-robot applications where robots have to interact are usually limited to software solutions and robots of one manufacturer. On the other hand, control design in engineering is often carried out by the usage of Scientific and Technical Computing Environments (SCEs) like MATLAB. The Robotic Control & Visualization Toolbox (RCV Tbx) for MATLAB/Simulink tries to close the gap between robot manufacturer-specific software solutions and SCEs. The current version of the RCV Tbx supports a uniform and integrated control development for KUKA and KAWASAKI robots in the MATLAB/Simulink environment. An extension to other robot types is straight forward. Thus, the implementation of heterogeneous multi-robot applications is considerably simplified.

1 Introduction

This paper is an extended version of [1] and aims to introduce the RCV Tbx for MATLAB/Simulink as an easy to use Rapid Control Prototyping (RCP) Tool for multivendor robot controls. The RCV Tbx has been under development by the research group *Computational Engineering and Automation* (CEA) at Wismar University since 2009 [2].

As research into robotics is proceeding rapidly and new fields of application for robots are being made up continually, the requirements concerning robot control development are increasing, too. Fast and easy control programming, integration of external hardware or software components and multi-robot operation are of particular importance. In this context it is often desirable to use a SCE, such as MATLAB, for RCP. RCP, according to Abel and Bolling [3], is understood as an integrated, continuous control development from early design to operating phase in a homogenous environment.

In addition to multi-robot capability, multivendor applications are one further key aspect. Today, various robot manufacturers are established on the market. They offer proprietary software environments with special robot programming languages such as KRL (VEN KUKA Robotics), AS (VEN Kawasaki

Robotics) or RAPID (VEN ABB Robotics). From a software engineering point of view, all these robot languages are pretty similar. Nevertheless, long-term standardization efforts like the IRL and its successor, the PLR are still unsuccessful. In addition, almost all robot manufacturers offer a Computer Aided Robotic (CAR) system, which is also referred to as 3D robot simulation software. CAR systems typically provide physics-based robot models for one manufacturer as well as interfaces to 3D CAD systems. Thus, a simulation and 3D visualization of complete robot cells is supported. Robot controls can be developed within these virtual environments using the proprietary robot languages. Such CAR systems simplify the development and commissioning of robot applications. However, the proprietary software limits applications to products from its manufacturer. There are some third-party CAR systems available, such as 3DRealize-R by Visual Components Corporation [4]. They offer a comprehensive solution for simulation and 3D visualization of heterogeneous robot types. However, the control programming is still based on the different proprietary robot languages. Hence, the development of interacting multi-robot controls is complicated for robots from different manufacturers.

For MATLAB/Simulink, besides the RCV Tbx [2], there exists a robot control toolbox for KUKA robots

(KUKA Control Toolbox, KCT) developed at the University of Sienna [5, 6]. The KCT connects a remote MATLAB computer via TCP/IP to the robot controller of a KUKA robot. Usage of KCT is limited to one single robot from one manufacturer, namely KUKA, so that multi-robot and multivendor applications cannot be addressed.

Inspired by the idea of RCP in control theory, the research group CEA began in 2004 to develop a MATLAB KRL toolbox. This toolbox supports control programming of KUKA robots within MATLAB including the usage of all available MATLAB features. Moreover, it provides a first MATLAB based CAR system [7]. As well as the KCT, the MATLAB KRL Tbx was limited to KUKA robots, but users already benefited from the powerful methods as well as the various interfaces provided by MATLAB and its toolboxes.

Almost all of the proprietary robot languages are imperative languages containing similar programming elements. Thus, the approach of the MATLAB KRL toolbox was generalized. A uniform robot control language for robot types from different manufacturers has been developed with the RCV Tbx for MATLAB. Moreover, the simulation and 3D visualization tools have been enhanced.

This paper is organized as follows: section 2 of this paper introduces the concept of RCP and relates it to robot controls. In section 3, the RCV Tbx for MATLAB is described. Design, implementation as well as user interface aspects are analyzed. Finally, section 4 gives a summary and identifies potentials for future work.

2 Rapid Control Prototyping

This section summarizes the RCP approach and how it can be used for robot control development. Systematic development of controls can be carried out following the V-model derived from Orth, Abel and Bollig [8, 3].

The V-model defines two main phases, the design phase and the commissioning phase. The design phase starts with the problem specification and continues with a draft and simulative testing of the control algorithms. It is completed with coding of an executable control program for the target hardware. This piece of software is then used to bring the con-

trol into operation. The commissioning phase starts with component tests and ends with a test of the control across the entire process. Each step of the V-model may have to be performed several times and through several iterations or it is possible that leap-back in the development process will occur.

2.1 Fundamentals

RCP generally requires either a well-adjusted tool chain to follow the V from specification phase down to coding phase and up to operational phase or support from an integrated development environment. This integrated development environment can be an SCE. Furthermore, Software-in-the-Loop simulation (SiL) and Hardware-in-the-Loop Simulation (HiL) techniques following Abel and Bollig [3] and explicit automatic code generation are key features of RCP-capable software systems.

Maletzki [9] adopted the general V-model for controls for robot control development as illustrated in Figure 1.

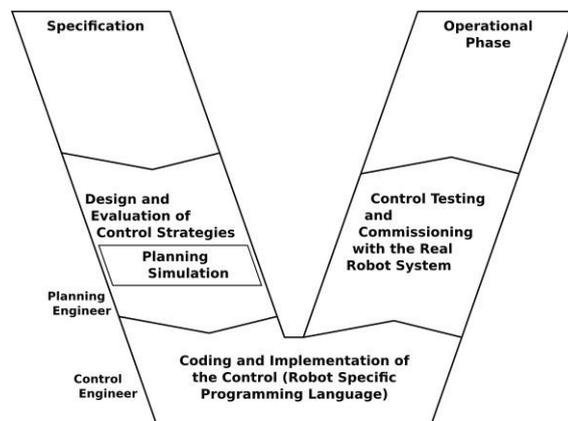


Figure 1. Adopted V-Model for Robot Control Development According to Maletzki [9].

A notably critical point is the transfer of results from planning simulation to the coding and implementation phases. As for industrial robots, the executable control program for the target hardware typically has to be written in a robot specific programming language; a continuous tool chain is not guaranteed. The control strategies that result from the planning simulation are handed over from the planning engineer to the control engineer. This kind of manual handing over is obviously fault-prone.

2.2 RCP and Proprietary Robot Controls

Conventionally, robot controls are coded in vendor-specific programming languages and tested using dedicated 3D-simulation software. Such CAR systems offer software libraries with robot models from the particular vendor and usually include interfaces to 3D-CAD software. In CAR systems, concrete control strategies can be implemented within the specific robot programming language so that extra coding after simulative testing of the control is not required. 3D visualizations, of e.g. robot movements and potential collisions, play a decisive role if CAR systems are deployed. Continuity as required for RCP is partly given but manual knowledge transfer from planning to control engineer is still necessary.

The conventional robot control programming approach supported by CAR systems and associated languages and software respectively are depicted in Figure 2.

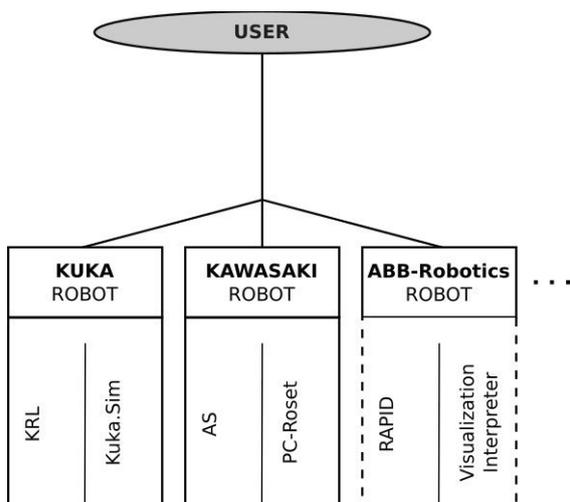


Figure 2. Conventional Robot Control Development Using Vendor Specific Software

After specification of the control requirements the planning engineer designs control strategies using planning simulation tools. Subsequently, he delivers a control strategy to a control engineer mostly in a textual manner. The control engineer implements the robot control using vendor specific tools as listed in Figure 2. At this point there is a break in the tool chain according to the definition of RCP by Abel et al. [3]. However, a continuous re-use of software components during this transition is hard to realize

for robot control development, because different concepts are used and are necessary in these two phases. Starting from this point, a control engineer can implement and deploy a robot control for vendor-specific robots compliant with the definition of RCP. For example, for KUKA robots, coding of controls is done using KUKA Robot Language (KRL) and simulation of the control with robot models is carried out using the Kuka.Sim software. Control testing and commissioning with the real robot system can be achieved by automatic code generation or a communication link as illustrated in Figure 3.

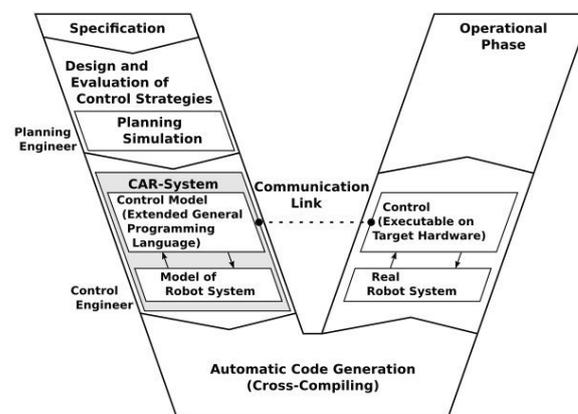


Figure 3. Detailed V-Model for Robot Control Development According to Maletzki [9].

In the case of industrial robots, an explicit code generation for target hardware is not necessary. The control program from the design phase is implemented in a robot-oriented language and running on a robot control computer that is used in operation phase. In the context of RCP, this approach is called *implicit code generation*. Hence, commissioning can be carried out via a simple communication link between a CAR system on a PC and a robot controller. This practice is called *Software in the Loop* (SiL) approach in [3]. The concept of explicit automatic code generation for target hardware is important for mobile robotic applications.

However, the robot control development software depicted in Figure 2 is vendor-specific and incompatible between types. This fact complicates the previously discussed control development or makes it nearly impossible to develop multivendor robot applications.

2.3 RCP and the RCV Toolbox for MATLAB/Simulink

Controls for interacting robots and multivendor robot controls are examples of advanced tasks in robot control development.

Figure 4 illustrates how the RCV Tbx eases such tasks by providing not just a generalized interface to robot control and robot visualization commands, but also to robots from different vendors and/or types.

The control engineer implements a robot control as a MATLAB coded sequence of control commands. These commands are independent from a specific robot vendor as well as from a specific robot language, but they are very similar to established robot programming languages such as KRL and AS. Furthermore, control commands can be combined with any MATLAB commands and also be generated from Stateflow or Simulink models which eases the implementation of complex controls.

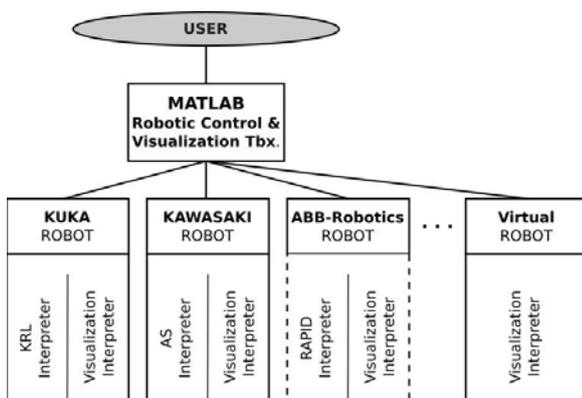


Figure 4. Robot Control Programming with the RCV Tbx for MATLAB/Simulink

Moreover, the RCV Tbx for MATLAB provides robot-specific interpreters coded by using the vendor-specific robot languages, which have to be installed on the robot controllers. The MATLAB based control PC and the robot controllers are connected via a bi-directional communication link. The interpreters are responsible for the identification and execution of control commands that are transmitted by the MATLAB based control PC and they deliver acknowledgements or sensor signals back. Figure 5 illustrates an RCV Tbx-based multi-robot configuration.

One or more computers with MATLAB/Simulink and the RCV Tbx installed act as the continuous software environment from the early design and evaluation phase via control testing with the real system and finally the operational phase. No recoding or reimplementa-tion is necessary, rather the control program can be extended successively until it meets the demands of the intended control task. SiL and HiL approaches as defined in [3], as requirements for RCP capability of control development, are met because the simulated control can be stepwise extended to become the real control for the operational phase.

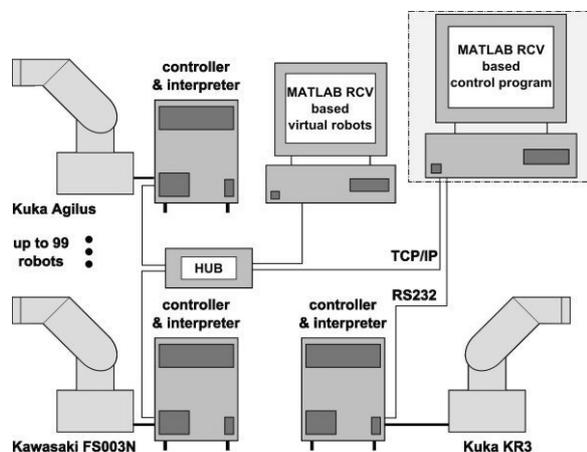


Figure 5. A Multi-Robot Configuration with Heterogeneous Robot Types Using the RCV Tbx

Short control sequences or single control tasks can already be tested with the real process during design and evaluation (SiL). HiL tests, in which a real control device is tested with a simulated process, are not usually relevant for industrial robot controls as mentioned in section 2.3. However, they could be of interest for other devices in a robot application. Furthermore, a control application can also be tested in a purely virtual environment using the RCV Tbx similar to the usage of a CAR system. For this purpose the RCV Tbx provides 3D-models of real robots and other virtual components.

3 Design and Usage of RCV Tbx for MATLAB/Simulink

This section describes and analyzes the software design of the RCV Tbx. It details the two toolbox

downloads as they are available at [2], namely the robot control download (*Robotic Control Tbx*) and the visualization download (*Robotic Visualization Tbx*). It is shown how a robot control program can be developed, tested by simulation, put into service (soft commissioning) and finally used as a real control. Continuity of the tool chain, as is needed to meet the requirements of RCP, is given. Hence, the same sequence of control commands coded in the same language can be sent either to a visualized or a real robot.

Figure 6 depicts the main functional parts of the RCV Tbx. *Control*, *Interpreter* and *Visualization* are the three main functionalities which are distributed across the two software packages.

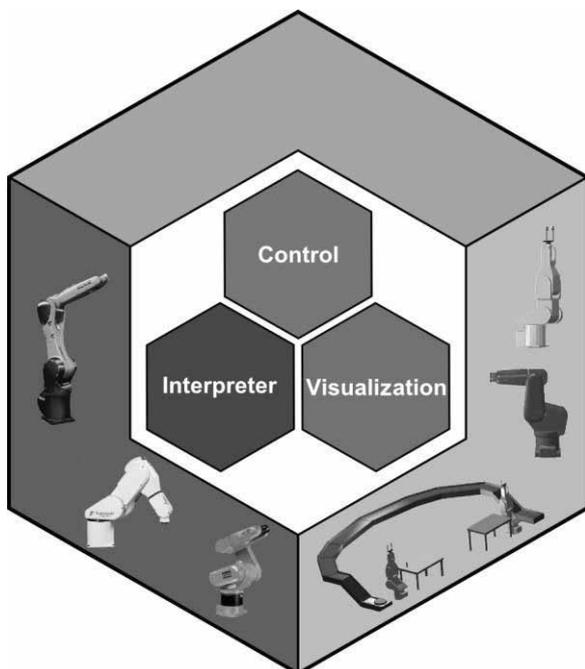


Figure 6. Main Parts of the RCV Toolbox for MATLAB/Simulink

A collection of control commands for real, as well as visualized, robots and interpreters for real robots are downloaded with the first part of the RCV Tbx, the *Robotic Control Tbx*. Interpreters for visualized robots, also called virtual robots, and the visualization software are downloaded with the second part of the RCV Tbx, the *Robotic Visualization Tbx*. Currently, both parts support interpreters for Kawasaki FS003N,

Kuka Agilus KR6 (TCP/IP connection) and for KUKA KR3 (RS232 connection).

3.1 Part I: Robotic Control Tbx

This subsection details robot control programming and gives some examples of robot control commands.

The control of real robots requires an interpreter program to be installed on the robot controller. The interpreter for each kind of real robot is written in the appropriate robot-specific language and copied to the robot controller. It establishes the communication link between control PC and robot, which can, depending on the robot type, be a TCP/IP link or a RS232 link. The interpreter program takes the commands from MATLAB and translates them into commands for the robot-specific language which are executable by the robot controller. Security considerations, such as workspace supervision and movement execution, are, therefore, still covered by the robot controller.

After the toolbox is installed on the control PC and the appropriate interpreter is copied to the robot controller, the interpreter program on the controller needs to be started. For RS232 connections, after startup of interpreter, the controller is ready to receive MATLAB control commands immediately. For TCP/IP connections, the interpreter on the controller acts as a server and waits for a suitable client to connect. This client is a robot object created on the control PC. Table 1 lists all available commands for robot control alphabetically.

rbrake ()	brake the motion of one or all robots directly
rcallback ()	define functions, that are executed automatically
rdisp ()	formatted display of position structures
rerror ()	set up a function, dealing with error codes received from robot controllers
rget ()	get interpreter-, toolbox- and motion-properties and current positions
ris ()	check the status of commands and processes
rkill ()	brake and stop the motion of one or all robots directly; then set up the original state of robot(s)
rmove ()	move a robot
robot ()	create or destroy a robot object that can be controlled
rpoint ()	define robot-positions with coordinates and motion-properties

rprocess()	define complex operations
rreset()	reactivate robot(s) after automatic switching-off of the interpreter(s)
rrun()	reactivate robot(s) after using rbrake() or rstop()
rset()	set up interpreter-, toolbox- and motion-properties
rstatus()	switch ON or OFF status report
rstop()	brake and stop the motion of one or all robots directly
rteach()	teach and save positions of a robot
rwait()	block the MATLAB-prompt until the status of commands or processes fulfills a condition

Table 1. List of Available Robot Control Commands

Of essential importance is the `robot()` command which creates and destroys the robot object which acts as an interface for control commands. Code 1 shows the syntax of the `robot()` command. Currently, robots can either be of type 'Kawasaki' or of type 'Kuka'.

```

1 h1 = robot( 'open', 'Kawasaki', 'tcpip',
  IP-ADRESS, PORT );
2 h2 = robot( 'open', 'Kuka', 'serial',
  COM-PORT);
3 robot( 'close', h1, h2);

```

Code 1. Create and Destroy Robot Objects

The command `robot()` creates a robot object and returns a MATLAB handle to the object. This handle can then be passed to other control commands to address this specific robot. Some control commands such as `rbrake()`, `rkill()`, `rreset()`, `rrun()` and `rstop()` affect all robots that the control PC is currently connected to, if no handle is passed to the command as a first parameter. Complex control operations can be coded by using the command `rprocess()`. With this command, tasks to be fulfilled by more than one robot can also be defined. Sequential as well as concurrent operations are possible, and control commands can be structured and grouped. Code 2 shows a short example for a concurrent operation with two robots.

```

1 load Points P
2 rprocess( {...
3     Kuka, P1,...
4     Kawasaki,{P2, 'speed', 50},...
5     },...
6     {
7     Kuka, 'home',...
8     Kawasaki, 'home',...
9     } );

```

Code 2. Two Robots in a Concurrent Operation Programmed with `rprocess()`

Kuka und *Kawasaki* are the handles for the robots. The robots move at the same time to the positions *P1* and *P2* which are defined by the loaded variable *P*. They start to move to their home positions only after they have both finished their moves.

The `rprocess()` command already allows some complexity, but furthermore, all standard and advanced programming features of MATLAB/Simulink can be used to create even more complex and also simulation based robot controls. An example of such integration with other tools is depicted in Figure 7 where RCV control commands are embedded in Stateflow.

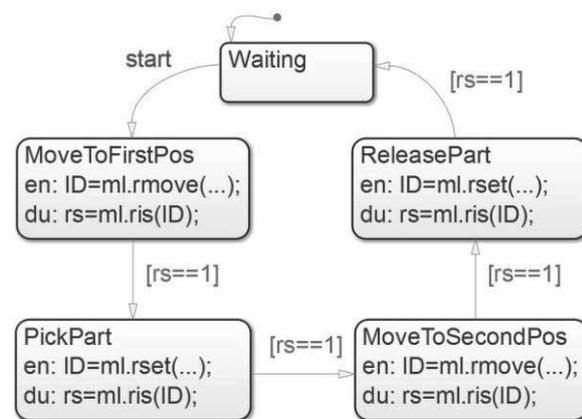


Figure 7. State Based Control Using Stateflow and RCV Tbx

The integration of arbitrary external hardware, which can act as sensors or actors, is also feasible. This would allow a big advantage compared to the restricted availability of additional hardware when developing robot controls with proprietary software and languages.

3.2 Part II: Robotic Visualization Tbx

The *Robotic Visualization Tbx* is the second software package of the RCV Tbx and can be used for testing and enhancement of controls developed with the Robotic Control Tbx. It offers distributed 3D-visualization of up to 99 robots in MATLAB and allows interactive control of visualized real robots and also pure virtual robots in a virtual environment. Thereby, safe development, testing and debugging of robot applications is possible. Furthermore, the toolbox includes an STL interface for importing user-defined graphical objects designed using external CAD software.

The control of a visualized robot requires an interpreter program, too. Interpreter functionality for virtual robots is included in the *Visualization Tbx*. Interpreters are part of virtual robot objects and establish a TCP/IP link between control PC and visualization PC. Control program and visualization can physically be located on the same PC, represented by two MATLAB instances. User toolbox's interface as well as virtual objects is designed as a MATLAB class. Figure 8 pictures visualizable object types that map entities of the real world.

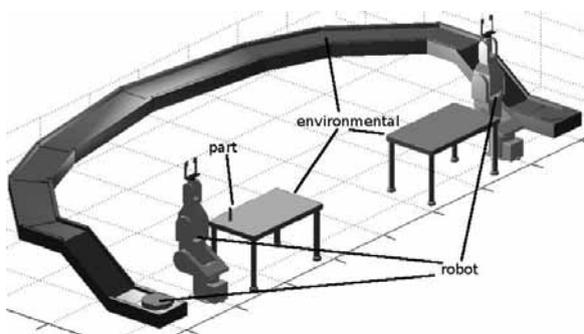


Figure 8. Types of Visualizable Object

Besides virtual robot objects environmental objects and part objects can also be visualized. Environmental objects are passive objects that cannot be moved by the robots while part objects are also passive objects which can be picked and moved by robots. In Figure 8 environmental objects are two tables and a conveyor; a test tube represents the part objects. Robot objects that represent the active visualization objects include kinematics which match the kinematics of the corresponding real robot. Furthermore, pure virtual robots or other active objects such as carts can

be visualized if the user defines appropriate kinematics. Figure 8 shows two visualized real robots and a pure virtual cart.

Table 2 lists commands of the Robotic Visualization Tbx. The first three commands are used to initialize, finish and monitor a visualization session. With the creation of a MATLAB 3D-figure at startup, a MATLAB timer object is also started. It ensures that the visualized objects are refreshed 20 times per second to achieve a smooth appearance of the animation.

ViSu.start	initialize the visualization, open an empty 3D window
ViSu.stop	stop the visualization, delete all virtual objects, close the figure
ViSu.info	display all virtual objects with their ID, type, position and additional information depending on object type
ViSu.create()	instantiate a robot object of type 'Kuka', 'Kawasaki' or user defined type
ViSu.repose_robot()	alter position of robot object identified by its id
ViSu.delete_robot()	delete a robot object identified by its id
ViSu.place_env()	instantiate an environmental object
ViSu.repose_env()	alter position of an environmental object identified by its id
ViSu.delete_env()	delete an environmental object identified by its id
ViSu.place_part()	instantiate a moveable object
ViSu.repose_part()	alter position of a part object identified by its id
ViSu.delete_part()	delete a part object identified by its id

Table 2. User Interface for Visualization

The other commands offer identical functionalities for the three different types of visualizable objects: robot, environmental and part objects can be I) initialized, II) repositioned, and III) deleted during a simulation session.

Currently, the toolbox is being revised extensively to harmonize the user interface and improve software stability and robustness.

3.3 Integrated RCV Tbx Usage Example

Figure 5 introduced in section 2 shows an example of multi-robot configuration. Notice that in that configuration we have integrated real robots from different vendors, i.e. from Kuka (Agilus, KR3) and Kawasaki (FS003N) and some virtual robots, too.

In this section we focus on an academic scenario where we have a robot control being applied only to some virtual objects. For this, it is necessary to start two instances of MATLAB; of these, one acts as the control PC (client) and the other is the visualization server. These MATLAB instances can either be located on different computers, as shown in Figure 5, or on the same computer as in the following example. The example includes two robots, the environmental object table and a test tube which is classified as a 'part'.

Code 3 illustrates how MATLAB control and visualization instances can interact. MATLAB commands in line 1, 2, 4, 6, 7, 11 to 12, and 14 have to be executed on the visualization instance, while the indented lines 3, 5, 8 to 10, and 13 are control commands which have to be executed on the control instance.

```

1 ViSu.start
2 ViSu.create('Kawasaki', 40000, [0, 0, 0,
  0, 0, 0])
3     r1=robot('open', 'Kawasaki', 'tcpip',
  'localhost', 40000)
4 ViSu.create('Kuka', 40001,[500, 500, 0,
  0, 0, 0])
5     r2=robot('open', 'Kuka', 'tcpip',
  'localhost', 40001)
6 ViSu.place_env('table.stl', [-800 0 0 0 0
  0], 'blue')
7 ViSu.place_part('test_tube.stl', [-800 0
  400 45 0 0], 'white')
8     rset(r1, 'signal', [-9, 10])
9     rmove(r1, 'home2');
10    rmove(r2, 'home2');
11 ViSu.info
12 ViSu.delete_part(1)
13    robot(r1, r2, 'close')
14 ViSu.stop

```

Code 3. Code Example of Interaction of the two Parts of RCV Tbx for MATLAB/Simulink

After a virtual robot has been created with the command `ViSu.create()`, the MATLAB prompt for the visualization instance is blocked until the TCP/IP

connection to the control instance is established. This is accomplished when the appropriate `robot('open', ...)` command is executed on the control instance which initiates a robot control object. The table, as well as the test tube, just exist virtually and are passive objects that are not controllable and therefore have no counterpart on control instance. After line 7 is executed, the visualization looks as depicted in Figure 9.

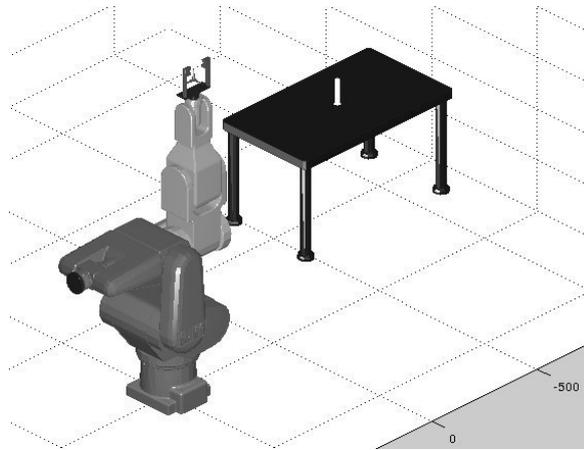


Figure 9. Visualization Example after Creation of Objects

Then some simple control commands, beginning with line 8, are executed. The command in line 8 closes the gripper of the Kawasaki robot. After that, both robots are moved to one of their predefined home positions, 'home2'. Back on the visualization instance, information on the current visualized objects is requested. With this information the user knows, that the test tube is a 'part' object with ID 1 and can be deleted during the visualization session. This feature is useful, if one wants, for example, to alter the surroundings of active robot objects without restarting the visualization.

After execution of line 12 the scenario looks as depicted in Figure 10. The gripper is closed, both robots have moved to their home position and the test tube has disappeared. The last two commands close the TCP/IP connections and finally close and delete the visualization figure.

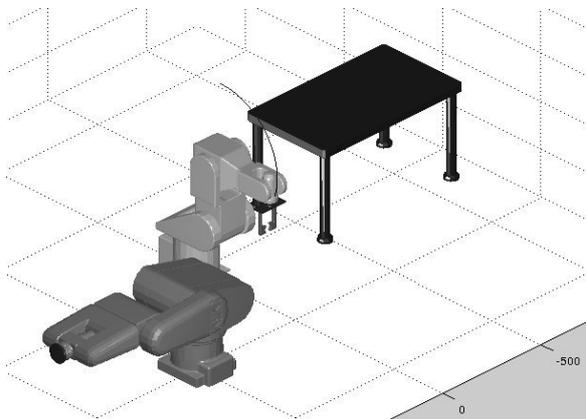


Figure 10. Visualization Example after Execution of Some Control Commands

Without larger modifications it is possible to apply the same sequence of control commands to real robots in a real environment, if we assume the real robots are of appropriate types and placed in the same positions. In this case, instead of the commands `Visu.create()`, that initialize the visualized real robots, TCP/IP and RS232 connections to the control PC need to be established by the robot controllers. For visualized robots all connections are of type TCP/IP, although for a real KUKA KR3, for example, an RS232 connection is necessary. Hence, the control command in line 5 has to be adapted. The control object `r2` needs to be opened with the parameter `'serial'` instead of with `'tcpip'`. Instead of a visualized table and a visualized test tube, a real table and a real test tube could be placed in the real robot cell. Motivation for such simple control tests by simulating the movements first in a virtual environment could be, for example, to avoid collisions between robot and table.

4 Summary and Further Work

The RCV Tbx offers excellent possibilities for developing multivendor robot controls in a homogeneous software environment. It fulfills the main requirements of a RCP capable tool. The key benefits of using the RCV Tbx are the possibility to develop monolithic control programs for interactive robots from different manufacturers and the easy, manufacturer-independent integration of external hardware. The MATLAB/Simulink environment is well established in the area of engineering and, today, is a

standard tool for engineers. Hence, the engineer can benefit from employment of all programming tools available in this environment. Especially for recent advanced control applications such as the Simulation Based Control (SBC) approach introduced by Maletzki [9] the integration of the RCV Tbx into MATLAB/Simulink is very advantageous. Simulation models are directly used as control programs in this approach and it is obvious that RCV Tbx control commands can easily be integrated into these kinds of simulative control. If one takes another step, it can be seen that models employed for control tasks can also be automatically generated from a knowledge base as proposed in the SBC and System Entity Structure (SES) approach by Schwatinski and Freymann [10, 11].

5 References

- [1] Pawletta, T., Freymann, B., Deatcu, C. and Schmidt, A. (2015). *Robotic Control and Visualization Toolbox for MATLAB*. Proc. of MATHMOD 2015 - 8th Vienna Int. Conf. on Mathematical Modelling, Breitenecker, F., Kugi, A. and Troch, I. (Eds.), February 18 - 20, ARGESIM Report No. 44 (ISBN: 978-3-901608-46-9), ARGESIM, Vienna/Austria UT, 2015, Pages 371-372 & Poster.
- [2] Research Group CEA. (2011). *Robotic Control & Visualization (RCV) Toolbox for MATLAB*, http://www.mb-hs-wismar.de/cea/sw_projects.html (date of reading 2015/03/28)
- [3] D. Abel and A. Bolling. (2006). *Rapid Control Prototyping – Methoden und Anwendungen*. Springer-Verlag Berlin.
- [4] Visual Components Corporation (2014). *3DRealize-R*, <http://www.visualcomponents.com/products/3drealizer-r>, (date of reading 2014/10/24).
- [5] F. Chinello, S. Scheggi, F. Morbidi, D. Praticchizzo. (2010). *KCT: a MATLAB toolbox for motion control of KUKA robot manipulators*. In Proc. IEEE Int. Conf. on Robotics and Automation, Anchorage, Alaska, Pages 4603-4608.

- [6] F. Chinello, S. Scheggi, F. Morbidi, D. Praticchizzo. (2011). *KUKA Control Toolbox. Motion control of robot manipulators with MATLAB*. Robotics Automation Magazine, IEEE, 18(4): Pages 69-79.
- [7] Maletzki, G., Pawletta, T., Pawletta, S. and Lampe, B.P. (2006). *A model-based robot programming approach in the MATLAB/Simulink environment*. In Advances in Manufacturing Technology – XX, 4th Int. Conf. on Manufacturing Research (ICMR06), Liverpool, UK, Sept. 05-07, 2006, Pages 377-382.
- [8] P. Orth, A. Bollig, D. Abel. (2004) *Rapid Control Prototyping diskreter Steuerungen in der Automatisierungstechnik*. In: SPS/IPC/Drives Congress, Nürnberg, 2004, Pages 143-152.
- [9] G. Maletzki. (2014) *Rapid Control Prototyping komplexer und flexibler Robotersteuerungen auf Basis des SBC-Ansatzes*. Phd Thesis, in German, Rostock University, Germany.
- [10] T. Schwatinski, T. Pawletta, S. Pawletta. (2012). *Flexible Task Oriented Robot Controls Using the System Entity Structure and Model Base Approach*. In: Simulation Notes Europe (SNE), Vol. 22(2), Special Issue 'Traffic Systems - Technical Systems', ARGESIM/ASIM Pub. TU Vienna, Austria, August 2012, Pages 107-114. (Print ISSN 2305-9974, Online ISSN 2306-0271, doi:10.11128/sne.22.tn.10133).
- [11] B. Freymann, T. Pawletta, T. Schwatinski, S. Pawletta. (2014). *Modellbibliothek für die Interaktion von Robotern in der MATLAB/DEVS-Umgebung auf Basis des SBC-Frameworks*. In: Proc. ASIM-Treffen STS/GMMS, Reutlingen 20./21.02.2014 - ARGESIM Report Nr. 42, ASIM Mitteilung AM 149, ARGESIM/ASIM Pub. Vienna, Austria, 2014, Pages 199-208. (Print ISBN 978-3-901608-42-1).

Multi-Robotersteuerungen mit variablen Interaktionsprinzipien auf Basis des Simulation Based Control Frameworks und dem Discrete Event System Specification Formalismus

Birger Freymann, Thorsten Pawletta, Sven Pawletta
Hochschule Wismar, Forschungsgruppe CEA
{birger.freymann, thorsten.pawletta, sven.pawletta}@hs-wismar.de

Softwarelösungen zur Programmierung von Robotersystemen sind heutzutage zumeist herstellerspezifisch und lassen sich nicht für Roboter anderer Hersteller verwenden. Dennoch besteht der Wunsch unterschiedliche Interaktionen zwischen Robotern verschiedener Hersteller zu ermöglichen. Ausgehend von einer Analyse möglicher Interaktionsprinzipien für Industrieroboter, wird mit dem Simulation Based Control (SBC) ein Framework zur durchgängigen Entwicklung von ereignisorientierten Steuerungen nach dem Rapid Control Prototyping (RCP) Ansatz vorgestellt. Die Umsetzung des SBC erfolgt hierbei mit dem Parallel Discrete Event System (PDEVS) Formalismus, welcher eine systemtheoretische Spezifikation und Ausführungsalgorithmen für dynamische Systeme definiert. Dieser wird durch den $PDEV_{SBC}$ -Ansatz um RCP-Fähigkeit erweitert. Basierend auf dem SBC-Framework und $PDEV_{SBC}$ wird ein Konzept zur Entwicklung einer Modellbibliothek für interagierende Roboter in der MATLAB-Umgebung vorgestellt.

1 Einleitung

Die angedachte Industrie 4.0 hat das Ziel, die Produktion an die Anforderungen voll vernetzter, intelligenter Fabriken anzupassen. Dies bedingt unter anderem ein hohes Maß an Wandlungsfähigkeit und Ressourceneffizienz der eingesetzten Fertigungsverfahren. In diesem Zusammenhang ist die Verwendung von Industrierobotern als Fertigungssystem angedacht. Industrieroboter werden von diversen Hersteller angeboten. Neben unterschiedlichen Robotertypen, welche speziell für bestimmte Anwendungsgebiete optimiert sind, unterscheiden sich auch die Softwarelösungen zum Programmieren dieser Roboter stark, sind herstellerspezifisch und lassen sich zumeist nicht für herstellerfremde Roboter verwenden. So wurden langjährige Normierungsbemühungen für die explizite Roboterprogrammierung, wie die IRL (Industrial Robot Language) und deren Nachfolger die PLR (Programming Language for Robots), durch die Roboterhersteller bis heute größtenteils ignoriert. Diese Gegebenheit führt zu der

Einschränkung bei einer gemeinsamen Nutzung von Robotern unterschiedlicher Hersteller beziehungsweise macht sie unmöglich. Flexible Applikationen, bei denen mehrere Roboter interagieren sollen, sind somit zumeist auf die Hard- und Softwarelösung eines Herstellers begrenzt oder nicht vollständig umsetzbar. Aufbauend auf dem Simulation Based Control (SBC) Framework für die ereignisorientierte Steuerungsentwicklung nach [13] 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 [16, 11] bereits erfolgreich bei der Steuerungsentwicklung für nicht interagierende Roboter unterschiedlicher Hersteller angewendet. Konkrete Robotertypen werden über eine spezielle Schnittstelle integriert. Die softwaretechnische Umsetzung des SBC-Frameworks erfolgte in bei-

den Fällen in der MATLAB-Umgebung. Die Arbeiten in [11] 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 [20] eine DEVS-Toolbox für MATLAB [5] entwickelt. Diese unterstützt eine systemtheoretische Modellierung und Simulation ereignisorientierter Systeme. Hinsichtlich eines durchgängigen Einsatzes der Modelle bis zum operativen Betrieb unter Echtzeitbedingungen wurde in [15, 17] der DEVS konforme $PDEV_{S_{RCP}}$ -Formalismus definiert.

Basierend auf SBC und DEVS wird in diesem Beitrag ein Konzept zur Implementierung einer Multi-Robotersteuerung mit variablen Interaktionsprinzipien vorgestellt. Dabei stellen insbesondere die zeitliche Synchronisation und der von der jeweiligen Interaktion abhängige Informationsaustausch zwischen Modellkomponenten eine große Herausforderung dar. Einige 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 bekannte Interaktionsprinzipien für Industrieroboter, das SBC-Framework und den $PDEV_{S_{RCP}}$ -Formalismus werden nachfolgend robotertypunabhängige Modellkomponenten zur Umsetzung einer prototypischen Robotersteuerung erläutert.

2 Interaktionen zwischen Industrierobotern

Die Entwicklung von Multi-Robotersteuerungen mit variablen Interaktionsprinzipien kann allgemein als Anwendungsentwicklung im Kontext verteilter Systeme (VS) aufgefasst werden. Ein VS ist nach [3] als ein System definiert, in dem sich Hard- und Softwarekomponenten auf vernetzten Computern befinden und über den Austausch von Nachrichten miteinander kommunizieren. Robotersysteme einer Multi-Roboterapplikation können hierbei als eigenständige Komponenten eines VS angesehen werden, welche diese Anforderungen erfüllen. Im Sinne einer Multi-Robotersteuerung werden diese Robotersysteme ihre Dienste teilen und damit Aufgaben realisieren, die mit einem einzelnen Robo-

tersystem nicht umsetzbar sind. Diese Form der Interaktion zwischen unterschiedlichen Robotersystemen wird nachfolgend als Interaktionsprinzip bezeichnet.

2.1 Kommunikationsmodelle

Zur Implementierung sämtlicher Interaktionsprinzipien für Industrieroboter (IAPs) ist ein Nachrichtenaustausch erforderlich. Einzelne IAPs unterscheiden sich in der Art der Nachricht und durch die Reaktion auf die enthaltenen Daten. Grundsätzlich können alle IAPs auf zwei wesentliche Kommunikationsmodelle (KM) zurückgeführt werden. Nach [8] beschreibt ein KM ein Interaktionsmuster zwischen Kommunikationpartnern. Hierbei werden im wesentlichen zwei KM unterschieden, die in Abbildung 1 dargestellt sind.

Synchrone Kommunikation

Bei der synchronen Kommunikation ist der Sender (S) einer Nachricht so lange in seiner weiteren Ausführung blockiert, bis er vom Empfänger (R) eine Antwort auf seine Nachricht erhält.

Asynchrone Kommunikation

Bei der asynchronen Kommunikation ist der Sender nicht blockiert. Sender und Empfänger werden parallel ausgeführt.

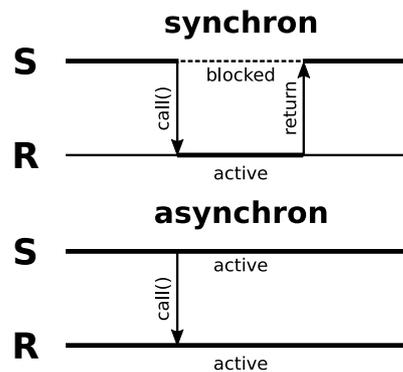


Abbildung 1: Kommunikationsmodelle nach [8]

KM bilden die Basis zur Umsetzung komplexer IAPs.

2.2 Interaktionsprinzipien

Interaktionsprinzipien von Knickarmrobotern lassen sich, basierend auf der allgemeinen Klassifizierung in [10, 18], 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 physische 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 durch Roboter in einer bestimmten Reihenfolge.

Mobilität

Allgemein wird unter Mobilität die Änderung von Arbeitsräumen einer physischen 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. Ein typisches Beispiel in der Robotik ist die gemeinsame Bewegung eines Objektes wie z.B. bei 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 dann 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.

3 Konzeptionelle Grundlagen

Nachfolgend wird das SBC-Framework, die Robotic Control and Visualization Toolbox for Matlab/Simulink (RCV) und der $PDEV_{S_{RCP}}$ -Formalismus vorgestellt. Sowohl das SBC-Framework, als auch $PDEV_{S_{RCP}}$ bilden die Basis für die Entwicklung der Modellbibliothek. Eine herstellerunabhängige Steuerung von Industrierobotern wird durch die RCV Toolbox als Middleware mit dazugehörigen Interface ermöglicht.

3.1 Simulation Based Control Framework (SBC)

Dieser Abschnitt basiert auf [11, 6] 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 wie in Abbildung 2 dargestellt. 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 (SM) wird schrittweise bis zu einem operativen Steuerungsprogramm (CS) 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) 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 mit einer Visualisierung unterstützt und in der gleichen Weise als Schnittstelle zum realen Prozess agieren kann.

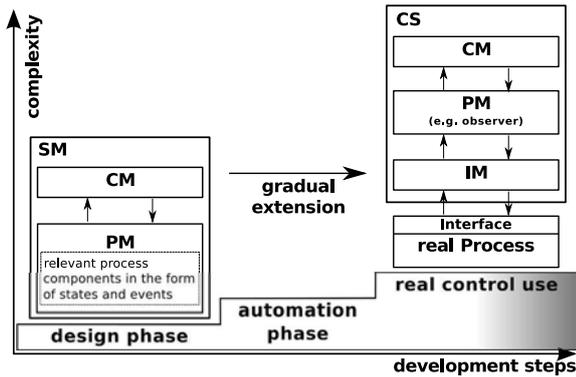


Abbildung 2: Vorgangsmodell und SBC-Framework

3.2 Discrete Event System Formalismus

DEVS wurde 1976 von Zeigler [20] 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 Modellarbeitung aus. Bei der Modellspezifikation wird zwischen zwei DEVS-Systemtypen unterschieden. Das dynamische Verhalten wird mit atomaren (atomic) DEVS-Systemen abgebildet. Daneben gibt es gekoppelte (coupled) DEVS-Systeme, welche eine Komposition aus atomic beziehungsweise coupled DEVS-Systemen beschreiben. Beide Systemtypen, atomic und coupled DEVS, definieren kompatible Ein- und Ausgänge. Die Spezifikation der Systemdynamik in atomic DEVS erfolgt mit einer Menge festgelegter Funktionen, ähnlich einem endlichen Zustandsautomaten. Der Parallel DEVS (PDEVS) Formalismus ist eine Weiterentwicklung des 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 kön-

nen. 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:

$$PDEVS = (X, Y, S, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta) \quad (1)$$

X	set of input values
S	set of state values
Y	set of output values
δ_{int}	internal state transition function $\delta_{int} : S \rightarrow S$
δ_{ext}	external state transition function $\delta_{ext} : Q \times X \rightarrow S$ $Q = \{(s, e) s \in S, 0 \leq e \leq ta(s)\}$ e time elapsed
δ_{con}	confluent state transition function $\delta_{con} : Q \times X \rightarrow S$
λ	output funktion $\lambda : S \rightarrow Y$
ta	time advance funktion $ta : S \rightarrow R_{0, \infty}^+$

Ein coupled PDEVS oder auch PDEVS-Netzwerk (PDEVN) ist formal definiert als:

$$PDEVN = (X, Y, D, \{M_d\}, \{I_d\}, \{Z_{i,d}\}) \quad (2)$$

X	set of input values $X = \{(p, v) p \in IPorts, v \in X_p\}$
Y	set of output values $Y = \{(p, v) p \in OPorts, v \in Y_p\}$
D	set of indexed subcomponents
M_d	model specification of $d \in D$ with $M_d = \{X, Y, S, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta\}$

- I_d set of influences to $d \in D$ with
 $I_d \subseteq D \cup \{N\}, d \notin I_d$
 $Z_{i,d}$ is the i -to- d output relation with
 $i \in I_d$
 $Z_{i,d} : X \rightarrow X_d, \quad \text{if } i = N$
 $Z_{i,d} : Y_i \rightarrow Y, \quad \text{if } d = N$
 $Z_{i,d} : Y_i \rightarrow X_d, \quad \text{if } d \neq N \text{ and } i \neq N$

Der $PDEV_{S_{RCP}}$ -Formalismus ist ebenfalls eine Erweiterung des Classic DEVS-Formalismus und wird in [16] detailliert beschrieben. $PDEV_{S_{RCP}}$ 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 $PDEV_{S_{RCP}}$ unmittelbar auf PDEVS aufbaut, können $PDEV_{S_{RCP}}$ -Modelle direkt mit einer PDEVS-Simulationsumgebung ausgeführt werden, wobei diese aufgrund der eingeführten Erweiterungen auch als Echtzeitumgebung genutzt werden kann. Dadurch ist es möglich, $PDEV_{S_{RCP}}$ -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 [20, 9] nicht zur Umsetzung des SBC-Frameworks geeignet. Die Namensgebung $PDEV_{S_{RCP}}$ weist auf die enge Verwandtschaft zu PDEVS hin und auf die durchgängige Nutzbarkeit der Modelle gemäß dem Vorgehensmodell des Rapid Control Prototyping (RCP) nach [1].

Ein $PDEV_{S_{RCP}}$ ist formal definiert als:

$$PDEV_{S_{RCP}} = (X, Y, S, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta, A) \quad (3)$$

- $Y, S, \delta_{ext}, \delta_{int}, \delta_{conf}, ta$ analog atomic PDEVS
 X set of input values
 $X = X_{model} \cup X_{clock}$
 set of model based input events
 $X_{model} = \{(p, v) | p \in IPorts, v \in X_p\}$
 set of input events from RTC
 $X_{clock} = \{('clock', v) | v \in R^+\}$
 λ combined output and activity function
 $\lambda : S \rightarrow Y \times A$

Die Definitionen von $Y, S, \delta_{int}, \delta_{ext}, \delta_{conf}, ta$ 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 Eingangsereignisse setzt sich aus den Mengen X_{model} und X_{clock} zusammen. Dabei umfasst X_{model} die Eingangsereignisse gewöhnlicher PDEVS-Modelle sowie von der Umgebung. Die Eingangsmenge X_{clock} wird von einer Echtzeituhr (RTC) generiert. Die Ausgabe-funktion λ definiert analog zu PDEVS die Berechnung von Ausgangsereignissen. 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 $PDEV_{S_{RCP}}$ -Modells zeigt Abbildung 3. Aus Übersichtsgründen ist die Behandlung zeitgleicher Ereignisse mit der δ_{conf} Funktion nicht dargestellt. Wie in [16] dargestellt, legt diese in der Regel fest, ob für den aktuellen Zustandsübergang zuerst δ_{int} oder δ_{ext} auszuführen ist.

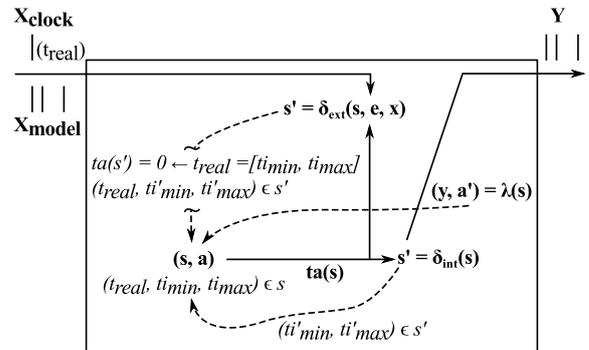


Abbildung 3: Dynamisches Verhalten eines atomic $PDEV_{S_{RCP}}$ nach [16].

3.3 Robotic Control & Visualization Toolbox for MATLAB

Wie eingangs erwähnt, ist die Programmierung von Robotersystemen heutzutage durch die Verwendung von roboterspezifischer Programmiersprachen geprägt. Diese

erschweren die Entwicklung von herstellerunabhängigen Multi-Robotersteuerungen bzw. machen diese unmöglich. Im Weiteren erfolgt in ingenieurtechnischen Anwendungen die Steuerungsentwicklung zumeist in wissenschaftlich-technischen Berechnungsumgebungen (SCEs), wie z.B. MATLAB. Um die Lücke zwischen herstellereigenen Programmiersprachen und SCEs zu schließen, wurde die Robotic Control & Visualization (RCV) Toolbox for MATLAB entwickelt [16, 11, 12, 2, 4]. Die RCV Toolbox besteht aus drei unterschiedlichen Modulen:

1. Menge abstrahierter roboterorientierter MATLAB Funktionen (ähnlich KUKA KRL Kommandos)
2. Menge von MATLAB Funktionen zur Prozessvisualisierung
3. roboterspezifische Kommandointerpreter

Den grundlegenden Aufbau einer Robotersteuerung auf Basis der RCV Toolbox zeigt Abbildung 4 mit:

Control PC

MATLAB-Instanz mit einem Steuerungsprogramm auf Basis von Modul 1

Visualization PC

weitere MATLAB-Instanz zur Prozessvisualisierung auf Basis von Modul 2

Roboter-Controller

Ausführung eines roboterspezifischen Kommandointerpreters (Modul 3)

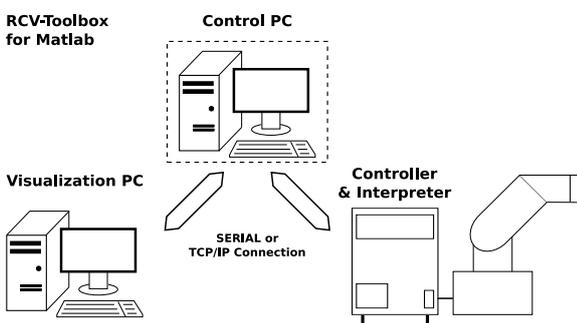


Abbildung 4: Aufbau einer Robotersteuerung auf Basis der RCV Toolbox

Die aktuelle Version der RCV Toolbox for MATLAB unterstützt eine integrierte Steuerungsentwicklung für

KUKA und KAWASAKI Roboter und kann einfach durch neue Kommandointerpreter um weitere Robotertypen erweitert werden. Ebenfalls angemerkt sei, dass durch die Verwendung einer SCE wie MATLAB die Integration von Drittherstellerhardware, z.B. zusätzlicher Sensoren/Aktoren in den Steuerungsprozess ermöglicht wird. Im Sinne des SBC erfüllt die RCV Toolbox for MATLAB alle Anforderungen an eine durchgängige Steuerungsentwicklung von der Entwurfsphase bis zum realen Steuerungseinsatz [4].

4 MATLAB/DEVS Modellbibliothek

Aufbauend auf den zuvor angeführten theoretischen und softwaretechnischen Grundlagen, wird in diesem Kapitel eine generische DEVS-Modellbibliothek zur Simulation und Ausführung von Multi-Robotersteuerungen vorgestellt. Die Implementierung der Komponenten erfolgt mit der DEVS-Toolbox for MATLAB [5] der Forschungsgruppe CEA.

4.1 Umsetzung des SBC mit $PDEV_{S_{RCP}}$

Abbildung 5 zeigt den schematischen Aufbau der Steuerungssoftware (CS) unter Verwendung des SBC, umgesetzt mit $PDEV_{S_{RCP}}$. Das Steuerungsmodell (CM), Prozessmodell (PM) und Interfacemodell (IM) werden als coupled DEVS Modelle, gemäß Definition 2 in Abschnitt 3.2, erstellt.

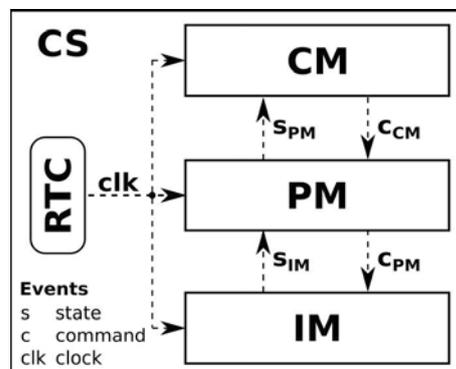


Abbildung 5: Struktur der Steuerungssoftware (CS) nach dem SBC unter Verwendung von $PDEV_{S_{RCP}}$

Basierend auf der mit $PDEV_{S_{RCP}}$ eingeführten Echtzeitsynchronisation wird die CS des SBC um eine Komponente RealTimeClock (RTC) erweitert. Die RTC ist eine spezielle atomic PDEVs Komponente, nach Definition 1 in Abschnitt 3.2, welche zyklisch Zeitereignisse (clk) an CM, PM und IM versendet. Basierend auf den clk-Ereignissen können CM und PM echtzeit-synchrone Steuerereignisse (c) senden. In der Regel werden als Antwort auf (c) Ereignisse Zustandsereignisse (s) gesendet. Durch die Integration einer RTC in CS können zeitabhängige und fehlertolerante Steuerungen realisiert werden. Hinsichtlich einer ausführlichen Darstellung des SBC mit $PDEV_{S_{RCP}}$ sei auf [16, 17] verwiesen.

4.2 Ausgewählte Komponenten der Modellbibliothek

Ausgehend vom SBC sind im PM die wesentlichen Komponenten des realen Prozesses abzubilden. Abbildung 6 zeigt schematisch den Aufbau einer Multi-Robotersteuerung mit den Prozesskomponenten. Eine Übersicht zu den Komponenten der Modellbibliothek (MB) mit Zuordnung zu den Komponenten des SBC zeigt Abbildung 7.

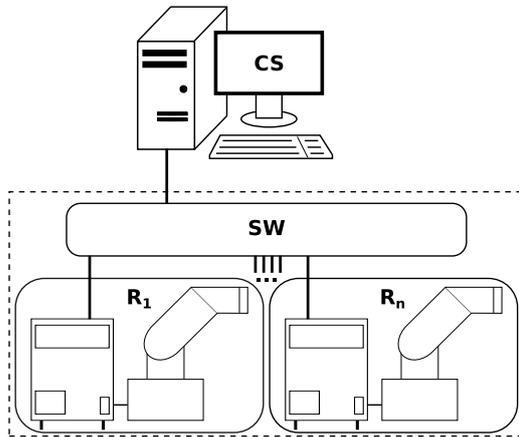


Abbildung 6: Struktur einer Multi-Roboter Applikation

Die Komponenten des CM dienen der Spezifikation der Steuerungslogik, welche applikationsabhängig ist. Darauf wird in Abschnitt 5 eingegangen. Die Komponente RealTimeClock (RTC) wurde in Abschnitt 4.1 erläutert. Die Interface Komponente (I) spezifiziert die Schnittstelle zwischen einer Roboterkomponente im

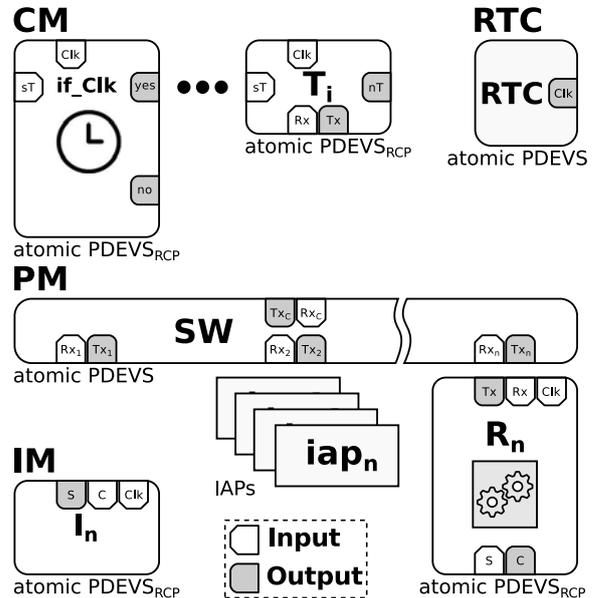


Abbildung 7: Ausgewählte Komponenten der MB

PM und dem Controller des realen Roboters. Sie ist ein atomic $PDEV_{S_{RCP}}$, nutzt zur Kommunikation die abstrahierten Funktionen des Modul 1 der RCV Toolbox gemäß Abschnitt 3.3 und ist dadurch nicht roboterspezifisch. Die Komponenten Switch (SW), Robot (R) und die IAPs realisieren maßgeblich die variablen Interaktionsprinzipien und werden nachfolgend erläutert.

Switch (SW)

Die Komponente ist als atomic PDEVs nach Definition (1) umgesetzt. Abbildung 8 zeigt die Komponente SW mit zugehörigen Input/Output Schnittstellen.

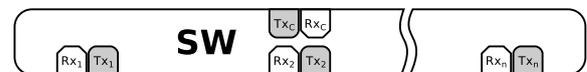


Abbildung 8: Input/Output Schnittstellen des SW

Der SW dient wie im realen Prozess der Kommunikation zwischen den Robotern (R_1, \dots, R_n) im PM sowie dem CM und übernimmt damit das Event-Routing. Dafür definiert SW ein Port-Paar, bestehend aus einem Input (Rx) und einem Output (Tx), pro Kommunikationspartner. Die Anzahl der Port-Paare ist variabel und wird durch einen Systemparameter festgelegt. Das Event-Routing über SW basiert auf Ereignissen, deren Aufbau an Nachrichtenpakete angelehnt ist. Das Ereignis ist ein 4-Tupel wie in (4) definiert.

$$EventPackage = (src, dest, cmd, data) \quad (4)$$

src name of sender
 $src \in D$ D analog Definition (1)
 $dest$ name of destination
 $dest \in D$
 cmd robot specific commands
 $cmd \in \{\emptyset, 'Set', 'Run', 'Stop', 'Done'\}$
 $data$ robot specific data
 $data \in Data$

Mit dem Ereignis können Kommunikationsprinzipien zwischen Komponenten realisiert werden. Der SW ist eine Schlüsselkomponente zur Umsetzung dynamischer IAPs zwischen Robotern.

Robot (R) und IAPs

Die Komponente R ist ein atomic $PDEV_{RCP}$ nach Definition (3). Sie bildet die Zustände eines realen Roboters im PM ab. Zur Realisierung variabler Interaktionsprinzipien speichert sie im Zustand ein konkretes $iap \in IAPs$. Das iap beeinflusst maßgeblich das dynamische Verhalten eines R. Im Kontext von $PDEV_{RCP}$ wird dieses durch die dynamikbeschreibenden Funktionen $(\delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta)$ abgebildet.

Wie in Abbildung 9a dargestellt, kann für jedes $iap \in IAPs$ eine separate atomic $PDEV_{RCP}$ Komponente R definiert werden. Bei diesem Ansatz kann das $iap \in IAPs$ zur Laufzeit nur durch Austausch der Komponente R im PM verändert werden. Diese Lösung erfordert einen strukturdynamischen DEVS Simulator gemäß [20, 7], welcher nicht die Anforderungen von RCP [1] und damit des SBC erfüllt. Ein anderer Ansatz ist der Multi-Modellansatz [14], bei welchem alle IAPs in einer Komponente R gemäß Abbildung 9b definiert werden. Diese Lösung führt zu einer komplexen und schwer wartbaren Komponente R, gestattet aber das Umschalten von IAPs zur Laufzeit durch einen gewöhnlichen PDEVS Simulator.

Ein dritter Ansatz ist in Abbildung 10 dargestellt, welcher sich an Agentenansätzen [19] orientiert. Die Komponente R definiert selbst kein $iap \in IAPs$. Die unterschiedlichen IAPs sind als Komponenten in der MB organisiert. Sie spezifizieren ein dynamisches Verhalten mit den PDEVS typischen Funktionen $(\delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta)$, repräsentieren aber keine eigenständigen PDEVS Komponenten. Die Komponente R empfängt vom CM ein Ereignis ('iap_name', {iap_conditions}) bezüglich eines umzu-

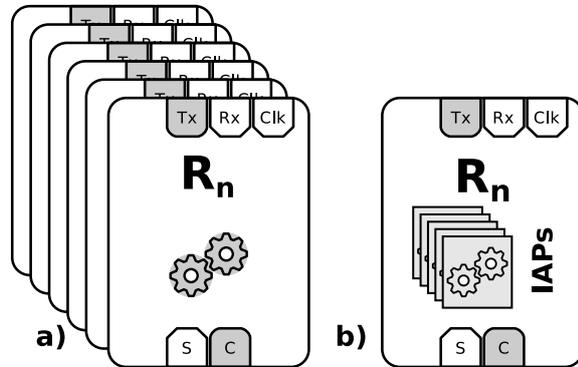


Abbildung 9: Konzepte zur Implementierung von atomic $PDEV_{RCP}$ R mit IAPs

setzenden $iap \in IAPs$. Durch die δ_{ext} beziehungsweise δ_{conf} Funktion der Komponente R wird das anzuwendende iap_i unter Verwendung der MB als Objekt im Zustand $s = (iap_i | iap_i \in IAPs, \dots) \in S$ der Komponente R gespeichert. Die Dynamik-Funktionen der Komponente R greifen über das im Zustand s gespeicherte iap_i auf dessen Dynamik-Funktionen zu, wie in Abbildung 10 gezeigt. Mit dieser Lösung können durch das CM die IAPs der Roboter (R_1, \dots, R_n) im PM zur Laufzeit verändert werden. Darüber hinaus können in einer Multi-Roboterapplikation Roboter nebenläufig mit unterschiedlichen IAPs arbeiten. Im Gegensatz zur zweiten Lösung auf Basis des Multi-Modellansatzes kann die MB einfach um neue IAPs erweitert werden ohne Modifikation der Komponente R.

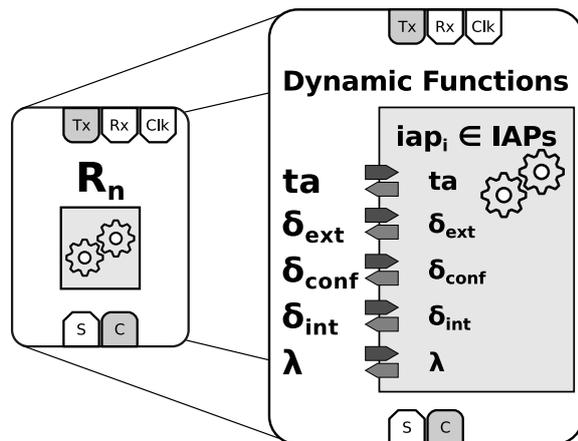


Abbildung 10: Atomic $PDEV_{RCP}$ R mit Schnittstelle zu den $iap \in IAPs$ der MB

5 Konzeptionelles Anwendungsbeispiel

Abbildung 11 zeigt konzeptionell den Aufbau einer Multi-Robotersteuerung für $n \in \mathbb{N}$ Roboter nach dem SBC Ansatz. Atomic DEVS sind durch einen runden Rahmen und coupled DEVS mit eckigen Rahmen dargestellt.

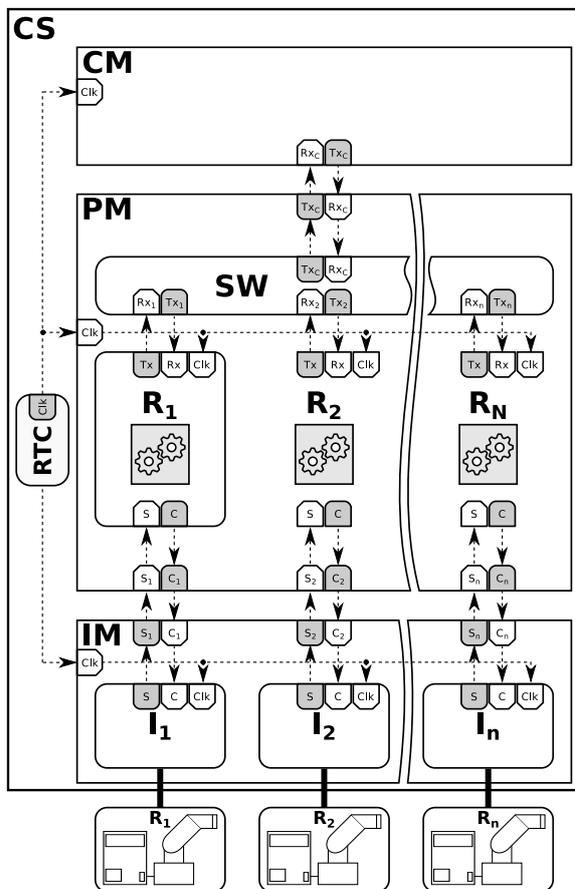


Abbildung 11: Struktur einer Multi-Robotersteuerung mit variablen IAPs

Der grundlegende Aufbau des PM und IM sowie die Verbindung zu den realen Robotern folgt gemäß den Ausführungen in Abschnitt 4. Die Spezifikation anwendungsspezifischer Steuerungsvarianten erfolgt im CM. Steuerungsvarianten können monolithisch mit einem atomic $PDEV_{SRCP}$, analog einem Zustandsautomaten, spezifiziert werden. Bei aufgabenorientierten Steuerungen besteht das CM aus einer Menge von Aufga-

ben (Tasks), die untereinander in Beziehung stehen. Wie in [16] aufgeführt, können Aufgaben durch einzelne $PDEV_{SRCP}$ Komponenten abgebildet und über Kopplungen verbunden werden, wie in Abbildung 12 schematisch gezeigt.

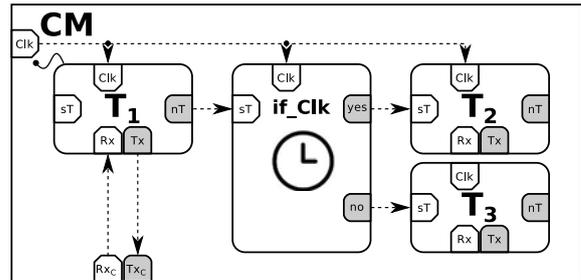


Abbildung 12: Schematische Darstellung einer aufgabenorientierten Steuerungsstruktur (Kopplungen nur teilweise dargestellt)

Das CM in Abbildung 12 besteht aus den drei Aufgaben T_1 , T_2 und T_3 . Nach Ausführung von T_1 entscheidet das $PDEV_{SRCP}$ if_Clk in Abhängigkeit von einer Zeitbedingung, ob T_2 oder T_3 nachfolgend auszuführen ist. Neben der Komponente if_Clk können weitere allgemeine Komponenten zur Ablaufsteuerung definiert und in der MB organisiert werden. Abbildung 13 zeigt einige weitere Beispiele wie: (for_i) für die iterative Abarbeitung von Aufgaben und (case) für Fallunterscheidungen bei der Aufgabenabarbeitung.

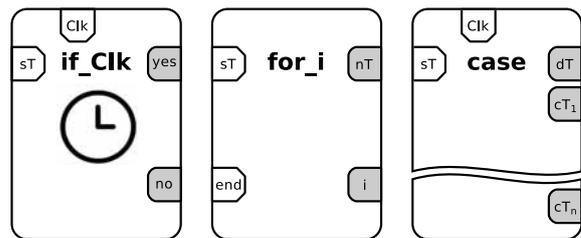


Abbildung 13: Einige Beispiele für Komponenten zur Ablaufsteuerung

6 Zusammenfassung & Ausblick

Es wurde ein Konzept zur Implementierung von Multi-Robotersteuerungen mit variablen Interaktionen vorgestellt. Basierend auf dem SBC und der Analyse des realen Prozesses wurden wesentliche Komponenten

identifiziert und modelliert. Dabei wurden verschiedene Ansätze der Modellierung diskutiert und wiederverwendbare Komponenten in Form von einer Modellbibliothek entworfen. Abschließend wurde ein konzeptionelles Beispiel einer Multi-Robotersteuerung auf Basis der entwickelten Modellbibliothek vorgestellt. In anschließenden Arbeiten sollen die vorgestellten Konzepte anhand von Applikationen validiert werden.

Literatur

- [1] ABEL, D ; BOLLIG, A.: *Rapid Control Prototyping, Methoden und Anwendung*. Springer, 2006
- [2] CHRISTERN, M. ; SCHMIDT, A. ; SCHWATINSKI, T. ; PAWLETTA, T.: *KUKA-KAWASAKI-Robotic Toolbox for Matlab*. http://www.mb.hs-wismar.de/cea/KK_Robotic_Tbx/KK_Robotic_Tbx.html. Version: März 2011. – accessed on: 2012.05.15
- [3] COULOURIS, G. ; DOLLIMORE, J. ; KINDBERG, T.: *Verteilte Systeme. Konzepte und Design*. Pearson Studium, 2002
- [4] DEATCU, C. ; FREYMAN, B. ; SCHMIDT, A. ; PAWLETTA, T.: MATLAB/Simulink Based Rapid Control Prototyping for Multivendor Robot Applications. In: *Proc. of ASIM-Treffen STS/GMMS*, 2015, S. 10, In diesem Tagungsband
- [5] DEATCU, C. ; SCHWATINSKI, T. ; PAWLETTA, T.: *DEVS Toolbox for Matlab*. http://www.mb.hs-wismar.de/cea/DEVS_Tbx/MatlabDEVS_Tbx.html. Version: November 2013. – accessed on 2014.03.05
- [6] FREYMAN, B.: *Entwicklung einer Modellbibliothek für die Interaktion von Robotern in der MATLAB/DEVS Umgebung*, Hochschule Wismar, Diplomarbeit, 2013
- [7] HAGENDORF, O. ; PAWLETTA, T. ; DEATCU, C.: Extended Dynamic Structure DEVS. In: *Proc. of 21th European Modelling & Simulation Symposium*, Puerto de la Cruz, Spanien, September 2009, S. 36–45
- [8] HAMMERSCHALL, U.: *Verteilte Systeme und Anwendungen*. Addison-Wesley Verlag, 2005
- [9] HONG, J. S. ; SONG, H.-S. ; KIM, T. G. ; PARK, K. H.: A Real-Time Discrete Event System Specification Formalism for Seamless Real-Time Software Development. In: *Discrete Event Dynamic Systems: Theory and Applications 7* (1997), Oktober, Nr. 4, S. 355–375
- [10] LÜTH, T.: *Technische Multi-Agenten-Systeme: verteilte autonome Roboter- und Fertigungssysteme*. Hanser, 1998
- [11] MALETZKI, G.: *Rapid Control Prototyping komplexer und flexibler Robotersteuerungen auf Basis des SBC-Ansatzes*, Hochschule Wismar and Universität Rostock, Diss., 2013
- [12] OTTO, J. ; SCHWATINSKI, T. ; PAWLETTA, T.: *KUKA-KAWASAKI-Visualization Toolbox for Matlab*. http://www.mb.hs-wismar.de/cea/MatlabKK_Robotic-and-Visualization_Tbx/MatlabKK_Robotic-Visualization_Tbx.html. Version: April 2011. – accessed on: 2012.05.15
- [13] PAWLETTA, T. ; PAWLETTA, S. ; MALETZKI, G.: Integrated Modeling, Simulation and Operation of High Flexible Discrete Event Controls. In: TROCH, I. (Hrsg.) ; BREITENECKER, F. (Hrsg.) ; TU Wien (Veranst.): *Proc. of Mathematical Modelling MATHMOD 2009*. Wien, Österreich : Argesim Report, Februar 2010 (35), S. 13
- [14] ÖREN, T. I.: Dynamic Templates and Semantic Rules for Simulation Advisors and Certifiers. In: FISHWICK, P. A. (Hrsg.) ; MODJESKI, R. B. (Hrsg.): *Knowledge-Based Simulation: Methodology and Application* Bd. 4, Springer-Verlag, 1991, S. 53–76
- [15] SCHWATINSKI, T. ; PAWLETTA, T.: An Advanced Simulation Approach for Parallel DEVS with Ports. In: *Proc. of Spring Simulation Multiconference 2010, Book 4 - Symposium on Theory of Modeling & Simulation - DEVS*. Orlando, Florida, USA, April 2010, S. 132–139
- [16] SCHWATINSKI, T. ; PAWLETTA, T. ; PAWLETTA, S.: Flexible Task Oriented Robot Controls Using the System Entity Structure and Model Base Approach. In: *SNE - Simulation Notes Europe, ARGESIM/ASIM Pub. TU Vienna 22* (2012), August, S. 107–114

- [17] 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* Bd. 2. Prag, Tschechien, 2010, S. 8
- [18] WETTSTEIN, H: *Systemarchitektur*. München,Wien : Carl Hanser, 1993
- [19] YILMAZ, L. ; ÖREN, T. I.: Dynamic model updating in simulation with multimodels: A taxonomy and a generic agent-based architecture. In: *Proceedings of SCSC 2004 - Summer Computer Simulation Conference*, 2004, S. 3–8
- [20] ZEIGLER, B.P. ; PRAEHOFER, H. ; KIM, T.G: *Theory of Modeling and Simulation: Integrating Discrete Event and Continous Complex Dynamic Systems*. 2nd ed Elsevier Academic Press, 2000

Anwendung einer iterativ lernenden Regelung auf einen Service Roboter mit Mehrkörperkinematik

Sandra Baßler¹, Peter Dünow¹, Mathias Marquardt¹

¹Hochschule Wismar - University of Applied Sciences Technology, Business and Design
s.bassler@stud.hs-wismar.de

Dieser Beitrag beschreibt die Untersuchung zweier Methoden der iterativ lernenden Regelung (ILR) kombiniert mit Computed Torque Control an einem, in SimMechanics simulierten, mobilen Service Roboter mit Mehrkörperkinematik. Diese Art von Robotern soll eingesetzt werden, um Gegenstände zu transportieren beziehungsweise zu manipulieren. Dadurch können zum Beispiel ältere oder in ihrer Bewegung eingeschränkte Menschen unterstützt werden. Der Einsatz von ILR ist bei dieser Anwendung aufgrund sich wiederholender Bewegungsabläufe und der Tatsache, dass die Roboter in der Regel nicht kontinuierlich in Gebrauch sind (und somit Zeit zum Trainieren der Bewegungsabläufe haben), besonders geeignet. Neben der Beschreibung der Steuerungsstruktur und der Trajektoriengenerierung wird eine kurze Ausführung zur Entwicklung des nichtlinearen Dynamikmodells anhand des Lagrange-Formalismus für die Konfigurationsänderung des Roboters gegeben. Eine im dynamischen Modell zu berücksichtigende Besonderheit des mobilen Service Roboters ist das Vorhandensein einer geschlossenen kinematischen Kette. Das aufgestellte Dynamikmodell weist erwartungsgemäß Ungenauigkeiten auf, welche zu einer Regelabweichung beim Einsatz von Computed Torque Control führen. Die zusätzliche Anwendung einer ILR wirkt dieser Regelabweichung entgegen und ermöglicht ein nahezu perfektes Folgen der Sollwert-Trajektorie. Dies wird anhand von Simulationsergebnissen, welche auch zur Gegenüberstellung der ILR-Methoden genutzt werden, gezeigt.

1 Einleitung

Neuste Entwicklungen in verschiedenen technischen Bereichen, wie zum Beispiel den elektrischen Antrieben, eingebetteten Systemen, Energiespeichern, Sensoren et cetera kombiniert mit sinkenden Kosten der Hardwarekomponenten treiben die Entwicklung von Service Robotern in neuen Anwendungsbereichen voran. In diesem Beitrag wird die Anwendung einer iterativ lernenden Regelung in Kombination mit Computed Torque Control für eine bestimmte Klasse mobiler Mehrkörper-Serviceroboter (siehe Abbildung 1) beschrieben. Diese Art von Robotern soll eingesetzt werden, um Gegenstände zu transportieren beziehungsweise zu manipulieren und somit zum Beispiel ältere oder in ihrer Bewegung eingeschränkte Menschen unterstützen. Dabei sollen verschiedene Aufgaben, wie beispielsweise das Steigen von Treppen oder der Transport des Einkaufes in die Wohnung,

erfüllt werden. Aufgrund variierender Umgebungsbedingungen, sollten die Roboter in der Lage sein, ihr Verhalten den Gegebenheiten, wie zum Beispiel einer Laständerung, anzupassen. Der Einsatz von ILR ist bei dieser Anwendung aufgrund sich wiederholender Bewegungsabläufe, wie zum Beispiel Konfigurationsänderungen oder Treppensteigen, besonders geeignet. Zudem sind die Roboter in der Regel nicht kontinuierlich in Gebrauch und haben somit Zeit zum Trainieren der Bewegungsabläufe. In diesem Beitrag wird insbesondere die Konfigurationsänderung betrachtet. Dabei werden zunächst die Steuerungsstruktur und die Trajektoriengenerierung der Achsvariablen erläutert. Anschließend wird auf die Entwicklung des vom Computed Torque Control benötigten dynamischen Modells eingegangen. An dieser Stelle ist zu beachten, dass der Roboter eine geschlossene kinematische Kette aufweist. Dies kann, wie in Abschnitt 3 gezeigt, über Schließbedingungen der kinematischen Kette im



Abbildung 1: Konfigurationen des Serviceroboters [4]

dynamischen Modell berücksichtigt werden. Weiterhin werden zwei ILR-Ansätze kombiniert mit Computed Torque Control auf ein SimMechanics-Modell des vorgestellten Roboters angewandt und anhand der resultierenden Simulationsergebnisse bewertet. Der betrachtete Roboter besteht aus vier Armen mit drei Konfigurationsantrieben und drei Fahrtrieben. Genauer betrachtet besteht der hintere Arm aus zwei Armen mit einem weiterem Antrieb zur Lenkung. Für die Betrachtungen in diesem Beitrag wird angenommen, dass die Räder immer gleich ausgerichtet sind und der Lenkantrieb somit nicht relevant ist.

2 Steuerungsstruktur und Trajektoriengenerierung

Die Steuerungsstruktur des Roboters wird in Abbildung 2 anhand des Beispiels der Konfigurationsänderung dargestellt. Die Sollwerte für ein Szenario (beispielsweise für eine in Abbildung 3 gezeigte Umkonfiguration) werden vom übergeordneten System als Ergebnis von Steuerungsfunktionen oder aus einer Datenbank vorgegeben. Für ein gutes Fahrverhalten beziehungsweise gleichmäßige Bewegungen erfolgt eine Interpolation zur Generierung von Zwischenwerten. Die Sollwerte repräsentieren für gewöhnlich die Position und Orientierung eines Tool Center Points (TCP). Für die betrachtete Anwendung beschreiben diese hingegen beispielsweise die Korbhöhe, die Lage des Schwerpunktes oder direkt, die in Abbildung 4 gezeigten, unabhängigen Roboterparameter, die eine einfache Generierung von Sollwerten ermöglichen. Die Roboterparameter werden mittels inverser Kinematik in die Achsvariablen umgerechnet. Anschließend wer-

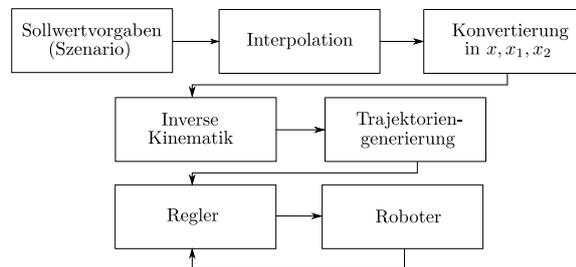


Abbildung 2: Steuerungsstruktur

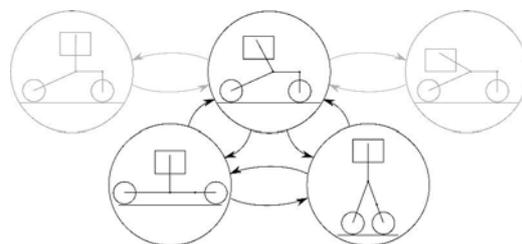


Abbildung 3: Beispiele für die Konfigurationsänderung

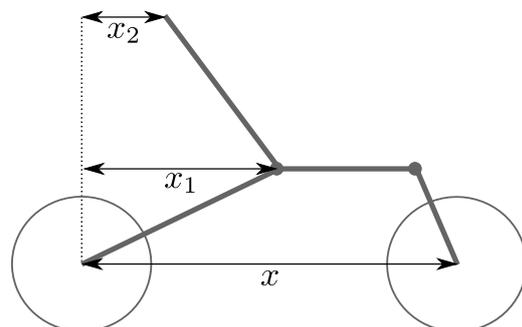


Abbildung 4: Roboterparameter

den die Sollwerttrajektorien, das heißt die Winkel-, Geschwindigkeits- und Beschleunigungsprofile generiert und an die Regelung übergeben.

2.1 Trajektoriengenerierung

In der Trajektoriengenerierung des Steuerungssystems werden die Winkel-, Geschwindigkeits- und Beschleunigungsprofile des Achsvariablen berechnet. Um eine weiche Bewegung des Roboters zu erzielen, wird ein Polynomansatz mit Polynomen fünfter Ordnung verwendet [3]:

$$q = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5. \quad (1)$$

Dabei finden die folgenden fünf Randbedingungen Anwendung:

1. $q(0) = 0$
(Winkel relativ zum Start $q(T) = \Delta q$)
2. $\dot{q}(0) = \dot{q}_1$ (Achsgeschwindigkeit im Startpunkt)
3. $\dot{q}(T) = \dot{q}_2$ (Achsgeschwindigkeit im Endpunkt)
4. $\ddot{q}(0) = 0$ (Achsbeschleunigung im Startpunkt)
5. $\ddot{q}(T) = 0$ (Achsbeschleunigung im Endpunkt)

Die Achsgeschwindigkeiten im Start- und Endpunkt werden je nach Szenario vorgegeben. Die Koeffizienten der Polynome werden wie folgt bestimmt:

$$\begin{aligned}
 a_0 &= 0; & a_1 &= \dot{q}_1; & a_2 &= 0 \\
 a_3 &= \frac{2}{T^3}(5q - 3T\dot{q}_1 - 2\dot{q}_2T) \\
 a_4 &= \frac{1}{T^4}(15q - 8T\dot{q}_1 - 7\dot{q}_2T) \\
 a_5 &= \frac{3}{T^5}(2q - T\dot{q}_1 - \dot{q}_2T).
 \end{aligned} \tag{2}$$

3 Dynamisches Modell

Computed Torque Control erfordert ein dynamisches Modell beziehungsweise die Bewegungsgleichung des Roboters:

$$\tau = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) \tag{3}$$

worin τ der Vektor der Achsmomente, $\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}}$ die Massenmatrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ der Anteil der Coriolis- und Zentrifugalkräfte, $\mathbf{g}(\mathbf{q})$ der Anteil der Gravitation und $\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}})$ die Reibung ist. Das dynamische Modell eines solchen Mehrkörpersystems kann, wie beispielsweise in [1], [3] oder [4] beschrieben, anhand des Lagrange-schen Algorithmus bestimmt werden.

Die Lagrangefunktion eines mechanischen Systems $\mathcal{L} = \mathcal{T} - \mathcal{U}$ entspricht der Differenz zwischen kinetischer \mathcal{T} und potentieller Energie \mathcal{U} des Systems. Die Lagrange-schen Gleichung, wie beispielsweise in [3] gegeben, lautet:

$$\tau = \frac{d}{dt} \frac{\delta \mathcal{L}}{\delta \dot{\mathbf{q}}} - \frac{\delta \mathcal{L}}{\delta \mathbf{q}}. \tag{4}$$

Tabelle 1: Parameter

	m	a	l
Arm 1	0.5 kg	0.3 m	0.15 m
Arm 2	0.5 kg	0.17 m	0.085 m
Arm 3	0.5 kg	0.13 m	0.065 m
Arm 4	0.5 kg	0.25 m	0.125 m
Box	0.5 kg	0.2 m	0.1 m

Zur Aufstellung der Bewegungsgleichungen des gegebenen Roboters nutzen wir einen auf Euler-Lagrange basierenden systematischen Algorithmus, welcher beispielsweise in [1] und [2] beschrieben wird.

Folgende vereinfachende Annahmen wurden getroffen:

- die Räder werden vernachlässigt,
- alle Arme sind Zylinder,
- die Kiste ist ein Quader,
- die Einflüsse der Motoren werden hier nicht berücksichtigt,
- Reibung wird erst einmal vernachlässigt,
- die Lenkung wird als starr angenommen.

Abbildung 5 zeigt die vereinfachte Darstellung des Roboters. Dabei sind m_1, \dots, m_4 die Massen der einzelnen Arme und m_5 die Masse der Kiste. Die Winkel $\theta_1, \dots, \theta_5$ sind die Achsvariablen. Weiterhin sind a_1, \dots, a_4 die Längen der jeweiligen Arme, l_1, \dots, l_5 die Abstände der Massenmittelpunkte zur entsprechenden Achse und I_1, \dots, I_5 die Trägheitsmomente im Bezug auf den Massenmittelpunkt der einzelnen Arme beziehungsweise der Kiste. Tabelle 1 fasst die verwendeten Roboterdaten zusammen.

Da alle Räder des Roboters auf dem Boden stehen, befindet sich zwischen den Rädern eine virtuelle translatorische Achse, welche durch die gestrichelte Linie in Abbildung 5 dargestellt ist. Diese virtuelle Achse hat, da sie keine Masse aufweist, keinen direkten Einfluss auf das dynamische Modell, dennoch ist sie als Schließbedingung einzubeziehen. Des Weiteren ist die Verbindung zwischen Aufhängung und Schwerpunkt der Kiste immer lotrecht, wodurch sich die zweite

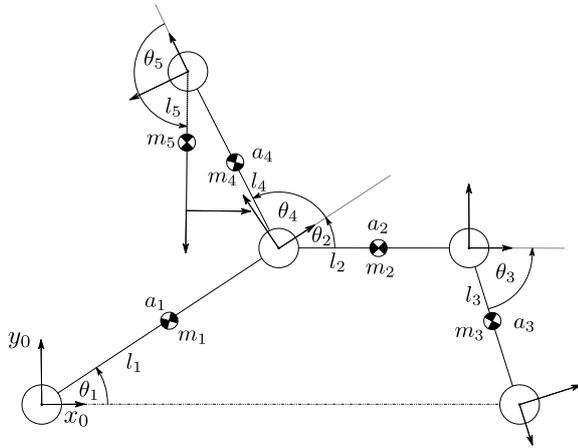


Abbildung 5: Der Roboter als Mehrkörpersystem

Schließbedingung ergibt. So ergeben sich für den Roboter zwei geschlossene Ketten, die bei der Aufstellung des dynamischen Modells einzubeziehen sind.

Das dynamische Modell eines Systems mit geschlossener Kette wird wie folgt bestimmt [1]:

1. Die geschlossene Kette wird an einer nicht angetriebenen Achse aufgeschnitten, sodass sich eine Baumstruktur beziehungsweise offene Kette ergibt.
2. Nach der Aufstellung des dynamischen Modells für das Mehrkörpersystem mit offener Kette wird dieses (anhand der aus den Schließbedingungen resultierenden Abhängigkeiten der nicht angetriebenen Achsen von den angetriebenen Achsen) in das Modell für das Mehrkörpersystem mit geschlossener Kette transformiert.

Nachfolgend werden die wichtigsten Schritte zur Aufstellung der Bewegungsgleichungen beziehungsweise zur Berechnung von τ_2 , τ_3 und τ_4 für die angetriebenen Achsen in zwei Abschnitten beschrieben. Zuerst wird das dynamische Modell für das Mehrkörpersystem mit Baumstruktur ohne geschlossene Kette bestimmt. Anschließend wird das Ergebnis in das Modell für das System mit geschlossener Kette transformiert.

3.1 Baumstruktur

In diesem Abschnitt wird das dynamische Modell des Mehrkörpersystems mit offener Kette bestimmt. Das System wird mit zwei zusätzlichen generalisierten Koordinaten beschrieben, sodass sich die fünf Winkel $\theta_1 \dots \theta_5$ ergeben.

$$\mathbf{q} = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5] \quad (5)$$

Wie bereits erwähnt verwendet der Lagrangesche Algorithmus die Differenz aus kinetischer und potentieller Energie, um die Bewegungsgleichungen aufzustellen. Die kinetische Energie ergibt sich wie folgt [1]:

$$\mathcal{T} = \sum_{i=1}^n \frac{1}{2} (m_i \dot{\mathbf{q}}^T \mathbf{J}_P^{iT} \mathbf{J}_P^i \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{J}_O^i \mathbf{R}_i \mathbf{I}_i^T \mathbf{R}_i^T \mathbf{J}_O^i \dot{\mathbf{q}}) \quad (6)$$

wobei \mathbf{J}_P^i die Jacobimatrix der translatorischen und \mathbf{J}_O^i die Jacobimatrix der rotatorischen Bewegung des i -ten Arms ist, m_i die zugehörige Masse, \mathbf{R}_i die Rotationsmatrix des i -ten Frames und \mathbf{I}_i^T die Trägheit bezüglich des Massenmittelpunktes des i -ten Arms ist. Die potentielle Energie ergibt sich aus:

$$\mathcal{U}(\mathbf{q}) = - \sum_{i=1}^n m_i \mathbf{g}_0^T \mathbf{p}_i \quad (7)$$

mit dem Vektor \mathbf{p}_i zum Massenmittelpunkt des jeweiligen Arms im Basisframe, der zugehörigen Masse m_i und dem Gravitationsvektor \mathbf{g}_0 im Basisframe beispielsweise $\mathbf{g}_0 = [0 \quad -g \quad 0]^T$ (y ist die vertikale Achse).

Die Anwendung des Lagrangeschen Formalismus führt zu den Parametermatrizen $\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}}$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ und $\mathbf{g}(\mathbf{q})$. Wie in [1] oder [2] gezeigt kann die Matrix $\mathbf{B}(\mathbf{q})$ systematisch mit den Jacobimatrizen, den Rotationsmatrizen der entsprechenden Frames \mathbf{R}_i , den Massen m_i und den Trägheitsmomenten \mathbf{I}_i^T aufgestellt werden:

$$\mathbf{B}(\mathbf{q}) = \sum_{i=1}^n (m_i \mathbf{J}_P^{iT} \mathbf{J}_P^i + \mathbf{J}_O^i \mathbf{R}_i \mathbf{I}_i^T \mathbf{R}_i^T \mathbf{J}_O^i) \quad (8)$$

Den Anteil der Gravitation erhält man durch partielle Ableitung der potentiellen Energie:

$$\mathbf{g}(\mathbf{q}) = \left[\frac{\partial \mathcal{U}}{\partial q_1} \quad \dots \quad \frac{\partial \mathcal{U}}{\partial q_n} \right]^T \quad (9)$$

und die Elemente der **C**-Matrix, welche die Beiträge der Coriolis- und Zentrifugalkräfte zusammenfasst, werden anhand der folgenden Formeln bestimmt [1]:

$$c_{ij} = \sum_{k=1}^n c_{ijk} \dot{q}_k \quad (10)$$

$$c_{ijk} = \frac{1}{2} \left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i} \right). \quad (11)$$

Die Jacobimatrizen werden folgendermaßen aufgestellt:

$$\mathbf{J}_P^i = [\mathbf{j}_{P1}^i \quad \dots \quad \mathbf{j}_{Pi}^i \quad \mathbf{0} \quad \dots \quad \mathbf{0}] \quad (12)$$

$$\mathbf{J}_O^i = [\mathbf{j}_{O1}^i \quad \dots \quad \mathbf{j}_{Oi}^i \quad \mathbf{0} \quad \dots \quad \mathbf{0}] \quad (13)$$

mit

$$\mathbf{j}_{Pj}^i = \begin{cases} \mathbf{z}_{j-1} & \text{translatorische Achse} \\ \mathbf{z}_{j-1} \times (\mathbf{p}_i - \mathbf{p}_{j-1}) & \text{rotatorische Achse} \end{cases} \quad (14)$$

und

$$\mathbf{j}_{Oj}^i = \begin{cases} \mathbf{0} & \text{translatorische Achse} \\ \mathbf{z}_{j-1} & \text{rotatorische Achse} \end{cases} \quad (15)$$

Hier sind sowohl eine Beschreibung des Roboters in der Framedarstellung, als auch die Vektoren zu den jeweiligen Massenmittelpunkten der Arme \mathbf{p}_i erforderlich. Dabei ist \mathbf{z}_{j-1} der Einheitsvektor der z-Achse des Frames $j-1$, während \mathbf{p}_{j-1} der Positionsvektor zum Ursprung des Frames $j-1$ im Bezugskoordinatensystem ist.

Die Jacobimatrizen für den gegebenen Roboter sind im Anhang zu finden. Die Ergebnisse für **B**, **C** und **g** können hier aufgrund der Komplexität der Gleichungen in den Matrizen nicht angegeben werden.

3.2 Geschlossene Kette

Durch eine geschlossene Kette ergibt sich eine funktionale Abhängigkeit der generalisierten Koordinaten, welche genutzt wird, um das dynamische Modell der offenen Kette τ in das dynamische System der geschlossenen Kette τ_c zu transformieren. Zuerst werden die generalisierten Koordination in angetriebene

\mathbf{q}_a^T und nicht angetriebene Achsen \mathbf{q}_u^T unterteilt.

$$\mathbf{q}_o = [\mathbf{q}_a^T \quad \mathbf{q}_u^T]^T \quad (16)$$

Anschließend wird die funktionale Beziehung zwischen den angetriebenen und nicht angetriebenen Achsen $\mathbf{q}_u = \mathbf{q}_u(\mathbf{q}_a)$ aufgestellt. Das heißt, die Variablen der nicht angetriebenen Achsen werden mithilfe der Variablen der angetriebenen Achsen ausgedrückt. Die Ableitung dieser Beziehung nach der Zeit entspricht dabei der Beziehung der Achsgeschwindigkeiten.

$$\dot{\mathbf{q}}_o = \Upsilon \dot{\mathbf{q}}_a \quad (17)$$

mit

$$\Upsilon = \begin{bmatrix} \mathbf{I} \\ \frac{\partial \mathbf{q}_u}{\partial \mathbf{q}_a} \end{bmatrix} \quad (18)$$

Bezugnehmend auf [1] werden die Momente der angetriebenen Achsen mittels der Transformationsmatrix Υ berechnet:

$$\tau_c = \Upsilon^T \tau_o \quad (19)$$

mit

$$\tau_o = [\tau_a^T \quad \tau_u^T]^T. \quad (20)$$

Die Elemente von τ_o können folgendermaßen bestimmt werden:

$$\tau = \mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (21)$$

wobei die Momente jeder Achse τ_i in Momente der angetriebenen τ_a^T und nicht angetriebenen Achsen τ_u^T aufgeteilt werden.

Für den gegebenen Roboter sind θ_1 und θ_5 die Winkel der nicht angetriebenen Achsen. So ergeben sich die folgenden Trägheitsmomente:

$$\tau_a = [\tau_2 \quad \tau_3 \quad \tau_4]^T \quad \tau_u = [\tau_1 \quad \tau_5]^T.$$

Die Transformationsmatrix ergibt sich zu:

$$\Upsilon = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ & F_1 & \\ & F_2 & \end{bmatrix}$$

wobei

$$F_1 = \begin{bmatrix} \frac{\partial \theta_1}{\partial \theta_2} & \frac{\partial \theta_1}{\partial \theta_3} & 0 \end{bmatrix}$$

und

$$F_2 = \begin{bmatrix} \frac{\partial \theta_5}{\partial \theta_2} & \frac{\partial \theta_5}{\partial \theta_3} & \frac{\partial \theta_5}{\partial \theta_4} \end{bmatrix}.$$

θ_1 wird wie folgt berechnet:

$$a = \sqrt{a_2^2 - 2 \cos(\pi - \theta_3) a_2 a_3 + a_3^2}$$

$$\alpha = \arcsin\left(\frac{a_3 \sin(\pi - \theta_3)}{a}\right)$$

$$\beta = \theta_2 - \alpha$$

$$b = \sqrt{a^2 - 2 \cos(\beta) a a_1 + a_1^2}$$

$$\theta_1 = -\arcsin\left(\frac{a \sin(\beta)}{b}\right)$$

und θ_5 wird folgendermaßen bestimmt

$$\theta_5 = \theta_4 + \frac{\pi}{2} + \theta_1.$$

Zur Bestimmung der Bewegungsgleichung wurde das Computeralgebrasystem MuPAD verwendet. Aufgrund der komplexen mathematischen Ausdrücke F_1 und F_2 in der Transformationsmatrix ist die Komplexität der Matrizen \mathbf{B} , \mathbf{C} und \mathbf{g} erhöht, sodass vereinfachende Maßnahmen in Betracht zu ziehen sind.

3.3 SimMechanics-Modell

Für die vergleichenden Untersuchungen der unterschiedlichen Regelungsalgorithmen wurde ein SimMechanics-Modell des Roboters entwickelt. Dabei wurden folgende Vereinfachungen vorgenommen:

- Die Arme des Roboters werden als Quader angenommen.
- Die Einflüsse der Motoren werden vernachlässigt.
- Die Räder werden als Zylinder betrachtet.

Abbildung 6 zeigt das entwickelte SimMechanics-Modell in der grafischen Ansicht von SimMechanics. Das Modell ist direkt mit der in Simulink im-

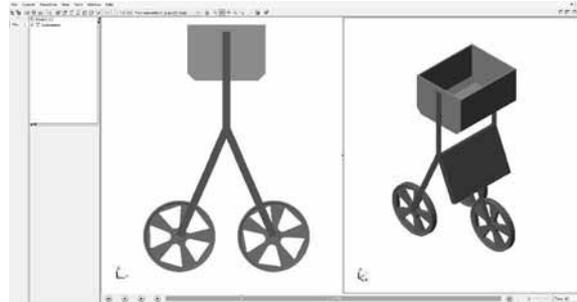


Abbildung 6: SimMechanics-Modell des Roboters

plementierten Regelungsstruktur verbunden, um diese zu untersuchen. Die Eingänge des SimMechanics-Modells sind die Momente der angetriebenen Achsen und die Ausgänge sind die Winkel ($\theta_2, \theta_3, \theta_4$) und die Winkelgeschwindigkeiten ($\dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4$) der entsprechenden Achsen. Das SimMechanics-Modell wird zur Untersuchung der Iterativ lernenden Regelung und des Computed Torque Control, welches das oben beschriebene dynamische Modell der Roboters enthält, verwendet.

4 Iterativ lernende Regelung

Die Konfigurationsänderungen des Roboters sind sich wiederholende Bewegungsabläufe, die wie oben erwähnt vom Roboter trainiert werden können, da dieser nicht kontinuierlich in Gebrauch ist. Auf diese Weise soll die Qualität der Bewegung mit jeder Wiederholung verbessert werden. Die Literatur bietet zahlreiche Ansätze hinsichtlich Analyse und Entwurf von iterativ lernenden Regelungen, wie zum Beispiel in [6], [8] und [7] gezeigt. Abbildung 7 zeigt die den prinzipiellen Aufbau einer iterative lernenden Regelung. Der Istwertverlauf y_k wird über einen Zyklus gespeichert und von der Sollwerttrajektorie w abgezogen. Aus dem resultierende Fehlervektor und dem gespeicherten Stellwertvektor u_k wird durch einen Lernoperator beziehungsweise anhand eines Lerngesetzes der Verlauf Stellgröße u_{k+1} für den nachfolgenden Zyklus bestimmt. Die in diesem Beitrag untersuchten ILR werden, wie in Abbildung 8 dargestellt, in Kombination mit Computed Torque Control angewandt. Der Computed Torque Controller nutzt das nichtlineare dynamische Modell zur Kompensation der Nichtlinearitäten und zur Entkopplung des Mehrgrößensystems[1].

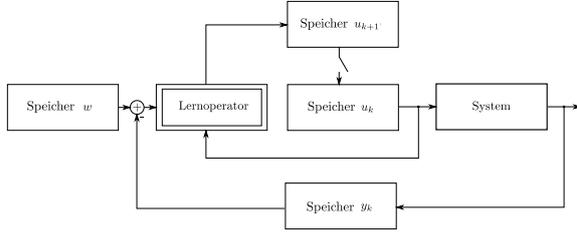


Abbildung 7: Allgemeine Struktur einer iterativ lernenden Regelung [5]

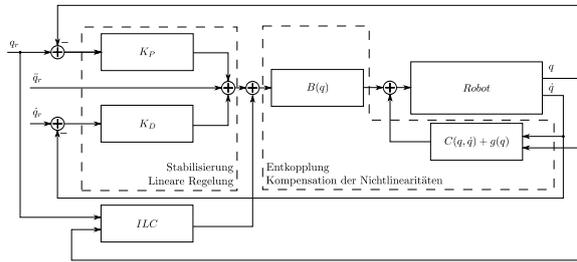


Abbildung 8: Computed Torque Control [1] kombiniert mit ILC

Allerdings führen die vereinfachenden Annahmen zur Aufstellung des dynamischen Modell (siehe Abschnitt 3) zu Modellungenauigkeiten, welche Regelabweichungen hervorrufen. Dies ist ein allgemeingültiges Problem, da es in der Regel nicht möglich ist, ein exaktes Modell aufzustellen. Durch den Einsatz einer iterativ lernenden Regelung gilt es nun diese Abweichungen zu kompensieren. Da eine ILR nur zyklische Störungen ausregeln kann, ist zusätzlich der Einsatz eines Feedback-Regler notwendig [6]. Dieser kann beispielsweise parallel zur ILR arbeiten (siehe Abbildung 8).

4.1 Systemmodellierung

Die im Folgenden untersuchten ILR-Ansätze erfordern ein lineares Systemmodell, einerseits um eine Prüfung auf Stabilität und Konvergenz durchzuführen und andererseits, um den Lernoperator zu entwerfen. Durch die Verwendung von Computed Torque Control vereinfacht sich das Gesamtsystem zu folgender Übertragungsfunktion:

$$G_{ges} = \frac{1}{p^2 + K_D p + K_P}. \quad (22)$$

4.2 Lifted Vector Framework

Das unter anderem in [6] und [9] beschriebene Lifted Vector Framework ist eine gängige Darstellungsform hinsichtlich Analyse und Entwurf iterativ lernender Regelungen. Das Lerngesetz in der Lifted Vector Darstellung lautet [6]:

$$\mathbf{u}_{j+1} = \mathbf{Q}[\mathbf{u}_j + \mathbf{L}e_j]. \quad (23)$$

Darin sind u_j der Stellgrößenvektor und e_j der Fehlervektor im aktuellen Zyklus, \mathbf{L} der Lernfilter (24), \mathbf{Q} der Robustheitsfilter (25) und u_{j+1} der Stellgrößenvektor für den nächsten Zyklus. Zudem ist j der Iterationsindex.

$$\mathbf{Q} = \begin{bmatrix} q_0 & q_{-1} & \cdots & q_{-(N-1)} \\ q_1 & q_0 & \cdots & q_{-(N-2)} \\ \vdots & \vdots & \ddots & \vdots \\ q_{N-1} & q_{N-2} & \cdots & q_0 \end{bmatrix} \quad (24)$$

$$\mathbf{L} = \begin{bmatrix} l_0 & l_{-1} & \cdots & l_{-(N-1)} \\ l_1 & l_0 & \cdots & l_{-(N-2)} \\ \vdots & \vdots & \ddots & \vdots \\ l_{N-1} & l_{N-2} & \cdots & l_0 \end{bmatrix} \quad (25)$$

An dieser Stelle ist zu beachten, dass der Fehlervektor um den relativen Grad m verschoben wird, um die Zeitverzögerung zwischen Ein- und Ausgang des Systems zu kompensieren:

$$\mathbf{e}_j = \begin{bmatrix} e^{(m)} \\ e^{(m+1)} \\ \vdots \\ e^{(m+N-1)} \end{bmatrix}. \quad (26)$$

Der Lernfilter und der Robustheitsfilter müssen so gewählt werden, dass das Stabilitätskriterium [6]

$$\rho(\mathbf{Q}(\mathbf{I} - \mathbf{L}\mathbf{P})) < 1, \quad \rho(\mathbf{A}) = \max_i |\lambda_i(\mathbf{A})| \quad (27)$$

und das Konvergenzkriterium [6]

$$\bar{\sigma}(\mathbf{P}\mathbf{Q}(\mathbf{I} - \mathbf{L}\mathbf{P})\mathbf{P}^{-1}) < 1 \quad (28)$$

$\bar{\sigma}(\cdot)$ ist der maximale Singulärwert

erfüllt werden. Dabei wird das System \mathbf{P} (29) anhand der Markovparameter $p_k = \mathbf{C}\mathbf{A}^{k-1}\mathbf{B}$ dargestellt. Der

Index m steht hier ebenfalls für den relativen Grad während k dem Zeitindex entspricht.

$$\mathbf{P} = \begin{bmatrix} p_m & 0 & \cdots & 0 \\ p_{m+1} & p_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_{m+N-1} & p_{m+N-2} & \cdots & p_m \end{bmatrix}, \quad (29)$$

In der Simulation wurden $\mathbf{Q}(q) = \mathbf{I}$ und $\mathbf{L}(q) = \gamma \mathbf{I}$ mit $\gamma = 300$ gewählt. So ergibt das Stabilitätskriterium

$$\rho(\mathbf{Q}(\mathbf{I} - \mathbf{L}\mathbf{P})) = 0.9905 < 1,$$

und das Konvergenzkriterium

$$\bar{\sigma}(\mathbf{P}\mathbf{Q}(\mathbf{I} - \mathbf{L}\mathbf{P})\mathbf{P}^{-1}) = 0.9992 < 1.$$

Die Ergebnisse der Simulation werden in Abschnitt 4.4 gezeigt.

4.3 Systeminverse als Lernfilter

Die Inverse des Systems ist eine direkte Lösung des Konvergenzkriteriums [5],[6]. Das heißt, bei einem exaktem Modell würde sich bereits nach einer Iteration ein Fehler von Null ergeben. In der Regel ist kein exaktes Modell gegeben, dennoch kann die Anwendung der Systeminversen als Lernoperator zu sehr guten Ergebnissen führen. Eine sich durch die Invertierung ergebende Nichtkausalität kann durch entsprechende Verschiebung des Fehlervektors und des inversen Systems kompensiert werden. Problematisch sind hingegen nichtminimalphasige Systeme, aus deren Invertierung ein instabiles System resultiert. Ein Lösungsansatz dafür wird in [10] gezeigt. Da Modellungenauigkeiten zu einem schlechten Einschwingverhalten führen können, wird typischerweise zusätzlich ein Q-Filter (Tiefpass) eingesetzt [6].

In der nachfolgenden Simulation wurde zunächst auf einen Q-Filter verzichtet und nur die Systeminverse als Lernfilter eingesetzt.

4.4 Simulationsergebnisse

Als Beispielszenario wurde die Änderung der Höhe des Roboters ausgewählt. Dabei richtet sich der Robo-

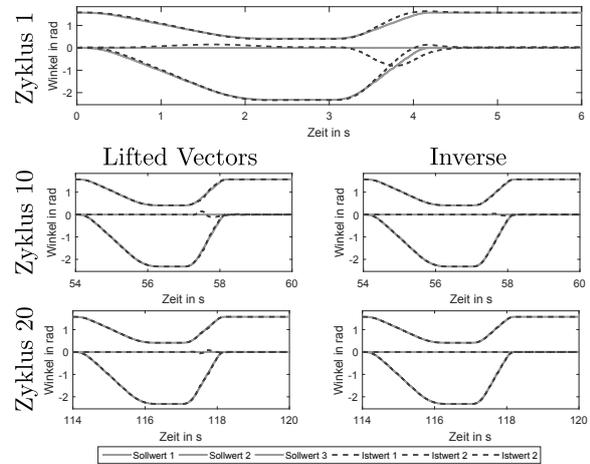


Abbildung 9: Simulationsergebnisse

ter auf, hält die Position circa 1 s und fährt anschließend wieder hinunter. Das Szenario hat eine Dauer von 6 s beziehungsweise 600 Abtastschritten. Die Sollwerttrajektorien für die drei Konfigurationsmotoren wurden mittels Bahnplanung berechnet. Die Simulationsergebnisse sind in Abbildung 9 dargestellt. Im oberen Diagramm ist das Ergebnis des ersten Zyklus beider ILR-Ansätze zu sehen. Da erst nach dem diesem Zyklus eine Stellwerttrajektorie durch den jeweiligen ILR berechnet wird, sind die Istwertverläufe für beide Ansätze identisch. Darunter sind die Ergebnisse für den zehnten Zyklus dargestellt. Beide ILR bewirken eine wesentliche Verbesserung im Vergleich zum ersten Zyklus, dennoch ist in beiden Diagrammen noch ein Regelabweichung zu erkennen. Die Anwendung der Inversen als Lernfilter bewirkt im 20. Zyklus (Diagramm unten rechts) ein nahezu perfektes Folgen der Sollwerttrajektorie, während bei der Lifted Vector Methode noch geringe Abweichungen sichtbar sind. Abbildung 10 zeigt den Verlauf der euklidischen Fehlerlernorm über 100 Zyklen. Hier ist zu erkennen, dass beide Methoden trotz Modellabweichungen Konvergenz aufweisen.

5 Zusammenfassung

Das Hauptanliegen dieses Beitrags war es eine Testumgebung mit einer lauffähigen Implementierung eines Computed Torque Controllers für den beschriebenen Roboter zu realisieren. Basierend auf dieser Te-

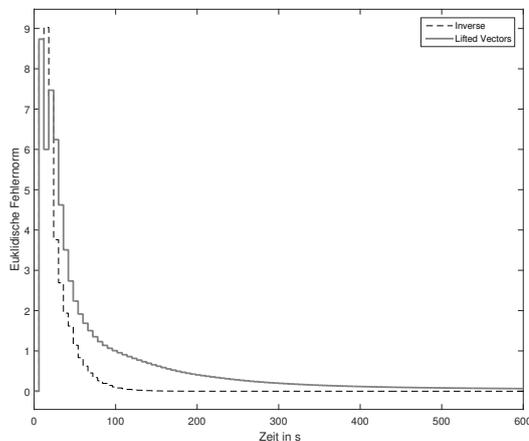


Abbildung 10: Enorm

stumgebung ist es nun möglich weiterführende Regelungsansätze, wie bereits am Beispiel zweier ILR Methoden demonstriert, zu untersuchen. Weiterhin konnte in der Simulation gezeigt werden, dass eine ILR die Möglichkeit bietet, die durch Modellungenauigkeiten entstehende Regelabweichung eines Computed Torque Controllers bei sich wiederholenden Bewegungsabläufen zu kompensieren. In nachfolgenden Arbeiten sind nun die Robustheit der vorgestellten und anderer Verfahren bezüglich Modellungenauigkeiten und Störungen zu untersuchen und die Regelalgorithmen auf den realen Prozess zu übertragen.

Literatur

- [1] B. Siciliano and L. Sciavicco and L. Villani and G. Oriolo, *Robotics, Modelling, Planning and Control* Springer, 2009.
- [2] R. J. Schilling, *Fundamentals of robotics, analysis and control* Prentice Hall, 1990.
- [3] J. Craig, *Introduction to robotics, mechanics and control* Pearson Education, 2005.
- [4] R. Popp, *Entwicklung einer Trajektorienfolge-
regelung für einen mobilen 5DOF Service Robo-
ter*, 3rd ed. Leibniz Universität Hannover, 2014.

[5] F. Schmidt, *Analyse und Entwurf iterativ ler-
nender Regelungen* University Kaiserslautern,
1996.

[6] D. A. Bristow and M. Tharayil and A. G. Alley-
ne, *A survey of iterative learning control* IEE-
EE Control systems magazine, pp. 96-114, June
2006.

[7] H. Ahn and Y. Chen and K. L. Moore, *iterative
Learning Control: Brief Survey and Categori-
zation* IEEE Transactions on systems, man, and
cybernetics-Part C: Applications and Reviews,
Vol. 37, No. 6, November 2007.

[8] K. L. Moore and Y. Chen and H.-S. Ahn, *Itera-
tive learning control: A Tutorial and big picture
view* Decision and Control, 45th IEEE Confe-
rence, 2006.

[9] H. Nahrstaedt and T. Schauer and S. Hesse
and J. Raisch, *Iterativ Lernende Regelung einer
Gang-Neuroprothese* at - Automatisierungs-
technik 56 (2008), S. 494-501

[10] M. G. E. Schneiders and M. J. G. van de Molen-
graft and M. Steinbuch, *Evaluation of (unstable)
non-causal system applied to iterative learning
control*

A Jacobimatrizen

$$\mathbf{J}_P = \begin{bmatrix} \mathbf{J}_P^1 \\ \mathbf{J}_P^2 \\ \mathbf{J}_P^3 \\ \mathbf{J}_P^4 \\ \mathbf{J}_P^5 \end{bmatrix}, \quad \mathbf{J}_O = \begin{bmatrix} \mathbf{J}_O^1 \\ \mathbf{J}_O^2 \\ \mathbf{J}_O^3 \\ \mathbf{J}_O^4 \\ \mathbf{J}_O^5 \end{bmatrix} = \begin{bmatrix} \mathbf{z} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{z} & \mathbf{z} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{z} & \mathbf{z} & \mathbf{z} & \mathbf{0} & \mathbf{0} \\ \mathbf{z} & \mathbf{0} & \mathbf{0} & \mathbf{z} & \mathbf{0} \\ \mathbf{z} & \mathbf{0} & \mathbf{0} & \mathbf{z} & \mathbf{z} \end{bmatrix}$$

$$\mathbf{J}_P = \begin{bmatrix} -l_1 s_1 & 0 & 0 & 0 & 0 \\ l_1 c_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -a_1 s_1 - l_2 s_{12} & -l_2 s_{12} & 0 & 0 & 0 \\ a_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \sigma_1 - a_1 s_1 & \sigma_1 & -l_3 s_{123} & 0 & 0 \\ \sigma_2 + a_1 c_1 & \sigma_2 & l_3 c_{123} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -a_1 s_1 - l_4 s_{14} & 0 & 0 & -l_4 s_{14} & 0 \\ a_1 c_1 + l_4 c_{14} & 0 & 0 & l_4 c_{14} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \sigma_3 - a_1 s_1 & 0 & 0 & \sigma_3 & -l_5 c_{145} \\ \sigma_4 + a_1 c_1 & 0 & 0 & \sigma_4 & l_5 c_{145} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} \sigma_1 &= -l_3 s_{123} - a_2 s_{12} & \sigma_2 &= l_3 c_{123} + a_2 c_{12} \\ \sigma_3 &= -l_5 c_{145} - a_4 s_{14} & \sigma_4 &= l_5 c_{145} + a_4 c_{14} \end{aligned}$$

Notation: $c_i = \cos \theta_i$, $s_i = \sin \theta_i$, $s_{ij} = \sin(\theta_i + \theta_j)$,
 $s_{ijk} = \sin(\theta_i + \theta_j + \theta_k)$ und $\mathbf{z} = [0 \ 0 \ 1]^T$

Sitzung 3

Raum 116

Maritime Anwendungen

(Sitzungsleitung: Franz Jendersie)

Robuste Regelung zur Dynamischen Positionierung von Schiffen mit Voith Schneider

Propellern

Charlotte Siebert

Universität Rostock

Aufbereitung von AIS-Daten der WSV-N für die Simulation von Seeverkehrssystemen

Carsten Hilgenfeld

Hochschule Wismar

Einsatz von Strömungssimulationen bei der Optimierung der Aerodynamik moderner

Kreuzfahrtschiffen

Holm Stückrad

MET GmbH

Robuste Regelung zur Dynamische Positionierung von Schiffen mit Voith Schneider Propellern

Charlotte Siebert¹, Adel Haghani¹, Philipp Koschorrek¹, Torsten Jeinsch¹

¹Universität Rostock, Institut für Automatisierungstechnik

{charlotte.siebert, adel.haghani, philipp.koschorrek, torsten.jeinsch}@uni-rostock.de

Schiffe sind auf See unterschiedlichen Störungen wie Wind, Wellen und Strömung ausgesetzt. Um dennoch auch in tiefen Gewässern, in denen beispielsweise Ankern nicht möglich ist, ihre absolute oder relative Position zu halten, werden Systeme zur Dynamischen Positionierung (DP) eingesetzt. Dabei wird allein durch die Aktuatoren erreicht, dass die aktuelle Position und Orientierung festgelegte Grenzen nicht überschreitet. In diesem Beitrag kommen Voith Schneider Propeller (VSP) zum Einsatz, welche durch sehr gute dynamische Eigenschaften ein präzises und effizientes Manövrieren ermöglichen. Ein weiterer Hauptbestandteil solcher DP-Systeme sind intelligente Regelalgorithmen, welche nach den klassischen Methoden als PID-Regler mit dem LQ- oder dem Polvorgabeverfahren entworfen werden können. Hierfür muss ein Modell der Regelstrecke vorliegen, welches häufig in einem Arbeitspunkt linearisiert wird, sollte es sich um eine nichtlineare oder sehr komplexe Strecke handeln. Werden die Abweichung zum Arbeitspunkt aufgrund unvorhergesehener großer Störung oder Parameterveränderungen zu groß, so kann der Regler die gewünschte Kompensation nicht mehr erfüllen. Dann ist es sinnvoll einen Regler einzusetzen, der ein robustes Verhalten der Strecke gegenüber solchen Abweichungen um den Arbeitspunkt erlaubt, einen robusten Regler. Das Entwurfsverfahren zielt außerdem auf eine genauere Differenzierung der Störungen ab, sodass hochfrequenten Anteile vernachlässigt werden können, um den Stellaufwand möglichst gering zu halten. Im Rahmen dieses Beitrags wird zunächst die Struktur und Funktionsweise eines robusten Reglers erläutert. Anschließend wird der Entwurf für ein lineares Modell der Schiffsdynamik in drei Freiheitsgrade entworfen und Simulationsergebnisse vorgestellt.

1 Einführung

Schiffe oder Plattformen sind auf See dem dort herrschendem Wetter und somit dem Wind, der Strömung und dem Wellengang ausgesetzt. Soll nun eine gewünschte absolute oder relative Position gehalten werden, kommen Systeme zur Dynamischen Positionierung (DP-Systeme) zum Einsatz. Der Begriff Dynamische Positionierung meint die Beibehaltung dieser gewünschten Position allein durch Einsatz der Antriebe. In diesem Beitrag sind das spezielle Voith Schneider Propeller (VSP), die sich durch ihre hohe dynamische Flexibilität auszeichnen und dadurch bestens für DP-Systeme geeignet sind. In Abbildung 1 ist ein VSP dargestellt. Vor über 85 Jahren entwickelte der Ingenieur Ernst Schneider diesen einzigartigen Schiffsantrieb. Er kann gleichzeitig als Antrieb und als Steue-



Abbildung 1: Voith Schneider Propeller [1]

rung dient. Dabei kann der Schub in Stärke und Richtung beliebig eingestellt werden. Der VSP besteht aus einer horizontal rotierenden Scheibe, an welcher senkrecht mehrere Flügelblätter angebracht sind. Über die

Drehzahl der Scheibe lässt sich der Schubbetrag einstellen. Die Verstellung der Flügel bestimmt die Richtung des Schubs.

DP-Systeme wurden in den 1960er Jahren im Zuge der Offshore-Ölindustrie notwendig, weil in immer tieferen Gewässern nach Öl gebohrt werden sollte. Heutzutage sind es nicht nur Bohrschiffe, die mit DP-Technik ausgestattet sind, sondern auch Schiffe zum Verlegen und Warten von Unterwasserkabeln und Pipelines, Minenjäger, Kreuzfahrtschiffe oder Wartungsschiffe für Offshore-Anlagen.

DP-Systeme bestehen im Allgemeinen aus einem DP-Regler, einer Allokation und einem Sensorsystem. All das sorgt dafür, dass die Antriebe den für die Kompensation der Störung benötigten Kraftvektor erzeugt.

Anfänglich wurde mit PID-Reglern dynamisch positioniert. Hierfür stehen unterschiedlichen Entwurfsverfahren zur Verfügung, wie Polvorgabe oder das LQ-Verfahren [2], [3]. In [4] wird ein LQ-Regler mit Kalman Filter vorgestellt, mit dem auch ungenaue Antriebsmodelle und Wellenstörungen zweiter Ordnung und Strömung ausgleichen werden können. Dieser hat den Nachteil, dass Änderungen in der Wellenfrequenz und Modellungenauigkeiten nicht ausgeregelt werden [5]. Um mit diesem Problem umzugehen, besteht die Möglichkeit die Systeme auf manuellen Betrieb umzuschalten, sobald der Arbeitspunkt des Reglers verlassen wird. Daraufhin ist ein adaptiver Ansatz verfolgt worden [6], mit dem diese Umschaltung umgangen werden sollte. Doch aufgrund der Komplexität und starken Nichtlinearität des entworfenen Reglers wurden diese nicht praktisch eingesetzt. Für den Fall, dass die Abweichungen vom Arbeitspunkt zu groß werden, ist ein Regler erforderlich, der ein robustes Verhalten der Strecke gegenüber solchen Abweichungen erlaubt. Hierfür wurde von [7] ein robuster Regler eingeführt. Neben der größeren erlaubten Abweichung vom Arbeitspunkt kann mit Hilfe dieses Reglers auch eine genauere Differenzierung der Störungen erlaubt werden, sodass hochfrequente Anteile vernachlässigt werden können.

2 Robuste Regelung

Ein robuster Regler arbeitet zuverlässig in einem definierten Bereich um einen Arbeitspunkt und erzeugt

somit ein robustes Verhalten der Strecke gegenüber Störungen. Das bedeutet im Umkehrschluss jedoch auch, dass er nicht für einen Punkt, sondern für einen Bereich optimal ausgelegt ist. Ein PID-Regler beispielsweise könnte in einem bestimmten Arbeitspunkt diese Bereiche weniger Überschwingen oder eine geringere Einschwingzeit erreichen. Der große Vorteil, den ein robuster Regler in diesem Paper gegenüber einem PID-Regler hat, ist die deutliche Verringerung der Stellenergie bei gleichen Gegebenheiten.

2.1 Verwendete Modelle

Die Schiffsbewegung wird durch die nichtlinearen Bewegungsgleichungen in sechs Freiheitsgrade aus [2] modelliert. Hierbei sind die Position und Orientierung des Schiffes in erdfesten Koordinaten im Vektor $\eta = [x, y, z, \phi, \theta, \psi]$ zusammengefasst. Für den DP-Fall wird die Bewegung entlang der z -Achse (tauchen), die Drehungen um die Längsachse (rollen) und die Drehung um die Querachse (stampfen) vernachlässigt, d.h. $z = \phi = \theta = 0$. Der resultierende Vektor der Position und Orientierung ist dann $\eta = [x, y, \psi]$. Der Vektor der dazugehörigen Geschwindigkeiten $v = [u, v, r]$ ist im körperfesten Koordinatensystem definiert. Abbildung 2 zeigt die verwendeten Koordinatensysteme.

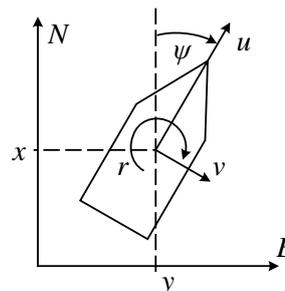


Abbildung 2: Koordinatensystem

Das mathematische Modell ist für den DP-Fall im Arbeitspunkt $u = v = r \approx 0$ linearisiert und gilt für kleine Kursänderungen, d.h. $\psi \approx 0$.

$$\dot{\eta} = R(\psi)v \quad (1)$$

$$M\dot{v} + C(v)v + D(v)v = \tau + \tau_{wind} + \tau_{wave} \quad (2)$$

R ist die Rotationsmatrix zwischen dem erdfesten und dem körperfesten Koordinatensystem. Hierbei wird

das erdfeste System schiffsparell gelegt, sodass für die Transformation lediglich der Kurswinkel ψ benötigt wird. Die Massenmatrix M berücksichtigt neben dem Gewicht auch die Geometrie und die Trägheitsmomente des Schiffes und beinhaltet die hydrodynamischen Massen. D ist die Dämpfungsmatrix, C die Matrix der Coriolis- und Zentripetalkräfte des Schiffes unter Beachtung der hydrodynamischen Massen. Der Vektor τ bezeichnet den Kraftvektor. Die externen Störungen wie Wind, Wellen und Strömung werden additiv der Strecke hinzugefügt und in niederfrequent und hochfrequente Anteile unterschieden. Der niederfrequenten Anteil der Störung R_l und der hochfrequente Anteil R_h werden mit den nachfolgenden Modell aus [5] abgebildet

$$R_l(s) = \frac{a_{1i}s + a_{2i}}{b_i s + 1} \quad (3)$$

$$R_h(s) = \frac{a_i s^2 + 2\zeta_{1i}\omega_{hi}s + \omega_i^2}{b_{hi}(s^2 + 2\zeta_{2i}\omega_{hi}s + \omega_{hi}^2)} \quad (4)$$

Der Index $i = 1, 2, 3$ steht für die einzelnen Bewegungen schnellen, versetzen und gieren. Interne Störungen werden in Form von Modellunsicherheiten berücksichtigt. Dabei kann es sich beispielsweise um eine Änderung der Masse oder Massenverteilung aufgrund von Be- und Entladung oder von Treibstoffverbrauch handeln. Modellunsicherheiten werden häufig multiplikative verknüpft [8]. Die Übertragungsfunktion dazu ist folgende

$$G_p(s) = (I + W_o(s)\Delta_o(s))G_m(s). \quad (5)$$

Wie diese drei Modelle in die Strecke einfließen ist in Abbildung 3 zu erkennen.

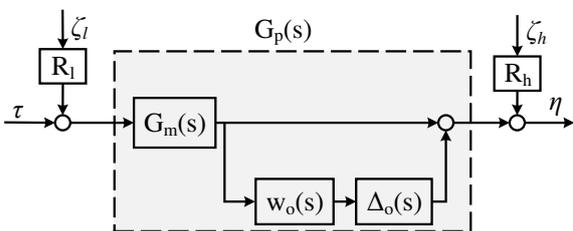


Abbildung 3: vollständiges Streckenmodell $G(s)$

Das vollständiges Streckenmodell $G(s)$ besteht aus dem Schiff inklusive Antriebe $G_m(s)$, den Modellunsicherheiten Δ_o (zusammen $G_p(s)$; p für "perturbed": gestört) und den externen Störmodellen R_l und R_h für

niederfrequente (low) und hochfrequente (high) Störungen. Die Eingänge der Strecke sind dementsprechend die Kraft τ , die der Regler zur Verfügung stellt, und die Störungen ζ_l und ζ_h . Der Ausgang der Strecke ist η .

2.2 Struktur der Regelung

Für einen robusten Regler wird die gegebene Strecke erweitert und für diese neue Ein- und Ausgänge definiert. Die für das vorgestellte Modell geeignete Regelstruktur ist eine Kaskade, da sich hier die Dynamik und die Statik des Regelkreises gut voneinander getrennt optimieren lassen. Dabei wird zunächst der innere Regler, der robust ausgelegte Geschwindigkeitsregler, so getuned, dass die Dynamik des Regelkreises das gewünschte robuste Verhalten aufweist. In diesem Fall wird der robuste Regler so eingestellt, dass hochfrequente Störungen keine Änderung der auf das Schiff wirkenden Kraft nach sich ziehen. Das bedeutet, dass keine Stellenergie dafür aufgebracht werden soll geringe Änderungen der Geschwindigkeit auszugleichen, sondern nur für große Differenzen, die durch niederfrequente Störung hervorgerufen werden. Ist dieser Regelkreis optimiert, wird der äußere Kreis betrachtet um die Beziehung zwischen der Referenzposition η_r und der dafür notwendigen Geschwindigkeit v_r zu optimieren. Abbildung 4 stellt die Verkopplung dar. Hier beschreiben e_η

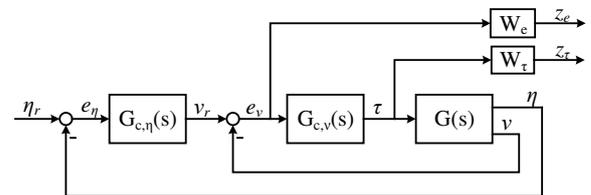


Abbildung 4: Regelstruktur als Kaskade

und e_v den Positions- bzw. Geschwindigkeitsfehler, $G_{c,\eta}(s)$ und $G_{c,v}(s)$ sind die Übertragungsfunktionen des Positionsreglers und des Geschwindigkeitsreglers. W_e und W_τ sind die Gewichtungsmatrizen des Fehlers und der Stellgröße.

2.3 Parametrierung des robusten Reglers

Für den Reglerentwurf wird die Strecke um die Gewichtungsfunktionen der neuen Ein- und Ausgänge erweitert. Die Struktur der erweiterten Strecke P ist in Abbildung 5 dargestellt. Daraus ergibt sich die Über-

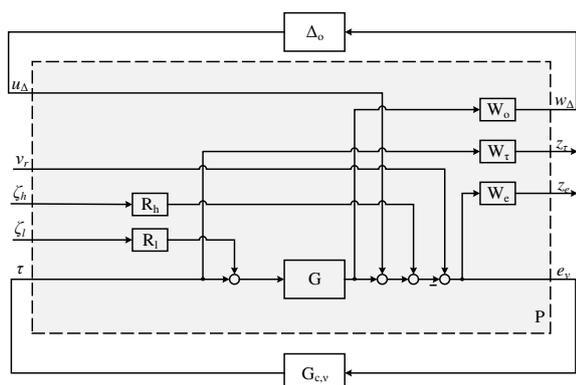


Abbildung 5: Standardsystem mit Regler $G_{c,v}$, erweiterter Strecke P und Modellunsicherheiten Δ_o

tragungsmatrix zwischen den neuen Ein- und Ausgängen

$$\begin{bmatrix} z_e \\ z_{\tau} \\ w_{\Delta} \\ e_v \end{bmatrix} = P \begin{bmatrix} v_r \\ \zeta_l \\ \zeta_h \\ u_{\Delta} \\ \tau \end{bmatrix} \quad (6)$$

mit

$$P = \begin{bmatrix} W_e & -W_e G_s R_l & -W_e R_h & -W_e & -W_e G_s \\ 0 & 0 & 0 & 0 & W_{\tau} \\ 0 & W_o G_s R_l & 0 & 0 & W_o G_s \\ I & -G_s R_l & -R_h & -I & -G_s \end{bmatrix} \quad (7)$$

Der Übersichtlichkeit halber bietet sich für die Reglersynthese die kompakte Darstellung der Strecke aus Abbildung 6 an. Die externen Störungen sind in $\zeta^T = [\zeta_l^T, \zeta_h^T]$ und die gewichteten Ausgänge in $z^T = [z_e^T, z_{\tau}^T]$ zusammengefasst.

Um den Regler einzustellen, müssen die Matrizen W_e zur Gewichtung des Regelfehlers, W_{τ} zur Gewichtung der Stellgrößen und W_o zur Gewichtung der Modellunsicherheiten geeignet gewählt werden. In [5] wird

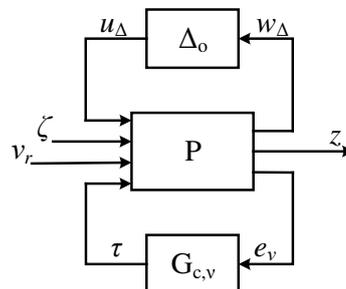


Abbildung 6: erweiterte Strecke zur Reglersynthese

W_{τ} als Hochpassfilter gewählt, um die Stellenergie bei hohen Frequenzen stärker zu bestrafen. Gleichzeitig sind W_e und W_{τ} so ins Verhältnis zu stellen, dass der Frequenzgang von W_{τ} den von W_e etwa eine Dekade nach dessen Erreichen der Hochfrequenzverstärkung schneidet [5]. Abbildung 7 macht das deutlich.

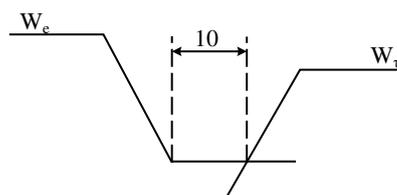


Abbildung 7: Verhältnis zwischen W_{τ} und W_e

Für W_o ist die Einheitsmatrix gewählt worden, so dass die Änderung der Massematrix um beispielsweise $\pm 10\%$ ohne zusätzliche Gewichtung auf die Strecke wirkt.

3 Ergebnisse und Diskussion

Hier sind die Ergebnisse der Simulation eines Versorgungsschiffes mit einer Länge von 82,8m und einem Gewicht von rund $6,362 \cdot 10^6$ kg dargestellt. Dabei werden keine Spezifikationen der Antriebssysteme getroffen, d.h. wie die durch den Regler berechnete Kraft auf das Schiff wirkt ist nicht relevant. Das bedeutet auch, dass die Antriebsdynamiken nicht berücksichtigt werden. Da VSP jedoch sehr gute dynamische Eigenschaften aufweisen sind keine großen Abweichungen der Ergebnisse bei Beachtung der Antriebe zu erwarten. Aufgrund der geringen Verstellzeiten und der Möglichkeit der schnellen Schubumkehr sind die

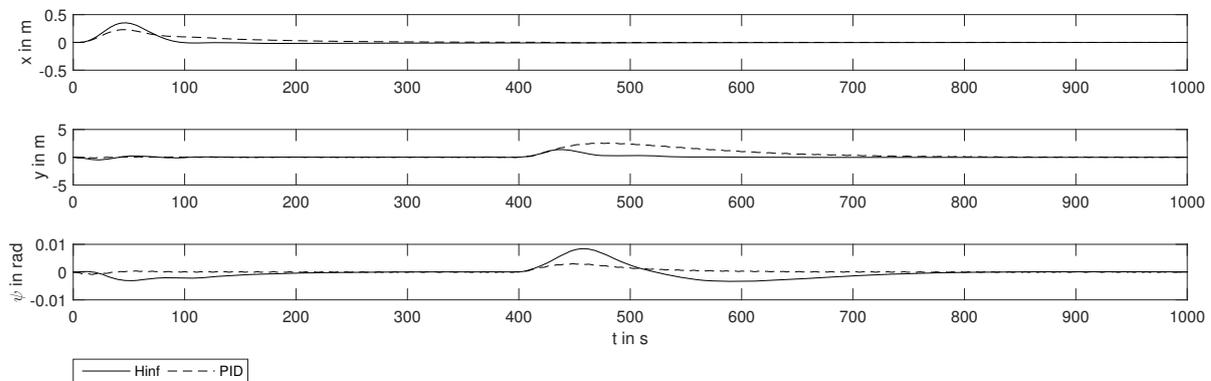


Abbildung 8: Position des Schiffes

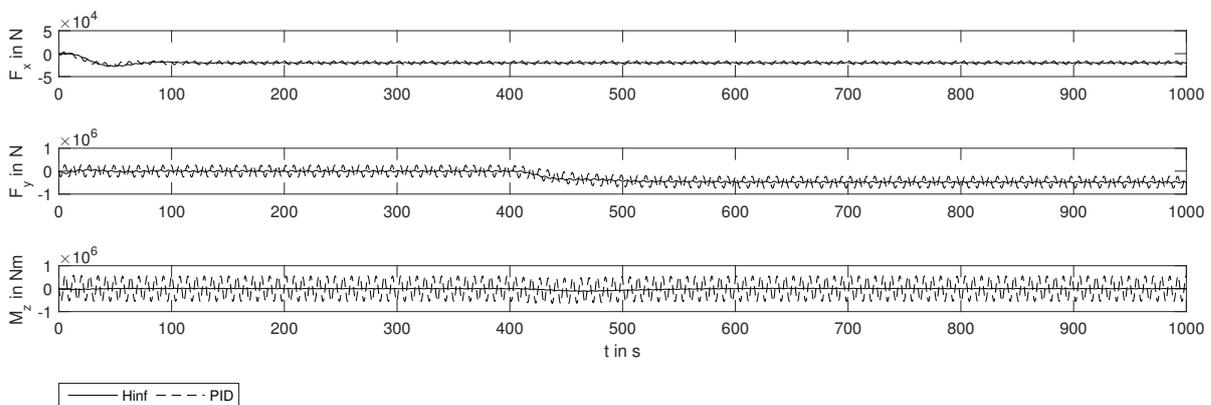


Abbildung 9: Stellgrößenverlauf für DP

durch diese spezielle Antriebe hervorgerufenen unerwünschten Querkräfte zunächst vernachlässigt worden. Das Schiff wurde in den Simulationen mit folgenden Störungen beaufschlagt:

- hoch- und niederfrequente Anteile von Wind und Wellen wirken von Beginn an in einem Winkel von 180° auf das Fahrzeug,
- niederfrequente Störungen setzt als Strömung mit einer Geschwindigkeit von $0,5 \frac{m}{s}$ aus 270° (Backbord) nach 400 Sekunden ein.

Da es sich hier um eine DP-Anwendung handelt, ist der zu haltende Sollwerte gegeben als $\eta_r = [0, 0, 0]^T$. In Abbildung 8 ist der Vergleich der erreichten Position mit einem PID-Regler und mit einem H_∞ -Regler dargestellt. Es ist zu erkennen, dass sowohl der robuste Regler als auch der PID-Regler beim Auftreten einer Störung das Schiff auf seiner Position halten.

Die Größe des Überschwingens und der Einschwingzeit unterscheiden sich dabei. Der entscheidende Unterschied zwischen beiden Reglern kommt jedoch erst in Abbildung 9 zum Vorschein. Hier ist zu erkennen, dass der PID-Regler für die Beibehaltung der Position große Stellgrößenänderungen fordert. Im Gegensatz dazu bleibt der Stellgrößenaufwand, der vom H_∞ -Regler gefordert wird, gering.

Was auf Grundlage dieser Simulationsergebnisse weiterhin untersucht werden sollte, ist die Berücksichtigung der Antriebe. Daran anschließend kann eine robuste Regelung für den Ausfall einer oder mehrerer Sensor- oder Aktorkomponente entworfen und simulativ getestet werden. Hierbei spielen die Fehlererkennung, Fehlerdiagnose und die Fehlerkompensation oder -korrektur eine wichtige Rolle.

Literatur

- [1] Voith, *Pressemitteilung*. http://voith.com/en/press/press-releases-99_27640.html, besucht: 20.04.2015.
- [2] Fossen, T.I., *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.
- [3] Juergens, D., Palm, M., Brandner, A., *Comparative investigation on influence of the positioning time of azimuth thrusters on the accuracy of DP*. Dynamic Positioning Conference, Houston, 2012.
- [4] Balchen, J.G., Jenssen, N.A., Mathisen, E., Saelid, S., *A dynamic positioning system based on Kalman filtering and optimal control*. MIC-Journal, Vol 1, No 3, 135-163, 1980.
- [5] Katebi, M.R., Grimble, M.J., Zhang, Y., *H_∞ robust control design for dynamic ship positioning*. Control Theory and Applications, IEE Proceedings, Vol 144, No 2, 110-120, 1997.
- [6] Fung, P.T.K., Grimble, M.J., *Dynamic ship positioning using a self-tuning Kalman filter*. Automatic Control, IEEE Transactions on, Vol 28, No 3, 339,350, 1983.
- [7] Zames, G., *Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses*. Automatic Control, IEEE Transactions on, Vol 26, No 2, 301-320, 1981.
- [8] Skogestad, S., Postlethwaite, I., *Multivariable Feedback Control, Analysis and design*. John Wiley & Sons, 2005.

Aufbereitung von AIS-Daten der WSV-N für die Simulation von Seeverkehrssystemen

Prof. Dr.-Ing. N. Vojdani¹, Prof. Dr.-Ing. M. Ahn², Dipl.-Ing. (FH) C. Hilgenfeld M.Sc.³

¹ Universität Rostock Lehrstuhl für Produktionsorganisation und Logistik

² Hochschule Wismar Professur für Verkehrsplanung und Stadtplanung

³ Hochschule Wismar wiss. Mitarbeiter Bereich Seefahrt

carsten.hilgenfeld@hs-wismar.de

After receiving data of the automatic identification system (AIS) from the Federal Waterways and Shipping Authority it is recommended to prepare the data. Unnecessary duplications increase the storage space and can lead instability issues in the simulation. As a first step the relative GPS antenna location is used to correct the ship's position to the actual center of the ship. Then a critical examination of the ship's speed and ship's course is advisable, because a faulty database leads to a significant contradiction of the datasets. The final step for the data preparation is the distinction, if a vessel is part of the vessel flow in the investigated area or not.

1 Einordnung

Für eine Simulation zur Verhaltensprognose von Seeverkehrssystemen ist die Bereitstellung von systemrelevanten Informationen unerlässlich. Dazu wurden im Rahmen eines Forschungsprojektes zwischen der Universität Rostock und der Hochschule Wismar reale Daten, des in der Schifffahrt verbreiteten Automatic Identification Systems (AIS), aufbereitet, um damit eine realitätsnahe Modellierung des Verkehrssystems vorzunehmen und diese realen Daten zur besseren Überprüfbarkeit der Ergebnisse der Simulation herangezogen.

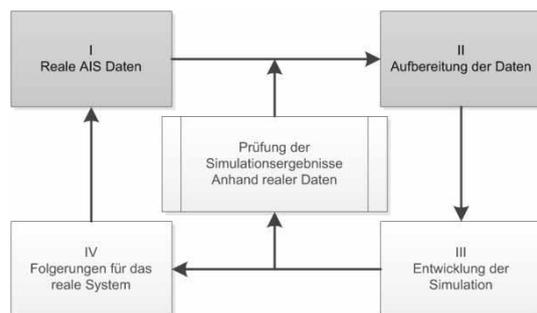


Abbildung 1. Einordnung in die Simulationentwicklung

Nachfolgend wird erläutert auf welche Weise die AIS-Daten der Schifffahrtsverwaltung des Bundes, Außenstelle Nord (WSV-N) editiert sind (vgl. Abbildung 1 - I) und wie die Aufbereitung dieser Daten erfolgte (II), um diese in eine Simulationsumgebung einzubetten und das Simulationsergebnis zu validieren. Das Anwendungsgebiet für die Simulation ist die Kadettrinne (KDR) in der Ostsee, dessen Verkehrssystem nachgebildet und analysiert werden soll.

2 Das Automatic Identification System (AIS) im Projektkontext

Beim AIS handelt es sich um ein broadcast Funkverfahren gemäß Recommendation ITU-R M.1371-5, in dem kodierte Informationen übertragen werden [IMEA 2013]. Ziel des am 06.12.2000 durch die International Maritime Organisation (IMO) bestätigten Standards und der in der SOLAS verankerten Regelung für AIS ist eine landseitige Verkehrslenkung durch den jeweiligen Küstenstaat - mit detaillierten Information über die Schiffe bzw. Ladung - und damit primär die Kollisionsverhütung der Schiffe zur Erhöhung der Sicherheit auf See.

Das AIS unterscheidet in Klasse A und B Transceiver Typen. Während Klasse A von allen ausrüstungspflichtigen Schiffen geführt wird, sind Klasse B AIS Anlagen auf Schiffen eingesetzt, deren Ausrüstung freiwillig geschieht (Privatschifffahrt).

Das AIS unterscheidet in 27 verschiedene Nachrichtentypen [Raymond 2015]. Für eine ganzheitliche Betrachtung ist das Zusammenführen der relevanten sechs Nachrichten erforderlich. Dies ist erforderlich, da in Nachricht 1, 2, 3 und 18 (Class B) u.a. die Position und der Kurs des Schiffes alle 3-10 Sekunden versendet wird, während in Typ 5 und 24 (Class B) statische Informationen wie Schiffsname, Länge und Breite sowie Position des GPS-Empfängers an Bord nur alle 6 min übertragen werden. Diese sechs Nachrichten sind alles Nachrichten, die Informationen über die Bewegung oder den Zustand des Schiffes beinhalten und sind in Tabelle 1 aufgelistet.

Nachrichtentyp	Bedeutung
Msg01	Position Report Class A
Msg02	Position Report Class A (Assigned schedule)
Msg03	Position Report Class A (Response to interrogation)
Msg05	Static and Voyage Related Data Class A
Msg18	Standard Class B CS Position Report
Msg24	Static Data Report Class B

Tabelle 1. Verwendete AIS-Nachrichtentypen

Die verwendeten AIS-Daten aus dem Bereich der KDR wurden bei der Wasser- und Schifffahrtsverwaltung des Bundes, Außenstelle Nord (WSV-N) beantragt, die als hoheitlich verwaltende Stelle seit 2011 die Daten von offizieller Seite zur Verfügung stellt [Hilgenfeld et al. 2014]. Diese offiziellen Daten sind abweichend von industriellen AIS-Datensätzen auf eine spezielle Weise editiert. So wurde bspw. ein eindeutiger Zeitstempel hinzugefügt, was das AIS sonst nur im beschränkten Maße enthält. Diese Stempel zeigen den Zeitpunkt des Empfangs der Nachricht und den Zeitpunkt der Verarbeitung.

Nachrichteninhalt	Bedeutung
00000090,00000000,...	DB Verweise
...110601000002814,...	Zeitstempel I
...110601000002791,...	Zeitstempel II
...!ABVDM,1,1,3,A,110GQ00011PW1 AtNbEM5o4n40D3E,0*78	AIS Nachricht

Tabelle 2. Erläuterung des AIS-Datensatzes

Weiterhin wird bsp. die offizielle neunstellige Maritime Mobile Service Identity (MMSI) durch die WSV-N mit einer im internen Sprachgebrauch titulierten WSV-ID ersetzt. Diese WSV-ID ist eine von 100.000.000 beginnende fortlaufende ID, um damit den besonderen Anforderungen des deutschen Datenschutzes Rechnung zu tragen. Ein Fahrzeug derselben MMSI erhält bei dieser sogenannten Pseudonymisierung innerhalb eines Datenabrufs nun eine konstante WSV-ID. Es ist jedoch zu beachten, dass bei jedem neuen Datenabruf bei der WSV-N diese Pseudonymisierung mit der beschriebenen Vorgehensweise erneut bei 100.000.000 beginnt und somit keine Verknüp-

fungen über die WSV-ID zwischen unterschiedlichen Datenabrufen möglich ist [Vojdani et al. 2013].

3 Umwandlung der WSV-N AIS-Datenbank

Nach Umwandlung des kodierten AIS-Datensatzes in ein lesbare Format, müssen die verschiedenen AIS-Nachrichtentypen logisch sortiert und die Nachrichtentypen untereinander ergänzt werden [ITU 2010]. Dazu wurden im Vorfeld der Umwandlung 19 AIS-Informationen definiert, die für die Entwicklung der Simulationsumgebung und der Programmierung notwendig waren. Dazu zählen u.a. die WSV-ID, der Zeitstempel, die Positionsangaben, der Kurs und die Geschwindigkeit sowie der Tiefgang oder der Ausrüstungstyp.

Maßgeblich für das Schreiben der dekodierten Datensätze in eine zusammenfassende AIS-Projektdatenbank sind die Msg01, 02, 03 und 18 (vgl. Tabelle 1). Die Daten wurden zeilenweise ausgelesen und nach den maßgebenden dynamischen Daten entschieden, ob die Informationen in eine temporäre Datenbank für statische Informationen oder die Projektdatenbank geschrieben wurde. Wenn es sich um eine Msg01, 02, 03 und 18 handelt, werden die dynamischen Daten mit den bisher vorhandenen statischen Informationen ergänzt und zeilenweise mit allen 19 Informationen abgelegt. Handelte es sich dagegen um eine Msg05 oder 24, wurden die statischen Daten in der temporären Datenbank für die jeweilige WSV-ID aktualisiert.

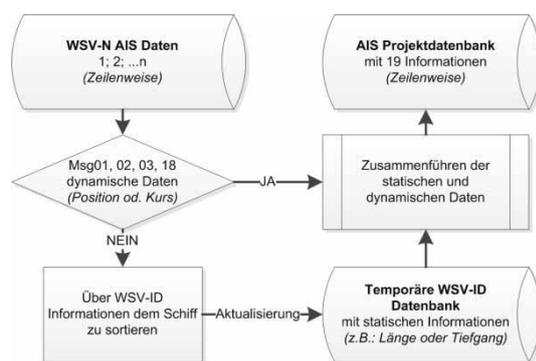


Abbildung 2. Zusammenführung der statischen und dynamischen AIS-Daten

Schon bei Projektbeginn war festzustellen, dass außer des WSV-N Zeitstempels und der Positionsangabe alle anderen Informationen äußerst kritisch zu hinterfragen sind [Baldauf et al. 2006; Heymann 2013]. Der Zeitstempel ist vertrauenswürdig, da dieser durch die WSV-N in einem landgestützten und durch interne

Prozesse abgesicherten Verfahren generiert wurde. Eine Analyse des Zeitstempels ergab, dass alle Datensätze wie zu erwarten in chronologischer Reihenfolge abgelegt wurden und der Empfangszeitpunkt (Zeitstempel II) vor dem Verarbeitungszeitpunkt lag (Zeitstempel I). Die Positionsangabe ist dahingehend ausreichend vertrauenswürdig, da einerseits im AIS selbst hierarchisch die geeignetste an Bord befindliche Positionsmesseinrichtung für die Daten verwendet wird [Hoppe 2006; Norris 2008] und andererseits falls die Positionsangabe tatsächlich, z.B. fehlerhaft empfangen werden würde, durch die Auswahlkriterien bei der WSV-Extraktion herausgefiltert wurde. Nehmen wir beispielsweise ein Schiff an, welches sich tatsächlich in der KDR befindet. Dieses Schiff sendet aber eine fehlerhafte Positionsmeldung ab, wonach es sich im englischen Kanal befindet. Diese Positionsangabe wurde schon bei der Extraktion aus der WSV-N Datenbank herausgefiltert, da das Auswahlkriterium die Position in der Kadettrinne ist.

Für die Verwendung der Daten in der Simulationsumgebung ist es zweckmäßig das Format der Positionsangabe umzuwandeln. Die Positionsangabe liegen originär nach dem geodätischen Referenzsystem WGS84 im Sexagesimalsystem in geographischer Länge und Breite vor. Nach einer Umwandlung über eine transversale Mercatorprojektion in ein winkeltreues kartesisches Koordinatensystem ergibt sich die Möglichkeit, die relativen und absoluten Entfernung als Universal Transverse Mercator (UTM) direkt über Meterangaben zu ermitteln.

Es wurde sich beim UTM für das Gauß-Krüger Format mit dem Streifen 12° östliche Breite entschieden und die Daten dahingehend umgewandelt [Böhme et al. 1984]. Das Gauß-Krüger Format unterscheidet in Rechtswert gemessen von 0° Breite und Hochwert gemessen vom Äquator in Metern.

4 Filtern von überflüssigen Datensätzen

Durch die hohe Netzabdeckung der von der WSV-N (vgl. Abbildung 3) betriebenen AIS-Basisstationen kann aufgrund der Tragweite des AIS-Signals über 20 sm dasselbe Signal zweimal aufgefangen, verarbeitet und abgespeichert werden. Dabei unterscheidet sich die vorhandene Information nur anhand des unterschiedlichen Empfangs- und Verarbeitungszeitpunktes.

Dem gegenüber können aber zwei aufeinanderfolgende Datensätze desselben Schiffes nahezu identische Informationen beinhalten, ohne dass es sich um die-

selbe AIS-Nachricht handelt. Einziges zuverlässiges Unterscheidungskriterium (bei den ausgewählten 19 Parametern) ist die Position. Aus diesem Grund wurde für die Datensätze, die derselben WSV-ID zugeordnet worden, überprüft, ob es bei einander folgenden Datensätzen eine Veränderung des Hoch- oder Rechtswertes gab. Wurde beides negiert erfolgte die Löschung des wiederholt empfangenen Datensatzes.

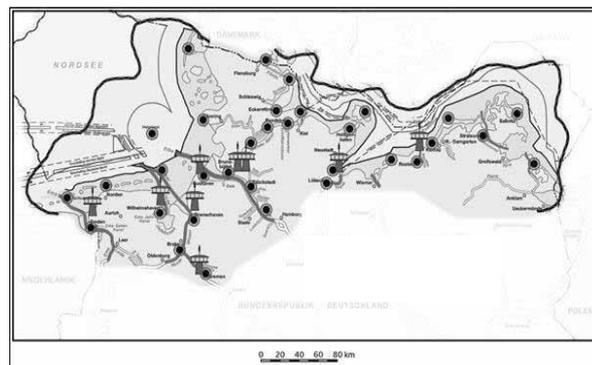


Abbildung 3. AIS-Abdeckungsbereich der WSV
[Quelle: WSV]

Die Entfernung von überflüssigen Datensätzen hat neben der Reduzierung des benötigten Speicherbedarfes auch erheblichen Vorteil in Bezug auf die Stabilität der verwendeten Skripte, da so bei der späteren Korrektur von Kurs und Geschwindigkeit über eine relative Positionsveränderung ein Divisor von null vermieden wird.

5 Ermittlung von bewegenden Signalen

Es gibt eine Vielzahl von AIS-Sendern, die zwar ein AIS Signal senden, aber keine Bewegung durchführen. Dazu gehören stationäre Einrichtungen wie z.B. auch der Bereich Seefahrt der Hochschule Wismar, welcher für Ausbildungszwecke ebenfalls AIS Signale über seine Einrichtungen versendet. AIS-Transceiver sind weiterhin frei verkäuflich und können von jeder Privatperson erworben werden.

Daher ist es insbesondere zur Speicherplatzreduzierung angebracht alle stationären Signale zu entfernen. Dies erfolgte über die Prüfung, ob das Signal eine Bewegung im Betrachtungszeitraum durchgeführt hatte. Das Ausschlusskriterium sollte ausreichend groß gewählt werden um damit eine falsch-positiv Meldung Aufgrund der GPS-Ungenauigkeit auszuschließen. Es wurde ein Radius von 208 m als 16-facher Standardfehler der 13 m gewählt (95% aller GPS Signale liegen in einem Radius von 13 m) [Mou et al. 2010]. Sollte das Signal die Fläche von 0,136

km² nicht verlassen haben, wird es als stationäres Signal identifiziert und alle verfügbaren Daten der WSV-ID gelöscht.

6 Korrektur der Position des Schiffes

Ein inhärenter Bestandteil der AIS-Informationen ist, wie u.a. im vorherigen Kapitel beschrieben, die Position des Schiffes zum Zeitpunkt der Sendung. Die Position wird mit Hilfe von (D-)GPS-Empfängern ermittelt. An Bord befinden sich immer mehrere Messeinrichtungen, die variabel angeordnet sein können. Aus technischen Gründen ist der per AIS gesendete Ort die Position der Positionsmesseinrichtung. Innerhalb der AIS-Daten wird jedoch die Position der an Bord befindlichen Positionsmesseinrichtung zum Zeitpunkt der Sendung ebenfalls mitgeteilt.

Damit ist der übermittelte Ort in Relation zum Schiffsmittelpunkt immer mit einer gewissen Differenz behaftet, der sich zu den sonstigen Messfehlern (GPS-Ungenauigkeit) hinzuaddiert.

Da Schiffe, die den Bereich der Kadettrinne durchlaufen, durchaus eine Länge von 250 m haben können, ist es für eine korrekte Reproduktion der Vorgänge erforderlich, die gesendete Position um die relative Lage des (D-)GPS Empfänger auf einen konstanten Referenzpunkt bei allen Schiffen zu korrigieren. Würde dies nicht korrigiert, so zeigt das folgende Beispiel, dass die gesendete Position des Schiffes (gefüllte Markierung) beispielsweise 125 m vom echten Mittelpunkt des Schiffes (ungefüllte Markierung) entfernt sein kann.

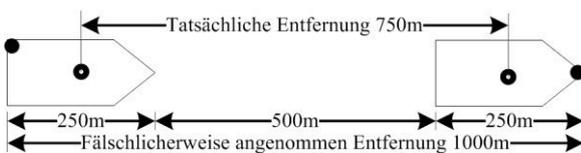


Abbildung 4. Korrektur der Antennenposition

Als Konsequenz könnte sich der Fehler bei zwei aufeinanderfolgenden Schiffen auf 250 m summieren. Dies wurde als unzulässiger Fehler bei der Projektbearbeitung identifiziert und eine Korrektur vorgenommen. Dazu wurde mit Hilfe einer Drehmatrix [Schneider 2014] die UTM des Schiffes um die Position des GPS-Empfängers an Bord auf den Mittelpunkt des Schiffes korrigiert.

7 Verfahren zur Korrektur der Schiffsgeschwindigkeit

Für eine korrekte Modellierung der Verkehrsvorgänge muss eine Überprüfung der gemeldeten Geschwindigkeit erfolgen. Durch technische Defekte der Geschwindigkeitsmeseinrichtung kann es zu einem vereinzelt, temporären oder dauerhaften Ausfall des Geschwindigkeitsmessgerätes kommen. Beispiele für diese Ausfälle sind in Abbildung 5 dargestellt. Über die Gründe des Ausfalls kann im Einzelfall nur spekuliert werden. Erkennbar ist dies durch eine im AIS kodierte Geschwindigkeitsmeldung (Speed over Ground [SoG]) von 102,3 kn.

Die Korrektur wurde mithilfe der relativen Positionsveränderung als euklidischer Abstand der Vorgänger zur Nachfolgermeldung in Relation zur vergangen Zeit in kn ermittelt. Die ermittelte Geschwindigkeit unterliegt durch die GPS-Ungenauigkeit Schwankungen, welche durch einen gleitenden Mittelwert angeglichen wurde.

Soweit im Datensatz eine fehlerhafte Geschwindigkeitsmeldung abgelegt wurde, ist dieser Wert mit den errechneten Geschwindigkeitswerten ersetzt worden.

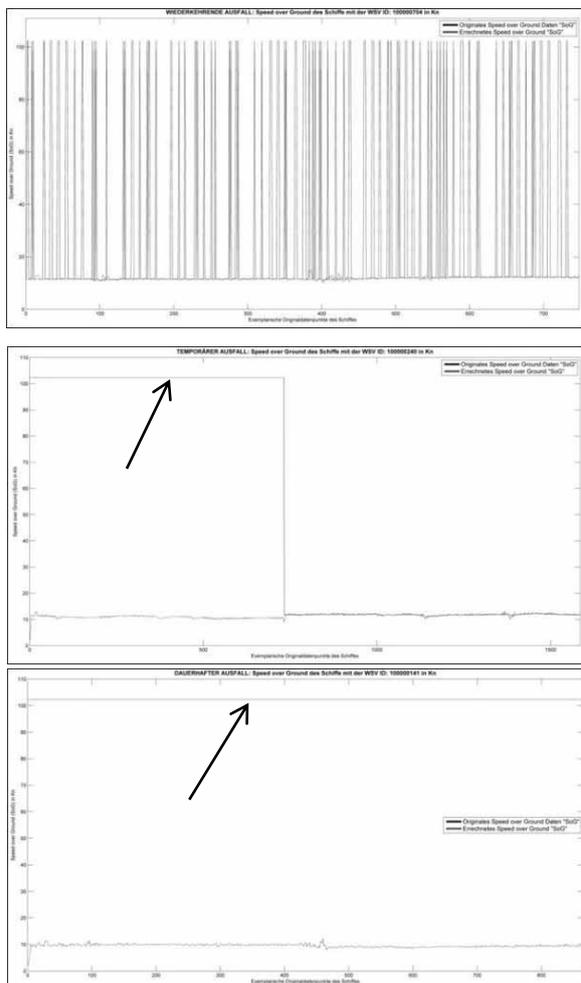


Abbildung 5. Vereinzelt auftretender (oben), temporärer (mitte) und dauerhafter (unten) Ausfall des SoG

8 Verfahren zur Korrektur des Schiffskurses

Der Kurs des Schiffes (Course over Ground [CoG]) ist eine weitere essenzielle Datengrundlage für die Realitätsabbildung. Wie die Geschwindigkeitsmess-einrichtung kann auch die Aufzeichnung des gesteu-erten Kurses fehlerhaft sein. Der Datenbereich des Kurses ist in Rechtsweisend-Nord von 0° bis 359° angegeben. Ein Fehler in der Aufnahme oder Weiter-leitung des Kurses wird durch ein CoG von 360° angezeigt.

Die Korrektur des CoG erfolgte mit Hilfe des jewei-ligen Skalarprodukts des Winkels des Richtungsvek-tors der Vorgängerposition \vec{P}_1 zur Nachfolgerposition \vec{P}_2 in Relation zur Ordinate mathematisch negativ drehend (da Rechtsweisend-Nord). Ebenso wie die Geschwindigkeit wurde auch der Kurswinkel γ über

einen gleitenden Mittelwert geglättet, um so der GPS-Ungenauigkeit entgegen zu wirken.

Auf den nachfolgenden Auswertungen sind der CoG und die auftretenden Ausfälle dargestellt. Da alle Passagen einer WSV-ID aus Performancegründen in einem Durchlauf untersucht werden (nord- wie süd-gehend), erscheint bei einer neuen Passage, die dann entgegen der vorherigen verläuft, ein Sprung in der Winkelanzeige (Ordinate). Auf der Abszisse ist der Datensatz des Schiffes angezeichnet.

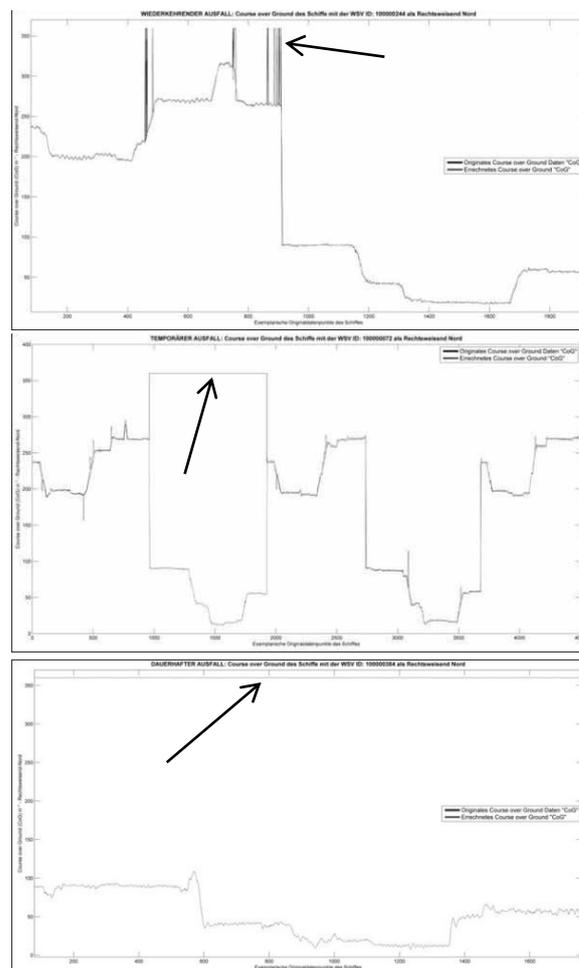


Abbildung 6. Vereinzelt auftretender (oben), temporärer (mitte) und dauerhafter Ausfall (unten) des CoG

9 Identifikation der Stromteilnehmer

Das Untersuchungsgebiet umfasst den Bereich der Kadetrinne, welches durch ein Verkehrstrennungsgebiet (VTG) „South of Gedser“ verkehrlich geordnet ist.



Abbildung 7. Bereich Kadetrinne [Quelle: Elektronische Seekarte]

An seiner nördlichen Seite ist dieses Gebiet natürlich begrenzt durch die dänische Insel Falster mit dem Ausläufer Gedser Riff. Südlich ist das VTG durch die 2007 eingerichtete Küstenverkehrszone „Fischland“ eingefasst.

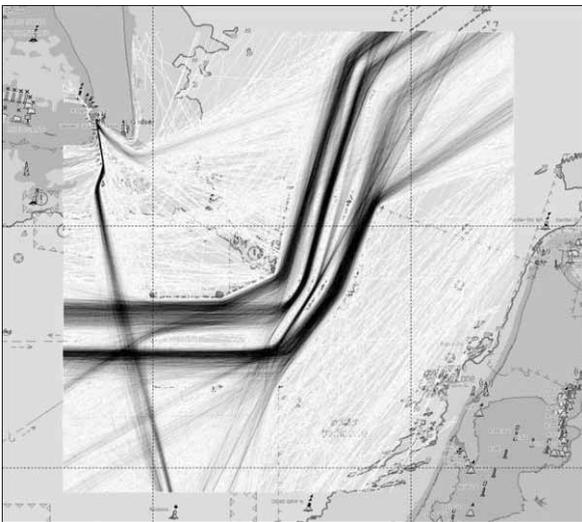


Abbildung 8. Untersuchungsgebiet mit Verkehr 03/2010 [Quelle: Hartmann; Hilgenfeld 2013]

Wie auf der Visualisierung in Abbildung 8 erkennbar ist, nutzen neben den Schiffen die dem VTG folgen etliche andere Schiffe das Seegebiet (weiße Linien), ohne dabei Bestandteil des Verkehrsstroms zu sein. Für die Erfassung der relevanten Schiffsbewegungen ist es damit unerlässlich zu identifizieren, welche Trajektorie dem definierten Verkehrsstrom entspricht und welcher Teilnehmer den Seeraum nutzt, aber kein Verkehrsteilnehmer ist. Ziel ist es, alle Fahrzeugpassagen, die nicht Bestandteil des definierten Verkehrs-

stroms waren, auszuwählen und zu entfernen ohne dabei tatsächliche Passagen fälschlicherweise zu eliminieren.

Im ersten Schritt wurde der Seeraum festgelegt, der den Kern der zu ermittelnden Datenmenge bildet. Dazu wurde sich für die See-Tonne 71 und 72 als Grenzen entschieden, da für diesen Bereich auch die Verkehrssimulation entwickelt werden soll. Als Entscheidungskriterium wurde festgelegt, dass die Trajektorie der WSV-ID einmal innerhalb der in dem Kerngebiet definierten zwei Dreiecke liegen sollte (Abbildung 9).

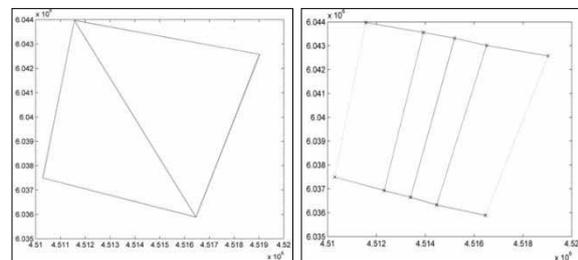


Abbildung 9. Kernfläche der KDR (links) und KDR mit Tief [Innen]- und Flachwasserbereich (rechts)

Liegt die Trajektorie nicht in der Kernfläche der KDR können alle existierenden Daten der WSV-ID gelöscht werden. Als nächstes empfiehlt sich, Daten die länger als 30 min vor Eintreten und nach Verlassen der KDR lagen zu löschen. Nach Erfahrungen im Projekt konnte 50% des benötigten Speicherplatzes eingespart werden.

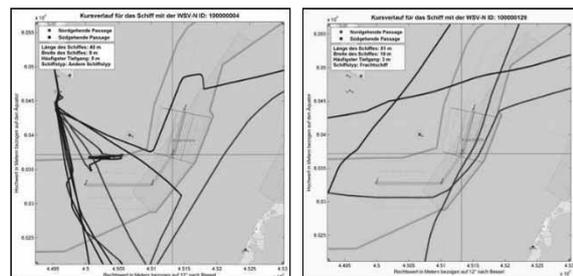


Abbildung 10. Beispiel für eine (links) und zwei (rechts) Passagen

Mit Entfernung der unnötigen Daten wurden den Trajektorien Passagen zugeordnet, um zu entscheiden, welche Kursinformationen zu einer Durchfahrt gehörten. Beispiele für eine oder zwei Passagen sind in Abbildung 10 dargestellt. Diese Passagen, die durch den Kernbereich der KDR führten, wurden hinsichtlich des Kreuzens untersucht. Fahrzeuge können den Bereich des VTG nach Kollisionsverhütungsregel 16 überqueren, dürfen dabei aber Schiffe, die dem Verkehrsstrom des VTG folgen, nicht behindern. Ein kreuzendes Fahrzeug ist in Abbildung 11 illustriert.

Im nächsten Schritt wurden die Verweildauer im Kerngebiet der KDR und die Schiffsgeschwindigkeit untersucht. Dabei wurden alle Schiffe identifiziert, die zeitweise weniger als 4 kn in diesem Bereich fahren und somit konnten z.B. alle Tonnenleger, Wartungsschiffe und Angler aus der AIS-Projektdatenbank entfernt werden.

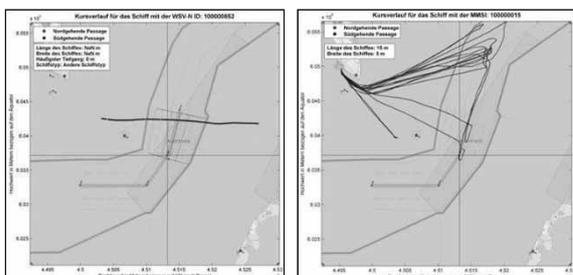


Abbildung 11. Beispiel für ein kreuzendes Fahrzeug (links) und eine Passagezeit kleiner 8 min (rechts)

Abschließend wurde überprüft, ob die Trajektorie weniger als 8 min Aufenthalt im Kernbereich hatte (vgl. Abbildung 11). 99,9% aller Schiffe fahren 24 kn oder weniger (vgl. Abbildung 12) in diesem Seegebiet und benötigen für die 3,66 sm lange Passage mindestens die o.g. 8 min.

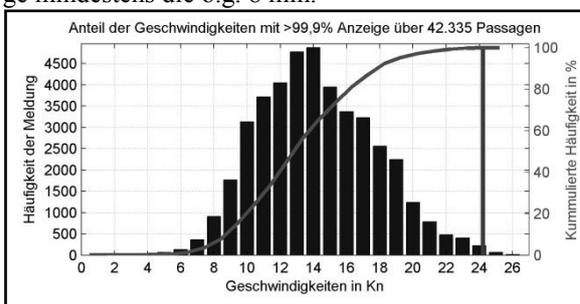


Abbildung 12. Geschwindigkeiten im Kernbereich

Schiffe mit weniger als 8 min Passierzeit und Geschwindigkeit von mehr als 4 kn, sind Fahrzeuge, die das Kerngebiet der KDR nur touchiert haben (Abbildung 13).

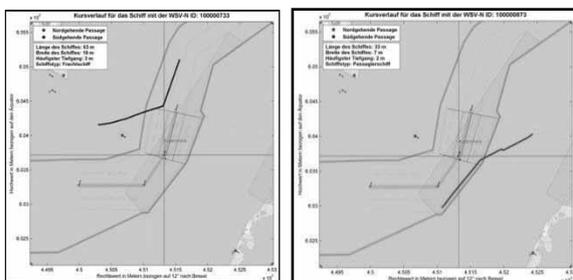


Abbildung 13. Beispiel für touchierende Passagen

10 Zusammenfassung

Es konnte dargestellt werden, wie die Daten nach dem Datenabruf bei der WSV-N aufzuarbeiten und nutzbar gemacht werden können. Hierdurch ist eine Implementierung in eine Simulationsumgebung und eine Validierung mittels Realdaten möglich. Einer der Schwerpunkte war die Identifizierung der Fahrzeuge, die tatsächlich Teilnehmer des Verkehrsstroms sind, was beispielhaft für eine WSV-ID in Abbildung 14 dargestellt ist. Die Datensätze aller anderen Schiffe, die kein Bestandteil des Verkehrsstroms sind, werden über die hier beschriebenen Abfragen entfernt. Dabei verdeutlichen die gezeigten Beispielen, wie schwierig teilweise eine Entscheidungslogik zu definieren ist, um eine Trajektorie dem Strom zuzuordnen oder zu verwerfen.

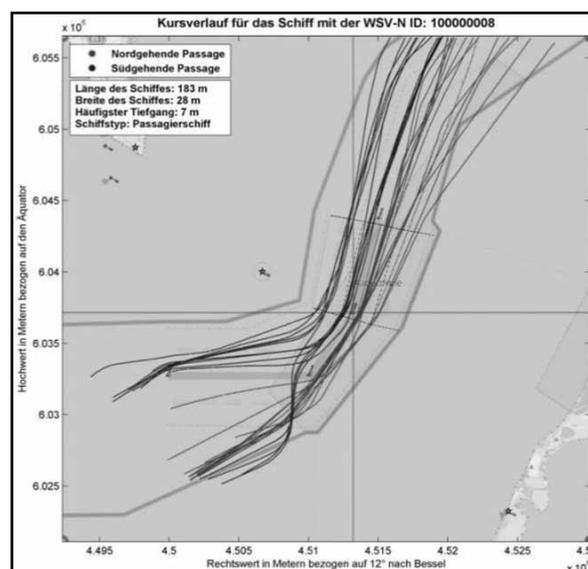


Abbildung 14. Beispiel alle Trajektorien eines Schiffs über 90 Tage

11 Literatur

- [Baldauf et al. 2006] Baldauf, M.; Benedict, K.; Fischer, S.; Hartmann, F.: *Technische Zuverlässigkeit von AIS - Durchführung und Ergebnisse einer Szenariostudie. Sicherer und effektiver Seeverkehr*. Schiffahrtsinstitut Warnemünde e. V. Schiffahrtskolleg (12.), Warnemünde, 12.11.2006.
- [Böhme et al. 1984] Böhme, J.; Steinfatt, W.; Uhlig, L. (Hrsg.): *Leitfaden der Navigation: Automatisierung der Navigation*. Transpress, Verl. für Verkehrswesen, Berlin, 1984.
- [Heymann 2013] Heymann, F.: *Plausibility analysis of navigation related AIS parameter based on time series*, Neustrelitz, Deutschland, 2013.
- [Hilgenfeld et al. 2014] Hilgenfeld, C.; Hartmann, F.; Ahn, M.: *AIS data as a basis for industry and science*. HANSA International maritime Journal 151 (2014), 8, S. 74–75.
- [Hoppe 2006] Hoppe, M.: *DGPS Dienst der WSV - IALA Referenzstation Groß Mohrdorf. Signalausbreitung und Nutzerreichweiten*, Koblenz, 2006.
- [IMEA 2013] International Marine Electronics Association (IMEA): *NMEA 0183 V.4.1: Standard for Interfacing Marine Electronic Devices*. Stand: Juni 2012, Severna Park, MD, USA, 2013.
- [ITU 2010] International Telecommunication Union (ITU): *ITU-R M.1371-4 - NMEA 0183 V.4.0: Technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band*, Genua, 2010.
- [Mou et al. 2010] Mou, J. M.; Tak, C. v. d.; Ligteringen, H.: *Study on collision avoidance in busy waterways by using AIS Data*. Ocean Engineering (2010), 5, S. 483–490.
- [Norris 2008] Norris, A. (Hrsg.): *Integrating ship bridge systems: Radar and AIS*. Nautical Institute, London, 2008.
- [Raymond 2015] Raymond, E. S.: *AIVDM/AIVDO protocol decoding: AIS-NMEA-Sendungstypen mit Koodierungsschlüssel und Bedeutung*, <http://catb.org/gpsd/AIVDM.html>; Zugriff am 07.05.2015.
- [Schneider 2014] Schneider, A.: *Mathebibel: Matrizenrechnung / Drehmatrix*, <http://www.mathebibel.de/drehmatrix>; Zugriff am 22.10.2014.
- [Vojdani et al. 2013] Vojdani, N.; Ahn, M.; Hilgenfeld, C.: *Nutzung von AIS-Daten der WSV in der Forschung. Tendenzen und Entwicklungen in der Anwendung des Seerechts*. Schiffahrtsinstitut Warnemünde e. V. Schiffahrtskolleg (19.), Warnemünde, 27.11.2013.

Einsatz von Strömungssimulationen bei der Optimierung der Aerodynamik moderner Kreuzfahrtschiffe

Holm Stückrad¹, Mark Bludszuweit¹

¹MET Motoren- und Energietechnik GmbH, Rostock

holm.s@met-online.com

Ausgehend von Anforderungen, die der weiter boomende Kreuzfahrtmarkt stellt, wird gezeigt, wie in der täglichen Praxis eines Berechnungsdienstleisters dem Stand der Technik entsprechende Strömungssimulationsstools (CFD) eingesetzt werden, um Schiffe mit - höchstem Komfort und Unterhaltungswert für die Passagiere und - sicherem und energieeffizientem Betrieb zu entwerfen. Im Fokus des Vortrags steht dabei die Lösung des „Konflikts“ zwischen den theoretischen Möglichkeiten der numerischen Simulation und den realen wirtschaftlich-technischen Rahmenbedingungen. Beispielhaft wird demonstriert, wie - die Außen-decksbereiche moderner Kreuzfahrtschiffe in ein numerisches Modell abgebildet werden, - das Schiff unter verschiedenen Randbedingungen simuliert wird, - welche Schlussfolgerungen hinsichtlich des Windkomforts getroffen werden, - wie die Beeinträchtigung durch gasförmige Emissionen vermieden wird, - Vorhersagen für das Klima großer Atrien und Freizeitbereiche erstellt werden.

1 Einleitung

Industriestandard für die Bestimmung des Luftwiderstands von Kreuzfahrtschiffen, der Rauchgasausbreitung und des Windkomforts auf den offenen Passagierdecks sind Untersuchungen an skalierten Modellen im Windkanal. Trotz einer dem Stand der Technik entsprechenden Ausstattung der Versuchseinrichtungen mit drucksensitiven Farben (PSP - pressure sensitive paint), der PIV-Technologie (particle image velocimetry) und des Rapid Manufacturing, haben Windkanäle entscheidende Nachteile gegenüber der Anwendung numerischer Strömungssimulationen (CFD). Während CFD-Methoden die gesamtheitliche Bewertung der komplexen Luftströmung um den Überwasserteil eines Schiffs bei gleichzeitig möglicher Variation einer Vielzahl von Randbedingungen wie Anströmrichtung, Anströmprofil, Bewegung des Schiffes, Luft-, Abgas- und Abluftein- bzw. -austritte, Wandrauheit usw. ermöglichen, liefern Modelluntersuchungen im Windkanal maßstabsbehaftete, punktuelle Ergebnisse der Geschwindigkeiten und Drücke bzw. qualitativ-phänomenologische Beobachtungen zur Abgasausbreitung. Aufgrund des begrenzten Messquerschnitts eines Windkanals müssen die erforderlichen Untersuchungen mit relativ kleinen Modellen durchgeführt werden. Das schränkt die Aussagekraft der Ergebnisse bezüglich der Großausführung ein und bestimmte Fragestellungen z.B. zu thermisch getriebenen Strömungen oder der Ausbreitung von Partikeln können nicht oder nicht befriedigend geklärt werden.

Mit zunehmendem Fortschritt von Hard- und Software gewinnen numerische Methoden der Strömungssimulation (CFD) immer mehr an Akzeptanz und Marktanteilen. Dazu muss der Aufwand für die Modellerstellung und die numerische Lösung der Gleichungssysteme in einem kommerziell vertretbaren Rahmen bleiben.

Um das sicherzustellen, werden geometrische Details mit einer Genauigkeit von ca. 0,5m bis 1m aufgelöst und raum- und zeitgemittelte Größen als stationäre Lösung ermittelt. Solche Analysen werden zur Vorhersage des aerodynamischen Widerstands, der Rauchausbreitung und auch von Wechselwirkungen zwischen verschiedenen Ein- und Austritten benutzt.

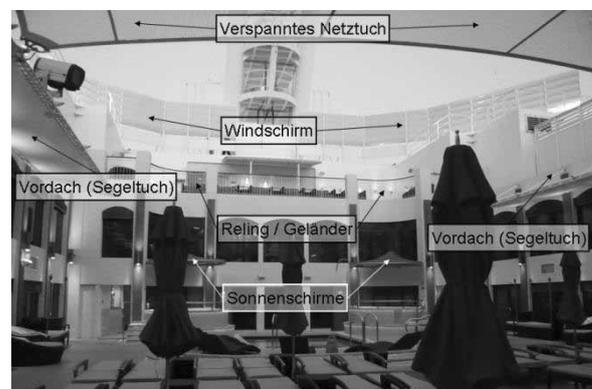


Abbildung 1: Lokale Designdetails

Schiffsbetreiber fordern aber von den Werften und Designbüros in zunehmendem Maß zuverlässige

Aussagen über die lokalen Auswirkungen geometrischer Details im Bereich $< 0,1\text{m}$ sowohl auf den Passagierkomfort (Wind und Abgaskonzentration) als auch auf die Bewertung von Risiko- und Gefahrensituationen. Windschirme, verspannte Netze, Relinge, Schutzdächer, Geländer, Balustraden (**Abbildung 1**) sowie relevante Ein- und Austritte gasförmiger Medien müssen dazu im Zentimeterbereich aufgelöst werden und auch zeitabhängige Effekte sind, wenn erforderlich, bei der Bewertung der Komfortsituation einzubeziehen. Das führt zu sehr großen Simulationsmodellen mit entsprechend langen Rechenzeiten. Dieser Aufwand ist aber gegenwärtig nur in besonderen Fällen wie etwa beim Bau von Mega-Yachten kommerziell zu rechtfertigen.

Um einen möglichst durchgängig simulationsgestützten Entwurfsprozess zu realisieren, ist es daher erforderlich, den hohen manuellen Aufwand beim Einpflegen einer Vielzahl von geometrischen Details während der Umsetzung komplexer Geometrien in handhabbare CFD-Modelle zu reduzieren, ohne die erforderliche Aussagefähigkeit der Simulationsergebnisse zu beeinträchtigen.

2 Wissenschaftlich technischer Stand

Das Widerstandskomitee der 25. ITTC-Konferenz (2008) [1] arbeitete erstmalig Untersuchungen zu Windeffekten und zu Luftnachlaufströmungen am Überwasserschiff auf. Im Ergebnis dieser Betrachtungen wurde festgestellt, dass, bis zu diesem Zeitpunkt, für die Vorhersage von aerodynamischen Kräften an Schiffen im Wesentlichen experimentelle Methoden (EFD – experimental fluid dynamics) eingesetzt wurden, während für die Vorhersage des Strömungsverlaufs hauptsächlich numerische Methoden (CFD – computational fluid dynamics) zur Anwendung kamen.

So konzentrierten sich frühe experimentelle Forschungsarbeiten aus den dreißiger bis in die fünfziger Jahre des vergangenen Jahrhunderts auf den Luftwiderstand der Schiffsaufbauten und die auf sie wirkenden Windkräfte.

Darauf aufbauend wurden die in den EFD-Studien gewonnenen Daten verwendet, um empirische Formeln für die Vorhersage aerodynamischer Kräfte und Momente zu entwickeln. So veröffentlichte Isherwood (1972) [2] eine auf linearer, multipler Regression beruhende Methode für Handelsschiffe, deren Ergebnisse von Inoue und Ishibashi (1972) [3] be-

nutzt wurden, um die Manövrierbarkeit von Schiffen und deren Kursstabilität zu untersuchen. Die Weiterentwicklung der experimentellen Methoden erlaubte schließlich auch die Untersuchung komplexerer Schiffsgeometrien bei realistischeren Windbedingungen. Blendermann (1995) [4] führte Windkanalmessungen in einem ungleichförmigen Luftstrom durch und entwickelte daraus eine Methode zur Abschätzung der Windlast auf Schiffe. Nimura u. a. (1997) [5] untersuchte ein Tankschiff unter Ballastbedingungen und führte Windkanaltests nicht nur für die Ermittlung der Luftkräfte und –momente sondern auch für die Visualisierung des Strömungsfelds durch.

Mit der Entwicklung numerischer Methoden zur Lösung der Navier-Stokes-Gleichungen einerseits und der Verfügbarkeit leistungsfähiger Rechner andererseits wurde seit Anfang der neunziger Jahre des vergangenen Jahrhunderts intensiv daran gearbeitet, die Luftumströmung um maritime Strukturen mit Hilfe von CFD-Methoden zu beschreiben.

Das Danish Maritime Institut (DMI) führte umfassende aerodynamische CFD-Untersuchungen für maritime Einrichtungen durch. Auf der Grundlage vorangegangener Arbeiten von Leer-Andersen und Hughes (1996) [6] an der Danish Technical University (DTU), beschreiben Aage u. a. (1997) [7], Hvid u. a. (1997) [8] RANSE Anwendungen auf ein Fährschiff und eine Offshore-Plattform zur Bestimmung der Windkräfte und der Rauchausbreitung.

Reddy u. a. (2000) [9] simulierten unter Verwendung eines kommerziellen CFD-Codes eine turbulente Luftströmung mit Windrichtungen von 0° , 45° und 90° um eine typische Fregatten-Form. Jin u. a. (2001) untersuchten die Ausbreitung von NO_2 aus dem Schornstein eines Tankers in Abhängigkeit von verschiedenen geometrischen Parametern der Schornstein-konstruktion. Für die Berechnung wurde die Buggeometrie vereinfacht und einige Details wie das Rohrleitungssystem und der Radarmast nicht berücksichtigt.

Yelland u. a. (2001) simulierten die aerodynamische Strömung um mehrere, vereinfachte typische Frachtschiffe um den Einfluss der Aufbauten auf Windmessenrichtungen für meteorologische Forschungen zu bewerten.

Ebenso wurden auch an der Hamburgischen Schiffbauversuchsanstalt (HSVA) im Laufe des letzten Jahrzehnts eine Reihe von CFD gestützten Untersuchungen von Schiffsaufbauten durchgeführt (El

Moctar u. a. (2001) [10] u. [11], El Moctar und Bertram (2002) [12], Lindenau u. a. (2002) [13], Schmode und Bertram (2002) [14], Bertram u. a. (2002) [15]). Die Anwendungen untersuchen unterschiedliche Schiffstypen (Containerschiff, SES, schnelles Fährschiff, Kreuzfahrtschiff) und umfassen u. a. die Rauchausbreitung und die thermodynamische Analyse des Schornsteins.

Die MET Motoren- und Energietechnik GmbH wendet seit Mitte der neunziger Jahre moderne CFD-Simulationsmethoden für die systematische Analyse der Aerodynamik von Kreuzfahrtschiffen im Auftrag weltweit führender Schiffbauunternehmen erfolgreich an. Unser Ziel ist es, Werften und Reeder bei der Suche nach der besten Kombination aus zeitgemäßer Schiffsarchitektur und maximalem Komfort für die Passagiere effizient zu unterstützen.

Zusammenfassend ergibt sich aus der Auswertung der Literatur, dass beim Entwurf von Kreuzfahrtschiffen zwar Windkanaluntersuchungen für die Bestimmung des Luftwiderstands, der Rauchgasausbreitung und der Abschätzung des Windkomforts auf den offenen Passagierdecks benutzt werden, diese aber den aktuellen Forderungen der Schiffsbetreiber nicht mehr gerecht werden [16].

3 Bauformen und Entwicklungstrends von Kreuzfahrtschiffen

Im Jahr 2004 waren Kreuzfahrtschiffe mit etwa 2.000 Passagieren die größten und luxuriösesten Schiffe im Einsatz. Als größtes Kreuzfahrtschiff zu dieser Zeit galt die *Voyager of the Seas* (Royal Caribbean) für über 3.000 Passagiere. In der Regel werden große Kreuzfahrtschiffe heute für ca. 4.000 Passagiere ausgelegt, wobei der Rekord gegenwärtig bei über 6.000 Passagieren liegt (*Oasis of the Seas* / *Allure of the Seas* (Royal Caribbean)). Die Auswertung der Neubauten der letzten 5 Jahre zeigt, dass auch kleinere Schiffe in Auftrag gegeben werden, jedoch ausschließlich für die höchste Luxusklasse oder für besondere Einsatzcharakteristiken wie z.B. Expeditionskreuzfahrten. Dementsprechend größer werden die Schiffe auch in ihren Abmessungen. Schiffslängen unter 200 Metern sind nur noch bei älteren oder speziellen Schiffen zu finden. Standard sind heute Längen von ca. 300 m. Die größten Einheiten sind sogar 360 m lang. Die Schiffshöhen erreichen dabei Werte von bis zu 72 m bei 17 Decks.

Reederei	Projekt	Bauwerft	Indienststellung	Vermessung [BRZ]	Passagiere
Royal Caribbean International	Oasis	STX Finland Oy, Turku	2009	225.262	5.400
	Sunshine	MEYER WERFT, Papenburg	2013	158.000	4.100
	Oasis	STX France, Saint-Nazaire	2016	225.262	5.400
Princess Cruises	Royal Princess	Fincantieri S.p.A., Monfalcone	2013	141.000	3.600
Norwegian Cruise Line	Norwegian Epic	STX France SA, Saint Nazaire	2010	155.873	4.600
	Norwegian Breakaway	MEYER WERFT, Papenburg	2013	144.000	4.000
	Norwegian Breakaway-plus	MEYER WERFT, Papenburg	2015	163.000	4.200
AIDA Cruises	AIDAmar (Sphinx)	MEYER WERFT, Papenburg	2012	71.304	2.500
	Hyperion	Mitsubishi Heavy Industries, Nagasaki	2014	125.000	3.250

Tabelle 1: Unter Vertrag befindliche Neubauten bis 2016 im Vergleich zu kürzlich abgelieferten Schiffen

In **Tabelle 1** sind wichtige unter Vertrag befindliche Neubauprojekte und kürzlich in Dienst gestellte Schiffe aufgelistet. Diese Schiffe überschreiten damit das „Panamax“-Maß von 32,30 m Schiffsbreite und 294 m Schiffslänge, welches die maximale Größe für Schiffe darstellt, die den Panamakanal nutzen können. Damit ist diese neue Generation von Kreuzfahrtschiffen nur noch bedingt geeignet, wirklich alle Weltmeere zu befahren. Die fast ausschließlich in Europa gebauten Kreuzfahrtschiffe sind eigentlich nur für den Atlantik und direkt angrenzende Meere konzipiert.

In Bezug auf die Tonnage der Schiffe bedeutet das, dass aktuell in Auftrag gegebene Neubauten eine Größe von 120.000 (MSC, Fantasia-Klasse) bis 220.000 BRZ (Royal Caribbean, *Oasis*) haben. Vor ca. 5 Jahren wurden die meisten Schiffe dagegen noch mit einer deutlich geringeren Tonnage von 60.000 bis 80.000 BZR (Schiffe für ca. 2.000 Passagiere) gebaut. Im Vergleich dazu haben traditionelle Kreuzfahrtschiffe wie z.B. die *Deutschland* und die *Europa* eine Vermessung von lediglich 22.000 bzw. 29.000 BRZ.

Entsprechend der zunehmenden Schiffsgröße stiegen auch die durchschnittlichen Bausummen von Kreuzfahrtschiffen von ca. 300 Mio. Euro vor ca. 5 Jahren auf etwa 400 bis 500 Mio. Euro für z.B. die neuen Schiffe von Costa oder MSC bis hin zu 900 Mio. Euro bei Schiffen der *Oasis*-Klasse oder der NCL *Norwegian Epic*.

Einteilen lassen sich Kreuzfahrtschiffe im Wesentlichen an den Selbstdarstellungen der Reedereien in Form ihrer Firmenphilosophie. Diese ist aus Gründen der Marketingstrategie oft verklausuliert umschrieben, jedoch an der Umsetzung auf den jeweiligen Schiffen und in der Bewirtschaftung der Schiffe sehr gut erkennbar (Abbildung 1).

Schiffsgröße	Angebote an Bord
Boutique 20 bis 200 Passagiere	
Klein 200 bis 500 Passagiere	
Mittelgroß 500 bis 1200 Passagiere	
Groß 1200 bis 2000 Passagiere	
Sehr Groß 2000 bis 3000 Passagiere	
Mega 3000 bis 4000 Passagiere	
Ultra über 4000 Passagiere	

Angebote an Bord
Fun - Erlebnis - Freizeitsport
Club - Unterhaltung
Leben - Gemächeln
Komfort - Luxus
Erlebnis - Expedition
Kultur - Erholung

Abbildung 2: Typisierung von Kreuzfahrtschiffen nach Jans [17]

Exemplarisch kann hier die Norwegian Cruise Line (NCL) angeführt werden, bei der auf jedem Schiff das Freestyle Konzept durchgängig umgesetzt wird. Bestens erkennbar ist dies anhand der Deckspläne und in der Bewirtschaftung im Restaurantbereich, ebenso wie im Freizeitangebot. **Abbildung 2** zeigt links die Einteilung der Kreuzfahrtschiffe nach Schiffsgröße, wobei aktuelle Trends der Entwicklung der Schiffsgröße berücksichtigt wurden. Das allerdings unter der Voraussetzung, dass die Unterteilung einen inhaltlichen Hintergrund hat und nicht willkürlich gesetzt ist. Die sechs Angebotstypen sind idealtypische Zusammenführungen, die durch Jans [17] anhand der Selbstdarstellung von Reedereien, in Katalogen und im Internet abgeleitet wurden. Reedereien lassen sich indirekt diesen Angebotstypen zuordnen – aber wie bei jedem Idealtypus sind auch stets Elemente der anderen Typen im konkreten Praxisfall mit betroffen. Die dunklen Schattierungen in den Balken zeigen, in welcher Schiffsgröße der jeweilige Angebotstypus primär verankert ist.



Abbildung 3: links: SeaWalk auf der *Royal Princess* [18]; rechts: Aquapark auf der *Norwegian Breakaway* [19]

Um sich auf dem nach wie vor wachsenden Kreuzfahrtmarkt gegen Mitbewerber zu behaupten, versuchen die Kreuzfahrtreedereien mit neuen konstruktiven Merkmalen an ihren Schiffen wie z.B. dem SeaWalk (*Royal Princess*, Princess Cruises) oder einem Aquapark mit Wasserrutschen und Pools in den verschiedensten Ausführungen (Norwegian Cruise Line, *Norwegian Epic / Breakaway*) (**Abbildung 3**), die Attraktivität ihrer Angebote zu erhöhen. Diese hoch über der Wasseroberfläche gelegenen Bereiche auf

den offenen Decks stellen besondere Anforderungen an die Aerodynamik des Schiffs, da diese unter allen Betriebsbedingungen frei von Abgasen und Ablüften gehalten werden müssen und ein entsprechender Windkomfort zu gewährleisten ist.

4 Referenzobjekte

Für die durchgeführten Untersuchungen wurde als Referenzobjekt ein generisches Schiff verwendet, welches alle typischen Designelemente eines modernen Kreuzfahrtschiffes für ca. 2.000 Passagiere enthält. Dazu zählen u. a.:

- ein umlaufendes Deck über dem Bootsdeck mit einer breiten Vielfalt an gastronomischen Einrichtungen und Erholungsmöglichkeiten
- Balkonkabinen an den Seiten (Standardkabinen), vorn und hinten (Suiten) an den Aufbauten
- Courtyard im Bereich der oberen Decks
- Ausgedehnter Freizeit- und Erholungsbereich auf den oberen Decks mit Pools, Liegeflächen, Sport- und gastronomischen Einrichtungen u.s.w.

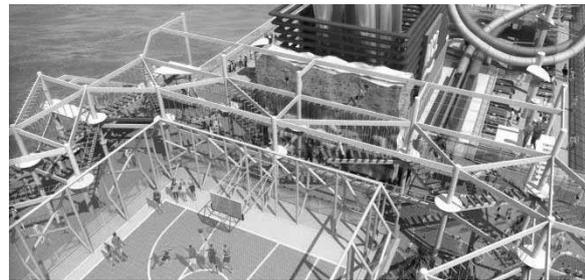


Abbildung 4: NCL *Breakaway* – Sportbereich (Basketballcourt, Klettergarten, Kletterwand [19])

Die Geometrie der generischen Kreuzfahrtschiffe wurde als Oberflächenmodell bzw. als Volumenmodell in unterschiedlichen Detaillierungsstufen (**Abbildung 5** und **Abbildung 6**) erzeugt und bildet die Grundlage für die Generierung der numerischen Gitter für die CFD-Modelle.

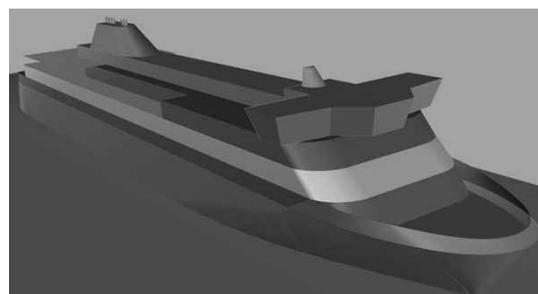


Abbildung 5: 3D-Oberflächenmodell eines generischen Kreuzfahrtschiffs (grobe Auflösung)

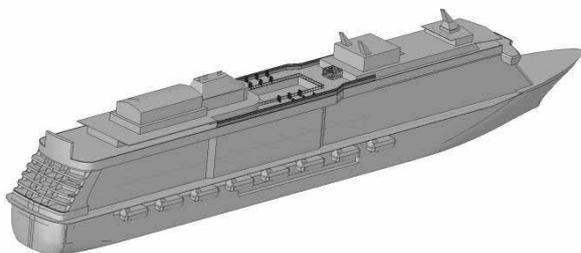


Abbildung 6: 3D-Volumenmodell eines generischen Kreuzfahrtschiffes (hohe Auflösung)

Um die Geometrie sich wiederholender kleinskaliger Baugruppen effizient beschreiben zu können, wurde ein Konzept zur parametrisierten Erfassung der Geometrie von Balkonen, Wind- und Sonnenschirmen, Relings, Balustraden und verspannten Netzen erarbeitet. Die Erstellung parametrisierter 3D-Volumenmodelle, deren automatische Vernetzung und die Berechnung und Analyse hochaufgelöster Strömungsfelder wurde anhand ausgewählter Beispiele realisiert. Die Verluste an den Ersatzmodellen werden dabei über einen von der Strömungsrichtung abhängigen quadratischen Widerstandsansatz berücksichtigt. **Abbildung 7** zeigt das Geschwindigkeitsfeld im Bereich eines durchströmten Geländers im Ergebnis der Optimierung der Widerstandskoeffizienten in Strömungsrichtung und quer zur Strömungsrichtung.

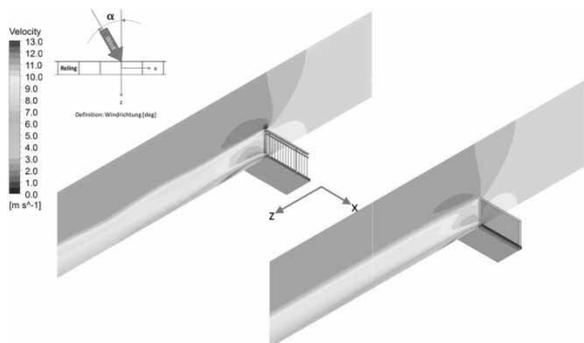


Abbildung 7 Geschwindigkeitsverteilung im Bereich eines Geländers bei 20 kn / 15° (links: HighRes Modell; rechts: Ersatzmodell)

Eine kritische Überprüfung der erzielten Ergebnisse zeigte, dass eine Substitution von Baugruppen durch geometrisch und/oder physikalisch vereinfachte Ersatzmodelle sinnvoll möglich ist.

5 Grobskaliges CFD-Modell

Unter Anwendung der durch die MET entwickelten Ersatzmodelle für Netze, Geländer u. ä. Designelemente wurde das Strömungsfeld um ein generisches

Kreuzfahrtschiff für eine charakteristische Geschwindigkeit und einen resultierenden Windangriffswinkel schräg von vorn simuliert, der entsprechend den Aussagen von Werften und Betreibern als kritisch für den Windkomfort anzusehen ist.

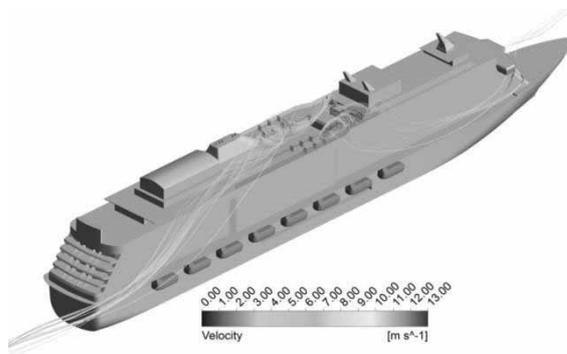


Abbildung 8: Stromlinien auf dem Passagierdeck

Das Strömungsfeld wurde für definierte Bereiche umfassend visualisiert. Dabei wurden Ansichten für eine spätere automatisierte Analyse vordefiniert und spezielle Auswerteskripte erstellt.

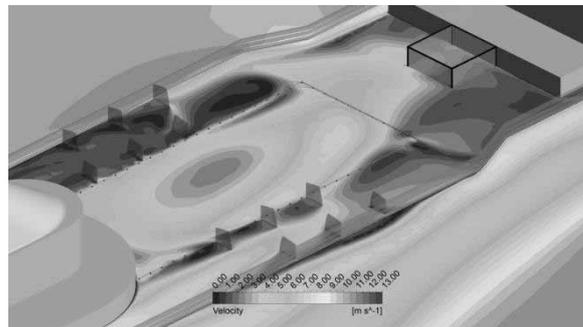


Abbildung 9: Geschwindigkeitsverteilung 0,95 m über dem Passagierdeck (Windschirme, Geländer, Netze – Ersatzmodell)

6 Hochaufgelöstes CFD-Modell

Als Referenzobjekt für die Untersuchungen wurde die äußere Hülle eines Schiffs entworfen, welches alle wesentlichen Gestaltungsmerkmale eines modernen, sehr großen Kreuzfahrtschiffs (2.000 bis 3.000 Passagiere) hat.

Das Schiff hat folgende Hauptabmessungen:

Länge über alles:	264 m
Breite in der Wasserlinie:	27 m
Höhe über Wasserspiegel:	48 m

Das Schiff verfügt über einen kompakten Hotelbereich, offene Passagierdecks im vorderen, mittleren und achterlichen Bereich, einen halbachterlich ange-

ordneten Schornstein sowie eine Videowand am Heck. Beispielhaft sind die hochaufgelösten Details wie Geländer (Abbildung 10) und Windschirme (Abbildung 11) dargestellt. Austritte für gasförmige Medien sind exemplarisch am Schornstein (Abgase der Hauptmaschinen) und an den mittschiffs angeordneten, doppelten Signalmasten (Küchenabluft) im Modell vorgesehen.

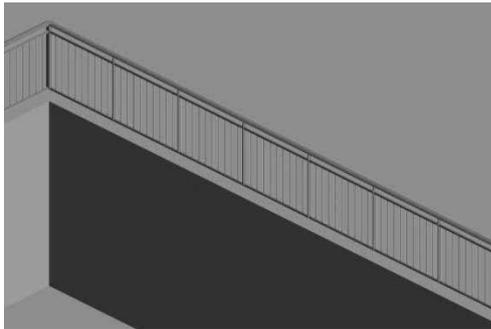


Abbildung 10: Geländer (Hochaufgelöstes Modell)

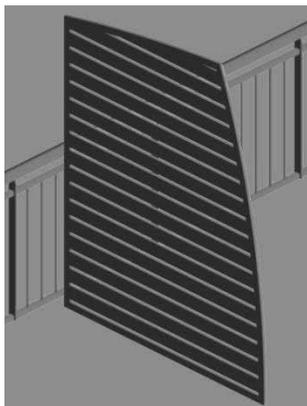


Abbildung 11: Windschirm (Hochaufgelöstes Modell)

Das hochaufgelöste CFD-Modell wurde unter identischen Randbedingungen wie das grob aufgelöste Modell berechnet. Die gewonnenen Ergebnisse wurden mit den Resultaten aus der Simulation des groben Modells verglichen.

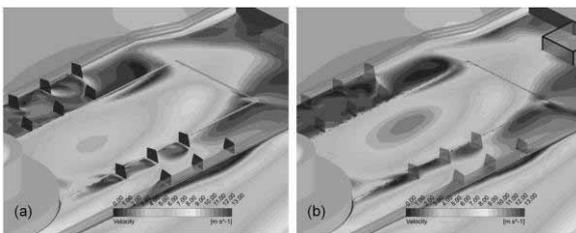


Abbildung 12: Geschwindigkeitsverteilung 0,95 m über dem offenen Passagierdeck; (a) Windschirme, Geländer, Netze – Geometrie vollständig aufgelöst; (b) Windschirme, Geländer, Netze – Ersatzmodell

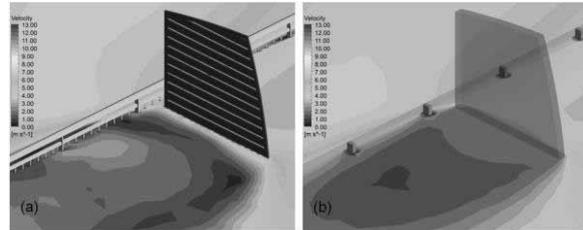


Abbildung 13: Geschwindigkeitsverteilung 0,95 m über dem offenen Passagierdeck; (a) Windschirme, Geländer, Netze – Geometrie vollständig aufgelöst; (b) Windschirme, Geländerstreben, Netze – Ersatzmodell; Pfosten - Solid

Abbildung 12 und Abbildung 13 zeigen anhand der Geschwindigkeitsverteilung in einer Ebene über dem offenen Passagierdeck den Vergleich zwischen dem CFD-Modell, welches bestimmte Konstruktionsdetails komplett auflöst (linke Darstellung), und dem Modell, welches entsprechende Ersatzmodelle verwendet (rechte Darstellung). Die Geschwindigkeitsverteilung, die unter Verwendung von Ersatzmodellen berechnet wurde, liegt sowohl qualitativ als auch quantitativ sehr dicht an dem Ergebnis, welches unter Verwendung eines hochauflösenden Gitters erzielt wurde. Der Vergleich der erforderlichen Hardwareressourcen zeigt, dass das Ersatzmodell mit einem Viertel des Aufwands, der für das hochauflösende Modell erforderlich ist, betrieben werden kann.

7 Zusammenfassung

Es wurde eine neuartige Methode für die Abbildung der komplexen dreidimensionalen Geometrie von modernen Kreuzfahrtschiffen in geeignete CFD-Modelle entwickelt und anhand ausgewählter Beispiele getestet. Je nach erforderlicher Detailtiefe werden die Genauigkeitsanforderungen der frühen, mittleren und späten Phase des Designprozesses effektiv unterstützt und so ein durchgängig simulationsgestützter Entwurfsprozess ermöglicht. Die Vorteile dieser Vorgehensweise liegen in einer schnellen Erzeugung numerischer Gitter mit minimierter Anzahl an Gitterelementen und dadurch erheblich reduzierten CPU-Zeiten pro Simulationsfall. Das ist eine wesentliche Voraussetzung, um entweder eine größere Anzahl von Varianten in die Betrachtungen einzubeziehen oder instationäre Analysen durchzuführen. Für kleinskalige Baugruppen, die einen entsprechenden Einfluss auf globale und/oder lokale Strömungszustände haben, wie z. B. Balkone, Windschirme, Relinge, Geländer, Balustraden und verspannte Netztücher, wurden hochaufgelöste Strömungsfelder für

charakteristische Betriebszustände berechnet. Aus diesen wurden geeignete Ersatzmodelle abgeleitet, die eine deutlich größere geometrische Struktur besitzen, aber ähnliche Strömungseigenschaften aufweisen wie die Originalgeometrien.

Mit Hilfe der erarbeiteten und getesteten Algorithmen kann die MET Motoren- und Energietechnik GmbH komplexe Fragestellungen zur Aerodynamik moderner Kreuzfahrtschiffe wie z.B. das Schornsteindesign, die Ausbreitung von Abgasfahnen, die Belüftungssituation, die Windsituation auf offenen Decks sowie den Passagierkomfort in geschlossenen oder halb offenen Aquaparks auf streng physikalischer Grundlage beantworten. Weiter bietet uns das die Möglichkeit, die Wirkung von unterschiedlichen Designvorschlägen mit minimalem Aufwand zu analysieren und schnellstmöglichst eine optimale Variante zu finden. Renommierete Werften und Reedereien haben derartige Dienstleistungen bei uns bereits in Anspruch genommen und die positiven Ergebnisse erfolgreich auf zahlreichen Schiffsneubauten umgesetzt.

8 Literatur / Quellen

- [1] ITTC 2008 Resistance Committee Report: *Proceedings of the 25th international towing tank conference*, Fukuoka, Japan. E. Beispiel und S. Exempel. *Titel der V. Zeitschrift*, S. 111–123, 2012.
- [2] Isherwood, R.M.: *Wind resistance of merchant ships*, The Royal Institute of Naval Architects, Vol. 115, 1972, pp. 327-338.
- [3] Inoue, S., Ishibashi, Y.: *The effects of wind on the ship manoeuvrability (I)*, J. West- Japan Society of Naval Architects, No. 44, 1972, pp. 111-128.
- [4] Blendermann, W.: *Estimation of wind loads on ships in wind with a strong gradient*, OMAE, Vol.1-A, 1995, pp. 271-277.
- [5] Nimura, T., Fujiwara, T., Ueno, M., Nonaka, K.: *Wind loads and flow visualization around an oil tanker model*, J. The Visualization Society of Japan, Vol. 17, No. 2, 1997, pp. 217-220.
- [6] Leer-Andersen, M.; Hughes, P.H.: *Computations of wind loads on ships and offshore structures*, Dept. Naval Arch. & Offshore Eng., Danish Technical Univ..
- [7] Aage, C.; Hvid, S.L.; Hughes, P.H.; Leer-Andersen, M.: *Wind loads on ships and offshore structures estimated by CFD*, 8th Int. Conf. Behaviour of Offshore Structures BOSS'97, Delft, pp.237-251.
- [8] Hvid, S.L.; Leer-Andersen, M.; Hughes, P.H.: *Wind load prediction on offshore structures using CFD*, *Conf. Application of Fluid Dynamics in the Safe Design of Topsides and Superstructures*, Inst. of Marine Eng., London, 1997.
- [9] Reddy, K.R., Tooletto, R., Jones, K.R.W.: *Numerical simulation of ship airwake*, *Computers & Fluids*, Vol. 29, 2000, pp. 451-465.
- [10] El Moctar, O.M.; Gatchell, S.; Bertram, V.: *RANSE simulations for aerodynamic flows around ship superstructures*, 4th Num. Towing Tank Symp., Hamburg.
- [11] El Moctar, O.M.; Gatchell, S.; Bertram, V.: *Aerodynamische Strömungssimulation für Schiffe*, *Hansa* 138/9, 2001, pp.20-21.
- [12] El Moctar, O.M.; Bertram, V.: *Computation of viscous flow around fast ship superstructures*, 24th Symp. Naval Hydrodyn., Fukuoka.
- [13] Lindenau, O.; Bertram, V.; El Moctar, O.M.; Gatchell, S.: *Aerodynamic simulations for an SES employing virtual reality post-processing techniques*, 3rd High-Performance Marine Vehicles Conf. HIPER'02, Bergen, pp.289-296.
- [14] Schmode, D.; Bertram, V.: *Aerodynamic flow computations for a Superfast ferry*, 3rd High-Performance Marine Vehicles Conf. HIPER'02, Bergen, pp.345-354.
- [15] Bertram, V.; Schmode, D.; El Moctar, O.M.: *Modeling Appendages in RANSE Simulations of Aerodynamic Flows around Ship Superstructures*, 5th Num Towing Tank Symp., Pornichet.
- [16] Bertram, V.; Köhlmoos, A.: *Megayachten im Computer. Anspruchsvolle Simulationen für anspruchsvolle Schiffe*, *nonstop* 1/2012, 2012, S. 35-40.
- [17] Jans, B.: *Kreuzfahrt-Tourismus. Konzepte, Trends, Perspektiven. Zur Einführung. In: Kreuzfahrt-Tourismus – die Online- Publikation von www.kreuzfahrt-forschung.de*, heraus-

gegeben von Walter Freyer, Bernhard Jans,
Philipp Röder, Grafenschaft/ Dresden 2010.

- [18] <http://maritimematters.com/2013/03/royal-princess-pre-decked-part-one-top-decks/>
- [19] Norwegian Cruise Line Bildmaterial :
<http://www.agents.ncl.de>

Sitzung 4

Raum 111

Blended Learning

(Sitzungsleitung: Andreas Körner)

*Individuelle Einrichtung einer Online-Lernumgebung auf Basis eines Einstufungstests - Ein
Werkstattbericht*

Karin Landefeld

HAW Hamburg

*Mathematik-Kompetenzen überprüfen und fördern - Automatisiert Lehren und Lernen mit
STACK*

Michael Kallweit

Ruhr Universität Bochum

Mathematiktests zwischen Schule und Hochschule

Thomas Schramm

Hafen Uni Hamburg

Test ist nicht gleich online Test – Anforderungen und Umsetzungen von elektronischen Tests

Stefanie Winkler

TU Wien

Individuelle Einrichtung einer Online-Lernumgebung auf Basis eines Einstufungstests - Ein Werkstattbericht

Karin Landenfeld¹

¹Hochschule für Angewandte Wissenschaften Hamburg

karin.landenfeld@haw-hamburg.de

Die Online-Lernumgebung viaMINT [1] bietet den Studienanfängerinnen und Studienanfängern modulbasierte Online-Kurse zur Vorbereitung auf das Studium an. Die Studierenden können ihr Wissen in einem Online-Einstufungstest testen und eventuelle Lücken identifizieren. Basierend auf den Ergebnissen des Einstufungstests wird die Online-Lernumgebung individuell gestaltet, so dass der Studierende seine Wissenslücken zielgerichtet aufarbeiten kann. In diesem Beitrag wird der dazugehörige Einstufungstest mit seinem Aufbau, der Aufgabengestaltung und der technischen Umsetzung vorgestellt sowie verschiedene Ansätze und beeinflussende Faktoren dazu diskutiert.

1 Ausgangslage

Hochschulstatistiken, Studienganganalysen sowie Aussagen von Lehrenden und Lernenden belegen in der Studieneingangsphase in vielen Fällen Probleme in den Grundlagenfächern, die häufig durch Wissenslücken in den Schulkenntnissen der Mathematik hervorgerufen werden. Außerdem lässt sich bei den Studienanfängerinnen und Studienanfängern bedingt durch unterschiedliche Schulformen und Lehrpläne oder unterschiedlich lange Zeiträume bis zum Studienbeginn ein sehr unterschiedlicher Leistungsstand im Bereich der Mathematik beobachten. Ausreichende Vorkenntnisse im Bereich der Mathematik sind jedoch für den Studienerfolg insbesondere für Studiengänge in den MINT-Wissenschaften unumgänglich. Diese gilt es vor Studienbeginn individuell zu erkennen sowie passgenau und zielgerichtet aufzuarbeiten.

2 Zielsetzung

Die Erkennung der Wissenslücken ist die Basis für eine zielgerichtete Aufarbeitung der fehlenden Kenntnisse. Insbesondere für eine Online-Lernumgebung ist dieses eine äußerst wichtige Voraussetzung, damit die Lernmodule passend empfohlen und bearbeitet werden können. Der in diesem Beitrag dargestellte Eingangstest der Lernumgebung viaMINT verfolgt mehrere Ziele:

(1) Der Test soll verdeutlichen, welche schulischen Vorkenntnisse für einen erfolgreichen Studienstart benötigt werden. Dieses ist den Studienanfängerinnen und Studienanfängern vor Beginn des Studiums nicht immer bewusst.

(2) Er soll dem Studierenden anhand der Ergebnisse sichtbar darstellen, welche Kenntnisse vorhanden sind und welche noch fehlen.

(3) Er soll Empfehlungen aussprechen, welche Themen vor Beginn des Studiums von dem Studienanfänger/ der Studienanfängerin noch aufgefrischt werden sollten.

(4) Er soll zeitlich deutlich vor Studienbeginn durchführbar sein, damit die notwendige Zeit zur Auffrischung gegeben ist.



Abbildung 1. viaMINT-Konzept für das Fach Mathematik

3 Konzept und Umsetzung

Das Konzept gliedert sich in vier Schritte (vgl. Abbildung 1). Ein wichtiger erster Schritt für die Studienanfänger/innen ist die Transparenz, welche Kenntnisse zu Beginn des Studiums erwartet werden. Der zweite Schritt ist das Erkennen, welche Kenntnisse vorhanden sind und welche noch nicht ausreichen. Der dritte Schritt ist das Aufarbeiten der fehlenden Kenntnisse, welches in einem vierten Schritt mit einem Abschlusstest pro Themengebiet als Lernerfolgskontrolle überprüft werden kann. Die Abbildung 1 visualisiert dieses Konzept, welches ergänzt mit den

passgenauen Präsenzveranstaltungen zu einem Blended Learning Konzept wird.

Online-Tests können unterschieden werden in (1) „Vorkenntnistest“ - Notwendige Kenntnisse und Kompetenzen für ein Studium oder eine Berufsausbildung testen, (2) „Lernerfolgstest“ - Prüfen eines Lernergebnisses nach Absolvieren einer Lerneinheit, (3) „Interessenstest“ – Passung zu einem Studiengang prüfen. Die verschiedenen Tests mit ihren Möglichkeiten und Zielsetzungen werden detailliert in [2] diskutiert. Neben dem Eingangstest, der ein Vorkenntnistest ist, werden in der Lernumgebung viaMINT auch Lernerfolgstests eingesetzt, die in den einzelnen Lernmodulen als Abschlusstest für ein Thema verwendet werden.

Mit dem Online-Einstufungstest können die Studierenden in viaMINT zunächst ihr Wissen testen und Lücken identifizieren. Die Aufgaben des Eingangstest orientieren sich

- zum einen am Mindestanforderungskatalog Mathematik, den die COSH-Initiative in Baden-Württemberg als notwendig Basis-Schulkenntnisse bei Aufnahme eines Studiums erarbeitet hat [3] und
- zum anderen an den spezifischen Erfordernissen des Studiengangs des einzelnen Studierenden, welche durch die Studiengangsverantwortlichen spezifiziert werden.

Der Eingangstest wird nach dem Login des Studierenden in die Lernumgebung entsprechend den Erfordernissen seines Studiengangs passgenau aufgebaut. Der Test erfüllt durch die dargestellten Aufgaben eine Informationsfunktion erfüllen, d.h. die Anforderungen werden sichtbar (vgl. [4]). Der Eingangstest ist als diagnostischer Test konzipiert, der dem Studierenden Rückmeldungen zu seinem individuellen Leistungsstand gibt. Für jeden Themenblock im Test werden die Aufgaben entsprechend der Lernziele gestaltet und mit aufsteigendem Schwierigkeitsgrad angeordnet. Basierend auf den Testergebnissen wird dem Studierenden eine individuelle Empfehlung zur Bearbeitung einzelner Lernmodule gegeben, damit seine Wissenslücken in den Themengebieten zielgerichtet aufgearbeitet werden können. Die Online-Lernumgebung baut sich entsprechend der gegebenen Empfehlung auf.

Der Online-Einstufungstest wird in viaMINT auf Basis des Testsystems der Lernplattform Moodle realisiert. Über das Moodle-Plugin STACK können

Aufgaben umgesetzt werden, die zur Auswertung numerischer Eingaben das Computer-Algebra-System MAXIMA nutzen. Hierdurch können komplexe Aufgaben mit vielfältigen Aufgabentypen, einer Bewertung von Zwischenergebnissen und einem detaillierten Feedback realisiert werden.

4 Zusammenfassung und Ausblick

Der Eingangstest wird an der HAW Hamburg aktuell durch die Studienanfängerinnen und Studienanfänger evaluiert. Für weitere Fächer z.B. Physik wird an der Erstellung eines Eingangstests gearbeitet.

Die Entwicklung der Online-Lernumgebung viaMINT ist ein Teilprojekt des hochschulweiten Projekts „Lehre lotsen – Dialogorientierte Qualitätsentwicklung für Lehre und Studium“ der HAW Hamburg. Dieses Vorhaben wird aus Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01PL11046 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.



5 References

- [1] K. Landefeld, M. Göbbels, A. Hintze, J. Priebe. *viaMINT – Aufbau einer Online-Lernumgebung für videobasierte interaktive MINT-Vorkurse*. Zeitschrift für Hochschulentwicklung (ZFHE), Ausgabe 9/5 "Übergang Schule-Hochschule", Dezember 2014
- [2] K. Winter. *Diagnostisches Potential von Online-Self-Assessments – Möglichkeiten und Umsetzung*. Beiträge zum Mathematikunterricht 2013 Digital, Vorträge auf der 47. Tagung für Didaktik der Mathematik, Jahrestagung der GDM vom 4.3.2013 bis 8.3.2013 in Münster
- [3] COSH Cooperation Schule-Hochschule. *Mindestanforderungskatalog Mathematik (Version 2.0) der Hochschulen Baden-Württembergs für ein Studium von WiMINT-Fächern*, Stand 23.7.2014; http://www.mathematik-schule-hochschule.de/images/Aktuelles/pdf/MAKatalog_2_0.pdf
- [4] Ch. Neugebauer, K. Winter. *Fehleranalysen bei Studienanfängerinnen als Basis zur individuellen Förderung in Mathematik*. Münster, <https://eldorado.tu-dortmund.de/bitstream/2003/33271/1/BzMU14-4ES-Neugebauer-197.pdf>, Stand 20.5.2015

Mathematik-Kompetenzen überprüfen und fördern - Automatisiert Lehren und Lernen mit STACK

Michael Kallweit

Ruhr-Universität Bochum

michael.kallweit@rub.de

STACK ist ein quelloffenes computergestütztes Assessmentssystem für mathematische Fragestellungen. Durch Verbindung mit einem Computeralgebrasystem können (Freitext-)Eingaben automatisch auf mathematische Eigenschaften getestet, Aufgaben ausgewertet und Bewertungen vorgenommen werden. Mit der Integration in die Lernplattform Moodle steht Lernenden und Lehrenden damit eine moderne und freie eLearning-Lösung für die Mathematik zur Verfügung. Diese wird an der Ruhr-Universität Bochum in verschiedenen Veranstaltungen und Lehrprojekten fruchtbar eingesetzt.

Einleitung

Der Einsatz von eLearning ist vielfach mit der Hoffnung verknüpft, Lehre trotz knapper Ressourcen modern und innovativ zu ergänzen und im Idealfall sogar zu entlasten. Neben didaktisch und medial aufbereitetem Lernmaterial sind es oft Routine-Übungsaufgaben, deren Bearbeitung und Überprüfung man in elektronische Systeme auslagert. Der sicherlich häufigste Typ zur automatischen Auswertung ist die MultipleChoice-Aufgabe.

MultipleChoice

Aufgaben vom MultipleChoice-Typ bestehen durch ihre Einfachheit bei der Korrektur, welche sehr leicht von automatischen Systemen erledigt werden kann. Deren Einsatz ist aber nicht unproblematisch. So ist z.B. die Frage der konkreten Bepunktungsregel für mehrere richtig und falsch angekreuzte Antworten nicht eindeutig zu beantworten. Noch viel schwieriger ist auch die Erstellung sinnvoller Distraktoren. Diese müssen mit sehr viel Sorgfalt entworfen werden, um z.B. typische Fehler und Fehlvorstellungen der Nutzer abzufangen bzw. aufzugreifen. Es sollte verhindert werden, dass die Aufgabe nicht ausschließlich durch gute „Ratetechniken“ zu lösen ist.

Insbesondere der Einsatz in der Mathematik zeigt weitere Probleme auf. Wie [1] beispielsweise ausführt, sorgen reversible Prozesse (wie sie in der Mathematik zuhauf vorkommen) dafür, dass intendierte Aufgabenstellungen überhaupt nicht erreicht werden können. Ein Beispiel ist in Abbildung 1 gegeben: Anstatt Kenntnisse über die Bildung von Stammfunktionen anzuwenden, kann die Lösung der Aufgabe auch einfach durch Ableitung der Antwortmöglichkeiten gefunden werden.

Bestimmen Sie das folgende unbestimmte Integral: $\int (x-7)^3 dx$

- | | |
|--|--|
| <input type="checkbox"/> $(x-7)^4$ | <input type="checkbox"/> $(x-7)^2$ |
| <input type="checkbox"/> $4 \cdot (x-7)^4$ | <input type="checkbox"/> $3 \cdot (x-7)^2$ |
| <input type="checkbox"/> $\frac{1}{4} \cdot (x-7)^4$ | <input type="checkbox"/> $\frac{1}{3} \cdot (x-7)^2$ |

Abbildung 1: Beispiel MC-Aufgabe

Generelle Bedenken, die auf geschlechtsbezogenen Verzerrungseffekten beruhen, trüben ebenfalls den Einsatz von MultipleChoice. So zeigen mehrere Untersuchungen, dass in dem Format selbst ein Gender-Bias inne wohnt [2], [3].

Freitexteingabe

Ein möglicher Ausweg sind Aufgaben mit freier Eingabe der Lösung. Nahezu jedes eLearning-System bietet solche Formate, doch meist wird nur der exakte Abgleich mit einer hinterlegten Musterlösung zur Auswertung vorgenommen. Für mathematische Fragestellungen ist es zudem oft möglich kleine arithmetische Rechnungen mit numerischen Toleranzbedingungen zu definieren. Für tiefere Mathematik ist man oft auf die Hilfe von menschlichen Korrektoren angewiesen. Der Grund ist offensichtlich: mathematische Objekte können nahezu immer mit unendlich vielen (Text-)Ausdrücken beschrieben werden. Ein System, welches mathematisch sinnvoll auswerten und Bewertungen vornehmen soll, muss also in der Lage sein, die Eingaben in semantisch sinnvolle Repräsentationen zu übersetzen und mit diesen weiter zu arbeiten. Im Laufe der Zeit sind einige solcher Systeme entstanden, die speziell für das automatische Assessment von Mathematik entwickelt wurden. Eine Einführung und Überblick über vorhandene Software findet man in [1].

STACK

Das Assessmentsystem STACK [4], ursprünglich an der Universität Birmingham ab 2004 von Chris Sangwin entwickelt, dehnt die Möglichkeiten des Testens entschieden aus. Es wird ein offener Aufgabentyp für Mathematik und Mathematik-nahe Aufgaben bereitgestellt. Es sind Freitext-Eingaben möglich, die dennoch automatisch und mathematisch sinnvoll ausgewertet werden können. Durch Rückgriff auf das freie Computeralgebrasystem Maxima [5] lassen sich die Eingaben der Nutzer nicht nur mit gespeicherten Antwortmöglichkeiten vergleichen, sondern auch auf mathematische Eigenschaften testen, was umfangreichere Überprüfungs- und Feedback-Möglichkeiten schafft.

Die einfachste so realisierbare Auswertung ist der direkte Vergleich mit einer hinterlegten Musterlösung. Hierbei kann auf die im Computeralgebrasystem (CAS) vorhandenen Funktionen zur Vereinfachung und Reduzierung symbolischer Ausdrücke zurückgegriffen werden, siehe Abbildung 2. Mit diesem Verfahren lassen sich bereits die vielfältigen verschie-

denen Eingaben als mathematisch äquivalent und damit ggf. als richtig erkennen.

```

IF simplify(input - solution) = 0
  THEN mark := 1
  ELSE mark := 0
    
```

Abbildung 2: Auswertung durch Vergleich mit Lösung

Die Möglichkeiten von STACK gehen jedoch weit über dies hinaus. So lässt sich hier eine Aufgabe durch die Angabe eines *Rückmeldebaums* gestalten. Dieser ist ein gerichteter azyklischer Graph, an dessen Knoten die Eingaben des Nutzers mittels verschiedener Tests überprüft werden können und je nach Ergebnis mit nächsten Knoten fortgefahren wird. Ähnlich zu einem Flußdiagramm mit Entscheidungen wird so der Rückmeldebaum von der Wurzel bis zu einem der Blätter durchlaufen. Diese Feedbackmodellierung ermöglicht es die Eingabe nacheinander auf verschiedene mathematische Eigenschaften zu testen und entsprechende Rückmeldungen zurückzuliefern. Ein Beispiel ist in Abbildung 3 gegeben.

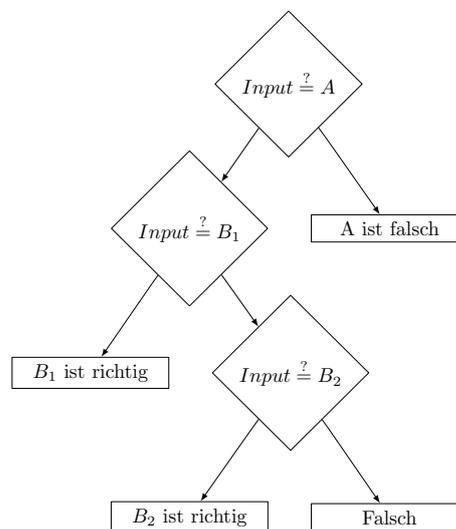


Abbildung 3: Rückmeldebaum

Die möglichen Tests an den Knoten sind vielfältig und können z.B. den (Daten-)Typ oder die mathematische Form der Eingabe überprüfen; eine Auflistung ist in Abbildung 4 gegeben. Den automatischen Rückmeldungen sind dabei in ihrer Komplexität keine Grenzen gesetzt.

Equality	CasEqual, EqualComAss, AlgEquiv, SubstEquiv, SameType, SysEquiv
Form	LowestTerms, Expanded, FacForm, SingleFrac, PartFrac, CompletedSquare
Factorisation of polynomials	
Precision	NumRelative, NumAbsolute, NumSigFigs, GT, GTE
Calculus	Diff, Int
Other	String, StringSloppy, RegExp

Abbildung 4: Mögliche Tests

Möglich werden z.B. offene Modellierungsaufgaben, in denen zu einem Problem eine Gleichung aufgestellt werden muss, oder Aufgaben, in denen der Nutzer gefordert ist, ein Beispiel für einen mathematischen Sachverhalt einzugeben.

Bei der Konstruktion der Aufgaben kann an nahezu jeder Stelle auf die Funktionen des CAS zurückgegriffen werden, so ist es u.a. möglich neue Variablen mit Zwischenergebnissen oder zufälligen Werten anzulegen. Damit lassen sich randomisierte Aufgaben erstellen, die dennoch didaktisch äquivalent sind. In Abbildung 5 ist eine Zusammenfassung der Bestandteile einer STACK-Aufgabe dargestellt.

Verbindung zu Moodle

STACK ist eine Open Source Software und steht als Plugin für das Learning-Management-System Moodle zur Verfügung [6]. Als eigenständiger Fragetyp ergänzt es die vorhandenen Aufgabenformate. Ein wesentlicher Vorteil dieses Gespanns liegt in ihrer Fähigkeit \LaTeX -Ausdrücke (mittels MathJax) an nahezu jeder Stelle anzuzeigen.

Einsatz an der RUB

Das Tandem Moodle und STACK wird seit einigen Jahren an der Ruhr-Universität Bochum eingesetzt. Zuletzt wurde 2014 bei dem eLearning-Wettbewerb

Aufgabenvariablen	Maxima
Aufgabentext	CASText
Generelles Feedback	CASText
Eingaben	Algebraische Eingabe, Einzelne Zeichen, Matrix, Textfeld, Wahr/Falsch
Feedback-Graph	Knoten mit Antworttests und spezifischem Feedback (CASText), sowie Kanten
Optionen	Optionen und CAS-Text
Testfälle	Testeingaben

Abbildung 5: Anatomie einer STACK-Aufgabe

5x5000 das Projekt „Aufbau einer freien Sammlung automatisch auswertbarer Mathematikaufgaben“ als einer der Gewinner ausgezeichnet. Dort werden Aufgaben entwickelt, die die besonderen Feedbackmöglichkeiten von STACK ausschöpfen, um den Studierenden ein besseres Lernerlebnis zu bieten. In Kooperation mit dem etablierten Helpdesk Mathematik des *Servicezentrum Mathematik und Anwendungen* [7], sollen diese Aufgaben im Rahmen der Professionalisierung in einem Aufgabenpool des Moodle-Systems der RUB für alle Veranstaltungen frei nutzbar sein und das Präsenz-Angebot auch unabhängig von dessen Öffnungszeiten ergänzen.

Die Ergebnisse fließen auch in Unterstützungsprogramme für Erstsemester wie MathePlus [8] ein. Dort soll der oftmals problematische Übergang von Schule zu Hochschule abgemildert und unnötiger Studienabbruch verhindert werden. Die fachliche Komponente wird in Zukunft intensiv vom STACK-System Gebrauch machen.

Folgende Vorteile haben sich u.a. herauskristallisiert:

unmittelbareres Feedback: Regelmäßige wöchentliche Übungsaufgaben sind Bestandteil jeder regulären Mathematik-Veranstaltung. Durch die vorgegebene Bearbeitungszeit und die Korrekturzeit, liegt die Rückgabe zu meist mehrere Wochen nach Behandlung der Inhalte in der Vorlesung. Durch automatisch ausgewertete Aufgaben

kann ein viel direkteres Feedback zum aktuellen Vorlesungsstoff und dem eigenen Verständnis gegeben werden.

neue Arten der Kooperation Die Möglichkeit neben gleichen Aufgaben für alle Studierenden, auch individuelle (aber dennoch vergleichbare) Aufgaben zur Bearbeitung zu veröffentlichen, fördert eine neue Art der Kooperation. Studierende können zwar über eine Aufgabenart reden, aber nicht mehr konkret Lösungen austauschen. Es muss dabei jedes mal noch eine Transferleistung erbracht werden.

mehr Übungsmöglichkeiten Durch die Randomisierung von Aufgaben können den Hörern gleich „unendlich“ viele Aufgaben zum weiteren Üben (außerhalb der abzugebenden Übungsaufgaben) zur Verfügung gestellt werden.

Inzwischen ist STACK bei vielen weiteren Hochschulen im Einsatz und es etablieren sich erste Nutzernetzwerke. Portierungen auf andere eLearning-Systeme sind im Gange, so dass die Verbreitung weiter steigen wird.

Literatur

- [1] C. J. Sangwin, *Computer Aided Assessment of Mathematics*, Oxford University Press 2013
- [2] P. Hassmén & D. P. Hunt, *Human self-assessment in multiple choice*, Journal of Educational Measurement 31(2), S. 149-60, 1994
- [3] G. C. Leder, G. Rowley & C. Brew, *Gender differences in mathematics achievement: Here today and gone tomorrow?*, International Comparisons in Mathematics: The State of the Art, S. 213-24, Falmer Press, 1999
- [4] C. J. Sangwin. Computer Aided Assessment of Mathematics Using STACK. In Proceedings of ICME 12, 2012.
- [5] <http://maxima.sourceforge.net>
- [6] https://github.com/mathsmoodle-qtype_stack
- [7] J. Buchsteiner & M. Kallweit, *Professionalisierung des Helpdesk Mathematik*, in H. Linneweber-Lammerskitten (Hrsg.), Beiträge zum Mathematikunterricht 2015. Münster: WTM.
- [8] H. Dehling, E. Glasmachers, B. Griese, J. Härterich, & M. Kallweit, *MP2 - Mathe/Plus/Praxis - Strategien zur Vorbeugung gegen Studienabbruch*, Zeitschrift für Hochschulentwicklung 9, S. 39-55, 2014.

Mathematiktests zwischen Schule und Hochschule

Thomas Schramm

Geomathematik, -informatik & Physik, HafenCity Universität Hamburg

tom.schramm@hcu-hamburg.de

An vielen Hochschulen wird an der Problematik gearbeitet, den Übergang von der Schule zu erleichtern oder überhaupt erst zu ermöglichen. Im WiMINT-Bereich ist Mathematik DAS Problem. Neben vielen Einzelaktivitäten entstehen konzertierte, landesweite Aktionen, die mit einem gemeinsamen inhaltlichen Anforderungsprofil Online-Brückenkurse anbieten. Innerhalb dieser Kurse ist ein formatives Assessment obligatorisch, vorgeschaltet wird aber ein Einstufungstest, der weitere Verwendung finden kann. Studiengangsspezifischer sind sogenannte Navigatoren, die ebenfalls Selbst-Assessments enthalten, aber zu ganz anderen Aussagen führen. Berichtet wird über die Schwierigkeiten der Erstellung und der Evaluation solcher Test und die Problematik ihrer Aussagekraft.

Test ist nicht gleich online Test – Anforderungen und Umsetzungen von elektronischen Tests

Stefanie Winkler¹, Andreas Körner¹, Tamara Vobruba¹

¹Technische Universität Wien

stefanie.winkler@tuwien.ac.at

Werden bestehende Lehrveranstaltung mit einem online Tool auszustatten wird häufig der Fehler gemacht, einen elektronischen Test im selben Stil gestalten zu wollen, wie einen klassischen Test. Im Folgenden soll die Entstehung und das Einsatzgebiet von online Tests in zwei konkreten Anwendungen vorgestellt werden. An der TU Wien wird zu Studienbeginn ein Auffrischkurs Mathematik für Studierende angeboten, welcher u.a. zwei Arten von elektronischen Tests einsetzt, einen anonymen und einen bewertenden Test. Des Weiteren werden in den Mathematik für Elektrotechnik Veranstaltungen Hausübungen, Klausuren und Prüfungen ebenfalls elektronisch abgebildet.

1 Einführung

Klassische Ingenieursdisziplinen waren immer bekannt für herausragende Rechenfertigkeiten. In den letzten Jahren hat sich dieses Bild aber gewandelt. An der TU Wien kamen vermehrt Rückmeldungen der Ingenieursfakultäten, dass die Rechenfertigkeiten der Studierenden zunehmend schlechter wurden. Diese Beobachtungen beruhen auf den Leistungen der Studierenden in mündlichen Prüfungssituationen oder Übungen in technischen Disziplinen. Auch Rückmeldungen von Studierenden zeigten, dass Ihnen in den technischen Disziplinen das mathematische Handwerkszeug fehlt um den technischen Sachverhalt, insbesondere zu Studienbeginn, zu verinnerlichen. Diese Erkenntnisse waren Ausgangspunkt zu den Überlegungen des Projekts Auffrischungs- und Grundkurs Mathematik (AK- & GKMATH) an der TU Wien. Diese Kurse setzen ein E-Learning System ein, welches ein modifiziertes Vorlesungs- und Übungskonzept erfordert.

2 Projektstruktur

Der Auffrischkurs beschäftigt sich hauptsächlich mit der Festigung von Fertigkeiten. Dieser Kurs

findet zu Beginn des Studiums im ersten Semester statt. Es werden die wichtigen, für die Universität notwendigen, Mathematikkenntnisse aus der Schule wiederholt. Da der Kurs nur der Auffrischung verschütteter Kenntnisse dient um den Einstieg in das technische Studium zu erleichtern dauert der Kurs nur 1-2 Wochen. Dadurch wird auch der übliche Semesterstart nicht zu stark beeinflusst. Der Kursaufbau besteht aus 6-8 Modulen mit je einer Vorlesung und einer Übung. Durch die zeitliche Limitation ist die online Unterstützung wichtig um Studierenden die Möglichkeit zu geben, individuell den erlernten Stoff zu verarbeiten. Der Grundkurs Mathematik richtet sich an die Studierenden der Elektrotechnik in den ersten 2 Semestern. Diese Lehrveranstaltungen bestehen aus einer traditionellen Vorlesung mit Tafel, Schwamm und Kreide sowie einer Übung. Der Übungsmodus hat sich nach Einführung des E-Learning Systems modifiziert um die Möglichkeiten besser ausschöpfen zu können. Die Übung beinhaltet im Prinzip drei Elemente: Übungsbeispiele, Hausübung und Klausuren. Die Übungsbeispiele umfassen wöchentlich 8-10 schriftliche Aufgaben, welche von den Studenten vorbereitet werden. In kleinen Übungsgruppen werden diese dann unter Leitung eines Tutors präsentiert. In dieser Präsenzübung wird vor allem auf den mathematischen Hintergrund und die zugrundeliegende Theorie wert

gelegt. Drei über das Semester verteilte Klausuren fokussieren vor allem auf die Überprüfung der praktischen Rechenfertigkeiten. Um die Motivation für die Vorbereitung der Klausuren zu erhöhen werden freiwillige Hausübungen kurz vor den Klausuren zur Verfügung gestellt.

3 E-Learning System

Das verwendete E-Learning System basiert auf dem Computer Algebra System Maple. Es stellt nicht nur eine Beispielsammlung für Studierende dar, sondern unterstützt die Studierenden beim Lernprozess. Alle vorhandenen Beispiele sind in Module und diese in Untergruppen aufgeteilt, so dass man gezielt Beispiele auswählen kann die man üben will. Diese Struktur ist in Moodle verlinkt verfügbar. Das System bietet verschiedenste Beispielstrukturen wie Multiple Choice oder drop-down Formate um die Beispiele zu den Stoffgebieten didaktisch sinnvoll aufzubereiten. Einer der Vorteile von Maple T.A. ist die Möglichkeit Beispiele zu entwickeln, die sich bei jedem Aufruf ändern. Die Beispielstruktur bleibt die selbe allerdings ändern sich die verwendeten Variablen und Funktionen. Daher ist es nicht sinnvoll die Übungsbeispiele gleich den schriftlichen Aufgaben, als statisches Beispiel, zu konzipieren. Zusätzlich zu der randomisierten Struktur der Beispiele kommen aber auch die unterschiedlichen Fragetypen hinzu. Ein weiterer Vorteil des Systems liegt in der Verknüpfung von Maple T.A. mit dem Moodle basierten Administrationssystem der Universität. Die Ergebnisse der abgelegten Prüfungen und Klausuren können von Studierenden jederzeit eingesehen werden. Bei jedem Beispiel können die Studierenden sofort nach Eingabe ihre Resultate auf Richtigkeit überprüfen. Das System liefert umgehend Feedback, siehe [1]. Zusätzlich besteht die Möglichkeit in einem Modus zu üben, in dem man sich einen möglichen Lösungsweg des ausgewählten Beispiels anzeigen lassen kann um es beim nächsten Versuch eigenständig lösen zu können.

In den ersten Jahren waren die Rückmeldungen der Studierenden teilweise negativ, da bei schwierigeren Aufgabenstellungen die Bewertung von Maple T.A. noch nicht optimal war. Vor allem die Themen Lineare Algebra und Partialbruchzerlegung waren betroffen. Lange Rechenwege mit kleinen Fehlern resul-

tierten in einer teilweise falschen Lösung und wurden damit vom System mit Null Punkten bewertet. Da dies nicht im Sinn der Studierenden war, wurden zwei Bibliothekdateien entwickelt die einerseits die Bewertung für Studierende ansprechender machten, andererseits aber auch die Arbeit für Entwickler erleichterten, [2]. Diese Routinen verbesserten die Haltung der Studierenden nachweisbar.

Eine sehr nützliche Eigenschaft ist die Erstellung unterschiedlicher Assignments. Für Hausübungen gibt es die Möglichkeit, die Resultate der Studierenden aufzuzeichnen, ihnen aber gleichzeitig beliebig viel Zeit für das Lösen zu geben. Andererseits ist auch ein anonymes Üben ohne Aufzeichnung möglich. Am Wichtigsten hat sich aber der Prüfungsmodus erwiesen. Dieser garantiert, dass die Studierenden nur unter Anwesenheit eines Zeichnungsberechtigten die Prüfung ablegen können.

4 Zusammenfassung

Die Einführung des E-Learning Systems eröffnet neue Möglichkeiten. Es ist nicht nur möglich praktische Fertigkeiten abzutesten sondern auch theoretische Fragestellungen aufzubereiten. Der Vorteil des online Tools ist auch in der Abwicklung der Vorlesungsprüfung einsetzbar. Die Zusammenstellung der Prüfung kann unter Umständen etwas langwieriger ausfallen dafür besteht der Vorteil darin, den Zeitaufwand der Bewertung gering zu halten. Dies geschieht gänzlich automatisch.

References

- [1] Körner, A., Winkler, S., Urbonaite, V. *E-Learning Einsatz im Auffrischkurs Mathematik an der TU Wien*. Proceedings of the 4th International eLBA Science Conference. S. 145-154. 2011.
- [2] A. Zimmermann, V. Urbonaite, A. Körner, S. Winkler, S. Krause, M Kleinert. *Advanced Randomization and Grading in the E-Learning System Maple T.A.*. Proceedings of the 7th Congress on Modelling and Simulation. ISBN: 978-80-01-04589-3; S. 1209 - 1214. 2010.

Sitzung 5

Raum 104

Physical Modeling / Modelica

(Sitzungsleitung: Heinz-Theo Mammen)

Automatisierte, parallele Simulation von Modelica-Modellen am Beispiel des OpenModelica-Compilers

Volker Waurich

TU Dresden

Strukturvariabilität für Modelica-Modelle

Daniel Gomez Esperon

TU Berlin

Simulating a simple pneumatics network using the Modelica Fluid library

Peter Junglas

PHWT Vechta

Modellierung elektrischer Antriebe mit Modelica

Leo Gall

LTX Simulation GmbH

Automatisierte, parallele Simulation von Modelica-Modellen am Beispiel des OpenModelica-Compilers

Volker Waurich¹, Ines Gubsch¹, Marcus Walther¹, Michael Klöppel¹

¹Technische Universität Dresden

volker.waurich@tu-dresden.de

Gleichungsbasierte und objektorientierte Modellierungssprachen wie Modelica ermöglichen eine effektive Modellbeschreibung für die Systemsimulation. Die Formulierung physikalischer Sachverhalte ist aufgrund der akasalen Modellierungsweise von der Systemstruktur losgelöst, was Übersichtlichkeit und Austauschbarkeit gewährleistet. Die anschließende Überführung der Modellbeschreibung in ein mathematisch lösbares Modell erfolgt vollautomatisch durch einen Modelica-Compiler. Um eine beschleunigte Simulation zu erreichen kann der Compiler so erweitert werden, dass ein parallel ausführbares Simulationsprogramm erstellt wird und die Programmausführung auf mehreren Prozessoren stattfindet. Der open-source OpenModelica Compiler ist ein quelloffener, leistungsstarker Modelica-Compiler, der es ermöglicht verschiedene Optimierungs- und Parallelisierungsstrategien zu implementieren und zu testen. Diese Arbeit erläutert die Prinzipien zur Übersetzung eines gleichungsbasierten Modelica-Modells in ein parallel ausführbares Simulationsprogramm. Dabei wurden domänenspezifische Besonderheiten festgestellt, welche kurz begründet werden. Die erzielten Verbesserungen werden basieren auf der Implementierung im OpenModelica-Compiler dargestellt.

1 Einleitung

Die Simulation technischer und physikalischer Systeme aus verschiedenen Domänen ist eine komplexe Aufgabe, die Erfahrung, Zeit und physikalischen sowie mathematischen Sachverstand erfordert. Mit Hilfe von objektorientierten, gleichungsbasierten Modellierungssprachen lässt sich die Simulationsaufgabe weitestgehend auf die Beschreibung des Systems reduzieren. Der Modellierer kann sich auf die Abstraktion der physikalischen Sachverhalte konzentrieren und überlässt die Übersetzung des Modells in ein mathematisch lösbares differential-algebraisches Gleichungssystem (DAE-System) einem Compiler. Die Einbettung in eine Laufzeitumgebung bestehend aus Zeitintegrator und Eventbehandlung kann ebenso vom Compiler übernommen werden. Die effiziente programmtechnische Umsetzung wird somit größtenteils automatisch vom Compiler realisiert. Da zur Compilezeit die algebraischen Beziehungen bekannt sind, können die Gleichungen gezielt manipuliert werden, um eine schnelle Simulation zu ermöglichen. Neben grundlegenden symbolischen Umformungen der Modellgleichungen und performanten numerischen Algorithmen, kann die Beschleunigung der Simulation durch Parallelisierung erreicht werden. Aus der Literatur sind verschiedene Ansätze bekannt, um eine Simulation zu parallelisieren. Abgesehen von anderen Parallelisierungsmöglichkeiten wie paralleler Zeitintegration und der Parallelisierung von entkop-

pelten Submodellen, wird in der vorliegenden Arbeit die automatische Parallelisierung der Modellgleichungen erläutert. Für diese Parallelisierungsstrategie müssen die Abhängigkeiten der Modellgleichungen untersucht werden. Eine hinreichend gute Aufteilung des Modells in parallel-ausführbare Aufgaben wird anschließend durch den „Scheduling“-Algorithmus getroffen. Eine parallele Ausführung ist jedoch nicht immer möglich oder sinnvoll, was wiederum physikalisch begründbar ist.

In dieser Arbeit wird in Kapitel 2 zunächst die Modellierungssprache Modelica vorgestellt. Kapitel 3 umreist den Übersetzungsprozess im OpenModelica Compiler vom Modell zum parallelen Simulationsprogramm. Kapitel 4 behandelt Techniken zur Verbesserung der parallelen Simulation und Kapitel 5 zeigt ausgewählte Ergebnisse der Parallelisierung. In Kapitel 6 wird ein abschließendes Fazit gezogen.

2 Modelica als Beschreibungssprache

Modelica ist bereits ein weit genutzter Sprachstandard zur Beschreibung physikalischer, technischer und anderer Systeme. Die Sprache ist objektorientiert und ermöglicht eine hierarchische Strukturierung der Modellelemente. Modelica erlaubt es zeitkontinuierliche und zeitdiskrete Prozesse abzubilden. Typischerweise ergibt sich aus den Modellgleichungen folgendes differential-algebraisches Gleichungssystem (DAE-System):

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

Dabei stellen $x(t)$ die Zustandsgrößen und $\dot{x}(t)$ die Zustandsableitungen dar. $v(t)$ umfasst alle anderen zeitkontinuierlichen, algebraischen Variablen und $q(t)$ alle anderen zeitdiskreten Variablen. Das Formelzeichen p beinhaltet alle Parameter und Konstanten die nicht von der Zeit t abhängig sind. Modelica ist im Gegensatz zu signalflussbasierten Modellierungsansätzen gleichungsbasiert. Das bedeutet, dass es eine ungerichtete Zuordnung zwischen den Variablen gibt und eine symbolische Umformung durch einen Compiler erfolgt. Ein Umstellen von Gleichungen durch den Modellierer ist somit nicht notwendig. Die Modellvariablen und Gleichungen werden im Modelica-Quelltext in die vorgesehenen Bereiche geschrieben. **Code 1** zeigt ein Modelica Codebeispiel mit zwei Variablen, einem Parameter und zwei Gleichungen.

```

1 model example
2   parameter Real p = 1.0;
3   Real x;
4   Real y;
5 equation
6   x = sin(p*time);
7   der(y) = x;
8 end example;
9

```

Code 1 Modelica-Model

Die Wirkrichtung der Komponenten kann sich aufgrund der akasalen Formulierung je nach Anwendungsfall umkehren. Das Modell eines Generators kann somit auch als Elektromotor fungieren. Die Verbindung der Modellelemente erfolgt über Konnektoren. Es bietet sich daher an, Modellelemente übersichtlich auf einer grafischen Benutzeroberfläche darzustellen. **Abbildung 1** zeigt eine solche grafische Darstellung am Beispiel eines elektrischen Schaltkreises.

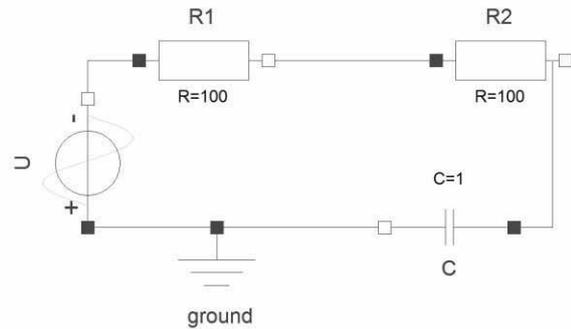


Abbildung 1 grafische Repräsentation eines Modelica-Modells

Modelica ist nicht auf einzelne Domänen beschränkt, sondern kann auf diverse Probleme angewandt werden, die sich mathematisch formulieren lassen. Demzufolge existiert bereits eine Vielzahl an Modelica-Bibliotheken, die Modellelemente verschiedenster Domänen, wie beispielsweise Mechanik, Thermodynamik, Elektrotechnik... umfassen. Diese Bibliotheken sind sowohl kommerziell als auch kostenlos verfügbar.

3 Modellübersetzung im OpenModelica Compiler

Modelica-Modelle stellen lediglich eine Beschreibung eines Systems dar. Zum Erstellen eines lauffähigen Simulationsprogramms benötigt es einen Compiler, von denen es kommerzielle als auch kostenfreie gibt. Für die vorliegende Arbeit wurde der OpenModelica-Compiler genutzt. Dieser ist quelloffen und eignet sich daher für den wissenschaftlichen Einsatz zur Entwicklung neuer Methoden. Der OpenModelica Compiler wird seit Jahren kontinuierlich von Wissenschaftlern und Industrieunternehmen weiterentwickelt. Die Arbeitsweise des OpenModelica Compilers gestaltet sich wie in **Abbildung 2** dargestellt:

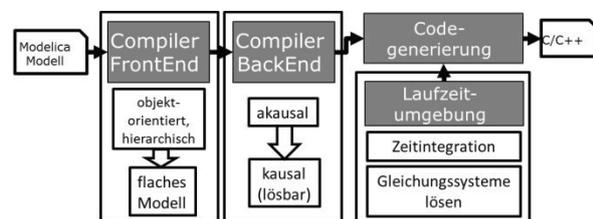


Abbildung 2 Arbeitsweise des OpenModelica Compilers

Im Compiler Frontend wird die hierarchische, objekt-orientierte Struktur des Modells in ein „flaches“ Modell überführt. Dieses umfasst sämtliche Gleichungen und Variablen. Im Backend werden die Abhängigkeiten zwischen Gleichungen und Variablen untersucht,

Optimierungen durchgeführt und eine Berechnungsreihenfolge aufgestellt. Bei DAE-Systemen mit höherem Index, wird zunächst eine Indexreduktion durchgeführt. Jeder Gleichung wird eine zu lösende Variable zugewiesen. Diesen Vorgang nennt man Kausalisierung. Gegebenenfalls werden algebraische Schleifen identifiziert, die später als Gleichungssystem geschlossen gelöst werden müssen. Aus dem kausalisierten Modell wird C bzw. C++ Code generiert und mit der bereitgestellten Laufzeitumgebung kann ein lauffähiges Simulationsprogramm kompiliert werden.

3.1 Das Compiler Backend

Die Algorithmen, die aus einem akasalen Modell eine geordnete Liste kausaler Gleichungen erzeugen, sind im Compiler Backend zu finden. Ausgangspunkt hierfür sind eine Liste von Gleichungen und eine Liste von Variablen, aus denen Abhängigkeitsbeziehungen hergeleitet werden können. Die Gleichungen und Variablen für das Schaltkreismodell aus **Abbildung 1** sind folgende:

```

1 U.v=U.a*sin(2*pi*U.f*time)
2 R1.v=R1.R_actual*U.i
3 R1.LossPower=R1.v*U.i
4 R1.v = -U.v-R1.n.v
5 R2.v = R2.R_actual*U.i
6 R2.LossPower = R2.v*U.i
7 R2.v = R1.n.v - C.v
8 U.i = C.C * der(C.v)
    
```

Listing 2 Modellgleichungen des Schaltkreises

Es ist anzumerken, dass zu diesem Zeitpunkt bereits triviale Gleichungen ($a = b$, $a = \text{const.}$) entfernt wurden. Die Algorithmen zur Kausalisierung sind graphenbasiert. Daher wurde zur Darstellung der Modellgleichungen und Variablen in **Abbildung 3** ein ungerichteter, bipartiter Graph genutzt. Hier sind die Variablen mit weißen Kreisen und die Gleichungen mit grauen Quadraten gekennzeichnet. Ist eine Variable Teil einer Gleichung, so existiert eine Kante zwischen Gleichungs- und Variablenknoten.

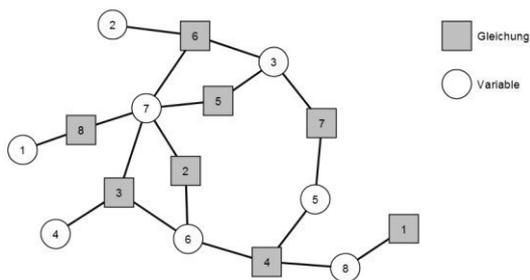


Abbildung 3 bipartiter Graph

Um für dieses Modell eine Berechnungsreihenfolge und eine Kausalisierung zu finden, müssen Gleichungen und Variablen einander zugeordnet werden. Dazu wird der Graph nach Gleichungsknoten durchsucht, die nur einen inzidenten Variablenknoten besitzen. In diesem Beispiel ist das Gleichung 1, die der Berechnung der Quellspannung $U.v$. entspricht wird. Somit kann Gleichung 1 der Variable 8 zugeordnet werden. Anschließend werden die zugeordneten Knoten aus dem Graphen entfernt und eine weitere Kausalisierung wird gesucht. Im vorliegenden Beispiel ist kein weiterer Gleichungsknoten kausalisierbar. Es gibt jedoch Variablenknoten, die eindeutig Gleichungsknoten zugeordnet werden können. In diesem Fall die Variable 1, 2 und 4 zu Gleichung 8, 6 und 3. Diese kausalisierten Gleichungen werden am Ende der Berechnung gelöst.

Es bleibt nun eine sogenannte algebraische Schleife übrig, bestehend aus Variable 3, 5, 6, 7 und Gleichung 2, 4, 5, 7. Diese bildet eine starke Zusammenhangskomponente im Graphen, die geschlossen gelöst werden muss. Da in diesem Fall die Gleichungen 2, 4, 5, 7 lineare Gleichungen sind, bildet diese starke Zusammenhangskomponente ein lineares Gleichungssystem, welches zur Laufzeit mit einem linearen Löser berechnet werden muss. Die Inzidenzstruktur des Graphen kann in eine Inzidenzmatrix überführt werden, wie in **Abbildung 4** zu sehen ist. Hierbei ist ein Matrixeintrag schwarz markiert, wenn eine Variable in einer Gleichung vorkommt. Die Reihen- und Spaltenreihenfolge entspricht der Berechnungsreihenfolge, wodurch sich eine untere Block-Dreiecksmatrix ergibt. Auf der Hauptdiagonalen befinden sich die kausalisierten Paare aus Gleichung und Variable. Die Einträge über der Hauptdiagonalen bilden einen Block. Dieser entspricht einer starken Zusammenhangskomponente im Graphen und muss, wie bereits erwähnt, als Gleichungssystem geschlossen gelöst werden. Zur besseren Veranschaulichung wurde der Block in **Abbildung 4** grau hinterlegt.

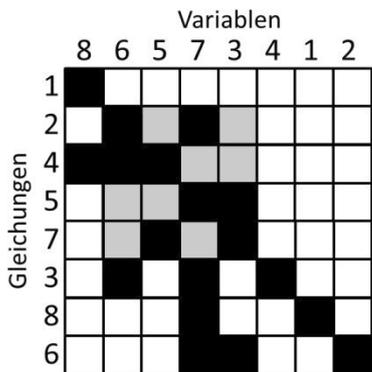


Abbildung 4 untere Block-Dreiecksmatrix

Die Blöcke auf der Hauptdiagonalen geben die serielle Ausführungsreihenfolge von Einzelgleichungen und Gleichungssystemen an.

3.2 Parallele Auswertung der Modellgleichungen

Die Theorie zur automatisierten, parallelen Auswertung der Modellgleichungen existiert bereits seit längerem. P. Arronson hat diesen Ansatz bereits 2006 vorgestellt [1]. Eine Umsetzung in etablierten Modelica-Compilern existiert jedoch erst seit 2014. Verschiedene Herangehensweisen an den vorgestellten Ansatz werden in [2], [3], [7] erläutert. Darüber hinaus gibt es weitere Möglichkeiten eine Simulation parallel auszuführen. Beispielhaft sei hier die „TLM“-Methodik (transmission line modelling) erwähnt, welche eine spezielle Modellierungsweise voraussetzt [4]. Bei Parallelisierungstechniken basierend auf parallelen Lösern sei auf das QSS-Verfahren verwiesen [5]. Im Gegensatz zu anderen Parallelisierungsstrategien hat der in dieser Arbeit vorgestellte Ansatz die Vorteile, dass das Modell nicht verändert werden muss und die gleiche numerische Stabilität erreicht wird, wie bei einer seriellen Simulation.

Der nun vorgestellte Ansatz zum Verteilen der Modellgleichungen auf parallele Prozesse ist ebenfalls graphenbasiert. Auf Basis der unteren Block-Dreiecksmatrix, kann ein gerichteter „Task-Graph“ erstellt werden, der Abhängigkeitsbeziehungen zwischen den Blöcken veranschaulicht. In **Abbildung 5** ist auf der linken Seite der Task-Graph des Beispielsmodells dargestellt. Ein Task ist ein Knoten und entspricht dabei einem Block in der unteren Block-Dreiecksmatrix. Dieser kann erst ausgeführt werden, wenn alle Vorgängerknoten berechnet worden sind. Der Task-Graph kann weiterhin mit Kanten- und Knoteninformationen über die Rechenzeiten erweitert

werden. Dieser Graph ist auf der rechten Seite dargestellt. Die Werte in den eckigen Feldern neben den Knoten stellen Ausführungszeiten dar, die notwendig sind, um den Task zu berechnen. Die Werte an den Kanten sind Kommunikationszeiten, die benötigt werden falls miteinander verbundene Knoten auf verschiedenen Prozessoren berechnet werden. Die konkreten Zahlenwerte können zum einen in einer vorhergehenden Simulation gemessen werden. Zum anderen kann eine umfassende Kostenmetrik zur Abschätzung der Laufzeiten genutzt werden, welche im OpenModelica-Compiler implementiert wurde. Dabei werden die Ausführungszeiten auf Basis der Gleichungsterme abgeschätzt.

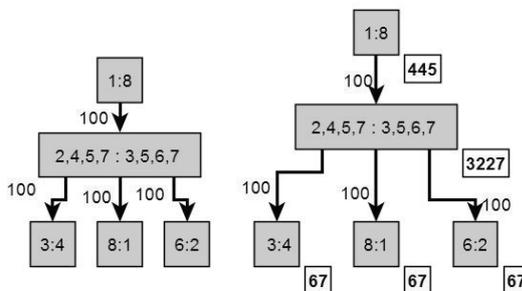


Abbildung 5 Task-Graph und Task-Graph mit Kommunikations- und Ausführungszeiten

Auf Basis des Task-Graphen kann eine Zuordnung der Tasks auf verschiedene Prozessoren erfolgen. Im vorliegenden Beispiel wäre es möglich, die Tasks 3:4, 8:1 und 6:2 auf unterschiedlichen Prozessoren zu berechnen

3.3 Scheduling

Die Zuordnung zwischen Tasks und Prozessoren wird als „Scheduling“ bezeichnet. Ein optimales Scheduling zu finden ist NP-schwer, weswegen heuristische Schedulingalgorithmen verwendet werden, die mit geringem Aufwand ausreichend gute Ergebnisse liefern [8]. Um die Problemgröße für den Schedulingalgorithmus zu verringern, werden zuvor ausgewählte Tasks zusammengefasst. Dabei werden nur solche Task-Gruppen zusammengefasst, bei denen sich eine Parallelisierung nicht lohnt. In **Abbildung 6** ist dargestellt, wie ein Graph vereinfacht wird. Die Ausführungszeiten werden dabei addiert.

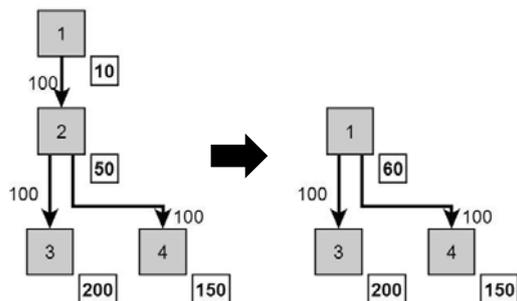


Abbildung 6 serielle Knoten zusammenfügen

In **Abbildung 7** werden die Tasks 1, 2 und 3 zusammengefügt. Würde Task 1 oder 2 auf einen anderen Prozessor ausgelagert, so erhöht die zusätzliche Kommunikationszeit die Gesamtausführungszeit und eine parallele Simulation wäre langsamer als eine serielle.

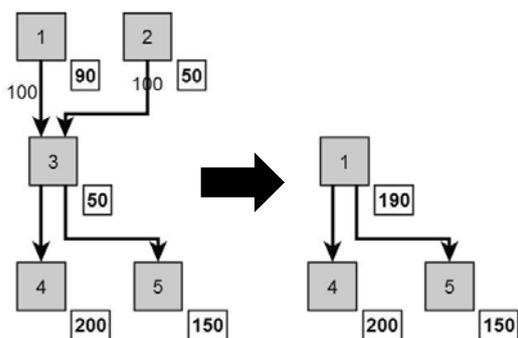


Abbildung 7 Elternknoten zusammenführen

Nach dem Zusammenfassen von Tasks, kann mittels eines Schedulingalgorithmus die Zuordnung zwischen Tasks und Prozessor erfolgen. Schedulingalgorithmen sind seit Jahrzehnten Gegenstand der Forschung und es existiert eine Vielzahl von Heuristiken. Die Algorithmen lassen sich für den vorliegenden Anwendungsfall in statisches und dynamisches Scheduling unterteilen. Beim statischen Scheduling wird jeder einzelne Task zur Compilezeit dem entsprechenden Thread zugewiesen. Das Scheduling bleibt somit zur Laufzeit konstant. Ein Beispiel dafür ist der „Modified Critical-Path-Scheduler“ [6]. In **Abbildung 8** ist ein beispielhafter Task-Graph dargestellt, welcher für zwei Prozessoren mit dem MCP-Algorithmus gescheduled wurde. Die grauen Tasks werden auf Prozessor 1 berechnet und die weißen Tasks auf Prozessor 2. Neben dem Task-Graphen ist ein entsprechendes Gantt-Diagramm dargestellt. Da-

bei sind alle Knoten, die auf einem Prozessor gerechnet werden, in einer Spalte angeordnet.

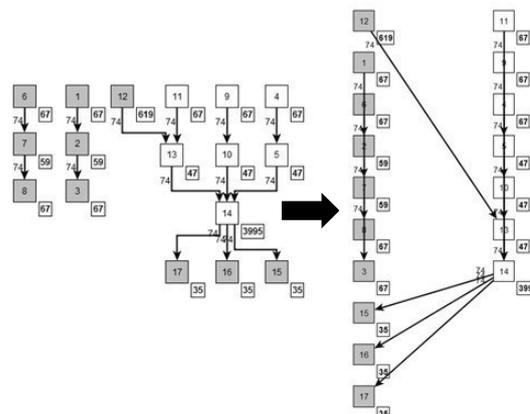


Abbildung 8 Task-Graph mit statischem Scheduling und Gantt-Diagramm

Beim dynamischen Scheduling wird die Zuordnung teilweise oder komplett zur Laufzeit neu bestimmt. Ein Beispiel dafür ist der „Level-Scheduler“ [7]. Der Graph wird dafür, wie in **Abbildung 9** zu sehen, in Ebenen unterteilt. Innerhalb der Ebenen besitzen die Knoten untereinander keine Kanten. Die Knoten einer Ebene werden zur Laufzeit der Reihe nach an die Prozessoren verteilt. Ist ein Prozessor frei, so kann er den nächsten, unbearbeiteten Task behandeln. Knoten der nächsten Ebene werden erst berechnet, wenn alle Knoten der vorangegangenen Ebene vollständig berechnet wurden.

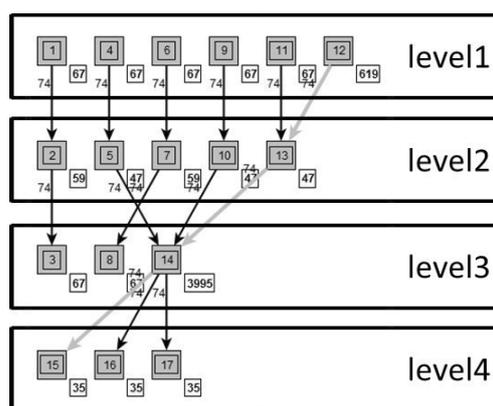


Abbildung 9 Task-Graph mit level-Scheduling

3.4 Domänenspezifische Task-Graphen

Mit Modelica ist die Beschreibung von Modellen verschiedenster Domänen möglich. Aufgrund der mathematischen Beschreibung der physikalischen Zusammenhänge weisen die zugehörigen Task-

Graphen gewisse Eigenarten auf. Ein typischer Anwendungsfall ist beispielsweise die Modellierung eines mechanischen Mehrkörpersystems. Dafür werden mehrere, starre Körper mit Gelenken verbunden und mit Lasten (Kräfte und Momente) beaufschlagt. Die starren Körper bilden meist einen kinematischen Baum, in dem die Gelenkpositionen und Gelenkbeschleunigungen für die Berechnung der Gelenkkräfte und -momente benötigt werden. Mit den Lastelementen lassen sich im Weiteren die Gelenk- und Körperbeschleunigungen errechnen. Da für jeden Körper die kinematischen Größen der angrenzenden Körper bekannt sein müssen, sind die Zustandsgrößen verbundener Körper voneinander abhängig und es ergibt sich ein großes Gleichungssystem. Dieses Gleichungssystem dominiert die Berechnung der Zustandsableitungen und macht bis zu 95% der Auswertungsdauer der Modellgleichungen aus. In **Abbildung 10** ist der Task-Graph eines Mehrfach-Pendels dargestellt. Das dominierende Gleichungssystem ist schwarz gefärbt. Nahezu alle Kanten münden in dieses Gleichungssystem. Mit der vorgestellten Task-graph-Parallelisierung sind solche Modelle nicht effektiv parallelisierbar. Eine parallele Auswertung des Gleichungssystems durch einen parallelen Löser wäre denkbar. Die erreichte Systemgröße der untersuchten Modelle (<1000) ist jedoch zu klein, um mit diesen Ansätzen verringerte Simulationszeiten zu erreichen.

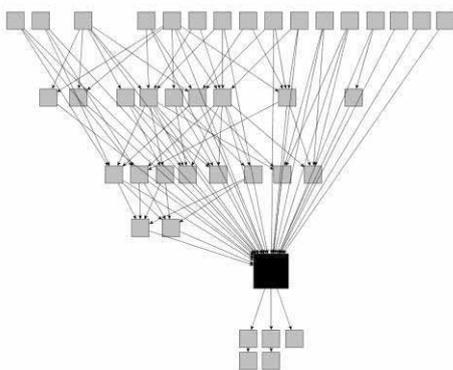


Abbildung 10 Typischer Task-Graph eines Mehrkörpersystems mit dominierendem Gleichungssystem

Neben mechanischen Modellen werden häufig hydraulische oder thermohydraulische Systeme modelliert. In diesen Systemen müssen die Druckänderungen in den verschiedenen Volumina berechnet werden. Dazu wird der Differenzdruck am Volumen ermittelt und daraus der Volumenstrom berechnet. Die Aufsummierung der Volumenströme in den Verzwei-

gungspunkten ermöglicht dann die Berechnung des Druckanstiegs. Prinzipiell kann jedes Volumen einzeln berechnet werden, woraus sich ein breiter Task-Graph ergibt. Dieser besitzt mehrere Gleichungssysteme, die einzeln auswertbar sind. **Abbildung 11** zeigt den Task-Graphen eines hydraulischen Modells. Es gibt vier parallel auswertbare, schwarz markierte Gleichungssysteme.

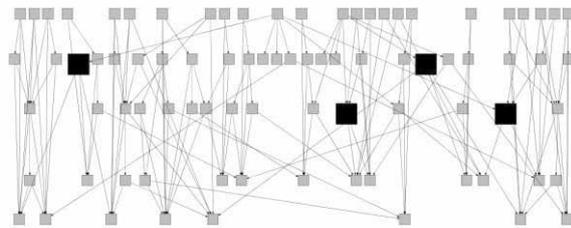


Abbildung 11 Task-Graph eines hydraulischen Modells

Hydraulische und thermohydraulische Modelle sind demzufolge sehr gut für eine Task-Graph-Parallelisierung geeignet, wie F. Casella bereits erläutert hat [9].

4 Möglichkeiten zur Verbesserung der parallelen Simulation

4.1 Gleichungssysteme aufspalten

Wie in Abschnitt 3.4 bereits herausgestellt wurde, können große Gleichungssysteme den Flaschenhals für eine parallele Simulation darstellen. In mechanischen Systemen ist dies systembedingt. Es treten jedoch auch Modelle auf, deren Gleichungssysteme in mehrere kleine Systeme zerlegt werden können. Diese Methode wurde in [10] gezeigt. Dabei wird ein graphentheoretischer Ansatz basierend auf dem bipartiten Graphen der algebraischen Schleife genutzt. Mittels einer teilweisen Gauss-Elimination werden Kanten in der Schleife neu angeordnet, wodurch sich kleinere Schleifen ergeben. Dadurch kann die Struktur des Task-Graphen verbessert werden, wie in **Abbildung 12** zu erkennen ist.

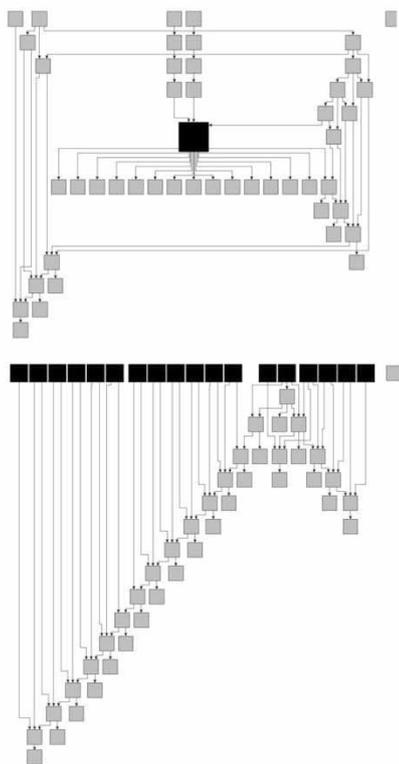


Abbildung 12 oben: ohne Aufspaltung des Gleichungssystems, unten: nach der Aufspaltung des Gleichungssystems

4.2 Optimierte Speicherverwaltung

Neben einer effizienten Verteilung der einzelnen Berechnungsaufgaben auf die vorhandenen Prozessoren, ist auch die Anordnung der Variablen im Programmspeicher von Interesse. Aufgrund des Aufbaus moderner Rechnerarchitekturen, werden für den Zugriff auf verschiedene Speicherbereiche unterschiedliche Zugriffszeiten benötigt. Besonders bei laufzeitkritischen Programmen, wie z.B. bei der Echtzeitsimulation oder bei der Simulation sehr rechenintensiver Modelle, muss dieser Fakt berücksichtigt werden. Im Compiler Backend werden daher die Speicherbereiche der einzelnen Threads voneinander separiert. Außerdem werden alle Variablen, die für die Berechnung eines Tasks benötigt werden, an möglichst fortlaufenden Speicheradressen abgelegt [11].

5 Ergebnisse

Die vorgestellte Methodik zur Parallelisierung des Task-Graphen ist im OpenModelica-Compiler implementiert und somit frei verfügbar. Die Parallelisierung läuft vollautomatisch. Für die folgenden Messungen wurde der standardmäßige „level“-Scheduler mit einer OpenMP Parallelisierung benutzt. Gemessen

wurden die Simulationszeiten auf einem i7-3930K mit 6 Kernen à 3.20 GHz. Der generierte C-Code wurde unter Linux mit dem GCC 4.8.2 kompiliert.

Wie bereits erwähnt, ergeben sich domänenspezifische Unterschiede. Daher werden im Folgenden ein mechanisches, ein hydraulisches und ein elektrisches Modell untersucht. Es wird jeweils die Simulationszeit über der Anzahl an genutzten Threads (Prozessoren) dargestellt. **Abbildung 13** zeigt die Simulationszeit für das kinematische Modell eines V6-Motors. Wie bereits in Abschnitt 3.4 erläutert sind mechanische Modelle für die Task-Graph-Parallelisierung ungeeignet, weswegen auch keine Verbesserung der Simulationszeit erreicht wird.

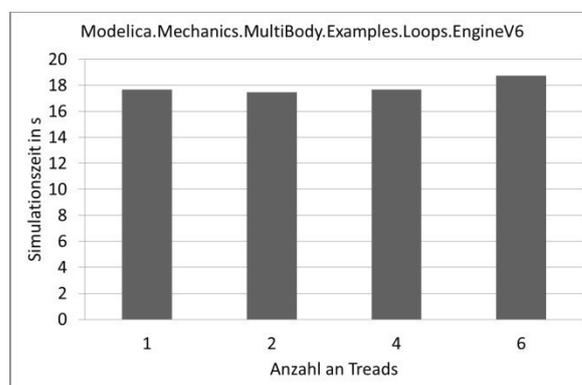


Abbildung 13 Simulationszeit des EngineV6-Modells

Abbildung 14 zeigt die Verbesserung der Simulationszeit für das hydraulische BranchingDynamicPipes-Modell. Hydraulische Modelle weisen einen breiten Task-Graphen mit parallelen Gleichungssystemen auf. Mit sechs verwendeten Prozessoren konnte eine Verbesserung der Simulationsdauer um den Faktor 4.3 erzielt werden.

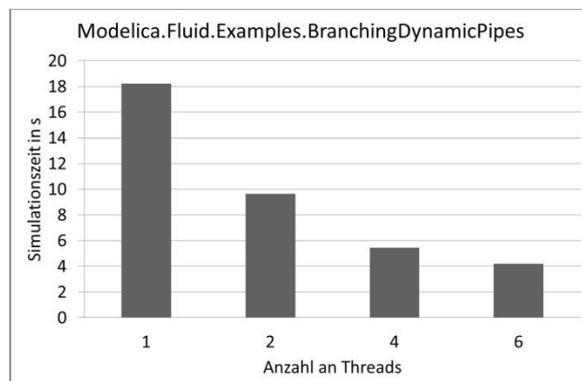


Abbildung 14 Simulationszeiten des BranchingDynamicPipes-Modells

Elektrische Modelle können ebenfalls durch Task-Graph-Parallelisierung schneller simuliert werden, wie in **Abbildung 15** zu erkennen ist. Die Verbesserung erreicht hier den Faktor 2.3.

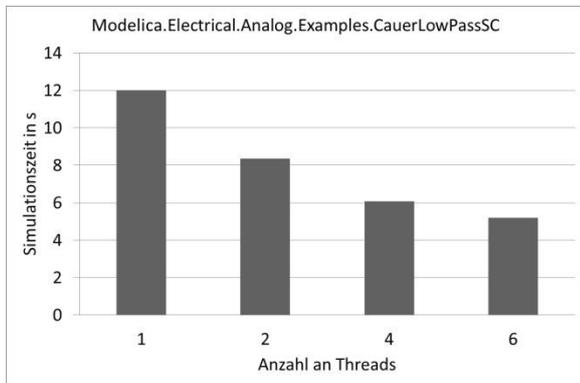


Abbildung 15 Simulationszeiten des Cauer-LowPassSC-Modells

6 Fazit

Die Task-Graph-Parallelisierung hat seit kurzem den Weg in die Anwendung in kommerzielle und nicht-kommerzielle Modelica-Compiler gefunden. Dieser Ansatz ist im Gegensatz zu anderen Parallelisierungsstrategien automatisiert und ohne Modelländerungen und ohne Veränderung der numerischen Stabilität einsetzbar. Die erreichbaren Geschwindigkeitsvorteile sind jedoch modellabhängig. Hydraulische und thermohydraulische Modelle lassen sich besonders gut unter Ausnutzung mehrerer Prozessoren simulieren.

Referenzen

- [1] P. Aronsson. *Automatic Parallelization of Equation-Based Simulation Programs*. Department of Computer and Information Science, Linköping University, Schweden, 2006
- [2] H. Elmqvist, S.E. Mattson und H. Olsson. *Parallel Model Execution on Many Cores*. In Proceedings of the 10th International Modelica Conference, Lund, Schweden, 2014.
- [3] M. Gebrehmedin, P.Fritzson. *Automatic task based analysis and parallelization in the context of equation based languages*. In Proceedings of the 6th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, Berlin, Deutschland, S.49-52, 2014
- [4] M. Sjölund, R. Braun und P. Fritzson. *Towards Efficient Distributed Simulation in Modelica using Transmission Line Modeling*. In Proceedings of the 4th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, Zürich, Schweiz, 2011
- [5] F. Bergero, X. Floros, J. Fernández. *Simulating Modelica models with a Stand-Alone Quantized State Systems Solver*. In Proceedings of the 9th International Modelica Conference, München, Deutschland, 2012.
- [6] M.-Y. Wu und D. D. Gajski. *Hypertool: A programming aid for message-passing systems*. IEEE TRAN. ON PARALLEL AND DISTRIBUTED SYSTEMS, vol. 1, pp. 330–343, 1990.
- [7] M. Walther und V.Waurich, *Equation based parallelization of Modelica models*. In Proceedings of the 10th International Modelica Conference, Lund, Schweden, 2014.
- [8] P. Quinton, *Scheduling*. In: Encyclopedia of Parallel Computing, Herausgeber: D.Padua, Springer, New York, USA, S. 1788-1789, 2011
- [9] F. Casella, *A Strategy for Parallel Simulation of Declarative Object-Oriented Models of Generalized Physical Networks*. In Proceedings of the 5th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, Nottingham, UK, 2013
- [10] V.Waurich, I.Gubsch, C.Schubert, *Reshuffling: a symbolic pre-processing algorithm for improved robustness, performance and parallelization for the simulation of differential algebraic equations*. In Proceedings of the 6th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, Berlin, Deutschland, S.3-10, 2014
- [11] M.Walther, M.Flehmg, U. Markwardt, *Performance analysis of equation based simulations*. In Proceedings of the 6th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, Berlin, Deutschland, S.53-65, 2014

Strukturvariabilität für Modelica-Modelle

Daniel Gomez Esperon
Technische Universität Berlin, Fachgebiet Softwaretechnik
gomezesperon@tu-berlin.de

Klassische Simulationsmodelle basieren auf einem monolithischen Gleichungssystem. Moderne Sprachen und Werkzeuge unterstützen die Entwicklung komplexer Modelle durch Abstraktion mit Hilfe von Objekt- oder Komponentenorientierung und den Einsatz von Bibliotheken. Für spezielle Modelle, bei denen eine grundlegende Verhaltensänderung auftritt, sind Änderungen am Gleichungssystem während der Simulation wünschenswert. Strukturvariable Modelle bieten ein Konzept zur Beschreibung eines solchen Verhaltens. Existierenden oder experimentellen Umsetzungen dieses Konzeptes mangelt es bislang an Wiederverwendbarkeit oder die Komplexität der Modelle und Gleichungssysteme ist eingeschränkt. Im Beitrag werden eine Methode und ihre Umsetzung in einem Werkzeug vorgestellt, die es ermöglichen, Strukturvariabilität nachträglich an existierende, komponentenorientierte Modelle anzufügen. Basierend auf einem Ausgangsmodell wird der Austausch einzelner Komponenten durch Instanzen eines anderen Typs beschrieben und automatisch durch Manipulation der Modellstruktur vorgenommen. Eine Simulation wird durch automatische Generierung von Variationen und sequenzielle Ausführung des Ausgangsmodells realisiert. Einzelne Variationen werden zu Functional Mock-up Interface-Modellen kompiliert und die Simulation zeitschrittweise überwacht. Beispielhaft wird dieses Vorgehen im Prototyp *MoVasE* zunächst für Modelica-Modelle erprobt.

1 Komponentenorientierte Simulationsmodelle

Ein hier betrachtetes Simulationsmodell besteht im Kern aus einem differential-algebraischen Gleichungssystem, das zur Simulation von Solvern zeitschrittweise gelöst wird. Modelle für den praktischen Einsatz können mehrere 10000 Gleichungen enthalten. Etablierte Sprachen und Entwicklungsumgebungen unterstützen die Modellbildung solcher komplexer Modelle durch Abstraktion und Wiederverwendbarkeit. Ein Beispiel ist das Konzept der Objektorientierung in der Sprache *Modelica* [2]. Simulationsmodelle können durch Verbindung von Objekten, hier und im weiteren *Komponenten* genannt, zusammengesetzt werden und selbst als Komponenten in anderen Modellen verwendet werden. Komponenten werden von Modellen instanziiert, ein Modell kann damit als *Typ* einer Komponente angesehen werden. Bibliotheken bieten Typen, die ein in verschiedenen Kontexten nutzbares Verhalten

kapseln. Sie enthalten Gleichungen und bieten Schnittstellen zur Verbindung mit anderen Komponenten und deren Gleichungen. Diese Teilmodelle aus Bibliotheken können dann beliebig oft als Instanzen bzw. Komponenten in einem Simulationsmodell genutzt werden. Eine Komponente kann selber Komponenten enthalten und dieses wieder andere Komponenten. Eine hierarchische Struktur liegt vor. Um ein solches objektorientiertes Modell simulieren zu können, müssen Umformungen zu einem monolithischen Gleichungssystem stattfinden. Die Strukturinformationen aus der objektorientierten Beschreibung gehen dabei verloren.

Für bestimmte Arten von Modellen, bei deren Simulation zum Beispiel eine grundlegende Verhaltensänderung auftritt, ist es wünschenswert, dieses monolithische Gleichungssystem anzupassen. Es kann nützlich sein, Gleichungen zu ersetzen, neue Gleichungen und Variablen hinzuzufügen oder zu entfernen. Einfaches Beispiel ist ein Ball, der von der Tischkante rollt: Solange er auf dem Tisch rollt,

lässt sich seine Bewegung mit anderen Gleichungen beschreiben, als wenn er sich im freien Fall befindet. Etablierte Sprachen und Entwicklungsumgebungen unterstützen die Umsetzung solcher Verhaltensänderungen nur in sehr geringem Umfang. Besondere Herausforderung ist dabei die Distanz zwischen abstrakter, objektorientierter Beschreibung und dem zur Simulation gelösten Gleichungssystem. Gewünschte Änderungen sollten wie gewohnt in der Komponentenabstraktion beschreibbar sein, wobei die Umsetzung der Änderungen im Gleichungssystem automatisch geschehen sollte.

1.1 Strukturvariabilität

Strukturvariabilität ist ein existierendes Konzept, das es ermöglicht, diese Veränderungen zur Simulationszeit zu beschreiben. Das zu einem Zeitpunkt der Simulation herangezogene Gleichungssystem wird dabei als *Mode* bezeichnet. Übergänge sind *Transitionen*. Diese werden durch *Schaltbedingungen (Trigger)* angestoßen und erfordern analog zur klassischen Simulation eine *Initialisierung* des folgenden Modes. Betrachtet man die Modes als Knoten und die Transitionen als Kanten, kann ein strukturvariables Modell als Zustandsübergangsdiagramm angesehen werden. Es existieren verschiedene experimentelle Sprachen und Werkzeuge, die unterschiedliche Interpretationen und Beschreibungsformen vorsehen und strukturvariable Modelle simulieren können. Die Sprache MOSILA mit ihrer Simulationsumgebung MOSILAB [3] ist eine objektorientierte Sprache, die die Verhaltensänderung durch Zustandsübergangsdiagramme innerhalb von Komponenten umsetzt. SOL [4] verfügt über spezielle Sprachelemente und Ptolemy II [5] führt Strukturvariabilität durch sogenannte Modal-Models und ebenfalls Zustandsübergangsdiagramme ein. Des Weiteren existiert DEVS [6] als Formalisierungsmöglichkeit, strukturvariable Systeme zu beschreiben. Nachteil der Ansätze ist die fehlende Wiederverwendbarkeit, da eine Implementierung in entsprechender Sprache notwendig ist. Die Simulierbarkeit komplexer Gleichungssysteme ist in diesen Werkzeugen begrenzt. Es existieren kaum Bibliotheken.

Das *DySMo*-Framework [7] bietet die Möglichkeit, etablierte Simulationsumgebungen für die strukturvariable Simulation durch ein Python-Skript anzu-

steuern. Dabei können existierende Modelle als Basis für Modes verwendet werden. Die Mächtigkeit der etablierten Umgebungen zur Entwicklung der Modelle und zur Lösung der Gleichungssysteme bleibt erhalten. Jeder Mode wird durch ein manuell erstelltes und simulierbares Modell dargestellt, was die Anzahl der möglichen Modes begrenzt. Bedingungen für den Wechsel eines Modes werden lokal durch Ergänzungen im entsprechenden Modell untergebracht, damit die Simulation in der Umgebung ohne externe Steuerung terminiert. Übergänge und die Auswahl des nächsten Modes werden im Skript beschrieben.

In diesem Beitrag werden eine Methode und ein Werkzeug vorgestellt, die es ermöglichen, Strukturvariabilität durch Erweiterung eines bestehenden Modelica-Modells zu erzeugen. Der Ansatz ist allgemein für komponentenorientierte Sprachen konzipiert, die Realisierung für andere Sprachen ist noch zu untersuchen. Der folgende Abschnitt beschreibt die Methode abstrakt und führt wichtige Begriffe ein. In Abschnitt 3 wird das Vorgehen an einem Beispiel mit Wechsel einer Spannungsquelle gezeigt. Die Methode wurde im prototypischen Werkzeug *MoVasE - Modelica Variable-structure Editor* umgesetzt, das im Abschnitt 4 vorgestellt wird. Der letzte Abschnitt beschreibt den aktuellen Stand und weiteres Vorgehen.

2 Methode

Die hier vorgestellte Methode zur Ergänzung von Strukturvariabilität abstrahiert im manuellen Modellbildungsprozess vom Konzept der Modes. Sie führt Strukturvariabilität lokal und unabhängig an den Komponenten eines Ausgangsmodells ein, die zur Simulationszeit ausgetauscht werden sollen. Eine austauschbare Komponente enthält dann ein Zustandsübergangsdiagramm von Typen, durch deren Instanzen sie ausgetauscht werden kann. Durch Transitionen verbunden kann eine Komponente dann ihren Typ wechseln. Modes werden basierend auf diesen Informationen automatisch generiert. Eine Transition enthält die Information

- *Wann* sie schaltet. *Trigger* können zeit- oder zustandsbasiert sein und logisch kombiniert werden.

- *Wie* der neue Typ an den alten Typ angepasst werden muss. Es besteht allgemein keine Vererbungsbeziehung, daher sind Anpassungen notwendig. Dies geschieht mit *Mappings*.
- *Welche* Startwerte die neue Komponente des neuen Typs erhält durch *Initializations*.

Informationen über austauschbare Komponenten und ihre Transitionen werden mit Unterstützung des Werkzeuges generiert und als separates *Metamodell* gehalten. Basierend auf diesen Informationen können jetzt die unterschiedlichen Ausprägungen des Modells mit verschiedenen Komponentenkonfigurationen abgeleitet werden. Die Zustandsübergangsdiagramme von verschiedenen Komponenten werden parallel interpretiert. Eine Komponentenkonfiguration besteht jeweils aus einer Komponente aus jedem Diagramm. Diese Komponentenkonfiguration bildet zusammen mit dem unveränderlichen Teil des Ausgangsmodells einen Mode. Das Ausgangsmodell, hier *Master-Modell* genannt, dient dann als Vorlage zur Generierung von simulationsfähigen Gleichungssystemen. Dazu wird das Ausgangsmodell in eine besser verarbeitbare Form als die reine Textdarstellung übersetzt. Der entstehende *Abstract Syntax Tree (AST)* ist eine übliche Darstellungsform für syntaktische Informationen von Programmcode. Sowohl das Ausgangsmodell als auch die Zustände der austauschbaren Komponenten enthalten AST-Elemente. Damit kann für jeden Mode mit den Beschreibungen in den Mappings ein simulierbarer AST zusammengesetzt werden. Dieser AST enthält keine strukturvariable Information. Er stellt ein klassisches Simulationsmodell dar. Zur Simulation werden die ASTs der einzelnen Modes mit dem gleichen Verfahren wie klassische Modelle zu einem simulierbaren Gleichungssystem übersetzt. Dabei können aus den ASTs der Modes auch Modelica-Modelle generiert werden. Die simulierbaren Gleichungssysteme werden dann für die strukturvariable Simulation zu *Functional Mock-up Interface (FMI)*-Modellen (<https://www.fmi-standard.org/>) kompiliert, die einen Solver enthalten. Transitionen in den Zustandsübergangsdiagrammen der austauschbaren Komponenten bestimmen die Ausführungsreihenfolge der einzelnen Modes. Für jeden Simulationsschritt wird geprüft, ob und welche Transitionen schalten. Entsprechend wird der Mode zu einer neuen Komponentenkonfiguration gewechselt und Initialisierungen der Transi-

tionen vorgenommen. Alle nicht explizit durch eine Transition betroffenen Komponenten werden durch ihren letzten Wert im vorherigen Mode initialisiert. Während der gesamten Simulation werden ausgewählte Daten der Modes gesammelt und zusammengefügt. Das Ergebnis ist als ein Datensatz plotbar.

3 Beispiel Wechsel einer Spannungsquelle

Ausgangssituation ist die Erkenntnis, dass ein existierendes Modell von Strukturvariabilität während der Simulation profitieren könnte. Dieses Modell wird als Master-Modell verwendet und ist das einzige benötigte simulierbare Modell. Zusätzlich werden Alternativimplementierungen für Typen von auszutauschenden Komponenten benötigt. Als einfaches Beispiel wird hier ein Modell aus einer Spannungsquelle und einem Widerstand betrachtet. Das Ausgangsmodell in Abbildung 1 zeigt schematisch das Modell mit einer Sinusspannungsquelle. Es ist ein klassisches, reines Modelica-Modell und kann mit jedem Modelica-unterstützenden Werkzeug simuliert werden.

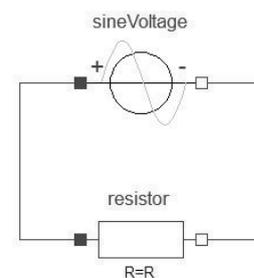


Abbildung 1: Ausgangsmodell mit Sinusspannung.

Ziel ist es, während der Simulation die Sinusspannungsquelle durch eine Rechteckspannungsquelle zu ersetzen, um den Verlauf der Spannung wie in Abbildung 2 während der Simulation zu verändern. Die initial eingestellte Frequenz und Amplitude sollen dabei erhalten bleiben. Der Übergang soll kontinuierlich verlaufen.

Die Methode umfasst fünf Schritte bis zur Simulation, die im Folgenden dargestellt sind.

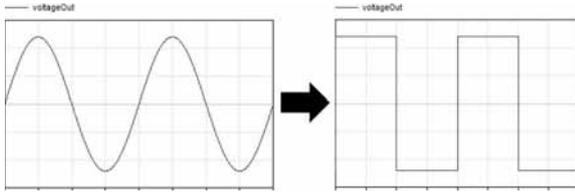


Abbildung 2: Sinusspannung und Rechteckspannung.

3.1 Schritt 1: Übersetzen der Modelle

Zusätzlich zum Master-Modell wird der Typ der Rechteckspannung benötigt und durch die *Reimplementations* zur Verfügung gestellt. Das Master-Modell mit der Sinusspannungsquelle und der Typ der Rechteckspannung werden in die AST-Form übersetzt. Abbildung 3 zeigt einen Ausschnitt der ASTs mit den wichtigsten Elementen.

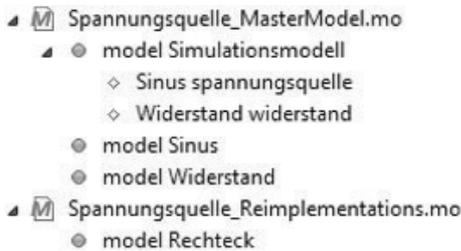


Abbildung 3: Ausschnitt Abstract Syntax Tree.

Das *model Simulationsmodell* ist das zu simulierende Modell. Es enthält die Komponenten *spannungsquelle* vom Typ *Sinus* und *widerstand* vom Typ *Widerstand*. Ebenfalls enthält das Master-Modell die Implementierungen der entsprechenden Typen *Sinus* und *Widerstand*. Die Reimplementations enthalten nur den Typ *Rechteck*, der zur Instanziierung der neuen Komponente herangezogen wird. Im Master-Modell und als Reimplementations können auch Referenzen auf Bibliothekselemente verwendet werden.

Relevante Master-Modell- und Reimplementations-Elemente wie Typen und Komponenten liegen jetzt in AST-Form für das Werkzeug vor. Darauf aufbauend können folgende Schritte umgesetzt werden.

3.2 Schritt 2: Typ-Transitionen

Zur Vorbereitung des beabsichtigten Komponentenwechsels werden mögliche *Typ-Transitionen* eingeführt. Dazu werden Gruppen aus Typen gebildet, deren Instanzen, die Komponenten, austauschbar sein sollen. Für dieses Beispiel besteht die einzige benötigte Gruppe aus den Typen *Sinus* und *Rechteck*.

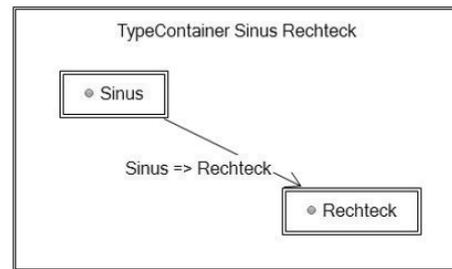


Abbildung 4: Austauschbare Typen Sinus und Rechteck mit Transition.

Der Pfeil in Abbildung 4 zwischen diesen Typen ist eine Typ-Transition. In dieser Transition *müssen* Mappings angegeben werden und *können* Initialisierungen vorgesehen werden. Der Austausch einer Komponente vom Typ *Sinus* durch eine Komponente vom Typ *Rechteck* findet im späteren Abschnitt 3.4 durch AST-Manipulationen statt. Die Auswirkung der Manipulationen muss lokal im AST-Knoten des Typs stattfinden, damit referenzierende Komponenten eine unveränderte Schnittstelle vorfinden. Da keine Vererbung zwischen den Typen verlangt wird, die eine Schnittstellengleichheit vorgeben könnte, müssen durch Mappings Anpassungen des neuen Typs an die Schnittstelle des alten Typs vorgesehen werden. Abbildung 5 zeigt einen Ausschnitt des Modelica-Programmcodes der Typen *Sinus* und *Rechteck*. Eingerahmt sind die nach außen sichtbaren Parameter und Variablen der beiden Typen.

model Sinus	model Rechteck
parameter Real amp;	parameter Real amp;
parameter Real f;	parameter Real freq;
Real volt;	Real volt;
equation	equation
...	...
end Sinus;	end Rechteck;

Abbildung 5: Variablen der Typen Sinus und Rechteck

Die Umgebung im Master-Modell erwartet durch die ursprüngliche Definition der *spannungsquelle* vom Typ *Sinus* Parameter *amp* und *f* sowie Variable *volt*. Um die Spannungsquelle durch eine Komponente vom Typ *Rechteck* ersetzen zu können, müssen die Variablennamen von *Rechteck* durch Mappings angepasst werden. Es wird ein Mapping definiert, das den Parameter *freq* zu *Parameter f* umbenennet.

Die Toolunterstützung bietet Analysen, notwendige Mappings zu identifizieren. Durch Name-, Typ-, Sichtbarkeits- und Modifizierbarkeitsgleichheit wird ein direktes Mapping von *amp* und *volt* vorgeschlagen und ein offenes Mapping für *f* identifiziert. Vorgeschlagene Mappings sind überschreibbar.

In der Typ-Transition optional anzugeben sind Initialisierungen. Anforderung an das strukturvariable Modell war der kontinuierliche Übergang von der Sinus zur Rechteckspannung. Dieser ist nicht alleine durch die Initialisierung möglich, da die Rechteckspannung nur zwei Werte kennt: ihr Maximum und ihr Minimum. Der kontinuierliche Übergang wird weiterhin mit dem Zeitpunkt des Komponentenwechsels in Abschnitt 3.3 sichergestellt. Hier wird zur Initialisierung der letzte Wert der Variablen *volt*, der der Ausgangsspannung entspricht, der *volt* Variablen in der *Rechteck*-Komponente zugewiesen. Als äquivalent angesehene Variablen werden durch Analysen automatisch zur Initialisierung mit dem letzten Wert der vorherigen Simulation vorgeschlagen.

In einer Transition zwischen zwei Typen werden nur Mappings und Initialisierungen für die Komponente des neuen Typs angegeben. Für den Wechsel der Komponente wird die Simulation unterbrochen und mit der neuen Komponente fortgesetzt. Für den Rest des Modells muss ebenfalls ein kontinuierlicher Übergang sichergestellt werden. Da durch die Mappings sichergestellt ist, dass sich die Schnittstellen zum Rest des Modells nicht ändern, bleiben alle anderen Modellvariablen erhalten und werden mit dem letzten Wert der vorherigen Simulation initialisiert.

Die Typen *Sinus* und *Rechteck* sind durch die Typ-Transition als austauschbar beschrieben. Mappings und Initialisierungen sind vollständig angegeben. Statisch wurde festgelegt, wie der Austausch von Komponenten der beiden Typen im AST des Master-Modells stattfindet und wie Startwerte der neuen Komponente gesetzt werden.

3.3 Schritt 3: Komponenten-Transitionen

Für Komponenten des Simulationsmodells können jetzt Typ-Transitionen verwendet werden, um den Austausch der Komponenten zu beschreiben. Im Beispiel wird die Komponente *spannungsquelle* durch eine neue Komponente vom Typ *Rechteck* ersetzt. Die Notation in Abbildung 6 ist analog zu der Zustandsübergangsdiagramm-Notation der Typ-Transitionen. Knoten sind dabei Komponenten.

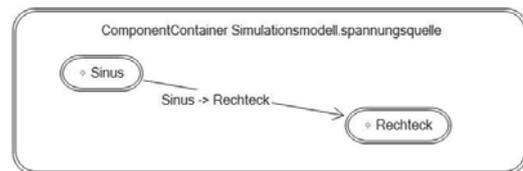


Abbildung 6: Austauschbare Komponenten.

Die eingeführte Komponenten-Transition *Sinus* \rightarrow *Rechteck* referenziert die entsprechende Typ-Transition und kann dort vorgesehene Mappings und Initialisierungen übernehmen. Diese können komponentenaustauschspezifisch überschrieben werden. Typ-Transitionen dienen der Wiederverwendung von Mappings und Initialisierungen. Gäbe es mehrere Komponenten vom Typ *Sinus*, könnten in entsprechenden Komponenten Transitionen die eine Typ-Transition als Vorlage genutzt werden.

Zusätzlich zu den Mappings und Initialisierungen sind in Komponenten-Transitionen Trigger anzugeben. Ein Trigger kann simulationszeit- oder zustandsabhängig von den Variablenwerten sein. Als Simulationszeit können die *absolute* Simulationszeit oder die *relative* Simulationszeit einer austauschbaren Komponente angegeben werden. Im Beispiel soll die Spannungsquelle unabhängig von der Parametrisierung mit der Amplitude und Frequenz mindestens zwei Perioden lang als Sinusspannung auftreten und dann an ihrem Maximum einen kontinuierlichen Übergang zum Maximum der Rechteckspannung finden. Die dazu benötigten Trigger sind:

- Absolute Simulationszeit \geq zwei Perioden
- Ausgangsspannung \geq parametrisierte Amplitude

3.4 Schritt 4: Mode-Generierung

Die Mode-Generierung dient der Erstellung der zur strukturvariablen Simulation herangezogenen, verschiedenen klassischen Simulationsmodelle. Sie kann dynamisch während der Simulation ausgeführt werden oder, wie im Werkzeug zu Untersuchungszwecken vorgesehen, vor dem Start der Simulation. Vorteil der dynamischen Mode-Generierung ist, dass nur während der Simulation betretene Modes erstellt werden müssen, wohingegen bei der Generierung vor der Simulation alle möglichen Modes generiert werden.

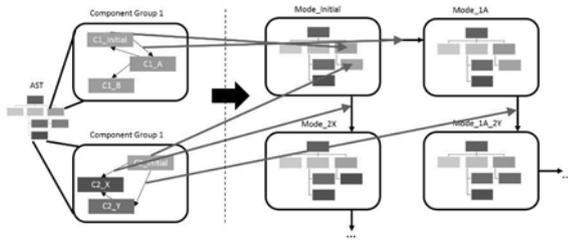


Abbildung 7: Algorithmus zur Mode Generation.

Im Trivialbeispiel der Spannungsquelle entstehen zwei Modes: einer mit den Komponenten *spannungsquelle* vom Typ *Sinus* und *widerstand* vom Typ *Widerstand* (nicht austauschbar) und einer mit den Komponenten *spannungsquelle* vom Typ *Rechteck* und *widerstand* vom Typ *Widerstand*. Dazwischen ist die Transition aus 3.3 mit den Triggern 'Absolute Simulationszeit > zwei Perioden' und 'Ausgangsspannung \geq parametrisierte Amplitude'. Zur Veranschaulichung des Algorithmus für mehrere Komponentengruppen und Transitionen wird die in Abbildung 7 dargestellte Situation herangezogen. Zu zwei Komponenten des ASTs existieren Zustandsübergangsdigramme mit je drei Komponenten und drei Transitionen. Aus den Zustandsübergangsdigrammen links in der Abbildung wird das Zustandsübergangsdigramm der Modes rechts abgeleitet. Dazu wird, beginnend mit dem initialen Mode, der dem Ausgangsmodell entspricht, je eine Komponenten-Transition verfolgt. Die Zielkomponente dieser Transition liefert eine auszutauschende Komponente. Aus dieser auszutauschenden Komponente und dem vorherigen Mode wird der nächste Mode generiert, der sich in einer Komponente unterscheidet. Dazwischen entsteht eine Transition, die di-

rekt aus der Komponenten-Transition abgeleitet wird. Dieses Vorgehen wird für alle austauschbaren Komponenten und ihre herausführenden Transitionen wiederholt. Modes identifizieren sich durch ihre Komponentenkonfiguration und können damit wiederverwendet werden. Das Resultat ist ein Zustandsübergangsdigramm mit Knoten für jede mögliche Komponentenkonfiguration und allen direkt möglichen Transitionen. Eine Transition verbindet dabei immer nur zwei Modes, die sich in ihrer Komponentenkonfiguration um eine Komponente unterscheiden. Aus der vollständigen Komponentenkonfiguration in einem Mode und dem übrigen Teil des Initialmodells wird dann durch Manipulation des ASTs ein neues, simulierbares Modell in Form eines neuen ASTs generiert.

3.5 Schritt 5: Simulation

Ein Mode enthält einen simulierbaren AST. Dieser AST repräsentiert ein simulierbares klassisches Simulationsmodell und kann durch vorhandene Mechanismen zu einem Gleichungssystem übersetzt werden. Dieses Gleichungssystem wird im Werkzeug zu einem FMI-Modell mit integriertem Solver übersetzt. Dadurch ist jeder Mode einzeln simulierbar. Die Simulation beginnt mit dem FMI-Modell des initialen Modes. Nach jedem Simulationsschritt werden herausführende Transitionen aus dem Zustandsübergangsdigramm und deren Trigger auf Auslösung geprüft. Hat eine Transition ausgelöst, wird die Simulation unterbrochen und eine neue Simulation mit dem FMI-Modell des folgenden Modes fortgeführt. Dabei werden Initialisierungen der Transition herangezogen. Ergebnisse der einzelnen Mode-Simulationen werden zusammengefasst und sind im Werkzeug als eine Datenmenge darstellbar.

Im Beispiel beginnt die Simulation mit dem Mode der Sinusspannung. Amplitude und Frequenz werden nur initial parametrisiert. Beim zweiten Mode werden die Werte wie in Abschnitt 3.2 beschrieben übergeben. Nach zwei Perioden evaluiert der erste Trigger zu wahr. Erst wenn die Sinusspannung ihr Maximum erreicht, sind die beiden verundeten Trigger wahr und der Modewechsel wird vollzogen. Die Rechteckspannung beginnt wie in der Initialisierung der Typ-Transition vorgesehen mit dem letzten Wert der Sinusspannung. Ein kontinuierlicher Übergang ist entstanden.

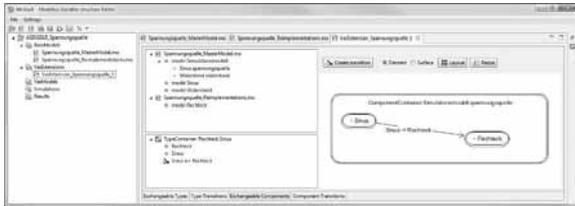


Abbildung 8: MoVasE mit Austauschbarer Komponente.

4 Umsetzung im Werkzeug

Die vorgestellte Methode wird im prototypischen Werkzeug *MoVasE - Modelica Variable-structure Editor* umgesetzt. Abbildung 8 zeigt die Beschreibung des Austausches der Komponenten aus dem Spannungsquellenbeispiel in Abschnitt 3. Die Architektur der Anwendung besteht aus drei Schichten:

1. Benutzeroberfläche: Visualisierung der Modellstruktur und Notation der Strukturvariabilität unter anderem durch Zustandsübergangsdiagramme.
2. Metamodell: Struktur- und strukturvariable Informationen werden separat vom Ausgangsmodell und der Sprache gehalten. Analysen basieren auf Metamodellobjekten.
3. Sprachschicht: Einlesen der Modelle in die AST- und Metamodellform, Manipulation des ASTs, Kompilierung der Modes.

Plattform der Anwendung ist die *Eclipse Rich Client Platform* (<http://wiki.eclipse.org/Eclipse4/RCP>) implementiert wie der Rest der Anwendung in Java. Für die Benutzeroberfläche werden Frameworks wie das *Standard Widget Toolkit* (<http://www.eclipse.org/swt/>), *Piccolo2D* (<https://code.google.com/p/piccolo2d/>) für die Zustandsübergangsdiagramme und *SWT XY Graph* (<https://code.google.com/p/swt-xy-graph/>) für Plots eingesetzt. Das Metamodell ist mit dem *Eclipse Modeling Framework* (<http://eclipse.org/modeling/emf/>) realisiert. Analysen für Mappings und Initialisierungen sowie der Algorithmus der Modegenerierung basieren auf dem Metamodell und sind damit weitestgehend sprachunabhängig. Für die Auswertung einzelner Trigger-Ausdrücke wird *JEval* (<http://jeval.sourceforge.net/>) verwendet.

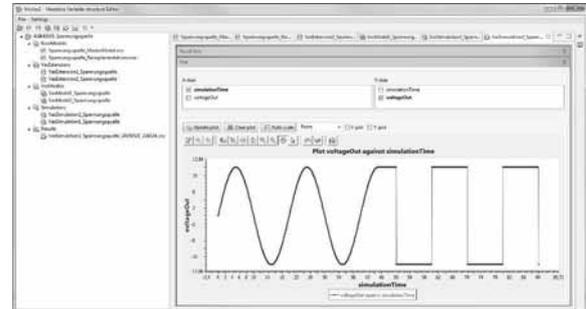


Abbildung 9: MoVasE Plot des Simulationsergebnisses.

Die Plugin-Architektur der Rich Client Platform ermöglicht es, die Funktionsgruppen in *Plugins* zusammenzufassen und lose zu koppeln. Die Kommunikation mit dem Plugin der Sprachschicht wird über Java Interfaces und Metamodellobjekte realisiert. Damit wird von der im Plugin verwendeten Sprache weitestgehend abstrahiert. Die Methode und das Werkzeug sind dafür konzipiert, auch andere Sprach-Plugins aufzunehmen. Anforderung an das Sprach-Plugin sind die Möglichkeiten, Modellcode in die AST-Form zu übersetzen und mit relevanten Elementen des Metamodells zu verbinden. Manipulationsoperationen müssen im Plugin zur Verfügung gestellt werden sowie zur Simulation die Möglichkeit, den manipulierten AST zu einem FMI-Modell zu übersetzen. *JModelica* (<http://www.jmodelica.org/>) erfüllt diese Anforderungen und bildet den Kern des ersten Sprach-Plugins. Es übernimmt das Übersetzen von Modelica-Code in AST-Form, das Übersetzen der manipulierten ASTs in ein simulationsfähiges Gleichungssystem und die Übersetzung zu einem FMI-Modell.

Die Beschreibung der Strukturvariabilität wird durch Wizards, Analysen und Hinweise unterstützt. Jeder Schritt der Entwicklung erzeugt Artefakte, die vorhergehende Artefakte referenzieren. Dadurch ist die Aufspaltung in Entwicklungszweige verschiedener strukturvariabler Ergänzungen für die gleichen Ausgangsmodelle möglich. Jedes Artefakt hat seinen eigenen Editor. Abbildung 9 zeigt das Simulationsergebnis des wie in Abschnitt 3 beschriebenen Modells für das letzte Artefakt der Methode, ein Result.

5 Stand und weiteres Vorgehen

Das vorgestellte Beispiel ist wie beschrieben im Tool umsetzbar. Darüber hinaus sind mehrere austauschbare Komponenten als getrennte Zustandsübergangsdiagramme realisierbar. Der Generierungsalgorithmus erzeugt entsprechend Modes und Transitionen. Transitionen zwischen diesen Modes ermöglichen immer nur den Austausch einer Komponente. Lösen mehrere Komponenten-Transitionen gleichzeitig aus, müssen mehrere Komponenten ausgetauscht werden. Der folgende Mode ist nicht direkt erreichbar. Durch eine anzugebende Priorisierung der Transitionen können die Übergänge nacheinander ausgeführt werden. Der Generierungsalgorithmus stellt sicher, dass es einen Pfad vom vorherigen Mode zum folgenden Mode mit beliebig vielen ausgetauschten Komponenten gibt. Priorisierung von Transitionen kann auch innerhalb eines Zustandsübergangsdiagramms verwendet werden, um Nicht-Determinismus zu verhindern. Ein weiteres implementiertes Mittel zur Vermeidung von Nicht-Determinismus ist ein Hit Counter für Trigger. Damit kann festgelegt werden, wie oft ein Trigger bzw. eine entsprechende Transition auslösen kann. Er kann als dynamische Priorisierung interpretiert werden, die sich im Verlauf der Simulation ändert.

Mit dem Werkzeug konnte gezeigt werden, dass die vorgeschlagene Methodologie umsetzbar ist. Die Nutzung der Sprache Modelica und die JModelica Platform ermöglichen es, Strukturvariabilität als Ergänzung zur Mächtigkeit der vorhandenen, klassischen Werkzeuge zu nutzen. Ausgangsmodelle können in etablierten Prozessen entwickelt und dann ergänzt werden. Die in den Abschnitten 3.4 und 3.5 beschriebene Mode-Generierung und Simulation dient nur der Evaluation des strukturvariablen Modells. Die Mode-Generierung, besonders vor der Simulation, erzeugt eine große Menge an Modellen, die sich nur gering unterscheiden. Für den praktischen Einsatz der Methode müssen Übersetzer entwickelt werden, die das Ausgangsmodell und Reimplementations sowie die strukturvariablen Metainformationen entgegennehmen und effizienter die Variationen auf einer tieferen Ebene als dem AST des Modellcodes generieren.

Das weitere Vorgehen sieht die Untersuchung komplexerer Ausgangsmodelle und Ergänzung kom-

plexerer Strukturvariabilität vor. Komplexere Strukturvariabilität bedeutet zum Beispiel bei einem Modewechsel die Simulation einige Zeitschritte vor Ende der letzten Simulation fortzuführen, um überschneidende und eventuell bessere Daten zu erhalten. Für komplexere Modelle werden Beispiele und Typen aus der Modelica Standard Library auf Austauschbarkeit untersucht. Die Anwendbarkeit der Methode auf andere komponentenorientierte Sprachen wird durch die Entwicklung neuer Sprach-Plugins untersucht. Um einen anderen Modellierungsansatz zu prüfen, ist geplant, *Matlab Simulink* (<http://www.mathworks.com/products/simulink/>) als Sprach-Plugin zu verwenden.

References

- [1] F. Breitenecker. *Development of simulation software - from simple ode modelling to structural dynamic systems*. Tagungsband 22nd European Conference on Modelling and Simulation, Vol. 5, S. 20–37, 2008.
- [2] The Modelica Association. *Modelica*® - A Unified Object-Oriented Language for Systems Modeling. <https://www.modelica.org/>
- [3] C. Nytsch-Geusen et al. *MOSILAB: Development of a Modelica based generic simulation tool supporting model structural dynamics*. Tagungsband 4th International Modelica Conference, S. 527-535, 2005.
- [4] D. Zimmer. *Equation-based modeling of variable-structure systems*. Eidgenössische Technische Hochschule Zürich, Schweiz, 2010.
- [5] C. Ptolemaeus. *System Design, Modeling, and Simulation Using Ptolemy II*. Ptolemy.org, 2014.
- [6] T. Pawletta et al. *A DEVS-Based Approach for Modeling and Simulation of Hybrid Variable Structure Systems*. In: Modelling, Analysis, and Design of Hybrid Systems, S. 107-129, 2002.
- [7] A. Mehlhase. *A Python framework to create and simulate models with variable structure in common simulation environments*. In: Mathematical and Computer Modelling of Dynamical Systems, V. 20, I. 06., S. 566–583, 2013.

Simulating a simple pneumatics network using the Modelica Fluid library

Patrick Drente¹, Peter Junglas²

¹Waskönig+Walter Kabel-Werk GmbH u. Co. KG

²PHWT Vechta/Diepholz/Oldenburg

peter@peter-junglas.de

The Modelica Fluid library provides a framework and a large number of components for thermo-fluid applications. This should simplify the task of modeling a large pneumatics system considerably. Nevertheless a lot of problems remain. Some of them are due to the lack of important components that are difficult to model, others are related to deficiencies of the used modeling software. But the really hard problems come from conceptual difficulties that are well known to experts and plague the practical modeler. The lessons learned from building and using a basic pneumatics library show what is feasible right now - and where future work is needed to make modeling of fluid systems easier.

1 Introduction

Pneumatic systems are used in industrial applications to distribute power in factory buildings. Such installations can be huge, consisting of large distribution networks, several compressors and many different kinds of consumers. Planning a new network or optimising an existing one involves a large number of parameters and possible configurations. A particular problem is the unknown timing behaviour of the consumers, which is often highly irregular. This obviously calls for an appropriate simulation tool.

The simulation of fluid systems with compressible media is a difficult task. A large step forward has been the introduction of the Modelica Fluid library [1] (in the following often abbreviated as MFL). It incorporates the fundamental behaviour of one-dimensional thermo-fluid systems and contains basic components for vessels, pumps, valves, pipes and other network elements. It uses the Modelica Media library [2], which allows to choose from a large list of predefined fluid media, compressible as well as incompressible.

Under these conditions Waskönig+Walter decided to start a simulation study in cooperation with the PHWT to optimise its existing pneumatics network. The simulation should help to find the reasons of bottlenecks

and to evaluate the effects of simple actions beforehand. The tool chosen was OpenModelica, which is open source and has a good support of the MFL. From the academic point of view two questions were of special interest: How easy is it to use the MFL for development of own libraries? What is the current status of OpenModelica concerning MFL?

In a first step a simple library has been constructed that contains all necessary components. It does not compete with any of the commercial pneumatics libraries available, but concentrates on the basic network. Notably missing are models for valves and actuators. A special feature is a focus on the tee branch, which is a common device in pneumatics networks and is astonishingly hard to model with reasonable accuracy.

The choice to work with OpenModelica had serious consequences. To make clear which of the difficulties are intrinsic to the modeled system and which are due to deficiencies of the software, the construction of the library and the models will be described in the following using Dymola, which is arguably the best platform for MFL applications at present. The particular problems coming from using OpenModelica are subject of a later section.

2 The PneuBib library

All components that are needed in the following to model simple pneumatics networks are combined in the PneuBib library. Two basic assumptions are made throughout: The temperature is constant and given by the ambient temperature, and the medium used is SimpleAir from the Modelica.Media library.

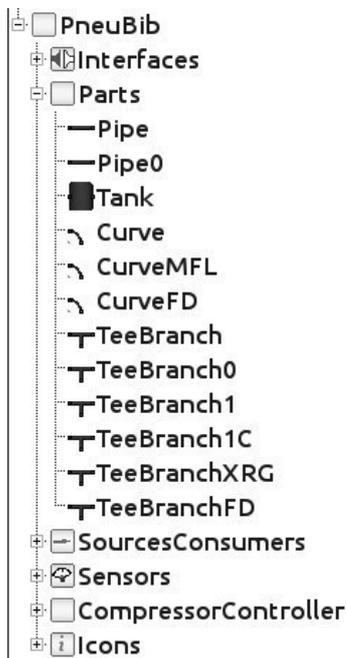


Figure 1: PneuBib library

The model names are in German, but will be translated here for better readability. The library consists of the following packages (fig. 1):

- `Interfaces` and `Icons` provide the common infrastructure of ports, base classes and functions.
- `Parts` contains the main components for the network and will be discussed in more detail below.
- `SourcesConsumers` includes the usual source blocks to define pressure, mass or volume flow as well as a generic consumer block. This is simply a wrapper around the linear valve component from the MFL connected to the ambient pressure. For convenience a special consumer is added that opens periodically with additional parameters for number of periods and start time.
- `Sensors` provide access to the values of pressure, mass or volume flow in a network model.
- `CompressorController` contains components for the modelling of the controlled compressors that are used at Waskönig+Walter.

The `Parts` sublibrary contains a few components that are wrappers around corresponding MFL blocks: The `Tank` is an isothermal version of the `ClosedVolume`, the `Pipe` consists mainly of a `DynamicPipe` with two nodes. The `CurveMFL` uses the `CurvedBend` from the MFL Fittings package. Unfortunately it doesn't work in OpenModelica, therefore a simpler version `Curve` has been added that relies only on the MFL function `dp_curvedOverall_DP` to compute the pressure loss.

Finally `Parts` provides several versions of a tee branch. They are the most complex part of PneuBib and will be explained in the following section.

3 Modeling a tee branch

The seemingly simple tee branch is difficult to model due to its several operational modes corresponding to the directions of the flows at its three ports: It can be used for splitting the main flow using the side branch either as one of the outgoing directions or for the incoming flow. One can join two flows with the combined outgoing flow either going straight or through the side branch. And for a compressible medium one could even use all three connections in the same direction, either outgoing or incoming.

The flow situations in all these cases are completely different – and always very complicated. Fortunately we are not interested in the exact flow but only in the overall pressure drops. Of course these depend on many details like the exact geometry of the pipes or the roughness of the inner pipe surfaces. But for our purpose of designing or analysing a pipe network, a simple approximation is often good enough. For this reason the PneuBib library contains several tee branch models with different levels of complexity and accuracy.

The MFL contains two models named `TeeJunctionIdeal` and `TeeJunctionVolume`, but they are not useful here, since they only describe

the mixing properties, not the pressure loss in the junction.

Fortunately this gap has been filled by the free package FluidDissipation [3], which contains a lot of functions for heat transfer and pressure loss in many important cases, among them the tee branch. For the computation of pressure losses the library relies heavily on [4], a large compilation of knowledge in form of formulae, tables and graphs. To implement these into numerically stable models they had to be complemented by a whole bunch of interpolation and regularisation formulae and combined with sophisticated schemes to discriminate between the different modes of the tee junction.

In addition to the necessary functions the FluidDissipation library contains a ready-to-use example component of a tee branch. This has been incorporated into the PneuBib as TeeBranchFD, which adds only a wrapper to fix many of the parameters that are not used in the pipeline context. The TeeBranchXRG component is a simplified version that uses the pressure loss functions directly.

A simpler version is TeeBranch1, which is reduced in three ways: First it implements only the two modes that are used in the following, namely splitting and joining along the straight direction. Second, it neglects all density changes and uses only the density at the straight input for all computations. And finally it computes the pressure drops by using only simple interpolation polynomials for the pressure drop coefficients ζ_i as functions of the volume flow ratio $Q_{\text{branch}}/Q_{\text{combined}}$. The task of finding an appropriate polynomial is not easy – not because it is hard to find one, but because there are many published versions. Fig. 2 gives an impression of the large variations in published values. The relations that are implemented in TeeBranch1 are based on data given in [5] for the split case and on the formulae in [6] for the join case. Fortunately the general conclusions for the models of interest here do not depend significantly on the details of the chosen curve.

The largest impact of the simplifications has the assumption of constant density. Therefore PneuBib contains the variant model TeeBranch1C that uses the appropriate variable densities for computing energy balances. Only for the computation of the ζ values it sticks to the mentioned polynomials – mainly because

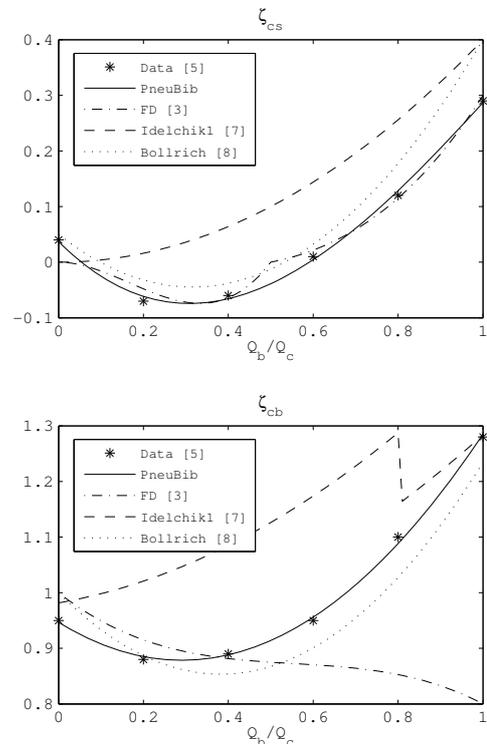


Figure 2: Pressure drop coefficients for the splitting tee branch

better results for compressible media are not available.

When dimensioning pneumatic networks in practice one often uses a very basic approach to include the pressure losses of tee branches: At each outgoing junction one adds a “virtual” substitutional pipe that reproduces the pressure loss of the tee branch [9]. By choosing an appropriate pipe length according to the dimensions of the junction one can get a rough approximation of the pressure losses. Values for the substitutional length can be found from vendors of pneumatic equipment, e. g. at [10]. The basic component TeeBranch implements this idea.

4 Testing the tee branch components

Several tests have been performed to check the basic functionality and to compare the different models of the tee branch. Fig. 3 presents one of the test models.

It is used to measure the pressure drops in the joining case, where a given time varying mass flow is inserted at the left port and a constant mass flow at the side port. The outgoing port on the right is connected to a fixed pressure.

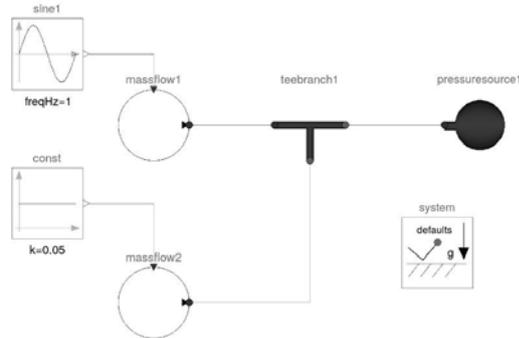


Figure 3: Model for testing a tee branch component

The resulting pressure drops from the straight and the side connection to the output are compared in fig. 4 for the three models TeeBranchFD, TeeBranch1 and TeeBranch. The curves show a good qualitative agreement, their differences are to be expected considering the variability of the different underlying data.

One important thing that has been learned here is how to choose the parameters for the basic TeeBranch: The values given for the substitutional length vary by more than 25 % (e.g. between [9] and [10]), the roughness k of the substitutional pipe is not given at all. But the test results have shown that the roughness has a considerable influence. Therefore the lengths have been chosen according to [10], which includes values for the straight direction as well, and the value for k has been adapted to reproduce approximately the results of the other components. The resulting value of $k = 0.25$ mm seems reasonable being in the range for used steel pipes that is given in [11].

Similar tests in the split case have an unexpected behaviour: The elaborate models show an increasing pressure in the straight direction, whereas the simple TeeBranch gives a pressure drop. A second thought explains this phenomenon: The pressure rise is the dynamical result of the velocity drop due to the splitting of the flow, an effect that is not incorporated in the simple “substitutional pipes” model of TeeBranch.

But this doesn't make the simple model useless, there are two different ways how to cope with it: First one

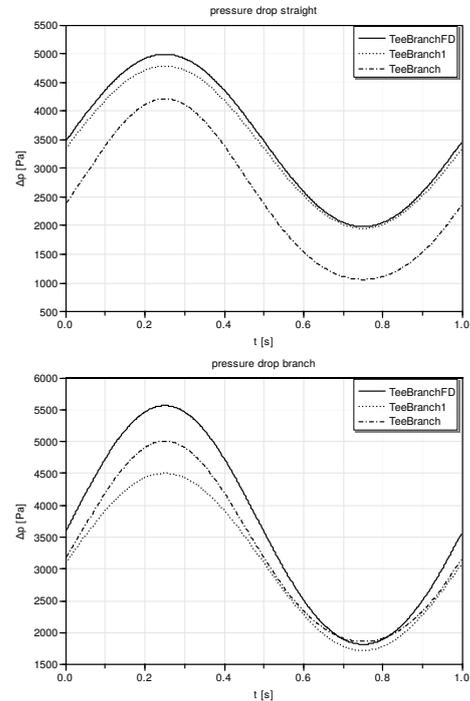


Figure 4: Comparison of the pressure drops in join mode

could just use it, including the pressure drop. After all the substitutional lengths (for the split situation!) are a reasonable rule of thumb coming from practical experience. Most likely it includes dissipative effects that have been neglected here so far. If one wants to reproduce the results of the other tee branch models instead, one can get a pressure rise just by using a negative length of the substitutional pipe. It is rather unexpected that this simple idea works with Modelica's DetailedPipeFlow model, but so it does! In the following the TeeBranch will be used with parameter values that roughly reproduce the results of the other models.

To see how the different tee branch models perform as part of a network, in the next test one short straight pipe is inserted between the tee branch and the pressure source. This leads to rapid pressure oscillations with amplitudes of several bar, which are strongly damped and converge to the expected result after a few seconds. Their origin can be easily traced back to the start values: The System component defines a default start pressure for all blocks that is set initially to the ambient pressure of about one bar. Changing

this to the value given by the pressure source reduces the amplitude of the oscillations to 0.2 bar. Setting the initial mass flow through the pipe to its steady-state value the amplitude goes down to 0.03 bar, which is in the order of the expected pressure losses. To get rid of the remaining oscillations too one had to supply more precise initial pressure values at all three ports, which are generally not known beforehand.

For the final test the position of the pipe has been changed: It is now inserted at the opposite side of the tee branch, directly connected to the mass flow source. In this case the simulation stops after a very short time with an error: The Newton solver is not able to produce reasonable initial values. Trying to fix more initial conditions doesn't help at all, apparently very precise values are needed here for the solver to converge. The problem is the same for all tee branch models in PneuBib, even for the simple TeeBranch. Only for the TeeBranch with positive substitutional pipe lengths the solver works fine and the results are as expected, but of course with a pressure drop along the tee branch instead of a rise.

A way how to cope with difficult initialisation problems has been proposed in [12]: One substitutes the problematic component with a simpler version that is used only for initialisation. Using a homotopy, i.e. a continuous path from the simple to the complete model, one adapts the initial values gradually, until proper values for the final model are found. This method has been applied successfully in different contexts, especially for thermo-fluid models [13].

According to this idea several simplified versions of the model have been developed, using a heavily reduced tee branch component, a pipe with a linear pressure drop law, a smoothly rising mass flow, a pressure source instead of one flow source or combinations thereof. All of them worked fine without the additional pipe, none of them lead to converging of the Newton solver. The only remaining tee branch model that actually works is the simple TeeBranch with positive substitutional pipe lengths.

5 Simulating complete networks

After the basic PneuBib library has been built and tested one can finally turn to the modeling of con-

crete pneumatics networks. A basic example consisting of several pipes, curves and branches together with a controlled compressor, a few consumers and an auxiliary tank is shown in fig. 5. According to the findings of the last section all branches are simple TeeBranch components with positive substitutional lengths given by [10]. If one replaces just one of them with one of the more elaborated models the solver can't find initial values.

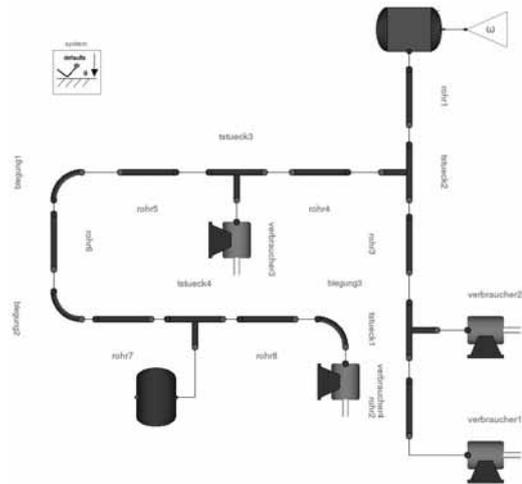


Figure 5: Model of a pneumatics network

Nevertheless such a model can be used to answer some of the questions that appear in real networks, e.g.:

- How large are the pressure drops in several parts of the network and where are the bottlenecks?
- Can the compressor provide the necessary mass flows everywhere, especially in the case of two neighbouring consumers with large demands?
- What size and position should auxiliary tanks have to buffer peak demands?

As a concrete example the situation at the two consumers on the lower right of fig. 5 is considered. Their timing behaviour is shown in the upper graph of fig. 6: Both are used periodically with the same frequency, but different start time, leading to a small overlapping period, where both are working simultaneously. The diagram in the middle of fig. 6 displays the resulting pressures at both positions: When only one is used the

pressure drops by about 0.15 bar, and it goes down by almost 0.4 bar, when they are working both. To resolve this problem an auxiliary tank can be installed between the two consumers. The resulting behaviour can be seen in the lower diagram of fig. 6: The total pressure drop is now reduced to almost half of the previous value.

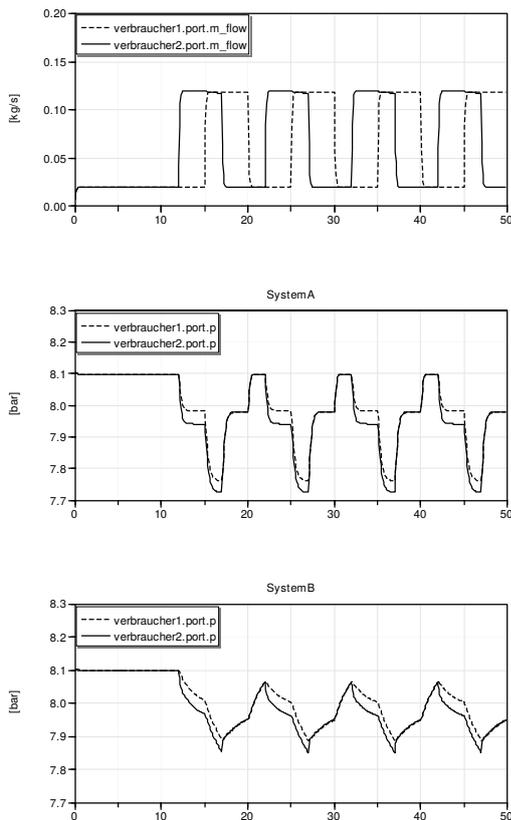


Figure 6: Simulation results of the example network

Large systems with more than 60 components and 5000 equations have been studied in this way. In spite of the severe limitations of the tee branch model their results have been used successfully to improve a real pneumatics network.

6 Working with OpenModelica

As has been mentioned in the introduction all simulations should be performed with the open source program OpenModelica. Though its user interface is not

as elaborated as those of commercial programs, its Modelica engine works generally very well in a wide range of applications [14]. Furthermore in the 1.9.1 release, which is the base of this investigation, it provides much better support of the Modelica Fluid library than most commercial programs except for Dymola. For this reason it seemed to be a cheap but promising alternative.

First tests using only MFL components showed promising results, most of the corresponding “wrapped” PneuBib components like Tank and Pipe worked as well. Only the Curve showed a strange behaviour, the simulation lead to undefined values for the mass flow. Using instead the corresponding component BendFlowModel from the FluidDissipation library made things worse: Now the simulation lead to an error at initialisation. A working model could be constructed by simplifying the MFL version: Instead of using the function `dp_curvedOverall_MFLOW` to compute the mass flow from the pressure difference and taking into account the different densities and viscosities at both ends, it uses the inverse `dp_curvedOverall_DP` and the density and viscosity only at one end. This worked in OpenModelica and lead to almost identical results in all cases of interest here (as verified later using Dymola).

The real challenge was to find a tee branch model that works in OpenModelica. Tests with the example component that is included in the FluidDissipation library did not succeed, the solver produced an error while flattening the equations. Simplifying this model by using the pressure loss functions directly did not remove the problem. Only after reducing the complexity largely one arrived at a working component that is included in PneuBib as TeeBranch1 and described above. That the structure of this model is on the brink of OpenModelicas capabilities becomes apparent, when one tries to remove some of its limitations: Using TeeBranch1C that includes some effects of the varying density, the simulation stops after a short time and produces very strange results that are due to completely wild initial conditions.

Since due to the initialisation difficulties all the problematic components could not be used in the final models anyhow, the prospects were good that the real pneumatics network could be analysed. Unfortunately in the case of the complete model with more than

100 components and 10000 equations OpenModelica gives up due to sheer size, the Modelica compiler crashes in an early phase. Several simplified versions have been tried with nonconclusive results: Some models with 80 components and 7000 equations run for a short simulation time, before the compiler stops with an error, other much smaller models crash immediately. A reduction of the network to 60 components and 5000 equations has lead to a model that generally runs long enough to produce interesting results even after several structural modifications or parameter changes. It was the basis of the final investigations, which lead to improvements of the real pneumatics system.

7 Conclusions

The modeling and simulation of the “simple” pneumatics network turned out to be much harder than had been expected at the beginning. This is a consequence of several different problems:

- The MFL components are very complex, they contain many parameters and subsystems that are difficult to understand for an unexperienced user. This problem gets much worse if one tries to create similar components from scratch.
- Some important components are missing, especially for the tee branch. The FluidDissipation library is a useful addition here, but it is not easy to use either. Especially the documentation and presentation of ready-to-use components leaves room for improvement.
- The fundamental problem of initialisation seems to be still far from being solved. Maybe the homotopy method is a feasible approach, but at least the authors were not able to find a working solution here.

The decision to use OpenModelica did not help either: Though it generally works fine in simple standard situations, it still has serious problems with models that are very complex or very large. Especially it did not cope well with the FluidDissipation library. In combination with the intrinsic problems listed above this lead to endless debugging sessions where it was al-

most impossible to isolate the reason of a specific failure.

Nevertheless in the end it all worked out: The industrial partner has now a working tool that helps to optimise his pneumatics installation. It has a sufficient accuracy and is open source. This miracle came about by simplifying the original system dramatically until all difficulties disappeared. Along the way one got rid of the limitations of the tool.

Is the MFL ready for end users? Principally yes – but only, if you are willing to accept simplified solutions and are prepared to work very hard!

References

- [1] R. Franke, F. Casella et al. *Standardization of Thermo-Fluid Modeling in Modelica.Fluid*. Proc. 7th Int. Modelica Conference, Como, p. 122-131, 2009.
- [2] F. Casella, M. Otter et al. *The Modelica Fluid and Media Library for Modeling of Incompressible and Compressible Thermo-Fluid Pipe Networks*. Proc. 5th Int. Modelica Conference, Vienna, p. 631-640, 2006.
- [3] T. Vahlenkamp, S. Wischhusen. *FluidDissipation for Applications – A Library for Modelling of Heat Transfer and Pressure Loss in Energy Systems*. Proc. 7th Int. Modelica Conference, Como, p. 132-141, 2009.
- [4] I. E. Idelchik. *Handbook of hydraulic resistance*. Jaico Publishing House, Mumbai, 3rd ed. 2006.
- [5] E. R. Rodriguez. *Technische Hydraulik. Vorlesungsskript*, FH Frankfurt. Online: <http://slideplayer.org/slide/209801/> (Aufruf 2015-04-15).
- [6] D. Aigner. *Gleichung zur Berechnung der hydraulischen Verluste der Rohrvereinigug - kalibriert mit Ergebnissen numerischer und physikalischer Modelle*. 3R-international, 47/1, 2008.
- [7] I. E. Idelchik. *Handbook of hydraulic resistance*. Israel Program for Scientific Translations, Jerusalem, 1966.

- [8] G. Bollrich. *Technische Hydromechanik, Band 1: Grundlagen*. Beuth, Berlin, 7th ed. 2013.
- [9] U. Bierbaum, J. Hütter. *druckluft kompendium*. Weka Business Medien, Darmstadt, 6th ed. 2004.
- [10] KAESER KOMPRESSOREN GmbH. *Berechnung des Druckabfalls*. Online: http://www.kaeser.at/Online_Services/Toolbox/Pressure_drop/default.asp (Aufruf 2015-04-15).
- [11] W. Bohl, W. Elmendorf. *Technische Strömungslehre*. Vogel Business Media, Würzburg, 15th ed. 2014.
- [12] M. Sielemann, F. Casella et al. *Robust Initialization of Differential-Algebraic Equations Using Homotopy*. Proc. 8th Int. Modelica Conference, Dresden, p. 75-85, 2011.
- [13] F. Casella, M. Sielemann, L. Savoldelli. *Steady-state initialization of object-oriented thermo-fluid models by homotopy methods*. Proc. 8th Int. Modelica Conference, Dresden, p. 1-11, 2011.
- [14] P. Junglas. *Praxis der Simulationstechnik*. Europa-Lehrmittel, Haan-Gruiten, 2014.

Modellierung elektrischer Antriebe mit Modelica

Leo Gall¹, Anton Haumer¹, Christian Kral²

¹LTX Simulation GmbH

²www.EDrives.eu

Leo.Gall@ltx.de, Anton.Haumer@edrives.eu, Christian.Kral@edrives.eu

Simulation ist ein unverzichtbares Hilfsmittel bei der Auslegung komplexer Systeme mit elektrischen Antrieben. Abhängig vom Zeitpunkt im Entwicklungsprozess und von der Aufgabenstellung sind unterschiedliche Abstraktionsebenen erforderlich: Beurteilung der Systemarchitektur, Untersuchung des Energie- und Leistungsverbrauches, Auslegung der Regelung, u.a.m. Die neue Modelica-Bibliothek EDrives bietet drei Abstraktionsebenen für elektrische Frequenzrichter: quasistationäre (unter Vernachlässigung der elektrischen transienten Vorgänge), mittelwertbildende (sinusoidale, unter Vernachlässigung der Schalteffekte) sowie schaltende Modelle – womit unterschiedliche Anforderungen erfüllt werden. Diese Umrichtermodelle können die Modelle für elektrische Maschinen aus der Modelica Standard Library direkt anspeisen: Magnetic.FundamentalWave sowie die neue Modelica-Bibliothek Magnetic.QuasiStatic.FundamentalWave.

1 Einleitung

Die Auslegung von Systemen mit elektrischen Antrieben wird wesentlich erleichtert durch die Anwendung von Simulationen, die das Verifizieren des Entwurfes gegen die Anforderungen schon in frühen Entwurfsstadien erlaubt [1]. Verschiedene Auslegungsaufgaben erfordern dabei verschiedene Detaillierungsstufen der Modelle:

Eine frühe Überprüfung des Konzepts kann auch ohne Kenntnis von Details wie z.B. die Regelung und den Pulsweiten-Algorithmus erfolgen. Die Bestimmung der Energie- und Leistungsaufnahme erfordert die exakte Berücksichtigung der Verluste und der Interaktion der Regelung mit dem Gesamtsystem. Dennoch können in diesen Fällen die Schalteffekte meist vernachlässigt werden. Für spezielle Untersuchungen wie z.B. Drehmomentpendelungen wird ein detaillierteres Modell verwendet, das die Einflüsse des Schaltens der Leistungshalbleiter berücksichtigt.

Deshalb ist ein flexibles modulares Konzept erforderlich, das alle Aufgaben unterstützt.

Die neue EDrives Bibliothek stellt Umrichtermodelle in drei Detaillierungsstufen zur Verfügung:

- Quasistationäre Umrichter können mit quasistationären Maschinenmodellen verbunden werden. Elektrische transiente Vorgänge werden vernachlässigt, aber der Einfluss der Verluste wird richtig erfasst. Diese Modelle haben die höchste Simulationsgeschwindigkeit und Robustheit, was sie für Langzeit-Simulationen geeignet macht.
- Sinusoidale Umrichtermodelle können mit transienten Maschinenmodellen verbunden werden. Schalteffekte werden vernachlässigt, aber die Interaktion der Regelung mit dem Gesamtsystem kann mit hoher Performanz getestet werden. Der

Einfluss der Verluste wird richtig erfasst. Die Bestimmung der Gleichstrom-Leistung basiert auf einer Leistungsbilanz.

- Schaltende Umrichtermodelle bieten den höchsten Detaillierungsgrad, was langsamere Simulationsgeschwindigkeit bedingt. Mit diesen Modellen können der Einfluss der Schaltungen der Leistungshalbleiter sowie neue Algorithmen zur Pulsweitenmodulation untersucht werden.

Die EDrives Bibliothek unterstützt nahezu beliebige Phasenzahlen m und ist daher vollkommen kompatibel mit den Modelica-Bibliotheken Magnetic.FundamentalWave und der neuen Magnetic.QuasiStatic.FundamentalWave. Phasenzahlen $m=2^n$ werden derzeit jedoch noch nicht unterstützt. Gründe dafür werden in [2] und [3] diskutiert.

Die modulare Trennung der Umrichter- und Maschinen-Modelle erlaubt den Austausch von Basismodellen gegen höher entwickelte Modelle der Maschinen, der Leistungselektronik und der Regelung. So kann z.B. das Maschinenmodell mit einem thermischen Modell der Maschine und des Kühlsystems gekoppelt werden und erlaubt die Untersuchung des thermischen Verhaltens des Systems über einen kompletten Lastzyklus.

Die Parameter des kompletten Antriebes werden in einem parameter record zusammengefasst. Dieser parameter record beinhaltet je einen record für die Maschine, für die Leistungselektronik und die Regelung. Die modulare Struktur ermöglicht dem Benutzer, parameter records für verschiedene Maschinen und Antriebe mit wenig Aufwand auszutauschen.

In der derzeitigen Version beinhaltet die EDrives Bibliothek die Regelungsstrukturen für permanenterregte Synchronmaschinen, synchrone Reluktanzma-

schinen und Asynchronmaschinen mit Kurzschlussläufer. Für künftige Versionen ist auch die Implementierung von Netzstromrichtern geplant.

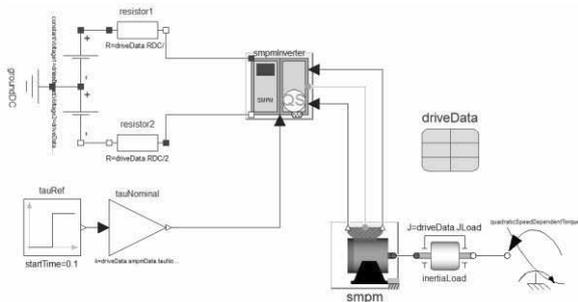


Abbildung 1. Permanenterrregte Synchronmaschine mit Umrichter

2 Antriebskonzepte

Jedes Maschinenmodell der EDrives Bibliothek beinhaltet ein Maschinenmodell der Modelica Standard Library (MSL) sowie zusätzliche Sensoren für Temperatur und Rotorposition. Die Schnittstellen sind elektrische, mechanische und thermische Konnektoren sowie die Sensorsignale, welche zum Umrichter verbunden werden.

Das Umrichtermodell besteht aus Leistungselektronik (einem Gleichstrom – Wechselstrom – Konverter), Sensoren für Ströme und Spannungen, der Regelung und dem Algorithmus für Pulsweitenmodulation. Der elektrische Mehrphasenausgang wird mit der Maschine verbunden, der Gleichstromeingang mit einer externen Gleichstromversorgung.

Ein Beispiel einer permanenterrregten Synchronmaschine samt Umrichter zeigt Abbildung 1. Eine detaillierte Beschreibung der Komponenten erfolgt in den Abschnitten 3 - 6.

3 Maschinen und Sensoren

Die EDrives Bibliothek bietet Hüllmodelle für jeden Maschinentyp wie in Abbildung 2 gezeigt:

- Eine Instanz eines quasistationären oder transienten Modells aus der MSL
- Einen Klemmenkasten
- Einen Sensor für die Rotorposition
- Temperatursensoren
- Den parameter record

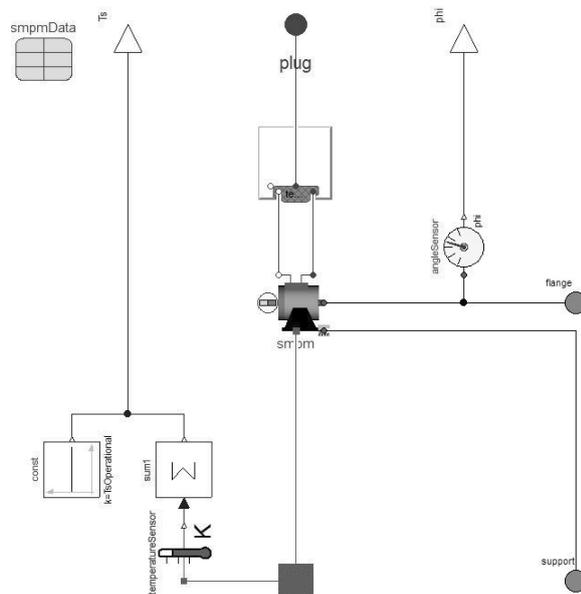


Abbildung 2. Hüllmodell für eine transiente Maschine

Der parameter record erlaubt es, alle Maschinenparameter mit nur einem record einzustellen. Das Hüllmodell propagiert die Parameter in das Maschinenmodell. Dieses Konzept erlaubt auch den Austausch von quasistationären und transienten Maschinenmodellen, da die parameter records und alle Konnektoren bis auf die elektrischen Anschlüsse kompatibel sind. Alle anderen Konnektoren inklusive der Sensorausgänge sind für beide Detailstufen der Modelle gleich. Beide Modelle berücksichtigen folgende Verluste: temperaturabhängige Wicklungsverluste, Eisenverluste (derzeit eingeschränkt auf Wirbelstromverluste), Reibungsverluste, Zusatzverluste sowie Verluste im Permanentmagneten – siehe [4] und [5]. Alle Verluste werden entweder intern bei fixer Betriebstemperatur dissipiert oder mit einem externen thermischen Modell über den thermischen Anschluss ausgetauscht. Ein externes thermisches Modell erlaubt die Simulation der aktuellen Betriebstemperaturen über einen Lastzyklus wie beispielsweise in [6] und [7] gezeigt.

4 Umrichter und Parametrierung

Jedes Umrichtermodell leitet sich von der selben Basisklasse ab, alle Modelle haben dieselben Anschlüsse mit Ausnahme der elektrischen Anschlüsse. Auch derselbe parameter record wird benutzt. Daher ist der Austausch verschiedener Detaillierungsgrade des Umrichtermodells leicht möglich, wobei der Detaillierungsgrad von Umrichter und Maschine jedoch passen zu wählen ist: ein quasistationärer Umrichter kann mit einer quasistationären Maschine gekoppelt werden, eine transiente Maschine kann von

einem sinusoidalen oder schaltenden Umrichter gespeist werden.

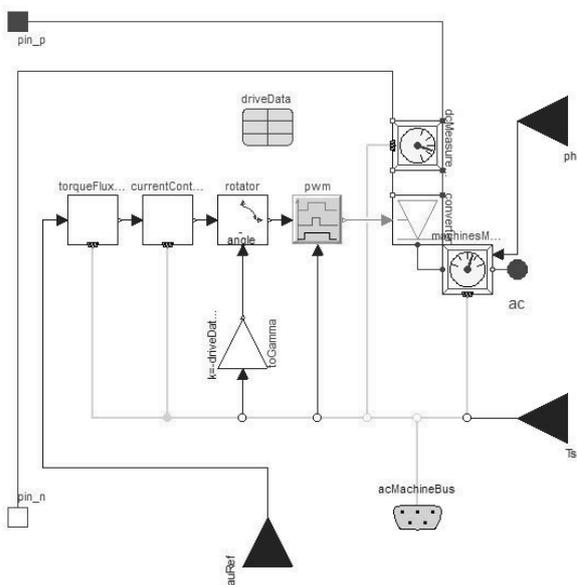


Abbildung 3. Schaltendes Umrichtermodell

Das in Abbildung 3 gezeigte Umrichtermodell existiert in drei Detaillierungsgraden:

- Das quasistationäre Modell basiert auf einer Leistungsbilanz zwischen Gleichstrom- und Multiphasen-Anschluss.
- Das sinusoidale Modell legt sinusförmige Spannungen an den Ausgang und stellt Leistungsgleichheit zwischen Gleichstrom- und Multiphasen-Anschluss sicher.
- Das schaltende Modell besteht aus m Umrichterphasen mit je zwei elektronischen Schaltern und zwei Freilaufdioden.

Messwerterfassung und Signalkonditionierung von Gleichstrom- und Wechselstromgrößen ist ein zwei Modellen gekapselt. Der Gleichstromsensor erfasst Gleichstrom und –spannung am Eingang, zusätzlich werden gefilterte Spannung, Strom und Leistung bestimmt. Die Filtereinstellungen werden automatisch an die Schaltfrequenz angepasst. Der Wechselstromsensor erfasst Ströme, Spannungen und Leistung, zusätzlich werden die Raumzeigergrößen (d- und q-Komponente) im drehfeldfesten Koordinatensystem berechnet. Alle Größen werden ebenfalls gefiltert, um schaltfrequente Anteile zu unterdrücken. Alle Messwerte werden ebenso wie die aktuellen Temperaturen und die Rotorposition über einen Signalbus kommuniziert.

Ebenfalls enthalten ist die Regelung, wie in Abbildung 3 gezeigt. Die Regelung basiert auf feldorientierter Regelung (FOR). Dabei werden die elektrischen Größen durch eine Komponente in Richtung

des Magnetfeldes (d-Komponente) und senkrecht dazu (q-Komponente) repräsentiert. Dafür wird die erfasste Rotorposition benutzt. Die d- und q-Komponente entsprechen dem Real- und Imaginärteil des Raumzeigers im drehfeldfesten Koordinatensystem. Die aktuelle Lage und Amplitude des Magnetfeldes wird im internen Maschinenbeobachter bestimmt. Dieser Beobachter schätzt auch das elektrische Drehmoment, wie in jedem realen Umrichter. Die Qualität der Regelung hängt von der Qualität des Maschinenbeobachters ab.

Für Maschinen mit Phasenzahlen größer oder gleich 3 werden elektrische Spannungen, Ströme und magnetische Flüsse durch Raumzeiger mit d- und q-Komponenten dargestellt. In der EDrives Bibliothek werden diese Raumzeiger durch Vektoren mit zwei Elementen repräsentiert.

Die Regelung ist hierarchisch aufgebaut; in jedem Detaillierungsgrad liegt am Eingang der Drehmoment- und Fluss-Regler wie in Abschnitt 6.2 beschrieben. Der Ausgang dieses Reglers sind die Sollwerte des d- und q-Stromes, wobei der q-Strom proportional zum Drehmoment und der d-Strom proportional zum magnetischen Fluss ist. Die Stromsollwerte werden vom Stromregler (siehe Abschnitt 6.3) geregelt und ergeben den Sollwert für den Spannungsraumzeiger, der in Phasenspannungen rücktransformiert werden muss.

Im quasistationären Umrichter werden die quasistationären Phasenspannungen an die quasistationäre Leistungselektronik weitergegeben.

Der sinusoidale Umrichter beinhaltet einen transienten Stromregler, wie er auch im schaltenden Umrichter verwendet wird. Ein Verzögerungsblock modelliert den Schaltverzug einer Schaltperiode der Leistungselektronik. Nach Rücktransformation des Spannungsraumzeigers werden die transienten (sinusoidalen) Phasenspannungen von der sinusoidalen Leistungselektronik umgesetzt (Abbildung 4).

Der schaltende Umrichter nutzt denselben Stromregler. Der Spannungsraumzeiger wird in das statorfeste Koordinatensystem transformiert und an den Pulsweitenmodulations-Algorithmus weitergeleitet. Die resultierenden Schaltbefehle werden an die schaltende Leistungselektronik gelegt.

Der parameter record des Antriebs enthält den parameter record der Maschine und weitere Parameter zur Konfiguration des Antriebes. Die Reglerparameter werden – ausgehend von den Maschinenparametern – berechnet und an die Reglermodelle propagiert. Zur Feinabstimmung der Regler kann der Benutzer über Abstimmparameter eingreifen.

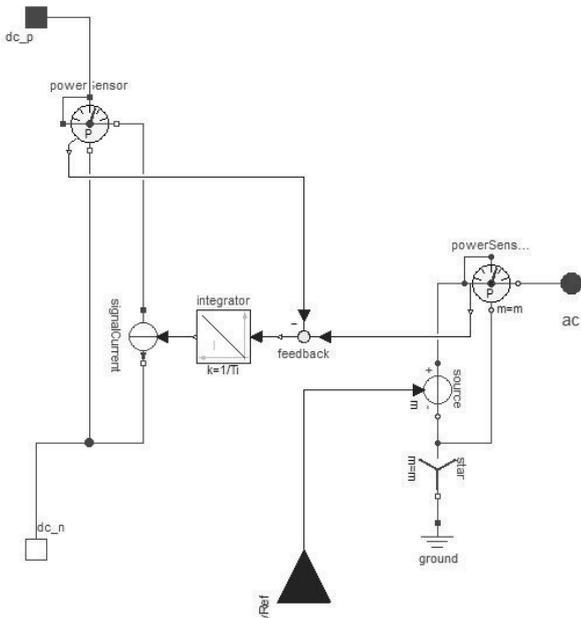


Abbildung 4. Sinusoidale Leistungselektronik

5 Leistungselektronik

Für die Leistungselektronik existieren drei Detaillierungsgrade:

Das quasistationäre Modell besteht aus einer quasistationären Spannungsquelle, zwischen Gleichstrom- und Multiphasen-Anschluss wird Leistungsgleichheit hergestellt. Die Verluste des Umrichters werden in künftigen Versionen durch eine Differenz in der Leistungsbilanz berücksichtigt werden.

Das sinusoidale Modell (Abbildung 4) legt die Sollwerte der Phasenspannungen an eine ideale Spannungsquelle. Wie beim quasistationären Modell wird der Gleichstrom aus der Leistungsbilanz bestimmt.

Das schaltende Modell benutzt zwei leistungselektronische Schalter und zwei Freilaufdioden aus der MSL pro Phase. Die Schaltbefehle sind logische Signale.

6 Regelung

Die Antriebsregelung basiert auf dem Prinzip der feldorientierten Regelung (FOR) und enthält:

- Einen optionalen Drehzahlregler
- Den Drehmoment- und Fluss-Regler
- Den Stromregler.

In allen Reglerkomponenten werden elektrische Größen durch Raumzeiger im drehfeldfesten Koordinatensystem repräsentiert, die über den Signalbus

kommuniziert werden. Der Eingang des Drehmoment-/Fluss-Reglers ist entweder das gewünschte Drehmoment oder wird vom Ausgang des optionalen Drehzahlreglers bestimmt.

6.1 Optionaler Drehzahlregler

Der externe Drehzahlregler ist als PI-Regler mit Anti-Windup implementiert. Das Reglermodell wird mit dem Signalbus des Antriebes verbunden. Die Parameter sind dem record speedControllerData, einer Komponente des parameter records des Antriebes, zu entnehmen und werden nach dem Symmetrischen Optimum bestimmt ([8]):

$$k_p = k_{Tune} \frac{(J_{Machine} + J_{Load})\omega_{Nominal}}{\tau_{Nominal}} \cdot \frac{2k_{Tune,iq}}{T_i}$$

$$T_i = T_{i,iq} \frac{4}{k_{Tune,iq}}$$

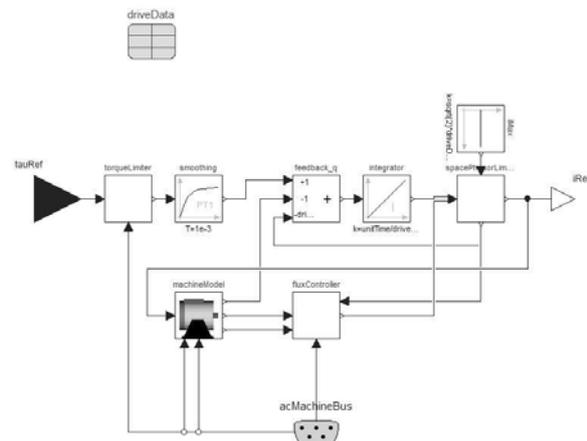


Abbildung 5. Drehmoment-/Fluss-Regler

6.2 Drehmoment- und Fluss-Regler

Der Drehmoment- und Fluss-Regler (Abbildung 5) limitiert das Solldrehmoment abhängig von der Ist-drehzahl. Im Feldschwächbereich nimmt das nutzbare Drehmoment mit dem Kehrwert der Drehzahl ab, im Grunddrehzahlbereich ist es konstant. Das Drehmoment, der Fluss und die Statorspannung werden mit einem internen Maschinenbeobachtermodell (Abschnitt 6.4) bestimmt, dessen Eingänge der Sollwert des Stromraumzeigers und die aktuelle Rotorposition sind. In einem realen Umrichter werden diese Größen ebenfalls mit einem Beobachter oder einem Modell, das Teil des Reglers ist, ermittelt. Die Abweichung zwischen dem Solldrehmoment und dem Drehmoment des Beobachtermodells stellt über einen schnellen Integralregler den drehmomentbildenden Strom i_q ein.

Der feldbildende Strom i_d wird durch den Flussregler mit einem einfachen Algorithmus – der leicht gegen einen komplexeren ausgetauscht werden kann – bestimmt. Einerseits wird die Spannung des Beobachters mit der maximal erlaubten und im aktuellen Betriebspunkt maximal erreichbaren Spannung verglichen. Weiters wird der aktuelle Fluss mit dem nominalen Fluss verglichen. Wenn die Spannung die maximale Spannung überschreitet, geht der Antrieb in Feldschwächung. Der Feldschwächbetrieb wird beendet, sobald der Fluss den Nennfluss übersteigt. Im Konstantflussbereich hält ein schneller Integralregler mit Hilfe des feldbildenden Stromes i_d den Fluss konstant, im Feldschwächbereich wird vom selben Regler die Spannung konstant gehalten.

Die Ausgänge des Drehmoment-/Fluss-Reglers sind die Sollwerte der d- und q-Komponente des Stromraumzeigers im drehfeldfesten Koordinatensystem. Der Stromraumzeiger wird auf den maximal zulässigen Strom (festgelegt durch den parameter record) begrenzt.

6.3 Stromregler

Der Stromregler besteht aus zwei PI-Reglern mit Anti-Windup. Die Spannungsgleichungen einer permanenterragten Synchronmaschine lauten im drehfeldfesten Koordinatensystem:

$$V_d = R_s I_d - \omega L_q I_q + L_d \frac{dI_d}{dt}$$

$$V_q = R_s I_q + \omega L_d I_d + L_q \frac{dI_q}{dt} + V_{PM}$$

ω ist die aktuelle mechanische Drehzahl, R_s ist der Statorwiderstand bei Betriebstemperatur und L_d und L_q sind die wirksamen Induktivitäten in d- und q-Achse. Das interne Beobachtermodell berechnet die beiden Spannungsabfälle $R_s I_d - \omega L_q I_q$ sowie $R_s I_q + \omega L_d I_d + V_{PM}$, welche zum Reglerausgang addiert werden. Dadurch bestimmt der Regler nur die transienten Abweichungen $\frac{dI_d}{dt}$ und $\frac{dI_q}{dt}$, in der Literatur wird diese Vorgangsweise auch Entkopplung genannt. Da die Maschinengleichungen ein PT1-Glied darstellen und zumindest die kleine Zeitkonstante der Filterung der Messwerte zu beachten ist, wird der Regler nach dem Betragsoptimum eingestellt:

$$k_{p,id} = k_{Tune,id} \frac{L_d \pi f_{filter}}{2}$$

$$T_{i,id} = \frac{k_{Tune,id} L_d}{k_{p,id} R_s}$$

$$k_{p,iq} = k_{Tune,iq} \frac{L_q \pi f_{filter}}{2}$$

$$T_{i,iq} = \frac{k_{Tune,iq} L_q}{k_{p,iq} R_s}$$

Die Reglerparameter werden im parameter record des Antriebes berechnet.

Der quasistationäre Stromregler ist einfacher aufgebaut: Da elektrische Transiente vernachlässigt werden, kann der limitierte Spannungsraumzeiger des internen Beobachtermodells direkt als Reglerausgang verwendet werden.

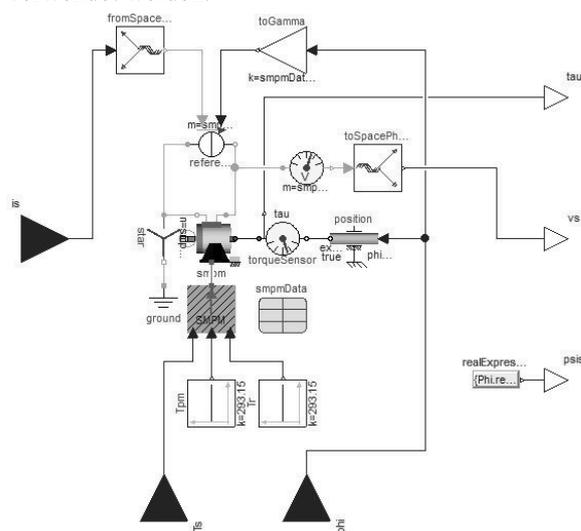


Abbildung 6. Internes Maschinenmodell

6.4 Maschinenbeobachter

Abbildung 6 zeigt das interne Maschinenmodell, welches den Spannungsraumzeiger, den Fluss und das Drehmoment basierend auf dem Sollwert des Stromraumzeigers und der aktuellen Rotorposition bestimmt. Durch die Verwendung eines quasistationären Maschinenmodells wie in [3] beschrieben ist das Beobachtermodell wesentlich modularer und flexibler aufgebaut als bei direkter Implementierung der quasistationären Gleichungen. Außerdem werden alle Eigenschaften (Temperaturabhängigkeit, Einfluss der Verluste) und Verbesserungen des quasistationären Maschinenmodells der MSL zugänglich gemacht.

6.5 Pulsweitenmodulation

Die Raumzeigermodulation (SVPWM; space vector pulse width modulation) wird in der Literatur als Stand der Technik angegeben, siehe z.B. [9] - [13].

Aktuell sind zwei PWM – Blöcke implementiert: Der erste Algorithmus ist auf Phasenzahl $m=3$ beschränkt; er tastet mit Schaltfrequenz den Sollwert des Spannungsraumzeigers und die aktuelle Gleichspannung ab und bestimmt die Einschaltdauer der beteiligten leistungselektronischen Schalter zur Erreichung des Spannungsraumzeigers durch Vergleich mit den benachbarten möglichen Schaltzuständen, siehe Abbildung 7. Eine Erweiterung auf Phasenzahl $m>3$ ist bereits in einem zweiten Block implementiert.

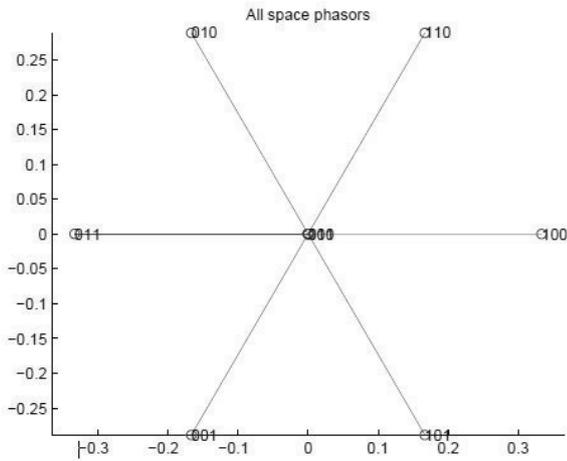


Abbildung 7. Schaltzustände in der d-q-Ebene für m=3 Phasen

7 Beispiele

Das erste Beispiel (Abbildung 1) zeigt eine quasistationäre permanenterrregte Synchronmaschine mit quasistationärem Umrichter. Für die angegebene Konfiguration werden die Simulationsergebnisse für die drei Detaillierungsstufen verglichen: Quasistationäre Maschine mit quasistationärem Umrichter, transiente Maschine mit sinusoidalem Umrichter und transiente Maschine mit schaltendem Umrichter.

Die Maschinenparameter werden aus den Maschinenmodellen der MSL entnommen (siehe Tabelle 1).

Phasenzahl	m	beliebig	
Polpaare	p	2	
Trägheit	J_r	0,29	kg m ²
Frequenz	$f_{S,Nominal}$	50	Hz
Leerlaufspannung	$V_{S,OpenCircuit}$	112,3	V
Nennspannung	$V_{S,Nominal}$	100	V
Nennstrom	$I_{S,Nominal}$	$100 \cdot \frac{m}{3}$	A
Statorwiderstand	R_S	$0,03 \cdot \frac{m}{3}$	Ω
Streuinduktivität	$L_{S\sigma}$	$\frac{0,1}{2\pi f} \cdot \frac{m}{3}$	H
d-Induktivität	L_{md}	$\frac{0,3}{2\pi f} \cdot \frac{m}{3}$	H
q-Induktivität	L_{mq}	$\frac{0,3}{2\pi f} \cdot \frac{m}{3}$	H

Tabelle 1. Maschinenparameter

Die Maschine ist ohne Dämpferkäfig ausgeführt. Alle Verluste außer den Wicklungswiderständen werden vernachlässigt. Die Maschine wird mit einem Lastträgheitsmoment und einem quadratisch von der Drehzahl abhängigen Lastdrehmoment belastet. Der Umrichter wird von einer Gleichspannungsquelle mit Innenwiderstand – ähnlich einer Batterie – gespeist. Zum Zeitpunkt $t=0,1$ s wird ein Sprung des Soll-drehmomentes von 0 auf 100 % Nennmoment eingepreßt. Nach der Beschleunigung erreicht der Antrieb einen Gleichgewichtszustand.

Die folgenden Abbildungen vergleichen die Ergebnisse einer quasistationären Maschine mit quasistationärem Umrichter, einer transienten Maschine mit sinusoidalem Umrichter und einer transienten Maschine mit schaltendem Umrichter.

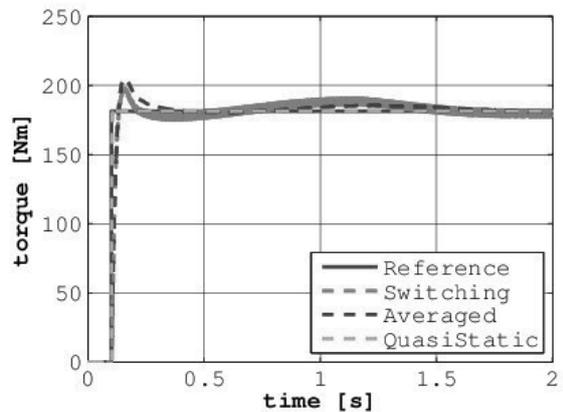


Abbildung 8. Beispiel 1: Drehmoment

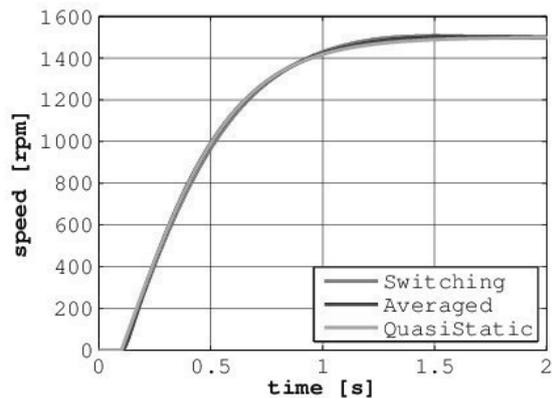


Abbildung 9. Beispiel 1: Drehzahl

In Abbildung 8 folgt das Drehmoment dem Sollwert in allen drei Detaillierungsstufen sehr gut. Das Drehmoment des schaltenden Modells zeigt einen mit der Schaltfrequenz (5 kHz) pulsierenden Drehmomentanteil. Die Drehzahlverläufe zeigen nur minimale Abweichungen.

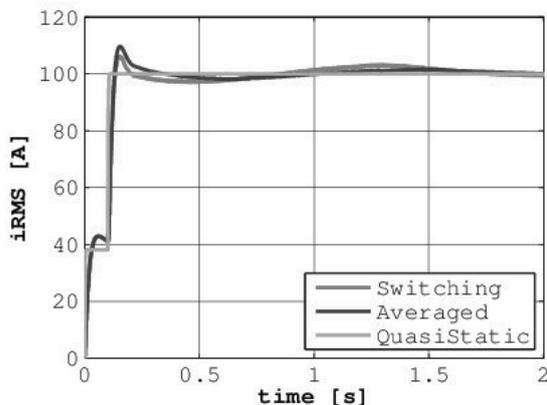


Abbildung 10. Beispiel 1: Länge des Stromraumzeigers

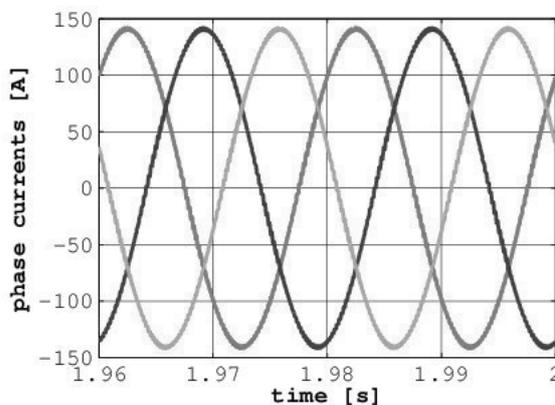


Abbildung 11. Beispiel 1: Phasenströme des schaltenden Umrichters

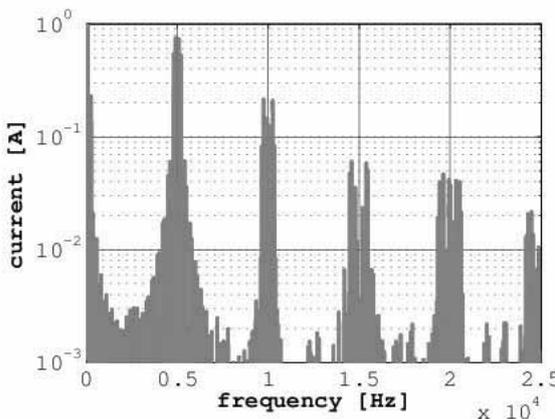


Abbildung 12. Beispiel 1: FFT der Ströme des schaltenden Umrichters

Der d-Strom steigt zu Beginn der Simulation an, um den Fluss aufzubauen; zum Zeitpunkt des Sollmomenten Sprungs steigt der q-Strom dann ebenfalls an (siehe Abbildung 10). Der Strom des schaltenden Umrichters zeigt eine Stromwelligkeit entsprechend

der Schaltfrequenz von 5 kHz. Die ungefilterten Phasenströme am Ende der Simulation werden in Abbildung 11 gezeigt, die Fast Fourier Transformation (FFT) des Stromes der Phase 1 in Abbildung 12. Neben der Grundschwingung sind die größten Anteile bei Vielfachen der Schaltfrequenz zu sehen, was zu erwarten war.

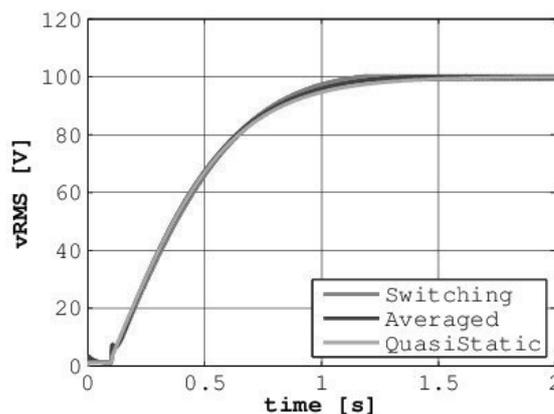


Abbildung 13. Beispiel 1: Statorspannung

Die gefilterte Statorspannung in Abbildung 13 steigt mit der Drehzahl an, übersteigt jedoch nicht die maximal erlaubte Spannung von 100 V.

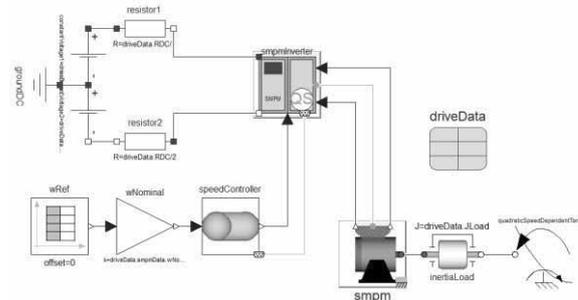


Abbildung 14. Beispiel 2: Drehzahlgeregelte permanentmagnetische Synchronmaschine

Das zweite Beispiel in Abbildung 14 ist abgeleitet vom Beispiel 1, jedoch mit einer Phasenzahl $m = 5$. Das Solldrehmoment wird vom externen Drehzahlregler bereitgestellt, die Solldrehzahl wird entsprechend einem Lastspiel aus einer Tabelle vorgegeben. Die Parameter des Drehzahlreglers werden im parameter record des Antriebs bestimmt. Die Simulationsergebnisse wurden mit einer transienten Maschine an einem sinusoidalen Umrichter gewonnen.

Abbildung 15 zeigt, dass das Drehmoment dem Solldrehmoment des Drehzahlreglers befriedigend folgt. Auch im Drehzahlverlauf (Abbildung 16) sind keine unerwarteten Abweichungen zu beobachten. Abbildung 17 zeigt alle fünf Phasenströme über den Zeitraum von einer Sekunde während der ersten Beschleunigungsphase. Das Modell funktioniert in beiden Dreh- und Energierichtungen zufriedenstellend.

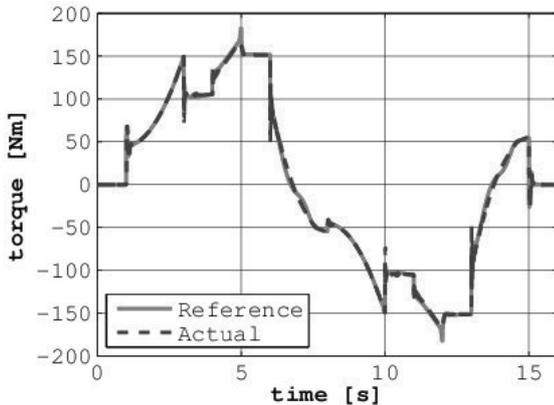


Abbildung 15. Beispiel 2: Drehmoment

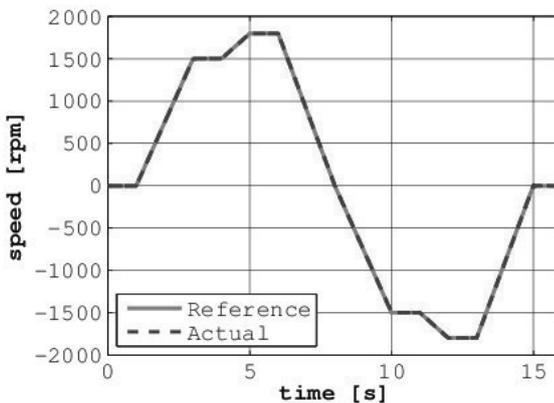


Abbildung 16. Beispiel 2: Drehzahl

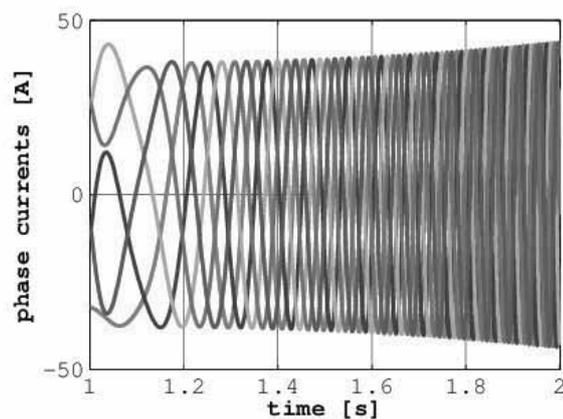


Abbildung 17. Beispiel 2: Phasenströme

8 Ausblick

Die EDrives Bibliothek soll als tool-unabhängige kommerzielle Bibliothek veröffentlicht werden. Das modulare Konzept erleichtert die bereits geplanten Entwicklungsschritte:

- Erweiterung des Leistungsteils und des Algorithmus zur Pulsweitenmodulation auf Multilevel-Konfigurationen.
- Einbau alternativer Flussregelstrategien.
- Die Implementierung von Diodengleichrichtern (quasistationär, sinusoidal und schaltend).
- Die Implementierung von aktiven Netzstromrichtern mit spannungsorientierter Regelung wie in [14] beschrieben. Die bereits implementierten Leistungselektronikmodelle und die Algorithmen zur Pulsweitenmodulation können benutzt werden.
- Die aktuell verwendeten Halbleitermodelle sind der MSL entnommen, welche die Leitverluste hinreichend abbilden. Verbesserungen zur Erfassung der Schaltverluste sind möglich. Für quasistationäre und sinusoidale Leistungselektronik sollen ebenfalls Leit- und Schaltverluste modelliert werden. Besonderes Augenmerk ist dabei auf eine benutzerfreundliche Parametrierung zu legen. Weiters sollen die Modelle der Leistungselektronik mit thermischen Anschlüssen ausgestattet werden, um die Erwärmung und Kühlung der Umrichter sowie den Einfluss ihrer Verluste auf das thermische Gesamtsystem simulieren zu können.
- Schließlich sind adaptive Regelungen denkbar, die Temperatureinfluss und Sättigung in die Betrachtung einbeziehen.

Die neue EDrives Bibliothek öffnet neue Anwendungsfelder in der Elektromobilität und Industrie: Quasistationäre Modelle mit höchster Performanz erlauben Langzeitsimulationen für Energieverbrauchsermittlung. Sinusoidale Modelle ermöglichen den Test der Interaktion zwischen Antriebsregelung und Gesamtsystem. Mit den detailreichsten schaltenden Modellen können Effekte der Welligkeit untersucht werden.

Die Bibliothek soll in einer kommerziellen Version veröffentlicht werden, mit quasistationären, sinusoidalen und schaltenden Umrichtern für permanenterregte Synchronmaschinen, synchrone Reluktanzmaschinen und Asynchronmaschinen mit Käfigläufer. Erweiterungen und Verbesserungen sind bereits geplant.

9 References

- [1] A. Haumer, T. Bäuml, and C. Kral, *Multiphysical simulation improves engineering of electric drives*, 7th EUROSIM Congress on Modelling and Simulation, September 2010.
- [2] C. Kral, A. Haumer, and R. Wöhrschimmel, *Extension of the fundamentalwave library towards multiphase electric machine models*, Modelica Conference, 2014.
- [3] C. Kral and A. Haumer, *New multi phase quasi static fundamental wave electric machine models for high performance simulations*, Modelica Conference, 2014.
- [4] A. Haumer, C. Kral, H. Kapeller, T. Bäuml, and J. V. Gragger, *The AdvancedMachines library: Loss models for electric machines*, Modelica Conference, 2009.
- [5] C. Kral and A. Haumer, *Object Oriented Modeling of Rotating Electrical Machines*. INTECH, 2011.
- [6] C. Kral, A. Haumer, and M. Plainer, *Simulation of a thermal model of a surface cooled squirrel cage induction machine by means of the SimpleFlow-library*, Modelica Conference, 2005.
- [7] A. Haumer, C. Kral, V. Vukovic, A. David, C. Hettfleisch, and A. Huzsvar, *A parametrization scheme for high performance thermal models of electric machines using Modelica*, MATHMOD VIENNA, 2012.
- [8] P.-I. H. Lutz and P. D. I. W. Wendt, *Taschenbuch der Regelungstechnik*, 5. erweiterte Auflage. Frankfurt am Main: Wissenschaftlicher Verlag Harri Deutsch, 2003.
- [9] M. J. Duran, F. Barrero, and S. Toral, *Multi-phase space vector pulse width modulation: Applications and strategies*, International Conference on Renewable Energies and Power Quality, 2007.
- [10] S. Halasz, *PWM strategies of multi-phase inverters*, 34th Annual Conference of IEEE Industrial Electronics, 2008.
- [11] D. Dujic, M. Jones, and E. Levi, *Generalised space vector PWM for sinusoidal output voltage generation with multiphase voltage source inverters*, International Journal of Industrial Electronics and Drives, vol. 1, no. 1, 2009.
- [12] M. Mengoni, *Modulation Techniques for Multi-Phase Converters and Control Strategies for Multi-Phase Electric Drives*. PhD thesis, University of Bologna, 2010.
- [13] D. G. Holmes and T. A. Lipo, *Pulse Width Modulation for Power Converters*. Wiley Interscience, 2003.
- [14] Haumer and C. Kral, *Modeling a mains connected PWM converter with voltage-oriented control*, Modelica Conference, 2011.

Sitzung 6

Raum 105

Grundlagen und Methoden I

(Sitzungsleitung: Christina Deatcu)

Zur Struktur von dynamischen Modellen für die Regelung von MIMO Prozesse

Bernhard Lampe

Universität Rostock

Towards a Newer Toolbox for Computer Aided, Polynomial Design of Sampled-Data Systems

Rudy Cepeda Gomez

Universität Rostock

Hybrid State Estimation of nonlinear index-1 DAE-Systems

Dirk Weidemann

Fachhochschule Bielefeld

Implementation and Simulation of Hybrid Systems Formulated as DEV&DES Using the QSS-based Simulator PowerDEVS

Franz Preyser

TU Wien

Zur Struktur von dynamischen Modellen für die Regelung von MIMO Prozessen

Bernhard P. Lampe
Universität Rostock
bernhard.lampe@uni-rostock.de

Sollen mathematische Modelle für performante Regelungen verwendbar sein, dann müssen bei der Modellbildung eine Reihe von Besonderheiten beachtet werden. Zum einen müssen die Modelle in gewissem Sinne invertierbar sein, was hinsichtlich ihrer Struktur und Komplexität enge Grenzen setzt. Im Vortrag wird gezeigt, dass darüber hinaus weitere Schwierigkeiten auftreten können, wenn die wahre Struktur des zu regelnden Prozesses im Modell auf bestimmte Weise verletzt wird. An Hand von einfachen Beispielen wird gezeigt, dass es sehr leicht zu derartigen Verletzungen kommen kann, etwa durch Parameteridentifikation aus Messdaten oder infolge von Rundungsfehlern. Eine höhere Genauigkeit der Modelle durch Erhöhung der Modellordnung kann, im Gegensatz zur offenen Kette, zur Instabilität des geschlossenen Regelkreises führen.

Keywords: MIMO control systems, Transfer matrices, State space realizations, Process identification, Structural instability

1 Einleitung

Prozesse mit mehreren Ausgangsgrößen, die von mehreren Eingangsgrößen beeinflusst werden, bezeichnet man als MIMO Prozesse. Insbesondere wird davon ausgegangen, dass jede Ausgangsgröße prinzipiell von jeder Eingangsgröße beeinflusst wird. Für die Regelung solcher Prozesse werden verschiedene Ansätze verfolgt [1, 2, 3, 4].

Zum einen sind es die sogenannten dezentralen oder autonomen Regelungen, wo jeweils eine Regelgröße durch eine Stellgröße geregelt wird. Die (ungewollten) gleichzeitigen Beeinflussungen der anderen Regelgrößen werden als Störungen angesehen, deren Wirkungen von den autonomen Regelkreisen genügend stark abgebaut werden muss. Diese simple Herangehensweise verzichtet von vornherein auf hohe Regelgüte, stellt andererseits aber vergleichsweise geringe Anforderungen an die Modellbildung, insbesondere wenn der MIMO Prozess stabil ist.

Der zweite Ansatz geht von einem Mehrgrößenregler aus, der prinzipiell immer alle Stellgrößen gleichzeitig bedient, auch wenn zum Beispiel nur eine Regel-

größe abweicht. Dieser Ansatz verspricht höhere Regelgüte, weil die Stellaktionen besser aufeinander abgestimmt werden können und nicht gegeneinander arbeiten. Dieses Potential wird aber nur wirksam, wenn genügend genaue Prozessmodelle beim Entwurf des MIMO-Reglers verwendet werden.

Insbesondere sollte die Struktur von Prozess und Modell passen. Als problematisch ist der Fall anzusehen, wenn das Modell eine höhere Ordnung als der Prozess hat, weil dann Größen geregelt werden, die in der Realität gar nicht existieren. Dadurch werden unnötige Aktionen ausgelöst, die mitunter die Instabilität des geregelten Systems zur Folge haben.

Wenn die Prozessmodelle durch experimentelle Prozessidentifikation gewonnen werden, kann eine zu große Ordnung des Modells durch Rundungs- oder Messfehler hervorgerufen werden, wenn die Struktur nicht a priori festgelegt wird. Eine nachträgliche Korrektur der Modellstruktur ist dann erforderlich. Diese Korrektur umfasst mehr als die Beseitigung von nicht steuerbaren oder nicht beobachtbaren Anteilen, wie es zum Beispiel die Matlab-Funktion `minreal` erledigt.

Da diese Phänomene bereits bei linearen zeitinvarianten (LTI) Prozessen auftreten, wird die strukturelle Instabilität daran erklärt. Darüber hinaus werden deren Auswirkungen auf Regelungen sowie Ansätze zur Korrektur der Modellstruktur besprochen. Hierbei wird neben den Begriffen Steuerbarkeit und Beobachtbarkeit auch das weniger bekannte Konzept der Normalität eine wichtige Rolle spielen [5, 6].

2 Darstellungen für LTI Systeme

Folgende Darstellungen für reale kontinuierliche Systeme werden betrachtet

Zustandsdarstellung

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (1)$$

mit Zustand $x \in \mathbb{R}^p$, Eingang $u \in \mathbb{R}^m$, Ausgang $y \in \mathbb{R}^n$ und konstanten Matrizen A, B, C, D passender Dimension. Wichtige Eigenschaften: Steuerbarkeit, Beobachtbarkeit, Stabilität (Dynamik), Zustandstransformation $\xi = Rx$.

Übertragungsmatrix Laplace Transf. von (1)

$$\begin{aligned} y(s) &= G(s)u(s) \\ G(s) &= C(sI - A)^{-1}B + D \end{aligned} \quad (2)$$

Formen für $G(s)$:

$$\text{Elementar} \quad \begin{bmatrix} G_{11}(s) & \dots & G_{1m}(s) \\ \dots & \dots & \dots \\ G_{n1}(s) & \dots & G_{nm}(s) \end{bmatrix} \quad (3)$$

$G_{ij}(s)$ ggf. in Pol-Nullstellen-Form

$$\text{Standard} \quad \frac{L(s)}{d(s)}, \quad \begin{matrix} \text{Polynommatrix} \\ \text{Polynom} \end{matrix} \quad (4)$$

$$\text{MFD} \quad M_l^{-1}(s)N_l(s), N_r(s)M_r^{-1}(s) \quad (5)$$

Polynommatrizen

Sprungantwort $y(t)$ bei $x(0) = 0, u(t) = 1(t)$

Frequenzgang $G(j\omega), -\infty < \omega < \infty$.

3 Wandlung von Darstellungen

Alle Darstellungen für LTI Systeme sind prinzipiell gleichwertig, wenn man nur die steuerbaren und beobachtbaren Anteile betrachtet. Sie sind aber unterschiedlich gut für verschiedene Aufgaben, etwa den Reglerentwurf, geeignet. Im Vortrag wird untersucht, wie sich Messfehler während der Prozessidentifikation oder Rundungsfehler bei der Konvertierung von Modelldarstellungen auswirken. Insbesondere wird die Rolle von Strukturinformationen diskutiert. Dazu werden unter einfachen Annahmen, die typischerweise bei der Prozessidentifikation erfüllt sind, die Auswirkungen auf die Prozessmodelle sowie Wirkungen im geschlossenen Regelkreis simulativ untersucht. Als Plattform wird MATLAB/SIMULINK benutzt, wobei solche Kommandos wie `step, impulse, lsim, frd, freqresp, tf, zpkr, residue, ss, ssdelay, minreal, svd` hinsichtlich ihrer Eignung für die Lösung der oben genannten Aufgaben betrachtet werden.

Literatur

- [1] R. E. Kalman. Mathematical description of linear dynamical systems. *SIAM J. Control*, A(1):152–192, 1963.
- [2] D. C. Youla, H. A. Jabr, and J. J. Bongiorno (Jr.). Modern Wiener-Hopf design of optimal controllers. Part II: The multivariable case. *IEEE Trans. Autom. Contr.*, AC-21(3):319–338, 1976.
- [3] T. Kailath. *Linear Systems*. Prentice Hall, Englewood Cliffs, NJ, 1980.
- [4] M. J. Grimble and V. Kučera, editors. *Polynomial methods for control systems design*. Springer-Verlag, London, 1996.
- [5] B. P. Lampe and E. N. Rosenwasser. Structural instabilities of frequency domain descriptions of multivariable sampled-data systems. In *Proc. 14th IFAC Triennial World Congr.*, volume D, pages 195–200, Beijing, P.R. China, 1999.
- [6] E. N. Rosenwasser and B. P. Lampe. *Multivariable computer-controlled systems – a transfer function approach*. Springer, London, 2006.

Towards a Newer Toolbox for Computer Aided, Polynomial Design of Sampled-Data Systems

Rudy Cepeda-Gomez¹, Bernhard Lampe¹

¹Institute of Automation, University of Rostock

rudy.cepeda-gomez@uni-rostock.de, bernhard.lampe@uni-rostock.de

This work presents recent steps taken in order to update the DIRECTSD toolbox for MATLAB. This toolbox realizes recently developed polynomial methods for the analysis and optimal design of sampled-data systems. Once released, the version under development will be compatible with the newest versions of MATLAB and include the possibility of working with both SISO and MIMO systems. The text describes the last published version, an interim version currently running and the updates planned for a future stable release. Some usage examples obtained with the interim version are also presented.

1 Introduction

Sampled-Data systems are systems in which a digital computer controls a continuous-time plant. This class of systems is widely used in almost any industry. Classical approaches to the design of controllers for such systems consider either a continuous time synthesis of a controller followed by its discretization, or a discretization of the plant followed by a controller synthesis in the discrete domain. Modern approaches to the design of optimal controllers are based in the so-called direct design methods, which take into account the continuous time behavior of the system.

The widely known *lifting* technique [1] allows the reduction of some hybrid optimization problems to similar problems for stationary discrete systems. This methodology was implemented in MATLAB as the *Sampled Data Control toolbox* [2, 3]. The lifting technique, however, is not applicable to some cases, e.g., to systems with time delays.

Another alternative for the direct optimal design of sampled-data systems is the frequency domain analysis based on the *Parametric Transfer Function* concept [4, 5]. This approach has some advantages over the lifting technique: it can be used even when the continuous plant has time delays and also allows to obtain the structure and order of the optimal controllers.

The DIRECTSD toolbox, in its initial version [6], was designed to implement these polynomial methods in the case of SISO systems. Its development generated a MIMO version [7], which uses the theory presented in [5]. Its latest version [8] was a mature project: albeit it focused on SISO systems, it included options to use either polynomial methods or the lifting technique to solve the problem and a comprehensive set of examples and demos integrated with the MATLAB help explorer.

The development of the DIRECTSD toolbox, however, halted with its version 3.0. After almost ten years since its last revision, the code became obsolete, mainly because of the change in the object-oriented programming (OOP) model introduced in MATLAB release 2012.

During the last year, an effort has been made within the institute of automation of the university of Rostock to update the code and to have, once again, a functional toolbox. The steps taken in this direction and the current development status of the DIRECTSD version 4.0 are presented in this work.

2 Structure of DIRECTSD 3.0

The DIRECTSD toolbox is currently in its version 3.0, which is available in [9]. The current code uses OOP

to define a new class to create and manipulate polynomial objects. The class `poln` and its methods are one of the most important parts of the toolbox. To model continuous and discrete LTI systems, the standard objects from the Control Systems Toolbox of MATLAB, i.e., `tf`, `zpk`, or `ss`, are used. Some functions for these objects were overloaded in order to change their behavior.

The toolbox provides functions to perform analysis and design of Sampled-Data systems using both the frequency domain methods presented in [4] and lifting technique [1]. Due to some unsolved numerical robustness problems encountered during the development of the MIMO toolbox only SISO controllers are considered in this version.

As the OOP model of Matlab changed with version 7.6 (Release 2008a) the syntax of class `poln` made it obsolete. This change also remove support of overloaded functions, making the methods newly defined for the standard objects unusable.

3 Interim: DIRECTSD 3.5

The first steps taken towards the newer version were aimed to recover a basic level of functionality while the new developer got familiarized with the structure of the code and its theoretical background. The result of this is an interim version, which is being used as a basis for further development.

For this intermediate version, designated as DIRECTSD 3.5, the `poln` class was rewritten to make it compatible with the new OOP model. To recover the functionality of the overloaded functions new classes were defined as subclasses of the standard MATLAB classes. For example, for the `tf` class a subclass called `sdtf` was defined. In this way, the class `sdtf` inherits all the properties and methods of a standard `tf` object replacing the old methods with those that were overloaded in previous versions. Albeit this solution required extensive editing of the codes, it was the most practical way to replace the overloaded functions.

Besides the recovery of the basic functionality some minor extensions were added to the toolbox. The most important is the possibility of using a first order hold with systems affected by a time delay.

Since this version is considered as an interim while the newer version is under development, we have no plans to make it publicly available. However, any interested reader may contact the authors to request a copy.

The following subsections present some examples of usage obtained with DIRECTSD 3.5.

3.1 H_2 Optimization of a system with delay using Zero- and First Order Hold

This case study consists on solving the H_2 optimization problem for a sampled-data system affected by a time delay, using two different hold devices. The objective of the H_2 optimization is to find a controller $C(\zeta)$ such that the minimum value of the mean variance of the output $\varepsilon(t)$ is obtained. This mean variance is defined as:

$$\bar{d}_\varepsilon = \int_0^T d_\varepsilon(t) dt \quad (1)$$

with $d_\varepsilon(t)$ is the instantaneous variance of the signal $\varepsilon(t)$ [4]. We would like to compare the results (the order of the controller and the optimal cost) using each hold device.

The system under study is shown in figure 1. It consists of a continuous plant $P(s)$, a digital controller $C(\zeta)$, a hold device $H(s)$ and a pure delay representing an actuator. The plant is taken as $P(s) = \frac{1}{s(s+1)}$, and a sampling period $T = 1$ s is used. A value of $\tau = 0.1$ s is set for the delay.

The two options considered for the hold device are a zero order hold (ZOH) for which:

$$u(t) = v_k \quad \text{for } kT \leq t < (k+1)T \quad (2)$$

and a first order hold (FOH) for which:

$$u(t) = v_k + (t - kT) \frac{v_k - v_{k-1}}{T} \quad \text{for } kT \leq t < (k+1)T. \quad (3)$$

In order to use the DirecSD toolbox, the system must be converted to the standard form of figure 2. For this case we have:

$$\begin{aligned} K(s) &= P(s) & L(s) &= P(s) e^{-\tau s} \\ M(s) &= -P(s) & N(s) &= -P(s) e^{-\tau s} \end{aligned}$$

The commands needed to run the example follow. First, the system is set-up:

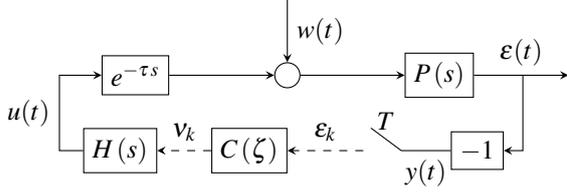


Figure 1: Structure of the system

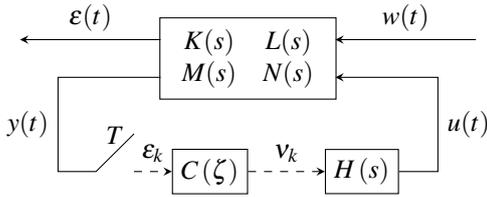


Figure 2: Standard sampled-data system.

```

>>T = 1; %Sampling Period
>>H0 = tf(1,[1 0]); %Zero-order hold
>>H1(:, :, 1,1)=ss(1);%First-order hold
>>H1(:, :, 2,1)=ss(1);%First-order hold
>>P = zpk([], [0 -1], 1); %Plant
>>tau = 0.1;
>>Fdelay = tf(1);
>>Fdelay.iodelay = tau;
>>Pdelay = P*Fdelay;
>>sys = [P Pdelay; -P -Pdelay];
    
```

The last line creates a rational matrix, which represents a system in the standard form. The function `sdh2` is used to synthesize the controllers and to find the optimal cost. The commands issued are:

```

>>[Kzohd, err0d] = sdh2(sys,T, [], H0);
>>[Kfohd, err1d] = sdh2(sys,T, [], H1);
    
```

For the ZOH the toolbox reports the controller:

$$C_0(\zeta) = \frac{-193.39(\zeta - 3.964)}{(\zeta + 171.1)(\zeta + 1.143)} \quad (4)$$

with a minimum cost of $\bar{d}_{\epsilon 0} = 0.6003$. On the other hand, using the FOH, we obtain the controller:

$$C_1(\zeta) = \frac{200(\zeta - 3.819)}{(\zeta + 118.4)(\zeta - 2.705)(\zeta + 0.9415)} \quad (5)$$

with an optimal cost $\bar{d}_{\epsilon 1} = 0.6538$.

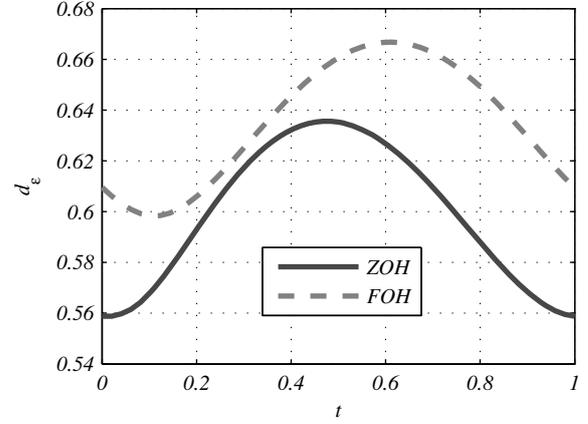


Figure 3: Comparison of the inter-sample variance for the cases with ZOH and FOH.

These results show that for this particular case the introduction of a FOH does not represent any advantage over the use of a ZOH. Besides the higher order of the reported controller, the performance index worsens when the FOH is used. To have a more decisive comparison, figure 3 presents the inter-sample variance (i.e., $\bar{d}_{\epsilon}(t)$) for both cases. To obtain this plot, we used the function `sdh2norm` which finds either the mean variance or the variance at a certain point within the sampling interval. For this particular case we issued the following commands to `DIRECTSD 3.5`:

```

>>t = linspace(0,T,50);
>>errs0d = sdh2norm(sys,Kzohd,t,H0);
>>errs1d = sdh2norm(sys,Kfohd,t,H1);
    
```

in order to obtain the variance at 50 equidistant points within the sampling interval. The results were then plotted to obtain the result observed in figure 3.

3.2 H_2 Optimization of a Generic System with Two Delays

This example considers the H_2 optimization of a closed loop sampled-data system, as shown in figure 4. This is the most general case for a single loop system with a SISO controller.

For optimization purposes, the output of the system is taken as:

$$\mathbf{z}(t) = \begin{bmatrix} z_y(t) \\ z_u(t) \end{bmatrix} \quad (6)$$

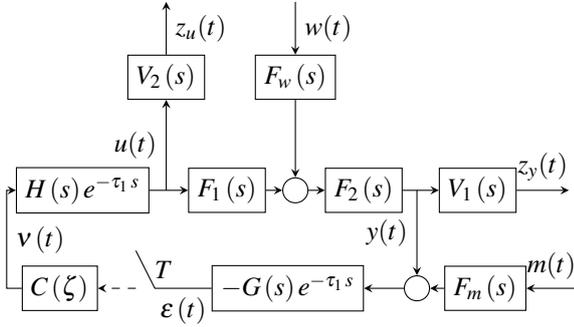


Figure 4: Structure of the system

and the cost function to be minimized is:

$$J = \bar{d}_{zy} + \bar{d}_{zu}. \quad (7)$$

In order to use the DIRECTSD toolbox, the system must be converted to the standard form of figure 2. Given that the input of the system is:

$$\mathbf{x}(t) = \begin{bmatrix} w(t) \\ m(t) \end{bmatrix}, \quad (8)$$

we have:

$$K(s) = \begin{bmatrix} V_1(s)F_2(s)F_w(s) & 0 \\ 0 & 0 \end{bmatrix} \quad (9)$$

$$L(s) = \begin{bmatrix} V_1(s)F_2(s)F_1(s)H(s)e^{-\tau_1 s} \\ V_2(s)H(s)e^{-\tau_1 s} \end{bmatrix} \quad (10)$$

$$M(s) = - \begin{bmatrix} G(s)F_2(s)F_w(s)e^{-\tau_2 s} & G(s)F_m(s)e^{-\tau_2 s} \end{bmatrix} \quad (11)$$

$$N(s) = -G(s)F_2(s)F_1(s)H(s)e^{-(\tau_1+\tau_2)s} \quad (12)$$

Consider a simple case in which the disturbance and measurement noise are both with noise. This means that the forming filters $F_w(s)$ and $F_m(s)$ are both equal to 1. The weights of the output signals are taken as $V_1 = 1$ and $V_2 = 2$. The other components take the following values:

$$F_1(s) = \frac{1}{s+1} \quad G(s) = 1$$

$$F_2(s) = \frac{2}{s+2} \quad H(s) = \frac{0.25}{s+0.25}$$

the two time delays are $\tau_1 = 0.05$ and $\tau_2 = 0.1$, whereas the sampling period is taken as $T = 0.5$.

This example is run by entering the following commands:

```
>>F1 = tf(1,[1 1]);
>>F2 = tf(2,[1 2]);
>>Fm = 1;
>>Fw = 1;
>>tau1 = 0.05;
>>tau2 = 0.1;
>>G = -tf(1,'iodelay',tau2);
>>H = tf(0.25,[1 0.25],'iodelay',tau1);
>>V1 = 1;
>>V2 = 2;
>>T = 0.5;
>>H0 = tf(1,[1 0]);
>>K = [V1*F2*Fw 0; 0 0];
>>L = [V1*F2*F1*H; V2*H];
>>M = [G*F2*Fw G*Fm];
>>N = G*F2*F1*H;
>>sys = [K L;M N];
>>[C, err] = sdh2(sys,1,[],H0)
```

Considering these values, the optimal controller is found as:

$$C_{opt}(\zeta) = \frac{0.88964(\zeta - 1.284)(\zeta - 2.718)}{(\zeta + 3.721)(\zeta - 3.134)(\zeta - 13.81)} \quad (13)$$

with an optimal cost $J_{opt} = 0.99995$.

4 Future Version: DIRECTSD 4.0

A complete refurbishing of the toolbox is currently undergoing. We aim to recover the full capabilities of the last public release while we add some bells and whistles. Some of the improvements being added or planned to be added are:

A new class for rational matrices: Up to this point, different classes have been used to represent polynomial matrices (class `poln`) and rational matrices (linear systems in `zpk`, `tf` or `ss` form). This created the need to some extra code to perform transformations when two objects of different classes were to be operated together. The new class will be a sub class of the standard matlab classes, to make it inherit their properties and methods, and will also have all the particular methods to operate with polynomial matrices, i.e., rational matrices with denominator equal to one.

Packages for different methods: In its version 3.0 DIRECTSD includes options to work with the lifting technique or with polynomial methods. These functions are being reorganized in the form of packages, according to the new OOP model of MATLAB.

Support for MIMO systems: The DIRSDM toolbox [7] was abandoned due to numerical instabilities of the MIMO version of the algorithms for polynomial design of Sampled-Data systems. An effort is being made in to develop numerically reliable algorithms for this case. We are attempting to include said algorithms in the newer version.

Updated examples, demos and help files: The *html* help files will be revised and updated once the final code is in place.

Besides this user-facing improvements, the codes of all the functions are being revised. The coding style is being standardized and the internal documentation is being improved. These steps aim to simplify the maintenance of the toolbox.

For future versions, we are already considering the possibility of developing a graphic user interface, similar to the *sisotool* included in the standard *Control Systems Toolbox*.

5 Concluding Remarks

This paper described the current development status of version 4.0 for the DIRECTSD, a MATLAB toolbox for the analysis and design of sampled-data systems. While an interim version, which recovered the basic functionality of the obsolete code, is in place, the work to obtain a completely revised version is on going.

References

- [1] T. Chen and B. Francis, *Optimal Sampled-Data Control Systems*. New York: Springer-Verlag, 1995.
- [2] H. Fujioka, Y. Yamamoto, and S. Hara, "Sampled-data control toolbox: A software package via object oriented programming," in *Proceedings of the IEEE International Symposium on Computer Aided Control Systems Design*, Hawaii, HI, USA, August 1999, pp. 404–409.
- [3] H. Fujioka, S. Hara, and Y. Yamamoto, "Sampled-data control toolbox: object oriented sof for sampled-data feedback control systems," in *Proceedings of the IEEE Conference on Computer Aided Control Systems Design*, Taipei, Taiwan, September 2004, pp. 19–24.
- [4] E. Rosenwasser and B. Lampe, *Computer Controlled Systems: Analysis and Design using Process Orientated Models*, ser. Communications and Control Engineering. London: Springer-Verlag, 2000.
- [5] —, *Multivariable Computer Controlled Systems: A Transfer Function Approach*, ser. Communications and Control Engineering. London: Springer-Verlag, 2006.
- [6] K. Polyakov, E. Rosenwasser, and B. Lampe, "Directsd - a toolbox for direct design of sampled-data systems," in *Proceedings of the IEEE International Symposium on Computer Aided Control Systems Design*, Hawaii, HI, USA, August 1999, pp. 357–362.
- [7] K. Polyakov, B. Lampe, and E. Rosenwasser, "Directsdm - a toolbox for polynomial design of multivariable sampled-data systems," in *Proceedings of the IEEE Conference on Computer Aided Control Systems Design*, Taipei, Taiwan, September 2004, pp. 95–100.
- [8] K. Polyakov, E. Rosenwasser, and B. Lampe, "Directsd 3.0 toolbox for matlab: further progress in polynomial design of sampled-data systems," in *Proceedings of the IEEE Conference on Computer Aided Control Systems Design*, Munich, Germany, October 2006, pp. 1946–1951.
- [9] K. Polyakov. (2015, May) Directsd: Package for the analysis and synthesis of digital control systems. Online. [Online]. Available: <http://kpolyakov.spb.ru/science/dsd.htm>

Hybrid State Estimation of nonlinear index-1 DAE-Systems

Dirk Weidemann¹, Ilja Alkov¹

¹Institut of System Dynamics and Mechatronics
University of Applied Sciences Bielefeld
dirk.weidemann@fh-bielefeld.de

Modern technical systems often contain components capable for extensive autonomous actions. Thus, an integrated algorithmic system supervision is essential enabling an adequate reaction for compensation of unpredictable substantial variations. This is typically addressed by fault detection, isolation, and identification techniques. Therefore, an overview is given about modeling of systems subject to faults, continuous state estimation utilizing an unscented Kalman filter, and hybrid state estimation. These methods are particularly generalized for application to nonlinear differential algebraic equations with index 1, i.e. index-1 DAE-systems. Furthermore, the proposed fault identification approach is applied to a hydraulic system and the related results are discussed in detail.

1 Introduction

As modern technical systems become more complex, often the likelihood increases that faults occur. An appropriate way to compensate for effects of faults is to use redundant hardware. For cost reasons, however, the deployment of redundant systems is uncommon if the faults do not lead to life-threatening situations. Rather, the aim is to compensate for the effect of faults by diagnosing them and using an appropriate fault-tolerant control strategy. Thus, model-based fault diagnosis and (active) fault-tolerant control play an increasingly important role for complex systems and are an active research area. However, active fault-tolerant control approaches usually require not only to locate the fault but also to estimate its type and magnitude which is known as fault identification.

A significant amount of model-based approaches are proposed for fault detection and fault isolation (cf. [2],[7]). In contrast, there are fewer approaches for fault identification. One of the first fault identification approaches was proposed in [11] for linear systems and is based on the assumption that faults lead to a change of the system parameters. Therefore, the faults can be identified using online parameter estimation approaches. In contrast, in [19] a sliding mode observer is proposed to identify actuator and sensor

faults. Note that this approach is also only applicable for linear systems. Most systems, however, are characterized by significant nonlinear behavior. An extension of the approach proposed in [19] to a specific class of nonlinear systems is given in [5]. Though, the latter approach is only applicable to identify actuator faults.

An alternative to identify sensor, actuator, and system faults is a combined state and fault estimation in which the faults are modeled by augmented state variables. Combined state and parameter estimation with a two-state adaptive Kalman-Filter is, for example, proposed in [22] as basis for a fault tolerant control scheme. If the system is only subject to one fault it is sufficient to use only one model. On the other hand, complex systems typically are subject to more than one fault such that in addition to a model, describing the behavior of the fault-free system, one model for each fault has to be used. In order to find which of these models fits the actual system behavior best a mode estimation scheme is appropriate. As indicated in [10], mode estimation can be done in the sense of hybrid estimation.

Common hybrid estimation schemes assume that the model is given in state space form. In the context of complex systems, object oriented modeling has gained increasing attention within science and industry, since it significantly reduces the effort needed to

model technical systems with high complexity. This advantage is obtained by hierarchical modeling, inheritance and the ability to interconnect component models. As a consequence of connecting component models, the resulting mathematical model describing the behavior of the complex systems consists not only of ordinary differential equations but also of algebraic equations. Such sets of equations are called differential algebraic equations (DAEs), also known as descriptor, implicit or singular systems.

In this paper we propose a hybrid estimation scheme for nonlinear index-1 DAE-systems for fault identification purposes. Thus, the estimates consist of discretely and continuously valued states. A hybrid modeling approach is given in which the discretely valued state represents the mode of the system such that the discrete dynamics include phenomena such as jumps in the state or switching of the vector field. The continuously valued states represent the corresponding continuous dynamics of each mode described by a nonlinear DAE-system.

2 Differential-Algebraic-Systems

Even though modeling of complex systems gets easier if an object oriented modeling approach is used, this modeling approach typically leads to a set of differential algebraic equations. The class of DAE-systems contains systems given in state space form as a special case, but in general the properties of DAE-systems are more complex. Therefore, analysis of DAE-systems and also synthesis of controllers, observers, and filters for DAE-systems are more challenging than for systems given in state space form. This section concerns the difference between DAE- and ODE-systems, giving a formal definition of a DAE-system.

The most general form of a DAE-system is a fully implicit system

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

where

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

and $f: \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \rightarrow \mathbb{R}^{n_x}$ is a sufficiently differentiable vector valued function, implicitly dependent on the generalized state $x \in \mathbb{R}^{n_x}$ of the DAE, the corresponding time derivative of the generalized state

$\dot{x} \in \mathbb{R}^{n_x}$, a time dependent input variable $u \in \mathbb{R}^{n_u}$, and the time $t \in \mathbb{R}$. The dimensions of the generalized state and the input variable are assigned with n_x and n_u respectively.

Since (2) holds, according to the implicit function theorem, the fully implicit system (1), which contains differential and algebraic equations, can not be solved for \dot{x} algebraically. Furthermore, the condition (2) may be satisfied only locally but not globally and the structural properties of the system may change significantly. For the sake of simplicity, only DAE-systems with invariable structural properties are considered in the following.

The core of the hybrid estimation presented in Section 5 consists of Kalman filters for nonlinear DAE-systems. Most Kalman filter approaches for nonlinear DAE-systems are applicable only for index-1 systems, whereas few approaches for higher index problems exist (cf. [1]). Therefore, the concept of index is explained briefly. Several concepts for the index of DAE-systems are established in the literature. A detailed overview of the concepts of differentiation, perturbation, geometric, strangeness, structural, and tractability indices is given in [16]. However, the differentiation index is the most widely used concept.

The differentiation index (cf. [4]) of a DAE system (1) is the smallest number of differentiations $\mu \in \mathbb{N}$ required to solve

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

algebraically for \dot{x}

$$\dot{x} = \phi(x, u, \dot{u}, \dots, u^{(\mu)}, t) \quad (4)$$

as a function of the generalized state, the input variable, time derivatives of the input variable, and the time. The equation ϕ is referred to as the underlying ODE corresponding to the original DAE and may be an implicit equation algebraically solvable for \dot{x} . The differentiation index can be interpreted as a criterion describing the structural properties of DAE systems. A frequently indicated disadvantage of the differentiation index concept is the requirement of unique solvability making it unsuitable for over- and underdetermined systems. In this contribution it is assumed that

the system is uniquely solvable which holds for most engineering problems.

Solving nonlinear differential-algebraic systems is still a research subject, however, several approaches are established in literature. A number of direct integration methods are applicable to specific classes of DAE systems (cf. [3], [6]). These methods often require a low differentiation index and/or a specific structure of the DAE system. The probably most widely used solution method is a combination of an index reduction technique (e.g. [18], [15]) and a direct numerical integration of an index reduced DAE formulation. This approach is implemented in popular commercial (e.g. Dymola or MapleSim™) and open source (e.g. OpenModelica) simulation environments. Furthermore, solver implementations based on an equivalent strangeness-free formulation of the DAE (cf. [13]), on Taylor series expansion (cf. [17]), and on nonlinear programming (cf. [8]) are relevant for scientific as well as industrial applications. The application of hybrid state estimation as presented in Section 5 requires that the system is modeled by an index-1 DAE-System

$$\dot{x}_d = \phi_d(x_d, x_a, u_\mu, t) \quad (5)$$

$$g_a(x_d, x_a, u_\mu, t) = 0 \wedge \det\left(\frac{\partial g_a}{\partial x_a}\right) \neq 0, \quad (6)$$

where $x_d \in \mathbb{R}^{n_{x,d}}$ are the differential and $x_a \in \mathbb{R}^{n_{x,a}}$ the algebraic states, with corresponding dimensions $n_{x,d} \in \mathbb{N}$ and $n_{x,a} \in \mathbb{N}$. In this formulation the differential state can be initialized arbitrarily, whereas the algebraic state is calculated dependently. Thus, the corresponding state transition function can be given directly as

$$x_{d,k} = x_{d,k-1} + \int_{t_{k-1}}^{t_k} \phi_d(x_d, x_a, u_\mu, t) \cdot dt \quad (7)$$

$$\begin{aligned} &= \xi_{d,k}(x_{d,k-1}, x_{a,k-1}, u_\mu(\cdot), t_{k-1}, t_k), \\ x_{a,k} &= \xi_{a,k}(x_{d,k}, u_{\mu,k}, t_k), \end{aligned} \quad (8)$$

where $t_{\{k-1,k\}} \in \mathbb{R}$ with $k \in \mathbb{N}$ are time instants, $x_{d,\{k-1,k\}} \in \mathbb{R}^{n_{x,d}}$ are differential states, $x_{a,\{k-1,k\}} \in \mathbb{R}^{n_{x,a}}$ are algebraic states, $\xi_{d,\{k-1,k\}}$ differential state transition functions, $\xi_{a,\{k-1,k\}}$ algebraic state transition functions, at these time instants respectively.

3 Modelling of Systems subject to Faults

Concerning modeling of systems subject to faults it is necessary to distinguish between the locations, where faults can occur, and between different fault characteristics. Since faults can affect sensors, actuators, and the system these faults commonly are referred to as sensor-, actuator-, and system-faults. Even though the fault magnitude can possess an arbitrary characteristic with respect to time, the fault magnitude is often modeled either by an abruptly changing signal (abrupt fault) or by a constantly increasing signal (incipient fault).

In order to identify the fault magnitude of an abrupt fault or the constant slope of an incipient fault it is appropriate to augment the differential states of the fault-free system which is given by a nonlinear index-1 DAE system

$$\dot{x}_d^{(f)} = \phi_d^{(f)}(x_d^{(f)}, x_a, u_\mu, t), \quad (9)$$

$$g_a^{(f)}(x_d^{(f)}, x_a, u_\mu, t) = 0 \wedge \det\left(\frac{\partial g_a^{(f)}}{\partial x_a}\right) \neq 0, \quad (10)$$

$$y = h^{(f)}(x_d^{(f)}, x_a, u_\mu, t). \quad (11)$$

where $x_d^{(f)}$ denotes the differential states of the fault-free system and $\phi_d^{(f)}$ the corresponding differential equations, $g_a^{(f)}$ the algebraic equations, and $h^{(f)}$ the output mapping. Let $x_d^{(ab)} = [(x_d^{(f)})^T, x_{d,aug}^{(ab)}]^T$ as well as $x_d^{(in)} = [(x_d^{(f)})^T, x_{d,aug}^{(in)}]^T$ denote the augmented state vector for the abrupt and incipient fault, respectively. Then, modelling an abrupt fault by $\dot{x}_{d,aug}^{(ab)} = 0$ and the dynamics of an incipient fault by $\dot{x}_{d,aug}^{(in)} = s$ with s characterizing the slope yields the augmented models

$$\dot{x}_d^{(ab)} = \phi_d^{(ab)}(x_d^{(ab)}, x_a, u_\mu, t) = \begin{bmatrix} \phi_d^{(f)}(x_d^{(ab)}, x_a, u_\mu, t) \\ 0 \end{bmatrix}, \quad (12)$$

$$g_a^{(ab)}(x_d^{(ab)}, x_a, u_\mu, t) = 0 \wedge \det\left(\frac{\partial g_a^{(ab)}}{\partial x_a}\right) \neq 0, \quad (13)$$

$$y = h^{(ab)}(x_d^{(ab)}, x_a, u_\mu, t), \quad (14)$$

and

$$\dot{x}_d^{(in)} = \phi_d^{(in)}(x_d^{(in)}, x_a, u_\mu, t) = \begin{bmatrix} \phi_d^{(f)}(x_d^{(in)}, x_a, u_\mu, t) \\ 0 \end{bmatrix}, \quad (15)$$

$$g_a^{(in)}(x_d^{(in)}, x_a, u_\mu, t) = 0 \wedge \det\left(\frac{\partial g_a^{(in)}}{\partial x_a}\right) \neq 0, \quad (16)$$

$$y = h^{(in)}(x_d^{(in)}, x_a, u_\mu, t). \quad (17)$$

for the fault cases. As indicated by the superscripts $(\cdot)^{\{f,ab,in\}}$ each of these models characterize the behavior of the system in the corresponding mode. Note that the approach of state augmentation is not restricted to abrupt or incipient faults but can also be applied to model more complex fault characteristics. The abrupt and incipient faults are just selected to illustrate the modeling of faults. Moreover, under the assumption that an appropriate estimation approach is used it is possible to distinguish between a high number of different faults if each fault model describes only one or a few faults. In contrast, if one fault model describes all faults, i.e. the fault-free system is augmented by all faults, the augmented system model might be unobservable. In general, the more faults are described by one model the more likely it is that observability is lost.

Depending on whether a fault occurs or not, the system operates in different modes. Let an operation mode $m^{(\cdot)}$ be an element of the set \mathcal{M} which contains all fault modes as well as the fault-free mode. As the transitions from one mode to another are assumed to be instantaneous, the transitions are modeled by an appropriate finite state automation, while the continuous-time dynamics within each mode is given by the corresponding DAE system. Thus, a system subject to faults can be modeled as a hybrid system. The state of the hybrid system is described by the triple $x_h = (z, x_d, x_a)$ with $z \in \mathcal{M}$, $x_d \in \mathbb{R}^{n_{x,d}}$, $x_a \in \mathbb{R}^{n_{x,a}}$ and corresponding dimensions $n_{x,d}, n_{x,a} \in \mathbb{N}$. Assume, for the sake of simplicity, that the system is subject to one abrupt and one incipient fault, then the related hybrid automaton is depicted in Fig. 1 with $\mathcal{M} = \{m^{(f)}, m^{(ab)}, m^{(in)}\}$. Here $m^{(f)}$ stands for the behavioral mode of the fault-free system, whereas $m^{(ab)}$ and $m^{(in)}$ indicate the modes if the system is subject to an abrupt and an incipient fault, respectively. The continuous dynamics in each mode (or state) is governed by a nonlinear index-1 DAE system.

4 Unscented Kalman Filter for Index-1 DAE Systems

The unscented Kalman filter (UKF) is an extension of the original Kalman filter algorithm for the application to nonlinear systems, where the eponymous technique is the unscented transformation (UT). The UT introduced in [12] is utilized for the calculation of the expectation and covariance of stochastic variables determined by the nonlinear system. Therefore, deterministic sets of model predicted state and output variables are evaluated. The number of elements in the evaluated sets is relatively small in comparison to a particle filter with a comparable accuracy. Thus, the key advantage of the UKF is the calculation method of the involved stochastic moments avoiding a determination of any Jacobian of the model equations. As a consequence, the estimation for highly nonlinear and stiff systems is of higher accuracy and reliability than by Kalman filter variants involving partial derivatives. The computational complexity order of the UKF is $\mathcal{O}(L^3)$, where L is the dimension of the estimated variable. However, the extended Kalman filter (EKF) has the same complexity order (c.f. [20]).

The application and parametrization of model based estimation methods to DAE systems is substantially more complex than the corresponding techniques for ODE systems. In the following a variant of the UKF for nonlinear DAE-systems with index-1 proposed in [14] is sketched.

The DAE formulation given by

$$x_{d,k} = x_{d,k-1} + \int_{t_{k-1}}^{t_k} \phi_d(x_d, x_a, u_\mu, t) \cdot dt + v_{x,d,k-1} = \quad (18)$$

$$= \xi_{d,k}(x_{d,k-1}, x_{a,k-1}, u_\mu(\cdot), t_{k-1}, t_k) + v_{x,d,k-1},$$

$$g_a(x_d, x_a, u_\mu, t) = 0 \wedge \det\left(\frac{\partial g_a}{\partial x_a}\right) \neq 0, \quad (19)$$

$$x_{a,k} = \xi_{a,k}(x_{d,k}, u_{\mu,k}, t_k), \quad (20)$$

$$y_k = h_k(x_{d,k}, x_{a,k}, u_{\mu,k}, t_k) + v_{y,k} \quad (21)$$

where $t_{\{k-1,k\}} \in \mathbb{R}$ with $k \in \mathbb{N}$ are time instants, $x_{d,\{k-1,k\}} \in \mathbb{R}^{n_{x,d}}$ are differential states, $x_{a,\{k-1,k\}} \in \mathbb{R}^{n_{x,a}}$ algebraic states, $u_{\mu,\{k-1,k\}} \in \mathbb{R}^{n_u \times \mu}$ input variables and corresponding time derivatives, $y_{\{k-1,k\}} \in \mathbb{R}^{n_y}$ output variables, $\xi_{d,\{k-1,k\}}$ differential state transition functions, $\xi_{a,\{k-1,k\}}$ algebraic state transition

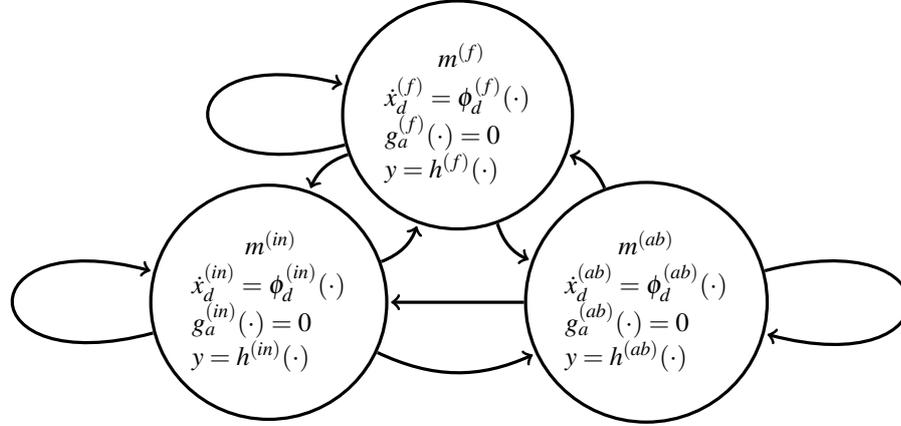


Fig. 1: Hybrid automaton for a system subject to two faults

functions, $h_{\{k-1,k\}}$ output functions, $v_{x,d,\{k-1,k\}} \in \mathbb{R}^{n_{x,d}}$ stochastic differential state noise signals, and $v_{y,\{k-1,k\}} \in \mathbb{R}^{n_y}$ stochastic output noise signals at these time instants. Furthermore, ϕ_d is the ODE part of the index-1 DAE system and g_a is the restrictive manifold with the local solution ξ_a for algebraic states. The stochastic processes $v_{x,d,\{k-1,k\}} \sim \mathcal{N}(0, C_{v_{x,d},v_{x,d},\{k-1,k\}})$ and $v_{y,\{k-1,k\}} \sim \mathcal{N}(0, C_{v_y,v_y,\{k-1,k\}})$ are assumed to be normal distributed white noise processes independent of each other and other system quantities with zero expectation and known covariances at corresponding time instants. Completing, $n_{x,d}, n_{x,a}, n_u, n_y \in \mathbb{N}$ are corresponding dimensions and $\mu \in \mathbb{N}$ the differential index of the original higher index DAE reduced to the given index one formulation. Consequently, $x_{d,\{k-1,k\}}$ and $y_{\{k-1,k\}}$ are stochastic quantities with $\mathbb{E}(x_{d,\{k-1,k\}})$, $\mathbb{E}(y_{\{k-1,k\}})$ denoting the expectations and $\text{Cov}(x_{d,\{k-1,k\}}, x_{d,\{k-1,k\}}) = C_{x_d,x_d,\{k-1,k\}}$, $\text{Cov}(y_{\{k-1,k\}}, y_{\{k-1,k\}}) = C_{y,y,\{k-1,k\}}$, $\text{Cov}(x_{d,\{k-1,k\}}, y_{\{k-1,k\}}) = C_{x_d,y,\{k-1,k\}}$ covariances.

As typical for Kalman filters dual time indices $k-1|k-1$, $k|k-1$ and $k|k$ are introduced, denoting quantities which are determined considering model information up to the time instant indicated by left and actual system information up to the time instant indicated by right index component. Furthermore, a function $\text{srm}(\cdot)$ is introduced, denoting the matrix square root operation applicable to square positive definite

matrices as

$$\text{srm}(A) \cdot \text{srm}(A)^T = A. \quad (22)$$

The UKF for DAE systems is represented using matrix operations instead of conventional set operations. Therefore, application of a function defined for a vector argument to a matrix argument with equal number of rows means that this function is applied to each column of the matrix. The result is a matrix with the 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. The estimation algorithm is structured as usual into the steps *Initialization*, *Prediction*, and *Correction* as given below. Thus, the initialization is executed at the beginning of the estimation, whereas prediction and correction are repeated sequentially for each taken measurement.

The weights for the scaled UT are defined by

$$w_m = \frac{\left[2 \cdot \alpha^2 \cdot (n_{x,d} + \gamma) - 2 \cdot n_{x,d}, 1_{1 \times 2 \cdot n_{x,d}} \right]^T}{2 \cdot \alpha^2 \cdot (n_{x,d} + \gamma)} \quad (23)$$

and

$$W_c = w_m + \left[(1 - \alpha^2 + \beta), 0_{1 \times 2 \cdot n_{x,d}} \right]^T \quad (24)$$

$$\Psi = I_{2 \cdot n_{x,d} + 1} - w_m \cdot 1_{1 \times 2 \cdot n_{x,d} + 1} \quad (25)$$

$$w_c = \Psi \cdot \text{diag}(W_c) \cdot \Psi^T, \quad (26)$$

where $0 < \alpha \leq 1$, $0 \leq \beta$, $0 \leq \gamma$ are the real scalar UT set-up parameters, $w_m \in \mathbb{R}^{2 \cdot n_{x,d} + 1}$ is the vector

valued weight for calculation of the expectation and $w_c \in \mathbb{R}^{2 \cdot n_{x,d} + 1 \times 2 \cdot n_{x,d} + 1}$ is the matrix valued weight for calculation of the covariance. Further, $I(\cdot)$, $0(\cdot)$, and $1(\cdot)$ denote identity, zeros and ones matrices of dimension specified in the index respectively. Remaining quantities are locally defined auxiliaries.

Initialization:

$$k - 1 = k_0$$

$$\mathbb{E}(x_{d,k-1|k-1}) = x_{d,0}$$

$$C_{x_d, x_d, k-1|k-1} = C_{x_d, x_d, 0}$$

At the initial instant t_{k_0} , an initial differential state estimation $x_{d,0}$ and a corresponding covariance $C_{x_d, x_d, 0}$ are assumed.

Prediction:

$$d_{x_d, k-1|k-1} = [0_{n_{x,d} \times 1}, -\text{srn}(C_{x_d, x_d, k-1|k-1}), \\ \text{srn}(C_{x_d, x_d, k-1|k-1})]$$

$$s_{x_d, k-1|k-1} = \mathbb{E}(x_{d, k-1|k-1}) 1_{1 \times 2n_{x,d} + 1} \\ + \alpha \sqrt{n_{x,d} + \gamma} d_{x_d, k-1|k-1}$$

$$s_{x_a, k-1|k-1} = \xi_{a, k-1}(s_{x_d, k-1|k-1}, u_{\mu, k-1}, t_{k-1})$$

$$s_{x_d, k|k-1}^* = \xi_{d, k}(s_{x_d, k-1|k-1}, s_{x_a, k-1|k-1}, u_{\mu}(\cdot), t_{k-1}, t_k) \\ + \mathbb{E}(v_{x_d, k-1}) 1_{1 \times 2n_{x,d} + 1}$$

$$\mathbb{E}(x_{d, k|k-1}) = s_{x_d, k|k-1}^* w_m$$

$$C_{x_d, x_d, k|k-1} = s_{x_d, k|k-1}^* w_c s_{x_d, k|k-1}^{*T} + C_{v_{x_d}, v_{x_d}, k}$$

$$d_{x_a, d, k|k-1} = [0_{n_{x,d} \times 1}, -\text{srn}(C_{x_d, x_d, k|k-1}), \\ \text{srn}(C_{x_d, x_d, k|k-1})]$$

$$s_{x_d, k|k-1} = \mathbb{E}(x_{d, k|k-1}) 1_{1 \times 2n_{x,d} + 1} \\ + \alpha \sqrt{n_{x,d} + \gamma} d_{x_d, k|k-1}$$

$$s_{x_a, k|k-1} = \xi_{a, k}(s_{x_d, k|k-1}, u_{\mu, k}, t_k)$$

$$s_{y, k|k-1} = h_k(s_{x_d, k|k-1}, s_{x_a, k|k-1}, u_{\mu, k}, t_k) \\ + \mathbb{E}(v_{y, k}) 1_{1 \times 2n_{x,d} + 1}$$

$$\mathbb{E}(y_{k|k-1}) = s_{y, k|k-1} w_m$$

$$C_{y, y, k|k-1} = s_{y, k|k-1} w_c s_{y, k|k-1}^T + C_{v_y, v_y, k}$$

$$C_{x_d, y, k|k-1} = s_{x_d, k|k-1} w_c s_{y, k|k-1}^T$$

In this calculation step, the expectations $\mathbb{E}(x_{d, k|k-1})$, $\mathbb{E}(y_{k|k-1})$, as well as the covariances $C_{x_d, x_d, k|k-1}$,

$C_{y, y, k|k-1}$, $C_{x_d, y, k|k-1}$ are determined using the model and the UT. The columns of the matrices composed of state and output values $s_{x_d, k|k-1}$, $s_{x_a, k|k-1}$, $s_{y, k|k-1}$ are commonly referred to as sigma points.

Correction:

$$K_{x_d, d, k} = C_{x_d, y, k|k-1} C_{y, y, k|k-1}^{-1}$$

$$\mathbb{E}(x_{d, k|k}) = \mathbb{E}(x_{d, k|k-1}) + K_{x_d, d, k} (y_{m, k} - \mathbb{E}(y_{k|k-1}))$$

$$C_{x_d, x_d, k|k} = C_{x_d, x_d, k|k-1} - K_{x_d, d, k} C_{y, y, k|k-1} K_{x_d, d, k}^T$$

In this final calculation step the Kalman gain $K_{x_d, d, k}$ is determined. Thus, the expectation $\mathbb{E}(x_{d, k|k})$ and the covariance $C_{x_d, x_d, k|k}$ of the estimated differential state can be obtained accordingly, considering the actual system measurement $y_{m, k}$. Note, that the algebraic states in each Kalman filter cycle are always calculated such that a consistent initialization of the differential state transition function is ensured. Thus, the algebraic constraint imposed on the generalized DAE state is satisfied by the estimation algorithm.

5 Hybrid Estimation for DAE-Systems

As described in Section 3 it is appropriate to model systems subject to faults as hybrid systems. Thus, in order to solve the fault identification task, both the discrete valued state as well as the continuous states have to be determined, which can be done using a hybrid state estimation approach. Discussing the drawbacks of both the simplest and the optimal hybrid estimation approach indicate why neither of them are applicable for fault identification purposes. A suitable alternative is a suboptimal hybrid estimation scheme such as the interacting multiple model (IMM) approach. This approach originated in the context of target tracking but has also shown to be effective for fault diagnosis if the continuous dynamics of each mode in the related hybrid automaton is modeled by a set of ordinary differential equations, i.e. by an index-0 DAE-system. Since the IMM approach assumes that the discrete dynamics is modeled by a Markov chain of order one, this will be introduced before proceeding with the aforementioned hybrid estimation schemes for DAE-systems.

5.1 Discrete-Time Markov Chains

A discrete-time Markov chain is a model for representing stochastic processes whereby state changes occur at discrete time instants $k \in \mathbb{N}$ and the state space is either a finite set or countable set. The characteristic property of a Markov chain is that the value of the discrete state z_k is only influenced by the value of the state z_{k-1} but not by the values the states z_i with $i \in 0, 1, \dots, k-2$. In mathematical terms, the Markov property is given by

$$P(z_k = m^{(i)} | z_{k-1} = m^{(j)}, \dots, z_0 = m^{(\ell)}) = P(z_k = m^{(i)} | z_{k-1} = m^{(j)}) \quad (27)$$

with $m^{(i)} \in \mathcal{M}$. The probability of z_k being in the discrete state i or, if it comes to fault diagnosis, in mode $m^{(i)}$ given that the z_{k-1} is in mode $m^{(j)}$ is called the state transition probability and denoted by

$$P_{k,i,j} = P(z_k = m^{(i)} | z_{k-1} = m^{(j)}). \quad (28)$$

Assume that the number of discrete states $n_z = |\mathcal{M}|$ is finite, then the state transition probabilities can be arranged in a transition probability matrix $T_k \in \mathbb{R}^{n_z \times n_z}$, where each element $t_{k,i,j} = P_{k,i,j}$ assigns the probability for a state transition from $z_{k-1} = m^{(j)}$ to $z_k = m^{(i)}$ at time instant k . Thus, the probability transition matrix possess the properties

$$t_{k,i,j} \geq 0 \wedge \sum_{i=1}^{n_z} t_{k,i,j} = 1 \quad (29)$$

for all $i, j \in \{1, \dots, n_z\}$. In the following it is assumed that the transition probabilities are independent of time k , such that the Markov chain has stationary transition probabilities $P_{k,i,j} = P_{i,j}$.

5.2 Hybrid Estimation Schemes

Non-switching hybrid estimation. The simplest hybrid estimation scheme is the non-switching approach, which is also known as fixed model approach or static case. In this approach it is assumed that the system stays in one mode for the entire time, i.e. the system does not switch to another mode during the estimation process. Furthermore, a bank of $q = |\mathcal{M}|$ Kalman filters is implemented to perform the estimation of the mode conditioned mean

$$\hat{x}_{k|k}^{(i)} = E(x_k | Y_k, m^{(i)}) \quad (30)$$

and covariance

$$P_{k|k}^{(i)} = E((x_k - \hat{x}_{k|k}^{(i)})(x_k - \hat{x}_{k|k}^{(i)})^T | Y_k, m^{(i)}) \quad (31)$$

of each mode $m^{(i)} \in \mathcal{M}$, where $x_k = [x_{k,d}^T, x_{k,a}^T]^T$ denotes the concatenation of differential and algebraic states and $Y_k = \{y_1, y_2, \dots, y_k\}$ the sequence of measurements up to time instant k .

In order to estimate the discrete valued variable, i.e. the mode the system is operating in, the most likely mode has to be determined. Applying Bayes' theorem the probability of each mode at time instant k can be calculated recursively by

$$\begin{aligned} \mu_k^{(i)} &= P(m^{(i)} | Y_k) = P(m^{(i)} | y_k, Y_{k-1}) \quad (32) \\ &= \frac{p(y_k | Y_{k-1}, m^{(i)}) P(m^{(i)} | Y_{k-1})}{p(y_k | Y_{k-1})} \\ &= \frac{p(y_k | Y_{k-1}, m^{(i)}) \mu_{k-1}^{(i)}}{\sum_{j=1}^q p(y_k | Y_{k-1}, m^{(j)}) \mu_{k-1}^{(j)}}. \end{aligned}$$

The conditional probability density function $p(y_k | Y_{k-1}, m^{(i)})$ in (32) can, according to [9], be replaced by the likelihood function

$$\Lambda_k^{(i)} = \frac{e^{-\frac{1}{2}(r_k^{(i)})^T S_k^{-1} r_k^{(i)}}}{\sqrt{|2\pi S_k^{(i)}|}}, \quad (33)$$

where $r_k^{(i)} = y_k - \hat{y}_{k|k-1}^{(i)}$ is the residual of the i -th Kalman filter and $S_k^{(i)}$ is the corresponding residual covariance matrix. Thus, fault identification can be done by exploiting the current hybrid estimate $\hat{x}_{h,k} = (\hat{z}_k, \hat{x}_{k|k}^{(i)})$, where $\hat{z}_k = m^{(i)}$ is the most likely mode with $\mu_k^{(i)} \geq \mu_k^{(j)}$ for all $j \in \{1, 2, \dots, q\}$ and $\hat{x}_{k|k}^{(i)}$ is according to (30) the mode conditioned mean.

Moreover, the mode probability $\mu_k^{(i)}$ is also used to determine the combined mean

$$\begin{aligned} \hat{x}_k &= E(x_k | Y_k) \quad (34) \\ &= \sum_{i=1}^q P(m^{(i)} | Y_k) E(x_k | Y_k, m^{(i)}) = \sum_{i=1}^q \mu_k^{(i)} \hat{x}_{k|k}^{(i)} \end{aligned}$$

and covariance matrix

$$\begin{aligned} P_k &= E((x_k - \hat{x}_k)(x_k - \hat{x}_k)^T | Y_k) \quad (35) \\ &= \sum_{i=1}^q P(m^{(i)} | Y_k) E((x_k - \hat{x}_{k|k}^{(i)})(x_k - \hat{x}_{k|k}^{(i)})^T | Y_k, m^{(i)}) + \end{aligned}$$

$$\begin{aligned} & \sum_{i=1}^q P(m^{(i)}|Y_k)(\hat{x}_{k|k}^{(i)} - \hat{x}_k)(\hat{x}_{k|k}^{(i)} - \hat{x}_k)^T \\ &= \sum_{i=1}^q \mu_k^{(i)} P_{k|k}^{(i)} + \sum_{i=1}^q \mu_k^{(i)} (\hat{x}_{k|k}^{(i)} - \hat{x}_k)(\hat{x}_{k|k}^{(i)} - \hat{x}_k)^T \end{aligned}$$

of the continuous valued state.

Optimal hybrid estimation. Concerning the assumption of the non-switching approach that the system has to stay in one mode, it is obvious that this approach is only appropriate for faults, which occurred before the estimation process has been started. In order to cope with faults, which occur during the estimation process, the optimal hybrid estimation approach can be used. The switching between different modes, or rather between different states of the hybrid automation, is modeled by a Markov chain with known state transition probabilities

$$P_{ij} = P(m_k^{(i)}|m_{k-1}^{(j)}), \quad (36)$$

where $m_k^{(i)}, m_{k-1}^{(j)} \in \mathcal{M}$ denote that the system is in mode $m^{(i)}$ at time k and in mode $m^{(j)}$ at time $k-1$, respectively.

Due to the switching it is necessary to track all r^k possible mode sequences

$$M_k^{(i)} = \{m_0^{(i_0)}, m_1^{(i_1)}, \dots, m_k^{(i_k)}\} \quad (37)$$

with $i \in \{1, 2, \dots, r^k\}$, $m_j^{(i_j)} \in \mathcal{M}$ being the mode of the i -th mode sequence at time instant $j \in \{0, 1, \dots, k\}$. In contrast to the non-switching case, the optimal hybrid estimation does not require one Kalman filter for each mode $m^{(i)} \in \mathcal{M}$ but a filter bank with one filter for each possible mode sequence $M_k^{(i)} \in \{M_k^{(1)}, M_k^{(2)}, \dots, M_k^{(r^k)}\}$. Note in this context, that the number of possible mode sequences increases exponentially over time. These Kalman filters yield

$$\hat{x}_{k|k}^{(i)} = E(x_k|Y_k, M_k^{(i)}) \quad (38)$$

and

$$P_{k|k}^{(i)} = E((x_k - \hat{x}_k^{(i)})(x_k - \hat{x}_k^{(i)})^T | Y_k, M_k^{(i)}) \quad (39)$$

for each mode sequence.

The estimation of the discrete valued state is done by first calculating the probabilities of the modes sequences as follows

$$v_k^{(i)} = P(M_k^{(i)}|Y_k) = P(M_k^{(i)}|y_k, Y_{k-1}) \quad (40)$$

$$\begin{aligned} &= \frac{1}{c} p(y_k|Y_{k-1}, M_k^{(i)}) P(M_k^{(i)}|Y_{k-1}) \\ &= \frac{1}{c} p(y_k|Y_{k-1}, M_k^{(i)}) P(m_k^{(i_k)}, M_{k-1}^{(\ell)}|Y_{k-1}) \\ &= \frac{1}{c} p(y_k|Y_{k-1}, M_k^{(i)}) P(m_k^{(i_k)}|Y_{k-1}, M_{k-1}^{(\ell)}) \\ & \quad P(M_{k-1}^{(\ell)}|Y_{k-1}) \end{aligned}$$

with the normalisation constant $c = \sum_{j=1}^{r^k} p(y_k|Y_{k-1}, M_k^{(j)}) P(M_k^{(j)}|Y_k)$. Since the discrete dynamics are modeled by a Markov chain the probability of the current mode depends only on the previous mode (40) simplifies to

$$\begin{aligned} v_k^{(i)} &= \frac{1}{c} p(y_k|Y_{k-1}, M_k^{(i)}) P(m_k^{(i_k)}|m_{k-1}^{(\ell_{k-1})}) P(M_{k-1}^{(\ell)}|Y_{k-1}) \\ &= \frac{1}{c} p(y_k|Y_{k-1}, M_k^{(i)}) P_{i\ell} v_{k-1}^{(\ell)}. \end{aligned} \quad (41)$$

Note, that even though the Markov property has been exploited the probability density function is still conditioned on the entire mode sequence. The discrete valued state at time k is then given by the current mode $m_k^{(i_k)}$ of the most likely mode sequence $v_k^{(i)} \geq v_k^{(j)}$ with $j \in \{1, 2, \dots, r^k\}$, such that the hybrid state is given by $\hat{x}_{h,k} = (\hat{z}_k = m_k^{(i_k)}, \hat{x}_{k|k}^{(i)})$.

Finally, the combined mean and covariance matrix are given by

$$\begin{aligned} \hat{x}_k &= E(x_k|Y_k) \\ &= \sum_{i=1}^{r^k} P(M_k^{(i)}|Y_k) E(x_k|Y_k, M_k^{(i)}) = \sum_{i=1}^{r^k} v_k^{(i)} \hat{x}_{k|k}^{(i)} \end{aligned} \quad (42)$$

and

$$\begin{aligned} P_k &= E((x_k - \hat{x}_k)(x_k - \hat{x}_k)^T | Y_k) \\ &= \sum_{i=1}^q v_k^{(i)} P_{k|k}^{(i)} + \sum_{i=1}^q v_k^{(i)} (\hat{x}_{k|k}^{(i)} - \hat{x}_k)(\hat{x}_{k|k}^{(i)} - \hat{x}_k)^T. \end{aligned} \quad (43)$$

Hybrid estimation with interacting multiple model approach. Even though it is possible to detect faults,

which occur during the estimation process a major drawback of the optimal approach is the exponential grow of possible mode sequences. Thus, the optimal approach can not be realized in practice. An appropriate alternative is a suboptimal hybrid estimation approach such as the interacting multiple model (IMM)

approach that merges mode sequences. As the non-switching case the IMM-approach requires only a filter bank with one Kalman filter for each mode. However, in contrast to the non-switching approach the Kalman filters share information with each other. For index-1 systems, this is done by calculating the initial mean and covariance of each mode according to the mixing step

$$\begin{bmatrix} \tilde{x}_{d,k-1}^{(i)} \\ \tilde{x}_{a,k-1}^{(i)} \end{bmatrix} = \sum_{j=1}^q \mu_{ij} \begin{bmatrix} \hat{x}_{d,k-1|k-1}^{(j)} \\ \hat{x}_{a,k-1|k-1}^{(j)} \end{bmatrix} \quad (44)$$

and

$$\begin{aligned} \tilde{P}_{k-1}^{(i)} &= \begin{bmatrix} \tilde{P}_{x_d x_d, k-1}^{(i)} & \tilde{P}_{x_d x_a, k-1}^{(i)} \\ \tilde{P}_{x_a x_d, k-1}^{(i)} & \tilde{P}_{x_a x_a, k-1}^{(i)} \end{bmatrix} \\ &= \sum_{j=1}^q \mu_{ij} \begin{bmatrix} P_{x_d x_d, k-1|k-1}^{(j)} & P_{x_d x_a, k-1|k-1}^{(j)} \\ P_{x_a x_d, k-1|k-1}^{(j)} & P_{x_a x_a, k-1|k-1}^{(j)} \end{bmatrix} \\ &\quad + \mu_{ij} \begin{bmatrix} \Delta x_{d,ji} \Delta x_{d,ji}^T & \Delta x_{d,ji} \Delta x_{a,ji}^T \\ \Delta x_{a,ji} \Delta x_{d,ji}^T & \Delta x_{a,ji} \Delta x_{a,ji}^T \end{bmatrix} \end{aligned} \quad (45)$$

at time instant $k-1$ with $\Delta x_{d,ji} = \hat{x}_{d,k-1|k-1}^{(j)} - \tilde{x}_{d,k-1}^{(i)}$, $\Delta x_{a,ji} = \hat{x}_{a,k-1|k-1}^{(j)} - \tilde{x}_{a,k-1}^{(i)}$, and the mixing probability

$$\lambda_{k-1}^{(ij)} = P(m_{k-1}^{(i)} | m_k^{(j)}, Y_{k-1}) = \frac{P_{ji} \mu_{k-1}^{(i)}}{\sum_{i=1}^q P_{ji} \mu_{k-1}^{(i)}}. \quad (46)$$

As indicated by (44) and (45) neither the mixed mean nor the mixed covariance matrix of the dynamic states depends on the algebraic states. Note, that the Kalman filter for index-1 systems only estimates the mean and covariance matrix of the differential states, whereas the algebraic states are calculated such that the algebraic constraints are met. Thus, the IMM algorithm only has to be set up for the differential states and has the same structure as for systems given in state space form. Taking this into account, it seems reasonable to sketch briefly the steps of the IMM algorithm. A detailed discussion can be found in [21].

1. Calculate the mixing probabilities $\lambda_{k-1}^{(ij)}$ for each mode $i \in \{1, 2, \dots, q\}$ according to (46).
2. Calculate the mixed initial mean $\tilde{x}_{d,k-1}^{(i)}$ and covariance $\tilde{P}_{x_d x_d, k-1}^{(i)}$ of the differential states at for each mode according to the corresponding parts of (44) and (45).

3. Perform Kalman filtering according to the algorithm given in Section 4 to calculate $\hat{x}_{d,k|k}^{(i)}$, $P_{x_d x_d, k|k}^{(i)}$ and the likelihood function $\Lambda_k^{(i)}$ for each mode.

4. Calculate the mode probabilities

$$\mu_k^{(i)} = P(m_k^{(i)} | Y_k) = \frac{\Lambda_k^{(i)} \sum_{j=1}^q P_{ij} \mu_{k-1}^{(j)}}{\sum_{i=1}^q \Lambda_k^{(i)} \left(\sum_{j=1}^q P_{ij} \mu_{k-1}^{(j)} \right)}. \quad (47)$$

5. Calculate the combined mean and covariance matrix according to (34) and (35), respectively, with the mode conditioned mean values and covariance matrices that have been computed in step 3 and the mode probabilities from step 4.

6 Numerical Results

The considered system is a hydraulic cylinder depicted in Fig. 2. The working cycle that is controlled by pressures imposed at the hydraulic ports pt_1 and pt_2 is intended to execute a mechanical work at the port pt_3 . Further, 5 energy storage elements are modeled. These are compliances of hydraulic compartments c_1 and c_2 , an inertia of moved masses aggregated to i_1 , a compliance of displacement limiting elements represented by se_2 , and a compliance included in the mechanical load model. The transformation from hydraulic to mechanical power takes place at the piston surfaces assigned by tf_1 and tf_2 . Dissipative elements in the hydraulic and mechanical domain are involved by r_1 to r_4 and r_5 to r_6 , respectively. Moreover, the ambient pressure is represented by the element se_1 . Pressures in the hydraulic compartments and positions of the piston and mechanical port are measurable output variables taken with 10 ms sample time uniformly.

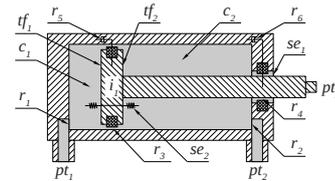


Fig. 2: Simulated Example System, Hydraulic Cylinder

The nonlinear characteristics of the model are induced by equations describing mechanical load, hydraulic resistances, and position limiting forces. Any switching behavior is avoided by introduction of steep sigmoid functions. Furthermore, the time characteristics of the differential states are extremely different. In consequence, the model exhibit highly nonlinear and stiff behavior. Since the index one DAE model include 5 differential and 41 algebraic equations the model is not presented here for sake of space but can be provided by the authors on request.

The working cycle can be splitted into four phases. During the first phase the actuator moves forward until the mechanical load effects a rapid deceleration. Then, in the second phase, the actuator executes effective work. Ensuing, after input pressures change, the actuator moves backward and resumes its original position. In the last phase the system is at rest in the initial position.

Three operation modes $z_1, z_2, z_3 \in \{0, 1\}$ are considered for the system designating nominal operation, operation with increased internal leakage, and operation with increased external leakage. Where $z_{\{1,2,3\}} = 1$ indicates an operation according to the corresponding mode and $z_1 + z_2 + z_3 = 1$. The internal leakage can increase, if the piston sealing of the hydraulic cylinder is worn or defective. Whereas if the piston rod sealing is affected, the system can loose hydraulic fluid and operates with increased external leakage. The operation modes with increased leakage are induced in the simulated model by a modification of the resistance coefficients of elements r_3 and r_4 by a factor $10^{s_{a,1}}$ and $10^{s_{a,2}}$, respectively. Within the hybrid state estimation concept the quantities $s_{a,1}$ and $s_{a,2}$ are estimated by mode allocated Kalman filters calculating corresponding estimates $x_{s,a,1}$ and $x_{s,a,2}$. Since, for the operation modes with increased leakage the system behavior deviates from nominal conditions, these are fault modes and by the calculation of the quantities $x_{s,a,1}$ and $x_{s,a,2}$ a fault identification is implemented.

A relative observable system disturbance $d_{y,\{1,2,3\}}$ for each mode is defined to rate measurable behavior deviation given as

$$d_{y,\{1,2,3\}} = \frac{y_s^{-1} \cdot |y_{\{1,2,3\}} - y_1|}{y_s^{-1} \cdot |y_1|}, \quad (48)$$

$$y_s = \frac{1}{\tau_{wc}} \cdot \int_{\tau}^{\tau + \tau_{wc}} |y_1^T| \cdot dt. \quad (49)$$

Where $y_1, y_2, y_3 \in \mathbb{R}^{n_y}$ are output variables with corresponding dimension $n_y = 4$ for nominal operation, operation with increased internal leakage, and operation with increased external leakage respectively. Further, $\tau_{wc} = 1s$ denotes the working cycle time range, t is the time, and y_s^{-1} is a component wise inverted output scaling factor. This particular definition is suitable for the considered system since $\exists y_s^{-1}$ as well as $y_s^{-1} \cdot |y_1| \neq 0$. Fig. 3 gives the relative observable system disturbances for a particular model and faults parameterization. In accordance to its definition $d_{y,1} = 0$,

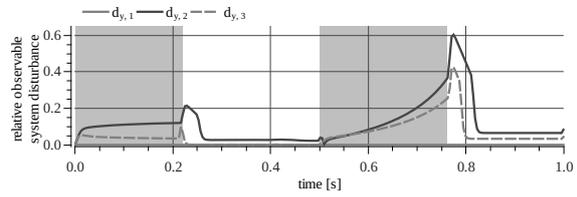


Fig. 3: Simulation results, relative observable system disturbance $d_{y,1}, d_{y,2}, d_{y,3}$

while $d_{y,2}$ and $d_{y,3}$ vary significantly within the working cycle time range. Thus, the system condition referring to fault detection, isolation, and identification (FDI) vary correspondingly. The time average values $\bar{d}_{y,2} \approx 0.11$ and $\bar{d}_{y,3} \approx 0.055$ are set appropriately by adaption of $s_{a,1}$ and $s_{a,2}$ for simulated faulty system operation modes.

A hybrid state estimation is set up for the simulated system utilizing interacting multiple model approach. The simulation results are depicted in the Fig. 4 for 8 working cycles. Altering operation modes $z_{\{1,2,3\}}$ are imposed to the simulated system in different working cycle phases. Obviously, the operation mode estimation is reliably possible for the example system. Remarkably, the algorithm performance correlates strongly with the previously described relative observable system disturbance. Thus, the operation with increased external leakage can be detected less accurate as the corresponding observable disturbance is set significantly smaller. Further notable, the short response time of the estimation process enables operation mode identification within approximately a fourth of the working cycle.

The operation mode estimation accuracy results from the efficiency of the implied Kalman filters. Particularly, the realized differential states estimation error

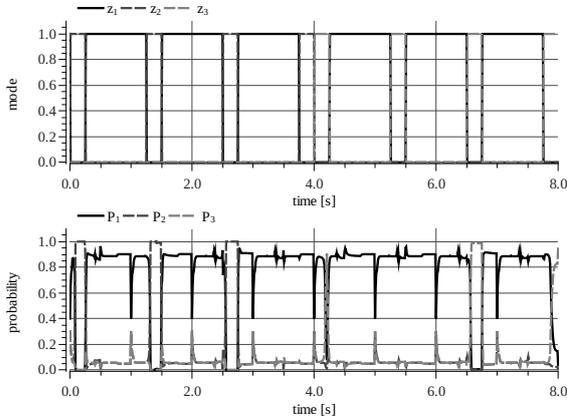


Fig. 4: Simulation results, operation mode estimation with IMM, system operation modes z_1 , z_2 , z_3 and calculated mode probabilities P_1 , P_2 , P_3

is relatively insignificant. On the other hand, the estimation accuracy of the values $s_{a,1}$, $s_{a,2}$ representing system variation in consequence of induced faults is more challenging and correlates again with the relative observable system disturbance. This is reflected in the Fig. 5.

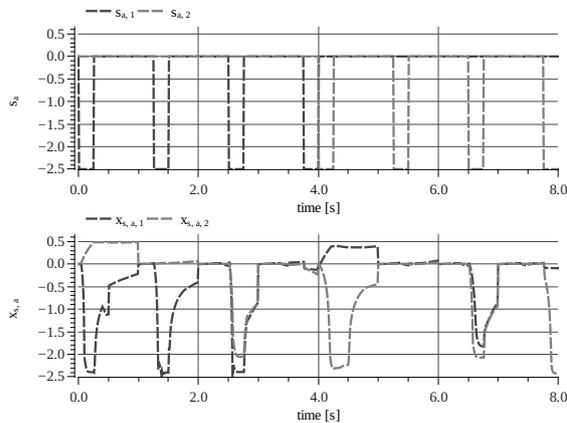


Fig. 5: Simulation results, fault identification, system modification quantities $s_{a,1}$, $s_{a,2}$ and corresponding estimates $x_{s,a,1}$, $x_{s,a,2}$

Concluding, the unscented Kalman filters and interacting multiple model approach is applicable for accurate hybrid state estimation of the described system modeled as an index one DAE.

7 Conclusion

In this article a hybrid estimation scheme for nonlinear DAE system with index-1 is proposed and used for fault identification. Since the hybrid estimation is based on the interacting multiple model approach a generalization of this approach for DAE systems is given. Therefore, properties of DAE systems and modeling of systems subject to faults are discussed in detail. A primary advantage of the presented approach is a direct integration of dynamic equations resulting from object oriented modeling. Object oriented modeling is a modeling concept widely established in industrial practice.

The fault identification approach is illustrated by a hydraulic cylinder that is modeled as a nonlinear DAE system. Furthermore, a set of operation modes is considered corresponding to system characteristics subject to probable faults. Even though the model properties are challenging and the magnitudes of the faults are rather small, the numerical results of the system subject to several mode changes show that, the operation mode as well as the state and fault estimation are estimated reliably.

References

- [1] I. Alkov and D. Weidemann. Unscented kalman filter for higher index nonlinear differential-algebraic equations. In *Proc. of the Methods and Models in Automation and Robotics (MMAR), 2014 19th International Conference*, pages 88–93, 2014.
- [2] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and fault-tolerant control*. Springer, 2006.
- [3] K. E. Brenan and L. R. Petzold. The numerical solution of higher index differential/algebraic equations by implicit methods. *SIAM Journal on Numerical Analysis*, 26(4):976–996, 1989.
- [4] S. L. Campbell and C. W. Gear. The index of general nonlinear daes. *Numerische Mathematik*, 72(2):173–196, 1995.
- [5] D.W. Chen and M. Saif. Observer-based strategies for actuator fault detection, isolation and

- estimation for certain class of nonlinear systems. *IET Control Theory and Applications*, 1(6):1672–1680, 2007.
- [6] P. Deuffhard, E. Hairer, and J. Zugck. One-step and extrapolation methods for differential-algebraic systems. *Numerische Mathematik*, 51(5):501–516, 1987.
- [7] S.X. Ding. *Model-based fault diagnosis techniques*. Springer, 2008.
- [8] J. D. Hedengren, R. Asgharzadeh Shishavan, K. M. Powell, and T. F. Edgar. Nonlinear modeling, estimation and predictive control in AP-Monitor. *Computers & Chemical Engineering*, 70(0):133 – 148, 2014.
- [9] M.W. Hofbaur. *Hybrid Estimation of complex systems*. Springer, 2005.
- [10] M.W. Hofbaur and B.C. Williams. Mode estimation of probabilistic hybrid systems. In *Intl. Conf. on Hybrid Systems: Computation and Control*, pages 253–266. Springer Verlag, 2002.
- [11] R. Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing - tutorial paper. *Automatica*, 29(4):815–835, 1993.
- [12] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc. of the American Control Conference*, volume 3, pages 1628–1632, 1995.
- [13] P. Kunkel, V. Mehrmann, and I. Seufer. GENDA: A software package for the solution of general nonlinear differential-algebraic equations. *Institut für Mathematik*, 730(2), 2002.
- [14] R. K. Mandela, R. Rengaswamy, and S. Narasimhan. Nonlinear state estimation of differential algebraic systems. *Advanced Control of Chemical Processes*, 7:792–797, 2009.
- [15] S. E. Mattsson and G. Söderlind. Index reduction in differential-algebraic equations using dummy derivatives. *SIAM Journal on Scientific and Statistical Computing*, 14(3):677–692, may 1993.
- [16] V. Mehrmann. Index concepts for differential-algebraic equations. *Institute of Mathematics, TU Berlin*, pages 3–2012, 2012.
- [17] N. S. Nedialkov and J. D. Pryce. Solving differential-algebraic equations by taylor series (i): Computing taylor coefficients. *BIT Numerical Mathematics*, 45(3):561–591, 2005.
- [18] C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM Journal on Scientific and Statistical Computing*, 9(2):213–231, 1988.
- [19] C.P. Tan and C. Edwards. Sliding mode observers for reconstruction of simultaneously actuator and sensor faults. In *Proc. of the IEEE Conference on Decision and Control*, pages 1455–1460, 2003.
- [20] R. van der Merwe and E. A. 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*, volume 6, pages 3461–3464, 2001.
- [21] Bar-Shalom Y. and Li X.-R. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, 1993.
- [22] Y.M. Zhang and J. Jiang. Active fault-tolerant control system against partial actuator failures. *IEE Proceedings on Control Theory Applications*, 149(1):95–104, 2002.

Implementation of Hybrid Systems Described by DEV&DESS in the QSS Based Simulator PowerDEVS

Franz Preyser¹, Irene Hafner², Matthias Rößler²

¹Institute for Analysis and Scientific Computing, Vienna University of Technology,
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

²Simulation Services, dwh GmbH, Neustiftgasse 57-59, 1070 Vienna, Austria
franz.preyser@tuwien.ac.at

In this article, a method for implementing hybrid models, formulated as DEV&DESS, in the QSS based simulator PowerDEVS is presented. PowerDEVS is actually a pure DEVS simulator. However, it is specialised for simulating continuous Models (DESS) using QSS (Quantized State System) to translate them into discrete event models (DEVS). Therefore, it is perfectly suited for the simulation of hybrid models (DEV&DESS). When designing and simulating coupled DEVS models though, very soon a lot of difficulties occur caused by concurrent events and feedback loops. Hence, first some concepts are introduced of how to implement an atomic DEVS in a way that avoids those difficulties. Afterwards, an approach of how to implement an atomic DEV&DESS is introduced which makes use of those concepts.

1 Introduction

The factor ‘costs for energy consumption’ is gaining more and more importance in terms of production process optimization. Since most of the time energy intense processes in a production line are of continuous nature a demand is rising for being able to formulate those continuous aspects in addition to the usual discrete logistical aspects in one simulateable model [9]. See [11] for preceding work on this issue.

Herbert Praehofer introduced DEV&DESS (Discrete Event & Differential Equation System Specification) [10] for the formal description of hybrid systems. DEV&DESS is based on the two formalisms DEVS and DESS invented by Bernard Zeigler [13] for the description of discrete event systems and continuous systems, respectively.

However, as digital computers work in a purely discrete way, DESS models and therefore also DEV&DESS models have to be discretised somehow before they can be simulated on a digital computer. The usual method is to apply an ODE solver onto the differential equations describing the continuous model. A rather new alternative is called QSS (Quan-

tized State System) [8] which is already mentioned in [13] as it transforms the continuous model (DESS) into an discrete event one (DEVS). Ernesto Kofman introduced a set of advancements to QSS [3], [4], [5], [6], [7] which have been implemented in PowerDEVS [1]. PowerDEVS is a DEVS simulator that supports graphical block orientated model description comparable to Simulink or Dymola.

Although PowerDEVS is specially designed for implementing continuous models in an discrete event manner, so far there has been no way to directly implement a DEV&DESS. A formal method of how to embed DEV&DESS in DEVS is presented in [12]. Based on this work, a generic PowerDEVS DEV&DESS block is developed.

However, creating and simulating coupled DEVS models turns out to be quite difficult. This mainly is owed to the various numbers of possible scenarios of concurrent events that may occur during simulation and that have to be considered when defining the DEVS of each single block involved. To relieve the modeller of these difficulties, a new PowerDEVS *Atomic DEVS* block is introduced, based on the DEVS extension *Parallel DEVS* (P-DEVS, see [2]). The

thereby developed methods for concurrency treatment are then also utilized for the mentioned generic DEV&DESS block, called *Atomic DEV&DESS*.

1.1 DEVS

DEVS denotes a formalism for describing systems that allow changes only at discrete points in time called *events*. An atomic DEVS is specified by the following 7-tuple.

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

where

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

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

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

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

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

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

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

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

Figure 1 illustrates a DEVS graphically. It basically consists of an inner state s that can be altered by the external and internal state transition functions which are evaluated each time an external or an internal event occurs. External events are triggered by arriving input messages x , whereas internal events are triggered when the current state s has not change for the duration of $ta(s)$. On each internal event, right before δ_{int} is called, the output function λ is evaluated which may result in an output message y .

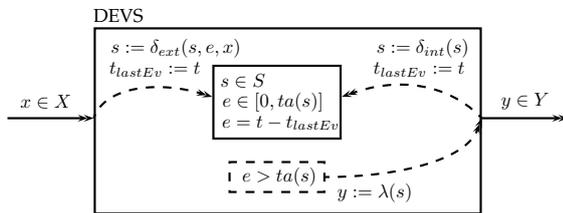


Figure 1: Graphical illustration of an atomic DEVS.

As an atomic DEVS has input and output ports, it can be coupled with other atomic DEVS resulting in a *coupled DEVS* whose behaviour again can be described by an atomic DEVS (see closure under coupling property in [13]). Figure 2 illustrates some pos-

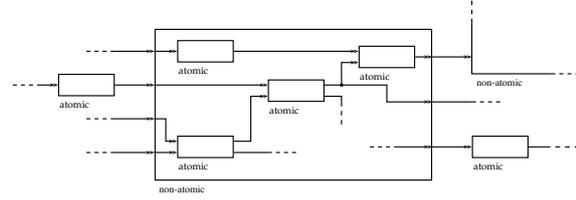


Figure 2: Coupling scheme of DEVS blocks.

sible coupling schemes of atomic and coupled DEVS. Two types of DEVS can be distinguished:

Definition 1.1 Mealy Type DEVS

A DEVS is called Mealy Type DEVS or of Type Mealy, if there exists an internal state s and an external input x in such a way that $ta(\delta_{ext}(s, x)) = 0$. Thus, a model described by a mealy type DEVS may produce an output as immediate response to an input.

Definition 1.2 Moore Type DEVS

A DEVS is called Moore Type DEVS or of Type Moore, if it is not of type mealy.

1.2 DEV&DESS

DEV&DESS formalism is a composition of DEVS and DESS and therefore capable of describing hybrid systems. An atomic DEV&DESS can be described by the following 11-tuple:

$$\langle X^{discr}, X^{cont}, Y^{discr}, Y^{cont}, S, \delta_{ext}, C_{int}, \delta_{int}, \lambda^{discr}, f, \lambda^{cont} \rangle$$

where

X^{discr}, Y^{discr} ... set of discrete inputs and outputs

X^{cont}, Y^{cont} ... set of continuous inputs and outputs

$S = S^{discr} \times S^{cont}$... set of states (=state space)

$$Q = \{(s^{discr}, s^{cont}, e) \mid s^{discr} \in S^{discr}, s^{cont} \in S^{cont}, e \in \mathbb{R}_0^+\}$$

$\delta_{ext} : Q \times X^{cont} \times X^{discr} \rightarrow S$... external transition fct.

$\delta_{int} : Q \times X^{cont} \rightarrow S$... internal transition function

$\lambda^{discr} : Q \times X^{cont} \rightarrow Y^{discr}$... discrete output function

$\lambda^{cont} : Q \times X^{cont} \rightarrow Y^{cont}$... continuous output function

$f : Q \times X^{cont} \rightarrow S^{cont}$... rate of change function

$C_{int} : Q \times X^{cont} \rightarrow \{true, false\}$... event condition fct.

As it can be seen, apart from ta , each component of an atomic DEVS is recurring in an atomic DEV&DESS. Additionally there are the right side of the ODE f and the continuous output function λ^{cont} describing the DESS part. The state event condition function C_{int} though, is new. It is responsible for triggering internal events in the DEV&DESS and therefore replaces ta . However, C_{int} does not only trigger time events, but also state events, i.e. events caused by the internal state q reaching some specific value. Nevertheless, both time and state events result in an execution of δ_{int} .

Figure 3 shows a graphical illustration of an atomic DEV&DESS. As well as DEVS also DEV&DESS is closed under coupling. However, continuous output ports cannot be coupled arbitrary to discrete input ports, but only if their output value is guaranteed to be piecewise constant. See [13] for more details about that and about DEVS and DEV&DESS in general.

2 DEVS Simulation in PowerDEVS

In PowerDEVS a DEVS of an atomic block consists of the specification of the six C++ functions: $init(t, parameters)$, $ta(t)$, $dint(t)$, $dext(x, t)$, $lambda(t)$ and $exit()$. These functions are member functions of a C++ class describing the block. Thus, state variables, block parameters and auxiliary variables are defined as member variables of that class making them accessible in all the member functions.

These member functions are called by the PowerDEVS simulation engine whenever their execution is necessary to calculate the models dynamic behaviour. The function $init(t, parameters)$ is called right before the actual simulation is started and can be used to initialise the system's state and to read block parameter values that have been entered by the modeller using PowerDEVS graphical user interface and the block's *Parameters Dialogue*. The function $exit()$ is called after the simulation is finished and thus, can be used for example to free allocated memory.

The other methods represent the corresponding functions of the DEVS formalism. In case of an internal event they are executed in the following order:

1. call of $lambda(t)$
2. $e = t - t1$
3. call of $dint(t)$
4. $t1 = t$
5. call of $ta(t)$: $t_n = t + ta(t)$

where $t1$ denotes the time of the last event, and t_n the time of the next event in the corresponding DEVS.

In case of an external event they are executed in the following order:

1. $e = t - t1$
2. call of $dext(x, t)$
3. $t1 = t$
4. call of $ta(t)$: $t_n = t + ta(t)$

In coupled DEVS the so-called *Select* function, a kind of priority ranking of the involved blocks, selects which DEVS is allowed to execute its internal transition function first, when several of them are scheduled simultaneously. In PowerDEVS the *Select* function is implemented as a list in which all blocks occurring in a coupling are sorted descending according their priority in case of concurrent internal events. However, if a block produces an output message, the external transition function at the receiving block is always executed immediately, independent of its priority.

3 Problems with DEVS and Solution Approaches

3.1 Problem Identification

The definition of an atomic DEVS behaving exactly as intended in every possible situation of concurrent input messages at different input ports turns out to be quite challenging. The *Select* function regulating the resolution of such concurrencies is part of the coupling but nevertheless influences the behaviour of an atomic DEVS in such situations and therefore, it is hard to consider when formulating the atomic DEVS. For example, if an atomic DEVS with the current internal state s receives the input messages x_1 at its input

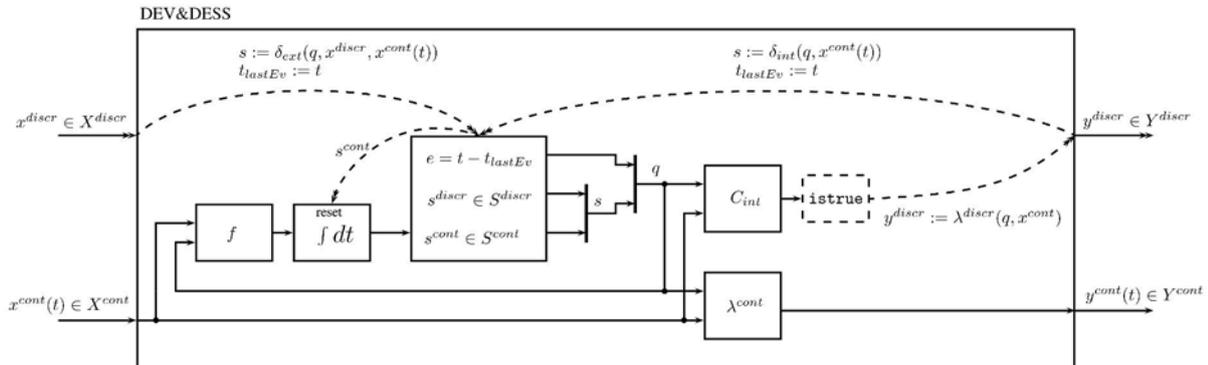


Figure 3: Graphical illustration of an atomic DEV&DESS.

port one and x_2 at its input port two at the same instant of simulation time, it depends on the *Select* function whether its new state calculates as $\delta_{ext}(\delta_{ext}(s, x_2), x_1)$ or as $\delta_{ext}(\delta_{ext}(s, x_1), x_2)$. Even more, if the block is of type Mealy, it may depend on the *Select* function if an output message is produced in reaction to the input messages or not.

Therefore, to define a DEVS in a rigorous way, it is necessary to consider each possible set of concurrent input messages and moreover, every possible treatment order.

3.2 Parallel DEVS Approach

The DEVS extension *Parallel DEVS* counters the problems mentioned in section 3.1 by collecting all input messages in so called *bags* before treating them all at once in one single call of δ_{ext} . If at the same time also an internal event is triggered, a so called *confluent transition function* $\delta_{conf} : Q \times X \rightarrow S$ is applied instead of δ_{ext} .

The idea is to first calculate λ of each immanent block, i.e. of each block experiencing an internal transition at the current simulation time, before calculating δ_{int} or δ_{conf} of any of them. However, due to blocks of type Mealy and due to feedbacks in the coupling it may be necessary to recalculate λ if the set of arrived input messages of the corresponding block has changed after its former calculation. Therefore, the determination of a stable set of input messages for each block is an iterative process. It even is possible that finally the stable set of input messages is empty although having not been empty in former iteration

which makes it necessary to undo the state changes that may have been accomplished by a call of δ_{ext} .

If a coupled model contains an algebraic loop, it may not be possible to determine a stable set of input messages for each block in a finite number of iterations. Such models are called *illegitimate*.

3.3 DEVS Approach in PowerDEVS

As in Parallel DEVS the execution of λ is decoupled from the succeeding execution of the corresponding δ function, P-DEVS works with a simulation engine different to the one used for DEVS. PowerDEVS does not support Parallel DEVS simulation. Nevertheless, it is possible to implement P-DEVS functionality in PowerDEVS.

For this purpose, three mechanisms have to be installed. The first one addresses the gathering of concurrent input messages in a set \mathbf{x} (bold letters denote sets). In order to do that the internal state of the DEVS is extended by an input buffer representing \mathbf{x} and the external transition function only is allowed to change the input buffer (by storing arriving messages with their arrival time in it). The actual state change caused by the external event is shifted into the internal transition which is executed every time the input buffer has been modified. Thus, it is made sure that all input messages coming from blocks with higher priority are gathered in \mathbf{x} before they are treated. However, due to feedback couplings it cannot be avoided that some input messages origin from blocks with lower priority. This problem is addressed by the second mechanism.

The second mechanism consists of a backing up of

the state s of the DEVS every time the first event at the current simulation time is triggered. In PowerDEVS this can be identified by $t1 < t$. The δ functions (δ_{int} , δ_{ext} , δ_{conf}) are then using the backup s_{old} instead of s to calculate a new state. Therefore, if \mathbf{x} is changed after one of the δ functions has already been evaluated it simply is re-evaluated: $s = \delta(s_{old}, \mathbf{x})$. As every calculation of δ is preceded by a calculation of λ though, it may occur that formerly output messages have been produced at output ports where finally no output messages are to be sent. However, these output messages have already altered the set \mathbf{x} of its receiving blocks. These alterations have to be withdrawn somehow. This is what the third mechanism is dedicated to.

Like \mathbf{x} also the set of output messages \mathbf{y} is iteratively changing and hopefully finally stabilising. Thus, also for the output messages an output buffer is installed as part of the state of the DEVS. Each calculated output message is stored in it at the corresponding output port with its time of last change and with an 'already sent - flag'. So if λ is recalculated and there exists an entry in the output buffer that has been already sent at the current simulation time but is not marked to be resent (as it is not included in the result of λ anymore) a *retrieve message* is sent instead informing the receiving block to ignore the formerly received message and to recalculate its own λ and δ function.

4 Atomic PDEVS Block

The created *Atomic PDEVS* PowerDEVS library block is intended to overcome the problems mentioned in section 3.1 by implementing the three mechanisms discussed in section 3.3. In the following, its working principle is expressed in form of a description of the content of the C++ functions corresponding to δ_{ext} , ta , λ , and δ_{int} . Of course, a complete description of each detail that need to be considered when programming the *Atomic DEVS* block would go beyond the scope of this article.

1. $d_{ext}(x, t)$:
 - If $t1 < t$ set $s_{old} = s$, $\sigma_n = \sigma - e$, $\sigma_{n,old} = \sigma_n$ and $flag = 'n'$ but if additionally $\sigma_n = 0$ set $flag = 'i'$.
 - If x is a retrieve message, remove the corresponding entry from \mathbf{x} , set $\sigma = 0$

and if $\mathbf{x} = \emptyset$ set $flag = 'n'$.

Else, if x differs from the last reception at the same port in value or in arrival time, modify \mathbf{x} accordingly and set $\sigma = 0$ and update $flag$: $'n' \mapsto 'e'$, $'i' \mapsto 'c'$.

2. $ta(t)$:
Return σ .
3. $lambda(t)$:
 - If $t1 < t$ set $s_{old} = s$ and $flag = 'i'$
 - If there is no pending output message, calculate $\mathbf{y} = \lambda(s_{old}, \mathbf{x})$, and add retrieve messages to \mathbf{y} .
 - If there are pending output messages left in \mathbf{y} , output one of them.
4. $d_{int}(t)$:
If there are no pending output messages and no newly arrived input messages:
 - if $flag = 'i'$ calc. $[s, \sigma_n] = \delta_{int}(s_{old})$.
 - if $flag = 'e'$ calc. $[s, \sigma_n] = \delta_{ext}(s_{old}, \mathbf{x})$.
 - if $flag = 'c'$ calc. $[s, \sigma_n] = \delta_{conf}(s_{old}, \mathbf{x})$.
 - if $flag = 'n'$ set $[s, \sigma_n] = [s_{old}, \sigma_{n,old}]$.
 - set $\sigma = \sigma_n$.

The variable $flag$ is used to identify the type of event and therefore, which particular δ function is to be used for calculating the new state. The message retrieving mechanism though, only works if all blocks involved in a coupling implement it.

5 Atomic DEV&DESS Block

Based on the Atomic PDEVS block, an *Atomic DEV&DESS* PowerDEVS library block is developed.

5.1 Structure

The structure of the DEV&DESS embedded in PowerDEVS is depicted in Figure 4. The idea is to divide a DEV&DESS into a continuous (DESS) part that can be implemented graphically as block diagram and into a discrete part that can be implemented as atomic PDEVS block. For this purpose the function C_{int} is also divided into two parts: one for detecting state events (C_{int}^{se}) and one for scheduling time events

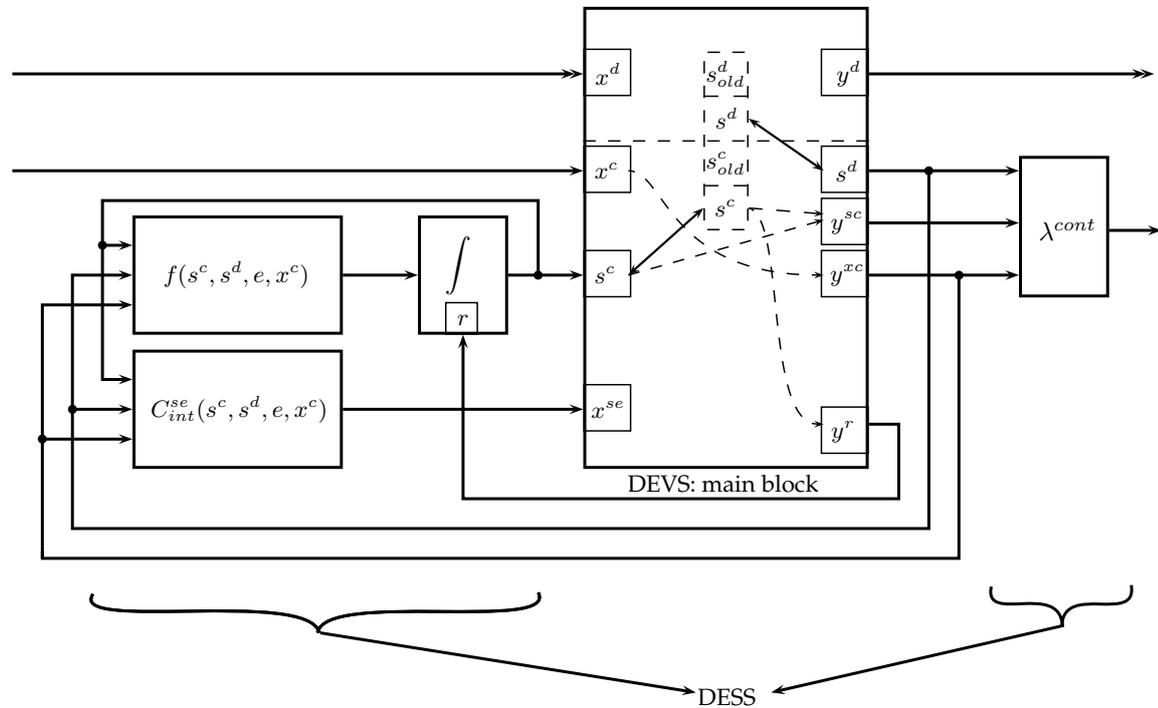


Figure 4: Graphical illustration of the construction of a DEV&DESS in PowerDEVS.

(C_{int}^e). The first one is a component of the continuous part and the second one is included in the atomic PDEVS definition.

The continuous part consists of λ^{cont} , f , an integrator \int , and of C_{int} . The discrete part, responsible for the implementation of δ_{ext} , δ_{int} , λ^{discr} , and C_{int}^e is realized as one single DEVS referred to as *main block*. The priority list of the coupling in Figure 4 is as follows: f , \int , C_{int}^{se} , *main block*, λ^{cont} .

5.2 QSS Signals

All continuous signals in DEV&DESS are described as QSS signals in PowerDEVS. That is, they are described as piecewise polynomial functions with jump discontinuities at the merging points of two polynomials which are smaller than a requested constant called *quantum* q . The polynomials represent the Taylor polynomials of the continuous signal with the expansion point at the discontinuities. Every time the difference between the current Taylor polynomial and the continuous signal or between Taylor polynomial and a Taylor approximation of higher order becomes equal

to q , the expansion point is advanced in time and the coefficients of the new polynomial are sent as DEVS message. In this way, a continuous signal can be described in a discrete event manner. As in DESS piecewise continuous signals are allowed, jump discontinuities bigger than q may appear as well. Anyway, they are not caused by the signal discretisation but have already existed before.

5.3 State Events

Using QSS, state events, triggered by the continuous state reaching a specific value, simply can be calculated as the points in time when the current polynomial signal representation reaches the specified value. Thus, state events are transformed into time events.

All the signal lines in Figure 4 may transmit vectorial signals. However, in PowerDEVS vectorial signals are not sent at once but index by index resulting in temporally non-valid signals during the time after the first changed index is sent and before the last changed index is sent. This in turn may lead to state events wrongly indicated by C_{int}^{se} .

Further there is the case in which a state transition in the main block leads to a state that immediately triggers a state event. This case is declared to be prohibited. That is, the DEV&DESS is not allowed to define state transitions leading immediately to a state event.

5.4 Main Block

As the main block calculates δ_{ext} and δ_{int} it has to administer the entire state of the system consisting of a discrete part s^d and of a continuous part s^c . However, in between two state transitions in the main block s^c may change due to Taylor polynomial expansion point advancement triggered by the integrator. Therefore, the third input port of the main block is coupled to the integrator's output port. The input buffer of that port is simultaneously used as storage for the continuous state s^c itself. The same holds for s^d and the output buffer of the second output port. The fourth input port of the main block receives the result of the calculation of C_{int}^{se} and thus, input messages at this port trigger internal transitions.

f , C_{int}^{se} , and λ^{cont} depend on the continuous input signal x^c . However, they are not coupled directly to it but indirectly through the main block. This is to not forward each single, maybe even only temporary index change in x^c immediately to the continuous part. Instead the whole new x^c is forwarded just before the main block calculates one of the δ functions as it has to evaluate C_{int}^{se} before to being able to decide which one.

In the following, the working principle of the main block is expressed in form of a description of the content of the C++ functions corresponding to δ_{ext} , ta , λ , and δ_{int} .

1. `dext(x, t)`:
 If $t_1 < t$ set $s_{old} = s$, $\sigma_n = \sigma - e$, $\sigma_{n,old} = \sigma_n$,
 phase='g' and flag='n',
 if additionally $\sigma_n = 0$ set flag='i'.
 If input at port:
 - x^d : add/remove x to/from \mathbf{x}^d .
 update flag:
 if $\mathbf{x} = \emptyset$ 'e' \mapsto 'n', 'c' \mapsto 'i'
 otherwise, 'n' \mapsto 'e', 'i' \mapsto 'c'
 - x^c : add/remove x to/from \mathbf{x}^c .
 - s^c : add change in s^c to \mathbf{y}^{sc} .
 if phase='g', apply change in s^c to s_{old}^c

- x^{se} : add x to \mathbf{x}^{se}
- If any input buffer changed, set $\sigma = 0$.
2. `ta(t)`:
 Return σ .
 3. `lambda(t)`:
 If $t_1 < t$ set $s_{old} = s$, $\sigma_n = \sigma - e$, $\sigma_{n,old} = \sigma_n$,
 phase='l' and flag='i'
 if phase=
 - 'g' Set phase='c'.
 - 'c' If \mathbf{x}^c changed, or $s \neq s_{old}$, forward \mathbf{x}^c to \mathbf{y}^{xc} , set $s^d = s_{old}^d$ and $\mathbf{y}^r = s_{old}^c$.
 Further set phase='r'.
 Else set phase='C'.
 - 'r' If there is a pending output left in \mathbf{y} ,
 set $\mathbf{x}^{se} = \emptyset$ and output next message.
 Else, set phase='C'.
 - 'C' If $x^{se} \neq \emptyset$, update flag: 'n' \mapsto 'I',
 'e' \mapsto 'C',
 phase='l'
 - 'I' if flag='i' | 'I':
 calc. $[s^d, s^c, \sigma_n] = \delta(s_{old}^d, s_{old}^c, e, \mathbf{x}^c)$,
 and $\mathbf{y}^d = \lambda^{discr}(s_{old}^d, s_{old}^c, e, \mathbf{x}^c)$.
 - if flag='e':
 calc. $[s^d, s^c, \sigma_n] = \delta(s_{old}^d, s_{old}^c, e, \mathbf{x}^c, \mathbf{x}^d)$,
 and $\mathbf{y}^d = \lambda^{discr}(s_{old}^d, s_{old}^c, e, \mathbf{x}^c)$.
 - if flag='c' | 'C':
 calc. $[s^d, s^c, \sigma_n] = \delta(s_{old}^d, s_{old}^c, e, \mathbf{x}^c, \mathbf{x}^d)$,
 and $\mathbf{y}^d = \lambda^{discr}(s_{old}^d, s_{old}^c, e, \mathbf{x}^c)$.
 Add retrieve messages to \mathbf{y}^d .
 Add each change in s^c to \mathbf{y}^r .
 Set phase='o'.
 - 'o' if unsent element of \mathbf{y}^r left, send it.
 Else, if s^c changed, update \mathbf{y}^{sc} .
 Else, if unsent element of \mathbf{y}^d left, send it.
 Else, if unsent change of s^d left, send it.
 Else, if unsent element of \mathbf{y}^{sc} left, send it.
 4. `dint(t)`:
 If pending output is left and there are no
 new input messages at input ports x^d and x^c ,
 set $\sigma = \sigma_n$.

As the new state is part of the output of the main block the order in which λ and δ are calculated is reversed here. This is why δ is now calculated in the C++ function `lambda` instead of in `dint`.

Since each time before δ and λ are calculated, C_{int}^{se} has to be evaluated, there are several different phases in

λ to be distinguished. This is what the variable phase is for.

Although the description above is quite elaborate, it still does not cover every detail of the complete source code of the *Atomic DEV&DESS* block. However, the basic principles are included.

6 Conclusion

With the ever growing computational power of modern computers also the possibilities to simulate more and more extensive and complex models increase. However, when including more and more details into a model, very soon a point is reached where pure discrete or pure continuous models do not suffice anymore. Production process models which include energy consumption behaviour provide an example for this trend. As a consequence the formulation and simulation of hybrid models is demanded. Concerning the formulation, DEV&DESS seems to be quite powerful. So far though, it lacks a simulator able to rigorously implement and simulate a DEV&DESS. An approach to implement DEV&DESS in PowerDEVS has been demonstrated in this paper. As the correct definition of coupled DEVS models is quite challenging, additionally a way of how to implement models in PowerDEVS similar to Parallel DEVS has been introduced. However, profound theoretical investigations and a formal proof of the correctness of the presented approaches are works that still needs to be done.

References

- [1] F. Bergero and E. Kofman. PowerDEVS. A Tool for Hybrid System Modeling and Real Time Simulation. *Simulation: Transactions of the Society for Modeling and Simulation International*, 87(1–2):113–132, 2011.
- [2] Alex Chung Hen Chow and Bernard P. Zeigler. Parallel devs: a parallel, hierarchical, modular, modeling formalism. In *Winter Simulation Conference'94*, pages 716–722, 1994.
- [3] E. Kofman. A Second Order Approximation for DEVS Simulation of Continuous Systems. *Simulation: Transactions of the Society for Modeling and Simulation International*, 78(2):76–89, 2002.
- [4] E. Kofman. Quantized-State Control. A Method for Discrete Event Control of Continuous Systems. *Latin American Applied Research*, 33(4):399–406, 2003.
- [5] E. Kofman. Discrete Event Simulation of Hybrid Systems. *SIAM Journal on Scientific Computing*, 25(5):1771–1797, 2004.
- [6] E. Kofman. A Third Order Discrete Event Simulation Method for Continuous System Simulation. *Latin American Applied Research*, 36(2):101–108, 2006.
- [7] E. Kofman. Relative Error Control in Quantization Based Integration. *Latin American Applied Research*, 39(3):231–238, 2009.
- [8] E. Kofman and S. Junco. Quantized State Systems. A DEVS Approach for Continuous System Simulation. *Transactions of SCS*, 18(3):123–132, 2001.
- [9] N. Popper, I. Hafner, M. Rössler, F. Preyser, B. Heinzl, P. Smolek, and I. Leobner. A General Concept for Description of Production Plants with a Concept of Cubes. *Simulation Notes Europe*, 24(2):105–114, 2014.
- [10] H. Praehofer. Systems Theoretic Formalisms for Combined Discrete-Continuous System Simulation. *International Journal of General Systems*, 19(3):219–240, 1991.
- [11] A. Schmidt and T. Pawletta. Hybride modellierung fertigungstechnischer prozessketten mit energieaspekten in einer ereignisdiskreten simulationsumgebung. In *ASIM 2014 – 22. Symposium Simulationstechnik, Berlin, 03.-05.09.2014*, pages 109–116. ARGESIM/ASIM, 2014.
- [12] B.P. Zeigler. Embedding dev&dess in devs. In *DEVS Integrative M&S Symposium (DEVS'06)*.
- [13] B.P. Zeigler, H. Praehofer, and T.G. Kim. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, 2000.

Sitzung 7

Raum 116

Adaptive Modellierung / Simulation

(Sitzungsleitung: Christine Wahmkow)

Richtlinien zum Erstellen von strukturvariablen Modellen

Alexandra Mehlhase

TU Berlin

Multi-Resolution Car2X-Simulation zur virtuellen Erprobung von Fahrerassistenzsystemen

Manuel Schiller

TU München

Potential of Dynamically Adaptable Simulation Models for Virtual Commissioning

Philipp Puntel Schmidt

TU Hamburg

2Simulate Modeling Language:

Wiederbelebung und Modernisierung von alter Simulationssoftware

Jürgen Gotschlich

DLR

Richtlinien zum Erstellen von strukturvariablen Modellen

Alexandra Mehlhase, TU Berlin Institut Softwaretechnik und theoretische Informatik

a.mehlhase@tu-berlin.de

Kurzfassung

In der Modellierung technischer Systeme geht es darum Modelle zu erstellen, welche zu jeder Zeit während einer Simulation genau die Dinge berücksichtigen, die in der jeweiligen Phase von Interesse sind. In den meisten Simulationen wird jedoch mit genau einem Modell gearbeitet, wodurch für eine Simulation genau festgelegt ist, was in ihr betrachtet wird. Reale Systeme durchlaufen aber oft verschiedene Phasen in denen unterschiedliche physikalische Eigenschaften von Interesse sind. Diese Phasen sollten mit unterschiedlichen Modellen beschrieben werden zwischen denen situationsbedingt gewechselt wird. Modelle, die aus mehreren Modellen (*Modi*) bestehen, werden *strukturvariable Modelle* genannt. Dabei wird von einem Modus zum anderen abhängig von Wechselbedingungen gewechselt. Während dieser Wechsel muss der nächste Modus initialisiert werden, damit die Simulation durch den neuen Modus fortgesetzt wird.

Die meisten Arbeiten über strukturvariable Modelle beschäftigen sich mit der Simulation dieser Modelle, so ist eine Simulation bereits in prototypischen Werkzeugen möglich (Bsp. DySMo, SOL, MOSILAB).

Es ist bisher jedoch wenig darüber bekannt, wie diese Modelle aufgebaut sein sollten, um vorteilhaft in der Modellierung und Simulation eingesetzt zu werden.

Aus diesem Grund sollen in diesem Vortrag die Herausforderungen zur strukturvariablen Modellierung vorgestellt und Hinweise zum Umgang mit diesen gegeben werden. Dabei wird zunächst auf die Architektur der Modelle eingegangen. Es geht dabei zum einen um den allgemeinen Aufbau und die notwendigen Eigenschaften der Modi und Transitionen zwischen diesen Modi. Zum anderen geht es um den hierarchischen und modularen Aufbau der Modelle. Die Idee ist, dass, wie auch in konventionellen Modellen, Bibliotheken von Komponenten erstellt werden. Dabei haben diese Komponenten jedoch bereits Modi. Ein Verbinden von mehreren Komponenten mit Modi bedeutet eine flexibel anpassbare Struktur der Modelle während der Simulation. Jedoch bringt gerade dieses Konzept neue Herausforderungen, so müssen die Schnittstellen der Komponenten, die Wechselbedingungen und Initialisierung bei Moduswechseln detailliert betrachtet werden.

Zusätzlich zur Architektur wird die Auslegung der Moduswechsel detailliert betrachtet; also unter welcher Bedingung sollte wie zu einem anderen Modus gewechselt werden. Dabei wird zum einen darauf eingegangen, wie die Wechselbedingungen ausgelegt sein sollten, um einem möglichst stetigen Übergang bei einem Moduswechsel zu erreichen. Zum anderen geht es darum, wie der neue Modus anhand von bekannten Daten initialisiert werden kann, welche Konzepte dazu notwendig sind und was dabei zu beachten ist.

Die Architektur, Wechselbedingungen und Initialisierungen werden detailliert betrachtet und die Herausforderungen im Vergleich zur konventionellen Modellierung erklärt. Es werden dann Hinweise bereitgestellt, um mit den neuen Herausforderungen umzugehen, sodass strukturvariable Modelle zukünftig vorteilhaft eingesetzt werden können.

Multi-Resolution Car2X-Simulation zur virtuellen Erprobung von Fahrerassistenzsystemen

Manuel Schiller¹, Alois Knoll¹

¹Lehrstuhl für Echtzeitsysteme und Robotik, Technische Universität München
manuel.schiller@in.tum.de

1 Problemstellung

Die Erprobung von Fahrerassistenzsystemen (FAS) bedingt umfangreiche Versuchsfahrten und Test-szenarien sowohl in der Vorentwicklung als auch in der Serienentwicklungsphase. Herkömmliche FAS die ausschließlich auf fahrzeugeigener Sensorik basieren werden in der Regel durch reale Erprobungs-fahrten validiert. Da auch in diesem Entwick-lungsfeld der Testraum mit zunehmender Komplexität der Funktionen der FAS wächst, werden verstärkt sim-ulationsbasierte Verfahren zur virtuellen Erprobung eingesetzt. Im Gegensatz zur detaillierten Simulation eines einzelnen Fahrzeugs und dessen Sensordaten-verarbeitung steht die Anforderung zukünftiger FAS welche die Informationen anderer Verkehrsteilnehmer durch Funkkommunikation mitberücksichtigen. Die Testkomplexität bei Validierung vernetzter FAS der nächsten Generation steigt durch die Einflüsse und Wechselwirkungen einer Vielzahl miteinander ver-netzter Fahrzeuge signifikant an. Es besteht da-her die Notwendigkeit umfangreiche Testszenarien in virtuellen Fahrversuchen abzubilden.

Das Gesamtsystem Car2X-Kommunikation kann durch die drei in Wechselwirkung stehenden Subsyste-me Verkehr, Anwendung und Kommunikation mod-elliert werden (vgl. Abbildung 1) [2]. Bisherige Ansätze der Car2X-Simulation stellen zum Großteil Grundlagenuntersuchungen dar und fokussieren sich hauptsächlich auf die Netzwerkdomeäne, bzw. auf einen Teilbereich davon, zum Beispiel Medienzugriffsverfahren oder Routing Protokolle. Die beiden anderen Subsysteme werden entweder stark vereinfacht dargestellt oder vollständig vernachlässigt, da sie keinen oder nur geringen Einfluss auf den Unter-

suchungsgegenstand der Teildomeäne Netzwerk haben.

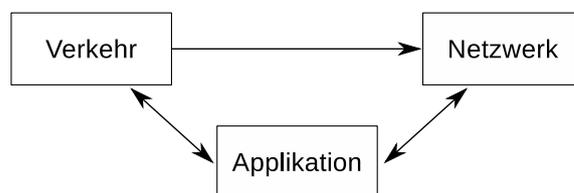


Abbildung 1: Domänen und Wechselwirkungen des Gesamtsystems Car2X-Kommunikation

Um reale Implementierungen von FAS, die auf Car2X basieren, in einer simulierten virtuellen Umgebung zu evaluieren, müssen alle drei Teilsysteme in hoher Realitätstreue abgebildet werden. Ein naiver Ansatz wäre der Einsatz existierender hochdetaillierter Sim-ulationsmodelle in allen Domänen für eine Vielzahl von Fahrzeugen. Die hohe Modellkomplexität und der damit verbundene Rechenaufwand machen diese Art der Umsetzung zum jetzigen Zeitpunkt jedoch ressourcenaufwändig und unwirtschaftlich.

2 Lösungsansatz

Dieser Beitrag stellt einen Ansatz zur simulativen Evaluierung von Car2X-FAS anhand von Multi-Resolution-Modelling [1] vor, um einen Kompromiss zwischen Rechenaufwand und Genauigkeit zu erzielen. Im Gegensatz zu den etablierten Sim-ulationsansätzen liegt der Betrachtungsfokus nicht auf der Gesamtheit aller Fahrzeuge, sondern auf einem speziellen Fahrzeug mit dem eine virtuelle Erprobungsfahrt durchgeführt wird (vgl. Abbildung 2). Durch die hohe Reichweite der Funkverbindung

müssen alle Fahrzeuge, die das virtuelle Erprobungsfahrzeug umgeben, in der Simulation berücksichtigt werden. Jedoch kann auf Basis der Entfernung zum virtuellen Erprobungsfahrzeug die Relevanz der einzelnen Fahrzeuge bewertet werden. Verkehrsteilnehmer in der unmittelbaren Umgebung stellen beispielsweise in Bezug auf die Sicherheit ein höheres Risiko dar. Basierend auf diesem Abstandskriterium wird der Simulationsraum dynamisch in zwei Teilgebiete aufgeteilt. In dem zentriert um das virtuelle Erprobungsfahrzeug angeordneten Bereich werden hochaufgelöste Modelle aller Teildomänen eingesetzt um eine realistische Abbildung des Gesamtsystems zu erreichen. Der restliche Simulationsraum, der typischerweise deutlich größer ist, wird durch weniger genaue Modelle repräsentiert.

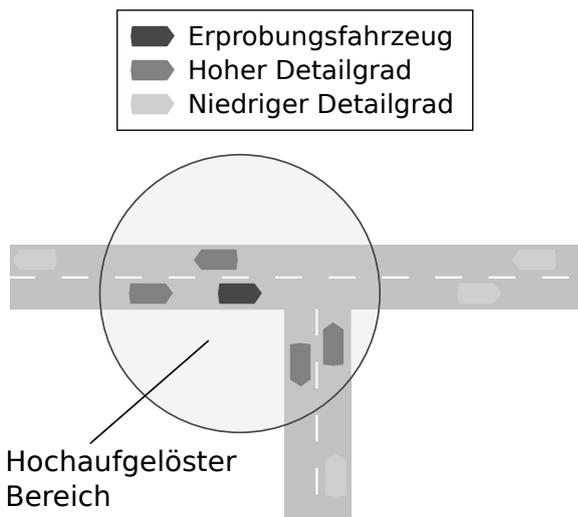


Abbildung 2: Dynamische Bereiche der Multi-Resolution Simulation

Das Anwendungsverhalten wird im hochaufgelösten Bereich mit Hilfe virtueller Maschinen realisiert, der Funkkanal auf Basis von Raytracing modelliert und die physikalischen Eigenschaften der Verkehrsteilnehmer anhand eines nanoskopischen Verkehrs- und Fahrzeugsimulators abgebildet. Für den vereinfacht modellierten Bereich werden die drei Teildomänen ebenfalls simuliert, jedoch in reduziertem Detailgrad.

References

- [1] P. K. Davis and R. Hillestad. Families of models that cross levels of resolution: Issues for design, calibration and management. In *Proceedings of the 25th conference on Winter simulation*, pages 1003–1012. ACM, 1993.
- [2] M. Killat. *The Impact of Inter-Vehicle Communication on Vehicular Traffic*. PhD thesis, Karlsruher Institut für Technologie, 2009.

Potential of Dynamically Adaptable Simulation Models for Virtual Commissioning

Philipp Puntel Schmidt¹, Fabian Köslin², Alexander Fay³

^{1,3} Helmut-Schmidt-University, Institute of Automation, Holstenhofweg 85, D-22043 Hamburg

² Dassault Systèmes, Wartenau 11, D-22089 Hamburg

¹ + 49 (0) 40 / 6541 – 3860, *philipp.puntelschmidt@hsu-hh.de* (corresponding author)

² + 49 (0) 40 / 6094693 – 42, *fabian.koeslin@3ds.com*

³ + 49 (0) 40 / 6541 – 2719, *alexander.fay@hsu-hh.de*

Virtual commissioning (VC) is used to test control code deployed on Programmable Logical Controllers. Simulation models of a plant are the core of any VC approach. Simulation models should represent the plant in a way so that the correct process execution can be tested under customers' conditions. Simulation models of a plant are usually not built monolithically, but by many partial simulation models that represent the modules or components of the investigated plant. To ensure that the VC is efficient and provides helpful results, these partial simulation models can be implemented at different levels of detail, depending on the current test scenario. Usually, the definition of the modules' and components' level of detail is fixed. However, situations exist where more than one level of detail can be adequate. A dynamically adaptable level of detail seems beneficial to e. g. keep computing time at a reasonable level and to ensure meaningful results of the plants simulation model. However, no method or approach exists so far to handle a dynamically adaptable level of detail. This paper discusses potentials and challenges of a dynamically adaptable level of detail and shows possible solution contributions that could yield benefits for a VC approach.

1 Introduction: a virtual commissioning based view on plants

Simulation models that represent the investigated plant are the basis of any virtual commissioning (VC) approach. Usually, these simulation models represent specific elements of a plant like modules and components [1] and are built by many partial simulation models. From a mechatronic point of view [2], these partial simulation models simulate modules that can be seen as noteworthy elements of the overall system and be described as substantial function holders.

Modules realise specific processes or tasks within the plant and bundle their capabilities together to perform the plants greater purpose. Modules themselves are comprised of (hierarchically lower and granular finer) components like sensors, actors and others (Figure 1). Components as typical elements of a module perform - in cooperation and together with an appropriate controller - the specific process/task of the module.

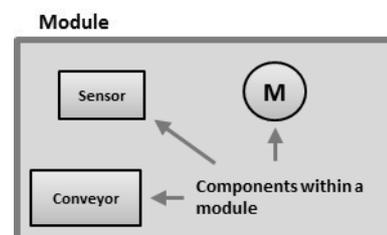


Figure 1: Components within a module

1.1 Model types and levels of detail for virtual commissioning

Simulation models of plants' modules and components must be sufficiently meaningful with regard to the respective test case [3]. Models can naturally only be abstractions and simplifications of reality and can therefore only show selected aspects of a system's behaviour. A systematic classification on what aspects should be simulated is beneficial for a VC approach. As defined in [1], four levels of detail (*LoD*) can be identified for a (discrete manufacturing) VC approach and can be described as follows:

- *Macroscopic LoD*: Rough granular perception of the respective plant; modules are rep-

represented by a single simulation model (without comprised components). Only time based behavior is considered, goods that are transported and/or processed are not considered.

- *Mesoscopic (not dynamical) LoD*: Fine granular perception of the plant; the module’s function is simulated by the interaction of simulation models of components that comprise the module. These components can be devices (like sensors or actors or frequency inverters) or mechanical/pneumatical/hydraulical systems (also called basic system according to [2]). Only time based behavior is considered, goods that are transported and/or processed are considered but movement is restricted to predefined paths (no free movement in space or collisions possible).
- *Mesoscopic (dynamical) LoD*: Fine granular perception of the plant as in the mesoscopic (not dynamical) LoD. Components show physical/dynamical behavior, where applicable. Goods that are transported and/or processed are considered but movement is restricted to predefined paths (collisions on these paths are possible, in contrast to mesoscopic (not dynamical) LoD).
- *Microscopic LoD*: Fine granular perception of the plant as in the mesoscopic (dynamical) LoD. Components as well as transported goods show physical/dynamical behavior. Movement of goods is possible in free space, including any type of collisions like interfering contours, etc.

Several model types (exemplary defined in [1], see also [4]) can be distributed over these four levels of detail and define the solution space used to build the respective plant’s simulation model. These model types differ in aspects like time-based and dynamical/physical behavior. The specified model types can be described as follows:

Dead time models (macroscopic LoD) are represented by a simple transformation of the control input to control signals, delayed only by a dead time. *Simple device and kinematic models (mesoscopic not dynamical LoD)* as well as *complex device and kinematic models (mesoscopic dynamical LoD)* describe a fine-granular modelling approach that considers the

plants’ components that comprise a module. *Kinetic models (microscopic LoD)* are similar to complex kinematic models, but movement in free space is possible.

Table 1 shows an overview of the defined levels of detail as well as several important aspects of the particular levels of detail that are important for a VC approach.

Aspect:	Macroscopic	Mesoscopic				Microscopic	
		dev.		mech.		dev.	mech.
		(nd)	(d)	(nd)	(d)		
Consideration of entire process mapping	✓	✓	✓	-	-	✓	-
Time-based behaviour	✓	✓	x	✓	x	x	x
Dynamical/physical behaviour (if applicable)	x	x	✓	x	✓	✓	✓
Failure states	✓	✓	✓	✓	✓	✓	✓
Consideration of processed good*:	x	-	-	✓	✓	-	✓
- Movement of goods on stated paths (= degrees of freedom)	x	-	-	✓**	✓	-	x
- Free movement of goods in space	x	-	-	x	x	-	✓
- Collision of goods possible	x	-	-	x	***	-	✓

dev: device, mech: mechanical /pneumatical/hydraulical basic system (see VDI2206 [2]), nd: not dynamical, d: dynamical
 * Good is synonym to bulk goods or piece goods
 ** Acceleration free movement
 *** Only possible in simulated degree of freedom

Table 1: Overview of defined levels of detail and several important aspects of VC.

1.2 Deficits when defining the level of detail for certain modules

As described before, each single module can be seen as a substantial function holder within a plant [1]. These function holders perform the customers’ defined processes.

Based on available engineering data and heuristics that consider the practical knowledge of the engineers involved, a situation-related (test-case specific) required level of detail can be assigned to each single module [5]. At the moment, this assignment is fix for the simulation run and cannot change. However, modules often not only perform one process, but many, depending on the character of the appropriate module. Situations can arise where modules can be modelled in multiple levels of detail, according to the appropriate situation and the process that is being executed. A module e. g. executes three different processes where each process requires a different

level of detail. An example will be given in the next section.

2 Identifying situations where an adaptable level of detail is appropriate

As a first step towards an adaptable level of detail, situations should be systematically identified where the approach is beneficial. An easy transport processes shall be given as example to demonstrate situations where an adaptable level of detail is appropriate.

2.1 Example

A work piece is to be transported over 2 conveyors (CV01, CV03) and a turntable with integrated conveying belt (TT02) there and back again (Process steps P1 to P6 in Figure 2).

All conveying belts consist of one motor; the turntable contains an extra motor for rotation movement. Conveying belt CV01 and CV03 have one sensor each (S1 placed at the beginning of CV01 and S6 placed at the end of CV03) for (accurate) position recognition. Turntable TT02 has 2 sensors (S2 and S3) on its conveying belt. The position of the (rotating) turntable is recognized by two mechanical switches (S4, S5). While sensors S2 and S3 can be described as standard capacitive sensors, sensors S1 and S6 allow for precise positioning (yellow shaded, laser distance sensors).

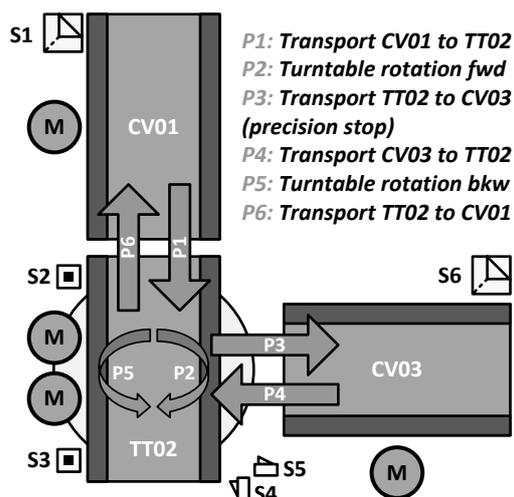


Figure 2: Example as described.

Performed processes are P1 to P6 (Figure 2, green arrows). P1, P2, P4 and P5 can be described as “standard” conveying processes or “technical cycles”

that do not have any special requirements e. g. regarding to positioning accuracy. Process P3 and P6 however need very accurate positioning ($\pm 0,1\text{mm}$) for further processing of the work piece (not part of this example model). Figure 3 shows an excerpt of the PLC program that exemplary shows execution of process steps P1 and P2.

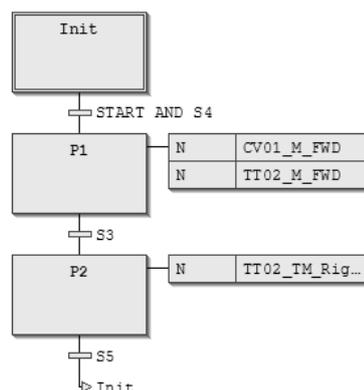


Figure 3: Exemplary PLC programm that shows execution of process steps P1 and P2.

Eventually, a situation-related required level of detail can be defined based on the processes. When performing process P1, P2, P4 and P5, no further requirements on the simulation emerge – the usage of a macroscopic level of detail (dead time models) seems to be adequate to test the programmed PLC-functions (see also Figure 3).

Following this idea, two levels of detail, depending on the process step, can be suitable for conveyors CV01 and CV03: When P1 and P4 are executed, a macroscopic level of detail (and the included dead-time-model) is sufficient. However, process P3 and P6 indicate that accurate positioning must be ensured. Parameters like ramps and physical effects like acceleration and friction become important elements of the module models in these process steps, as they influence the quality of the simulation results and, finally, the quality of the PLC-Code itself. The needed physical behavior can be simulated by a mesoscopic (dynamical) level of detail (complex device and kinematic models, see Chapter 1.1).

2.2 A systematic way to identify modules that can potentially switch regarding to their level of detail

In case several levels of details are possible, from all suitable levels of details often the highest and most complex one is chosen to ensure correct behavior of a module in any case. This can lead to situations where

a high and complex level of detail is useless for most of the simulation run since a low (curtailed in the meaning of functionality) level of detail would be sufficient for most of the time. Situations are possible where 1) the computing time increases to a level that prevents VC from running smoothly and 2) where effort on validation can increase to a level where the overall effort/benefit ratio of a simulation is compromised. An adaptable level of detail where simulation models can switch through different levels of detail (depending on the test case) can be promising. However, there is no known solution for this challenge at the moment. To describe the issue of switching simulation models (regarding to their level of detail) in a standardized and systematic way, the Domain-Mapping-Matrix (DMM) [6] can be used as a supporting tool. The DMM, represented by an $(n \times m)$ matrix, can combine two different domains (in this consideration *resources* and *processes*). Rows are represented by appropriate modules (representing the resources of the system investigated) while columns are filled with the process steps that are executed by the modules.

For each module which is involved in a particular process step, the cell at this intersection must be filled with a score. The overall quantification scheme of the score should be based on the amount of levels of detail available for modelling. If e. g. the already described four levels of detail (Chapter 1.1) are provided, the scores' values could be based on a range from 1 to 4, in reference to each available level of detail. The numbers present the overall needed level of detail (where 1 is defined as macroscopic and 4 as microscopic), ascertained by engineers. Figure 4 shows an example on how the interactions of modules and processes (based on the example in Chapter 2.1) could be weighted.

DMM		Process 1	Process 2	Process 3	Process 4	Process 5	Process 6
CV01		1					3
TT02		1	2	1	1	2	1
CV03		!	3	1			

Figure 4: Example of a Domain Mapping Matrix where processes are assigned to modules within a plant. Processes refer to process steps as defined in Chapter 2.1.

Attention should be paid to situations where a module has different scores (exemplarily shown by the red boxes in Figure 4). This can be identified by examining the modules associated row. As a conclusion, the DMM indicates in an easy way to identify situations where more than one level of detail is appropriate. As already described, the highest (and most complex) level of detail of all possible is often chosen for the simulation run to ensure correct behavior of a module in any case (*max()* function over the module's associated row), despite useless for e. g. CV03 when performing process P4 or TT02 performing Processes P1, P3, P4 and P6.

3 Potentials and challenges of an adaptable level of detail

A second step in introducing an adaptable level of detail should focus on possible implementation strategies regarding to the used simulation tool and overall simulation framework and procedure. The main benefit (lower computing time) of an adaptable level of detail should be in focus.

3.1 Possible implementation strategies

Simulation is always strongly depending on the used simulation tool as well as framework and the overall simulation approach like discrete event simulation or continuous simulation. The computing time of a simulation is influenced by multiple factors according to [7]: The type of the used numerical solver (for a detailed comparison of common solvers see [8]), amount of continuous state variables as well as the sheer quantity of events that must be handled by the solver [9]. If an adaptable level of detail is demanded, two very basic implementation strategies can be identified:

- a) Implementation in one single tool (e. g. in Modelica [10]): One tool provides all levels of detail in e. g. a (copious) library. The tool must be able to handle discrete event simulation as well as continuous simulation approaches. The tool must be able to swap (partial) simulation models during runtime. This is also called "integrated simulation" [7].
- b) Implementation in different tools according to tool specific abilities and strengths (e. g. Modelica [10] and Matlab/Simulink [11]): Several libraries in different modelling tools provide objects in a specific level of detail. Matlab/Simulink could e. g. be used for dead-time

models (since dead time models have a discrete event character) while a Modelica library can provide models where physical/dynamical behavior is needed. A middleware (like the Functional-Mockup-Interface (FMI) [12] or implementations with respect to the High-Level Architecture (HLA) [13] standard) is necessary to interconnect these (partial) simulations. The middleware must be able to swap (partial) simulation models during runtime. This principle is also called “seperated simulation” according to [7].

Both implementation strategies have their advantages and drawbacks: When implementing all levels of detail feasible in just one single simulation tool, it must be capable to handle both discrete (macroscopic, mesoscopic not dynamical) and continuous (mesoscopic dynamical, microscopic) behavior (“Hybrid-Modelling”, compare [7]). Additionally, the simulation tool must provide (numerical) solvers that are capable of handling not only discrete event simulation or continuous simulation, but both. Cardinality of the included simulation library is estimated to be very high, exacerbating servicing as well as model building effort (regardless if models are built by an automatic model approach [14], [15], or manually).

Segregation of discrete and continuous behavior in different tools (e. g. Matlab/Simulink for discrete event simulation, Modelica for continuous simulation) requires a middleware with standardized interfaces to ensure interoperability of the respective models. Liu and Frey [7] describe this as an intermediate communication and synchronization layer. The well known Functional Mockup Interface (FMI) [12] is one approach that follows this idea and has already found widespread acceptance in industrial applications.

FMI was introduced with the intention to deliver an interface to develop complex systems where different parts of the system can be modelled in different simulation tools. Simulation models are provided in form of a Functional Mockup Unit (FMU) (exported by the respective simulation tool) and are implemented in the standardized framework. However, FMI increases the complexity of the simulation approach to a level that could make an automatic generation approach according to [14], [15] unfeasible. Efforts on parameterization as well as adjustment of the FMI framework itself are considered to be rather high.

3.2 Influences on computing time when simulating in different levels of detail

Different levels of detail (precisely: different model types) can be indicative of different workloads needed to calculate the plant simulation model, being one of the main reasons an adaptable level of detail might be suitable to avoid unnecessary high computing time. A highly detailed simulation model indicates, due to the high amount of events to be calculated, that computing time may increase to a level that prevents VC from running “smoothly”. An example on how smoothness can be defined is provided by [16]: A fixed step of 1ms (hard real-time) of the simulation must be ensured so that the PROFINET communication from simulation model to the PLC is not corrupted within a VC test bench.

As already mentioned, the numbers of triggered events have a large impact on computing time of a simulation model. Each time an event is triggered (e. g. when a sensor triggers or control variables within the PLC change), the deployed solver restarts calculating the whole simulation model. Depending on the solver, these calculations can be iterative (called event-iterations) to ensure the specified simulation accuracy. This increases computing time even further. More complex, very detailed simulation models (that consider physical/dynamical behavior) have naturally a high amount of events and need potentially more computing time than an easy, time based simulation model which is based only on dead-time models (that should only have a low number of events). The defined models types (Chapter 1) should therefore not only differ in the ability to simulate time based and physical/dynamical behavior, but also mainly in the number of events they potentially generate. Furthermore, the used simulation models should be highly optimized to a specific solver (e. g. in [16], the Euler solver is used) to ensure a hard real-time at any time whenever needed.

3.3 Conclusions and thoughts on an adaptable level of detail

As a conclusion, the used simulation models, e. g. stored in a model library (see [17] for further details and an example), should be optimized in a way so they generate as less events as possible. Additionally, it must be ensured that the number of events grows with the respective level of detail and not vice versa: models at a macroscopic level of detail should generate (clearly) less events than models at a microscopic level of detail. This highly correlates with

the solver used and must be taken into consideration when building or optimizing a simulation library that contains models at different levels of details. The main benefit and the ultimate goal of an adaptable level of detail, is therefore mainly depending on the events generated and the solver used, not necessarily from the simulation principle itself (discrete event/continuous simulation). Additionally, the interoperability of the simulation models must be ensured at all time, not only when the level of detail is determined static, but especially when simulation models should change during runtime.

On the technical side, no possibility to change models during runtime exists, to the authors' best knowledge, so far, and this holds for both implementation possibilities (integrated/separated simulation). This can be seen as a major point that should be taken into consideration in future investigations.

4 Requirements for implementing an adaptable level of detail

As concluded in Chapter 3, the used simulation tool, simulation framework and number of events have a huge impact on the implementation of an adaptable level of detail. Several requirements can be identified that must be accomplished regardless of the used simulation environment. These requirements represent important factors that not only concentrate on the simulation model or the used framework, but also on the simulation (building and executing) process itself and shall be considered when an adaptable level of detail is to be implemented in future applications:

- It must be possible to generate the simulation model automatically: Information potentially necessary for a dynamical switching should not prevent an automatic generation approach (e. g. described by [14] or [15]).
- The level of detail may only be switched considering modules: Only complete modules (including all comprised components) may be switched through different levels of detail, not the components comprising a module (the goal is to keep complexity of the simulation model as well as intricacy of the switching process itself on a reasonable and manageable level).
- Interoperability must be ensured: Interfaces between partial simulation models should always

be compatible, even when these models are switched between several levels of detail.

- Computing time shall not increase while “switching” the simulation models: Effort on “switching” simulation models to different levels of detail should be kept to a minimum regarding to computing time to ensure real time ability.
- “Switching” simulation models within a process step must be prevented: A specific level of detail assigned to a module must be assigned fix until the process is terminated. Switching levels of detail is therefore not allowed within a process, but between two subsequent processes.
- It must be clearly identifiable what process is being executed: In case a module is able to execute several processes, the simulation tool/framework must be capable to identify the specific process.
- Models used for VC should always be optimized to generate the minimum possible number of events: The model library (an example can be found in [17]) must be optimized and it must be ensured that simulation models in a low level of detail create significant fewer events than models at a high level of detail. This can mean that models should be optimized to run with a specific solver in a specific simulation environment.

5 Conclusion

The paper presents considerations regarding an adaptable level of detail when running a simulation with the purpose of virtual commissioning. Furthermore, an outlook on how an adaptable level of detail might be implemented according to simulation library and simulation framework was given. While different levels of detail have already been implemented in several test cases [1], at the moment no known simulation tool as well as simulation framework is capable of switching simulation models during runtime of the simulation. Future work on the topic should be focused on two aspects: Firstly, a refined method to identify situations where an adaptable level of detail is suitable (based on the method introduced in this paper) and secondly a technical solution that enables swapping simulation models at different levels of detail.

6 References

- [1] P. Puntel-Schmidt, A. Fay: Levels of Detail and Appropriate Model Types for Virtual Commissioning in Manufacturing Engineering. In: Proceedings MATHMOD 15 – Vienna Conference on Mathematical Modelling, Vienna, 17.-20. February 2015.
- [2] VDI2206: Design methodology for mechatronic systems. VDI-Verlag, Düsseldorf, 2004.
- [3] Haberfellner, R., de Weck, O., Fricke, E., Vössner, S.: Systems Engineering. Grundlagen und Anwendungen. Orell Füssli Verlag AG, Zürich, 2012.
- [4] VDI3693: Virtuelle Inbetriebnahme. VDI-Richtlinien-Entwurf. VDI-Verlag, Düsseldorf, 2015.
- [5] P. Puntel-Schmidt, A. Fay: Applying the Domain-Mapping-Matrix to Identify the Appropriate Level of Detail of Simulation Models for Virtual Commissioning. In: 2nd IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control Schedule (CESCIT), Maribor, Slovenia, 2015.
- [6] Pimpler, Thomas U.; Eppinger, Steven D. (1994): Integration analysis of product decompositions. Cambridge, Mass: Alfred P. Sloan School of Management, Massachusetts Institute of Technology (Working paper / Alfred P. Sloan School of Management, WP # 3690-94-MS).
- [7] Liu, L.; Frey, G.: Efficient Simulation of Hybrid Control Systems in Modelica/Dymola. Proceedings of the 6th Vienna International Conference on Mathematical Modelling (MATHMOD 2009), pp. 1344-1352, Vienna, Austria, Feb. 2009.LIU12
- [8] Liu, L.; Felgner, F.; Frey, G.: Comparison of 4 Numerical Solvers for Stiff and Hybrid Systems Simulation. Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2010), ISBN 978-1-424-46849-2, 8p, Bilbao, Spain, Sept. 2010.
- [9] Liu, L.; Felgner, F.; Frey, G.: Modellierung und Simulation von Cyber-Physical Systems, Proceedings of the 12th Fachtagung Entwurf komplexer Automatisierungssysteme (EKA 2012), ISBN 978-3-940961-72-3, pp. 149-157, Magdeburg, Germany, May 2012.
- [10] Modelica Association: www.modelica.org, last downloaded: 05.05.2015.
- [11] MATLAB/Simulink v8.4. Natick, Massachusetts: The MathWorks Inc., 2014.
- [12] Modelisar, FMI for Model Exchange 1.0 Specification: https://svn.modelica.org/fmi/branches/public/specifications/FMI_for_ModelExchange_v1.0.pdf, 2010.
- [13] IEEE 1516: Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules. 2010.
- [14] P. Puntel-Schmidt, A. Fay, W. Riediger, T. Schulte, F. Köslin, S. Diehl: Validierung von Steuerungscode mit Hilfe automatisch generierter Simulationsmodelle. In: at – Automatisierungstechnik 2015; 63(2), S. 111–120.
- [15] Barth, Mike: Automatisch generierte Simulationsmodelle verfahrenstechnischer Anlagen für den Steuerungstest. Fortschritt-Berichte VDI Reihe 20 Nr. 438: Rechnergestützte Verfahren. Düsseldorf: VDI Verlag 2011.
- [16] W. Riediger, P. Puntel Schmidt, F. Köslin, T. Schulte: Hardware-in-the-Loop-Simulation fertigungstechnischer Anlagen. In: SPS/IPC/DRIVES 2014, Nürnberg, 25. -27. Nov. 2014.
- [17] F. Köslin, P. Puntel Schmidt, W. Riediger, A. Fay: Entwurf einer Modelica Simulationsbibliothek für die virtuelle Inbetriebnahme fertigungstechnischer Anlagen. In: 7th International Symposium on Automatic Control, AUTSYM 2014, 25.–26.09.2014, Wismar.

**2Simulate Modeling Language –
Wiederbelebung und Modernisierung von alter Simulationssoftware**

Jürgen Gotschlich, Umut Durak
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
Institut für Flugsystemtechnik
{ juergen.gotschlich, umut.durak }@dlr.de

In den vergangenen Jahrzehnten wurden zahlreiche Tools und Technologien zur Echtzeit Simulation entwickelt und angewendet. Beginnend mit der Analogrechner-Ära und der Programmierung von Patchfeldern, folgte eine Reihe von speziellen Hochleistungs-Multiprozessor-Rechnersystemen. In der 80er-Jahren waren die Computer der Fa. Applied Dynamics International (ADI), AD10 und AD100 herausragende Vertreter dieser Gattung. Die Programmiersprache ADSIM diente als Simulationssprache dieser Rechner. Die Sprache verfügt über Eigenschaften, die speziell zur Programmierung von dynamischen Systemen benötigt werden, wie der Simulationssteuerung, Integratoren, Funktionen zur linearen Interpolation, etc. Einige Programme sind bis heute im Einsatz. Heutzutage werden jedoch bevorzugt standardisierte Betriebssysteme und COTS-Hardware benutzt um entweder Modell-basierte Design-Praktiken anzuwenden oder General-Purpose Sprachen wie C++. Auf der anderen Seite bleibt das Problem, alten Programm-Code zu modernisieren und evtl. wiederzuverwenden.

Das Institut für Flugsystemtechnik des Deutschen Zentrums für Luft- und Raumfahrt (DLR) in Braunschweig hat eine lange Tradition und Erfahrung beim Aufbau und Betrieb von InFlight- und Flugsimulatoren. Die Motivation für diesen Beitrag ist begründet durch den Wunsch die Modernisierung alten Simulationscodes zu vereinfachen. Dieser Ansatz betrifft die ADSIM Programme zur Modellierung des ausserdienstgestellten Flugversuchsträgers „Advanced Technologies Testing Aircraft System“ (ATTAS) sowie des Flugversuchsträgers Flying Helicopter Simulator (FHS). Diese Programme waren bzw. sind im Einsatz in den Bodensimulatoren der jeweiligen Großanlagen.

2Simulate ist die Simulations-Infrastruktur des „Air Vehicle Simulator“ (AVES) des DLR Instituts für Flugsystemtechnik. Dieses Werkzeug ermöglicht eine einfache Anbindung von Simulations-Hardware und eine Integration der erforderlichen Software-Komponenten unter Berücksichtigung von Echtzeit-Anforderungen in verteilten Simulationen. Es stellt eine generische Funktionalität zur Verfügung, und wird in einer Reihe von Simulatoren angewandt. Die Präsentation zeigt eine Modellingsprache mit einer Reihe von elementaren Funktionen zur Simulation von Dynamischen Systemen. Hiermit wird die manuelle Portierung von alten ADSIM Modellen nach C++ wesentlich vereinfacht. Es wird aufgezeigt, wie eine einfache Integration der portierten Modelle in die 2Simulate-Infrastruktur möglich ist. Dies wird anhand von 2 Beispielen belegt.

Sitzung 8

Raum 111

Engineering Anwendungen

(Sitzungsleitung: Leander Marquardt)

*Kopplung von Prozess- und CFD-Simulation zur Untersuchung des Wärmeverlusts beim
Flüssigaluminiumtransport*

Jörg Schmidt

Hochschule Ansbach

Simulation eines Kontaktschalters mit COMSOL Multiphysics

Sonja Weinbrecht

COMSOL

Spritzgussimulation mit SOLIDWORKS Plastics

Tobias Kamp

Solidline

Vorausberechnung des Betriebs- und Emissionsverhaltens von kleinen luftgekühlten

Industrie-Dieselmotoren

Leander Marquardt

Fachhochschule Stralsund

Kopplung von Prozess- und CFD-Simulation zur Untersuchung des Wärmeverlusts beim Flüssigaluminiumtransport

Jörg Schmidt¹, Wolfgang Schlüter¹, Ansgar Ringleb¹

¹Hochschule Ansbach

jo.schmidt@hs-ansbach.de

Die Studie konzentriert sich auf die Simulation von Wärmeverlusten beim Flüssigaluminiumtransport. Dabei wird aus Gründen der Energieeffizienz Aluminium in einem Schmelzbetrieb geschmolzen und das 900 °C heiße Aluminium in großen Behältern mit LKWs zum weiterverarbeitenden Druckgussbetrieb transportiert, wobei aufgrund nachgeschalteter Prozesse ein bestimmtes Temperaturfenster eingehalten werden muss. Dazu ist es wichtig, die Endtemperatur bei einer Transportfahrt zu prognostizieren und die Einflussfaktoren zu kennen, die diese beeinflussen.

Um dies zu erreichen, wird ein dynamisches Simulationsmodell, bestehend aus einer Kopplung von CFD- und Prozesssimulation, entwickelt. Während die dynamische Prozesssimulation den gesamten Transportvorgang abbildet, dient die CFD-Simulation zur Bestimmung von Wärmeübergangskoeffizienten. Die Simulationen werden mit Messungen von realen Transportfahrten validiert. Der Einfluss von verschiedenen Faktoren, wie Position des Behälters auf dem LKW, die Fahrtgeschwindigkeit und die Füllmasse der Behälter, auf das Abkühlverhalten, wird quantifiziert. Schwachstellen in der Behälterisolierung können ermittelt werden. Es werden verschiedene Detaillierungsgrade der CFD-Simulation auf ihre Eignung für die Kopplung mit der dynamischen Prozesssimulation untersucht.

Das hier vorgestellte Konzept ist auf weitere konvektive Wärmeübertragungsprozesse übertragbar, bei denen zeitlich variierende physikalische Parameter nur über Strömungsanalysen zu bestimmen sind.

1 Einleitung

Zu einem der größten Endenergieverbraucher in Deutschland zählt der Industriesektor (29 % am Gesamtenergieverbrauch [1]), bei welchem die energieintensiven Schmelzprozesse in Gießereien eine bedeutende Rolle spielen. Aktuelle Bedingungen, wie die CO₂-Problematik, steigende Energiepreise und Effizienzförderung durch Bund und Länder [2], sorgen für einen stetigen Bedarf, die vorliegenden Energieeinsparungspotentiale auszuschöpfen.

In einer Studie des Bayerischen Landesamts für Umweltschutz (LfU) [3] konnte aufgezeigt werden, dass eine Anlieferung von flüssigem Aluminium (sog. Flüssigaluminiumtransport), energetisch günstiger ist, als das Aufschmelzen von Festmaterial in der Gießerei (siehe Abbildung 1). Entscheidend ist der Transportweg des flüssigen Aluminiums von der Hütte zur Gießerei, da das Flüssigaluminium in einem bestimmten Temperaturfenster angeliefert werden muss. Beim Flüssigaluminiumtransport ist der Energieaufwand zur Behältervorwärmung bereits enthalten.

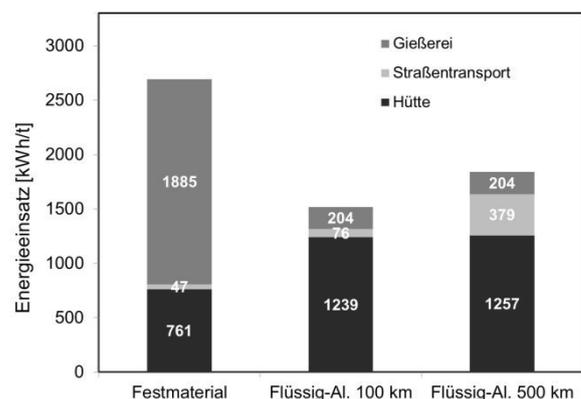


Abbildung 1. Vergleichende Betrachtung des Flüssigmetalltransports mit Anlieferung von Festmaterial

Im Rahmen des Forschungsverbundes energieeffizienter Technologien und Anwendungen (FORETA) wurde das Abkühlverhalten während einer Transportfahrt mittels Simulation betrachtet [4]. Das wesentliche Ziel der Simulation ist eine Darstellung des Temperaturverlaufs während der Fahrt und die Vorhersage

der Flüssigaluminiumtemperatur am Zielort, unter Berücksichtigung von Umwelteinflüssen (im folgenden Kapitel 2 näher erläutert). Das dynamische Simulationsmodell selbst besteht aus einer „Kopplung“ von CFD- und Prozesssimulation: Während die dynamische Prozesssimulation die Werte einer Fahrt ermittelt, wird mithilfe der CFD-Simulation der Wärmeübertragungskoeffizient ermittelt und in die Prozesssimulation integriert (Verfahren wird in Kapitel 3 erläutert). Die Simulation wurde zudem mit Messwerten von diversen Fahrten validiert.

In dieser Arbeit soll geklärt werden, welche Genauigkeit die CFD-Simulation für ein optimales Aufwand/Nutzen-Verhältnis besitzen muss.

2 Flüssigaluminiumtransport

Das Flüssigaluminium wird in der Hütte in vorgewärmte Transportbehälter (Tiegel) abgefüllt, auf LKWs verladen und gelangt von dort in den drei 5-Tonnen-fassenden Behältern zur Gießerei (vgl. Abbildung 2).



Abbildung 2. LKW mit Tiegeln auf dem Auflieger

Das Aluminium wird in der Hütte überhitzt (je nach Fahrtstrecke Temperaturen zwischen 800 °C und 975 °C), um trotz Abkühlung während des Transportes die Temperaturvorgabe am Zielort zu erfüllen. Folgende Einflussfaktoren wirken sich auf den Wärmeverlust und damit auch die Abkühlung des Flüssigaluminiums aus:

- Fahrdauer
- Fahrtgeschwindigkeit
- Außentemperatur
- Anfangstemperatur flüssiges Aluminium
- Flüssigaluminiummenge im Behälter
- Geometrie des Behälters
- Position des Behälters auf dem LKW.

Die Tiegel selbst sind aus einer Reihe von Isolationsschichten aufgebaut, um die Wärmeverluste möglichst gering zu halten. Der Aufbau der Schichten

unterscheidet sich je nach Behältersegment, wie Boden oder Zylinderwand und setzt sich aus Materialien wie Calcium-Silikat-Platten, Stahl und Beton zusammen. Für neuwertige Behälter mit funktionierender Isolationsschicht gibt es Richtwerte für die Abkühlung des Flüssigaluminiums von 8 °C/h (Oberflächentemperatur Behälter unter 70 °C). Gebrauchte Behälter weisen einen höheren Abkühlungswert von ca. 12 °C/h bis 20 °C/h auf (Oberflächentemperatur Behälter rund 100 °C, lokal oft höher); in [5] wird dieser gar mit 20 °C/h bis 40 °C/h angegeben.

Neben Energieeffizienz liegen die Vorteile des Flüssigaluminiumtransports u. a. in der optimierten Wertschöpfungskette (Just-in-Time-Lieferung, Lageraufwand entfällt) und einer erhöhten Materialeffizienz, da erneutes Einschmelzen zu Materialverlusten führen würde. [6]

3 Simulationsmodell

Zur Simulation des Abkühlverhaltens wurde ein dynamisches Modell entwickelt, das die Geometrie des Tiegels (CAD-Modell) berücksichtigt (siehe Abbildung 3). Ein wesentliches Problem der dynamischen Prozesssimulation ist die Bestimmung des Wärmeübertragungskoeffizienten α [7]. Analytisch ist dieser Wert nur mit Vereinfachungen zu ermitteln, bspw. über dimensionslose Kennzahlen wie der Nusselt-Zahl (Ähnlichkeitstheorie) [8]. Eine geeignete Möglichkeit, den α -Wert über einen bestimmten Zeitraum zu berechnen, stellt eine CFD-Simulation dar. Der Vorteil liegt in der Berechnung mithilfe der Erhaltungsgleichungen für Masse, Energie und Impuls [9].

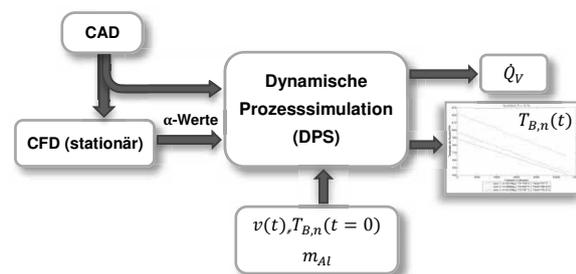


Abbildung 3. Schema dynamisches Simulationsmodell

Theoretisch wäre der Transportvorgang mit einer CFD-Simulation abbildbar. Problematisch ist jedoch der hohe Zeitaufwand, da ein instationäres Berechnungsmodell eingesetzt wird, in der ein gekoppeltes Wärme- und Strömungsmodell über einen langen Zeitraum (Fahrzeit) berechnet wird. Sinnvoller ist es

daher, die mit CFD berechneten α -Werte in die Prozesssimulation zu integrieren. Zusätzlich werden die Flüssigaluminiumtemperaturen $T_{B,n}(t=0)$ der Behälter n ($n = 1 \dots 3$) vor Fahrtbeginn, sowie die Fahrtgeschwindigkeit $v(t)$ und die Flüssigaluminiummasse m_{Al} in der Prozesssimulation vorgegeben. Als Ergebnis liefert die Simulation den Temperaturverlauf vom Flüssigaluminium in den jeweiligen Behältern $T_{B,n}(t)$, sowie den Gesamtwärmeverlust \dot{Q}_V nach Ende der Fahrzeit.

4 Dynamische Prozesssimulation

Die Simulation des Abkühlverhaltens erfolgt mit der Software Matlab[®]. Das physikalische Modell beruht dabei auf einer Energiebilanz. Da die Temperaturdifferenz ΔT genügend hoch ($\Delta T \approx 850 \text{ °C}$), die Biot-Zahl Bi genügend klein ($Bi \ll 1$) und die Fourier-Zahl Fo genügend groß ($Fo \gg 1$) sind, kann die Temperaturverteilung im Flüssigaluminiumbehälter als konstant angenommen werden [10]. Die Biot-Zahl beschreibt die zeitliche Entwicklung der Temperaturverteilung in einem Körper, bei plötzlicher Änderung der Umgebungstemperatur, während die Fourier-Zahl die Zeit ausdrückt, in der sich eine Temperaturinhomogenität ausbreitet. Abhängig vom behälterspezifischen Wärmeverlust $\dot{Q}_{V,n}$ und unter Berücksichtigung der Masse m_{Al} und der spezifischen Wärmekapazität $c_{p,Al}$ im Behälter, ergibt sich für die Temperaturänderung des Flüssigaluminiums in einem Behälter folgende Differentialgleichung:

$$\frac{dT_{B,n}}{dt} = -\frac{\dot{Q}_{V,n}}{m_{Al} \cdot c_{p,Al}} \quad (1)$$

Die Berechnung des Wärmeverlustes vom Transportbehälter muss aufgrund des Behälteraufbaus in separaten Bereichen erfolgen. So wird der Transportbehälter nach geometrischen Bedingungen aufgeteilt, da Segmente wie der Boden und die zylindrische Behälterwand unterschiedliche Isolationsschichten und -materialien aufweisen (vgl. Kapitel 2). Nach der Aufteilung besteht der Transportbehälter aus i Segmenten ($i = 1 \dots 5$), welchen jeweils unterschiedliche α -Werte zugeordnet werden (siehe Abbildung 4).

Der Gesamtwärmeverlust $\dot{Q}_{V,n}$ je Behälter ergibt sich nach (2) aus der Summe der Wärmeverluste pro Segment $\dot{Q}_{V,Segment\ i}$.

$$\dot{Q}_{V,n} = \sum \dot{Q}_{V,Segment\ i} \quad (2)$$

Der Wärmeverlust $\dot{Q}_{V,Segment\ i}$ pro Segment setzt sich nach (3) zusammen aus dem Quotient des Produkts der Oberfläche A und der Temperaturdifferenz zwischen der Temperatur des Flüssigaluminiums im Behälter $T_{B,n}$ und der Umgebungstemperatur T_u , sowie der Summe aus dem Wärmeleitwiderstand $R_{ges,i}$ des Bauteils und dem Wärmeübergangswiderstand $1/\alpha_{außen,i}$ der Behälteroberfläche. Der Wärmeübertragungskoeffizient im Behälterinneren ($\alpha_{innen,i}$) wird in diesem Fall vernachlässigt, da keine bzw. nur geringe Konvektion stattfindet.

$$\dot{Q}_{V,Segment\ i} = \frac{A \cdot (T_{B,n} - T_u)}{R_{ges,i} + \frac{1}{\alpha_{außen,i}}} \quad (3)$$

Der Faktor $R_{ges,i}$ berücksichtigt die geometrischen Bedingungen und die Wärmeleitfähigkeit der jeweiligen Isolationen. Die Gleichung macht deutlich, dass bei höherem α -Wert der Anteil an der Summe im Nenner geringer wird und dadurch der Einfluss der Konvektion auf die Wärmeverluste sinkt und die Wärmeleitung an Bedeutung gewinnt. Unter folgenden Rahmenbedingungen werden die Temperaturänderung und Wärmeverluste in der dynamischen Prozesssimulation untersucht:

- unterschiedliche Fahrzeiten bei Tag und bei Nacht (geringere Dauer)
- unterschiedliche Fahrtgeschwindigkeiten des LKWs während Fahrzeit (18 km/h, 36 km/h, 54 km/h, 72 km/h und 90 km/h)
- unterschiedliche Anfangstemperaturen des Flüssigaluminiums.

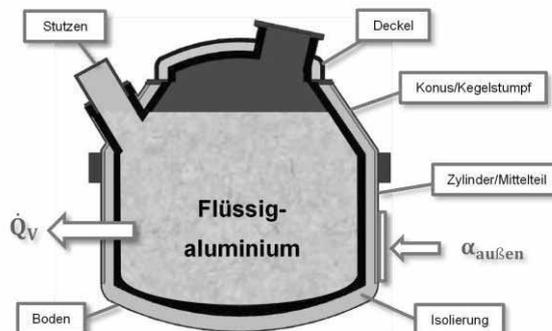


Abbildung 4. Flüssigaluminiumbehälter mit Segmenten und Isolierung im Querschnitt

Abbildung 5 zeigt den Wärmeverlust der jeweiligen Behältersegmente, bei einer Fahrtzeit von 4,3 h und 5 t Flüssigaluminium, mit einer Anfangstemperatur von 800 °C. Die Betrachtung ist bezogen auf den dritten Behälter, am Ende des Transporters.

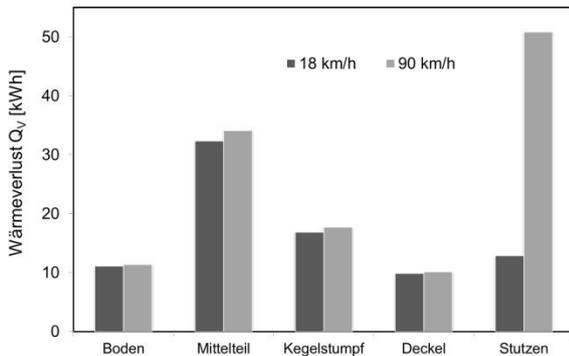


Abbildung 5. Vergleich von simulierten Wärmeverlusten der Behältersegmente bei unterschiedlichen Fahrtgeschwindigkeiten

Während sich die Wärmeverluste bei den ersten vier Segmenten, bei steigender Geschwindigkeit, nur leicht erhöhen ist beim Stutzen ein enormer Anstieg zu erkennen und macht bei der höheren Fahrgeschwindigkeit rund 41 % des Gesamtverlusts aus. Weiterhin auffällig ist, dass beim Stutzen ein hoher Wärmeverlust im Verhältnis der Größe zur Gesamtoberfläche vorliegt (nur 1,6 % Anteil an der Gesamtoberfläche). Hier hat der α -Wert (und damit auch der konvektive Wärmeverlust) einen höheren Einfluss, da der Gesamtwiderstand R_{ges} , aufgrund der kleineren Fläche, verhältnismäßig gering ist (vgl. (3)).

5 CFD-Simulation

Die CFD-Simulation (Star-CCM+[®]) setzt sich aus einem CAD-Geometriemodell des LKWs mit den Tiegeln zusammen. Um die Wärmeübertragung der Tiegel zu simulieren, wird auf deren Oberfläche eine Temperatur von 75 °C festgelegt. Der Strömungsraum um den LKW wird mithilfe eines Fernfeldes gebildet und mit einem polyedrischen Netz diskretisiert, ebenso wie die Oberfläche des LKWs und der Tiegel. Zusätzlich werden prismatische Schichten um den LKW angelegt, um die hohen Strömungsgradienten in Wandnähe annähernd exakt berechnen zu können. Als Turbulenzmodell wird ein k - ϵ -Modell eingesetzt. Die Geometrie des LKWs und der Tiegel selbst werden in vier verschiedenen Generationen (Entwicklungsstufen) der CFD-Simulation untersucht, wobei die Realitätsnähe der Geometrie mit zunehmender

Generation steigt. Die Entwicklungsstufen werden im Folgenden erläutert.

In Generation 1 wird eine simple Geometrie des Fahrerhauses und der Tiegel verwendet (siehe Abbildung 6). Die Transportbehälter sind in dieser Generation nicht fest montiert, sondern „schweben“ in der Luft. Es existieren zwei Varianten der Generation: Eine mit und eine ohne Stutzen.



Abbildung 6. CAD-Geometrie Generation 1

Abbildung 7 zeigt den LKW der Generation 2, welcher um den Auflieger für die Tiegel erweitert wurde. Weiterhin wurde das Fahrerhaus leicht angepasst. Die Tiegel selbst sind noch nicht fest auf dem Auflieger montiert. Wie auch in Generation 1 existiert eine Variante mit und eine ohne Stutzen.

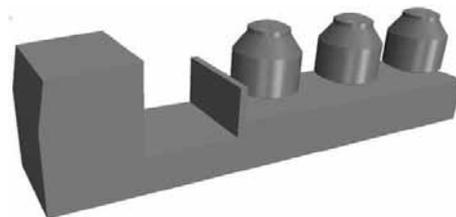


Abbildung 7. CAD-Geometrie Generation 2

Generation 3 beinhaltet eine Anpassung des Aufliegers und Fahrerhauses bzgl. des Bodens (u. a. Reifenstruktur). Auch hier „schweben“ die Tiegel in der Luft (vgl. Abbildung 8). In dieser Abbildung sind die Stutzen zu erkennen, welche in Fahrtrichtung angebracht sind.

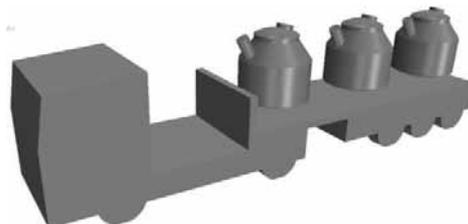


Abbildung 8. CAD-Geometrie Generation 3

In Generation 4 findet eine allgemeine Anpassung der Geometrien statt (siehe Abbildung 9):

- detailliertes Fahrerhaus mit Anbauten
- Reifenstruktur mit Radkappen
- detaillierterer Auflieger
- Tiegel um Anbauten erweitert und mittels Füßen fest am Auflieger montiert
- Stützen um 180° gedreht (nun entgegen der Fahrtrichtung ausgerichtet)

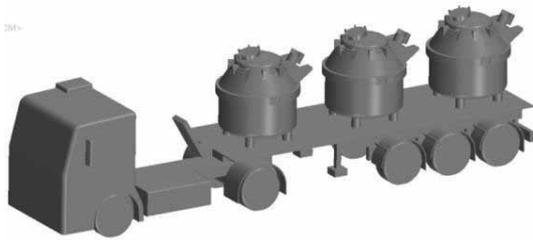


Abbildung 9. CAD-Geometrie Generation 4

In Generation 1 und 2 wird die Vernetzung um den LKW und die Tiegel lokal verfeinert (*Volumetric Control*). In Generation 3 und 4 findet eine andere Art von Netzverfeinerung statt: Hier wird die Vernetzung direkt an den Randgebieten (*Boundaries*) der jeweiligen Bauteile im Rechengebiet vorgenommen, d. h. bei den Netzbedingungen des jeweiligen Randgebietes wird eine spezifische Oberflächengröße (*Custom Surface Size*) vorgegeben. Abbildung 10 zeigt die Vernetzung des LKWs der Generation 3 mit den Tiegeln und dessen Umgebung (Fernfeld).

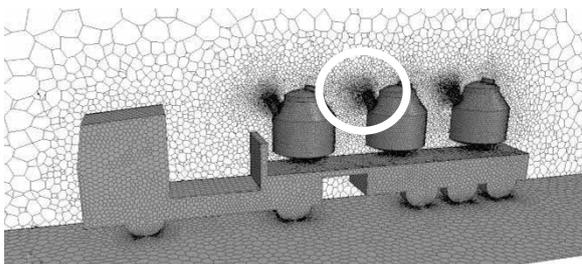


Abbildung 10. Vernetzung des Rechengebietes von Generation 3; feinere Vernetzung an Bauteiloberfläche (siehe Kreis)

Zur Darstellung des dreidimensionalen Strömungsfeldes eignen sich in diesem Fall Vektoren und insbesondere Stromlinienpfade. Auffallend sind dabei vor allem geringe Strömungsgeschwindigkeiten im Bereich der Tiegel und eine verstärkte Bildung von

Totgebieten, hervorgerufen durch die Ablösung der Strömung am Fahrerhaus. Abbildung 11 zeigt die

Strömungsgeschwindigkeit (*Velocity Magnitude*) um den LKW, anhand von Stromlinien und den α -Wert (*Heat Transfer Coefficient*) an der Tiegeloberfläche.

Es ist außerdem zu erkennen, dass die Strömung mit höherer Geschwindigkeit in der Mitte des Deckels und des Kegelstumpfs von Tiegel 1 auftrifft, während dies bei den nachfolgenden Tiegeln eher den äußeren Bereich des Deckels und des Kegelstumpfs betrifft. In diesen Bereichen sind auch die α -Werte entsprechend größer. Während bei der Prozesssimulation die α -Werte in den Segmenten gemittelt werden (da nicht konstant über Fahrzeit), sieht man in der CFD-Simulation deutlich, dass die Verteilung keineswegs homogen ist.

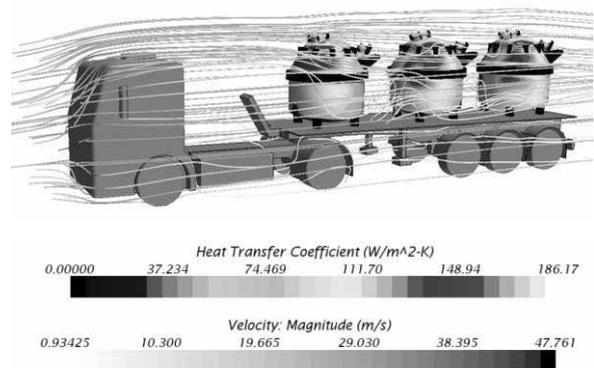


Abbildung 11. CFD-Simulation von Generation 4; Darstellung des Wärmeübertragungskoeffizienten an der Behälteroberfläche, Stromlinien stellen den Geschwindigkeitsverlauf dar

Faktoren, welche die Komplexität der Simulation und damit auch die Dauer der Simulation beeinflussen, sind die

- Anzahl der Zellen (in Verbindung mit der Geometrie des LKWs und der Tiegel)
- Kopplung von Strömungs- und Energiegleichung
- URF-Faktoren
- reale Fahrzeit (bei instationärer Simulation)
- und Zeitschritt (bei instationärer Simulation).

Ein wichtiges Vergleichskriterium ist eine Aussage bezüglich der Zeit, welche die Simulation benötigt, um die Berechnung unter den oben genannten Kriterien durchzuführen. In Kapitel 7 findet eine Betrachtung hinsichtlich der Simulationsdauer von verschiedenen Generationen statt. Nachfolgend sind die Ergebnisse der CFD-Simulation dargestellt.

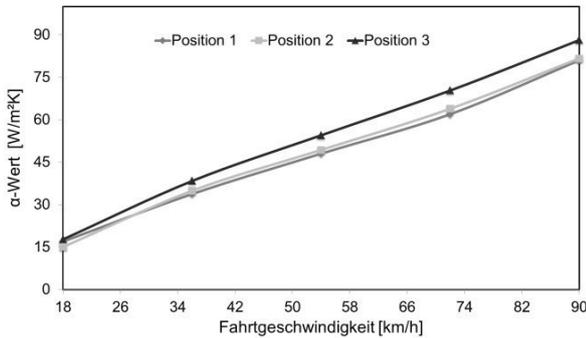


Abbildung 12. Vergleich der α -Werte an den verschiedenen Behälterpositionen in Abhängigkeit der Fahrtgeschwindigkeit

Abbildung 12 zeigt den Verlauf der α -Werte in Abhängigkeit der Fahrtgeschwindigkeit. Allgemein ist ein annähernd linearer Anstieg der α -Werte für steigende Geschwindigkeiten zu erkennen. Die Behälterpositionen auf dem Auflieger sind vom Fahrerhaus aus mit 1 bis 3 durchnummeriert (Position 3 am Ende des Aufliegers). An Position 3 liegen stets die höchsten Werte für α vor, da die Strömung mit hoher Geschwindigkeit auftrifft. An Position 1 und 2 dagegen ist die auftreffende Strömungsgeschwindigkeit geringer, da diese sich am Fahrerhaus ablöst.

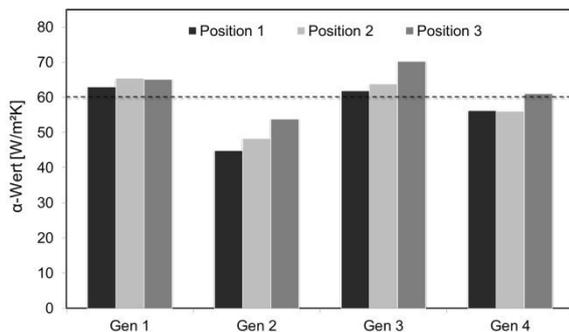


Abbildung 13. Vergleich von α -Werten der unterschiedlichen Generationen (Gen) an den verschiedenen Behälterpositionen

In Abbildung 13 ist eine Übersicht von α -Werten der verschiedenen Generationen dargestellt. Die gestrichelte Gerade deutet in dieser Abbildung den Mittelwert von α über alle Generationen an. Wie auch in der vorangegangenen Abbildung, sieht man hier den Einfluss der verschiedenen Behälterpositionen. Deutliche Unterschiede zeigt Generation 2 auf. Hier liegt der Unterschied zur Generation 3 bei rund 30 %. Die Werte von Generation 2 weichen ebenso deutlich von

der Standardabweichung σ ($\sigma = 6,5$) ab. Betrachtet man die Geometrie der 2. Generation (vgl. Abbildung 7), werden die Unterschiede deutlich: Hier ist die Geometrie des Aufliegers noch zu undetailliert. In Generation 4 ändert sich zudem die Verschattung durch die senkrechte Wand am vorderen Teil des Aufliegers.

6 Validierung des Simulationsmodelles

Durch die Arbeiten von Vatterodt [11] und Reint [12], in der das Abkühlverhalten der Behälter während einer realen Fahrt gemessen wurde, können die Simulationsmodelle validiert werden. Die Messwerte werden auch verwendet, um die Aluminiumtemperatur direkt nach der Einfüllung, die Masse des eingefüllten Aluminiums, die Fahrzeit und die Durchschnittsgeschwindigkeit vorgeben zu können. Aus den Messdaten geht hervor, dass der Tiegel am Fahrerhaus in der Regel die geringste Flüssigaluminiummenge enthält, an Behälterposition 3 am wenigsten Wärme übertragen wird und die durchschnittliche Fahrtgeschwindigkeit bei rund 65 km/h liegt. Die Werte werden über 38 Fahrten mit insgesamt 104 transportierten Tiegeln gemittelt.

Um eine Aussage bzgl. der Genauigkeit der simulierten α -Werte von Generation 1 bis 4 treffen zu können, werden die berechneten, bzw. simulierten Werte mit den Messwerten aus der Arbeit von Reint verglichen. Die Genauigkeit x_G spiegelt die relative Abweichung der Temperaturverläufe $|\Delta T_{mess} - \Delta T_{sim}|$, von gemessenem und simuliertem Temperaturverlauf, zum gemessenen Temperaturverlauf ΔT_{mess} dar (siehe Gleichung 4). Die hochgestellten Indizes deuten auf den Endwert e nach der Fahrt, bzw. den Anfangswert a vor Fahrtbeginn hin.

$$x_G = 1 - \frac{|\Delta T_{mess}^e - \Delta T_{sim}^e|}{\Delta T_{mess}^e} \quad (4)$$

Da die Anfangswerte der Simulation dessen der Messung entsprechen (T_{mess}^a), kürzt sich der Zähler entsprechend zu $|T_{mess}^e - T_{sim}^e|$ (vgl. Gleichung 5).

$$x_G \equiv 1 - \frac{|T_{mess}^e - T_{sim}^e|}{T_{mess}^a - T_{mess}^e} \quad (5)$$

Besonders gravierend sind die unterschiedlichen Abweichungen zwischen Generationen mit und denen ohne Stützen (vgl. Abbildung 14). Dies macht den hohen Einfluss des Stützens auf den Wärmeverlust

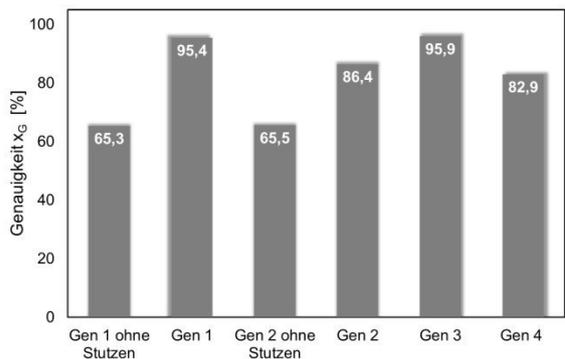


Abbildung 14. Genauigkeit der verschiedenen CFD-Generationen; berechnet über relative Abweichung

und damit α -Wert der Tiegel deutlich, da der Widerstand deutlich geringer ist als bei den restlichen Segmenten. Die Unterschiede zwischen den Generationen 1 bis 4 mit Stutzen fallen dagegen deutlich geringer aus. Generation 1 und 3 liefern dennoch die kleinsten Abweichungen und damit beste Genauigkeit. Ein Grund für den Unterschied zwischen Generation 3 und 4 ist, dass sich der Stutzen bei Generation 3 in Fahrtrichtung befindet, in Generation 4 entgegen der Fahrtrichtung (um 180° gedreht). Die zusätzlichen Tigelanbauten in Generation 4 beeinflussen ebenso das Ergebnis. Generation 1 überrascht bei diesen Ergebnissen mit hoher Genauigkeit der Werte. Die Schwankungen der einzelnen Temperaturabweichungen sind hierbei aber höher als in Generation 3 und daher vorsichtiger zu bewerten. Eine allgemeingültige Voraussage der Genauigkeit ist aufgrund des unstetigen Zusammenhangs zwischen Generation (Detailierungsgrad der CAD-Geometrie) und Genauigkeit nicht möglich.

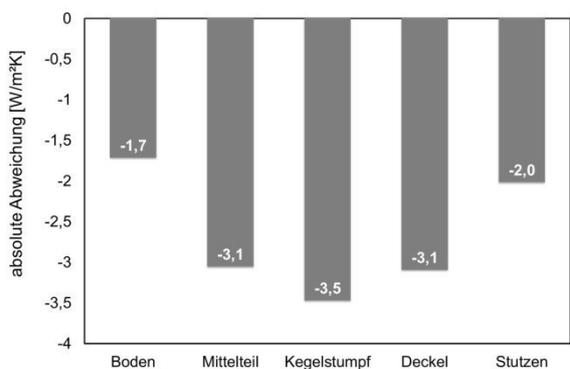


Abbildung 15. Abweichung zwischen simulierten α -Werten mit CFD und interpolierten Werten (positionsgemittelte Werte von Generation 4 bei 72 km/h)

Weiterhin wurde untersucht, wie sich die Interpolation der unstetigen α -Werte in Matlab, auf die Genauigkeit der Simulation auswirkt. Beim Vergleich dieser interpolierten Werte mit denen der CFD-Simulation verbleibt ein Differenzbetrag, da die Interpolierten einen geringeren Betrag aufweisen (vgl. Abbildung 15). So führt dies zu einer Abweichung der Aluminiumtemperatur - je nach Position - um bis zu 3,5 °W/m²K. Geht man davon aus, dass sich das Aluminium während einer Fahrt mit 72 km/h um rund 120 °C abkühlt und die Differenz 3,1 °W/m²K beträgt, führt dies zu einer Abweichung von 2,6 %. Je kleiner die Fahrtgeschwindigkeit ist, desto geringer ist aber der Fehler.

7 Simulationsdauer

Die Ergebnisse zeigen den zeitlichen Aufwand der CFD-Simulationen, auch in Verbindung mit der Genauigkeit der Simulationen. Es wird eine Aussage bezüglich Aufwand (Simulationsdauer) und Nutzen (Genauigkeit der Simulation) getroffen. Die Simulationen wurden mit einer i7-CPU (2,8 GHz) und einem Arbeitsspeicher von 8 GB durchgeführt. Zudem wurde die Courant-Zahl auf 5 (stationär) und 50 (instationär), sowie die Maximum Inner Iterations mit einem Wert von 5 und die Zahl der Iterationen mit einem Wert von 500 eingestellt.

Generation	Anzahl der Zellen in Mio	Simulationsdauer [h]	Zeit/Zelle [ms]
Gen1 ohne Stutzen	0,47	1,9	1,51
Gen1	0,51	1,9	1,40
Gen2 ohne Stutzen	0,44	1,6	1,34
Gen2	0,53	2,0	1,38
Gen3	0,40	1,2	1,09
Gen4	1,89	6,3	1,19

Tabelle 1. Simulationsdauer von Generationen der stationären Simulation

Tabelle 1 stellt den Vergleich der verstrichenen Zeit für die Simulationen der verschiedenen Generationen dar, welche jeweils mit einer CPU-Recheneinheit berechnet werden (daher keine parallele Berechnung). Aufgrund der geringsten Zahl an Zellen liegt für

Generation 3 die niedrigste Simulationszeit vor. Theoretisch müsste bei allen Generationen der gleiche Wert für die benötigte Zeit einer Zelle (Zeit/Zelle) vorliegen. Hier spielen aber weitere Parameter eine Rolle, wie bspw. der Wert der Courant Number (vgl. Kapitel 6). So ergeben sich bis zu 40 % Unterschiede beim Verhältnis Zeit/Zelle. Den größten Einfluss hat hier allerdings die Zahl der Zellen.

Abbildung 16 zeigt den Vergleich der Simulationszeiten als Diagramm. Hier sieht man, wie sich die Simulationszeit der Generation 4 deutlich von denen der anderen abhebt (ca. dreifacher Aufwand). Generation 3 besitzt die geringste Simulationsdauer.

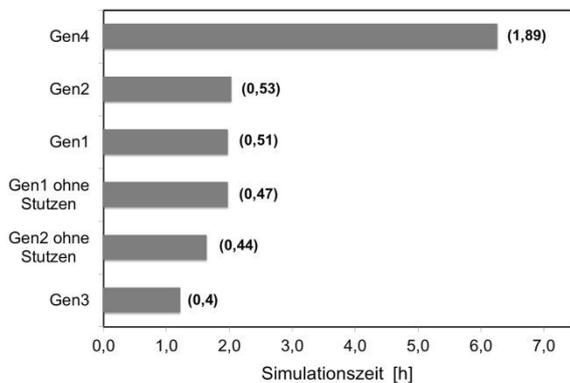


Abbildung 16. Vergleich der Simulationsdauer für 500 Iterationen (stationäre Simulation); Zahlen in Klammern entsprechen der jeweiligen Zellenanzahl

Um zu zeigen, dass es nicht sinnvoll ist, den kompletten Transportvorgang mit CFD-Simulation abzubilden, wird eine vergleichende Berechnung mit Generation 1 ohne Stützen durchgeführt, bei verschiedenen Zeitschritten (*Time-Steps*). Die Ergebnisse sind Tabelle 2 zu entnehmen. Die physikalische Zeit entspricht der realen Zeit, z. B. entsprechen 100 s physikalischer Zeit 100 s Fahrtzeit des LKWs. Je höher der Zeitschritt, desto geringer die gesamte Simulationsdauer. Der Zeitschritt, bei der die genauesten Ergebnisse zu erwarten sind, konnte ermittelt werden (0,01s).

Wenn man den besten Zeitschritt ansetzt (0,01s), würde die instationäre Simulation einer 6-stündigen Fahrt ca. 4,5 Jahre benötigen (bei den gegebenen Einstellungen). Zum Vergleich: Die dynamische Prozesssimulation benötigt für den Berechnungsvorgang weniger als eine Sekunde. Mit einem parallelen Rechenvorgang (weitere CPU-Einheiten zuschalten) könnte man die Simulationsdauer entsprechend verkürzen. Die durch das dynamische Simulationsmodell verkürzte Simulationsdauer ist besonders vorteilhaft

bei der Durchführung von zeitaufwendigen Parameterstudien.

Zeitschritt [s]	phys. Zeit [s]	Sim.-dauer [h]	Sim.dauer für 6-h-Fahrt [a]
0,001	0,1	2,8	68,7
0,01	1	1,8	4,5
0,1	10	5,9	1,5
1,0	100	25,5	0,6

Tabelle 2. Simulationsdauer von Generationen der stationären Simulation (Gen1, ohne Stützen)

8 Zusammenfassung

In der vorliegenden Studie wurde ein gekoppeltes Simulationsmodell, bestehend aus CFD- und Prozesssimulation erfolgreich eingesetzt, um den Wärmeverlust beim Flüssigaluminiumtransport simulieren zu können. Es konnte aufgezeigt werden, dass eine Verbesserung der Ergebnisse hinsichtlich Geometrie und Vernetzung, auf Kosten erhöhter Simulationszeit, nur bis zu einem bestimmten Grad rentabel ist. So liefert Generation 4 trotz genauerer Geometrie und feinerer Vernetzung keine verbesserten Ergebnisse, bei gleichzeitig stark gesteigener Simulationszeit. Die Genauigkeit der Generationen ohne Stützen ist unzureichend, die von Generation 1 und 3 dagegen am besten. Generation 3 liefert dank verbesserter Vernetzung die geringste Simulationszeit und stellt insgesamt die beste Wahl dar. Berücksichtigt man aber den Aufwand zur Erstellung der Geometrie, sind auch die Ergebnisse der Generation 1 zufriedenstellend. Durch Anwendung des dynamischen Modellansatzes konnte allgemein die Simulationsdauer um ein Vielfaches reduziert werden. Dies ist besonders vorteilhaft für das Durchführen von Parameterstudien.

Eine Schwierigkeit stellt die Simulation der realen Gegebenheiten dar, da Messungen gezeigt haben, dass die verschiedenen Tiegel nicht die gleiche Menge an Aluminium und unterschiedliche Starttemperaturen enthalten.

Durch eine Aufteilung des Behälters in Segmente konnte das Segment mit dem größten Wärmeverlust aufgezeigt werden (Stützen).

Das Prinzip des dynamischen Simulationsmodells ist auf ähnliche Strömungsprobleme in der konvektiven Wärmeübertragung anwendbar, in der physikalische

Werte über einen langen Zeitraum berechnet werden müssen. Ein Beispiel hierfür ist die Simulation eines Aluminium-Schachtschmelzofens: Die Basis bildet eine Prozesssimulation des Schmelzofens; mittels CFD-Simulation kann die Erwärmung des Aluminiums näher betrachtet werden. Zusätzlich sind Messungen notwendig, um die Simulation zu validieren und diese an reale Gegebenheiten anzupassen.

9 References

- [1] Arbeitsgemeinschaft Energiebilanzen. *Auswertungstabellen zur Energiebilanz 1990 bis 2013*. Berlin, 2014.
- [2] Bundesministerium für Wirtschaft und Energie (BMWi). *Förderung von energieeffizienten und klimaschonenden Produktionsprozessen*. Berlin, 2014.
- [3] W. Stephan, et. al. *Effiziente Energieverwendung in der Industrie - Teilprojekt "Metallschmelzbetriebe" - Effiziente Energienutzung in Nicht-Eisen-Metall-Schmelzbetrieben*, Bayer. Landesamt für Umweltschutz, Augsburg, 2005.
- [4] A. Dentel, et. al. *Rationeller Energieeinsatz in der aluminiumverarbeitenden Industrie. Abwärmennutzung und verbesserte Isolierung von Transportbehältern für flüssiges Aluminium*. In: Ergebnisse des Forschungsverbundes Energieeffiziente Technologien und Anwendungen (FORETA), Herausgeber: W. Mayer, Attenkofer, Straubing, S. C1-C24, 2013.
- [5] K. Krone. *Aluminiumrecycling*. Aluminium-Verlag, Düsseldorf, 2000.
- [6] Herrmann et. al. *Energie- und ressourceneffiziente Produktion von Aluminiumdruckguss*. Springer, Berlin, 2013.
- [7] H. Pfeifer et. al. *Praxishandbuch Thermoproszessstechnik*. Vulkan-Verlag, Essen, 2010.
- [8] Verein Deutscher Ingenieure. *VDI-Wärmeatlas*. Springer, Berlin, 2006.
- [9] J. Ferziger und M. Peric. *Numerische Strömungsmechanik*. Springer, Berlin, 2008.
- [10] W. Polifke und J. Kopitz. *Wärmeübertragung*. Pearson Studium, München, 2009.
- [11] M. Vatterodt. *Untersuchung zur thermischen Isolierung von Transportbehälter für flüssiges Aluminium*. Diplomarbeit, Hochschule Ansbach, 2010.
- [12] F. Reinl. *Verifikation numerischer Strömungssimulationen zur Voraussage des Abkühlverhaltens von flüssigem Aluminium während des Transports*. Bachelorarbeit. Hochschule Ansbach, 2012.

Simulation eines Kontaktschalters mit COMSOL Multiphysics

Sonja Weinbrecht, Nancy Bannach und Dennis Cronbach

Comsol Multiphysics GmbH

dennis.cronbach@comsol.de

Kontaktschalter werden in elektronischen Systemen zur Kontrolle des maximalen Stromflusses und zum Schutz vor Überhitzung von Bauteilen eingesetzt. Um solche Schalter zu simulieren, muss neben der Strom- und Energiebilanzgleichung auch der mechanische Kontaktdruck berücksichtigt werden, da dieser die Strom- und Wärmeleitfähigkeit an der Kontaktfläche signifikant beeinflusst.

Es wird ein Modell vorgestellt, das dieser Tatsache Rechnung trägt. Dabei wird gezeigt, wie man mit der Simulationssoftware COMSOL Multiphysics® ein solches Modell aufbaut - von der Auswahl der relevanten physikalischen Gleichungen über die Definition der Kontaktflächen, die Vernetzung, bis hin zur Berechnung und Auswertung der Ergebnisse.

Wie dabei von der Möglichkeit von High Performance Computing profitiert werden kann, wird exemplarisch gezeigt.

Das Modell vereint alle relevanten physikalischen Effekte über ein- und zwei-Wege-Kopplung und dennoch kann diese komplexe Aufgabenstellung vergleichsweise einfach und schnell in COMSOL umgesetzt werden.

1 Einleitung

Kontaktschalter regulieren, ob ein elektrischer Strom von einer Stromquelle zu einem elektrischen Bauteil fließen soll oder nicht. Solche Schalter sind in zahlreichen Geräten zu finden und kontrollieren die abgegebene Leistung von einer Steckdose an ein angeschlossenes Gerät; die Ströme, die über eine Leiterplatte/Platine eines Computers transportiert werden, oder den Strom, der eine eingeschaltete Lampe zum Glühen bringt.

Wegen ihres verbreiteten Einsatzes ist die Simulation von Kontaktschaltern ein wichtiger Schritt beim Design elektronischer Komponenten und Anwendungen.

Die prinzipielle Funktion eines solchen Schalters ist einfach: Zwei leitfähige Metallstücke, zwischen denen eine Potentialdifferenz herrscht, werden miteinander verbunden und es fließt Strom. Wird die Verbindung unterbrochen, stoppt auch der Stromfluss. Der Stromfluss durch die Kontakte führt zur Erwärmung des Schalters und kann deren Material- und Oberflächeneigenschaften beeinflussen. Dieser Effekt muss bei der Modellierung eines solchen Schalters berücksichtigt werden, denn steigt die Temperatur zu stark an, kann es sogar zum Durchbrennen des

Schalters kommen. Die Stromtragfähigkeit des Schalters muss also ebenfalls analysiert werden.

Darüber hinaus können auch der mechanische Druck auf die Kontakte deren elektrische und thermische Eigenschaften lokal beeinflussen – ein Effekt, der bei einer quantitativen Simulation der Stromtragfähigkeit mit hohen Anforderungen an die Genauigkeit ebenfalls berücksichtigt werden muss.

Dieses Anwendungsbeispiel ist daher ein wahrhaft multiphysikalisches.

2 Das Modell:

2.1 Modellaufbau

Dieses Beispiel zeigt, wie ein multiphysikalischer Kontakt simuliert werden kann. Dabei wird das thermische und elektrische Verhalten der beiden Kontakte des Schalters simuliert. Bei geschlossenem Schalter findet der Stromfluss und der Wärmetransport nur über die Kontaktfläche des Schalters statt. Das Schalter-Bauelement ist aus Kupfer mit zwei zylindrischen Elementen, die an den eigentlichen Kontakten befestigt sind. Am Ende jedes Kontakts sind abgeflachte Haken, die den Kontakt sicherstellen. Dort werden

thermischer und elektrischer Scheinwiderstand mit dem mechanischen Kontaktdruck gekoppelt.

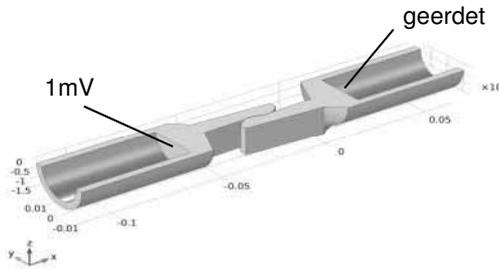


Abbildung 1: Geometrie des Kontaktschalters: Zwischen den beiden Kontakten wird eine Potentialdifferenz von 1 mV angenommen.

Die thermische und elektrische Kontaktleitfähigkeit hängen allein vom Kontaktdruck ab. Über die Schalteroberflächen wird Wärme durch natürliche Konvektion an die Umgebung abgegeben. In der Simulation wird dies mit Hilfe von Wärmeübergangskoeffizienten umgesetzt. Alternativ könnte hier die Strömung der Umgebungsluft ebenfalls simuliert werden.

Der Kontaktschalter ist symmetrisch, deshalb wurde nur eine Schalterhälfte simuliert, wie in Abbildung 1 dargestellt. Alle Ränder außer den Symmetrierändern sind frei beweglich und elektrisch isoliert.

Das Modell wird zunächst nach dem mechanischen Kontaktdruck gelöst. Basierend auf diesen Ergebnissen wird dann eine voll gekoppelte Simulation des Stromflusses und der Jouleschen Erwärmung durchgeführt.

2.2 Kontaktbedingungen für Stromfluss, Wärmetransport und Strukturmechanik

In diesem Beispiel soll besonders auf die Modellierung der Kontaktfläche eingegangen werden.

Voraussetzung dafür ist die Möglichkeit, die in Kontakt tretenden Ränder einander zuordnen zu können. Dazu werden in der Software COMSOL Multiphysics korrespondierende Ränder als sogenannte Paare definiert. Die entsprechenden Paarbedingungen werden dann je nach Position der Ränder zueinander aktiv.

Für die Analyse des strukturmechanischen Kontaktes stehen die „Augmented Lagrangian“-Methode und die „Penalty“-Methode zur Verfügung.

In diesem Beispiel kommt die genauere Augmented Lagrangian Methode zum Einsatz, bei der zusätzliche Freiheitsgrade für den Kontaktdruck und die

Komponenten der Reibungskraft eingeführt und mit gelöst werden.

Im Folgenden wird die Behandlung des thermischen und elektrischen Kontakts näher beschrieben. Auf mikroskopischem Level besitzen die Kontaktflächen eine gewisse Oberflächenrauigkeit und berühren sich daher nicht überall, sondern nur an gewissen Stellen. Die Oberflächenrauigkeit liegt in der Größenordnung von Mikrometern wohingegen die Kontaktfläche in der Größenordnung von Zentimetern liegt. Demnach ist eine Betrachtung auf mikroskopischer Ebene wenig effizient und eine Randbedingung, die mikroskopische Effekte berücksichtigt ohne dabei die Kontaktfläche explizit aufzulösen, ist notwendig. COMSOL bietet dafür spezielle Kontaktbedingungen.



Abbildung 2: Kontaktoberfläche auf mikroskopischer Ebene

Für den thermischen Kontakt wird eine gemeinsame Leitfähigkeit h der Kontaktfläche definiert. Die Wärmeströme an den beiden Kontaktflächen werden dann folgendermaßen bestimmt:

$$-\mathbf{n}_d \cdot (-k_d \nabla T_d) = -h(T_u - T_d) + rQ_b \quad (1)$$

$$-\mathbf{n}_u \cdot (-k_u \nabla T_u) = -h(T_d - T_u) + (1-r)Q_b \quad (2)$$

wobei u und d für die obere bzw. untere Kontaktfläche stehen.

Diese gemeinsame Leitfähigkeit h an der Kontaktfläche setzt sich folgendermaßen zusammen:

$$h = h_c + h_g + h_r \quad (3)$$

wobei h_c die Leitfähigkeit an den Kontaktstellen bezeichnet, h_g (g für „gap“) die Leitfähigkeit zwischen den Kontaktstellen und h_r die Leitfähigkeit durch Strahlung. Die Leitfähigkeit zwischen den Kontaktstellen h_g liefert dann einen signifikanten Beitrag, wenn das Medium zwischen den Kontakträndern eine nicht zu vernachlässigende Wärmeleitfähigkeit besitzt.

Da in diesem Beispiel die Leitfähigkeit von Kupfer deutlich höher ist als die der umgebenden Luft, kann h_g gegenüber h_c vernachlässigt werden, ebenso wie der Einfluss der Wärmestrahlung h_r .

Die mikroskopisch kleinen Unebenheiten lassen sich durch ihre durchschnittliche Höhe σ_{asp} beschreiben, sowie durch eine durchschnittliche Steigung m_{asd} (siehe Abbildung 2). Mit Hilfe der von Cooper, Mikic und Yovanovich beschriebenen Beziehungen ([1] und [2]) für isotrope raue Oberflächen läßt sich daraus h_c wie folgt berechnen:

$$h_c = 1.25 k_{contact} \frac{m_{asp}}{\sigma_{asp}} \left(\frac{p}{H_c} \right)^{0.95} \quad (4)$$

mit p als Kontaktdruck, H_c der Mikrohärtigkeit des weichen Materials und dem harmonischen Mittel der Leitfähigkeiten an den Kontaktoberflächen

$$k_{contact} = \frac{2k_u k_d}{k_u + k_d} \quad (5)$$

Fast analog wird der elektrische Kontakt behandelt. Auch hier wird die Stromdichte an den beiden Kontaktoberflächen bestimmt:

$$\begin{aligned} \mathbf{n} \cdot \mathbf{J}_1 &= -h_E (V_1 - V_2) \\ \mathbf{n} \cdot \mathbf{J}_2 &= -h_E (V_2 - V_1) \end{aligned} \quad (6)$$

wobei J_i die Stromdichte auf den beiden Kontaktoberflächen 1 und 2 darstellt, V_i das elektrische Potential und h_E die elektrische Leitfähigkeit.

Zur Berechnung der elektrischen Leitfähigkeit h_E wird wie beim Wärmetransport der Kontakt auf der Mikroskala betrachtet (siehe Abbildung 2), allerdings wird hier nur der Beitrag an den tatsächlichen Kontaktstellen berücksichtigt, die Lücken zwischen den Rändern wirken isolierend.

Die elektrische Leitfähigkeit h_E kann analog zur thermischen Leitfähigkeit h_c in Formel (4) an den Kontaktpunkten wieder über die Cooper-Mikic-Yovanovich-Beziehung beschrieben werden:

$$h_E = 1.25 \sigma_{contact} \frac{m_{asp}}{\sigma_{asp}} \left(\frac{p}{H_c} \right)^{0.95} \quad (7)$$

$\sigma_{contact}$ ist dabei das harmonische Mittel der elektrischen Leitfähigkeiten auf den beiden Kontaktoberflächen:

$$\sigma_{contact} = \frac{2\sigma_u \sigma_d}{\sigma_u + \sigma_d} \quad (8)$$

Die multiphysikalisch Kopplung von strukturmechanischer, elektrischer und thermischer Analyse geschieht also zum einen über den Kontaktdruck p , der bei der mechanischen Analyse berechnet wird, zum

anderen über einen im gesamten Modellgebiet aktiven Quellterm in der Wärmeleitungsgleichung, der sich aus Stromdichte und elektrischem Feld berechnet:

$$\begin{aligned} Q_e &= \mathbf{J} \cdot \mathbf{E} \\ \rho C_p \mathbf{u} \cdot \nabla T &= \nabla(k \nabla T) + Q_e \end{aligned} \quad (9)$$

Das Modell wird mit Tetraederelementen vernetzt, wobei die Kontaktoberflächen besonders fein aufgelöst werden müssen, um die Konvergenz des Algorithmus zur mechanischen Kontaktberechnung zu gewährleisten.

3 Ergebnisse

Bei geschlossenem Schalter führt der Stromfluss durch das Schalterelement zu einem Anstieg der Temperatur. Gleichzeitig wird über die Ränder des Schalters Wärme an die Umgebung abgegeben und es stellt sich mit der Zeit ein Temperaturgleichgewicht ein. Abbildung 3 zeigt die Temperaturverteilung auf der Oberfläche sowie die Stromdichte in Stromliniendarstellung für diesen Gleichgewichtszustand. Die Temperatur an der Kontaktfläche liegt dabei etwa 5 K über der Umgebungstemperatur.

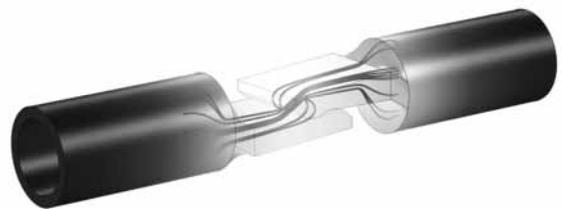


Abbildung 3: Temperaturverteilung und Stromdichte im Kontaktschalter: Warme Bereiche sind hell dargestellt, kalte Bereiche dunkel. Die Stromdichte wird mit Stromlinien dargestellt.

Führt man die Beobachtung auf mikroskopischer Ebene durch, kann man sehen, dass die Temperatur direkt an den Kontaktoberflächen gegenüber der Temperatur im sonstigen Bauteil leicht erhöht ist. Dies zeigt die detaillierte Darstellung einer Kontaktfläche in Abbildung 4. Gut zu erkennen ist hier auch die Drängung der Stromlinien, die wiederum die Stromdichte darstellen, an den Kontaktstellen.

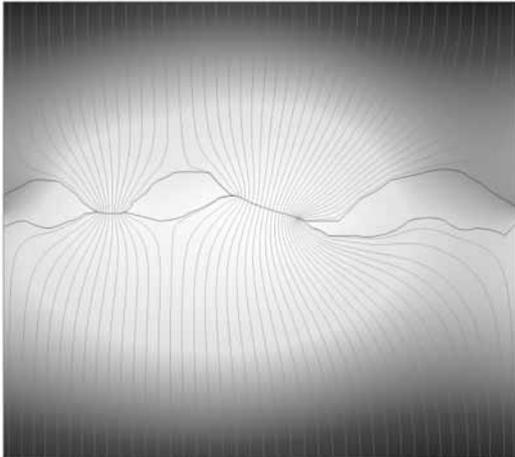


Abbildung 4: Detaillierte Ansicht einer Kontaktfläche: Die Stromliniendichte entspricht der Stromdichte, helle Bereiche sind Bereiche hoher Temperatur. Dieses Beispiel dient lediglich der Illustration.

4 Zusammenfassung und Ausblick auf HPC-Nutzung

Das Beispiel des Kontaktschalters demonstriert, wie ein Modell, in dem mechanische, elektrische und thermische Effekte auftreten, effizient aufgesetzt werden kann und welche Gleichungen dafür in der Software zur Verfügung stehen.

Multiphysikalische Modelle dieser Art werden umso rechenaufwändiger je mehr physikalische Effekte auftreten, denn für jeden dieser Effekte müssen zusätzliche Gleichungen gelöst werden. Steigt außerdem die Komplexität der Modell-Geometrie, so steigt der Rechenaufwand nochmals. Die Anforderung an eine Simulationssoftware ist neben der Fähigkeit, alle physikalischen Effekte in einem Modell vereinen zu können, also auch die Möglichkeit, von Multicore- und Clustersystemen zu profitieren. Darauf wird hier im letzten Abschnitt eingegangen.

COMSOL Multiphysics ist vollständig parallelisiert und profitiert sowohl von Shared- als auch Distributed Memory Systemen. Einfach zusammengefasst: Je mehr Speicher und Rechenleistung verfügbar ist, umso größer kann das Modell sein oder umso mehr Modelle gleicher Größe können in der gleichen Zeit berechnet werden.

Die Lösungsgeschwindigkeit eines einzelnen Modells steigt mit zunehmender Prozessoranzahl, jedoch nicht uneingeschränkt, da die Kommunikation zwischen den Prozessoren ebenfalls Zeit beansprucht. Die Ver-

teilung mehrerer einzelner Modelle auf Prozessoren erfährt diese Verlangsamung nicht.

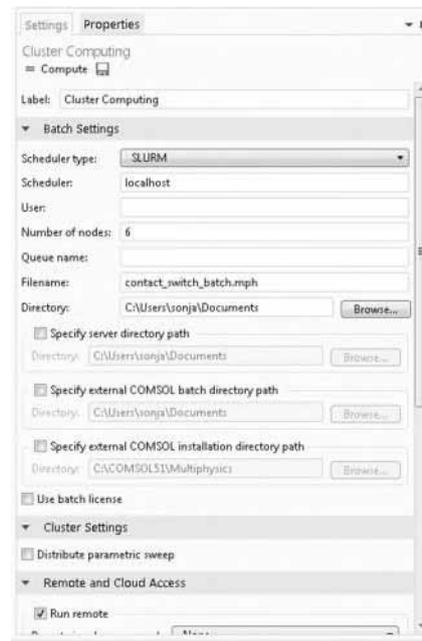
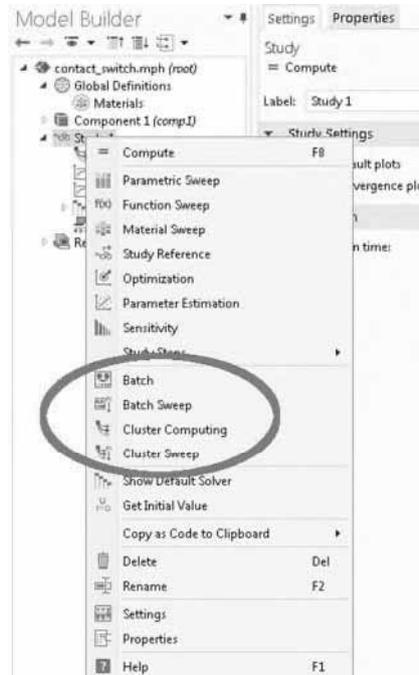


Abbildung 5: Auswahlmenü (oben) und Einstellungsmöglichkeiten (unten) für Clusterberechnungen in COMSOL Multiphysics.

Die Nutzung von Clustern oder auch das Starten von Batch-Jobs auf leistungsfähigen Rechnern ist dabei direkt aus der Software-Oberfläche möglich. Wie dies aussieht, zeigen das Auswahl-Menü und die Einstellungsmöglichkeiten für eine Cluster-Berechnung in Abbildung 5.

5 Referenzen

- [1] A. Bejan et al., *Heat Transfer Handbook*, John Wiley & Sons, 2003.
- [2] M.G. Cooper, B.B. Mikic, M.M. Yovanovich, Thermal contact conductance, *Int. J. Heat Mass Trans.* 12, S. 279–300, 1969.
- [3] COMSOL, *Heat Transfer Module User's Guide*. Handbuch zur Simulationssoftware COMSOL Multiphysics, Version 5.1, 2015.
- [4] COMSOL, *AC/DC Module User's Guide*. Handbuch zur Simulationssoftware COMSOL Multiphysics, Version 5.1, 2015.

Simulation des Spritzprozesses

SolidLine
Ein Unternehmen der Bechtle Gruppe



Tobias Kamp
Anwendungsberatung SolidWorks Plastics

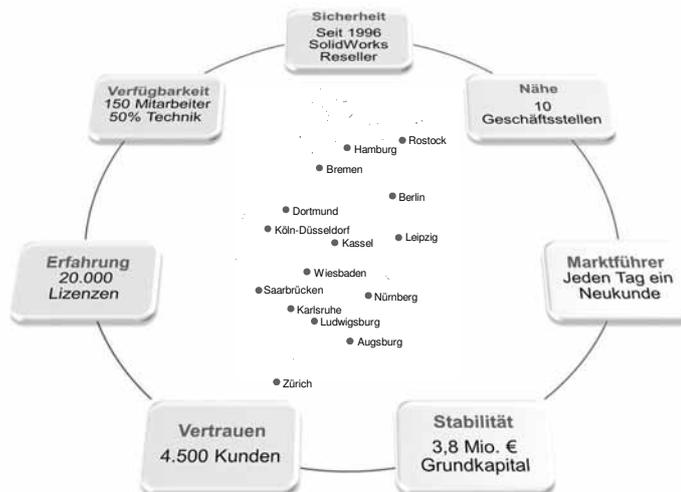


... freecall 0800 - 76 54 396

www.solidline.de

Ein starker Partner – SolidLine AG

SolidLine
Ein Unternehmen der Bechtle Gruppe



... freecall 0800 - 76 54 396

www.solidline.de

Optimierung Ihres Entwicklungsprozesses

SolidLine
Ein Unternehmen der Bechtle Gruppe



SOLIDWORKS



.. freecall 0800 - 76 54 396

www.solidline.de

SolidWorks Plastics

SolidLine
Ein Unternehmen der Bechtle Gruppe

Herausforderungen im Kunststoffspritzguss Prozess

- >80% der Kunststoffbauteile werden im Spritzgiessverfahren hergestellt
- Spritzgiessprozess ist ein komplexer Mix aus Zeit, Temperatur, Druck, Material & Werkzeugeinstellungen
- Vorhersage & Vermeidung Fertigungsfehlern so früh, wie möglich
- Teure Werkzeugnacharbeit bei Fertigungsfehlern
- Bauteilqualität
- Lange Entwicklungs- und Produktionszeiten durch anfallende Werkzeugnacharbeit



SOLIDWORKS



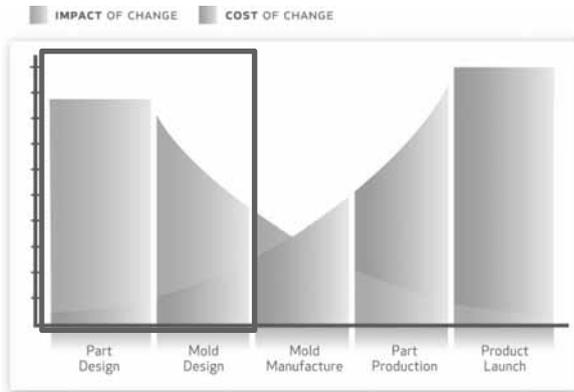
.. freecall 0800 - 76 54 396

www.solidline.de

SolidWorks Plastics

Mehrwert durch Einsatz von SolidWorks Plastics

SolidLine
Ein Unternehmen der Bechtle Gruppe



Großer Einfluss bei geringen Kosten während der Konstruktion!

DESIGN TO MANUFACTURING PROCESS



.. freecall 0800 - 76 54 396

www.solidline.de

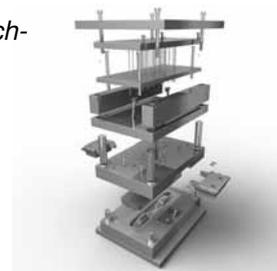
SolidWorks Plastics

Mehrwert durch Einsatz von SolidWorks Plastics

SolidLine
Ein Unternehmen der Bechtle Gruppe

- Für Hersteller von Kunststoffbauteilen oder Spritzgießwerkzeugen
- Simulation von verschiedenen Konfiguration, Anlaufpunkten und Materialien
- Werkzeugkosten >€10K - €1M+ und Werkzeugnacharbeit sehr zeit- und kostenaufwendig
- Optimierung der Oberflächenqualität

Einsatz von SW Plastics zur Vermeidung von nur 1x Werkzeugnacharbeit kann Positiven ROI bedeuten



.. freecall 0800 - 76 54 396

www.solidline.de

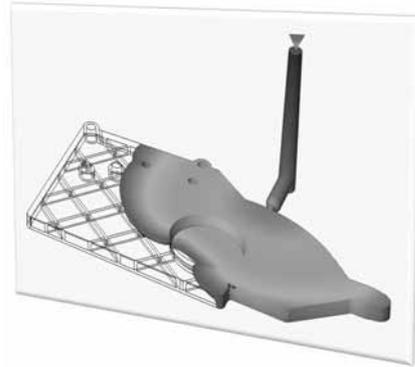
SolidWorks Plastics

Mehrwert durch Einsatz von SolidWorks Plastics

SolidLine
Ein Unternehmen der Bechtle Gruppe

Hauptvorteile von SolidWorks Plastics

- Voll in SolidWorks integriert
- Easy to use
- Umfangreiche Materialdatenbank mit mehr als 5000 Materialien
- Kosteneffizient
- Schnelle und genaue Simulation
- Hilft im Vertriebsprozess



.. freecall 0800 - 76 54 396

www.solidline.de

SolidWorks Plastics

SolidLine
Ein Unternehmen der Bechtle Gruppe

Simulation



Flow Simulation



Plastics



Sustainability



.. freecall 0800 - 76 54 396

www.solidline.de

SolidWorks Plastics

Mehrwert durch Einsatz von SolidWorks Plastics

SolidLine
Ein Unternehmen der Bechtle Gruppe

Plastics Standard			Plastics Professional				Premium
Oberflächen & Volumen Netz 	Füllanalyse & Short Shots 	Bindenähte 	Gasinnendruck & Angussventile 	Mehrfach-Werkzeuge 	Anguss-Balancierung 	Sandwich-Verfahren 	Verzugsanalyse 
Einfallstellen 	Luft-Einschlüsse 	Mehrere Anspritzpunkte 	Einleger & Überspritzung 	Schwindung 	Kühlzeit-Analyse 	Faseranalyse & Doppelbrechung 	Werkzeug Kühlung 

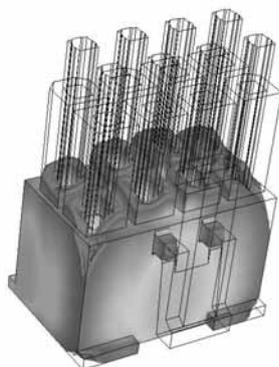
 **SOLIDWORKS**



.. freecall 0800 - 76 54 396

www.solidline.de

SolidLine
Ein Unternehmen der Bechtle Gruppe



*Vorstellung
SolidWorks Plastics*

 **SOLIDWORKS**



.. freecall 0800 - 76 54 396

www.solidline.de

SolidWorks Plastics

SolidLine
Ein Unternehmen der Bechtle Gruppe



Kunststoff-Konstruktion und Werkzeugauslegung

Aufgaben



... freecall 0800 - 76 54 396

www.solidline.de

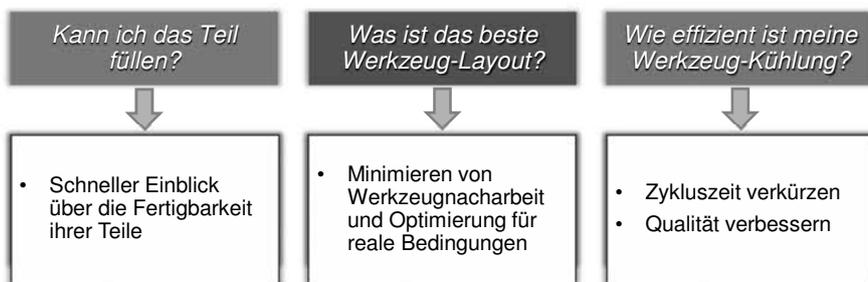
SolidWorks Plastics

SolidLine
Ein Unternehmen der Bechtle Gruppe



Kunststoff-Konstruktion und Werkzeugauslegung

Aufgaben



... freecall 0800 - 76 54 396

www.solidline.de

SolidWorks Plastics

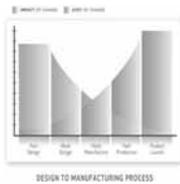
Mehrwert durch Einsatz von SolidWorks Plastics

SolidLine
Ein Unternehmen der Bechtle Gruppe

Zusammenfassung

Frühzeitige Anwendung von SolidWorks Plastics in der Kunststoffteil- & Werkzeugentwicklung hilft bei:

- Vorhersage und Vermeidung von Fertigungsfehlern
- Verbesserung der Bauteilqualität
- Reduzierung von teurer Werkzeugnacharbeit
- Verringerung von Time-to-Market



.. freecall 0800 - 76 54 396

www.solidline.de

SolidWorks Plastics

Mehrwert durch Einsatz von SolidWorks Plastics

SolidLine
Ein Unternehmen der Bechtle Gruppe

Referenzen

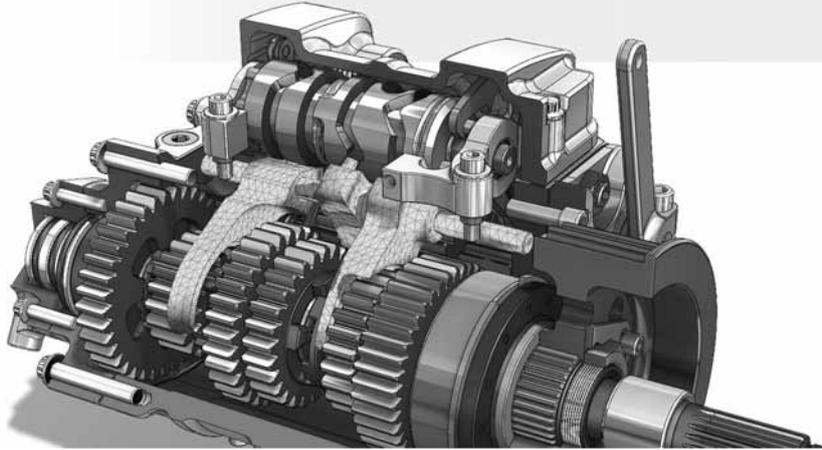


.. freecall 0800 - 76 54 396

www.solidline.de

SolidLine
Ein Unternehmen der Bechtle Gruppe

Vielen Dank!



 **SOLIDWORKS**



... freecall 0800 - 76 54 396

www.solidline.de

Vorausberechnung des Betriebs- und Emissionsverhaltens von kleinen luftgekühlten Industrie-Dieselmotoren

Leander Marquardt, Heiner-Joachim Katke, Andreas Reinke, Torsten Scheller

Fachhochschule Stralsund

Fachbereich Maschinenbau

leander.marquardt@fh-stralsund.de

Klein-Dieselmotoren bieten ein vielfältiges Anwendungsspektrum. Während in Regionen mit unterentwickelten Infrastrukturen häufig derartige Maschinen zur Wasserversorgung der Bevölkerung und der Landwirtschaft Anwendung finden, werden sie hierzulande vor allem dann eingesetzt, wenn notwendige Antriebsenergie preisgünstig über sehr lange Zeiträume unter verschärften Betriebsbedingungen zur Verfügung gestellt werden muss. Die Verbrennungsentwicklung erfolgt auch für solche Motoren in zunehmendem Maße auf theoretischem Wege. Mit einer geeigneten Prozess- und Emissionsrechnung lassen sich Voraussagen zum Betriebs- und Umweltverhalten unter veränderten Betriebsbedingungen treffen, so dass hiermit die für eine Entwicklung tatsächlich auszuführenden experimentellen Untersuchungen minimiert werden. Zeitgleich verlangen ebenso nachgeordnete konstruktive Entwicklungsarbeiten und Feldanalysen die korrekte Vorhersage des Druckverlaufes im Zylinder oder integraler Betriebswerte als Eingangsdaten. Prozessberechnungen für derartige Motoren sind aus der Literatur nicht bekannt. Gerade in den Teilgebieten Wärmefreisetzung, Brennverlaufsumrechnung und Stickoxidemission war mit erheblichen Problemen zu rechnen. Die modifizierten Berechnungsansätze sowie deren experimentelle Validierung sind Gegenstand dieses Beitrages.

1 Anlass der Untersuchungen

Klein-Dieselmotoren bewähren sich immer dann im Einsatz, wenn die zuverlässige Bereitstellung mechanischer Leistung über einen langen Zeitraum und unter widrigsten Bedingungen bezüglich der Aufstellung, Wartung und Pflege gefordert wird. Gerade für den Einsatz in den Bereichen Katastrophenschutz, Brandbekämpfung, Tiefbau sowie Trinkwasserversorgung und Landwirtschaft in Entwicklungsländern sind sie prädestiniert. Gefragt sind hier luftgekühlte Maschinen mit mechanischer Direkteinspritzung.

Die Verwendung empfindlicher elektronischer Hilfsysteme für Motorüberwachung, Einspritzung (Common Rail) oder Drehzahlregelung verbietet sich hierbei naturgemäß von selbst. Forderungen bezüglich Kraftstoffverbrauch und Leistungsdichte sind meist untergeordneter Natur.

Den Markt für derartige Maschinen teilen sich nur wenige Hersteller untereinander auf. Zu den bekanntesten zählen neben der Motorenfabrik HATZ ebenso FARYMANN Diesel als zweiter deutscher Anbieter sowie LOMBARDINI und GREAVES. Der Kreis potenzieller Interessenten für die im Folgenden beschriebenen Untersuchungen ist aus diesem Grund relativ klein. Untersuchungen an Maschinen dieser

Größe lassen sich zudem sehr preisgünstig ebenso auf rein experimentellem Wege durchführen.

Trotz der für diese Motorenklasse sehr preisgünstigen experimentellen Untersuchungen kann der verstärkte Einsatz von Simulationsmodellen die Entwicklungsarbeit beschleunigen, indem ausschließlich nur noch rechnerisch erfolversprechende Varianten tiefer gehenden Versuchen unterzogen werden. Gleichzeitig bieten geeignete Rechenmodelle die Möglichkeit einer schnellen Reaktion auf spezielle Anforderungen bei Kundenprojekten (z.B. Höhenaufstellung, besondere Abgasführungen, Leistungsanhebungen). Dieses Vorgehen ist bei Großdieselmotoren seit ca. 25 Jahren aus Kosten- und Zeitgründen zwingend. Zudem verlangen nachgeordnete Entwicklungsprozesse, wie z.B. die Auslegung der instationär belasteten Gleitlager des Kurbeltriebes oder Drehschwingungsanalysen die Bereitstellung von Zylinderdruckverläufen, ohne dass hierfür eine physisch existente Maschine zur Verfügung stehen würde.

Die Anpassung der bewährten Rechenmodelle an die Anwendung „Klein-Dieselmotor“ sowie ein quantifizierter Vergleich der berechneten Ergebnisse mit denen der Messung sind Gegenstand der hier beschriebenen Untersuchungen.

2 Zielsetzung und geplantes Vorgehen

Das grundlegende Ziel der beschriebenen Untersuchungen ist die Vorausberechnung des Betriebsverhaltens eines typischen Vertreters dieser Motorklasse. Ausgewählter Versuchsträger wurde der größte kommerziell verfügbare Einzylinder-Motor mit Luftkühlung. Abb. 1 zeigt ein Schnittbild des Motors.

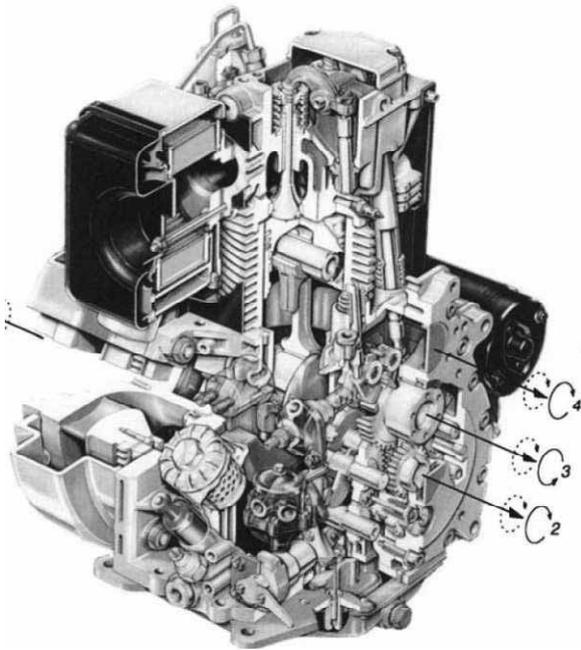


Abbildung 1. Schnittbild 1D90 [Werksbild HATZ]

Wie dem Schnittbild zu entnehmen ist, handelt es sich um einen handstartfähigen 4-Takt-Dieselmotor. Die direkte Einspritzung des Dieselöls in den Zylinder erfolgt durch ein mechanisches Pumpe-Leitung-Düse-System. Bei einem Zylinderhubvolumen von $V_h = 722 \text{ cm}^3$ entwickelt der Motor eine Nennleistung von $P_{eZ} = 11,2 \text{ kW} @ 3000 \text{ min}^{-1}$.

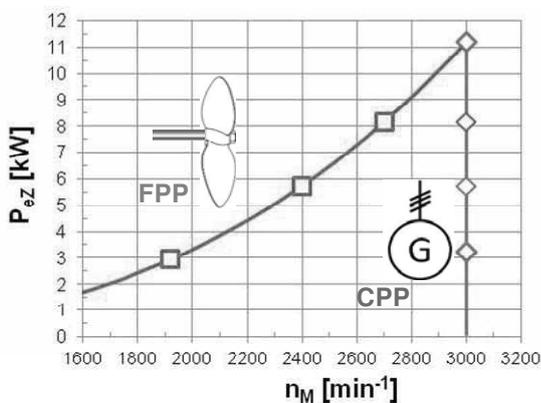


Abbildung 2. Untersuchter Betriebsbereich

Untersucht werden soll entsprechend Abb. 2 der Betrieb in Viertellast-Schritten auf der Festpropeller-Kurve (FPP) sowie im Konstantdrehzahl-Betrieb (CPP).

Die Zielsetzung selbst ist quantitativ zu formulieren. In Auswertung eigener Untersuchungen werden Simulationsergebnisse für industriell verwendete Verbrennungsmotoren dann verwertbar, wenn die folgenden Toleranzen erreicht werden:

- Zünddruck: $\Delta p_{Z_{\max}} \leq 3 \text{ bar}$
- Spez. Verbrauch: $\Delta b_{e42,7} \leq 3 \%$
- Spez. Luftdurchsatz: $\Delta l_e \leq 3 \%$
- Abgastemperatur: $\Delta t_{\text{Abg}} \leq 15 \text{ K}$

Mit abnehmender Motorlast werden die bisher bekannten Berechnungsmodelle zunehmend ungenauer. Sowohl aus ökonomischen Gründen als auch im Hinblick auf die Belastung der Bauteile des Motors spielen Leistungen unterhalb von 50 % des Nennpunktes eine untergeordnete Rolle, sodass die Anforderungen hier entschärft werden können, indem die Toleranzen nur für Leistungen über Halblast eingefordert werden sollen.

Das Hauptwerkzeug zur Erreichung dieser Zielsetzung stützt sich dabei auf ein einfaches Einzonen-Modell [1] zur Berechnung großvolumiger Dieselmotoren für den Schiffs- und Kraftwerksbetrieb, das auch vielen heute kommerziell verwendeten Berechnungsvorschriften zugrunde liegt. Dieses ist zunächst unverändert mit den Daten des Versuchsträgers zu parametrieren.

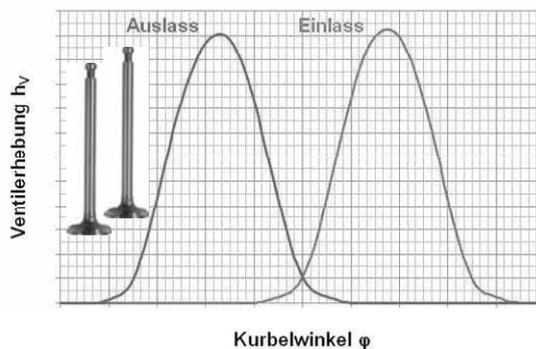


Abbildung 3. Ventilsteuerzeiten und -erhebungen

Neben den konstruktiven Grundabmaßen Hub, Bohrung, Pleuellänge sind wie aus Abb. 3 ersichtlich

ebenso die Abmaße der Ventile sowie deren kurbelwinkelabhängige Erhebungen zu implementieren.

Mit den Abmaßen und der Anzahl der Ein- und Auslassventile sind die Querschnitte geometrisch bekannt, die der in den Zylinder ein- und aus dem Zylinder ausströmenden Gasmasse zur Verfügung stehen. Aufgrund von Einschnürungen der Stromlinien und Diffusorwirkungen bei der Durchströmung ist ein Strömungsbeiwert experimentell in Abhängigkeit der Ventilerhebung h_v zu bestimmen, sodass aus dem Produkt von geometrischem Querschnitt und Strömungsbeiwert der kurbelwinkelabhängige effektive Ventilquerschnitt A_{eff} für Ein- und Auslass zu berechnen und dem Simulationsmodell vorzugeben ist (Abb. 4).

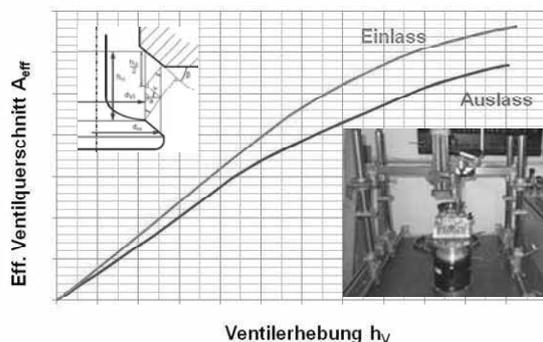


Abbildung 4. Effektive Ventilquerschnitte

Als zweckmäßig stellte sich heraus, das Simulationsmodell nunmehr mit einer experimentell aus dem indizierten Zylinderdruckverlauf ermittelten Wärmefreisetzung im Zylinder zu beaufschlagen. Das Vorgehen kann so verstanden werden, dass zunächst ein ausgewählter Betriebspunkt inklusive kurbelwinkel aufgelöstem Zylinderdruck vermessen wird. Hieraus wird mit dem Einzonenmodell die Brennstoffumsetzungsrate im Zylinder kurbelwinkelabhängig berechnet:

$$\frac{dq}{d\phi} = \frac{1}{m_{z,ES}} \cdot \frac{dQ}{d\phi}$$

Dieser sogenannte Real-Brennverlauf wird dem Einzonenmodell als Eingangsgröße vorgegeben, das hieraus wieder den Zylinderdruckverlauf sowie die integralen Betriebswerte berechnet. Das Teilmodell „Brennstoffumsetzung“ mit seinen Unsicherheiten entfällt zunächst. Im Idealfall liegen der gemessene und der berechnete Druckverlauf übereinander. Die berechneten integralen Größen (Abgastemperatur,

Verbrauch, Luftdurchsatz) wären mit den gemessenen identisch.

Realistisch betrachtet wird an dieser Stelle zunächst die Überarbeitung und Anpassung diverser Teilmodelle (u.a. Zündverzug, Wandwärme) ohne den Einfluss synthetischer Brennverläufe ermöglicht.

Erst nach Abschluss dieser Phase ist es zweckmäßig, die Wärmefreisetzung zu beschreiben. Das bisher standardisierte Vorgehen ist der bestmögliche Abgleich des synthetischen Brennverlaufes in einem ausgewählten Referenz-Betriebspunkt. Von diesem aus soll mit bekannten oder noch zu erarbeitenden Reduktionsgleichungen die Umrechnung der Wärmefreisetzung für Betriebspunkte mit abweichenden Bedingungen (Last, Drehzahl, Umgebung) erfolgen. Dieses Grundprinzip soll für die folgenden Untersuchungen beibehalten werden.

3 Nachrechnung mit Real-Brennverlauf

Entsprechend dem skizzierten Vorgehen wird der Versuchsträger im festgesetzten Betriebsbereich vermessen und indiziert. Mit den so ermittelten Real-Brennverläufen werden die einzelnen Betriebspunkte nachgerechnet. Abb. 5 zeigt den indizierten Druckverlauf sowie die hieraus ermittelte reale Wärmefreisetzung für den Nennpunkt des Motors. Zusätzlich ist der im Anschluss berechnete Zylinderdruckverlauf eingetragen.

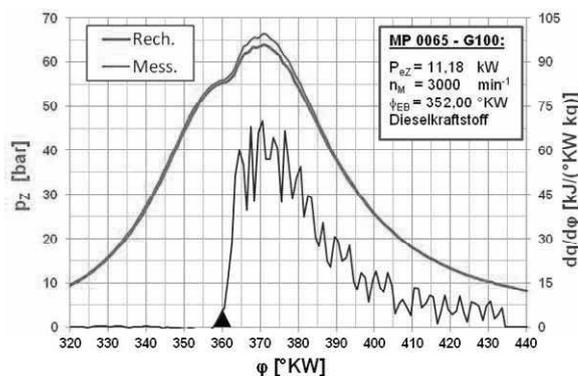


Abbildung 5. Druckverläufe in Messung und Rechnung

Während die Druckverläufe in der Kompressionsphase noch gut übereinander liegen, zeigt die Simulation bereits einen zu niedrigen Zylinderdruck während der Verbrennung.

Größe	Messung (vom 17.12.2013)	Rechnung (Datei: mp0065wo.cim)
pZmax (Zünddruck)	66,5 bar	64,1 bar
be42.7 (Kraftstoffverbr.)	276 g/kWh	245 g/kWh
le (Luftdurchsatz)	5,6 kg/kWh	5,6 kg/kWh
tAbg (Abgastemp.)	655 °C	642 °C
Q'w (Wandwärme)	6,85 kW	4,38 kW

Tabelle 1. Integrale Betriebswerte für den Nennpunkt

Tabelle 1 stellt die berechneten integralen Betriebswerte den gemessenen gegenüber. Deutlich wird hierbei insbesondere der um 11,2 % zu niedrig berechnete Kraftstoffverbrauch. Diese Abweichung ist so nicht akzeptabel. Zur Klärung der Frage, warum das für Großdieselmotoren seit Jahrzehnten bewährte Berechnungsmodell hier versagt, werden weitere Betriebswerte gesichtet. Auffallend ist hierbei, der trotz aller Unsicherheiten bei der experimentellen Bestimmung deutlich zu niedrig berechnete Wandwärmeverlust, der mit der Kühlluft dem Zylinder entzogen wird.

Der Wärmeübergang vom Zylindergas an die Zylinderwände wird üblicherweise als rein konvektiver Vorgang abgebildet, sodass als einzige zu beeinflussende Größe der Wärmeübergangskoeffizient α_w in Frage kommt. Verschiedene bekannte halbempirische Ansätze zu seiner Berechnung wurden untersucht. Neben der universell einsetzbaren Beziehung nach Woschni [2] wurde auch die für Pkw-Motoren gebräuchliche Beschreibung nach Hohenberg [3] verwendet, ohne dass eine grundsätzliche Änderung der berechneten Betriebswerte eintrat. Als Konsequenz wurde ein eigener Ansatz entwickelt, der den Wärmeübergang während der Verbrennungsphase gegenüber den bekannten Beschreibungen deutlich erhöht. Abb. 6 zeigt dessen Ergebnisse im Vergleich zu den Woschni-Angaben.

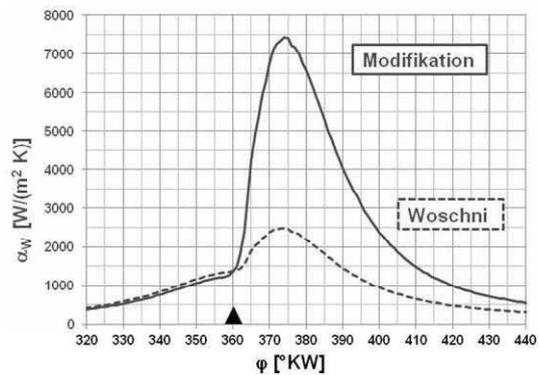


Abbildung 6. Modifizierter Wandwärmevergang

Kritisch anzumerken ist hierbei, dass diese modifizierte Beziehung lediglich an die globalen Betriebswerte angepasst wurde. Eine Überprüfung mit schnellen Wärmestromsonden konnte aus finanziellen Gründen noch nicht erfolgen.

Mit dem modifizierten Wandwärmevergang wurden die Betriebswerte für Propeller- und Konstandrehzahl-Linie berechnet und in Abb. 7 den Messergebnissen gegenüber gestellt. Sowohl bei den dargestellten Größen Verbrauch, Zünddruck und Abgastemperatur, als auch beim nicht dargestellten Luftdurchsatz konnte die Vorgaben bezüglich der zulässigen Toleranzen unterschritten werden.

Aufgrund dieser Ergebnisse konnte noch ein weiterer Schritt zur rechnerischen Beschreibung dieses Motors gegangen werden, indem ein an [4] angelehntes Zweizonenmodell zur Berechnung der spezifischen NOx-Emission aufgesetzt wurde. Auch dessen Ergebnisse sind zufrieden stellend.

Das modifizierte Einzonmodell liefert auf der Basis von Real-Brennverläufen, die aus indizierten Zylinderdruckverläufen gewonnen wurden, verwertbare Ergebnisse. Die Erweiterung des Modells auf die mathematische Beschreibung von Ersatz-Brennverläufen, inklusive deren Umrechnung auf abweichende Betriebsbedingungen erscheint sinnvoll.

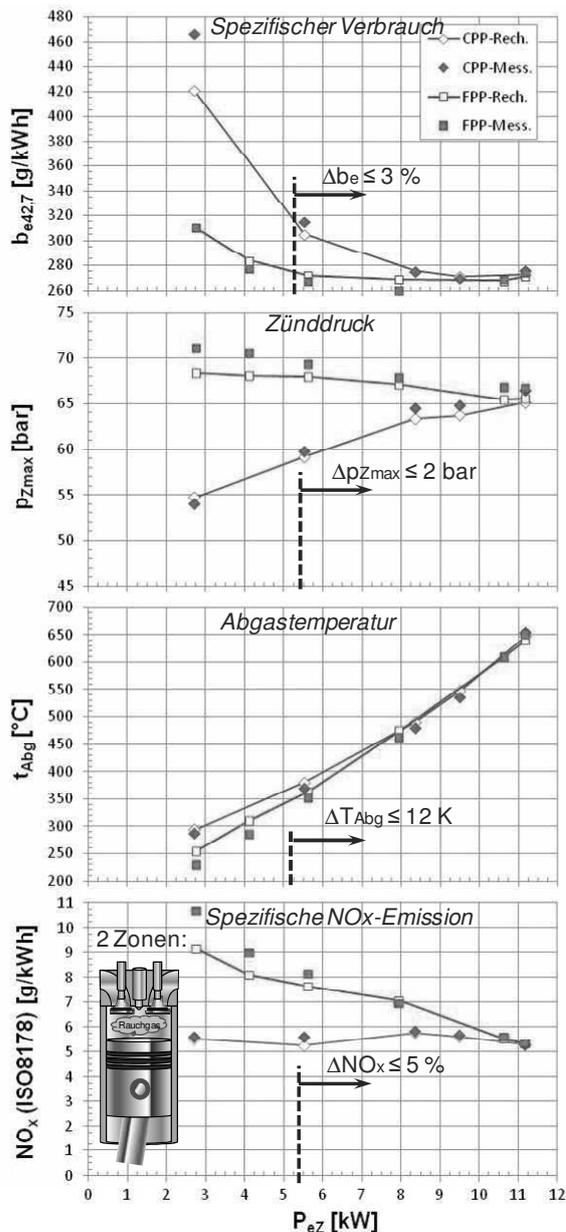


Abbildung 7. Messung/Rechnung-Vergleich mit Real-BV

4 Beschreibung mit Ersatz-Brennverlauf

4.1 Auswahl einer geeigneten Ersatz-Funktion

Klassischer Weise wird die Wärmefreisetzung im Dieselmotor mit einer mathematischen Ersatz-Funktion nach I.I. VIBE [5] beschrieben (Abb. 9 oben), die neben ihrer relativ einfachen Struktur zudem über eine allgemein gültige Umrechnungsvorschrift für abweichende Betriebsbedingungen verfügt [6].

Betrachtet man einen Hochlast-Brennverlauf wie in Abb. 8 oben gezeigt, wird die einfache VIBE-Funktion diesen in Bezug auf seine grundsätzliche Form gut abbilden können. Für Berechnung bei Teillast, bei denen naturgemäß der Anteil der vorge-mischten Verbrennung anwächst (Abb. 8 unten), wird deutlich, dass hier diese Art der Beschreibung an ihre Grenzen gerät.

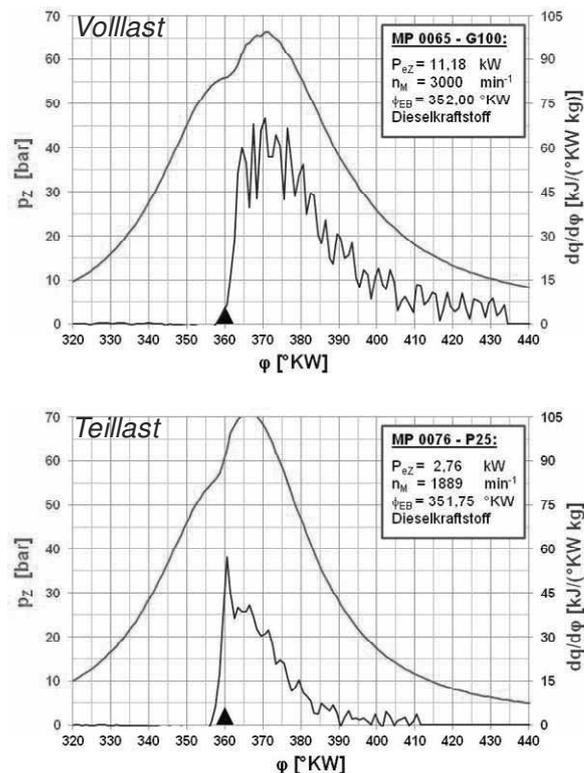


Abbildung 8. Real-Brennverläufe bei Hoch- und Teillast

Um dem zweistufigen Charakter (vorge-mischt/Diffusion) der dieselmotorischen Verbrennung, gerade in Verbindung mit hohen Drehzahlen und niedrigen Einspritzdrücken gerecht zu werden, bedarf es komplizierterer Ansätze wie von OBERG [7] oder SCHREINER/JÄGER [8, 9] formuliert (Abb. 9 mitte + unten).

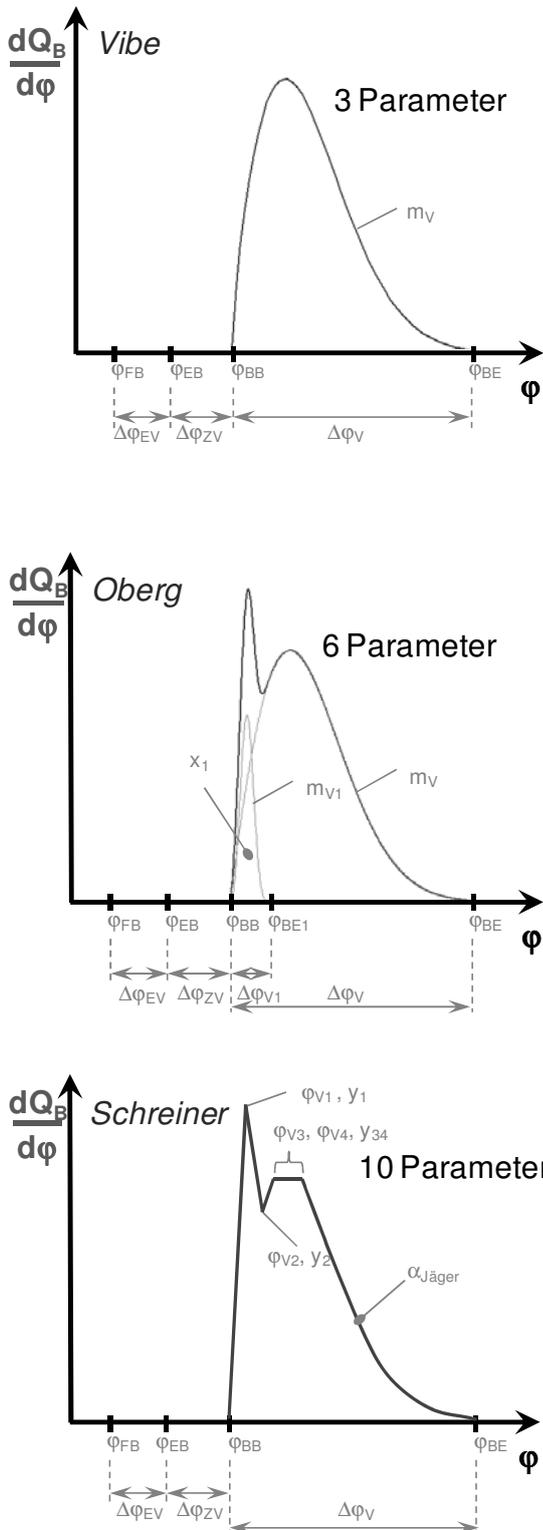


Abbildung 9. Mögliche Ersatz-Brennverläufe

Nachteilig bei den beiden letzten Beschreibungsvarianten ist die Zunahme der zu parametrierenden Größen, die zudem mit einer entsprechenden Anzahl von

Reduktionsgleichungen auf die vom Abgleichpunkt abweichenden Betriebspunkte umgerechnet werden müssen. Zudem fehlen bei beiden allgemein anerkannte Umrechnungen, wie sie für die einfache VIBE-Funktion aus [6] bekannt sind.

Mit diversen ausgewählten Probeberechnungen wurde versucht, den Beschreibungsaufwand ins Verhältnis zum erzielten Gewinn an Berechnungsgenauigkeit zu setzen. Dabei zeigte sich, dass im Falle der Beschreibung kleiner Industrie-Dieselmotoren die OBERG-Funktion mit ihren 6 Parametern für Berechnungen, die den oben definierten Toleranzgrenzen entsprechen ausreicht.

Als Konsequenz aus dieser Erkenntnis waren entsprechende Reduktionsgleichungen in Anlehnung an [6] selbst zu entwickeln.

4.2 Umrechnung des OBERG-Brennverlaufes

Die grundsätzliche Idee der Verwendung von Ersatz-Brennverläufen ist deren Abgleich an einem bekannten, experimentell vermessenen Referenzpunkt und deren Umrechnung für abweichende Betriebsbedingungen.

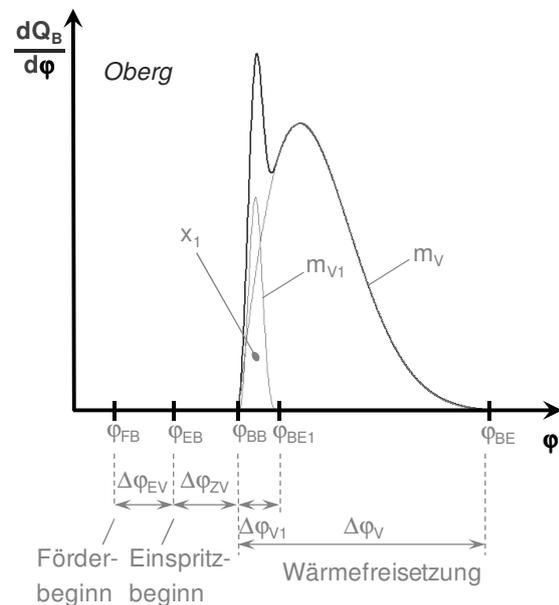


Abbildung 10. Ersatz-Brennverlauf nach OBERG

Der OBERG-Brennverlauf (Abb. 10) wird durch die Parameter Brennbeginn ϕ_{BB} , die Brenndauern von vorgemischter und Diffusionsverbrennung $\Delta\phi_{V1}$ und $\Delta\phi_V$, deren Formfaktoren m_{V1} und m_V sowie dem Energieanteil der vorgemischten Verbrennung x_1

beschrieben. Hinzu kommt bei bekanntem Einspritzbeginn φ_{EB} der Zündverzug $\Delta\varphi_{ZV}$.

Der Einspritzbeginn ist aus der Messung des Nadelhubes der Einspritzdüse bekannt. Des Weiteren existieren auch hier eine Reihe halb-empirischer Funktionen, die die Dauer des Zündverzuges in Abhängigkeit der Parameter mittlere Zylindertemperatur \bar{T}_Z und mittlerer Zylinderdruck \bar{p}_Z während der Zündverzugszeit. Auch waren zahlreiche Proberechnungen durchzuführen, die zeigten, dass eine modifizierte Form des Ansatzes nach WOLFER/MARZOUK [10] die besten Resultate liefert und somit für die weitere Berechnung zur Anwendung kommt.

Analog zu [6] konnten einfache Funktionsansätze gefunden werden, die die OBERG-Parameter des Abgleichpunktes in Abhängigkeit globaler Betriebsgrößen wie Verbrennungsluftverhältnis λ_V , Motordrehzahl n_M , Nutzmitteldruck p_e und Zündverzug $\Delta\varphi_{ZV}$ umrechnen. In Anlehnung an [6] wurde auch hier die Strategie verfolgt, jene Einflussgrößen auf deren Wert im Referenzpunkt (= Abgleichpunkt) zu normieren (Abb. 11).

Reduktionsgleichungen:

$$m_{V1} = \text{konstant}$$

$$\Delta\varphi_{V1} = \Delta\varphi_{V1_A} \cdot f(\lambda_V^*, n_M^*, \Delta\varphi_{ZV}^*)$$

$$x_1 = x_{1_A} \cdot f(\lambda_V^*, n_M^*, \Delta\varphi_{ZV}^*, p_e^*)$$

$$m_V = m_{V_A} \cdot f(\lambda_V^*, n_M^*)$$

$$\Delta\varphi_V = \Delta\varphi_{V_A} \cdot f(\lambda_V^*, n_M^*)$$

mit * = auf den Abgleichpunkt „A“ normierte Betriebsgröße

Abbildung 11. Umrechnung des Ersatz-Brennverlaufes

Erfreulich war hier die Tatsache, dass der Formfaktor m_{V1} der vorgemischten Verbrennung aufgrund ihrer kurzen Brenndauer ohne nennenswerten Einfluss war und auf einen konstanten Wert belassen werden konnte.

4.3 Nachrechnungen mit OBERG-Brennverlauf

Analog zu der Nachrechnung mit Real-Brennverläufen wurden diese für die Verwendung von den OBERG-Ersatz-Brennverläufen wiederholt. Die Parameter des jeweiligen Brennverlaufes wurden dabei ausgehend vom Abgleichpunkt mit den Reduktionsgleichungen nach Abb. 11 entsprechend umgerechnet. Die Abweichungen zwischen Messung und

Rechnung sind nun naturgemäß größer als bei der Verwendung von Real-Brennverläufen (Abb. 12).

Selbst hiermit ergeben sich im Leistungsbereich oberhalb von 50 % der Nennleistung sowohl im Propeller-, als auch im Konstantdrehzahl-Betrieb Abweichungen zwischen Messung und Rechnung, die nahezu den maximal zulässigen Werten der Vorgabe entsprechen. Die Vorgabetoleranz im spezifischen Kraftstoffverbrauch wird um 0,6 %, die der Abgastemperatur um 3 K überschritten.

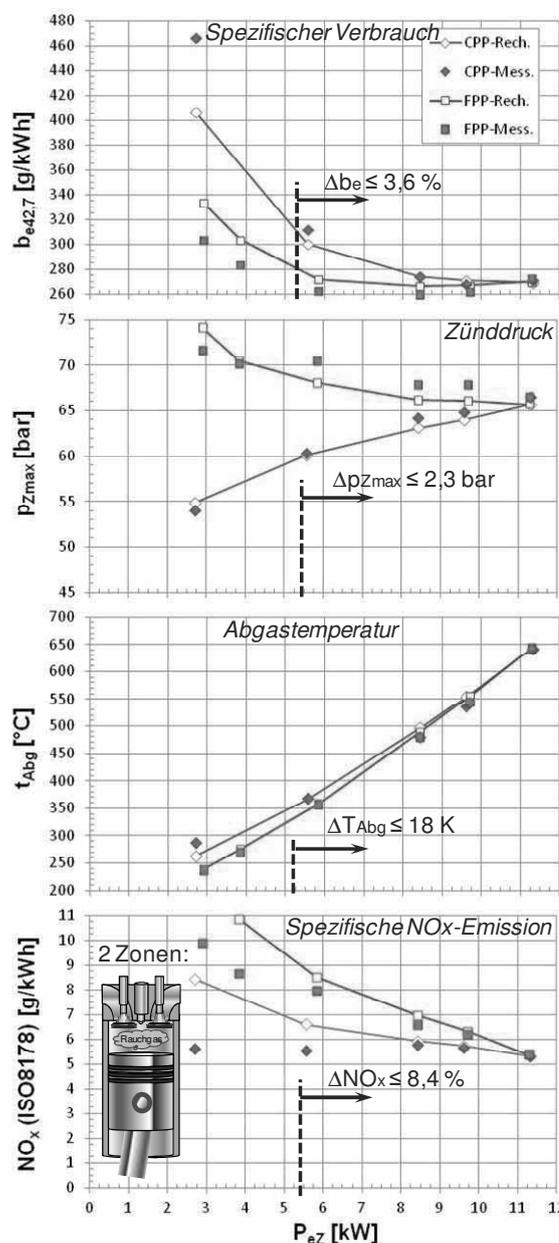


Abbildung 12. Messung/Rechnung-Vergl. mit OBERG-BV

Auch hier erschien es sinnvoll, das aufgesetzte Zweizonen-Modell zu erproben. Mit Maximalabweichungen kleiner als 10 % ergeben sich auch hier praktisch verwertbare Ergebnisse.

Wie aus Abb. 8 zu erkennen ist, wächst der Anteil der vorgemischten Verbrennung vom Nennlastpunkt, der zeitgleich als Abgleichpunkt für den Brennverlauf fungierte, zur Teillast hin stark an. Das bedeutet, dass die Reduktionsgleichung zur Berechnung der Energieanteils der Vormischphase x_1 eine massive Verstärkung aufweisen muss, sodass sich bereits kleine Fehler beim Abgleich stark auf die Energieaufteilung bei Teillast auswirken. Hier wäre die Beschreibung von x_1 als Funktion absoluter Betriebsgrößen eine sinnvolle Alternative, um die Auswirkungen von Fehlern bei Abgleich zu minimieren.

5 Zusammenfassung der Ergebnisse

Im Zuge eines Klein-Forschungsprojektes wurde eine bestehende Kreisprozessberechnung, deren Anwendung sich vornehmlich auf Großdieselmotoren mit Zylinderleistung größer 100 kW und Drehzahlen kleiner 1000 min^{-1} erstreckte, auf die Nutzung für luftgekühlte Klein-Dieselmotoren erweitert. Hierzu mussten die Teilmodelle für den Ersatz-Brennverlauf und dessen betriebsgrößenbedingte Umrechnung, den Zündverzug sowie den Wandwärmeübergang überarbeitet und entsprechend angepasst werden. Als zu untersuchender Betriebsbereich wurden die Festpropeller- sowie die Konstantdrehzahllinie definiert.

Tabelle 2 zeigt die maximalen Abweichungen zwischen Messung und Berechnung für den Leistungsbe- reich ab $P_e^* \geq 50 \%$. Die ursprünglich vereinbarten zulässigen Toleranzen (she. Pkt. 2) werden leicht überschritten. Das Modell liefert dennoch verwertbare Rechenergebnisse.

Größe	Real-BV <small>Dateien: MP00XXdx.cim</small>	OBERG-BV <small>mit Umrechnung</small>
Δp_{Zmax} (Zünddruck)	$\leq 2 \text{ bar}$	$\leq 2,3 \text{ bar}$
$\Delta be_{42.7}$ (Kraftstoffverbr.)	$\leq 3 \%$	$\leq 3,6 \%$
Δt_{Abg} (Abgastemp.)	$\leq 12 \text{ K}$	$\leq 18 \text{ K}$
ΔNO_x (Stickoxide)	$\leq 5 \%$	$\leq 8,4 \%$

Tabelle 2. Maximale Abweichungen für $P_e^* \geq 50\%$

Als zusätzliche, hier nicht dargestellte Untersuchungen wurden mit der ebenfalls zur Verfügung stehenden Konditionierungsanlage für die Verbrennungsluft die Auswirkungen unterschiedlicher klimatischer Verhältnisse auf den Motorbetrieb und die Stickoxidemissionen experimentell erforscht und mit guter Genauigkeit rechnerisch nachvollzogen. Weiterhin gelang es, den Einfluss der Temperatur des eingesetzten Kraftstoffes zufrieden stellend rechnerisch abzubilden.

Weiterführende Arbeiten ergeben sich sinnvollerweise aus der Erkenntnis zur Beschreibung der x_1 -Reduktion mit absoluten Betriebsgrößen. Mit geringem finanziellen Zusatzaufwand (< 2000 Euro) kann eine experimentelle Überprüfung der bisher verwendeten Modellierung des Wandwärmeüberganges erfolgen. Ein Reibdruckmodell sowie die Untersuchung weiterer Motorentypen würden diese Arbeiten abrunden.

6 Literaturangaben

- [1] Flenker, H.; Woschni, G.:
Programmiertes Berechnungsverfahren zur Bestimmung der Prozeßdaten aufgeladener Vier- und Zweitakt Dieselmotoren bei geänderten Betriebsbedingungen.
Bericht des Inst. für Verbrennungskraftmaschinen TU Braunschweig Nr. 74/2 / CIMAC Working Group "Supercharging", Frankfurt/M. 1974
- [2] Woschni, G.:
Die Berechnung der Wandverluste und der thermischen Belastung der Bauteile von Dieselmotoren.
Motortechnische Zeitschrift 31 (1970), S. 491ff.
- [3] Hohenberg, G.:
Experimentelle Erfassung der Wandwärme von Kolbenmotoren.
Habilitationsschrift, TU Graz, 1980
- [4] Heider, G.:
Rechenmodell zur Vorausberechnung der NO-Emissionen von Dieselmotoren.
Dissertation, TU München, 1996

[5] Vibe, I.I.:

Brennverlauf und Kreisprozeß von Verbrennungsmotoren.

VEB Verlag Technik, Berlin 1970

[6] Woschni, G.; Anisits, F.:

Eine Methode zur Vorausberechnung der Änderungen des Brennverlaufs mittelschnelllaufender Dieselmotoren bei geänderten Betriebsbedingungen.

Motortechnische Zeitschrift 34 (1973), S. 106 ff.

[7] Oberg, H.-J.:

Darstellung des Brennverlaufes eines schnelllaufenden Dieselmotors durch zwei überlagerte Vibe-Funktionen.

Dissertation TU Braunschweig, 1976

[8] Schreiner, K.:

Untersuchungen zum Ersatzbrennverlauf und Wärmeübergang bei schnelllaufenden Hochleistungsdieselmotoren.

4. Tagung „Der Arbeitsprozess des Verbrennungsmotors“, Graz 1993

[9] Jäger, G.:

Vorschlag zur Modifikation des Polygonzug/Hyperbel-Brennverlaufes in der Ausbrandphase.

Arbeitspapier FH Stralsund, 2013

[10] Gerstle, M.:

Simulation des instationären Betriebsverhaltens hochaufgeladener Vier- und Zweitakt-Dieselmotoren.

Dissertation Uni Hannover, 1999

7 Danksagung

Die Autoren danken der Forschungskommission der FH Stralsund für die finanzielle Förderung der beschriebenen Untersuchungen.



Der Prüfstand mit dem verwendeten Versuchsträger wurde im Rahmen des Programmes „Großgeräte der Länder“ beschafft.



Die Anlage zur Luftkonditionierung, die Totgewichtswaage zur Prüfung der Indiziermesskette sowie die verwendete Abgasanalyse wurden durch den EFRE mitfinanziert.



EUROPÄISCHE UNION

Europäischer Fonds für
Regionale Entwicklung

Investition in Ihre Zukunft!

Weiter Dank für die Unterstützung mit Sachmitteln und Prüfstandszeiten gebühren der HATZ-Niederlassung Nord, dem Institut für Verbrennungskraftmaschinen der TU Braunschweig sowie dem Institut für Energietechnik der TU Hamburg-Harburg.

Plenarvortrag 2

Raum 111

Sitzungsleitung: Heinz-Theo Mammen

Simulationsgestützter Entwurf von ActiveNoiseControl-Systemen für Kabinenanwendungen

Joachim Waßmuth
Fachhochschule Bielefeld

Simulationsgestützter Entwurf von ActiveNoiseControl-Systemen für Kabinenanwendungen

Prof. Dr.-Ing. Joachim Waßmuth

Fachhochschule Bielefeld, Institut für Systemdynamik und Mechatronik,

33602 Bielefeld, Deutschland

joachim.wassmuth@fh-bielefeld.de

Der Entwurf von technischen Systemlösungen zur Störschallkompensation lässt sich durch die Nutzung von Modellen für den virtuellen und realen Systemtest systematisieren und optimieren. In diesem Beitrag wird der Einsatz von Modellbildung und Simulation genutzt, um einerseits die Wechselwirkungen der sensorischen und aktuatorischen Teilkomponenten, der Transferpfade und der zu untersuchenden ActiveNoiseControl-(ANC-)Algorithmen in der Entwurfsphase beurteilen zu können. Die Verwendung von Modellen unterschiedlicher Detaillierungstiefe ermöglicht dabei auch Analysen, inwieweit ein ANC-System auch unter Verwendung kostengünstiger Teilkomponenten realisierbar ist, da der Kostenfaktor i.d.R. ein großes Hemmnis bei einem Einstieg in diese Technologie darstellt. Die hier aufgeführten Arbeiten beschreiben exemplarisch die Anwendung von ANC für Kabinen, bei denen extern generierter Störschall einen Bedienerarbeitsplatz in einer geschlossenen Kabine beeinträchtigt.

1 Einleitung

Technische Systeme zur Reduktion von Störschall sind in den vergangenen Jahren zunehmend am Markt zu finden, der Einsatz beschränkt sich aber i.d.R. noch auf hochpreisige Anwendungen im Komfortbereich. Gleichsam besteht aber ein Bedarf in zahlreichen weiteren Anwendungen, da Lärm aus gesundheitlicher wie volkswirtschaftlicher Betrachtung [1] negative Folgen hat. Ansätze zur Störschallreduktion sind in der Literatur zahlreich beschrieben [2, 3], ein Einsatz bleibt vielen Unternehmen aber oft aufgrund der Komplexität der Fragestellung verwehrt. Zusätzlich wird die Option für einen erfolgreichen Einsatz der Technologie durch die fehlende Verfügbarkeit und durch die Leistungseinschränkungen der verwendbaren, notwendigerweise kostengünstigen Systemkomponenten als zu risikoreich bewertet. Begrenzende Faktoren sind hier wie in jedem mechatronischen System die Leistungsfähigkeit der Aktuatoren oder aber auch die zur Verfügung stehende Rechenleistung der eingebetteten Informationsverarbeitung. Das Projekt „Low Cost Mechatronic Systems“ (LoCoMS) verfolgt die Fragestellungen, wie sich simulationsgestützt die relevanten Systemkomponenten im Entwurf analysieren lassen, wobei zunächst exemplarisch eine Anwendungsklasse Kabinenschall betrachtet wurde. Über eine weitere Anwendungsklasse wurde in [4] berichtet. In diesem Beitrag werden daher zunächst

das Grundprinzip der aktiven Störschallreduktion (Active Noise Control, ANC) für die Anwendung in Kabinen und die dabei sich ergebenden Fragestellungen erläutert. Im dritten Abschnitt wird auf die Entwicklungsmethodik und die unterschiedlichen Modellbildungen eingegangen. Die daraus resultierenden Ergebnisse sowohl hinsichtlich der gewählten Methodik als auch der Anwendung werden in Kapitel vier dargestellt.

2 ANC für Kabinenschallanwendungen

Die hier betrachtete Anwendungsklasse umfasst Szenarien, bei denen ein extern generierter Störschall über ein oder mehrere Transferpfade (Körper- oder Luftschall) in eine geschlossene Kabine eindringt und innerhalb der Kabine an mindestens einer Position, i.d.R. an einem Bedienerarbeitsplatz, eine Beeinträchtigung in Form von Lärm darstellt. Damit vereinheitlicht diese Anwendungsklasse zahlreiche praktische Situationen, wie sie bspw. bei Bedienerarbeitsplätzen von mobilen Maschinen (Agrartechnik, Baufahrzeuge,...) aber auch bei Einhausungen zur Bedienung und Überwachung von ortsfesten Maschinen im industriellen Umfeld vorkommen können. Die zu betrachtenden Störschallsignale hängen dabei von den jeweils lärm erzeugenden Prozessen ab, so dass hier eine Unterscheidung notwendig ist.

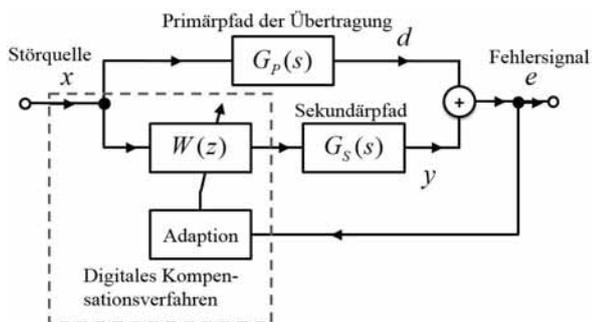


Abbildung 1. Übersicht ANC-System in Feedforward-Struktur

Der für die genannte Anwendung gewählte Ansatz entspricht einer sog. Feedforward-Struktur [3], d.h. Informationen über das Störsignal x müssen zunächst über einen Referenzsensor zur Verfügung stehen (Abbildung 1). Mit Hilfe eines digital realisierten Kompensationsverfahrens werden Ansteuersignale berechnet, die über einen Lautsprecher als Aktuator innerhalb der Kabine abgestrahlt werden. An einem definierten Ort in der Kabine, i.d.R. im Kopfbereich des Bedieners wird ein Fehlersignal erfasst. Neben dem Primärpfad der Störschallübertragung wird damit ein weiterer sog. Sekundärpfad mit dem Ziel realisiert, am Bedienerplatz ein Überlagerung des zusätzlichen Schalls mit dem Störschall zu erzielen, so dass das Fehlersignal und damit der Schalldruck am Bedienerplatz idealerweise reduziert werden. Das Fehlersignal wird im gewählten Ansatz dazu genutzt, eine Adaption des Kompensationssystems durchzuführen, da dieses einerseits nicht statisch vorausberechnet werden kann und andererseits somit Änderungen der Transferpfade und Störsignale berücksichtigt werden können. Nach erfolgreicher Adaption entsprechen sich Primärpfad $G_P(s)$ und die Kombination (Gesamtübertragungsfunktion) aus digitalem Kompensationssystem $W(z)$ und Sekundärpfad $G_S(s)$, zumindest in einem relevanten Frequenzbereich, einerseits hinsichtlich der Amplitudenwirkung, andererseits muss sich zur Kompensation eine Phasendifferenz von 180° ergeben (destruktive Interferenz).

3 Entwicklungsmethodik und Modellbildung

Die Untersuchungen zur Anwendungsklasse Kabinenschall wurden an einem Demonstrator unter Laborbedingungen durchgeführt, der zunächst erläutert werden soll. Anschließend wird kurz auf die verwendeten Verfahren und Algorithmen eingegangen, in

zwei weiteren Abschnitten werden die verwendeten Ansätze zur Modellierung der Übertragungspfade (Primärpfad, Sekundärpfad) sowie zur Modellierung der Störquelle erläutert. Die überlagerte Entwicklungs- und Testmethodik beschließt das Kapitel.

3.1 Anwendungsdemonstrator

Mangels Verfügbarkeit verschiedener Maschinen wurde zur Verallgemeinerung der Anwendungsklasse und zur Reproduzierbarkeit der Untersuchungen ein Anwendungsdemonstrator im Labor aufgebaut.

Dieser besteht aus einer Fahrerkabine einer Landmaschine, bei der am Sitzplatz des Fahrers mehrere Mikrofone eingebracht wurden (Abbildung 2). An der Kabinenrückwand dient ein kostengünstiges Elektretmikrofon zur Erfassung des für das Kompensationsverfahren benötigten Fehlersignals e , an den Hörpositionen erfassen zusätzliche Messmikrofone die Leistungsfähigkeit des gerade untersuchten Verfahrens. Über zusätzliche in der Kabine verbaute Lautsprecher wird der Kompensationsschall abgestrahlt. Die Kompensationssignale y werden auf einem echtzeitfähigen Entwicklungssystem (RapidControlPrototyping-System) berechnet, das notwendige Referenzsignal x liefert ein weiteres Mikrofon.

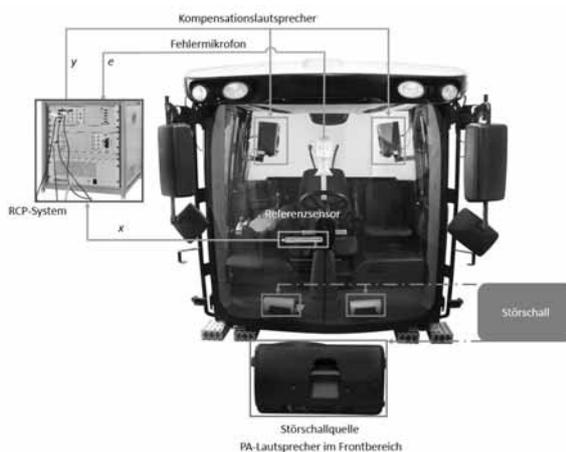


Abbildung 2. Demonstrator für die Anwendungsklasse Kabinenschall

Die Störschallsituation wird auf zweierlei Weise nachgebildet. Eine realitätsnahe externe Beschallung mit sowohl real gemessenen als auch synthetisch generierten Signalen verschiedener Anwendungsszenarien kann über ein PA-Lautsprecher im Frontbereich geschehen, zur Reduktion des Lärms im Gebäude kann alternativ mit zwei kleineren im Fußraum der Kabine eingebrachten Lautsprechern gearbeitet werden.

3.2 Verfahren und Algorithmen

Aufgrund des bereits gewählten Ansatzes einer sog. Feedforward-ANC Struktur müssen nun noch in Frage kommende Algorithmen analysiert werden. In der Literatur [3,5] sind die Grundlagen der hier genutzten Verfahren umfassend beschrieben, so dass hier nur auf das Grundkonzept und die umgesetzten Anpassungen und eigenen Erweiterungen eingegangen wird. Für die zeitdiskrete Berechnung des Kompensationssignals $y(k)$ aus dem Referenzsignal $x(k)$ wird häufig ein nichtrekursives Digitalfilter (FIR-Filter) der Filterlänge L mit der z -Übertragungsfunktion $W(z)$ eingesetzt, bei dem das Ausgangssignal $y(k)$ durch eine diskrete Faltung der Filterkoeffizienten $\mathbf{w}(k)$ mit verzögerten Werten des Eingangssignals $x(k)$ gebildet wird:

$$y(k) = \mathbf{w}^T(k) \cdot \mathbf{x}(k) = \sum_{i=0}^{L-1} w_i(k) \cdot x(k-i) \quad (1)$$

$$\mathbf{w}(k) = [w_0(k) \ w_1(k) \ w_2(k) \ \dots \ w_{L-1}(k)]^T \quad (2)$$

$$\mathbf{x}(k) = [x(k) \ x(k-1) \ \dots \ x(k-L+1)]^T \quad (3)$$

Die Filterkoeffizienten entsprechen dabei gleichsam der Impulsantwort des Digitalfilters. Zusammen mit dem Sekundärpfad ergibt sich eine Gesamtübertragungsfunktion, die dem Primärpfad, zumindest in den relevanten Frequenzbereichen, im Amplitudenbereich entsprechen muss und dabei eine Phasendifferenz von 180° erzielt. Hierzu wird das Digitalfilter adaptiv betrieben, indem aus dem Fehlersignal

$$e(k) = d(k) + y(k) \quad (4)$$

als Kostenfunktion der aktuelle quadratische Fehler (als Schätzwert für den mittleren quadratischen Fehler)

$$J(k) = e^2(k) \quad (5)$$

gebildet wird. Zur Anpassung der Filterkoeffizienten wird der recht einfach zu bildende Gradient der Kostenfunktion gemäß

$$\begin{aligned} \nabla J(k) &= \frac{\partial e^2(k)}{\partial \mathbf{w}(k)} = 2e(k) \cdot \frac{\partial e(k)}{\partial \mathbf{w}(k)} \\ &= 2e(k) \cdot \frac{\partial}{\partial \mathbf{w}(k)} (d(k) + \mathbf{w}^T(k) \cdot \mathbf{x}(k)) = 2e(k) \cdot \mathbf{x}(k) \end{aligned} \quad (6)$$

benötigt. Mit Hilfe des Verfahrens des steilsten Abstiegs ergibt sich die Adaptionsvorschrift

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \nu \cdot \nabla J(k) \quad (7)$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \cdot e(k) \cdot \mathbf{x}(k) \quad (8)$$

zur Berechnung der Filterkoeffizienten des nächsten Zeittakts. Dabei stellt $\mu (=2\nu)$ eine geeignete Schrittweite dar, so dass sich die Filterkoeffizienten in jedem neuen Zeittakt so ändern, dass die Kostenfunktion minimiert wird (LeastMeanSquare-Verfahren/LMS) und das Fehlersignal idealerweise Null wird. Für diese LMS-Verfahren existieren zahlreiche Erweiterungen, die im Rahmen dieser Arbeiten ebenfalls untersucht wurden.

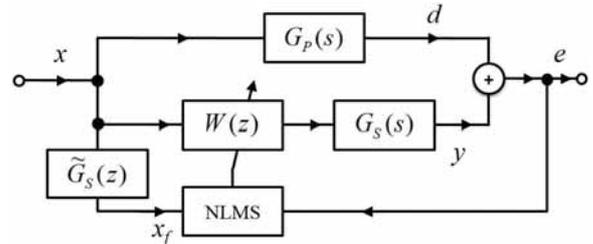


Abbildung 3. ANC -FxNLMS Struktur

Im Hinblick auf die Stabilität des Verfahrens ist es hilfreich [3,5], den Sekundärpfad $G_s(s)$ in Form einer zeitdiskreten Schätzung $\tilde{G}_s(z)$ zusätzlich zu berücksichtigen, indem das Referenzsignal x zunächst gefiltert wird (Filtered-x-LMS-/ FxLMS-Verfahren). Für diese Verfahren ist der Algorithmus um eine Vorabpfadschätzung zu erweitern. Desweiteren haben sich Modifikationen hinsichtlich der Schrittweitensteuerung des Algorithmus bewährt [5], etwa durch Normierung der Schrittweite auf die Signalleistung des Referenzsignals (NLMS-/FxNLMS-Verfahren), so dass sich eine modifizierte Schrittweite

$$\mu = \frac{\tilde{\mu}}{\mathbf{x}_f^T \cdot \mathbf{x}_f} \quad (9)$$

ergibt.

Im Hinblick auf eine kostengünstige Lösung sind weitere Modifikationen eingebracht worden, die den zu erwartenden Implementierungsaufwand reduzieren. Besonders aufwandsarm lässt sich bspw. die Berechnung der Signalleistung $\mathbf{x}_f^T \cdot \mathbf{x}_f$ durch eine einfache Quadrierung und anschließende Tiefpassfilterung mit einem rekursiven Filter niedriger Ordnung realisieren. Generell lässt sich die Rechenleistung durch Wahl einer möglichst niedrigen Betriebsrate reduzieren. Hierzu werden die Verfahren aber zur Anpassung an die externen Schnittstellen um Dezi-

mations- und Interpolationsfilter ergänzt (Abbildung 4).

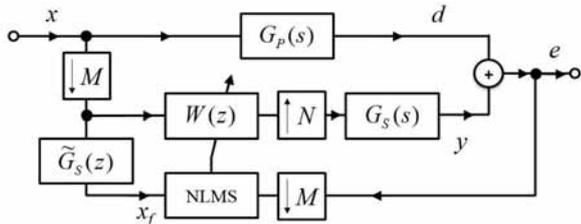


Abbildung 4. ANC –FxNLMS Struktur mit Dezimations- und Interpolationsfiltern

Die Wahl der Betriebsrate hat wiederum direkt Einfluss auf die notwendige Filterlänge L , da mit zunehmender Phasenwirkung des Primärpfades auch die Impulsantwort des Kompensationsfilters zeitlich länger wird. Aus der Literatur [5] ist weiterhin bekannt, dass ab einer bestimmten Filterlänge eine Implementierung der Filter im Frequenzbereich mit Hilfe der FFT (FastBlockLMS-Filter) aufwandsärmer wird, da eine Faltung im Zeitbereich einer Multiplikation im Frequenzbereich entspricht. Zu berücksichtigen ist allerdings, dass hiermit aufgrund der blockweisen Bildung der FFT auch eine zusätzliche Totzeit (Delay) der Verarbeitung einhergeht, die es zu bewerten gilt. Hinsichtlich der Stabilität der Algorithmen und Anwendbarkeit der Verfahren hat sich eine gehörig richtige also wahrnehmungsangepasste Filterung (z.B. sog. A-Bewertung) des Fehlersignals als hilfreich erwiesen, wie sie auch in [6] vorgeschlagen wird.

Die Auswahl, Anpassung und Parametrierung eines Algorithmus ist letztlich von den Übertragungspfaden und Störsignalen abhängig, eine simulationsgestützte Analyse ist folgerichtig also angeraten.

3.3 Pfadmodellierung

Der erfolgreiche Einsatz eines ANC-Systems hängt von der Kenntnis und Auslegung der Übertragungspfade ab. Dabei wird zwischen dem Primärpfad von der Störquelle zum Ort der Reduktion (Bedienerplatz) und dem zusätzlichen Sekundärpfad zur Realisierung der Kompensation unterschieden. Offensichtlich ist zunächst die zeitliche Verzögerung zwischen beiden Pfaden für die grundsätzliche Kausalität des Systems maßgeblich. Ist bspw. die Laufzeit über die digitale Kompensation und die Elemente des Sekundärpfades länger als die des Primärpfades, so wird der Kompensationsschall nicht rechtzeitig am Kompensationsort eintreffen können. Dann lassen sich bestenfalls

schmalbandige Signale mit tonalen Komponenten aufgrund von Periodizitäten in der Signalform kompensieren. Aus Kausalitätsgründen ist somit für beliebige Signalformen die Laufzeit des Sekundärpfades durch die des Primärpfades begrenzt. Des Weiteren sind die verwendeten sensorischen und aktuatorischen Komponenten einzeln hinsichtlich ihres Übertragungsverhaltens maßgeblich. Neben der Phasenwirkung ist hier natürlich die Amplitudenwirkung bedeutsam. Die Komponenten des Sekundärpfades sind dabei alle elektronischen (AD/DA-Umsetzer, Verstärker, Signalkonditionierung, Antialiasing- und Rekonstruktionsfilter), aktuatorischen (Lautsprecher) und sensorischen (Mikrofone, Beschleunigungsaufnehmer) Elemente und zusätzlich die akustische Schallausbreitung (Abbildung 5). Im Sinne einer umfassenden Modellbildung ist hier natürlich auch die Erfassung des Referenzsignals zu berücksichtigen, auch wenn diese nicht Bestandteil des Sekundärpfades ist.

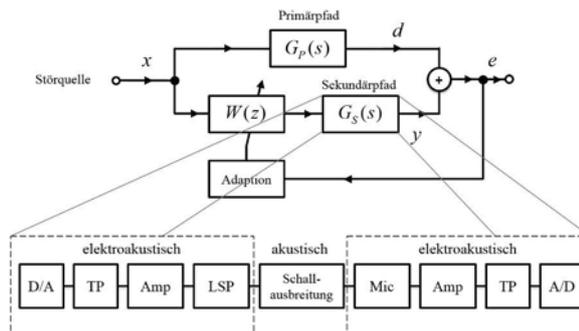


Abbildung 5. Elemente des Sekundärpfades

Die hier gewählte Modellierungsstrategie ermöglicht die Auswahl von Teilmodellen des Sekundärpfades in verschiedenen Modellierungstiefen. Dabei berücksichtigt die einfachste Modellierung lediglich ein Laufzeit- sowie Amplitudenverhalten (Sensitivität) der Komponenten. Für weitergehende Analysen lassen sich umfangreichere Modelle auf Basis von Übertragungsfunktionen oder linearen und nichtlinearen Differenzialgleichungen bei Bedarf auswählen. Dieser ergibt sich insbesondere dann, wenn konkrete Komponenten ausgewählt wurden und deren Einfluss auf das Gesamtsystem analysiert werden soll. Insbesondere die Analyse von Leistungseinbußen, die aus der Verwendung kostengünstiger Komponenten zu erwarten sind, wird durch diese Strategie effizient möglich. Alle Komponenten sind aber auch in einer trivialen Variante ohne Wirkung (das Signal wird unverändert weitergegeben) vorhanden, so dass die

Wirkung der Teilkomponente in der Simulation auch gezielt deaktiviert werden kann.

Der akustische Pfad beschreibt durch ein einfaches Modell die Abnahme des Schalldrucks bei der Ausbreitung im freien Raum, d.h. Reflexionen werden zunächst nicht berücksichtigt. Damit bleiben akustische Eigenschaften der Kabine wie Raummoden zunächst unberücksichtigt. Abhilfe würde hier eine dynamische FEM-Analyse schaffen. Um allerdings aufwändige und begrenzt parametrierbare FE-Modelle der Schallausbreitung in der Kabine umgehen zu können und darüber hinaus lange Rechenzeiten einer dynamischen FEM-Berechnung vermeiden zu können, lassen sich gemessene Impulsantworten als Modelle importieren. Die folgende im Demonstratoraufbau gemessene Impulsantwort des Primärpfades zeigt in Abbildung 6 neben einer Totzeit deutlich das Auftreten von einzelnen unterschiedlich schnell abklingenden Resonanzfrequenzen, die sich durch Raummoden erklären lassen, die letztlich aus der Kabinengeometrie resultieren.

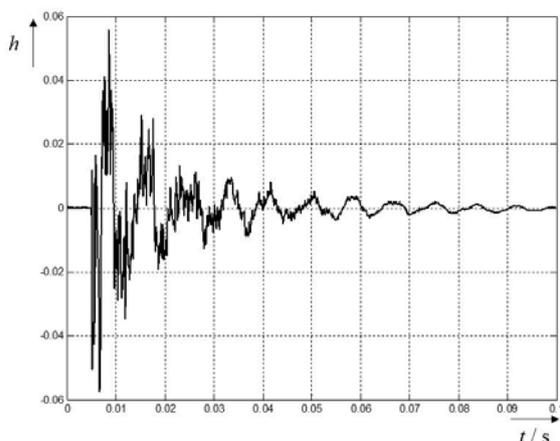


Abbildung 6. Gemessene Primärpfad-Impulsantwort

3.4 Störmodellierung

Auch die genaue Kenntnis der Störsignale entscheidet über den Einsatz eines ANC-Systems. Einerseits haben die signaltheoretischen Eigenschaften (z.B. Frequenzbereiche, schmalbandig/breitbandig, Stationarität) Einfluss auf Auswahl und Parametrierung der Algorithmen. Andererseits ist die Untersuchung und Auswahl einer geeigneten (aber kostengünstigen) Referenzsensorik essentiell für eine Feedforward ANC-Lösung. Die hier vorgestellte Anwendungsklasse wurde exemplarisch weiter eingeschränkt und für die Störmodellierung als externe Quelle das Häckselwerk eines Feldhäckslers ausgewählt. Während rein tonale Komponenten mit fester Frequenz

mit ANC-Verfahren schnelle Erfolge hinsichtlich Ihrer Unterdrückung versprechen, stellen die hier untersuchten Signale eine gewisse Herausforderung dar. Diese weisen neben bandbegrenzten Rauschkomponenten tonale Komponenten auf, die sich eindeutig aus dem nicht drehzahlstarren Häckselprozess ergeben. Bei diesem wird die Häckseltrommel mit einer variablen Grunddrehzahl angetrieben, die während des Häckselprozesses z.B. aufgrund der Menge des Häckselgutes noch schwanken kann.

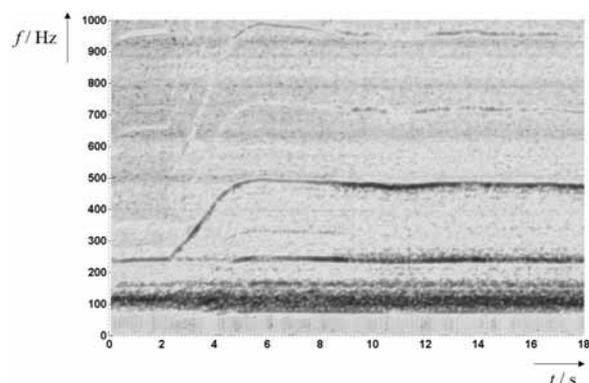


Abbildung 7. Spektrogramm realer Störsignale, Grauwert entspricht Signalstärke

Die Häckseltrommel selbst verfügt über Häckselmesser, die je nach Häckselgut auch unterschiedlich bestückt sein können (z.B. 6, 9 oder 18 Messer auf dem Umfang). Somit lässt sich ein Störmodell entwickeln, das aus einer gegebenen aber veränderlichen Drehzahl ein Signal generiert, das neben stationären bandbegrenzten Rauschkomponenten weitere drehzahlabhängige tonale Komponenten erzeugt, die sich aus einer vorgegebenen Messerbestückung errechnen lassen. Vorteilhaft ist hier, dass sich damit aufgrund der eindeutigen Beziehung das Drehzahlssignal als Ausgangsbasis für ein im Algorithmus benötigtes Referenzsignal eignet.

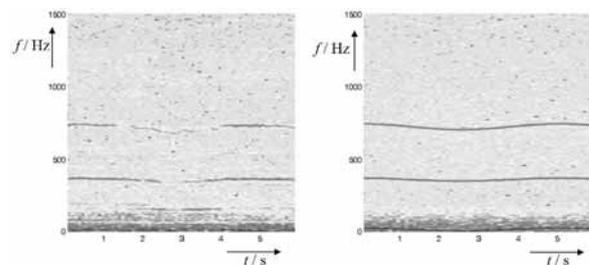


Abbildung 8. Vergleich Spektrogramm realer Störsignale (links) und simulierter Störsignale (rechts), Grauwert entspricht Signalstärke

3.5 Testmethodik RCP-MiL

Wie auch in [6] vorgeschlagen, lehnt sich die gewählte Methodik an die VDI2206 an [7]. Als sehr hilfreich hat sich hier eine Kombination aus realem Versuch mittels *Rapid Control Prototyping* (RCP) und virtuellem Versuch mittels *Model in the Loop* (MiL) erwiesen. Der in 3.1 beschriebene Anwendungsdemonstrator verfügt dazu über ein dSPACE-RCP-System, auf dem die unter MATLAB/Simulink entwickelten Algorithmenmodelle (3.2) nach automatischer Codegenerierung in Echtzeit ausgeführt und so in der realen Umgebung getestet werden können. Durch Erstellung der Modelle für die Übertragungspfade (3.3) und die Störsignale (3.4) lassen sich zusätzlich virtuelle Systemtests durchführen, da diese ebenfalls in MATLAB/Simulink aufgebaut wurden und somit direkt in Kombination mit den Algorithmenmodellen für die MiL-Simulation verwendet werden können. Die Festlegung eines modularen Aufbaus, eines einheitlichen Schnittstellenkonzepts und die Verwendung konfigurierbarer Subsysteme [8] erlauben dabei einerseits einen flexiblen Wechsel zwischen MiL- und RCP-Versuchen. Erkenntnisse aus dem RCP-Versuch können so zu Anpassungen und Erweiterungen der Störsignal- und Pfadmodelle führen, Parametrierungen und neue Variationen der Algorithmen können dagegen zunächst gefahrlos virtuell getestet werden. Andererseits lassen sich durch die konfigurierbaren Subsysteme sehr einfach die jeweiligen Detaillierungsgrade der Teilmodelle der Übertragungspfade sowie die zu verwendenden Adaptionalgorithmen variieren.

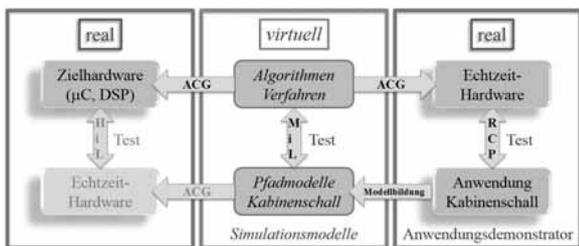


Abbildung 9. Übersicht Entwicklungsmethodik
ACG: automatische Codegenerierung, MiL: Model in the loop, HiL: Hardware in the loop, RCP: Rapid Control Prototyping

Bei der simulationsgestützten Auslegung eines ANC-Systems sind die Auswahl des Algorithmus selbst, die Auslegung der zugehörigen Parameter (z.B. Betriebsrate, Filterlängen, Schrittweite der Adaption), sowie Stabilität und Konvergenz der Adaptionalgorithmen von Interesse. Dazu kann in einer frühen Phase zu-

nächst noch mit den sehr einfachen Modellen der niedrigsten Detaillierungsstufe für die Pfadkomponenten gearbeitet werden, mit denen Laufzeiten und Amplitudenwirkungen (wesentliche Wandlungseffekte) erfasst werden. Da eine Adaption das Kompensationsfilter $W(z)$ so verändert, dass es inklusive Sekundärpfad den Primärpfad phasengedreht nachbildet, ist die Adaption neben dem Primärpfad auch von den Eigenschaften des Störsignals selbst abhängig. Enthält bspw. das Störsignal in einem Frequenzband keine Leistung, so kann in diesem Band keine Adaption erfolgen. Taucht nun in diesem Frequenzband plötzlich ein Störsignal auf, so wird durch die Adaption erst die Übertragungsfunktion in diesem Band angepasst. Die Dynamik dieser Anpassung sowie das generelle Nachführverhalten bei sich ändernden Störsignalen hängen hierbei natürlich von der Schrittweite der Adaptionalgorithmen ab. Durch Modellierung der Störsignale lässt sich das Nachführverhalten virtuell testen.

Während in einer frühen Phase noch mit leistungsfähigen Komponenten für Lautsprecher und Mikrofone gearbeitet werden kann, ist für eine kostengünstige Gesamtlösung die Auswahl und Beurteilung kostengünstiger Systemkomponenten zu betrachten. Auch die Einschränkungen durch Nichtlinearitäten im Signalpfad gilt es zu beurteilen. Mit Hilfe der Teilmodelle unterschiedlichen Detaillierungsgrades lassen sich die Leistungseinbußen simulieren und beurteilen. Dynamisch auftretende Einschränkungen durch Nichtlinearitäten (Sättigungsverhalten der AD/DA-Umsetzer, Übersteuerung der Verstärker) werden überwacht und dem Benutzer mitgeteilt.

4 Ergebnisse

Im genannten Projekt hat sich eine Einkanallösung als praktikabel erwiesen, wenn dabei der Kompensationsschall gleichzeitig über zwei Aktuatoren in Form von kostengünstigen Lautsprechern in den rückwärtigen Eckwinkeln oben in der Kabine abgestrahlt wird. Hieraus resultiert aufgrund der Symmetrie eine optimale Schallführung zum Bedienerarbeitsplatz von beiden Seiten. Hinter dem Sitzplatz dient ein einfaches Elektretmikrofon als Fehlersensor, so dass die verwendeten sensorischen und aktuatorischen Komponenten der Zielsetzung einer kostengünstigen Systemlösung entsprechen.

Die Verwendung einer Einkanallösung beschränkt dabei den notwendigen Rechenaufwand zunächst auf ein Minimum. Bei der Wahl der

Adaptionsalgorithmen haben sich FxNLMS-Verfahren als robust erwiesen, da der im realen Versuchsaufbau auftretende Sekundärpfad nicht zu vernachlässigen ist (FxNLMS) und Schwankungen der Störschallleistung berücksichtigt werden können (FxNLMS). Die generelle Leistungsfähigkeit der Störschallkompensation ist erwartungsgemäß stark abhängig von der Art der Störschallsignale. So lassen sich für ideale Sinussignale hohe Absenkungen erzielen, die aber nicht der realen Situation entsprechen. Reale Störsignale eines Feldhäckslers weisen (Abschnitt 3.4) drehzahlsynchrone aber variable Frequenzkomponenten auf, die von breitbandigen Rauschkomponenten überlagert sind. Für diese Signale ist in Abbildung 10 das Spektrum des in der Kabine aufgenommenen Fehlersignals jeweils über einen Zeitraum von 3 Sekunden mit und ohne die Aktivierung des ANC-Verfahrens dargestellt. Deutlich erkennbar sind Anteile im Bereich 100-500Hz, die sich mit dem untersuchten Verfahren wirksam reduzieren lassen. Eine gehörrichtige Messung ergibt hierbei Werte von durchschnittlich 15-20 dB(A) Absenkung des A-bewerteten Schalldruckpegels.

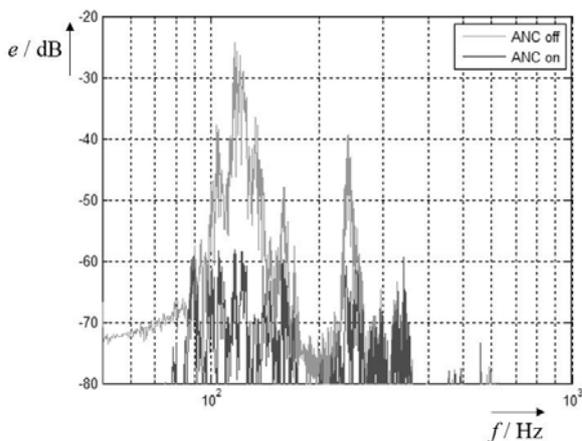


Abbildung 10. Spektrum des Fehlersignals in der Kabine mit aktivierter (dunkel) und deaktivierter (hell) ANC-Funktion

Aufgrund der zeitlichen Varianz der Störschallsignale ist eine einfache spektrale Darstellung nur bedingt geeignet. Mit Hilfe eines Spektrogramms lässt sich aber die zeitliche Veränderung analysieren. Bei der in Abbildung 11 dargestellten Analyse ist der Schalldruckpegel in der Kabine im Zeit- und Frequenzbereich aufgelöst. Über einen Zeitraum von 25 Sekunden ist dabei das ANC-Verfahren manuell insgesamt drei Mal für jeweils einige Sekunden

aktiviert worden. Deutlich erkennbar sind hier neben den dominierenden Komponenten im Frequenzband 100-200 Hz zahlreiche drehzahlvariable Anteile. In den Phasen der drei Aktivierungen adaptiert der ANC-Algorithmus und lässt eine deutlich erkennbare Reduktion der Störsignalkomponenten zu.

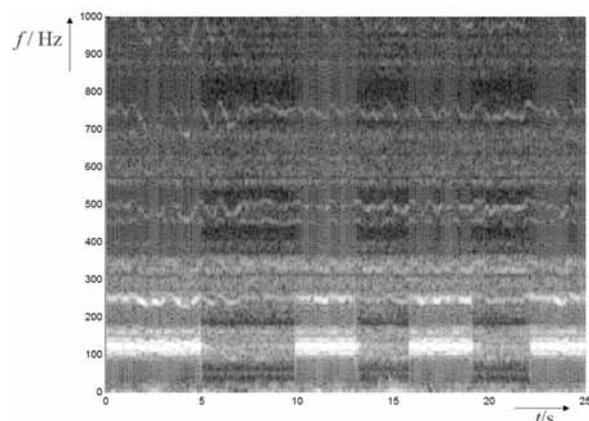


Abbildung 11. Spektrogramm des Schalldruckpegels in der Kabine mit mehrfach abwechselnd deaktivierter und aktivierter ANC-Funktion (Helligkeit entspricht Schalldruckpegel)

Hinsichtlich der untersuchten Methodik ergeben sich auch einige Feststellungen. Die Verwendung von virtuellen Systemtests bringt zahlreiche Erkenntnisse mit sich. So lassen sich durch Verwendung der einfachsten Modelle der Übertragungspfade wesentliche Grundparametrierungen der verwendeten Algorithmen (zeitliche Länge der Filter, Wahl der Betriebsrate) systematisch erarbeiten. Gerade die Wahl der Schrittweiten der adaptiven Filter für Sekundärpfadschätzung und Kompensation bedarf zahlreicher Untersuchungen, da sich hierbei ein Kompromiss aus Nachführverhalten, Adaptionsgüte und Konvergenz ergibt. Hierbei hat sich insbesondere die Verwendung von detaillierten Teilmodellen als hilfreich erwiesen, da eine ausbleibende Konvergenz der Adaption nicht nur auf eine zu große Schrittweite sondern auch auf ein temporäres Auftreten von nichtlinearen Effekten in den Übertragungspfaden (Übersteuerung der DA-Umsetzer, Verstärker, Lautsprecher) zurückgeführt werden kann.

Während in einem ersten Schritt die Teilmodelle an die Gegebenheiten des Anwendungsdemonstrators angepasst werden, ist für einen weiteren Schritt die Simulation von Leistungseinschränkungen bei der Verwendung kostengünstiger Komponenten von

Bedeutung. Hierbei lassen sich die genannten Anpassungen zur Aufwandsreduktion der Algorithmen, die zur Implementierung auf einem kostengünstigen Mikroprozessorsystem (Digitaler Signal Prozessor - DSP) vorbereitet werden, sowohl im virtuellen MiL-Test als auch im realen RCP-Test untersuchen. Aber auch die Verwendung von kostengünstigen Systemkomponenten lässt sich in der Simulation beurteilen. So kann bspw. die Verwendung zweier möglicher Lautsprecher durch Austausch der sog. ThieleSmall-Parameter in der Simulation untersucht werden. Wie in Abbildung 12 anhand des simulierten Schalldrucks zu sehen ist, ergibt sich aufgrund der Leistungseinschränkungen des einen Lautsprechers überhaupt keine Kompensationswirkung.

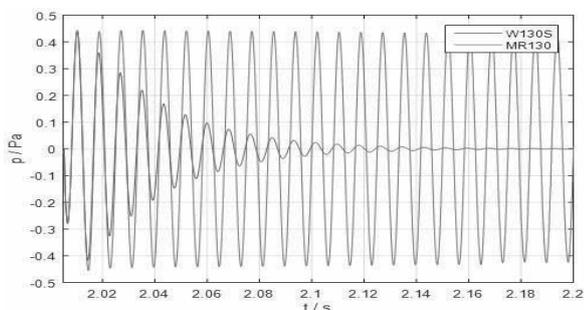


Abbildung 12. Simulation des resultierenden Schalldrucks bei Verwendung eines geeigneten und eines ungeeigneten Lautsprechers

5 Zusammenfassung und Ausblick

Die weitere technische Nutzbarmachung von ActiveNoiseControl-Systemen zur Schallkompensation hängt von drei wesentlichen in einander greifenden Faktoren ab:

1. zur Verfügung stehende Rechenleistung eingebetteter Systeme
2. Verwendung kostengünstiger Systemkomponenten
3. Systemverständnis der komplexen Wechselwirkungen

Die hier dargestellten Arbeiten liefern hierzu einen integrierten Ansatz. Durch eine umfassende Modellbildung der notwendigen technischen Komponenten (Sensoren, Aktuatoren, Signal- und Leistungskomponenten), der elektroakustischen Transferpfade und Störsignale, sowie der in Frage kommenden aufwandsarmen ANC-Verfahren lässt sich Systemver-

ständnis sowohl im virtuellen MiL- als auch realen RCP-Versuch aufbauen. Durch eine flexible Austauschbarkeit von Komponentenmodellen unterschiedlichen Detaillierungsgrades, die zusätzlich auch mögliche aus der Forderung nach Kostenreduktion resultierende Leistungseinschränkungen berücksichtigen, lässt sich schon frühzeitig analysieren, ob für den untersuchten Anwendungsfall eine kostengünstige Lösung absehbar ist.

Für den hier dargestellten Anwendungsfall „Kabinenschall“ lassen sich damit hinsichtlich der Reduktion realer Störschallsignale praxistaugliche Ergebnisse erzielen.

Weitergehende Untersuchungen verfolgen im Wesentlichen die Zielsetzung, die genannten Verfahren weiter in Richtung einer eingebetteten DSP-Lösung zu entwickeln. Dazu gehört u.A. die Modellierung von Festkommaverhalten für die Algorithmen, eine Modifikation, die sich in der gewählten Modellumgebung weiterhin testen lässt.

Darüber hinaus laufen Untersuchungen zur Modellierung der akustischen Pfade. Hier ist insbesondere von Interesse, FEM-Berechnungen zur Generierung von dynamischen Ersatzmodellen der akustischen Pfade durchzuführen.

Letztendlich besteht ebenfalls die Aufgabe, das Wissen und die Methodik auf neue Anwendungsklassen zu erweitern. Hier laufen erste Arbeiten zu Realisierung von auf ANC-Basis basierenden industriellen Gehörschutzlösungen.

Danksagung

Der Autor möchte Dennis Kaupmann und Tobias Lehmann für die im genannten Projekt geleisteten Beiträge Dank sagen.

Förderhinweis

Diese Arbeiten entstammen dem vom BMWi und ZIM geförderten „Low-Cost Mechatronic Systems“ Forschungsprojekt an der Fachhochschule Bielefeld.

Gefördert durch:



Bundesministerium
für Wirtschaft
und Energie

aufgrund eines Beschlusses
des Deutschen Bundestages



6 Referenzen

- [1] G. Neubauer und A. Gmeiner, „Volkswirtschaftliche Bedeutung von Hörschäden und Möglichkeiten zur Reduktion deren Folgekosten,“ *Studie des Instituts für Gesundheitsökonomik München*, August 2011
- [2] D. Guicking, „An overview of ASVC: from laboratory curiosity to commercial products,“ *Active Sound and Vibration Control: theory and applications*, pp. 3-24, 2002
- [3] S. M. Kuo und D. Morgan, „Active Noise Control: A Tutorial Review,“ *Proceedings of the IEEE*, Bd. 87, Nr. 6, pp. 943-973, Juni 1999
- [4] R. Naumann: „Theoretische und experimentelle Untersuchungen der Vertikaldynamik eines elektrischen Rollstuhls mit dem Ziel der Optimierung des Fahrverhaltens durch semi-aktive Dämpfer“, Workshop der ASIM/GI-Fachgruppen STS und GMMS, Reutlingen, 20.-21.2.2014
- [5] Stephen Elliot. *Signal Processing for Active Control*. Academic Press, 2001
- [6] T. Kletschkowski. *Adaptive Feed-Forward Control of LowFrequency Interior Noise*. Springer, 2012.
- [7] V.-R. 2206, *Entwicklungsmethodik für mechatronische Systeme*, Berlin: Beuth Verlag, 2004
- [8] Kaupmann, D., Lehmann, T. Waßmuth, J.: *Methodische Entwicklung kostengünstiger Störschallkompensationssysteme*, Tagungsband VDI Mechatronik 2015, S. 295-300, 12.-13.3.2015, Dortmund

Sitzung 9

Raum 105

Modellbildung / Regelung II

(Sitzungsleitung: Arnold Lange)

*Simulationsgestützte Bewertung kritischer Einflussgrößen auf das
Vorderradblockieren bei Elektrofahrzeugen*

Oliver Maier

Hochschule Pforzheim

Ein Ansatz zur datenbasierten Reglerparametrierung

Alexander Schöley

Universität Rostock

*Anwendung Subspace basierter Identifikationsverfahren auf
gesteuerte Mehrkörpersysteme*

Mathias Marquardt

Hochschule Wismar

Simulationsgestützte Bewertung kritischer Einflussgrößen auf das Vorderradblockieren bei Elektrofahrrädern

Oliver Maier¹, Benedikt Györfi¹, Andreas Kubatschek¹,
Prof. Dr.-Ing. Martin Pfeiffer², Prof. Jürgen Wrede³

¹Hochschule Pforzheim, Institut für Angewandte Forschung - IAF, 75175 Pforzheim

²Hochschule Pforzheim, Fakultät für Technik, Bereich Informationstechnik, 75175 Pforzheim

³Hochschule Pforzheim, Fakultät für Technik, Bereich Maschinenbau, 75175 Pforzheim

oliver.maier@hs-pforzheim.de

Das vorliegende Manuskript konzentriert sich auf die Bewertung kritischer Einflussgrößen für die Entwicklung der Funktion „Vorderradblockieren verhindern“ eines bremsdynamischen Assistenzsystems für Elektrofahrräder. Hierfür werden die wichtigsten Grundlagen und charakteristischen Gleichungen der Bremsdynamik dargelegt sowie das aufgebaute gleichungsbasierte Dreikörper-Simulationsmodell skizziert. Nachfolgend steht die Herausforderung der Parametrierung, bedingt durch den geringen wissenschaftlichen Forschungsstand bei Fahrrädern, im Fokus. Dabei werden sowohl die Ergebnisse einer Literaturrecherche resümiert als auch umgesetzte Methoden der experimentellen Parameterbestimmung vorgestellt. Die Ergebnisse resultieren aus mit dem Simulationsmodell durchgeführten Sensitivitätsanalysen über diverse Einflussgrößen wie Zustandsgrößen, Regelstrecken- und Umweltparameter. Einerseits zeigt sich die dominierende Rolle des Fahrers für das bremsdynamische Verhalten, andererseits sind Bremsdruck, Bremsenkennwert und Zustand der Fahrbahnoberfläche die wichtigsten während der Fahrt veränderlichen Einflussgrößen auf das Vorderradblockieren.

1 Einleitung

Die Leichtigkeit der Fortbewegung bei der Fahrt mit einem durch elektrischen Hilfsantrieb unterstützten Fahrrad (engl.: Electrically Power Assisted Cycle, kurz: EPAC) begeistert. So ist die Gesamtzahl der EPACs in Deutschland auf 2,1 Mio. angewachsen. Mittelfristig werden EPACs einen Marktanteil von 15% im gesamten Fahrradmarkt besitzen [1]. Neben der Entlastung des Fahrers beim Beschleunigen des Fahrrads durch den elektrischen Hilfsantrieb bietet die verfügbare elektrische Energie ebenfalls die Möglichkeit, den Fahrer beim Verzögern des Fahrrads zu unterstützen. Genau an diesen Sachverhalt knüpft das interdisziplinäre Forschungsprojekt BikeSafe mit der Entwicklung eines bremsdynamischen Assistenzsystems (engl.: Braking Dynamics Assistance System, kurz: BDA) für Fahrräder mit hydraulischen Bremsen an. Ziel ist es, die beiden bei Einspurfahrzeugen besonders kritischen Fahrsituationen Vorderradblockieren und Überschlagen über den Lenker zu verhindern.

Die Bestimmung und Quantifizierung der Anforderungen, als Ausgangspunkt der Entwicklung des BDAs, erfordert sowohl ein tiefgreifendes Verständnis der Bremsdynamik von Fahrrädern als auch die Kenntnis des Einflusses kritischer Parameter der Regelstrecke

und Umwelt. Zu diesem Zweck werden im Rahmen von BikeSafe die beiden angesprochenen Fahrsituationen analytisch, virtuell und real untersucht. Erkenntnisse aus der Entwicklung von bereits am Markt befindlichen fahrdynamischen Sicherheitssystemen bei Einspurfahrzeugen, wie bspw. dem ABS bei Krafträdern, sind bezüglich der physikalischen Beschreibung und mathematischen Modellbildung der Bremsdynamik nutzbar. Allerdings zeigen abgeschlossene Studien wie bspw. [2] hinsichtlich der Parameter im Vergleich von Fahrrad und Motorrad erhebliche Unterschiede. Das vorliegende Manuskript untersucht drei für die Entwicklung des BDAs und der Funktion „Vorderradblockieren verhindern“ wesentliche Forschungsfragen:

1. Unter welchen Randbedingungen tritt bei der Bremsung Vorderradblockieren, Hinterradabheben oder keine kritische Fahrsituation auf?
2. Welche Zeit steht für den Regelvorgang zur Verfügung bevor das Vorderrad kritische Schlupfwerte erreicht?
3. Welche Informationen benötigt ein Fahrzustandsbeobachter wie er bspw. aus dem ESP-Bremschlupfregler bekannt ist?

2 Grundlagen

2.1 Fahrradtyp Trekkingrad

In der Modellgruppeneinteilung aller Fahrräder für Deutschland im Jahr 2014 des Zweirad-Industrie-Verbands (kurz: ZIV) dominiert der Typ Trekkingrad mit 32,5% Anteil deutlich [3]. Unter der Annahme der gleichen Modellgruppeneinteilung für EPACs liegt der Fokus im Projekt BikeSafe ausschließlich auf diesem Fahrradtyp.

Die Typisierung von Fahrrädern erfolgt nach dem Haupteinsatzzweck. Für Trekkingräder liegt dieser gleichermaßen im Alltag wie für Touren und Reisen auf befestigten und unbefestigten Wegen. Charakteristisch ist eine Sitzposition mit einer Neigung des Oberkörpers um ca. 45° nach vorne. Das Gewicht des Fahrers verteilt sich zu 50% auf Sattel und Lenker sowie zu 50% auf die Pedale. Bei den Rahmenformen dominieren der klassische Diamantrahmen für Herren-, der Trapezrahmen für Damenmodelle. Der überwiegende Teil der Trekkingräder ist mit einer Tauchfederung an der Vorderradgabel ausgerüstet und verfügt über hydraulische Scheibenbremsen. Der Fahrbahnkontakt wird durch 28 Zoll Drahtspeichenräder unter Verwendung von Naturstraßen-Radialreifen hergestellt. [4, 5]

2.2 Regelungstechnische Betrachtung

Das Führen des Fahrrads, als ein physikalisch prinzipiell instabiles Einspurfahrzeug, findet in einem geschlossenen Regelkreis statt. Bestandteile dieses Regelkreises sind der Fahrer (Regler), welcher sich selbst auf dem Fahrrad (Regelstrecke) unter dem Einfluss der Umwelt (Störgrößenquelle) stabilisiert. Der Fahrer ist dabei gleichzeitig Teil der Regelstrecke und Regler. Beim Übergang der Regelstrecke in den instabilen Bereich, also dem Überschreiten der physikalischen Grenzbedingungen, wie es bei den beiden kritischen Fahrsituationen Vorderradblockieren und Überschlag der Fall ist, gelangt der Fahrer als Regler an seine Grenzen [6].

Hier helfen Fahrerassistenzsysteme wie bspw. das BDA, welches die Regelaufgabe übernimmt, wenn eine der beiden kritischen Fahrsituationen auftritt (vgl. Abbildung 1). Der in diesem Zustand als Regler ausgeschaltete Fahrer kann jedoch durch Loslassen des Bremshebels die Regelung des BDAs abbrechen. Tiefergehende Informationen zum Funktionsprinzip des BDAs finden sich in [7].

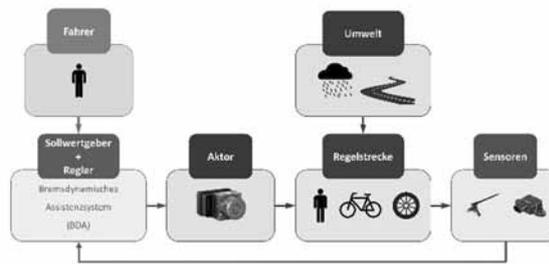


Abbildung 1. Regelkreis bei aktivem BDA

2.3 Bremsdynamik

Beide angesprochenen Fahrsituationen treten vorrangig bei einer Verzögerungsbremung auf, d.h. Verringerung der Geschwindigkeit des Fahrzeugs bis gegebenenfalls zum Stillstand durch die Reibungsbremsen [8]. Um die während einer Verzögerungsbremung wirkenden Kräfte zu betrachten, wird die Regelstrecke Fahrrad und Fahrer nach den Gesetzen der Technischen Mechanik analysiert.

Dabei wird Fahrrad und Fahrer als ein Starrkörper zusammengefasst und die komplexe Querdynamik sowie der Einfluss aller drehenden Komponenten (bspw. Laufräder, Kettentrieb) am Fahrrad vernachlässigt. Abbildung 2 zeigt das Freikörperbild nach dem Prinzip von d'Alembert, Gleichungen (1), (2) und (3) beschreiben die Längs-, Vertikal- und Nickbewegung.

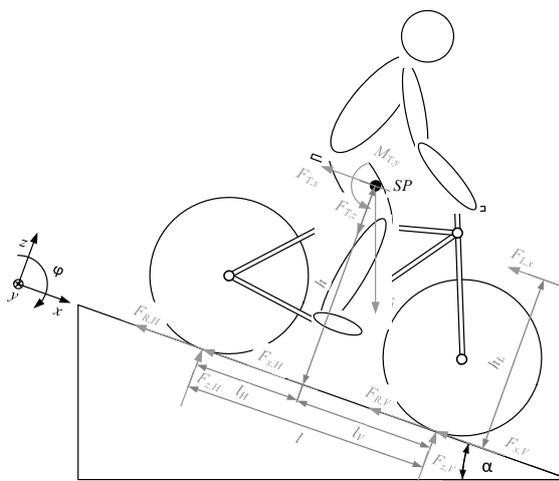


Abbildung 2. Freikörperbild Fahrrad und Fahrer

$$-F_{x,V} - F_{x,H} - F_{R,V} - F_{R,H} + G \cdot \sin \alpha - F_{L,x} - F_{T,x} = 0 \quad (1)$$

$$F_{z,V} + F_{z,H} - G \cdot \cos \alpha - F_{T,z} = 0 \quad (2)$$

$$F_{L,x} \cdot (h - h_L) + (F_{x,V} + F_{x,H} + F_{R,V} + F_{R,H}) \cdot h - F_{z,V} \cdot l_V + F_{z,H} \cdot l_H - M_{T,y} = 0 \quad (3)$$

Entscheidend für die Bremsdynamik sind die Radaufstandskräfte am Vorderrad $F_{z,V}$ und Hinterrad $F_{z,H}$, die zwischen den Reifen und Fahrbahn wirkenden Umfangskräfte $F_{x,V}$ und $F_{x,H}$ und die beim Bremsen im Schwerpunkt des Systems Fahrrad und Fahrer angreifende Trägheitskraft $F_{T,x} = m \cdot \ddot{x}$. Letztere verursacht mit der Luftwiderstandskraft $F_{L,x} = \frac{1}{2} \cdot \rho \cdot c_{w,x} \cdot A \cdot \dot{x}^2$ eine Veränderung der Radaufstandskräfte, welche als dynamische Radlastverlagerung $F_{z,dyn}$ bezeichnet wird. Bei einer starken Verzögerungsbremung bewirkt diese Radlastverlagerung das Hinterradabheben ($F_{z,H} = 0$), kennzeichnet also den Beginn der Überschlagbewegung.

Im Fokus des vorliegenden Manuskripts liegt die kritische Fahrsituation Vorderradblockieren. Für diese ist die Radaufstandskraft am Vorderrad $F_{z,V}$, welche sich unter den Annahmen des Bodenkontakts beider Räder und dem Angriffspunkt der Luftwiderstandskraft im Schwerpunkt nach (4) berechnet, maßgeblich.

$$F_{z,V} = F_{z,V,stat} - F_{z,V,dyn} = \frac{m \cdot g}{l} \cdot (l_H \cdot \cos \alpha + h \cdot \sin \alpha) - \frac{h}{l} \cdot \left(m \cdot \ddot{x} + \frac{1}{2} \cdot \rho \cdot c_{w,x} \cdot A \cdot \dot{x}^2 \right) \quad (4)$$

Tabelle 1 fasst Einflussparameter auf die Radaufstandskraft am Vorderrad zusammen. Aufgrund der Dominanz des Fahrers im Vergleich zum Fahrrad hinsichtlich geometrischer und physikalischer Parameter (hellgrau hinterlegt) zeigt sich dessen wichtige Rolle für die Eigenschaften der Regelstrecke.

2.4 Laufradeigenschaften

Drahtspeichenräder bestehen aus Nabe, Speichen, Felge und Reifen. Letzterer spielt durch seinen direkten Kontakt mit der Fahrbahn bezüglich des Bremsverhaltens eine wichtige Rolle. Aufgrund der bei Fahrrädern sehr kleinen Radaufstandsflächen (Latsch), ist der Druck pro Flächeneinheit Reifengummi hoch, wodurch der Reifengummi aufgrund seiner Elastizität teilweise die Unebenheiten der Fahrbahn umschließt.

System	Parameter	Symbol
Regelstrecke	Masse	m
	Abstand Schwerpunkt zu Hinterrad	l_H
	Höhe Schwerpunkt zu Fahrbahn	h
	Luftwiderstandsbeiwert	$c_{w,x}$
	Querspanfläche	A
	Radstand	l
Umwelt	Steigung Fahrbahn	α

Tabelle 1. Parameter Radaufstandskraft Vorderrad

Fahrradreifen beziehen folglich ihren „Grip“ nicht allein über Kraft-, sondern zu einem gewissen Teil auch durch Formschluss [4]. Im vorliegenden Manuskript werden die Annahmen getroffen, bei Trekkingradreifen überwiegt der Kraftschlussanteil (geringe Einsenkung, kaum Profiltiefe) und die Radwiderstände sind vernachlässigbar (wenig Formänderungsarbeit).

Der Kraftschluss μ zwischen Reifen und Fahrbahn als eine Funktion des Schlupfs S ergibt sich nach (5). Für den Schlupf im Bremsfall S_{Brems} gilt (6). Den größten Einfluss auf diese Kurve hat der Zustand der Fahrbahnoberfläche (bspw. trocken/ nass) [8].

$$\mu_V(S_V) = \frac{F_{x,V}}{F_{z,V}} \quad (5)$$

$$S_{V,Brems} = \frac{\dot{\varphi}_0 - \dot{\varphi}_V}{\dot{\varphi}_0} \quad (6)$$

Die für die Berechnung des Schlupfs notwendige Winkelgeschwindigkeit des momentenbeaufschlagten Rades $\dot{\varphi}$ ergibt sich aus der Anwendung des Drallsatzes um den Radmittelpunkt (7).

$$\int \frac{M_V - F_{x,V} \cdot r_{V,dyn}}{J_V} = \dot{\varphi}_V \quad (7)$$

Für die Untersuchung der kritischen Fahrsituationen sind nur negative Vorderradmomente M_V , welche sich

durch das hydraulische Bremssystem ergeben, relevant. Ausgehend vom Bremsdruck $p_{V,Brems}$ sowie unter der Annahme, dass dynamische Effekte und die Kompressibilität des Fluides vernachlässigbar sind, gilt (8). Tabelle 2 fasst die wichtigsten Parameter hinsichtlich der Kraftübertragung am Vorderrad zusammen.

$$M_V = (p_{V,Brems} \cdot A_{Nehmer} \cdot C_V^*) \cdot r_{V,Reib} \quad (8)$$

System	Parameter	Symbol
Regelstrecke	Dynamischer Rollhalbmesser	$r_{V,dyn}$
	Trägheitsmoment Laufrad	J_V
	Reibhalbmesser Bremse	$r_{V,Reib}$
	Bremsenkennwert	C_V^*
	Fläche Nehmerkolben	A_{Nehmer}
Umwelt	Zustand Fahrbahnoberfläche	-

Tabelle 2. Parameter Kraftübertragung Vorderrad

3 Modellparametrierung

Voraussetzung für die analytische als auch virtuelle Untersuchung der beiden kritischen Fahrsituationen ist es, die Parameter von Regelstrecke und Umwelt systematisch zu erfassen. Neben der korrekten Abbildung der physikalischen Zusammenhänge ist die Qualität der Parameter entscheidend für die Aussagekraft der Untersuchungen. Grundsätzliche Bestandteile der Regelstrecke sind Fahrer und Fahrrad. Aufgrund der Wichtigkeit des Laufrads für den Vorgang des Vorderradblockierens werden dessen Parameter nachfolgend getrennt vom Fahrrad behandelt.

3.1 Gleichungsbasiertes Dreikörpermodell

Zur simulativen Untersuchung der beiden kritischen Fahrsituationen Vorderradblockieren und Überschlag liegt ein gleichungsbasiertes Modell mit dem Freiheitsgrad vier vor. Dieses Modell besteht aus den drei Starrkörpern Aufbau (Rahmen, Gabel und Fahrer),

Vorderrad sowie Hinterrad und stellt die ebene Bewegung der Regelstrecke in einem hinreichenden Detaillierungsgrad dar. Ebenfalls können Einflüsse der Umwelt berücksichtigt werden. Eingänge des Modells sind die Zustandsgrößen Fahrgeschwindigkeit, Bremsdruck und Bremsdruckgradient am Vorderrad. Wichtigste Ausgangsgrößen sind die Drehzahl am Vorderrad, die Längs- und Vertikalbeschleunigung des Aufbaus sowie die Nickrate (siehe [7]). Weitere Informationen zur mathematischen Beschreibung, Modellierung und Umsetzung des Modells finden sich in [9].

3.2 Fahrer

Die erwähnte wichtige Rolle des Fahrers für die Eigenschaften der Regelstrecke ergibt sich aus dessen Abmaßen, Schwerpunktlage, Masse und Trägheitsmomenten relativ zum Fahrrad. Ein elementares Werkzeug zur Ermittlung dieser Parameter des menschlichen Körpers ist die Anthropometrie. Sie unterscheidet beide Geschlechter in 5., 50. und 95. Perzentil Personen [10]. Auf Basis der in den gängigen Normen [10, 11] zur Anthropometrie enthaltenen Abmaße des Körpers sowie der Nutzung von CAD-Software lässt sich die in Tabelle 3 gezeigte Zuordnung der Rahmengrößen h_{Rahmen} von Trekkingrädern und der Querspanflächen A_{Fahrer} zu den Perzentil Personen vornehmen. Der Luftwiderstandsbeiwert $c_{W,x}$ wird aufgrund der einheitlichen Haltung des Fahrers auf Trekkingrädern über die Perzentil-Personen als konstant angenommen.

Perzentil #	m_{Fahrer} kg	h_{Rahmen} m	A_{Fahrer} m ²	$c_{W,x}$ #
m	5	62,28	0,50	0,43
	50	81,43	0,54	0,49
	95	100,08	0,59	0,53
w	5	47,51	0,47	0,33
	50	62,20	0,51	0,37
	95	80,35	0,56	0,43

Tabelle 3. Wertetabelle Parameter Fahrer

Eine besondere Herausforderung besteht in der Ermittlung der Schwerpunktlage und Trägheitsmomente des Fahrers in Abhängigkeit von der Sitzposition auf dem Fahrrad. Eine Literaturrecherche blieb diesbezüglich

ergebnislos. Abhilfe bietet die Berechnung der beiden Größen aus den Schwerpunktlagen, Massen und Trägheitsmomenten der einzelnen Glieder. Die gängigen Normen zur Anthropometrie beinhalten hierzu kaum Informationen. Sehr aussagekräftig ist hingegen eine Erhebung der Raumfahrtbehörde NASA aus dem Jahr 1999, in welcher die benötigten Daten der einzelnen Glieder in experimentellen Untersuchungen an Leichen bestimmt wurden [12].

3.3 Fahrrad

Neben dem Fahrradtyp, welcher im Rahmen des Projekts BikeSafe auf Trekkingräder eingeschränkt ist, besitzt die Rahmengröße großen Einfluss auf die Parameter des Fahrrads und damit auf die Regelstrecke. Die Rahmengröße definiert die Abmaße (bspw. Radstand l), Schwerpunktlage, Masse, Trägheitsmomente und Querspanntfläche des Fahrrads sowie die exakte Sitzposition der jeweiligen Perzentil-Person. Letztere beeinflusst, wie im vorangegangenen Abschnitt angeführt, Schwerpunktlage und Trägheitsmomente des Fahrers.

Zur Bestimmung der fahrradspezifischen Parameter diente neben einer Datenblatt- und Literaturrecherche (Abmaße) sowie einfachen Wiegeversuchen nach [13] (Schwerpunktlage und Masse) ein eigenentwickelter Prüfstand zur experimentellen Bestimmung von Trägheitsmomenten diverser Komponenten (vgl. Abbildung 3).



Abbildung 3. Prüfstand Messung Trägheitsmomente

Zur Messung wird am Prüfstand das Prinzip der Dreh-schwingung genutzt. Dieses beruht auf der Wirkung

der Gravitationskraft auf einen Probenkörper. Die rechnerische Ermittlung des Trägheitsmoments um den Aufhängepunkt (Drehpunkt A) erfolgt durch einen Koeffizientenvergleich der Differentialgleichung der harmonischen Schwingung mit dem linearisierten Drallsatz (9). Das tatsächliche Trägheitsmoment des Probenkörpers um seinen Schwerpunkt J_{SP} resultiert aus einer Rückrechnung, ausgehend vom ermittelten Trägheitsmoments J_A , nach dem Satz von Steiner (10).

$$T = \frac{2\pi}{\omega} = 2\pi \sqrt{\frac{J_A}{m \cdot g \cdot \overline{ASP}}} \quad (9)$$

$$J_{SP} = J_A + m \cdot \overline{ASP}^2 \quad (10)$$

Die Querspanntfläche des Fahrrads A_{Fahrrad} und die genauen Sitzpositionen über die Perzentil-Verteilung wurden unter Zuhilfenahme von CAD-Software bestimmt.

Bei der Auswahl der zur Parameterbestimmung betrachteten Fahrräder wurde die bisherige Einschränkung auf den Fahrradtyp Trekkingrad nochmals verschärft. Berücksichtigung fanden, aufgrund des aktuellen Trends in der Fahrradbranche (bspw. Bosch, Shimano, Panasonic, etc.), nur EPACs mit einem Mittelmotor- oder Tretlagerantriebskonzept. In Summe umfasste die Vermessung 34 Fahrräder. Großen Wert wurde auf eine möglichst gleiche Verteilung der Rahmengrößen und verbauten Komponenten über die Stichprobe gelegt.

Tabelle 4 fasst die Mittelwerte der Parameter Masse m_{Fahrrad} , Radstand l und Querspanntfläche A_{Fahrrad} des Fahrrads zusammen. Hinsichtlich des Luftwiderstandsbeiwerts $c_{w,x}$ gilt die Annahme, dass dieser hauptsächlich durch die Haltung des Fahrers definiert und nicht von der Rahmengröße abhängig ist.

Hydraulische Scheibenbremsen am Fahrrad stellen ein von Kraftfahrzeugen abgeleitetes sehr leistungsfähiges Bremssystem dar. Ist im sportlichen Bereich das aggressive Bremsverhalten durchaus erwünscht, birgt es für den Trekkingradfahrer, vor allem in Schrecksituationen, die Gefahr eines Überbremsens des Vorderrades [14]. Dabei kann eine der beiden mit dem BDA zu verhindernden kritischen Fahrsituationen auftreten.

Aufgrund der Wichtigkeit des Bremssystems wurde zur Parameterbestimmung direkt mit dem Projektpartner Gustav Magenwirth GmbH & Co. KG zusammengearbeitet. Gängigstes Produkt bei Trekkingrädern

ist die Bremse Magura MT4 in Kombination mit der Scheibe Storm Rotor Ø 180 mm am Vorderrad.

Eine Schwierigkeit stellt die Bestimmung des Rollhalbmessers $r_{V,Reib}$ und des Bremskennwerts C_V^* dar. Ersterer verändert sich während eines Bremsvorgangs permanent und ist damit schwer erfassbar [15]. Aus diesem Grund wurde zur Ermittlung des Rollhalbmessers angenommen, dass dieser sich näherungsweise aus dem Abstand des Flächenschwerpunkts des Belags zum Radmittelpunkt ergibt. Der Bremskennwert ist während einer Bremsung stark abhängig von der Temperatur der Scheibe und Beläge, der Winkelgeschwindigkeit des Laufrades und der Flächenpressung auf die Beläge [15]. Für die Bestimmung des Bremskennwerts steht bei der Gustav Magenwirth GmbH ein Präzisions-Schwungmassenprüfstand zur Verfügung. Aufgrund der dargestellten Veränderlichkeit von Rollhalbmesser und Bremskennwert sind in Tabelle 5 jeweils die Mittelwerte angegeben.

Perzentil #	m_{Fahrrad} kg	l m	A_{Fahrrad} m ²
m	5	19,67	1,12
	50	20,02	1,19
	95	20,37	1,26
w	5	17,94	1,11
	50	19,25	1,13
	95	20,10	1,15

Tabelle 4. Wertetabelle Parameter Fahrrad

$r_{V,Reib}$ m	C_V^* #	A_{Nehmer} m ²
0,085	1,1	3,8e-4

Tabelle 5. Wertetabelle Parameter Bremssystem

3.4 Laufrad

Wichtigster Parameter des Laufrades ist die Kurve des Kraftschlusses μ zwischen Reifen und Fahrbahn in Abhängigkeit vom Schlupf S . Zur numerischen Simulation muss diese sogenannte μ -Schlupf-Kurve entweder

als mathematische Funktion beschrieben sein oder anhand bekannter Stützstellen interpoliert werden. Im vorliegenden Manuskript wurde aufgrund fehlender veröffentlichter Messdaten realer μ -Schlupf-Kurven von Trekkingradreifen der Ansatz nach [16] gewählt. Dieser basiert auf der Annäherung der μ -Schlupf-Kurve durch eine Exponentialfunktion. Hierfür müssen Haftbeiwert μ_h , Gleitbeiwert μ_g und der kritische Schlupf S_c bekannt sein (vgl. Tabelle 6)

Zustand Fahrbahnoberfläche	μ_h #	μ_g #	S_c %
trocken	1,0	0,8	10
	0,9	0,8	10
nass	0,8	0,7	15
	0,7	0,6	15

Tabelle 6. Wertetabelle Näherung μ -Schlupf-Kurve

Typisch für Trekkingräder sind 28 Zoll Drahtspeichenräder, woraus sich der statische Rollhalbmesser $r_{V,stat}$ errechnen lässt (vgl. Tabelle 7). Während der Fahrt verändert sich der statische Rollhalbmesser in Abhängigkeit der wirkenden Radaufstandskraft (4) um den Einfederweg des Reifens. Der resultierende Rollhalbmesser wird als dynamischer Rollhalbmesser $r_{V,dyn}$ bezeichnet. Charakteristisch für die Beschreibung des Reifeneinfederwegs sind dessen Federungs- und Dämpfungseigenschaften in radialer Richtung. Gängig ist in diesem Zusammenhang die Annäherung des viskoelastischen Verhaltens mittels des Gehmann-Modells [8]. Aufgrund der weit verbreiteten Tauchfederung an der Vorderradgabel von Trekkingrädern kann der Einfluss der Reifendämpfung vernachlässigt werden und der Einfederweg mit dem Hook'schen Gesetz beschrieben werden (11).

$m_{\text{Laufrad,V}}$ kg	$r_{V,stat}$ m	c_V kN/m	J_V kgm ²
1,97	0,35	200 [17]	0,15 [18]

Tabelle 7. Wertetabelle Parameter Laufrad

$$r_{V,dyn} = r_{V,stat} - \frac{F_{z,V}}{c_V} \quad (11)$$

3.5 Kombinierte Parameter

Zur Beschreibung der bremsdynamischen Grundlagen sind Fahrrad und Fahrer als ein Starrkörper zusammengefasst worden (vgl. Abschnitt 2.3). Folglich ergibt sich ein Teil der zur Untersuchung notwendigen Parameter erst durch Kombination der spezifischen Parameter von Fahrrad und Fahrer. Für die zur Berechnung der Radaufstandskraft am Vorderrad $F_{z,V}$ notwendigen Masse m gilt (12), die Querspanntfläche A ergibt sich nach (13).

$$m = m_{\text{Fahrer}} + m_{\text{Fahrrad}} + m_{\text{Laufrad,V}} + m_{\text{Laufrad,H}} \quad (12)$$

$$\text{mit } m_{\text{Laufrad,H}} = 3,58 \text{ kg}$$

$$A = A_{\text{Fahrer}} + A_{\text{Fahrrad}} \quad (13)$$

Gleiches gilt für die Schwerpunktlage, welche maßgeblich von der Sitzposition des Fahrers und der Rahmengröße des Fahrrads abhängt. Tabelle 8 fasst die Schwerpunktlagen in Abhängigkeit der Perzentil-Personen zusammen.

	Perzentil #	l_H	h
		m	m
m	5	0,51	0,87
	50	0,55	0,92
	95	0,58	0,99
w	5	0,50	0,80
	50	0,51	0,84
	95	0,52	0,91

Tabelle 8. Wertetabelle kombinierte Parameter

4 Simulative Einflussgrößenbewertung

Die nachfolgend dargestellten Ergebnisse resultieren aus mit dem gleichungsbasierten Dreikörpermodell durchgeführten Sensitivitätsanalysen und beantworten die in der Einleitung des Manuskripts gestellten Forschungsfragen.

4.1 Konfiguration der Sensitivitätsanalysen

Ziel der Sensitivitätsanalysen ist die Einflussbewertung von diversen Zustandsgrößen, Regelstrecken- und Umweltparameter auf die kritische Fahrsituation Vorderradblockieren.

Maßgeblich für das Verhalten der Regelstrecke ist der Fahrer. Seine Eigenschaften variieren gemäß der Perzentil-Verteilung. Aufgrund der Zuordnung von Fahrrad und Fahrer durch die Rahmengröße hängen folgende weitere Parameter mit der jeweiligen Perzentil-Person zusammen: Masse m , Schwerpunktlage l_H und h , Luftwiderstandsbeiwert $c_{w,x}$, Querspanntfläche A und Radstand l (vgl. Tabelle 1). Die in die Sensitivitätsanalysen übernommenen Werte entsprechen den Tabellen 3, 4 und 8. Neben dem Einfluss des Fahrers ist für die Untersuchung des Vorderradblockierens nach den Gleichungen (6), (7) und (8) die Zustandsgröße Bremsdruck von äußerster Bedeutung. Der angenommene Maximalwert des Bremsdrucks entstammt realen Messungen.

Die Variation der Zustandsgrößen Fahrgeschwindigkeit und Bremsdruckgradient dient der Untersuchung von Randbedingungen, bei denen Vorderradblockieren, Hinterradabheben oder keine kritische Fahrsituation auftritt. Ebenfalls werden die beiden Zustandsgrößen zur Ermittlung der für einen Regelvorgang zur Verfügung stehenden Zeit, bis das Vorderrad kritische Schlupfwerte erreicht, benötigt. Die berücksichtigten Fahrgeschwindigkeiten orientieren sich an für Elektrofahrräder typischen Werten [19, 20]. Der maximale Bremsdruckgradient resultiert aus realen Messungen.

Der Auswahl weiterer Regelstreckenparameter für die Sensitivitätsanalyse liegt die Überlegung zugrunde, welche Parameter sich während der Fahrt ändern und somit nicht vor Fahrtbeginn appliziert werden können (vgl. Tabelle 2). Die Abhängigkeit der Reifensteifigkeit c_V vom Reifenluftdruck, welche nach (11) eine Veränderung des dynamischen Rollhalbmessers $r_{V,dyn}$ bewirken könnte, findet aufgrund fehlender Werte keine Berücksichtigung. Nach Angaben der Gustav Magenwirth GmbH schwankt der Bremsenkennwert C_V^* fertigungsbedingt um bis zu 10% bezogen auf den Mittelwert aus Tabelle 5. Mögliche sich daraus ergebende Auswirkungen auf die Bremsdynamik werden bei den Sensitivitätsanalysen untersucht.

Wesentliche Einflüsse der Umwelt auf das Vorderradblockieren sind die Steigung α und der Oberflächenzustand der Fahrbahn (vgl. Tabelle 1 und 2). Letzterer wird anhand einer trockenen und nassen Fahrbahn in

jeweils zwei Ausprägungen untersucht (vgl. Tabelle 6). Eine Zunahme der Fahrbahnsteigung (positive Werte entsprechen gemäß Freikörperbild einem Gefälle, vgl. Abbildung 2) bewirkt nach Gleichung (4) eine Zunahme der Radaufstandskraft am Vorderrad, wodurch nach Gleichung (5) eine gegebene Umfangskraft bei kleinerem Reibungskoeffizient übertragen werden kann. Konsequenz ist ein geringerer Bremsschlupf, also ein minimiertes Risiko des Vorderradblockierens. Gegenteilig zeigt sich das Risiko des Hinterradabhebens, welches sich nach [21] mit Zunahme der Fahrbahnsteigung erhöht. Tabelle 9 fasst alle variierten Zustandsgrößen, Regelstrecken- und Umweltparameter zusammen.

4.2 Auftreten kritischer Fahrsituationen (Forschungsfrage 1)

Zur Untersuchung der Randbedingungen, bei denen eine oder keine der beiden kritischen Fahrsituationen auftritt, wurde bei jeder Sensitivitätsanalyse – aufgrund des dargelegten großen Einflusses – die Perzentil-Person und der Bremsdruck variiert. Neben diesen beiden Einflussgrößen ist stets noch eine weitere/r Zustandsgröße, Regelstrecken- oder Umweltparameter gemäß Tabelle 9 verändert worden. Alle übrigen Einflussgrößen blieben konstant und entsprechen den in Tabelle 9 als fett formatierten Werten.

Für die Vergleichbarkeit unterschiedlicher Sensitivitätsanalysen ist eine einheitliche dreidimensionale Ergebnisdarstellung definiert worden. Auf der horizontalen Achse findet sich stets die Perzentil-Person, auf der vertikalen Achse der Bremsdruck und auf der dritten Achse in die Raumtiefe die weitere variierte Einflussgröße. Ein Kreis symbolisiert keine kritische Fahrsituation, ein Kreuz das Hinterradabheben und ein Stern das Vorderradblockieren.

Die Abbildungen 4 und 5 stellen die Sensitivität der kritischen Fahrsituationen von der Fahrgeschwindigkeit und dem Bremsdruckgradient dar. Auffallend ist zunächst das Ausbleiben einer kritischen Fahrsituation bis zu einem Bremsdruck von 40bar über alle Perzentil-Personen.

Die kritische Fahrsituation Vorderradblockieren ist nur bei der 5-Perzentil Frau erkennbar. Aufgrund ihrer geringen Masse bewirkt sie die geringste Radaufstandskraft am Vorderrad, wodurch bei gegebenem Kraftschluss weniger Umfangskraft auf die Fahrbahn übertragbar ist. Nach (7) wird das momentenbeaufschlagte Vorderrad so stärker als bei anderen Perzentil-Personen verzögert und nimmt nach (6) größere Brems-

Einflussgröße	Einheit	Werte					
		m			w		
Perzentil-Person	#	5	50	95	5	50	95
Bremsdruck	bar	20	40	60	80	100	
Fahrgeschwindigkeit	km/h	10	15	20	25	30	
Bremsdruckgradient	bar/s	200	400	600	800	1000	
Bremsen-Kennwert	#	0,99	1,1	<i>1,21</i>			
Gefälle Fahrbahn	°	<i>0</i>	5	10	15	20	
Zustand Fahrbahnoberfläche μ_H / μ_G	#	trocken		nass			
		1,0/ 0,8	0,9/ 0,8	0,8/ 0,7	<i>0,7/ 0,6</i>		

Tabelle 9. Wertetabelle Sensitivitätsanalyse (Fett = Ausgangswerte, Grau = Worst-Case)

schlupfwerte an. Hinterradabheben tritt ab einem Bremsdruck von 80 bzw. 100bar bei allen Perzentil-Personen außer der 5-Perzentil Frau auf.

Am ungefährdetsten bzgl. einer kritischen Fahrsituation ist der 95-Perzentil Mann. Der Argumentation für die 5-Perzentil Frau folgend, lässt sich dies für das Vorderradblockieren leicht nachvollziehen. Zu diskutieren ist dieses Ergebnis hinsichtlich des Hinterradabhebens. In einer statischen Betrachtung nach [22] benötigt der 95-Perzentil Mann aufgrund seiner hohen Schwerpunktlage die geringste Längsverzögerung für das Hinterradabheben aller Perzentil-Personen. Aufgrund seiner hohen Masse erreicht er diese sogenannte kritische Längsverzögerung nach dem zweiten Newton'schen Axiom erst durch eine hohe Umfangskraft am Vorderrad bedingt durch einen hohen Bremsdruck. Risikominimierend für den 95-Perzentil Mann wirkt zudem die Schwerpunktverlagerung während der Bremsung, ermöglicht durch die Tauchfederung an der Gabel. Unter der Annahme einer ähnlichen Federsteifigkeit über die Perzentil-Personen, wie sie auch

der Sensitivitätsanalyse zugrunde lag, bewirkt die dynamische Radlastverlagerung insbesondere eine Reduktion der Schwerpunkthöhe. Konsequenz ist eine Erhöhung der kritischen Längsverzögerung ab der es zu einem Hinterradabheben kommt.

Eine Sensitivität der beiden kritischen Fahrsituationen hinsichtlich der variierten Zustandsgrößen Fahr- geschwindigkeit und Bremsdruckgradient ist aus den Ab- bildungen 4 und 5 nicht erkennbar (Symbol auf der dritten Achse bleibt stets unverändert).

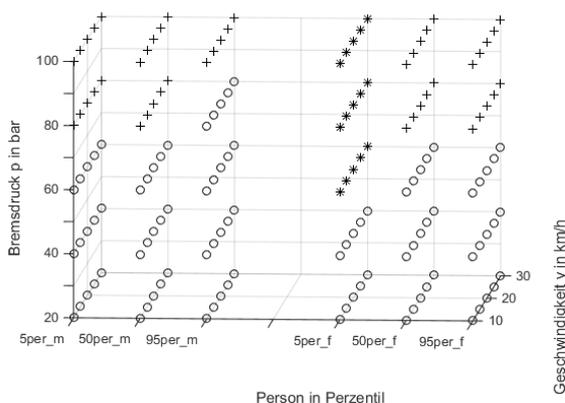


Abbildung 4. Sensitivität Fahrgeschwindigkeit

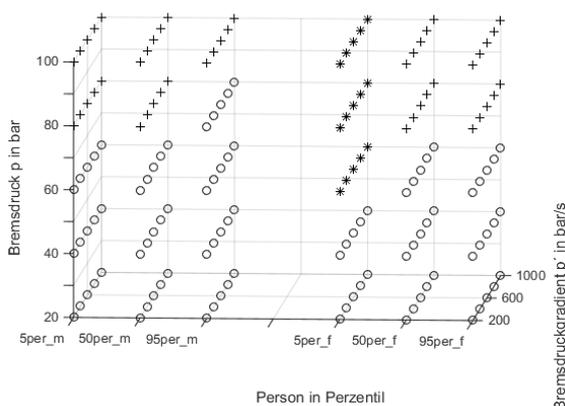


Abbildung 5. Sensitivität Bremsdruckgradient

Abbildung 6 zeigt die Auswirkung eines veränderten Bremskennwerts auf die kritischen Fahrsituationen. Zunächst sind ähnliche Effekte bzgl. des Auftretens von Vorderradblockieren und Hinterradabheben wie bei der Einflussuntersuchung von Fahrgeschwindigkeit und Bremsdruckgradient zu erkennen. Auffallend ist in Abbildung 6 jedoch, dass sich die Symbole auf

der dritten Achse in Abhängigkeit des Bremskennwerts ändern. Eine Sensitivität des Auftretens der beiden kritischen Fahrsituationen vom Bremskennwert ist folglich gegeben. So kann die fertigungsbedingte Schwankung des Bremskennwerts um bis zu 10% in einem ansonsten unveränderten Zustandsgrößen- und Parameterraum zum Auftreten bzw. nicht Auftreten einer kritischen Fahrsituation führen.

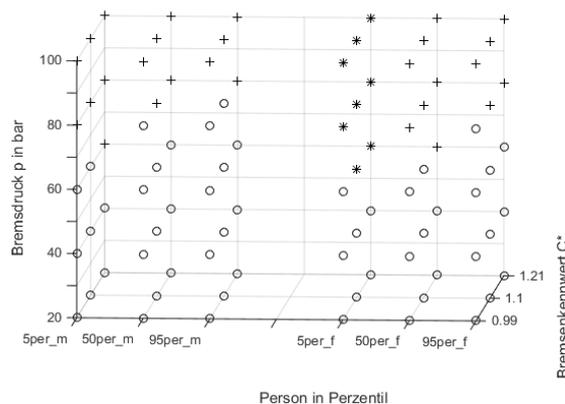


Abbildung 6. Sensitivität Bremskennwert

Sowohl Steigung als auch Oberflächenzustand der Fahrbahn haben maßgeblichen Einfluss auf die kritischen Fahrsituationen. Eine Zunahme der Steigung wirkt sich, wie in Abbildung 7 auf der dritten Achse durch den Übergang des Symbols Kreis zu Kreuz erkennbar, erhöhend auf das Risiko eines Hinterradabhebens auf.

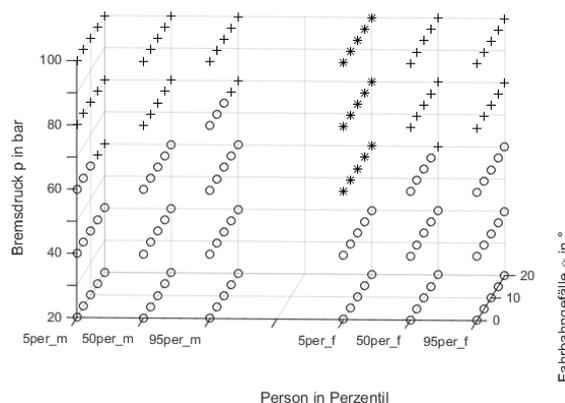


Abbildung 7. Sensitivität Steigung Fahrbahn

Die größte Sensitivität neben der Perzentil-Person und dem Bremsdruck aller variierten Einflussgrößen auf die beiden kritischen Fahrsituationen hat der Zustand der Fahrbahnoberfläche (vgl. Abbildung 8). So tritt bei allen Perzentil-Personen ab unterschiedlichen Bremsdrücken auf regennasser Fahrbahn mit geringem Reibungskoeffizient ($\mu_h = 0,7$) das Vorderradblockieren ein. Je trockener die Fahrbahn ist, also je höher der Reibungskoeffizient zwischen Reifen und Fahrbahn, desto mehr verlagert sich das Risiko des Vorderradbloc-kierens entweder hin zu einer nicht kritischen Fahrsituation oder zum Hinterradabheben. Beispielsweise ist die 5-Perzentil Frau auf trockener und sehr griffiger Fahrbahn ebenfalls durch die kritische Fahrsituation Hinterradabheben gefährdet.

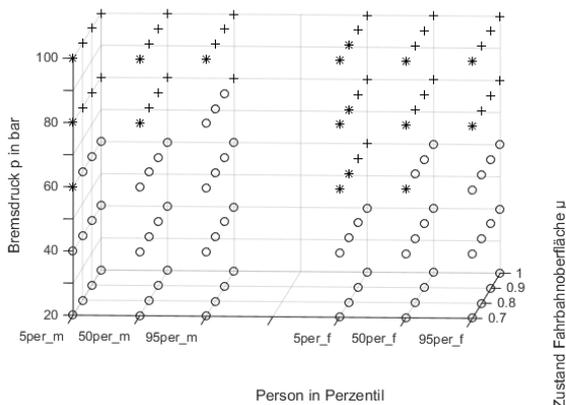


Abbildung 8. Sensitivität Zustand Fahrbahnoberfläche

4.3 Zustandsabhängige Radblockierzeit (Forschungsfrage 2)

Für die Auslegung eines Regelsystems ist die für den Regelvorgang zur Verfügung stehende Zeit entscheidend. Im Falle der Funktion „Vorderradbloc-kieren verhindern“ des BDAs ist dies die minimale Blockierzeit, also die Zeit, welche ein Rad in einem bestimmten Zu-stand braucht, bis es stehenbleibt. Größte Bedeutung für die minimale Blockierzeit hat die Fahrgeschwin-digkeit. Ist sie niedrig, wird die Blockierzeit klein. [23]

Abbildung 9 zeigt die minimale Blockierzeit des Vor-derrades in Abhängigkeit von der Fahrgeschwindigkeit und vom Bremschlupf. Für alle weiteren Zu-standsgrößen, Regelstreckenparameter und Umwelt-parameter ist der ungünstigste Fall hinsichtlich des Vorderradbloc-kierens angenommen (grau markiert in Tabelle 9). Weiterhin ist bei der Untersuchung der mi-nimalen Blockierzeit das Anlegen der Beläge an die

Scheibe mit einer Dauer von ca. 10 ms nicht berück-sichtigt. Dieser Vorgang, welcher die minimale Blo-ckierzeit verlängert, wirkt sich auf die in Abbildung 9 gezeigten Kurven als Nullpunktfehler aus.

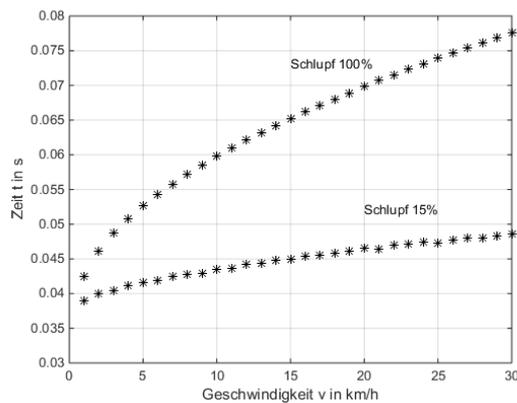


Abbildung 9. Minimale Blockierzeit

4.4 Einsatz Fahrzustandsbeobachter (Forschungsfrage 3)

Kennzeichnend für das Elektronische Stabilitätspro-gramm (kurz: ESP) ist die Fahr-dynamikregelung, wel-che die Fahrzeugbewegung mittels über- und unterla-gerter Regler regelt [24]. Unter anderem kommt dabei ein Bremsschlupfregler zum Einsatz, dessen Potential im Rahmen von BikeSafe zur Darstellung der Funk-tion „Vorderradbloc-kieren verhindern“ überprüft wird. Voraussetzung für den Einsatz des Bremsschlupfreg-lers ist das Vorliegen der aktuellen Aufstandskraft am Vorderrad zur Bestimmung von Sollschlupfwerten am Rad. Die Information der Radaufstandskraft wird bei der Fahr-dynamikregelung des ESPs durch einen Be-obachter geschätzt [24].

Im vorliegenden Manuskript wird untersucht, welche Abweichung sich bei einer Schätzung basierend auf (4) ergibt, wenn statt eines angenommenen und appli-zierten Fahrers eine andere Person mit ihren individu-ellen physikalischen Eigenschaften das EPAC mit appli-ziertem BDA nutzt. Als Ausgangswert wird ange-nommen, das EPAC sei jeweils auf den 50-Perzentil Mann (Herrenmodell) respektive die 50-Perzentil Frau (Damenmodell) eingestellt.

Tabelle 10 fasst die Ergebnisse der Untersuchung zu-sammen. Für die Schätzung der statischen Radlast $F_{z,V,stat}$ ist eine Abweichung von größer 20 % erkenn-bar. Unter der Annahme einer typischen Verzögerung

bei einer Vorderradbremung auf trockener Straße für Fahrräder von $\dot{x} = -3,4 \frac{\text{m}}{\text{s}^2}$ [25] und einer EPAC charakteristischen Fahrgeschwindigkeit von $\dot{x} = 17,4 \frac{\text{m}}{\text{s}}$ [19] ergibt sich ein zusätzlicher Schätzfehler von nochmals über 20 %. Eine Schätzung der Radaufstandskraft am Vorderrad ohne genaue Information über den aktuellen Fahrer ist folglich nicht zielführend. Eine Mehrwertfunktion wie die „Load-Adaptive Control Mode“ aus [24], welche die Fahrzeugmasse bei Kraftwagen schätzt, könnte in diesem Zusammenhang eine deutliche Verbesserung darstellen.

Perzentil		$F_{z,v,stat}$		$F_{z,v,dyn}$	
		N	%	N	%
m	5	278,21	-24,65	170,36	-22,75
	50	369,21	0	220,57	0
	95	451,93	22,41	274,43	24,42
w	5	209,94	-23,77	120,88	.25,49
	50	275,39	0	162,22	0
	95	356,42	29,42	222,19	36,96

Tabelle 10. Abhängigkeit Radaufstandskraft & Fahrer

5 Zusammenfassung und Ausblick

Die im vorliegenden Manuskript durchgeführte simulationsgestützte Einflussbewertung diverser Zustandsgrößen, Regelstrecken- und Umweltparameter auf die kritische Fahrsituation Vorderradblockieren liefert für die Funktionsentwicklung eines Blockierverhinders wesentliche Erkenntnisse. So identifizierte die detaillierte Diskussion der bremsdynamischen Grundlagen wichtige Einflussgrößen auf das Vorderradblockieren. Die Ergebnisse einer ausführlichen Literaturrecherche ergänzt durch eigene experimentelle Versuche lieferte die zur Durchführung der Sensitivitätsanalysen notwendige hohe Güte der Modellparameter. Der Bremsdruck, die Regelstreckenparameter Perzentil-Person und Bremsenkennwert sowie der Umweltparameter Zustand der Fahrbahnoberfläche zeigten sich als Haupteinflussgrößen, ob Vorderradblockieren, Hinterradabheben oder keine kritische Fahrsituation auftritt. Die in einer Worst-Case Betrachtung ermittelte minimale Blockierzeit in Abhängigkeit

der Geschwindigkeit lieferte die für den Regelvorgang zur Verfügung stehende Zeit. Aufgrund der Dominanz des Fahrers bei Fahrrädern zeigte sich die eingeschränkte Verwendbarkeit eines Fahrzustandsbeobachters, wie er aus dem ESP bekannt ist.

Eine wichtige zukünftige Arbeit ist die Weiterentwicklung des gleichungsbasierten Simulationsmodells. Aufgrund der Zusammenfassung von Fahrer mit dem Aufbau des Fahrrads zu einem Massepunkt ist die Schwerpunktverlagerung während des Bremsvorgangs nur eingeschränkt untersuchbar. Hinsichtlich der Parametrierung ist eine Substitution der bisher genäherten μ -Schlupf-Kurven durch reale vermessene Kurven von Fahrradreifen unverzichtbar. Weiterhin ist eine Perzentil-Person abhängige Steifigkeit und Dämpfung der Tauchfederung an der Gabel, zur realistischeren Abbildung der dynamischen Radlastverlagerung auf die Schwerpunktlage notwendig. Durchzuführende Ergänzung hinsichtlich der Sensitivitätsanalysen ist die Einbindung und Untersuchung des Reifenluftdrucks.

6 Danksagung

Die Autoren danken allen akademischen und industriellen Partnern für die Unterstützung von BikeSafe sowie dem Bundesministerium für Bildung und Forschung (BMBF) für die Förderung des Projekts (FKZ 03FH063PX3).

Literaturverzeichnis

- [1] D. Eisenberger. *Zahlen-Daten-Fakten zum Deutschen E-Bike-Markt 2014*. Pressemitteilung des Zweirad-Industrie-Verbands (ZIV), Berlin, 2015.
- [2] K. Enisz, I. Szalay, K. Nagy, G. Kohlrusz, D. Fodor, R. Jakab. *Development of a bicycle anti-lock braking system prototype*. International Journal of Automotive Engineering and Technologies, S. 111-118, 2014.
- [3] D. Eisenberger. *Zahlen-Daten-Fakten zum Fahrradmarkt in Deutschland*. ZIV Wirtschaftspressekonferenz, Berlin, 2015.
- [4] P. Barzel, M. Bollschweiler und C. Smolik. *Die neue Fahrradtechnik: Material, Konstruktion, Fertigung*. Bielefelder Verlag, Deutschland, 2008.
- [5] M. Gressmann et al. *Fachkunde Fahrradtechnik*. Europa Verlag, Deutschland, 2009.

- [6] C. Wolff. *Grundlegendes zum Bremsvorgang*. In: Bremsenhandbuch, Herausgeber: B. Breuer und K. Bill, Springer Vieweg Verlag, Deutschland, S. 13-64, 2012.
- [7] J. Wrede. *Überschlagschutzsystem für Zweiräder*. Offenlegungsschrift, Deutschland, 2009.
- [8] M. Mitschke und H. Wallentowitz. *Dynamik der Kraftfahrzeuge*. Springer Verlag, Deutschland, 2004.
- [9] O. Maier, M. Pfeiffer und J. Wrede. *Bremsdynamisches Assistenzsystem für elektrifizierte Fahrräder: Aufbau und Validierung eines Umgebungsmodells als Grundlage der modellbasierten Entwicklung*. Tagungsband VDI Mechatronik 2015, Dortmund, S. 271-276, 2015.
- [10] DIN 33402. *Ergonomie – Körpermaße des Menschen*. Beuth Verlag, Deutschland, 2008.
- [11] DIN EN ISO 7250. *Wesentliche Maße des menschlichen Körpers für die technische Gestaltung*. Beuth Verlag, Deutschland, 2010.
- [12] NASA. *International Space Station Flight: Cre Integration Standard*. Publikation, USA, 1999.
- [13] J. Reimpell und K. Hoseus. *Grundlagen: Schwerpunkt von Einzelmassen*. In: Fahrwerktechnik: Fahrzeugmechanik, Herausgeber: J. Reimpell, Vogel Verlag, Deutschland, S. 36-50, 1992.
- [14] C. Landerl, H. Köhler, H.-A. Wagner, R. Lewien, G. Vilsmeier, G. Heyl und S. Fritschle. *Bremsverhalten und Bremsen von Einspurfahrzeugen*. In: Bremsenhandbuch: Grundlagen, Komponenten, Systeme, Fahrdynamik, Herausgeber: B. Breuer und K.-H. Bill (Hrsg.), Springer Vieweg Verlag, Deutschland, S. 227-270, 2012.
- [15] C. Kleinlein und D. Severin. *Eigenschaften der Reibpaarungen im Bremsprozess*. In: Bremsenhandbuch: Grundlagen, Komponenten, Systeme, Fahrdynamik, Herausgeber: B. Breuer und K.-H. Bill (Hrsg.), Springer Vieweg Verlag, Deutschland, S. 529-546, 2012.
- [16] M. Burckhardt. *Der Reifen als Regelgröße*. In: Fahrwerktechnik: Radschlupf-Regelsysteme, Herausgeber: J. Reimpell, Vogel Verlag, Deutschland, S. 15-62, 1993.
- [17] B. Corves, J. Breuer, F. Schoeler und P. Ingelath. *A Three-Dimensional Multibody Model of a Full Suspension Mountain Bike*. Proceedings of the MUSME Conference, in Huatulco, Mexico, S. 133-141, 2014.
- [18] J. D. G. Kooijman, A. L. Schwab und J. P. Meijaard. *Experimental validation of a model of an uncontrolled bicycle*. Multibody Systems Dynamics, S. 115–132, 2008.
- [19] K. Schleinitz, L. Franke-Bartholdt, T. Petzoldt, S. Schwanitz, T. Gehlert, M. Kühn. *Pedelec - Naturalistic Cycling Study*. Forschungsbericht Nr. 27, Gesamtverband der Deutschen Versicherungswirtschaft e. V., Unfallforschung der Versicherer, Deutschland, 2014.
- [20] M. Dozza und G. F. Bianchi Piccinini. *Do cyclists on e-bikes behave differently than cyclists on traditional bicycles?* Proceedings of the International Cycling Safety Conference 2014, Göteborg, Schweden, 2014.
- [21] O. Maier, E. Keller, M. Pfeiffer und J. Wrede. *Development of a Dynamic Safety System for Electrically Power Assisted Cycles: Design, Implementation and Evaluation of Road Tests*. IEEE/ ASME Transactions on Mechatronics, Übermittelt zur Veröffentlichung, 2014.
- [22] V. Cossalter. *Motorcycle Dynamics*. Gardners Books Ltd. Eastbourne, Italien, 2006.
- [23] M. Burckhardt. *Zeitabläufe bei Schlupfregelungen*. In: Fahrwerktechnik: Radschlupf-Regelsysteme, Herausgeber: J. Reimpell, Vogel Verlag, Deutschland, S. 63-86, 1993.
- [24] A. van Zanten und F. Kost. *Bremsenbasierte Assistenzfunktionen*. In: Handbuch Fahrerassistenzsysteme, Herausgeber: H. Winner et al., Springer Verlag, Deutschland, S. 356-394, 2012.
- [25] DIN EN ISO 4210. *Fahrräder – Sicherheitstechnische Anforderungen an Fahrräder*. Beuth Verlag, Deutschland, 2015.

Ein Ansatz zur datenbasierten Reglerparametrierung

Alexander Schöley¹, Wolfgang Drewelow¹, Torsten Jeansch¹

¹Universität Rostock, Institut für Automatisierungstechnik, Lehrstuhl Regelungstechnik,
18119 Rostock, Germany

alexander.schoeley@uni-rostock.de

Steuerungen und Regelungen für technische Prozesse werden häufig auf der Grundlage eines Modells entworfen. Für diese Vorgehensweise existieren viele Methoden wie z.B. Modellinvertierung, Exakte Linearisierung, LQR und Polvorgabe. Die Güte der Regelung ist dabei von der Genauigkeit des Modells und von getroffenen Vereinfachung abhängig. Bei komplexen und stark nichtlinearen Systemen gestaltet sich die Modellbildung und Identifikation als schwierig und das Modell kann das reale Prozessverhalten nur in eingeschränkten Arbeitsbereichen widerspiegeln. Problematisch ist auch die Synthese einer Vorsteuerung eines nicht-minimalphasigen Systems durch Modellinvertierung. Diese Schwierigkeiten wirken sich negativ auf die Regelqualität aus. Im Gegensatz zu den modellbasierten Ansätzen existieren datenbasierte Methoden, bei denen die Angabe eines expliziten Modells nicht erforderlich ist. Auf der Basis von Messdaten können je nach Vorgehensweise Vorsteuerungen und Regelungen entworfen oder bestehende optimiert werden. Beispiele für datenbasierte Entwurfsverfahren sind "Virtual Reference Feedback Tuning" (VRFT), "Iterative Feedback Tuning" (IFT) und Iterativ Lernende Regelungen (ILR). In diesem Beitrag wird die Methodik des datenbasierten Reglerentwurfs mit VRFT demonstriert. Dabei werden Messdaten der Ein- und Ausgänge des Prozesses als Messung innerhalb eines geschlossenen Regelkreises angesehen. Der Reglerentwurf wird anschließend auf eine Identifikation der Reglerparameter zurückgeführt. Die Anwendung der VRFT-Methode wird an einem BLDC-Motorprüfstand dargestellt.

1 Einleitung

Mit steigenden Rechenleistungen lassen sich in Industrieprodukten immer mehr leistungsfähigere Steuerungs- und Regelungsverfahren realisieren. Beispielsweise existieren in modernen Motorsteuergeräten eine große Anzahl von Regelkreisen. Diese müssen für neue Motoren passend bedatet und gegebenenfalls an veränderte Betriebsbedingungen oder an Verschleiß und Alterung angepasst werden. Dem Applikateur stehen für die Parametrierung dieser Regelkreise verschiedene Werkzeuge zur Verfügung. Dazu zählen grobe Einstellverfahren auf der Basis von einfachen Streckenmodellen oder Testsignale zur Charakterisierung der Regelstrecke auf deren Grundlage ein Reglerentwurf erfolgt. Da reale Systeme ein hohes Maß an Komplexität wie Nichtlinearitäten und Zeitvarianzen aufweisen, führen diese

modellbasierten Einstellregeln mit vereinfachenden Annahmen im Allgemeinen nicht auf optimale Regler.

Andererseits können mit moderner Sensorik viele Prozessgrößen schnell erfasst und aufgezeichnet werden. Mit leistungsfähiger Rechenhardware ist es möglich, diese Messdaten online auszuwerten. Erfolgt die Auswertung offline, können auch akasale Rechenoperationen durchgeführt werden, beispielsweise eine Filterung ohne Phasenverschiebung. In den letzten Jahren wurden verschiedene Verfahren entwickelt, um mit vorhandenen Messdaten Regler auszulegen. Dazu zählen Verfahren des "Iterative Feedback Tuning" (IFT) [1], das "Virtual Reference Feedback Tuning" [2], [3] und das "Fictitious Reference Iterative Tuning" (FRIT) [4]. Mit diesen datenbasierten Algorithmen können Regler ohne explizites Modell der Regelstrecke bestimmt, angelernt oder optimiert werden.

Das Ziel dieses Beitrags ist es, die Eignung eines datenbasierten Verfahrens zu überprüfen. Dazu wird die VRFT-Methode an einem BLDC-Motorprüfstand simulativ und praktisch angewandt.

Der Beitrag ist wie folgt aufgebaut. Im Abschnitt 2 wird die VRFT-Methode vorgestellt. Anschließend wird in Abschnitt 3 der BLDC-Prüfstand beschrieben und eine Modellierung vorgenommen. Im 4. Abschnitt wird ein Regler nach dem VRFT-Ansatz eingestellt und mit der bestehenden Regelung des Prüfstandes verglichen. Der letzte Abschnitt gibt eine Zusammenfassung des Beitrags.

2 Die VRFT-Methode

Der "Virtual Reference Feedback Tuning" (VRFT) - Ansatz stellt eine Möglichkeit dar, einen Regler zu parametrieren, ohne dass ein Streckenmodell explizit bekannt ist. Die VRFT wurde erstmals von Campi, Lecchini und Savaresi in [2] vorgeschlagen. Mit dieser Methode werden Reglerparameter bestimmt, d.h. die Reglerstruktur muss bereits im Vorfeld festgelegt werden. Die Bestimmung der Reglerparameter erfolgt auf der Grundlage von Prozesseingangs- und Prozessausgangsdaten. Darüber hinaus sind keine weitere Messungen erforderlich. Im Gegensatz zu iterativen Verfahren wird die VRFT-Methode daher als direkte Methode oder "One-Shot"-Methode bezeichnet [2], [3], [5].

2.1 Einführung

Als Ziel einer Regelung kann das optimale Folgen der Regelgröße $y(t)$ zum Referenzwerte $w(t)$ definiert werden. Dieser Sachverhalt lässt sich als Gütefunktional J_{allg} ausdrücken, das minimiert werden soll:

$$J_{allg}(\theta) = \|y(t, \theta) - w(t)\|^2. \quad (1)$$

In (1) werden die Reglerparameter im Spaltenvektor θ zusammengefasst und $\|\cdot\|^2$ bezeichnet das Quadrat der 2-Norm.

Ein Nachteil ist, dass $y(t, \theta)$ auch vom Prozess abhängt und das Gütekriterium J_{allg} daher ohne Prozesswissen nicht minimiert werden kann.

Statt des Kriteriums (1) kann ein Model-Matching-Ansatz verfolgt werden. Ziel ist es dann, dass sich der geschlossene Kreis $G_{GK}(s, \theta)$ bestehend aus Regler, Prozess und Rückführung, wie ein Referenzmodell in Form der Übertragungsfunktion $M(s)$ verhält:

$$J(\theta) = \|G_{GK}(s, \theta)[w(t)] - M(s)[w(t)]\|^2. \quad (2)$$

Mit der Schreibweise $M(s)[w(t)]$ wird der Ausgang des Systems $M(s)$ im Zeitbereich bezeichnet, wenn das Signal $w(t)$ am Eingang anliegt.

Jedoch lässt sich auch (2) nicht ohne Prozesswissen minimieren.

2.2 Ablauf

Die VRFT-Methode nutzt den Model-Matching-Ansatz um ein Gütekriterium zu entwickeln, welches sich ohne Informationen über den Prozess minimieren lässt. Das Gütekriterium J wird durch das Kriterium J_{VRFT} ersetzt, sodass J und J_{VRFT} dasselbe globale Minimum besitzen. Ist die vorgegebene Reglerstruktur linear in den Reglerparametern, ist J_{VRFT} ein quadratischer Ausdruck, dessen Minimum sich analytisch mit der Least-Squares-Methode berechnen lässt.

Bei der VRFT-Methode wird folgendermaßen vorgegangen. Zunächst wird ein Regler $C(s, \theta)$ und das Referenzmodell $M(s)$ gewählt. Außerdem muss ein Experiment am Prozess durchgeführt werden, sodass Ein- und Ausgangsdaten ($u(t)$, $y(t)$) des Prozesses vorliegen. Dieses Experiment kann direkt am Prozess oder innerhalb einer geschlossenen Kette stattfinden. Anschließend wird angenommen, dass die Messdaten aus einem Experiment an einem virtuellen, geschlossenen Regelkreis mit dem Regler $C(s, \theta)$ stammen und sich dieser virtuelle Regelkreis wie das Referenzmodell $M(s)$ verhält. Unter diesen Annahmen existiert für den virtuellen Regelkreis eine virtuelle Führungsgröße $w_{virt}(t)$, die den gemessenen Verlauf $y(t)$ erzeugt. Es gilt der folgende Zusammenhang:

$$y(t) = M(s)[w_{virt}(t)]. \quad (3)$$

Der virtuelle Regelfehler $e_{virt}(t)$ ist:

$$e_{virt}(t) = w_{virt}(t) - y(t). \quad (4)$$

Das Blockschaltbild in Abbildung 1 stellt das Vorgehen der VRFT dar. Die durchgezogenen Linien kennzeichnen die realen Signale $u(t)$ und $y(t)$. Die unterbrochenen Linien stellen den virtuellen Regelkreis dar. Die gepunktete Verbindung stellt den Zusammenhang zwischen der Regelgröße $y(t)$ und der virtuellen Führungsgröße $w_{virt}(t)$ über das Referenzmodell $M(s)$ her.

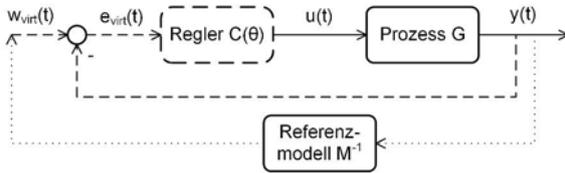


Abbildung 1: Blockschaltbild zum VRFT-Ablauf

Aufgrund der Annahme eines virtuellen Regelkreises mit dem Regler $C(s, \theta)$ liegen mit den Signalen $e_{virt}(t)$ und $u(t)$ die Ein- und Ausgangsdaten des Reglers vor, mit denen die Reglerparameter bestimmt werden können. Das Gütefunktional J_{VRFT} ist:

$$J_{VRFT}(\theta) = \|u(t) - C(s, \theta)[e_{virt}(t)]\|^2. \quad (5)$$

J_{VRFT} ist nicht vom Prozess abhängig und kann daher minimiert werden. Das Problem der Reglerparametrierung wird auf diese Weise in eine Identifikation der Reglerparameter umgewandelt.

Die Überführung des Gütefunktionals J in das Funktional J_{VRFT} erfolgt unter der Annahme, dass beide Funktionale dasselbe Minimum besitzen. Um diese Voraussetzung zu erfüllen, werden Bedingungen an das Stellsignal gestellt. $u(t)$ muss Frequenzanteile besitzen, sodass der Prozess in den relevanten Arbeitspunkten angeregt wird. Weiterhin muss beachtet werden, dass das gewählte Referenzverhalten M am realen Prozess realisierbar ist [3], [6].

3 Der BLDC-Prüfstand

Die VRFT-Methode wird an einem BLDC-Motorprüfstand demonstriert. Dies erfolgt zunächst in einer Simulation. In diesem Abschnitt wird der

Prüfstandsaufbau beschrieben und eine Modellierung des Prüfstandes vorgenommen.

3.1 Aufbau des Prüfstandes

Der reale Motorprüfstand besteht aus zwei BLDC (brushless direct current) - Motoren, die mit einer Momentenmesswelle verbunden sind. Einer der Motoren wirkt in diesem Szenario als Prüfling, der andere als Lastmaschine. Physikalische Größen des Prüfmotors werden mit dem Index $(\cdot)_1$, Größen des Lastmotors mit dem Index $(\cdot)_2$ gekennzeichnet.

Beide Motoren werden jeweils von einer kommerziellen Leistungselektronik angesteuert, welche die drei Phasenspannungen für den BLDC-Motor erzeugen und die Kommutierung durchführt. In diese Ansteuerung kann mit einer Steuerspannung U_i ($i = 1, 2$) eingegriffen werden, die daher als Stellgröße für den jeweiligen Motor wirkt. Beide Motoren sind mit hochauflösenden Inkrementalgebern ausgestattet, mit denen die Rotorposition gemessen und die Rotordrehzahl durch Ableitung berechnet werden kann.

Mit diesem Aufbau ergibt sich die Möglichkeit, das Drehmoment an der Messwelle und die Drehzahl der Messwelle zu regeln und so vorgegebene Fahrzyklen abzufahren. Damit stellt der Prüfstand ein MIMO-System mit zwei Eingängen und zwei Ausgängen dar, das in Abbildung 2 zu sehen ist.

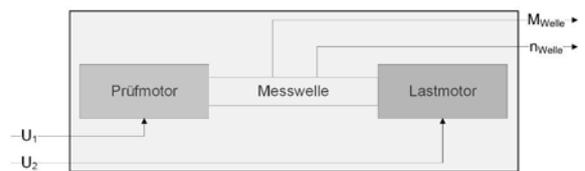


Abbildung 2: Stell- und Regelgrößen am BLDC-Prüfstandes

3.2 Modellierung des Prüfstandes

Das Modell des Prüfstandes wird in zwei Komponenten unterteilt. Ein Motormodell beschreibt den Zusammenhang zwischen der Steuerspannung U_i und dem abgegebenen Motormoment M_i . Aus beiden Motormomenten setzt sich anschließend das Wellenmoment M_{Welle} und die Wellendrehzahl n_{Welle} zusammen.

men. Diese Aufteilung des Modells ist für Motorprüfstände bereits in anderen Veröffentlichungen angewandt worden [7], [8].

Das Motormoment M_i setzt sich aus einem vom Motor indiziertem Moment $M_{i,indiziert}$ und einem Reibmoment $M_{i,Reibung}$ zusammen: $M_i = M_{i,indiziert} - M_{i,Reibung}$. Mit der Messwelle wird bereits die Summe aus beiden Anteilen erfasst, wodurch die Reibung im Motormodell enthalten ist. Das statische Übertragungsverhalten von der Steuerspannung U_i zum Motormoment M_i wird durch ein Motorkennfeld k_i beschrieben, das zusätzlich von der aktuellen Drehzahl abhängt:

$$M_{i,static} = k_i(U_i, n_{Welle})$$

Die dynamische Komponente des Motormodells besitzt Tiefpasscharakter [7]. Eine Modellverifikation zeigte, dass die Annahme eines PT1-Totzeit-Elementes das genaueste Modell liefert. Das Motormodell lässt sich dann symbolisch schreiben:

$$M_i = k(U_i, n_{Welle}) \frac{1}{1 + sT_i} e^{-sT_{ti}}. \quad (6)$$

Das Wellenmodell verknüpft beide Motormomente zum Wellenmoment und zur Wellendrehzahl. In Abbildung 3 ist der Aufbau der mechanischen Welle skizziert. Mit J_1 bzw. J_2 wird das Trägheitsmoment der Prüf- bzw. Lastseite bezeichnet. Es setzt sich jeweils aus dem Trägheitsmoment des Motors, des Getriebes und einem Anteil der Messwelle zusammen. Mit den in Abbildung 3 definierten Größen ergeben sich folgende Momentengleichgewichte:

$$J_1 \dot{\omega}_1 = M_1 - M_{Welle} \quad (7)$$

$$J_2 \dot{\omega}_2 = -M_2 + M_{Welle} \quad (8)$$

Unter der Annahme einer starren Welle gilt: $\omega_1 = \omega_2 = \omega_{Welle}$. Damit ergibt sich aus (7) und (8) für das Wellenmoment:

$$M_{Welle} = M_1 \frac{J_2}{J_1 + J_2} + M_2 \frac{J_1}{J_1 + J_2} \quad (9)$$

und für die Wellendrehzahl mit $n_{Welle} = \frac{1}{2\pi} \omega_{Welle}$:

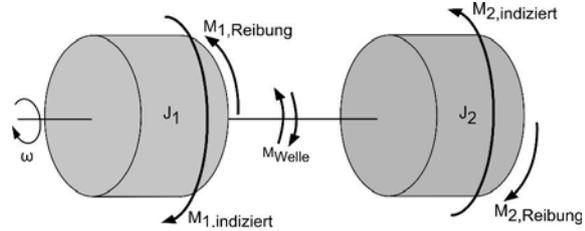


Abbildung 3: Definitionen der physikalischen Größen an der mechanischen Welle

$$n_{Welle} = \frac{1}{2\pi} \int \frac{M_1 - M_2}{J_1 + J_2}. \quad (10)$$

Die Gleichungen (9) und (10) bilden das Wellenmodell.

Um für die Regelgrößen Wellenmoment und Wellendrehzahl getrennte Regler entwerfen zu können, wurde ein Entkoppelnetzwerk ermittelt. Auf diese Weise werden die Querverbindungen eliminiert und es liegen aus Sicht der Regelung zwei separate Regelstrecken vor. Das Entkoppelnetzwerk besteht aus den inversen Motorkennfeldern und dem inversen Wellenmodell.

Die Modellgleichungen (6), (9), (10) und das Entkoppelnetzwerk wurden in Matlab/Simulink realisiert und bilden die Simulation des Prüfstandes.

4 Reglerentwurf und Ergebnisse

Am BLDC-Prüfstand werden beim Abfahren eines Fahrzyklus die Wellendrehzahl und das Wellenmoment geregelt. Dabei kommen PI-Regler zum Einsatz, die nach einem modellbasierten Entwurfsverfahren parametrisiert wurden. Auf der Basis von Testsignalen wurden die Regelstrecken charakterisiert und die Reglerparameter anschließend berechnet. Die Performance dieser Regler ist in Abbildung 4 und 5 zu sehen. Während die Regelung der Wellendrehzahl eine gute Regelgüte erzielt, ist die Qualität der Regelung des Wellenmomentes unbefriedigend. Die Wellenmomentregelung soll daher mit der VRFT-Methode verbessert werden.

Mit dem PI-Regler ist die Reglerstruktur für die VRFT-Methode festgelegt. Die Verstärkungen des

Proportional- und Integralteils (K_p und K_i) bilden den Parametervektor θ . Es gilt in kontinuierlicher Schreibweise:

$$\begin{aligned} C(s, \theta) &= K_p + \frac{K_i}{s} \\ &= \begin{bmatrix} K_p & K_i \end{bmatrix} \begin{bmatrix} 1 \\ \frac{1}{s} \end{bmatrix} \\ &= \theta^T C'(s). \end{aligned} \quad (11)$$

Als Referenzmodell wird ein PT1-Element mit der Verstärkung K und der Zeitkonstante T gewählt. Die virtuelle Führungsgröße muss nach (3) so bestimmt werden, dass

$$y(t) = \frac{K}{sT+1} [w_{virt}(t)] \quad (12)$$

gilt. Der virtuelle Regelfehler $e_{virt}(t)$ wird mit (4) berechnet.

Setzt man den Regler (11) in das VRFT-Gütekriterium (5) ein, erhält man:

$$J_{VRFT}(\theta) = \|u(t) - \underbrace{\theta^T C'(s)}_{\varphi} [e_{virt}(t)]\|^2.$$

Der Term φ ist die Regressionsmatrix. Sie enthält die Reaktionen des P- und I-Anteils auf den Eingang $e_{virt}(t)$ mit einer Proportional- und Integralverstärkung von 1. Unter Berücksichtigung der Dimensionen der Vektoren und Matrizen gilt:

$$J_{VRFT}(\theta) = \|u(t) - \varphi\theta\|^2 \quad (13)$$

Das Gütekriterium (13) wird mit der Least-Squares-Methode minimiert. Der optimale Parametervektor θ_{opt} ergibt sich zu:

$$\theta_{opt} = (\varphi^T \varphi)^{-1} \varphi^T u(t). \quad (14)$$

Die notwendigen Prozessein- und Prozessausgangsdaten werden generiert, indem mit dem Prüfstand ein Fahrzyklus mit den vorhandenen PI-Reglern nachgefahren wird (siehe Abbildungen 4 und 5).

Die mit der VRFT-Methode neu bestimmten Parameter für den Wellenmomentregler wurden zunächst an

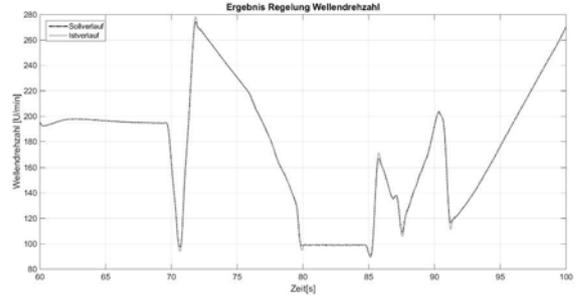


Abbildung 4: Ergebnis der Drehzahlregelung (modellbasierte Parametrierung)

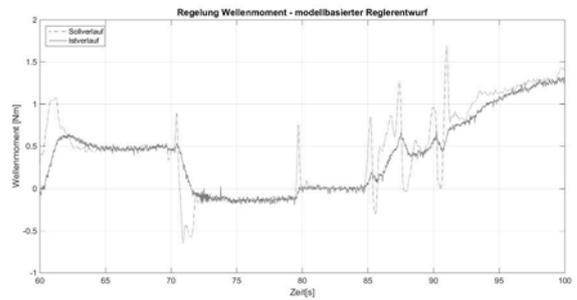


Abbildung 5: Ergebnis der Drehmomentregelung (modellbasierte Parametrierung)

der Simulation des BLDC-Prüfstandes erprobt. Das Ergebnis dieser Regelung ist für das Wellenmoment in Abbildung 6 dargestellt. Es ist zu erkennen, dass sowohl in statischen Phasen (z.B. zwischen 80s und 85s) als auch bei schnellen Änderungen des Sollverlaufs (z.B. bei ca. 90s) ein gutes Führungsverhalten erreicht wird.

In Abbildung 7 ist das Reglergebnis für das Wellenmoment am realen BLDC-Prüfstand zu sehen. Hier werden die Erkenntnisse aus der simulativen Erprobung bestätigt. Aufgrund von Störungen wird eine schlechtere Regelperformance als in der Simulation erreicht, es ist aber eine verbesserte Regelqualität im Vergleich zur modellbasierten Parametrierung festzustellen.

5 Zusammenfassung

In diesem Beitrag wurde ein datenbasiertes Verfahren verwendet, um die Regelgüte eines BLDC-

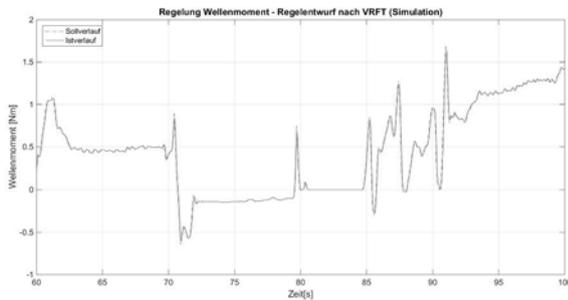


Abbildung 6: Ergebnis der Drehmomentregelung in der Simulation (VRFT-Parametrierung)

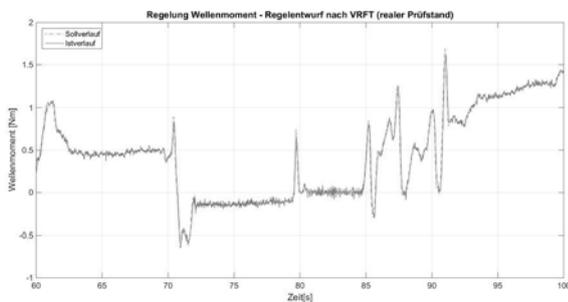


Abbildung 7: Ergebnis der Drehmomentregelung am realen Prüfstand (VRFT-Parametrierung)

Prüfstandes zu verbessern. Die vorher bestehenden Reglerparameter wurden durch ein modellbasiertes Einstellverfahren ermittelt. Es wurde nachgewiesen, dass eine Parametrierung nach der datenbasierten VRFT-Methode in einer Simulation und im praktischen Test eine gesteigerte Reglerperformance bietet. Die VRFT-Methode kann daher als weiteres Werkzeug und Hilfsmittel bei der Bedatung von Regelkreisen angesehen werden.

References

- [1] H. Hjalmarsson, M. Gevers, S. Gunnarson, O. Lequin. *Iterative Feedback Tuning: Theory and Applications*, IEEE Control Systems 18, S. 26-41, 1998.
- [2] M. Campi, A. Lecchini, S. Savaresi. *Virtual Reference Feedback Tuning: a direct method for the design of feedback controllers*, Automatica 38, S. 1337-1346, 2002.
- [3] M. Campi, S. Savaresi. *Direct Nonlinear Control Design: The Virtual Reference Feedback Tuning (VRFT) Approach*, IEEE Transactions on Automatic Control, 2006.
- [4] O. Kaneko, S. Soma, T. Fujii. *A Fictitious Reference Iterative Tuning (FRIT) in the two-degree of freedom control scheme and its application to closed loop system identification*, 16th IFAC World Congress, 2005.
- [5] S. Savaresi, G. Guardabassi. *Approximate I/O Feedback Linearization of Discrete-Time Non-Linear Systems via Virtual Input Direct Design*, Automatica 34, S. 715-722, 1998.
- [6] A. Bazanella, L. Campestri, D. Eckhard. *Data-Driven Controller Design - The H2 Approach* Springer, Brasilien, 2012.
- [7] E. Grünbacher. *Robust Inverse Control of a Class of Nonlinear Systems* Dissertation, Johannes Kepler Universität Linz, Österreich, 2007.
- [8] E. Grünbacher, L. del Re. *Robust Inverse Control for Combustion Engine Test Benches*, American Control Conference, 2008.

Anwendung Subspace basierter Identifikationsverfahren auf gesteuerte Mehrkörpersysteme

Mathias Marquardt¹, Peter Dünow¹, Sandra Baßler¹

¹Hochschule Wismar - University of Applied Sciences Technology, Business and Design

mathias.marquardt@hs-wismar.de

In diesem Beitrag werden unterschiedliche Subspace basierte Systemidentifikationsverfahren auf ein gesteuertes mechatronisches System angewandt, um ein arbeitspunktabhängiges lineares Modell des zugrunde liegenden Systems zu identifizieren. Da der untersuchte Prozess sowohl stabile als auch instabile Arbeitspunkte aufweist, werden die Identifikationsexperimente im geschlossenen Regelkreis durchgeführt. Zudem besitzt der Prozess mehrere Ein- und Ausgangsgrößen, wodurch die Anforderungen an das Identifikationsverfahren nochmals erhöht werden. Die in den letzten Jahren entwickelten Subspace basierten Identifikationsverfahren bieten die Möglichkeit lineare Zustandsraummodelle auf Basis von gemessenen Ein-/Ausgangsdaten mit geringem Vorwissen über den Prozess zu schätzen. Zudem können die Verfahren mit Daten aus geschlossenen Kreisen umgehen und die Anwendung auf Mehrgrößensysteme ist ohne Modifikationen möglich. Um das Verhalten der unterschiedlichen Verfahren zu untersuchen, werden Identifikationsexperimente an einem simulierten Mehrkörpersystem durchgeführt und die Ergebnisse hinsichtlich ihrer Modellierungsgüte untereinander verglichen.

1 Einleitung

Für die datenbasierte Systemidentifikation haben sich in den letzten Jahrzehnten zwei Ansätze als besonders geeignet erwiesen. Zum einen die Prädiktionfehlermethoden (engl. prediction error methods PEM) [6] und zum anderen Subspace basierte Methoden (engl. subspace state space identification SSID). Wenn es um die Identifikation von Mehrgrößensystemen geht, haben die Subspace Verfahren einen entscheidenden Vorteil, da sich nur die Dimensionen der in den Algorithmen aufgestellten Matrizen ändern und ansonsten das selbe Verfahren wie im Eingrößenfall angewandt wird. Weitere Vorteile der Subspace Methoden sind das nicht iterative Verhalten und die Schätzung der Systemordnung als Zwischenschritt eines solchen Verfahrens.

Die ersten entwickelten Subspace Verfahren waren bei ihrer Anwendung auf Ein-/Ausgangsdaten beschränkt, die aus Experimenten im offenen Regelkreis gewonnen wurden, da bei geschlossenem Regelkreis

eine Korrelation zwischen vergangenen Innovationen und zukünftigen Eingangsdaten vorliegt [5]. Um die Subspace Identifikation auch im geschlossenen Kreis anzuwenden, wurden die existierenden Verfahren wie die *Numerical Subspace State Space System Identification* (N4SID) [4], die *Canonical Variate Analysis* (CVA) [1] und die *MIMO Output-Error State Space model identification* (MOESP) [3] modifiziert und erweitert. Das ermöglichte die erwartungstreue Identifikation von Systemen mit Rückführung.

Ljung und McKelvey [5] schlagen vor, zunächst ein ARX Modell aus den gemessenen Daten zu schätzen, welches eine höhere Systemordnung als das zu identifizierende System aufwies (high order ARX), um daraus einen „j-step ahead“ Prädiktor zu konstruieren. Dieser ist erwartungstreu solange das System mindestens eine Verzögerung von einem Abtastschritt zwischen Regelgröße und Stellgrößen aufweist. Dieser Ansatz wurde von Chiuso [9] aufgegriffen, woraus die Klasse der Prädiktor basierten Subspace Identifikation entwickelt wurde. Ein weiterer Ansatz ist, mit Hilfe des HOARX Modells eine Innovationssequenz zu

schätzen, wie von Qin und Ljung vorgeschlagen [8]. Beide Ansätze verfolgen die Minimierung der Korrelation zwischen Innovation und Stellgröße in einem ersten Schritt, um anschließend die unkorrelierten Daten für die Subspace Identifikation zu nutzen.

Diese Algorithmen wurden, in ihren unterschiedlichen Formen, von de Korte [10] zu einer gemeinsamen Basis vereinheitlicht und in die Klassen Ein-, Zwei- und Mehrschrittverfahren eingeteilt. In diesem Beitrag wird je ein Algorithmus einer Klasse beschrieben und für die Identifikation eines mechatronischen Mehrgrößensystems mit zwei Ein- und Ausgängen verwendet.

Die unterschiedlichen Verfahren werden zunächst in Abschnitt 2 erläutert. In Abschnitt 3 und 4 wird der verwendete Prozess und das Identifikationsexperiment beschrieben, dessen Ergebnisse im Abschnitt 5 dargestellt und diskutiert werden.

2 Subspace Systemidentifikation

In diesem Abschnitt wird die Funktionsweise der untersuchten Identifikationsalgorithmen erläutert. Den Algorithmen liegt das folgende Zustandsraummodell in Innovationsform zugrunde:

$$x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k + \mathbf{K}e_k \quad (1)$$

$$y_k = \mathbf{C}x_k + \mathbf{D}u_k + e_k \quad (2)$$

Mit der Systemmatrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, der Eingangsmatrix $\mathbf{B} \in \mathbb{R}^{n \times m}$, der Ausgangsmatrix $\mathbf{C} \in \mathbb{R}^{l \times n}$ sowie der Durchgriffsmatrix $\mathbf{D} \in \mathbb{R}^{l \times m}$ und dem Kalman Gain $\mathbf{K} \in \mathbb{R}^{n \times l}$. $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, $y_k \in \mathbb{R}^l$ und $e_k \in \mathbb{R}^l$ sind der Zustandsvektor, der Eingangsvektor, der Ausgangsvektor und der Innovationsvektor. Des Weiteren ist e_k ein mittelwertfreies gaußverteilt Rauschsignal mit der Kovarianzmatrix \mathbf{R} (3). δ_{ij} ist das Kronecker Delta.

$$E(\mathbf{e}_i \mathbf{e}_j^T) = \mathbf{R} \delta_{ij} \quad (3)$$

Vor der detaillierten Beschreibung der einzelnen Algorithmen, werden einige Notationen und Formulierungen festgelegt. Die gemessenen Daten werden zu

Block Hankel Matrizen mit folgender Notation zusammengesetzt:

$$\mathbf{Y}_{k,\tau,\sigma} = \begin{bmatrix} y_k & \cdots & y_{k+\sigma-1} \\ \vdots & \ddots & \vdots \\ y_{k+\tau-1} & \cdots & y_{k+\tau+\sigma-2} \end{bmatrix} \quad (4)$$

Die Matrizen für vergangen Daten werden mit $\mathbf{Y}_p = \mathbf{Y}_{0,p,N}$ und die für zukünftige Daten mit $\mathbf{Y}_f = \mathbf{Y}_{p,f,N}$ festgelegt. Die Notation gilt ebenfalls für Eingangsdaten $\mathbf{U}_p, \mathbf{U}_f$ und Innovationsdaten $\mathbf{E}_p, \mathbf{E}_f$. Aus dieser Notation ergibt sich die Anzahl der Spalten einer Hankel Matrix aus \mathcal{N} gemessenen Datenpunkten zu

$$N = \mathcal{N} - l_p - l_f + 1 \quad (5)$$

wobei l_p der vergangene Horizont und l_f der Zukunftshorizont ist. In einigen Algorithmen wird eine zeilenweise separierte Notation der Hankel Matrix benötigt (6).

$$\mathbf{Y}_i = \begin{bmatrix} \mathbf{Y}_{f1} \\ \vdots \\ \mathbf{Y}_{fi} \end{bmatrix} = \mathbf{Y}_{p,i,N} \quad \text{für } 1 \leq i \leq l_f \quad (6)$$

Der Markov Zustand des k-ten Abtastzeitpunktes besitzt die Form:

$$x_k = (q\mathbf{I} - \bar{\mathbf{A}})^{-1} (\bar{\mathbf{B}} \quad \mathbf{K}) \begin{pmatrix} u_k \\ y_k \end{pmatrix} \quad (7)$$

mit $\bar{\mathbf{A}} = \mathbf{A} - \mathbf{K}\mathbf{C}$ und $\bar{\mathbf{B}} = \mathbf{B} - \mathbf{K}\mathbf{D}$. Wird der Zustand über einem Vergangenheitshorizont von p Abtastschritten erweitert, ergibt sich die Markov Zustandsfolge zu:

$$\mathbf{X}_{p,N} = [x_p \quad x_{p+1} \quad \cdots \quad x_{p+N-1}] = \bar{\mathbf{A}}^p \mathbf{X}_{0,N} + \bar{\mathcal{H}} \mathbf{Z}_p \quad (8)$$

$$\approx \bar{\mathcal{H}} \mathbf{Z}_p \quad (9)$$

worin

$$\bar{\mathcal{H}} = [\bar{\mathbf{A}}^{p-1} \bar{\mathbf{B}} \quad \cdots \quad \bar{\mathbf{A}} \bar{\mathbf{B}} \quad \bar{\mathbf{B}} \quad \bar{\mathbf{A}}^{p-1} \mathbf{K} \quad \cdots \quad \bar{\mathbf{A}} \mathbf{K} \quad \mathbf{K}] \quad (10)$$

$$\mathbf{Z}_p = [\mathbf{U}_p^T \quad \mathbf{Y}_p^T]^T \quad (11)$$

Einige Algorithmen nutzen eine Reihe von Ausgangsprädiktoren deren Repräsentation in Innovationsform sich wie folgt darstellt:

$$\mathbf{Y}_f \approx \Gamma_f \mathcal{H} \mathbf{Z}_p + \mathbf{H}_f \mathbf{U}_f + \mathbf{G}_f \mathbf{E}_f + \mathbf{E}_f \quad (12)$$

mit der erweiterten Beobachtbarkeitsmatrix

$$\Gamma_f = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{f-1} \end{bmatrix} \quad (13)$$

und den Block Toeplitz Matrizen, welche die Markovparameter des Systems enthalten.

$$\mathbf{H}_f = \begin{bmatrix} \mathbf{D} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CB} & \mathbf{D} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{f-2}\mathbf{B} & \mathbf{CA}^{f-3}\mathbf{B} & \cdots & \mathbf{D} \end{bmatrix} \quad (14)$$

$$\mathbf{G}_f = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CK} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{f-2}\mathbf{K} & \mathbf{CA}^{f-3}\mathbf{K} & \cdots & \mathbf{0} \end{bmatrix} \quad (15)$$

Somit liegt das Hauptproblem der Subspace Identifikation in einer Schätzung von $\Gamma_f \mathcal{H}$, wie in (12) formuliert, und der darauf folgenden Bestimmung der Zustandsfolge $\mathbf{X}_{p,N}$. Um diese Zustandsfolge zu bestimmen wird mit Hilfe einer Singulärwertzerlegung eine Reduzierung der Ordnung vorgenommen (16). Eine optimale Systemordnung n kann entweder durch Auswertung der dominanten Singulärwerte oder durch Anwenden eines Informationskriteriums, wie dem *Akaike Information Criterion* (AIC), ermittelt werden.

$$\widehat{\Gamma_f \mathcal{H}} \mathbf{Z}_p = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \approx \mathbf{U}_n \mathbf{\Sigma}_n \mathbf{V}_n^T \quad (16)$$

Die geschätzte Zustandsfolge setzt sich aus den ersten n rechten Singulärvektoren zusammen.

$$\hat{\mathbf{X}}_{p,N} = \mathbf{V}_n^T \in \mathbb{R}^{n \times N} \quad (17)$$

Aus der geschätzten Zustandsfolge und den gemessenen Ein-/Ausgangsdaten können mit der Methode der kleinsten Quadrate die Systemmatrizen ermittelt werden. Um auch den Kalman Gain bei der Identifikation zu erhalten werden die Residuen der Lösung für die Ausgangsgleichung (18) als Schätzung der Innovationsfolge genutzt und in (21) eingesetzt.

$$\mathbf{Y}_{p,N-1} = \mathcal{Y} = \underbrace{[\hat{\mathbf{C}} \ \hat{\mathbf{D}}]}_{\Theta} \underbrace{\begin{bmatrix} \hat{\mathbf{X}}_{p,N-1} \\ \mathbf{U}_{p,N-1} \end{bmatrix}}_{\mathcal{W}} + \mathcal{E}_Y \quad (18)$$

$$\hat{\Theta} = \mathcal{Y} \mathcal{W}^\dagger \quad (19)$$

$$\hat{\mathcal{E}}_Y = \mathcal{Y} (\mathbf{I} - \mathcal{W}^\dagger \mathcal{W}) \quad (20)$$

$$\hat{\mathbf{X}}_{p+1,N-1} = [\mathbf{A} \ \mathbf{B} \ \mathbf{K}] \begin{bmatrix} \hat{\mathbf{X}}_{p,N-1} \\ \mathbf{U}_{p,N-1} \\ \hat{\mathcal{E}}_Y \end{bmatrix} \quad (21)$$

Unter bestimmten Bedingungen wirkt der Kalman Gain, der mit (21) bestimmt wurde, nicht stabilisierend auf das System. Ein weiterer Ansatz umgeht das Problem indem zunächst \mathbf{A} und \mathbf{B} mit Gleichung (22) sowie die Kovarianz der Innovationsfolge mit Gleichung (23) bestimmt werden. Durch Lösen der diskreten algebraischen Riccati Gleichung (24)-(25) kann ein stabilisierender Kalman Gain berechnet werden, was allerdings den Nachteil eines höheren Rechenaufwandes mit sich bringt.

$$\hat{\mathbf{X}}_{p+1,N-1} = [\mathbf{A} \ \mathbf{B}] \begin{bmatrix} \hat{\mathbf{X}}_{p,N-1} \\ \mathbf{U}_{p,N-1} \end{bmatrix} + \mathcal{E}_X \quad (22)$$

$$\begin{bmatrix} \hat{\mathbf{Q}} & \hat{\mathbf{S}} \\ \hat{\mathbf{S}}^T & \hat{\mathbf{R}} \end{bmatrix} = \lim_{N \rightarrow \infty} \frac{1}{N-1} \begin{bmatrix} \mathcal{E}_X \\ \mathcal{E}_Y \end{bmatrix} [\mathcal{E}_X \ \mathcal{E}_Y] \quad (23)$$

$$\hat{\mathbf{P}} = \hat{\mathbf{A}} \hat{\mathbf{P}} \hat{\mathbf{A}}^T + \hat{\mathbf{Q}} - \left(\hat{\mathbf{S}} + \hat{\mathbf{A}} \hat{\mathbf{P}} \hat{\mathbf{C}}^T \right) \left(\hat{\mathbf{C}} \hat{\mathbf{P}} \hat{\mathbf{C}}^T \right)^{-1} \left(\hat{\mathbf{S}} + \hat{\mathbf{A}} \hat{\mathbf{P}} \hat{\mathbf{C}}^T \right)^T \quad (24)$$

$$\hat{\mathbf{K}} = \left(\hat{\mathbf{S}} + \hat{\mathbf{A}} \hat{\mathbf{P}} \hat{\mathbf{C}}^T \right) \left(\hat{\mathbf{R}} + \hat{\mathbf{C}} \hat{\mathbf{P}} \hat{\mathbf{C}}^T \right)^{-1} \quad (25)$$

Wie zuvor beschrieben besteht das Hauptproblem bei der Subspace Identification darin, eine Schätzung für

$\Gamma_f \bar{\mathcal{K}}$ zu erhalten, um daraus den Markov Zustand zu konstruieren und die Systemmatrizen zu bestimmen. Die folgenden Algorithmen unterscheiden sich in der Art, wie diese Schätzung $\widehat{\Gamma_f \bar{\mathcal{K}}}$ aufgestellt wird.

2.1 Einschrittverfahren

Der erste Algorithmus gehört zur Klasse der Einschrittverfahren, welche nur ein Least-Squares Problem lösen, um die Zustandsfolge zu schätzen. Dazu wird ein ARX Modell hoher Ordnung mit Hilfe der ersten Blockzeile aus Gleichung (12) geschätzt (26)-(27). Da in realen Systemen eine Verzögerung entweder im Regler oder in der Strecke vorhanden ist, sind Systemeingang \mathbf{u}_k und Innovation \mathbf{e}_k unkorreliert und die Schätzung ist erwartungstreu.

$$\mathbf{Y}_{f1} = \mathbf{Y}_{p,N} = \mathbf{C} \bar{\mathcal{K}} \mathbf{Z}_p + \mathbf{D} \mathbf{U}_{f1} + \mathbf{E}_{f1} \quad (26)$$

$$\begin{bmatrix} \widehat{\mathbf{C} \bar{\mathcal{K}}} & \widehat{\mathbf{D}} \end{bmatrix} = \mathbf{Y}_{p,N} \begin{bmatrix} \mathbf{Z}_p \\ \mathbf{U}_{p,N} \end{bmatrix}^\dagger \quad (27)$$

Aus $\mathbf{C} \bar{\mathcal{K}}$ und \mathbf{D} können die Markovparameter gewonnen werden, aus denen, wie in [10] beschrieben, die Schätzung für $\Gamma_f \bar{\mathcal{K}}$ konstruiert werden kann.

2.2 Zweischrittverfahren

Bei den Zweischrittverfahren wird die Lösung aus Gleichung (27) benutzt, um eine Vorabschätzung der Toeplitzmatrix $\hat{\mathbf{H}}_f$ zu bilden und diese in Gleichung (12) einzusetzen, um die Korrelation zwischen Eingangssignal und Innovation zu beseitigen.

$$\mathbf{Y}_{f-} = \mathbf{Y}_f - \hat{\mathbf{H}}_f \mathbf{U}_f = \Gamma_f \bar{\mathcal{K}} \mathbf{Z}_p + \mathbf{G}_f \mathbf{E}_f + \mathbf{E}_f \quad (28)$$

Daraus ergibt sich das neue Problem in Gleichung (28), welches anschließend mit Hilfe der Canonical Variate Analysis zwischen \mathbf{Y}_{f-} und \mathbf{Z}_p gelöst wird.

$$\widehat{\Gamma_f \bar{\mathcal{K}}} = \left(\mathbf{Y}_{f-} \mathbf{Y}_{f-}^T \right)^{-1/2} \cdot \mathbf{Y}_{f-} \mathbf{Z}_p^T \cdot \left(\mathbf{Z}_p \mathbf{Z}_p^T \right)^{-1/2} \quad (29)$$

2.3 Mehrschrittverfahren

Der dritte Algorithmus gehört zur Klasse der Mehrschrittverfahren und basiert auf der Vorabschätzung der Innovation nach [8]. Hierbei wird Gleichung (12) zeilenweise gelöst, für $i = 1 \dots f$, was im Vergleich mit den anderen beiden Methoden den höchsten Rechenaufwand bedeutet.

$$\mathbf{Y}_{fi} = \Gamma_{fi} \bar{\mathcal{K}} \mathbf{Z}_p + \mathbf{H}_{fi} \mathbf{U}_i + \mathbf{G}_{fi} \hat{\mathbf{E}}_{i-1} + \mathbf{E}_{fi} \quad (30)$$

Um die Schätzung für $\Gamma_f \bar{\mathcal{K}}$ zu erhalten, werden aus den Teillösungen (31) die $\Gamma_{fi} \bar{\mathcal{K}}$ extrahiert und zusammengesetzt.

$$\begin{bmatrix} \widehat{\Gamma_{fi} \bar{\mathcal{K}}} & \hat{\mathbf{H}}_{fi} & \hat{\mathbf{G}}_{fi} \end{bmatrix} = \mathbf{Y}_{fi} \begin{bmatrix} \mathbf{Z}_p \\ \mathbf{U}_i \\ \hat{\mathbf{E}}_{i-1} \end{bmatrix}^\dagger \quad (31)$$

2.4 Schätzung der Systemordnung

Die Schätzung der optimalen Systemordnung durch Identifizierung der dominanten Singulärwerte kann sich, besonders in automatisierter Anwendung der Verfahren auf unbekannte Systeme, als unzuverlässig erweisen. Zu diesem Zweck ist die Nutzung eines Informationskriteriums sehr hilfreich. Das in diesem Beitrag verwendete Kriterium ist das Akaike Information Criterion (AIC), welches die logarithmische Likelihoodfunktion des Schätzfehlers berechnet und zusätzlich die Anzahl der geschätzten Systemparameter M_n als Bestrafungsterm nutzt, um eine Überanpassung des Modells durch eine zu hoch gewählte Ordnung zu verhindern.

$$AIC(n) = -2 \log p(\mathbf{Y}^N | \mathbf{U}^N, \hat{\boldsymbol{\theta}}(n)) + 2M_n \quad (32)$$

Die Anzahl der unabhängigen Parameter für ein Modell in Zustandsraumdarstellung ist gegeben durch:

$$M_n = n(2l + m) + lm + \frac{l(l+1)}{2} \quad (33)$$

Da das AIC für eine große Anzahl an Messwerten ausgelegt ist, wird ein folgender Korrekturterm für kleine Stichproben eingeführt.

$$f = \frac{N}{N - \frac{M_n}{n} + \frac{n+1}{2}} \quad (34)$$

Nach [2] kann die Berechnung des AIC für eine gaußverteilte Innovation zu (35) vereinfacht werden.

$$AIC(n) = N(l(1 + \ln(2\pi)) + \ln |\text{cov}(\mathcal{E}_Y)|) + 2M_n f \quad (35)$$

Da das im nächsten Abschnitt beschriebene Mehrkörpersystem durch Newtons Bewegungsgleichungen beschrieben werden kann, ist es möglich die „wahre“ Systemordnung aus dieser Beschreibung zu erhalten und bei der Identifikation zu nutzen.

3 Das Zweiarmsrobotersystem

Als Prozess zur Untersuchung der unterschiedlichen Subspace Verfahren wurde ein Zweiarmsroboter gewählt, wie in Abbildung 1 dargestellt. Der Roboter besteht aus zwei Armen, die über Rotationsgelenke miteinander verbunden sind. Die Arme sind als starre Körper modelliert und besitzen eine individuelle Masse m_i sowie Trägheitstensor I_i in Bezug auf den jeweiligen Schwerpunkt c_i . Die beiden Gelenke sind durch externe Momente τ_i (Eingangsgrößen) angetrieben und die Gelenkwinkel θ_i (Ausgangsgrößen) werden gemessen. Die Nulllage des Systems ($\theta_{1,2} = 0$) ist gegeben, wenn beide Arme in einer Linie stehen und in die negative z-Richtung zeigen. Weitere Bemessungsgrößen sind die Abstände zwischen den Gelenkachsen und den Schwerpunkten der Arme l_1 und l_2 sowie der Abstand der beiden Gelenkachsen a_1 . Der Roboter ist in diesem Experiment so ausgerichtet, dass die Erdbeschleunigung in Richtung der negativen z-Achse wirkt und somit einen Einfluss auf die Lage des Systems hat.

3.1 Kinematik

Die Position der Schwerpunkte in der x-z-Ebene beider Roboterarme in Abhängigkeit der Gelenkwinkel wird durch die Gleichungen (36) und (37) beschrieben.

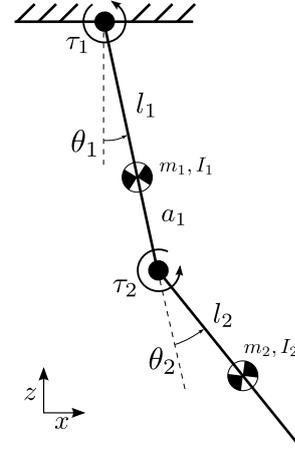


Abbildung 1: schematische Darstellung des Zweiarmsroboters

$$c_1 = \begin{pmatrix} l_1 \sin(\theta_1) \\ l_1 \cos(\theta_1) \end{pmatrix} \quad (36)$$

$$c_2 = \begin{pmatrix} a_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \\ a_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \end{pmatrix} \quad (37)$$

3.2 Dynamik

Das dynamische Verhalten des Robotersystems kann durch die nichtlineare Matrixdifferentialgleichung (38) beschrieben werden. Darin ist $\mathbf{M}(\theta)$ die Massenmatrix in Abhängigkeit der Gelenkwinkel und $\mathbf{V}(\dot{\theta}, \theta)$ eine nichtlineare Vektorfunktion, die die Zentrifugal-, die Coriolis- sowie die Gravitationskräfte enthält. Zusätzlich beschreibt der Term $\mu \dot{\theta}$ die viskosen Reibungskräfte in den Gelenken des Roboters. Für das Aufstellen von \mathbf{M} und \mathbf{V} wurde der Lagrange Formalismus nach [11] und [12] angewandt, woraus sich aus den gegebenen Größen die Gleichungen (39) - (45) für die dynamische Beschreibung des Systems ergeben.

$$\mathbf{M}(\theta)\ddot{\theta} + \mathbf{V}(\dot{\theta}, \theta) = \tau - \mu \dot{\theta} \quad (38)$$

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \quad (39)$$

$$m_{11} = m_2 a_1^2 + 2m_2 \cos(\theta_2) a_1 l_2 + m_1 l_1^2$$

$$+ m_2 l_2^2 + I_1 + I_2 \quad (40)$$

$$m_{12} = m_{21} = m_2 l_2^2 + a_1 m_2 \cos(\theta_2) l_2 + I_2 \quad (41)$$

$$m_{22} = m_2 l_2^2 + I_2 \quad (42)$$

$$\mathbf{V} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (43)$$

$$\begin{aligned} v_1 = & -g m_2 (a_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)) \\ & -g l_1 m_1 \sin(\theta_1) \\ & -\dot{\theta}_2 a_1 l_2 m_2 \sin(\theta_2) (2\dot{\theta}_1 + \dot{\theta}_2) \end{aligned} \quad (44)$$

$$v_2 = -l_2 m_2 (g \sin(\theta_1 + \theta_2) - \dot{\theta}_1^2 a_1 \sin(\theta_2)) \quad (45)$$

4 Simulationsexperiment

Um die beschriebenen Subspace Identifikationsverfahren zu untersuchen, wurde folgendes Experiment für die Simulation aufgestellt. Der Roboterprozess befindet sich in einem geschlossenen Regelkreis, zusammen mit zwei Proportionalreglern, die je einen Gelenkwinkel im Arbeitspunkt stabilisieren sollen. Die Anregung des Prozesses erfolgt durch zwei Pseudorandomsequenzen mit einer Periode von $M = 1023$, die an den Momentenschnittstellen der Gelenke simultan eingekoppelt werden. Zusätzlich werden die Messsignale der Winkelstellung von mittelwertfreien gaußverteilten Rauschsignalen \mathbf{v} mit einer Varianz von 1 überlagert. Weitere Parameter in dem Experiment sind die Abtastzeit mit $T_s = 50ms$ und die Horizonte der Subspace Algorithmen $l_p = l_f = 20$.

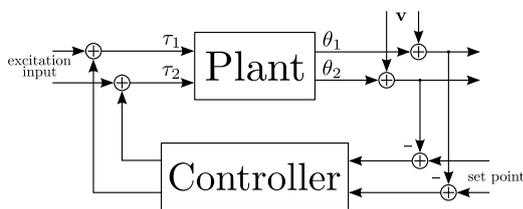


Abbildung 2: Blockschaltbild des Identifikationsexperiments

In dem beschriebenen Experiment werden neun Arbeitspunkte für die Systemidentifikation gewählt, welche in Tabelle 1 aufgezählt werden. Die Simulationen

werden mit Matlab/Simulink und der Toolbox SimMechanics (Bild 3) durchgeführt. Tabelle 1 listet die Achswinkel der einzelnen Arbeitspunkte auf. Für die Auswertung im nächsten Abschnitt werden die Arbeitspunkte 2 und 7 näher untersucht.

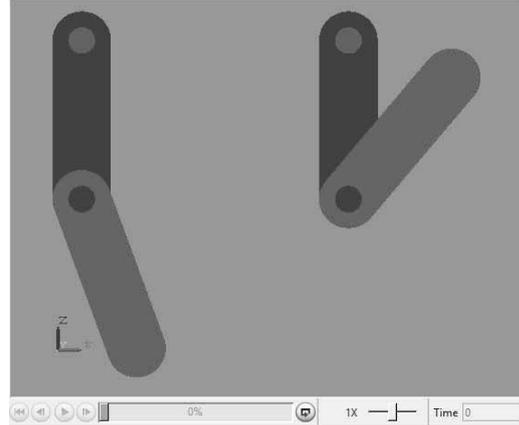


Abbildung 3: Modell des Zweiarroboters in SimMechanics

setpoint	θ_1	θ_2
1	0°	0°
2	0°	20°
3	0°	40°
4	0°	60°
5	0°	80°
6	0°	100°
7	0°	120°
8	0°	140°
9	0°	160°

Tabelle 1: Arbeitspunkte für die Systemidentifikation

5 Simulationsergebnisse

Im Folgenden werden die Ergebnisse des Simulationsexperiments dargestellt und diskutiert. Zunächst wurde mit Hilfe der identifizierten Systemmatrizen ein Beobachter aufgestellt und dessen geschätzte Ausgangsdaten anschließend mit denen des simulierten Prozesses verglichen. Diese Validierung wurde eben-

falls in allen neun Arbeitspunkten durchgeführt. In den Bildern 4 und 5 sind Zeitausschnitte der Vergleichsmessung über 20 Sekunden dargestellt. Der Beobachter ist zu den gezeigten Zeitpunkten bereits eingeschungen. Es ist eine gute Übereinstimmung zwischen den geschätzten und den realen Systemausgängen, sowohl in dem stabilen Arbeitspunkt 2, als auch in dem lokal instabilen Arbeitspunkt 7, zu beobachten. Bild 6 zeigt die Eigenwerte der Systemmatrix in beiden Arbeitspunkten. Es ist zu erkennen, dass von den Identifikationsalgorithmen im Arbeitspunkt 7 ein System mit einem instabilen Eigenwert identifiziert wird.

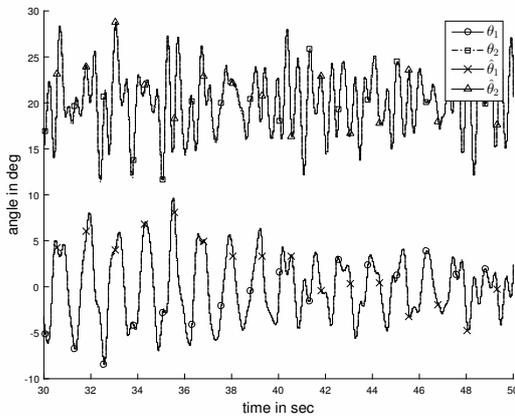


Abbildung 4: Vergleich zwischen System- und Beobachterausgang im Arbeitspunkt 2

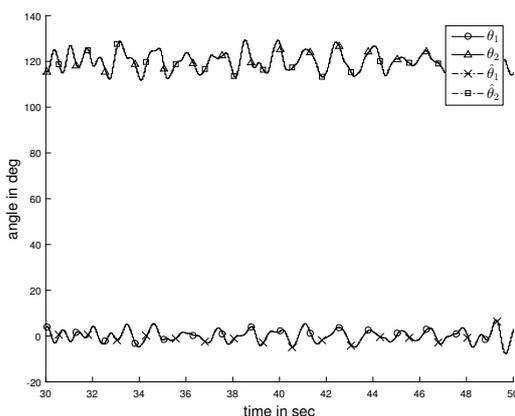


Abbildung 5: Vergleich zwischen System- und Beobachterausgang im Arbeitspunkt 7

Um die Übereinstimmung der identifizierten Modelle für die unterschiedlichen Verfahren zu bewerten,

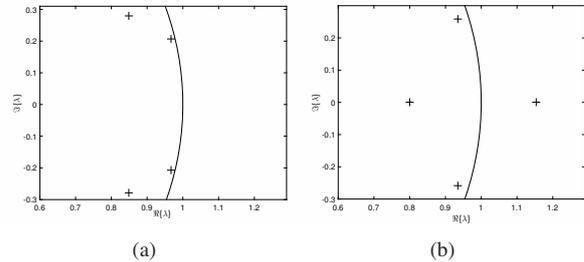


Abbildung 6: Eigenwerte der Systemmatrix für die Arbeitspunkte 2 und 7

wurde der Determinationskoeffizient (46) gebildet. Die Abbildungen 7 und 8 stellen den Determinationskoeffizienten für die drei Identifikationsverfahren in allen neun Arbeitspunkten jeweils für θ_1 und θ_2 dar. Zu beobachten ist, dass das Ein- und Mehrschrittverfahren in jedem Arbeitspunkt bessere Ergebnisse in der Modellübereinstimmung liefern als das Zweischrittverfahren. Eine weitere Auffälligkeit ist, dass die Übereinstimmung in den mittleren Arbeitspunkten geringer ist. Eine mögliche Ursache kann der größere Einfluss der Schwerkraft sein, der bei einem Winkel von $\theta_2 = 90^\circ$ sein Maximum erreicht. Da die Testsignale in jedem Arbeitspunkt die selben Deltamomente für die Anregung nutzen, kann der erhöhte Schwerkrafteinfluss zu einer erhöhten Auslenkung der Roboterarme führen, wodurch ein größerer Arbeitsbereich des nichtlinearen Systems von dem identifizierten linearen Modell abgebildet werden muss. Der selbe Einfluss wirkt ebenfalls bei der Validierungsmessung und beeinflusst das Ergebnis somit in zweifacher Weise.

$$R^2 = 1 - \frac{\sum_{k=1}^N (y_i(k) - \hat{y}_i(k))^2}{\sum_{k=1}^N (y_i(k) - \bar{y}_i)^2} \quad (46)$$

6 Zusammenfassung

In diesem Beitrag wurden zunächst drei unterschiedliche Subspace basierte Systemidentifikationsverfahren erläutert, welche sich für die Identifikation von diskreten linearen Zustandsraummodellen im geschlossenen Regelkreis eignen. Diese Verfahren wurden anschließend an einem Zweiarmeroboter, der ein nicht-lineares Mehrkörpersystem darstellt, angewandt, um in mehreren Arbeitspunkten lokale lineare Systemmo-

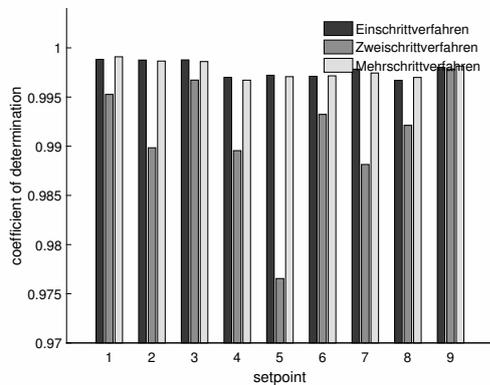


Abbildung 7: Vergleich der R^2 Werte für die unterschiedlichen Algorithmen in den untersuchten Arbeitspunkten für θ_1

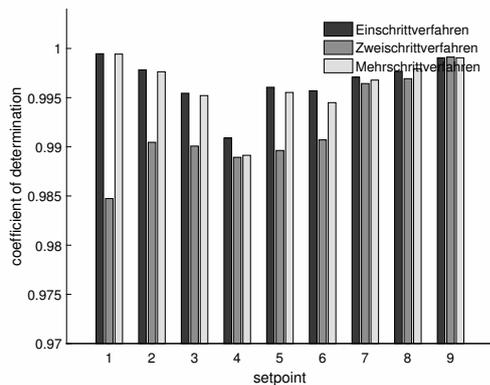


Abbildung 8: Vergleich der R^2 Werte für die unterschiedlichen Algorithmen in den untersuchten Arbeitspunkten für θ_2

delle zu identifizieren. Das Identifikationsexperiment wurde im geschlossenen Regelkreis durchgeführt, um das System in instabilen Arbeitspunkten zu stabilisieren. Die Ergebnisse zeigen, dass jedes der drei genutzten Verfahren in der Lage ist das System sowohl in stabilen als auch in instabilen Arbeitspunkten zu identifizieren.

Literatur

[1] Larimore, W.E., *Canonical Variate Analysis in Identification, Filtering and Adaptive Control*, Proceedings of the 29th Conference on Decision and Control, Hawaii, 1990

[2] Peloubet, R.P. and Haller, R.L. and Bolding, R.M., *On-Line Adaptive Control of Unstable Aircraft Wing Flutter*, Proceedings of the 29th Conference on Decision and Control, Hawaii, 1990

[3] Verheegen, M. and Dewilde, P., *Subspace model identification, Part I: The output-error state space model identification class of algorithms*, Int. J. Control, Vol. 56, 1187-1210, 1992

[4] Van Overschee, P. and DeMoor, B., *Subspace Identification of Linear Systems: Theory, Implementation, Application*, Kluwer Academic Publishers, 1996

[5] Ljung, L. and McKelvey, T., *Subspace identification from closed loop data*, Signal Processing 52, 209-215, 1996

[6] Ljung, L., *System Identification: Theory for the user*, second ed., Prentice-Hall, Inc., 1999

[7] Larimore, W.E., *Automated Multivariable System Identification and Industrial Applications*, Proceedings of the American Control Conference, San Diego, California, 1999

[8] Qin, S.J. and Ljung, L., *Closed-loop Subspace Identification with Innovation Estimation*, Proceedings of SYSID 2003, Rotterdam, 2003

[9] Chiuso, A., *The role of Vector Auto Regressive Modeling in Predictor-Based Subspace Identification*, Automatica, Vol. 43, No. 6, 2007

[10] de Korte, R., *Subspace-Based Identification Techniques for a 'Smart' Wind Turbine Rotor Blade*, Delft University of Technology, M.Sc. Thesis, 2009

[11] Murray, R.M. and Li, Z. and Sastry, S.S., *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994

[12] Craig, J.J., *Introduction to Robotics: Mechanics and Control*, third ed., Pearson Prentice Hall, 2005

Sitzung 10

Raum 116

Grundlagen und Methoden II

(Sitzungsleitung: Christine Wahmkow)

Domain-Specific Languages for Flexibly Experimenting With Models

Tom Warnke

Universität Rostock

Towards a Common Description of Interdisciplinary Aspects Relevant for Holistic Energy

Analysis of Production Facilities

Irene Hafner

dwh GmbH

Design and Optimization of an Energy Manager for an Office Building

Kristin Majetta

Fraunhofer-Institut, Dresden

Domain-Specific Languages for Flexibly Experimenting With Models

Tom Warnke, Danhua Peng, Adelinde M. Uhrmacher
Universität Rostock, Lehrstuhl für Modellierung und Simulation
tom.warnke@uni-rostock.de

Domain specific languages have traditionally been used for model description. In recent years, efforts toward enabling simulation reproducibility have motivated the use of domain specific languages to also express experiment specifications. Domain specific languages serve to specify individual sub-tasks in this process, such as configuration, execution, observation, analysis, and evaluation of experiments. A specific type of analysis, i.e., statistical model checking, can again benefit from domain specific languages. To allow a flexible specification of model behavior, different variants of temporal logic have been developed. In this talk, we will present the combination of two domain specific languages and their usage for statistical model checking.

1 Domain Specific Languages in Modeling & Simulation

The advantages of domain specific languages for model design are well-known. They enable domain experts to build models using the vocabulary of the domain, while hiding implementation details. For example, languages for cell biological models such as ML-Rules adopt a rule-based modeling style that resembles biochemical reaction equations [4], whereas the object-oriented style of Modelica can easily be mapped to components of technical systems [1].

However, domain specific languages can not only be used to create models, but also to specify and execute simulation experiments. The language SESSL (Simulation Experiment Specification via a Scala Layer) [2] constitutes an embedded domain-specific language and exploits SCALA features, such as meta-programming, to allow flexible experiment set-ups. A SESSL specification can incorporate simulation algorithm, model parameters, simulation run time, parallel execution, stopping conditions, replication numbers, observation, result evaluation a.s.o., however only the specification of the model file is mandatory. The actual experiment is then performed with an arbitrary simulation software that is controlled by SESSL based

on a specific binding. Currently, a number of bindings to different simulation systems exist; additional bindings can be added straightforwardly. As SESSL forms an additional software layer between the user and systems used for modeling and simulation, it allows to combine different optimization, simulation and visualizations systems within one experiment [2].

2 Statistical Model-Checking

The term model-checking refers to the automatic verification of systems against a formal specification. Traditional numerical approaches explore the complete state space of a system to prove that it satisfies the specification or find a counterexample. Statistical model-checking techniques employ simulation-based verification [3]. Through experiments they obtain a number of execution traces of the model and check how many of these traces satisfy the specification. Using hypothesis tests it is then possible to determine whether the model satisfies the specification with a probability greater than a given threshold.

The specification to check usually refers to the dynamics of the model and is thus expressed using temporal logics. However, traditional temporal logics are

```

1 import sessl._           // SESSL core
2 import sessl.james._    // JAMES II binding
3
4 execute {
5   new Experiment with Observation with Hypotheses {
6     model = "file:/path/to/the/model"
7     simulator = SimulationAlgorithm
8     scan(
9       "a" <~ range(1, 1, 50),
10      "b" <~ range(0.1, 0.1, 1.0))
11    stopCondition = AfterSimTime(10)
12    observe("output")
13    observeAt(range(0.0, 0.1, 10))
14    assume(Probability >= 0.8)(
15      P(Peak("output", "peakHeight"), time >= 2 and time <= 4, "peakTime"),
16      E(Max("output", "maximumAfterPeak"), end = 10, "afterPeak"),
17      Id("peakTime") STARTS Id("afterPeak"),
18      Id("peakHeight") >= Id("maximumAfterPeak")
19    )
20  }
}

```

Figure 1: SESSL specification of a verification experiment employing statistical model checking. Line 5 states that the experiment will use observation and hypothesis testing. Line 6 specifies the model location, line 7 the simulation algorithm to use. In lines 8-10 the parameter space is defined: For parameter *a* all values between 1 and 50 with a step size of 1 are used, for *b* 0.1 to 1 with steps of 0.1. Thus, 500 parameter combinations are checked. A model variable called *output* is observed from time 0 until the end of the simulation at time 10 with steps of 0.1 (lines 11-13). Finally (lines 14-18), the model specification is formulated: with a probability of at least 0.8, the model variable *output* shall peak between times 2 and 4 and afterwards reach no value greater than the height of the peak until the simulation ends at time 10.

limited when it comes to stochastic models. For example, it is typically possible to state that a property holds during an interval with a probability of 80%, but not that the property holds during 80% of the interval. This is however necessary to deal with models whose outputs are subject to stochastic noise. We propose a novel domain specific language to specify model behavior, based on first-order temporal logic. The language can be combined with SESSL to conduct verification experiments using statistical model checking. Figure 1 shows an example of such an experiment specification. Predicates are always evaluated with respect to time points or intervals; thus, it is possible to take model stochasticity into account. In line 16 of the specification in Figure 1 for example, the maximum of a variable during an interval can be defined to ignore spikes that are significantly shorter than the interval. We have implemented an algorithm to check models for satisfaction of a specification in the new language and are currently evaluating it in a cell-biological use case.

References

- [1] Hilding Elmqvist and Dynasim Ab. Modelica – The Next Generation Modeling Language. An International Design Effort. In *Proceedings of the WCSS '97*, 1997.
- [2] Roland Ewald and Adelinde M. Uhrmacher. SESSL: A Domain-specific Language for Simulation Experiments. *ACM Trans. Model. Comput. Simul.*, 24(2):11:1–11:25, 2014.
- [3] Axel Legay, Benoît Delahaye, and Saddek Bensalem. Statistical Model Checking: An Overview. In *Proceedings of the RV'10*, pages 122–135, Berlin, Heidelberg, 2010. Springer-Verlag.
- [4] Carsten Maus, Stefan Rybacki, and Adelinde M. Uhrmacher. Rule-based multi-level modeling of cell biological systems. *BMC Systems Biology*, 5(1):166, 2011.

Towards a Common Description of Interdisciplinary Aspects Relevant for Holistic Energy Analysis of Production Facilities

Bernhard Heinzl¹, Irene Hafner², Peter Smolek³, Ines Leobner^{3,5}, Georgios Gourlis⁴, Martin Obermair⁵,
Niki Popper², Wolfgang Kastner¹

¹Institute of Computer Aided Automation, Vienna University of Technology, Austria

²dwh GmbH Simulation Services, Austria

³Institute for Energy Systems and Thermodynamics, Vienna University of Technology, Austria

⁴Institute of Interdisciplinary Construction Process Management, Vienna University of Technology

⁵Institute for Production Engineering and Laser Technology, Vienna University of Technology, Austria

bernhard.heinzl@tuwien.ac.at

1 Introduction

As energy and resource efficiency is becoming increasingly important in the industrial sector [1], a demand arises for software tools that provide assistance and decision support for operating a production facility in an energy-efficient manner, including the assessment of optimization potentials and their financial impact [2]. Simulation-based approaches [3] are a way to accomplish this, allowing for example to make qualified short-term predictions of energy demands in multiple future production scenarios.

A challenge in such simulations is however the need to incorporate aspects from different domains, like production machinery [4], logistics, energy infrastructure and even the building, in order to be able to model their dynamic dependencies and interactions and gain insight into additional optimization potentials.

2 Approach

To address the high system complexity and heterogeneity, we follow a systematic approach that divides the overall system from an energetic point of view into well-defined manageable modules, called cubes [5]. These cubes are aimed at integrating different viewpoints and areas of engineering relevant for comprehensive energy analysis (machinery, energy system, logistics and building) into a single generic system description, see Figure 1.

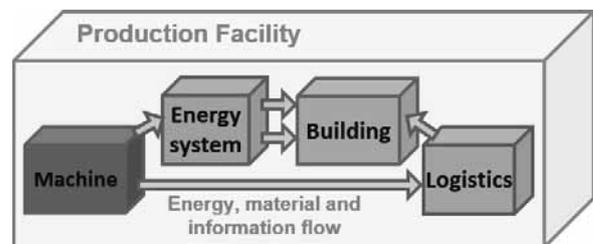


Figure 1. Production facility as interacting cubes

A general specification regarding properties and interfaces of a cube has to be established. Cubes consolidate all information and resource flows (energy, materials, etc.) within identical system boundaries, which not only promotes transparency during simultaneous analysis of energy and material flows, but the obtained modularity also increases flexibility for adaptation to different conditions. Cubes have uniformly and consistently defined interfaces through which they interact with each other by exchanging energy, material and information flows, see Figure 2.

The material flow incorporates the immediate value stream (e.g. work piece, baking goods, etc.) described as discrete entities. All necessary energy flows (electrical, thermal, etc.) are represented as continuous variables and are quantified inside the cube boundaries using balance equations. Information flow represents operating states and monitoring values for the higher-level control as well as control actions for the cube module.

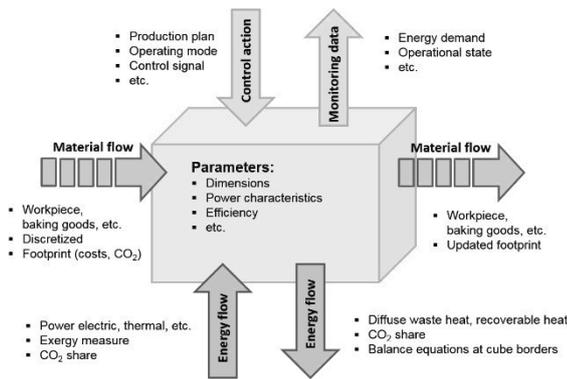


Figure 2. Generic cube interfaces with energy, material and information flows.

Based on the generic interface and attributes definition of a cube, more detailed classes and subclasses can be derived, which are grouped into four domains (see Figure 3) that allow domain-specific focus for a more detailed model description.

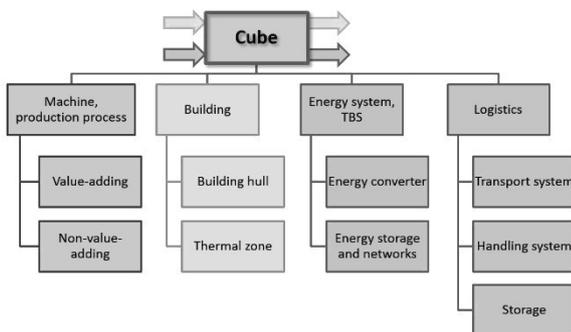


Figure 3. Four classes with various subclasses of cubes.

3 Goal

The cube formalism with its specified interfaces enables analysing the internal behaviour independent from its surroundings. For experimental analysis based on measurement data, cube interfaces can be equipped with measuring devices to detect incoming and outgoing flows.

Simulation models implemented based on the domain-specific cube descriptions [6] provide the core for a software tool chain that will be developed in the scope of the research project BaMa - Balanced Manufacturing [7]. For real-world applications, this tool chain will be coupled with industrial automation systems and thus allow monitoring, simulation-based prediction and optimization of energy and resource demands in

manufacturing plant operations under consideration of the economic success factors time, costs and quality. The tool chain will introduce energy efficiency as an additional steering parameter, enabling manufacturing companies to balance energy efficiency and competitiveness in their continuous operation strategies.

4 References

- [1] R. Neugebauer. *Energieeffizienz in der Produktion: Untersuchung zum Handlungs- und Forschungsbedarf – Abschlussbericht*. Fraunhofer Gesellschaft (FhG), München, 2008.
- [2] K. Bunse, M. Vodicka, P. Schönsleben, M. Brühlhart and F. Ernst. *Integrating energy efficiency performance in production management – gap analysis between industrial needs and scientific literature*. Journal of Cleaner Production, Vol. 19, pp. 667-679, 2011.
- [3] F. Bleicher, F. Duer, I. Leobner, I. Kovacic, B. Heinzl and W. Kastner. *Co-simulation environment for optimizing energy efficiency in production systems*. CIRP Annals – Manufacturing Technology, Vol. 63, pp. 441-444, 2014.
- [4] S. Thiede, G. Bogdanski and C. Herrmann. *A systematic method for increasing the energy and resource efficiency in manufacturing companies*. Procedia CIRP, Vol. 2, pp. 28-33, 2012.
- [5] N. Popper, I. Hafner, M. Rössler, F. Preyser, B. Heinzl, P. Smolek and I. Leobner. *A General Concept for Description of Production Plants with a Concept of Cubes*. SNE Simulation Notes Europe, Vol. 24(2), pp. 105-114, 2014.
- [6] F. Preyser, I. Hafner and M. Rößler. *Implementation of Hybrid Systems Described by DEV&DESS in the QSS Based Simulator PowerDEVs*. In Proceedings of the Workshop of the ASIM/GI Section STS and GMMS, Stralsund, 2015. In press.
- [7] <http://bama.ift.tuwien.ac.at/> (last accessed May 26, 2015).

Design and Optimization of an Energy Manager for an Office Building

Kristin Majetta¹ Christoph Clauss¹ Jürgen Haufe¹ Stephan Seidel¹ 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
{kristin.majetta, christoph.clauss, juergen.haufe, stephan.seidel}@eas.iis.fraunhofer.de

Abstract

Nowadays saving energy in the building sector is more important than ever. To achieve a reduction of energy consumption and also CO₂ emissions of buildings, energy efficient Building Energy Management Systems (BEMS), respectively called energy managers, are developed in the research project enerMAT. Therefore, enerMAT creates 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 optimization referenced to energy uses 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 energy manager in interaction with the building. This system model allows a simulation-based, energy-aware, global, dynamic, multi-criterial optimization of the energy manager. In this paper, the idea, the modeling of an office building and its energy manager and different optimization studies and their results are presented.

Keywords: Building, Energy Management, Green Building Library, FMI, PSO Optimization

1 Introduction

To save energy with the aim to not exhaust the not renewable sources (oil, gas, coal) is uncontradicted-

ly one of the most essential tasks. One of the important energy “consumers” are buildings. For heating, cooling, air conditioning, lightening and personal electronic devices, buildings consume more than one third of primary energy at all [1]. To decrease that part a lot of ideas were created. Besides passive measures (insulation) a promising way is consuming energy in buildings more ingenious than in the past. Energy shall only be consumed if it is absolutely necessary. One way to achieve this is the development as well as the introduction of Building Energy Management Systems (BEMS), shortly spoken, of energy managers. Operating on a runtime system energy managers take information from the building (via sensors) with the goal to calculate the energetically aware actions by controlling e.g. heating, cooling, air conditioning (HVAC).

At the ASIM workshop 2014 in Reutlingen-Rommelsbach we presented an approach of the development of energy managers which is investigated in the research project enerMAT [2]. Starting with an overall model of building, HVAC, ambience and user behavior an energy manager is designed (e.g. using state machines) which possesses some parameters. These parameters are used to adapt the energy manager in such a way that it is optimal regarding a cost function which bases on the energy consumption. The parameter adaption employs optimization techniques. Once the energy manager is developed it can be transformed to a runtime system without any manual interaction.

In the enerMAT research project there are three demonstrator examples: a conference room, an office building, and a residential building. The above mentioned paper covers the conference room,

whereas this paper is focused to the office building demonstrator. The office building (Figure 1) is the headquarters of the enerMAT project partner FASA AG in Chemnitz. By applying the enerMAT approach the development of energy managers is shown.

2 Modeling

The research project enerMAT investigates a model-based approach which means that the development of energy management systems is done under utilization of a building model. Therefore, the modeling of the office building of the project partner FASA AG will be presented in this section.



Figure 1: FASA office building in Chemnitz

The complex model of the FASA office building demonstrator comprises the building “itself”, under-floor heating, heat “generation” (solar panel, boiler, heat pump, stove), control devices (UVR [3]), ambience (weather), user behavior, and the energy manager to be developed.

Modeling bases on the GreenBuilding library [4] which is developed by EA Systems Dresden. The library contains a large set of models for the development of energy systems with energy supply and storage systems, e.g. building component models, heating system models, heat pump models and ambience models. Since the GreenBuilding library is based on the free, object oriented, equation based Modelica language [5] it is possible to develop additional models that are required in the project but are not part of the GreenBuilding library. The models are simulated using SimulationX that is developed by the ITI GmbH.

From an energetic point of view the office building consists of two parts, the energy producing part and

the energy consuming part. The energy producing side (schematic shown in Figure 2) consists of a solar heating system, a stove and a heat pump. All those components are heating up the water inside a 110 m³ buffer. The water of the buffer is temperature depending layered.

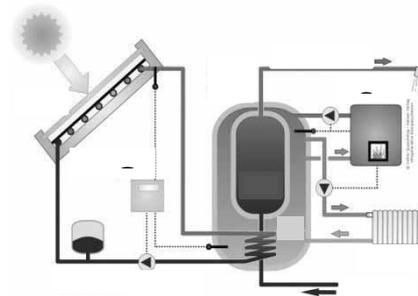


Figure 2: Energy producing part of the office building

The energy generating system is controlled via the UVR1611 (Universalregelung 1611) which is a controller developed by the Austrian company Technische Alternative. To use the functionality of the UVR in the energy supply model, the UVR function blocks were modeled in Modelica and stored in a Modelica library.

On the energy consumption side the warm water of the buffer is used to heat the offices in the building via a floor heating system. The offices are modeled with the BuildingZone model of the GreenBuilding Library. The BuildingZone is a prepared, complex one-point model of a room, which calculates the heat losses taking ambience conditions, heat sources and losses into account. The modeler has to parametrize the zone with e.g. wall properties, room size or geographical direction. Similar office rooms were combined to a bigger building zone. The so developed model of the consumption side contains 25 building zones and needs about 5 hours for a one-year-simulation. Since this is far too long for later optimization, the 25 building zones were combined to two building zones which are representing the ground floor and the first floor of the office building.

The models for the energy generating and the energy consuming part are connected to the complete building model which helps developing the energy manager for the office building.

To round the complete building model off, a model representing the users’ behavior is added. This model gives information about the presence of the employees on working days. This knowledge is used

to specify users thermal comfort ranges that's compliance is ensured by the energy manager.

3 Energy manager development

By means of the complete model of the office building two building energy management systems are developed, one for the energy producing side and one for the energy consuming side. This separation is reasonable since the aim of the energy supply side is to maximize the solar earning over the year and the aim of energy consuming side is to minimize the energy usage.

A BEMS (energy manager) prescribes set points as well as parameters of local controllers by keeping their structures. In the literature several kinds of BEMS are known e.g. rule based BEMS [6], BEMS that are based on artificial neural networks [7], fuzzy logic [8], optimization [9] or ontologies [10] and context-aware BEMS [11]. enerMAT prefers UML statecharts (Unified Modeling Language) as a basic design approach of BEMS since states are induced naturally, like, in case of a temperature, "too cold" or "too hot". Since the aim of the BEMS is controlling the set points of local HVAC devices, at each state of the statechart valid set points have to be calculated.

3.1 Energy manager for energy consumption

Most of the energy is consumed in the office rooms. Since minimizing the whole energy consumption will be achieved by minimizing the energy consumption of a single room, the energy manager is developed for one office room. Two different approaches were investigated.

The first approach is to work with lookup tables that contain all states that the room can have under defined surrounding conditions and values of the controlling variables to achieve a certain goal.

Such a table could, for example, contain temperature ranges for the ambient temperature and for the start value of the room temperature. The aim would be to know when to turn on the heating system to achieve a certain room temperature which means, it must be known how much time the room will need to reach the target temperature. An example is shown in Table 1. Although the values in Table 1 fictitious, they are inspired by real values

which are results of estimating another one-zone-room model developed with the GreenBuilding Library.

Table 1: Example lookup table

Ambient temp.	Start room temp.	Heating up time
< -5 °C	< 0 °C	10 h
	0 °C – 10 °C	8 h
	10 °C – 20 °C	5 h
	> 20 °C	4 h
- 5 °C – 15 °C	< 0 °C	8 h
	0 °C – 10 °C	7 h
	10 °C – 20 °C	4 h
	> 20 °C	3 h
15 °C – 30 °C	< 0 °C	7 h
	0 °C – 10 °C	2 h
	10 °C – 20 °C	1 h
	> 20 °C	30 min
> 30 °C	< 0 °C	3 h
	0 °C – 10 °C	2 h
	10 °C – 20 °C	30 min
	> 20 °C	10 min

During the energy managers' runtime it permanently compares the actual room conditions with the conditions in the lookup table and decides permanently whether to switch on the heating system or not.

The obvious drawback of this method is the huge amount of data that is needed already for relatively small use cases such as the investigated single office room. Besides the ambient temperature and the start value of the room temperature, there is a number of other influence values like the temperature of the adjacent rooms and the supply temperature of the floor heating system. Before developing the lookup table its data items must be known. Therefore a great number of simulations had to be done to investigate the heating up time for each combination of ranges of the influence variables. Since this way of developing an energy manager for the office room seems to be not really feasible, a second investigation was done and is shown in the following.

The simulation of one heating up process of a single office room shows, that the heating curve has the shape of an exponential function $f(t)$ (Figure 3).

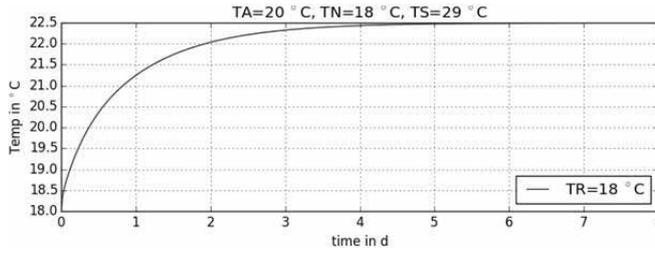


Figure 3: Heating up curve of single office room

$f(t)$ is characterized by its steady-state value g , its start value s and its slope $a(t=0)$ and can be described by (1).

$$f(t) = g - (g - s)e^{\frac{-a}{g-s}t} \quad (1)$$

The variables g and a depend on the supply temperature $T_S(t)$ (TS in Figure 3), the start value of the room temperature $T_R(t)$ (TR in Figure 3), the ambient temperature $T_A(t)$ (TA in Figure 3) and the temperature of the adjacent rooms $T_N(t)$ (TN in Figure 3). For simplicity reasons the temperatures of the neighbor rooms are chosen to be identical. The variables g and a are identified using results from particular simulations with defined values of $T_S(t)$, $T_R(t)$, $T_A(t)$, and $T_N(t)$. Using as an example g , the identification process is shown exemplarily. To identify the dependency of the steady-state value g from $T_A(t)$, $T_N(t)$ and $T_S(t)$, the following linear approach was chosen:

$$g = T_A \cdot x_1 + T_N \cdot x_2 + T_S \cdot x_3 \quad (2)$$

Since (2) has three unknowns (x_1, x_2, x_3), three simulations under defined conditions were undertaken and the linear system $Ax = b$ with

$$A = \begin{bmatrix} T_{A,1} & T_{N,1} & T_{S,1} \\ T_{A,2} & T_{N,2} & T_{S,2} \\ T_{A,3} & T_{N,3} & T_{S,3} \end{bmatrix} \text{ and } b = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}$$

was solved. Using (2) for each value of T_A , T_N , and T_S , g can be calculated. A similar approach is applied for identifying the parameter a . The parameter s does not have to be identified since it is the start value of the room temperature which simply can be taken from the simulation. Knowing g , s and $a(t=0)$, (1) is parameterized during the whole simulation of the room model and calculates the points in time for turning on or off the heating system online by transforming (1) to

$$t_{on,off}(t) = -\frac{g-s}{a} \ln\left(-\frac{w-g}{g-s}\right) \quad (3)$$

where w is the desired target temperature in the room.

The cooling down process of the room after turning off the heating system is also described as an

exponential function in the same way as the heating process is described. For technical reasons, the number of turning on and off the heating system is restricted to one per day. Figure 4 shows the heating and cooling process of the room for a five day periode. In the first two and a half days the heating system is always turned on since the target temperature w (blue dashed curve) is not reached. In the second two and a half days the heating system is turned on and off in a way that the room temperature (red, continuous curve) reaches the target temperature exactly at the beginning of occupancy and falls below the target temperature at the end of occupancy.

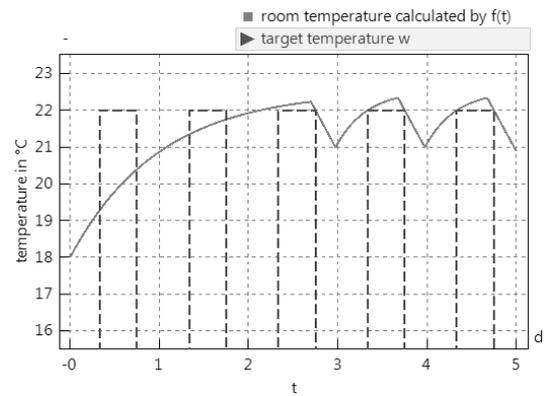


Figure 4: Heating and cooling of the office room (room temperature calculated by $f(t)$)

The switching points for the heating system belong to the room temperature that is calculated by the exponential function (1). Since the aim is to develop an energy manager for a “real” office room of the office building, in the following the calculated switching points are used to control the heating system of the GreenBuilding room model. The results in Figure 6 show, that the room temperature of the GreenBuilding model (black dashed dotted curve) shows the expected behavior by trend. This means, that the heating period in the first two and a half days is nearly the same as for the room temperature calculated by $f(t)$ (continuous red curve) and that also the switching of the heating system can be seen in the second two and a half days. Anyway the room temperatures are not exactly the same. A reason for that is that the exponential function (1) does not exactly represent the heating and cooling process of the room. That is because the slope of the “real” process cannot be represented by simply parameterize the exponential function (1) with a constant slope $a(t=0)$. In fact a slope that varies in time needs to be used.

other energy source (state g) that could be used to heat the water in the buffer.

To calculate the Boolean limit values (limit 1, limit 2) which indicate if there is enough heat in the buffer, a reference temperature T_{ref} is computed (4) with the help of the adjusting parameters c_0, c_1 (that will be optimized later) and the outdoor temperature T_{out} .

$$T_{ref} = \min(80, c_0 + c_1(25 - T_{out})) \quad (4)$$

The Boolean limit values are calculated by comparing T_{ref} with several underfloor heating supply temperatures T_{s1}, T_{s2}, T_{s3} and T_{s4} measured at the tank.

$$\text{limit1} = (T_{s1} < T_{ref} + 5 \text{ and } T_{s2} < T_{ref}) \text{ or } (T_{s3} < T_{ref} + 5 \text{ and } T_{s4} < T_{ref}) \quad (5)$$

Further adjusting parameters c_2, c_3 (that also will be optimized later) are used to calculate the burn time Bt , which is proportional to the mass of wood to be burned:

$$Bt = c_2 + c_3(25 - T_{out}) \quad (6)$$

This developed energy manager for the heat generating side of the office building is simulated and optimized together with the developed building models as shown in section 4.

4 Optimization studies

After the building models and the energy manager are combined to a complete building model, the overall energy consumption is optimized. Therefore, the model is exported as a functional mockup unit (FMU) and connected to a particle swarm optimizer via the functional mockup interface (FMI). Different optimization studies and their results are shown in this section.

As described in section 2, two differently detailed buildings models are available. Those models include, besides the stove, a heat pump which is not controlled by the energy manager. When simulating the more detailed model including the developed energy manager for the first six months of the year using an arbitrarily chosen set of adjusting parame-

ters, the energy supplied by stove and heat pump is calculated (second line in Table 2).

Unfortunately, this simulation takes around 5 hours (used hardware: Windows 7 Enterprise 64bit, Intel® Core™ i7-4600U CPU @ 2.7 GHz, 8 GB RAM). Since this is far too much for the application of optimization methods, the less detailed model is used since its simulation time is around 5 to 10 minutes. The results of this simulation (third line of Table 2) show that the energy consumption of the stove and the heat pump is higher compared to the simulation of the more detailed model. Nevertheless, the less detailed building model is an approximation of the detailed model which describes the building behavior in principle correctly. Since its simulation time is much shorter, it is used for the optimization of the energy manager.

A further acceleration was reached by defining the adjusting parameters c_0 and c_2 to be of the type integer during optimization instead of real. The following intervals for the adjusting parameters were defined heuristically:

$$\begin{aligned} 0 \leq c_0 \leq 30, & \quad 0 \leq c_1 \leq 2 \\ 0 \leq c_2 \leq 20 & \quad 0 \leq c_3 \leq 1 \end{aligned}$$

The objective function is the sum of energy supplied by the heat pump as well as by the stove. Furthermore, penalty terms are added to ensure some temperature restrictions at the buffer.

$$C = E_{\text{stove}} + E_{\text{heat pump}} + \text{penalty} \quad (7)$$

Most important is a low energy consumption of the heat pump since it uses electric power that is expensive compared to the wood for the stove. The optimization results for the parameters and energy consumptions are shown in line 4 in Table 2. It can be seen, that the optimized adjusting parameters lead to a lower energy consumption compared to the simulation of the less detailed model with arbitrary parameters (line 3 of Table 2)

The optimized adjusting parameters are now used to simulate the original, more detailed model (line 5 in Table 2) and the results show that the optimized parameters of the less detailed model also lead to better results in the more detailed model (second line in Table 2). This gives the conclusion, that the model simplifications that led from the more detailed model to the less detailed one were allowable.

Table 2: Optimization studies

	c_0	c_1	c_2	c_3	Energy stove in kWh	Energy heat pump in kWh
detailed model	30	1	18	0.5	115	0
less detailed model	30	1	18	0.5	425	1421
less detailed model optimized	30	1.97	20	1	310	626
more detailed models with optimized parameters	30	1.97	20	1	94	0

As could be seen during the optimization work, the performance of one simulation run can be quite bad since building models naturally are relatively extensive. Therefore, it is often too time consuming to finish optimization runs within reasonable time. To overcome these difficulties, several strategies can be applied:

- Take simplified models instead of accurate ones.**
It is possible to over-simplify models. Therefore, it needs to be investigated, how simple a model can be to generate still reasonable optimization results. At least the following consideration is useful: If simplified models are used for optimization, the parameters obtained from optimization should be applied to the more accurate model. Such a verification simulation should prove that the parameters still produce good results.
- Reduce the number of parameters during optimization.**
A low number of parameters is advantageous for optimization runs. Parameters which have not a great influence on the optimization result can be identified by performing a sensitivity analysis.
- Chose suitable time intervals for the parameter optimization.**
Often some states become active within dedicated time intervals only. Therefore, it is sometimes possible to optimize groups of parameters separately within shorter time intervals. This improves the performance. E.g. typically parameters which influence heating devices should not be optimized within summer months.

- Do not simulate unreliable parameter values.**
Sometimes, the optimization algorithm choses unreliable as well as not realistic parameter constellations, which cause bad performant simulations or simulation crashes. Such obviously bad parameter constellations should be selected before simulation. This selection can be included into the simulation model.
- Use parallelization.**
Both optimization as well as simulation can be accelerated by parallelization.
- Choose good process parameters of the optimization method.**
The optimization method can be adapted by suitable optimization process parameters. This influences the performance of optimization drastically.

5 Summary

During the enerMAT project different BEMS respectively energy managers were developed. Since enerMAT follows a model based approach, the modelling of the FASA office building (two differently detailed models) was introduced. Additionally the development of two energy managers for the FASA office building was presented in this paper. The energy manager for the energy generating part is developed as a statechart with adjusting parameters that are optimized to reach minimal energy consumption. The energy manager for the heat consuming part was developed by process identification of the heating and cooling process using the example of one room. A bottleneck in this process is the often bad simulation and optimization performance. Therefore some strategies are discussed to improve the performance.

The potential of this approach is by far not exhausted yet: The optimization can comprise both adjusting parameters of the energy manager as well as constructing parameters of the building. Thus more common target functions are possible. Other investigations are necessary to support the development of the statechart.

Acknowledgement

The research is funded by	<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] Clauss, C.; Haufe, J.; Blochwitz, T.; et. al.: Model Based Optimization of Building Control Systems. Proceedings of the ASIM-Workshop STS/GMMS 2014, pp.131-138., February 20-21, 2014, Reutlingen-Rommelsbach, Germany.
- [3] Website of the Austrian company "Technische Alternative" : <http://www.ta.co.at/de/produkte/uvr1611/frei-programmierbare-regelung-uvr1611.html>
- [4] Unger, R.; Schwan, T.; Mikoleit, B.; et. al.: „Green Building“ – Modelling renewable building energy systems and electric mobility concepts using Modelica. Proceedings of the 9th International Modelica Conference, pp. 897-906, September 3-5, 2012, Munich, Germany.
- [5] www.Modelica.org
- [6] Doukas, H.; Patlitzianas, K. D.; Iatropoulos, K.; et al.: Intelligent building energy management system using rule sets. Building and Environment, Volume 42, Issue 10, pp. 3562-3569, October 2007.
- [7] Kalogirou, S.A.: Applications of artificial neural-networks for energy systems. Applied Energy, Vol. 67, pp. 17-35, 2000.
- [8] Kolokotsa, D.; Stavarakakis, G.S.; Kalaitzakis, K.; et al.: Generic algorithms optimized fuzzy controller for the indoor environmental management in buildings implemented using PLC and local operating networks. Engineering Applications of Artificial Intelligence, Volume 15, Issue 5, pp. 417-428, 2002.
- [9] Castilla, M.; Álvarez, J.D.; Berenguel, M.; et. al.: A comparison of thermal comfort predictive control strategies. Energy and Buildings, Vol. 43, pp. 2737-2746, 2011.
- [10] Rosselló-Busquet, A.; Brewka, L. J.; Soler, J.; et. al.: OWL Ontologies and SWRL Rules Applied to Energy Management. UKSim 13th International Conference on Modelling and Simulation. pp. 446-450, Emmanuel College Cambridge, 2011, United Kingdom.
- [11] Byun, J.; Park, S.: Development of a Self-adapting Intelligent System for Building Energy Saving and Context-aware Smart Services. IEEE Transactions on Consumer Electronics, Vol. 57, No.1, 2011.

Sitzung 11

Raum 111

Elektrotechnische Systeme

(Sitzungsleitung: Peter Roßmanek)

*Modellbasierte Entwicklung des elektrischen Energiemanagements für Elektrofahrzeuge mit
einem Zweispannungsbordnetz*

Sören Scherler

Hochschule Ostfalia

Effiziente Modellierung heterogener Batteriemanagementsysteme (BMS)

Thomas Markwirth

Fraunhofer-Institut, Dresden

Determination of Radiation Models Based on Measured Electromagnetic Field Distribution

Zongyi Chen

TU Dortmund

Optimierung des Regelalgorithmus zur aktiven Fahrzeugfederung für Elektrofahrzeuge

Matthias Fritsch

Hochschule Ostfalia

Modellbasierte Entwicklung des elektrischen Energiemanagements für Elektrofahrzeuge mit einem Zweispannungsboardnetz

Sören Scherler, Florian Quantmeyer, Xiaobo Liu-Henke
Ostfalia Hochschule für angewandte Wissenschaften, Fakultät Maschinenbau,
Institut für Mechatronik, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel
so.scherler@ostfalia.de

Im vorliegenden Beitrag wird die modellbasierte Entwicklung eines elektrischen Energiemanagements (EEM) für Elektrofahrzeuge mit einem Zweispannungsboardnetz dargestellt. Das EEM sorgt für eine bedarfsgerechte und prioritätsbasierte Verteilung der im Elektrofahrzeug begrenzt verfügbaren elektrischen Leistung auf die elektrischen Verbraucher bei gleichzeitiger Gewährleistung von Fahrsicherheit und möglichst nicht wahrnehmbaren Einbußen im Fahrkomfort. Grundlage für dieses Verfahren sind zentrale Leistungskordinatoren für beide Spannungsebenen sowie dezentral verteilte Energieassistentenfunktionen (EAF). Die Leistungskordinatoren erhalten von den EAF Informationen über das Leistungsangebot der Energiespeicher und die Leistungsbedarfe der elektrischen Verbraucher, verarbeitet diese und gibt an die EAF Schaltfreigaben für die Verbraucher zurück. Das EEM steuert zudem, abhängig von Last und Ladestand der Batterie der Nieder-
voltebene, den unidirektionalen Tiefsetzsteller, welcher zur Leistungsübertragung von der Hochvolt- in die Nieder-
voltebene dient.

1 Einleitung

Das Thema Elektromobilität rückte in den vergangenen Jahren immer stärker in den Fokus der Öffentlichkeit und der Automobilindustrie.

Eine große Herausforderung bei der Entwicklung von Elektrofahrzeugen stellen die, im Vergleich zu fossilen Energieträgern, geringen Energiedichten und -kapazitäten der elektrischen Energiespeicher dar, aus denen die momentan noch relativ geringen Reichweiten von Elektrofahrzeugen resultieren. Die Reichweite allerdings ist ein entscheidendes Kaufargument, sodass Elektrofahrzeuge momentan gegenüber konventionell betriebenen Fahrzeugen für den Käufer weniger attraktiv sind. Deshalb werden derzeit die verschiedensten Ansätze zur Reichweitenerhöhung wie Start-Stopp-Systeme oder rekuperatives Bremsen entwickelt.

Das Problem der relativ geringen Energiespeicherkapazitäten wird durch den steigenden elektrischen Leistungsbedarf im Zuge der fortschreitenden Elektrifizierung von Fahrzeugfunktionen noch vergrößert. Der steigende Leistungsbedarf ist zum einen auf den Einsatz von „X-by-Wire“-Technologien, bei welchen mechanische durch elektrische Systeme ersetzt werden (z. B. „Steer-by-Wire“ oder „Brake-by-Wire“), und zum anderen auf den steigenden Anteil von Fahrerassistenz- und Komfortsystemen zurückzuführen. Um dem Leistungsbedarf möglichst aller Verbraucher im System gerecht zu werden und dabei den Ladestand der Batterie zu berücksichtigen, müssen Strategien

zum intelligenten Verteilen der begrenzt vorhandenen Leistung entwickelt und umgesetzt werden.

Die modellbasierte Entwicklung einer solchen Strategie, einem elektrischen Energiemanagement (EEM), wird in diesem Beitrag dargestellt.

2 Methodik

Das EEM ist informationstechnisch mit einem übergeordneten Fahrzeugmanagement und unterlagerten Systemen wie dem Batteriemanagementsystemen oder den elektrischen Verbrauchern gekoppelt.

Um die Systemkomplexität bei der Entwicklung dieses hochgradig vernetzten Systems zu beherrschen, wird das Gesamtsystem gemäß der mechatronischen Methodik [1] modular und hierarchisch strukturiert, sodass die gekapselte Entwicklung einzelner Funktionen möglich ist [2].

Die einzelnen Funktionen werden modellbasiert ausgelegt und in einem durchgängigen Prozess basierend auf Model-in-the-Loop-(MiL), Software-in-the-Loop-(SiL) und Hardware-in-the-Loop-(HiL)-Simulationen validiert und optimiert. Durch diesen Prozess finden bereits in frühen Entwicklungsphasen Erprobungen an Prototypen (u.a. Modellen) statt, sodass die Funktion frühzeitig abgesichert werden kann.

In diesem Beitrag wird die MiL/ SiL-Simulation im Rahmen der modellbasierten Entwicklung eines EEM für Elektrofahrzeuge dargestellt.

3 Energiemanagementstrategie

In diesem Abschnitt werden der Algorithmus des EEM und seine Bestandteile beschrieben.

3.1 Fahrzeugkonfiguration

Das diesem Beitrag zugrunde liegende Elektrofahrzeug ist in **Abbildung 1** schematisch dargestellt. Das Fahrzeug verfügt über eine Steer-by-Wire-Allradlenkung und radindividuelle Direktantriebe. Die Antriebsmaschinen dienen zum einen der Traktion und zum anderen der Rekuperation, also dem Bremsen mit den Antriebsmaschinen bei gleichzeitiger Stromrückspeisung in das Bordnetz. Bei Überschreiten der maximalen Rekuperationsleistung der Antriebsmaschinen können konventionelle Reibbremsen zugeschaltet werden.

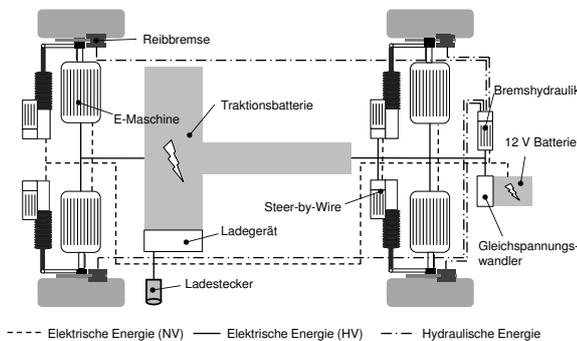


Abbildung 1. Fahrzeugkonfiguration

3.2 Zwei-Spannungs-Bordnetz

Für die beschriebene Fahrzeugkonfiguration ist ein Zwei-Spannungs-Bordnetz (vgl. Abbildung 2) zwingend erforderlich. Von den Antriebsmaschinen werden Leistungen benötigt, welche ein konventionelles 12V-Bordnetz aufgrund der Strombelastung nicht aufbringen könnte. Aus diesem Grund wird für die Antriebsmaschinen eine Spannungsebene von 350V vorgesehen, wodurch die Strombelastung nach dem

Ohm'schen Gesetz deutlich reduziert wird. Nachteilig sind an dieser hohen Spannung allerdings erhöhte Kosten für erforderliche Sicherheitsmaßnahmen des elektrischen Berührungsschutzes, weshalb möglichst viele Verbraucher weiterhin aus dem konventionellen 12V-Bordnetz versorgt werden, da für dieses keine Maßnahmen vorgesehen werden müssen.

Neben den Verbrauchern sind in beiden Spannungsebenen Energiespeicher (Lithium-Ionen-Zellen im Hochspannungs- und Bleiakumulatoren im Niederspannungsnetz) vorgesehen.

Verbunden sind die beiden Spannungsebenen durch einen unidirektionalen Tiefsetzsteller. Dieser kann Leistung von der Hochvoltebene in die Niedervoltebene übertragen. Die Niedervoltebene wird nicht nur durch den herkömmlichen Blei-Akkumulator, sondern auch durch den Tiefsetzsteller mit Leistung versorgt. Zum Laden der Hochvolt-Energiespeicher ist ein Lader vorgesehen.

3.3 Anforderungen an das EEM

Die zentrale Aufgabe des EEM ist die bedarfsgerechte Verteilung der verfügbaren elektrischen Leistung bei gleichzeitiger Gewährleistung von Fahrsicherheit und möglichst nicht wahrnehmbaren Einbußen im Fahrkomfort. Hierzu werden an das EEM zahlreiche Anforderungen gestellt:

- Gewährleistung sicherheitsrelevanter Funktionen bei Leistungsengpässen (ggf. durch Abschaltung nicht sicherheitsrelevanter Verbraucher).
- Möglichst nicht wahrnehmbare Abschaltung von (Komfort-)Verbrauchern bei Leistungsengpässen.
- Möglichkeit zur Leistungsübertragung aus dem Hochvoltnetz in das Niedervoltnetz unter Verwendung des Tiefsetzstellers.

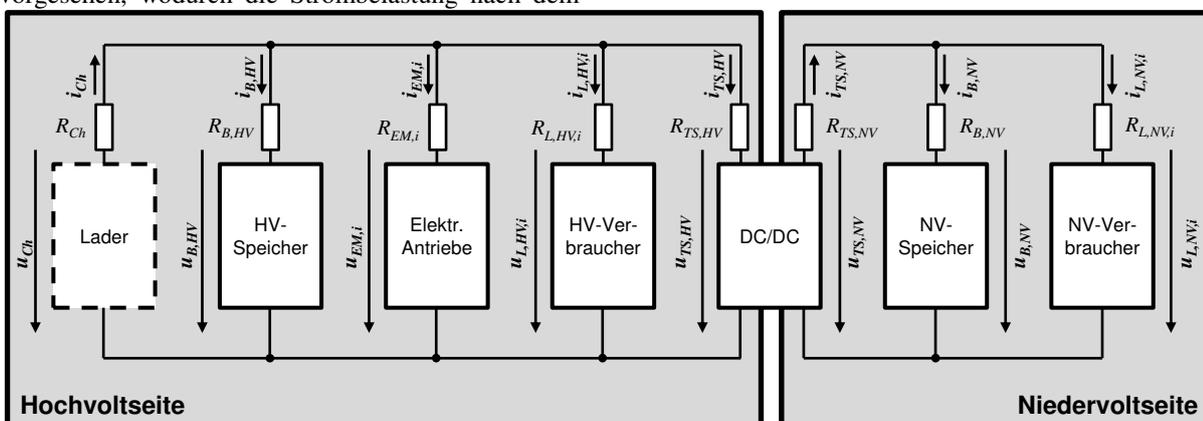


Abbildung 2. Zwei-Spannungs-Bordnetz

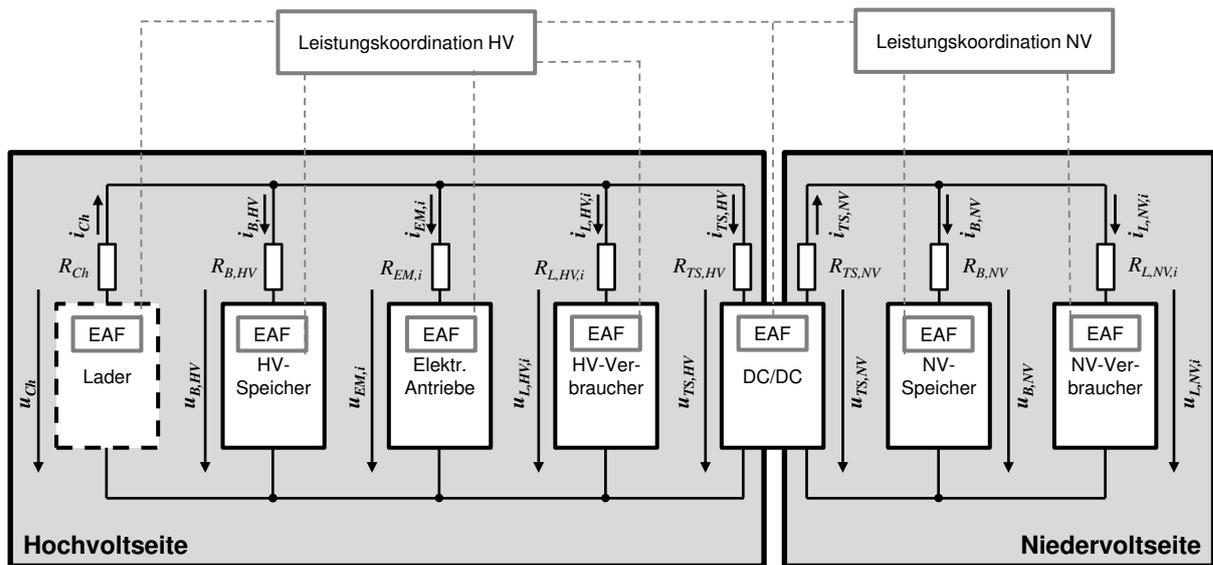


Abbildung 3. Zwei-Spannungs-Bordnetz erweitert um Komponenten des EEM

- Möglichkeit zum Laden des Hochvoltspeichers am Stromnetz.
- Begrenzung der maximalen Rekuperationsleistung abhängig von Stromlast und Ladezustand der Speicher.
- Kopplung des EEM an eine globale Fahrzeugregelung zur Begrenzung der Stellgrößen.
- Stabilisierung der Bordnetzspannung im NV-Netz in einem Bereich von 11 bis 13,5 V.
- Verhindern der Tiefentladung von HV- und NV-Batterien, um irreparable Schäden zu vermeiden.
- Schutz der Batterien vor Überlast (max. 200 A dauerhaft).

3.4 Stand der Technik

Es kann unterschieden werden zwischen zentralen und dezentralen EEM-Strategien.

Büchner [3] beschreibt zwei zentrale Ansätze für konventionelle Fahrzeuge, bei denen EEM wirtschaftswissenschaftlich entweder als Markt oder Auktion betrachtet wird. Ein zentrales Steuergerät ermittelt aus Angebot und Nachfrage bzw. Geboten, welche Verbraucher mit Leistung versorgt werden.

Schmidt [4] beschreibt ebenfalls einen zentralen Ansatz für konventionelle Fahrzeuge, bei welchem den Verbrauchern und Energiequellen feste Prioritäten zugeordnet werden. Mit diesen Prioritäten und ihrem

Leistungsbedarf oder -angebot melden sie sich bei einem zentralen Steuergerät an, welches basierend auf einer Matrix ermittelt, welche Verbraucher mit Leistung versorgt werden.

Heitel et al. [5] beschreiben hingegen einen dezentralen Ansatz angelehnt an die Agentenorientierte Softwareentwicklung. Demnach werde jeder Verbraucher durch einen Agenten vertreten und durch die Verhandlung dieser Agenten miteinander wird gemeinsam über die zu schaltenden Verbraucher entschieden.

3.5 Aufbau des EEM

Für das vorliegende Elektrofahrzeug wird die zentrale Strategie mit fester Priorisierung von Verbrauchern vorgesehen. Der Ansatz nach Büchner wird umfangreich und ganzheitlich erweitert, sodass die für ein konventionelles Bordnetz entworfene Strategie für ein Zwei-Spannungs-Bordnetz des Elektrofahrzeugs verwendet werden kann.

Den Verbrauchern und Speichern werden Klassen (KL) und Prioritätskennzahlen (PK) zugeordnet (im Folgenden wird unter Priorität die Kombination aus KL und PK verstanden). Ein Leistungskordinator pro Spannungsebene addiert ihre Leistungen in der Reihenfolge ihrer Priorität auf [6]. Übersteigt der Leistungsbedarf die verfügbare Leistung, werden die Verbraucher mit geringerer Priorität deaktiviert.

Die Informationen über die Zuordnung der Prioritäten liegen dezentral bei den Verbrauchern und Speichern und werden mithilfe der Energieassistentzfunktionen (EAF) verwaltet [7]. Die EAF dienen u. a. der Priorisierung der Verbraucher und Speicher, der Übermittlung

lung der Leistungsbedarfe/-angebote und dem Schalten der Verbraucher. Dies eröffnet eine hohe Flexibilität für den Einsatz bei einer hohen Variantenvielfalt, da zusätzliche Verbraucher einfach durch Vergabe einer Priorität in das System eingebunden werden können.

Eine Erweiterung des Bordnetzes um die Komponenten des EEM ist in **Abbildung 3** dargestellt. Für beide Spannungsebenen wird je eine zentrale Leistungskoordination vorgesehen. Diese erhält Informationen der EAF über die Leistungsbedarfe und -angebote der jeweiligen Spannungsebene und verarbeitet diese. Daraufhin gibt sie die Priorität, bis zu welcher alle Elemente geschaltet werden dürfen, an die EAF zurück, welche abhängig von dieser die Elemente des Bordnetzes schalten.

3.6 Algorithmus des Leistungskordinators

Dem Leistungskordinator liegt eine Matrix, die Leistungsmatrix, zugrunde, in welcher alle Leistungen der Verbraucher und Speicher anhängig von KL und PK eingetragen werden.

Eine Leistung ist wichtiger, je kleiner die zugehörige Klasse ist. Innerhalb der Klassen ist eine Leistung wichtiger, je kleiner die Prioritätskennzahl ist. Eine Leistung mit KL1 und PK2 ist demnach wichtiger als eine Leistung mit KL2 und PK1 und eine Leistung mit KL1 und PK2 ist wichtiger als eine Leistung mit KL1 und PK3. Die freizugebenden Elemente (bis einschließlich einer Grenzklasse KL_{Grenz} und einer Grenzprioritätskennzahl PK_{Grenz}) werden ermittelt, indem innerhalb einer Zeile die Leistungen von links nach rechts aufsummiert werden. Bei Zeilenende wird links in der nächsten Zeile weitergemacht. Diese Berechnung wird durchgeführt bis entweder die kumulierte Leistung größer als 0 wird oder bis alle Elemente aufsummiert wurde.

Ein Beispiel ist in **Abbildung 4** gegeben.

	PK1	PK2	PK3
KL1	-800 (-800)	-150 (-950)	0 (-950)
KL2	600 (-350)	300 (-50)	100 (50)

Abbildung 4. Exemplarische Leistungsmatrix mit 2 Klassen und 3 Prioritätskennzahlen. In Klammern die kumulierte Leistung.

Verfügbare Leistungen werden negativ und benötigte Leistung positiv berücksichtigt. Zusätzlich zu den Leistungen ist in Klammern die kumulierte Leistung nach Aufsummieren des jeweiligen Elements aufgeführt. In diesem Beispiel werden alle Leistungen bis

KL2 und PK2 freigeschaltet, da in diesem Element die kumulierte Leistung zum letzten Mal negativ ist, was bedeutet, dass die verfügbare letztmalig größer als die angeforderte Leistung ist.

3.7 Energieassistentzfunktionen

Für Verbraucher und Speicher werden unterschiedliche EAF vorgesehen.

Die EAF der Speicher sind Bestandteile des Batteriemanagementsystems (BMS), welches u.a. zur Überwachung und zum Ladeausgleich von Lithium-Ionen-Zellen benötigt wird. Mithilfe eines an der Ostfalia entwickelten Ladezustandsschätzers und einem Prädiktionsalgorithmus werden der Ladezustand des Batteriepakets sowie die maximale Abgabeleistung berechnet [8]. Diese Informationen werden an die Leistungskordinatoren weitergegeben. Das BMS kann mithilfe eines Relais' das Batteriepaket vom Netz trennen, um es z.B. vor Überlast oder Tiefentladung zu schützen.

Die EAF der Verbraucher müssen neben der Ermittlung und Kommunikation der Leistungsbedarfe und dem Schalten der Verbraucher auch externe Signale (z.B. Fahrerwünsche) auswerten können. Weiterhin sorgen sie für eine dynamische Priorisierung wahrnehmbarer Verbraucher. Bei einem Leistungsengpass werden Verbraucher mit niedriger Priorität deaktiviert. Wenn diese Deaktivierung für den Fahrer wahrnehmbar ist, wie z.B. die Abschaltung der Sitzheizung, führt die Deaktivierung zu signifikanten Komforteinbußen. Zur Vermeidung dieser Einbußen wird der deaktivierte Verbraucher, in diesem Beispiel die Sitzheizung, bei Überschreiten einer Wahrnehmbarkeitsschwelle kurzzeitig höher priorisiert, sodass ihm Leistung zugeteilt wird. Bei Erreichen einer definierten Grenze wird der Verbraucher wieder in seine Ursprungspriorität gestuft. Durch Ausnutzung der hohen Trägheit thermischer Systeme dauert es einen relativ langen Zeitraum, bis die Wahrnehmbarkeitsschwelle wieder erreicht wird.

Der Vorgang der dynamischen Priorisierung wird an einem Beispiel erläutert (vgl. **Abbildung 5**).

	PK1	PK2	PK3
KL1	-1000 (-1000) -1000 (-1000)	0 (-1000) 0 (-1000)	0 (-1000) 0 (-1000)
KL2	0 (-1000) 500 (-500)	750 (-250) 750 (250)	500 (250) 0 (250)

↑ Grenzelement bei hoher Priorisierung
↑ Grenzelement bei niedriger Priorisierung

Abbildung 5. Exemplarische Leistungsmatrix mit dynamischer Priorisierung

In der ersten Zeile ist ein Szenario mit einem Speicher (negative Leistung 1000 W in KL1/PK1) sowie zwei Verbrauchern (750 W in KL2/PK2 sowie 500 W in KL2/PK3). Die Leistung reicht nicht für beide Verbraucher, sodass lediglich alle Verbraucher bis einschließlich KL2/PK2 freigegeben werden. Dadurch erhält der Verbraucher in KL2/PK3 keine Leistung. Bei Erreichen der Wahrnehmbarkeitsschwelle wird er in KL2/PK1 gestuft, sodass er auf Kosten des Verbrauchers in KL2/PK2 mit Leistung versorgt wird.

4 Modellierung und Implementierung

In diesem Abschnitt wird die Modellierung und Implementierung der elektrischen Verbraucher und Speicher dargestellt, da diese zur Erprobung des EEM benötigt werden.

4.1 Automotive Simulation Models

Um das EEM zu erproben wird es in ein Gesamtfahrzeugmodell implementiert. Verwendet werden die Automotive Simulation Models (ASM) der Firma dSpace, bei denen es sich um echtzeitfähige Fahrdynamikmodelle handelt, welche in der Softwareumgebung von Matlab/Simulink ausgeführt werden. Basierend auf einem Mehrkörpersystem mit 24 Freiheitsgraden werden die physischen Eigenschaften des Fahrzeugs dargestellt.

Zusätzlich zum EEM wird das gesamte Zweispannungs-Bordnetz implementiert.

4.2 Batterie

Bei der Batterie handelt es sich um Lithium-Ionen-Batterien, welche mithilfe der Elektrochemischen Impedanzspektroskopie (EIS) im Frequenzbereich identifiziert wurden [9].

Eine für die Batterie wesentliche Größe ist ihr Ladezustand (State of Charge, SoC). Dieser ergibt sich aus einem Start-SoC S_{oC_0} , dem effektiven Strom $i(t)$, einem Coulombschen Wirkungsgrad η und der Zellkapazität C_n nach (1).

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

Für die Batterie wurde ein elektrisches Ersatzschaltbild (vgl. **Abbildung 6**) bestehend aus einer Spannungsquelle (Abbildung der Open-Circuit-Voltage, OCV, Ruhespannung), einem Widerstand R_S sowie 4 RC-Gliedern zur Abbildung der Dynamik.

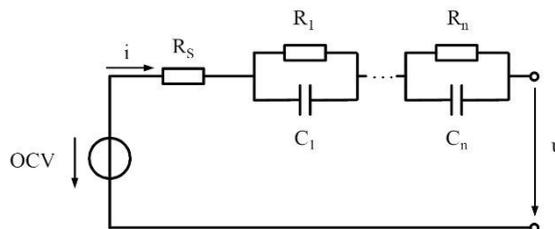


Abbildung 6. Elektrisches Ersatzschaltbild der Batterie

Mathematisch lässt sich der Frequenzgang laut [9] nach (2) beschreiben.

$$\frac{u(j\omega)}{i(j\omega)} = R_S + \frac{R_1}{R_1 C_1 j\omega + 1} + \dots + \frac{R_n}{R_n C_n j\omega + 1} \quad (2)$$

Der Frequenzgang ist vom Ladezustand der Zelle abhängig. Dementsprechend wurde der Frequenzgang für verschiedene SoC gemessen (vgl. **Abbildung 7**).

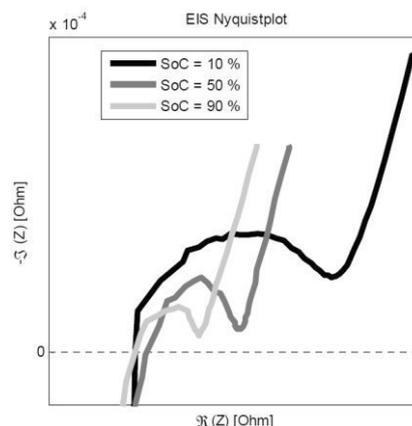


Abbildung 7. Messergebnis der EIS [9]

Zur Berücksichtigung des nichtlinearen Verhaltens der Ruhespannung abhängig vom SoC wurden Messungen im Zeitbereich durchgeführt. Das Ergebnis der Messungen zeigt **Abbildung 8**.

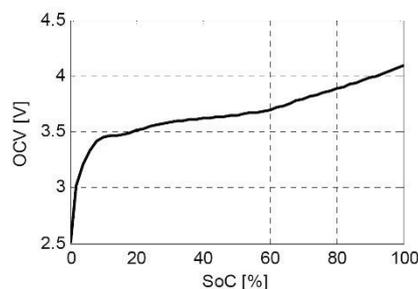


Abbildung 8. OCV-Kennlinie [9]

Das Modell der Batterie liefert abhängig vom Bordnetzstrom eine Spannung und ändert seinen SoC,

sodass das Batterieverhalten hinreichend gut wiedergegeben wird.

4.3 Verbraucher

Für die Verbraucher werden zum einen elektrische Modelle nach Gl. (3) aufgestellt, welche abhängig von der Leistung P des Verbrauchers, der Bordnetzspannung U und dem Schaltsignal z einen Strom liefert, welcher die Batterie belastet. Das dynamische Verhalten durch den Einfluss von Induktivitäten oder Kapazitäten wird vernachlässigt.

$$I = \frac{P}{U} \cdot z \quad (3)$$

Zum anderen werden thermische Modelle für Heizelemente implementiert. **Abbildung 9** zeigt ein thermisches Ersatzschaltbild.

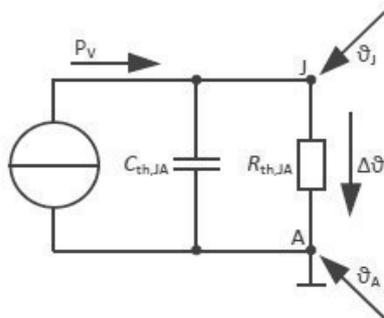


Abbildung 9. Thermisches Ersatzschaltbild für Heizelemente

Die Verbraucher werden als Parallelschaltung einer thermischen Kapazität $C_{th,JA}$ und eines thermischen Widerstandes $R_{th,JA}$ betrachtet. Die Temperatur ϑ ergibt sich nach Gl. (4) aus dem Integral der zugeführten Leistung P_V abzgl. der am thermischen Widerstand verlorenen Leistung $\Delta\vartheta/R_{th,JA}$.

$$\vartheta(t) = \frac{1}{C_{th}} \int_0^t \left(P_V - \frac{\Delta\vartheta}{R_{th}} \right) dt + \vartheta_A \quad (4)$$

Durch die Implementierung der thermischen Modelle kann das Energiemanagement die Leistung von Heizelementen dynamisch priorisieren. Wenn bspw. ein Heizelement eingeschaltet werden soll, aber für dieses keine Leistung zur Verfügung steht, wird die Temperatur den vom Fahrer gewünschten Sollwert nicht erreichen. In diesem Fall wird das Heizelement eingeschaltet bis die Solltemperatur erreicht wird.

4.4 Tiefsetzsteller

Der Tiefsetzsteller dient der Übertragung von Leistung des Hochvolt- in das Niederspannungsnetz. Er wandelt die elektrische Leistung P_{HV} mit der Spannung der

Hochvoltebene verlustbehaftet mit dem Wirkungsgrad η in eine elektrische Leistung P_{NV} mit der Spannung der Niedervoltebene Gl. (5).

$$P_{NV} = \eta \cdot P_{HV} \quad (5)$$

Der Wirkungsgrad wiederum hängt zum einen von der Spannung der Hochvoltebene u_{HV} und zum anderen von der zu übertragenden Leistung P_{HV} ab. **Abbildung 10** aus [10] zeigt ein Wirkungsgradkennfeld für einen Tiefsetzsteller. Für eine Spannung von 350 V sind abhängig von der zu übertragenden Leistung Wirkungsgrade zwischen 0,76 und 0,87 erreichbar. In Matlab/Simulink wurde das Wirkungsgradkennfeld als Matrix in einem Lookup-Table hinterlegt.

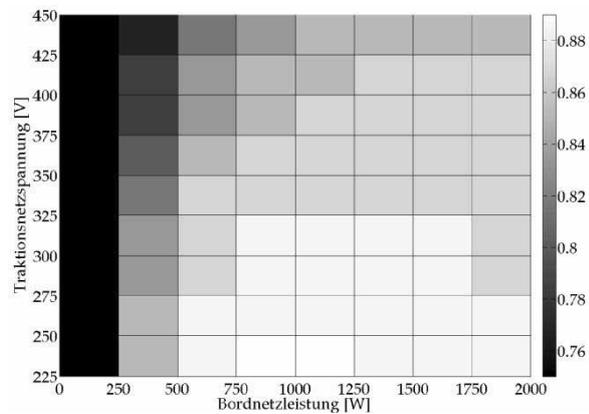


Abbildung 10. Wirkungsgradkennfeld des Tiefsetzstellers [10]

4.5 Integration in die Fahrdynamikregelung und das regenerative Bremssystem

Abbildung 11 zeigt die Schnittstelle zwischen der Fahrdynamikregelung mit dem regenerativen Bremssystem und dem Energiemanagement [7]. Unterschieden wird zwischen den beiden Fällen Antreiben und Beschleunigen des Fahrzeugs.

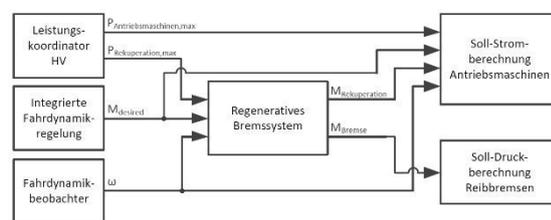


Abbildung 11. Schnittstellen des Energiemanagements an die Fahrdynamikregelung und das regenerative Bremssystem

Im Antriebsfall wird der Soll-Stromberechnung der Antriebsmaschinen die maximal verfügbare Leistung für die Antriebsmaschinen $P_{Ant,max}$ als Begrenzung

übergeben. Von der integrierten Fahrdynamikregelung wird das gewünschte Antriebsmoment der einzelnen Räder radindividuell vorgegeben.

Im Bremsfall wird von der integrierten Fahrdynamikregelung ein gewünschtes Bremsmoment M_{desired} an das regenerative Bremssystem gegeben. Durch Multiplikation mit den Radwinkelgeschwindigkeiten ω werden radindividuell die notwendigen Bremsleistungen berechnet. Durch den Leistungskordinator wird die maximal rekuperierbare elektrische Leistung $P_{\text{Rekup,max}}$ vorgegeben. Die Bremsleistung wird bis zu $P_{\text{Rekup,max}}$ vom regenerativen Bremssystem auf die Antriebsmaschinen verteilt, welche zur Soll-Stromberechnung die Bremsmomente M_{Rekup} für jedes Rad bekommt. Eine über $P_{\text{Rekup,max}}$ hinausgehende Bremsleistung wird von der Reibbremse aufgebracht, welcher als Sollwert das Bremsmoment M_{Brems} vorgegeben wird.

5 Erprobung

In diesem Abschnitt wird die modellbasierte Erprobung des EEM dargestellt.

5.1 Testbedingungen

Die Tests werden, wenn nicht anders beschrieben gekennzeichnet, bei einer Temperatur von 20 °C und unter Volllast, also bei Einschalten sämtlicher Verbraucher des Bordnetzes, durchgeführt.

Um das EEM möglichst realitätsnah erproben zu können wird der aus dem EU-Projekt ARTEMIS hervorgegangene Common Artemis Driving Cycle (vgl. **Abbildung 12**) verwendet, da dieser Stadt-, Überland- und Autobahnanteile beinhaltet und realistische Beschleunigungen abbildet.

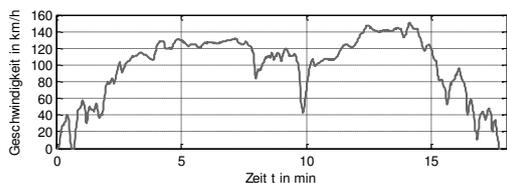


Abbildung 12. CADC-Fahrzyklus

5.2 Vergleich des Systems ohne und mit EEM

Zur Beurteilung des Einflusses des EEM wurden je eine Messreihe ohne und mit EEM aufgenommen. **Abbildung 13** zeigt oben den State of Charge (SoC) der Hochvolt-Batterie. Erkennbar ist, dass ohne Energiemanagement (hellgrau) die Batterie nach ca. 110 Minuten bis auf 0 % entladen wird. Mit Energiemanagement (dunkelgrau) wird erst nach 115 Minuten ein minimaler Ladestand von 3 % erreicht. Durch das

Energiemanagement wird also die Batterie vor einer kompletten Entladung auf 0 % geschützt.

Im unteren Teil von **Abbildung 13** sind die Ströme im Niederspannungsnetz dargestellt. Ohne Energiemanagement beträgt der Strom im Durchschnitt ca. 250 A mit Spitzen bis zu 400 A. Durch das Energiemanagement wird der Strom auf 200 A begrenzt, wodurch Batterie und Bordnetz vor Überlastung geschützt werden. Allerdings werden durch die Begrenzung des Stroms zwangsweise Verbraucher nicht eingeschaltet. Da das NV-Netz durch den Tiefsetzsteller mit Energie aus dem HV-Netz versorgt wird, sinkt mit Energiemanagement der Ladestand der HV-Batterie langsamer, da die Belastung geringer ist.

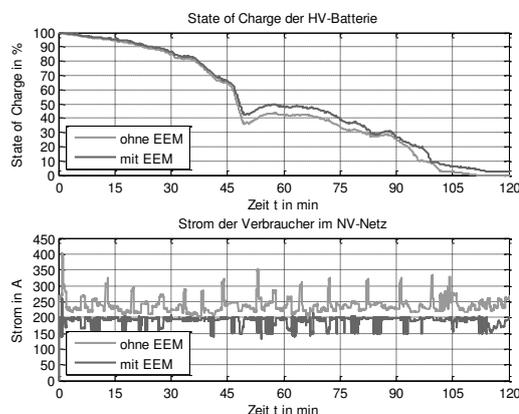


Abbildung 13. Begrenzung des Batterieladestandes auf mindestens 3 % und des Stromes auf 200 A

In **Abbildung 14** (oben), stellen die Referenzgeschwindigkeit und die tatsächliche Fahrzeuggeschwindigkeit gegenüber. Die tatsächliche Fahrzeuggeschwindigkeit sinkt nach 115 Minuten auf 0 und folgt der Referenzgeschwindigkeit nicht mehr (das Auto bleibt stehen). Dies liegt daran, dass die Batterie bis zur kritischen SoC-Grenze von 3 % (mittig) entladen wurde und keine Energie mehr für die Antriebsmaschinen bereitstellen kann.

Unten in **Abbildung 14** sind die Begrenzung der Antriebsleistung sowie die tatsächliche Antriebsleistung dargestellt. Bei Erreichen des Ladestandes von 3 % wird die Antriebsleistung komplett begrenzt, sodass das Fahrzeug nicht weiter angetrieben werden kann.

Aus **Abbildung 15** kann der Einfluss des Tiefsetzstellers auf das NV-Netz abgelesen werden. Oben sind der SoC des HV-Netzes sowie des NV-Netzes dargestellt. Der SoC des NV-Netzes wird bis 115 Minuten durch eine Zweipunktregelung zwischen 90 und 95 % gehalten. Dies am Tiefsetzsteller (2. von oben), welcher bei Erreichen eines SoC im NV-Netz von 90 % eine Leistung (verlustbehaftet) aus dem HV in das

HV-Netz einspeist, bis der SoC 95 % beträgt. Dort wird die Leistungszufuhr komplett abgeschaltet.

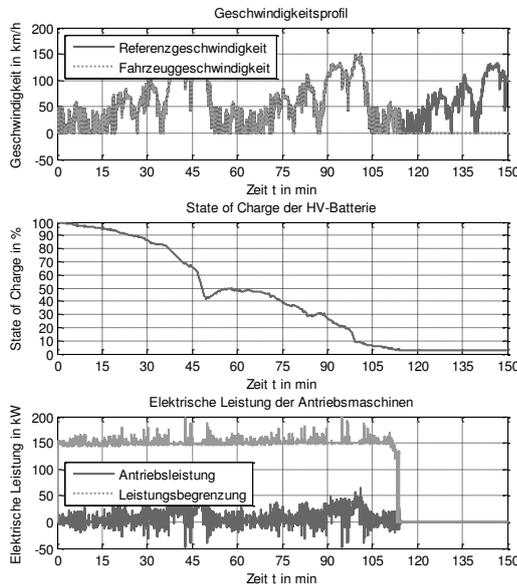


Abbildung 14. Begrenzung der Antriebsleistung

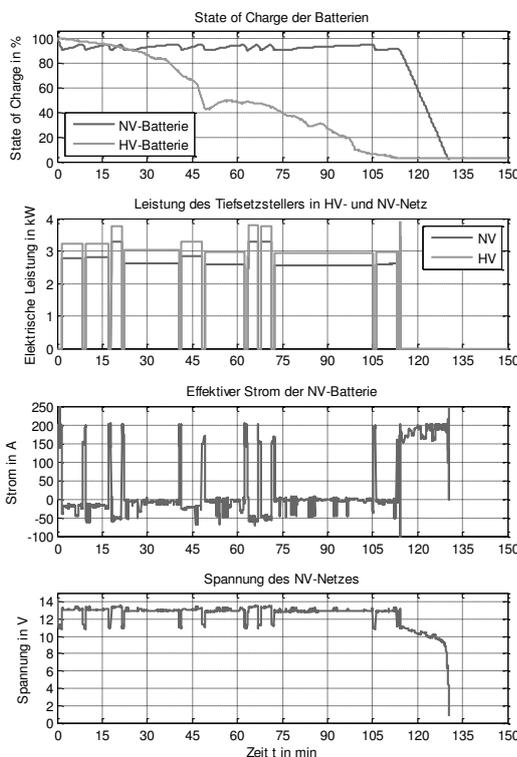


Abbildung 15. Belastung des NV-Netzes

Problematisch ist allerdings, dass hieraus eine große Stromdifferenz zwischen den Phasen mit Leistungsübertragung (-50 A) und ohne (200 A) resultiert

(Abbildung 15, 2. von unten). Aus dieser Stromdifferenz von 250 A folgen Spannungsschwankungen (Abbildung 15, unten) von 2,5 V (11 bis 13,5 V), wodurch die Bordnetzstabilität gefährdet ist.

5.3 Kopplung mit Fahrdynamikregelung und regenerativem Bremssystem

Abbildung 16 zeigt oben die Leistung der Antriebsmaschinen sowie die Leistungsbegrenzung der Rekuperation. Erkennbar ist, dass die Rekuperation nur zu Beginn der Fahrt begrenzt ist. Dies liegt daran, dass die Batterie mit der Zeit immer weiter entladen wird und somit immer mehr Energie wieder speichern kann.

Aus diesem Grund wird die Rekuperationsbegrenzung zu Fahrtbeginn (mittig) innerhalb der ersten eineinhalb Minuten betrachtet. Die Antriebsleistung stellt eine Rekuperationsleistung dar, wenn sie kleiner als 0 ist. Bei 0,65 Sekunden, 0,75 Sekunden, 0,85 Sekunden sowie um 1 Sekunde wird die Rekuperation begrenzt.

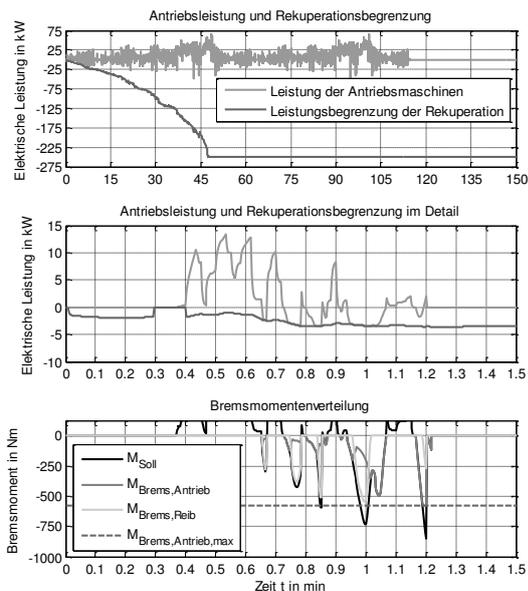


Abbildung 16. Funktionsnachweis des regenerativen Bremssystems

Unten in Abbildung 16 sind das von der Fahrdynamik gewünschte Antriebsmoment, das von den Antriebsmaschinen aufgebrauchte Bremsmoment, das von den Reibbremsen aufgebrauchte Bremsmoment sowie das maximal durch die Antriebsmaschinen aufbringbare Bremsmoment (vorgegeben durch die maximale Belastbarkeit der Antriebsmaschinen) dargestellt. Es zeigt, dass durch die Reibbremse immer ein Bremsmoment aufgebracht wird, wenn die Antriebsmaschine nur ein begrenztes Bremsmoment aufbringen kann. Das regenerative Bremssystem arbeitet also

korrekt und nutzt die Reibbremse, wenn die Bremsleistung der Antriebsmaschinen nicht ausreicht.

6 Fazit und Ausblick

Im Rahmen dieses Beitrags wurde die modellbasierte Entwicklung des elektrischen Energiemanagements vorgestellt. Die gestellten Anforderungen konnten erfüllt werden, sodass durch den entwickelten Energiemanagementalgorithmus die Belastung des Bordnetzes und der Batterien begrenzt werden konnte und die Batterien vor einer Tiefentladung und daraus folgenden irreparablen Schäden geschützt werden.

Momentan wird an einer Realisierung des Systems auf einem HiL-Prüfstand zur Validierung der Simulationsergebnisse gearbeitet. Weiterhin wird der Einfluss von Schaltspitzen, welche ein Vielfaches des Nennstroms betragen können, auf die Bordnetzstabilität untersucht.

7 Danksagung

Dieser Beitrag wurde im Rahmen des Forschungsprojekts *Energy Consumption Optimization for electrical Cars* (kurz ECOCar) durch das Bundesministerium für Bildung und Forschung unter dem Förderkennzeichen 03FH03513 gefördert. Die Verantwortung für den Inhalt liegt bei Autoren. Für die Förderung bedanken sich diese herzlichst.



8 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] Quantmeyer, F.; Liu-Henke, X.; Scherler, S.: *Design and Test of the Vehicle Management of Electrical Vehicles with Decentralized Direct Drives*. 12. Symposium: Hybrid- und Elektrofahrzeuge, Braunschweig, 24.-25. Februar 2015.
- [3] Büchner, S.: *Energiemanagement-Strategien für elektrische Energiebordnetze in Kraftfahrzeugen*. Dissertation, Technische Universität Dresden, Deutschland, 2008.
- [4] Schmidt, M.: *Ein selbstadaptierender, dynamischer Energiemanagementansatz für das elektrische Kraftfahrzeugbordnetz*. Dissertation, Universität Kassel, Deutschland, 2009.
- [5] Heitel, B.; Wagner, T.; Holm, G.: *Agentenorientiertes dezentrales Energie- und Funktionsmanagement für Kfz*. Entwicklerforum Kfz-Elektronik, Ludwigsburg/Stuttgart, Deutschland, 2004.
- [6] Scherler, S.: *Modellbasierte Entwicklung des elektrischen Energiemanagements für Elektrofahrzeuge*. Bachelorarbeit, Ostfalia Hochschule für angewandte Wissenschaften, Wolfenbüttel, 2014
- [7] Quantmeyer, F.: *Fahrzeugmanagement für ein Elektrofahrzeug mit dezentralen Direktantrieben*. Dissertation, Universität Magdeburg, Deutschland (erscheint demnächst).
- [8] Quantmeyer, F.; Liu-Henke, X.: *State of Charge Estimation for Lithium-ion batteries in Electric Vehicles using Extended Kalman Filtering*. 1st International Symposium on Energy & Mechanics (ECM), Aberdeen, UK, July 8 - 10, 2014.
- [9] Quantmeyer, F.; Liu-Henke, X.: *Modeling the Electrical Behavior of Lithium-ion Batteries for Electric Vehicles*. Solid State Phenomena Vol. 214, Trans Tech Publications, Switzerland, S. 40-47, 2014.
- [10] Lange, S.; Schimanski, M.: *Energiemanagement in Fahrzeugen mit alternativen Antrieben*. Dissertation, Technische Universität Braunschweig, Deutschland, 2006.

Effiziente Modellierung heterogener Batteriemanagementsysteme(BMS)

Thomas Markwirth, Matthias Gulbins, Karsten Einwich, Joachim Haase

Fraunhofer IIS/ EAS, Zeunerstraße 38, 01068 Dresden

Thomas.Markwirth@eas.iis.fraunhofer.de

Vorgestellt wird eine Simulationsumgebung für den BMS-Entwurf auf Basis von SystemC/SystemC AMS. Modelle der Batteriezellen, der Sensoren zur Temperatur- und Strommessung, der Komponenten für den Ladungsausgleich, der integrierten Schaltungen zum Batteriemonitoring und die Schnittstelle zur Übertragung der Daten an den Controller, auf dem die BMS-Software läuft, beschreiben die Hardware. Die Anwendersoftware kann über eine AUTOSAR RTE konforme Programmierschnittstelle in die Simulationen einbezogen werden. Es kann sowohl das nominale als auch das fehlerhafte Verhalten von Komponenten berücksichtigt werden. Basis für die Simulationsumgebung ist die Entwurfsumgebung COSIDE.

1 Einleitung

Aufgrund der vergleichsweise geringen Energiedichte aktuell verfügbarer elektrischer Energiespeicher (Batterien), bilden diese momentan den kritischen Punkt bei der Umsetzung der Energiewende im Bereich der Elektromobilität. Umso wichtiger ist die optimale Ausnutzung der aktuell verfügbaren Speicher in Bezug auf Kapazität (Reichweite), Anzahl der maximalen Zyklen (Lebensdauer), Sicherheit usw. Um dies zu erreichen, ist ein optimales und intelligentes Batteriemangement essentiell. Dieses muss die optimale Ladung und Entladung der Batterie über deren gesamte Lebensdauer gewährleisten und sicherstellen, dass die Batterie nie einen kritischen Zustand erreicht, welcher zu deren Zerstörung, Bränden oder Explosion führen kann.

2 Realisierung

Im Rahmen des Vorhabens IKEBA [1] wird eine virtuelle Entwurfsplattform prototypisch erstellt, die es gestattet, mittels Simulation die Eignung und das Zusammenspiel ausgewählter Hardwarekomponenten (Li-Ion-Batterien und Halbleiterschaltungen zum Batteriemonitoring) und der Software eines Batteriemangementssystems (BMS) zu überprüfen und die Entwicklung der Batteriemangement-Software (BM-Software) zu unterstützen. Durch die Verwendung der Entwurfsplattform sind u.a. Aussagen zur Erhöhung der Reichweite elektrisch betriebener Fahrzeuge für unterschiedliche Fahrscenarien in Abhängigkeit von der BM-Software möglich. Partner im Vorhaben IKEBA sind neben der Fraunhofer-Gesellschaft e.V. die Firmen Hella KGaA Hueck & Co und Atmel

Automotive GmbH sowie das Institut für Angewandte Materialien - Angewandte Werkstoffphysik des Karlsruher Instituts für Technologie. Basis für die virtuelle Entwurfsplattform ist die Entwurfsumgebung COSIDE [2] für die Entwicklung elektronischer Systeme. COSIDE ist ein Tool, mit dem hochkomplexe elektronische oder heterogene Systeme zusammen mit der Software, die auf diesen Systemen läuft, modelliert und simuliert werden können.

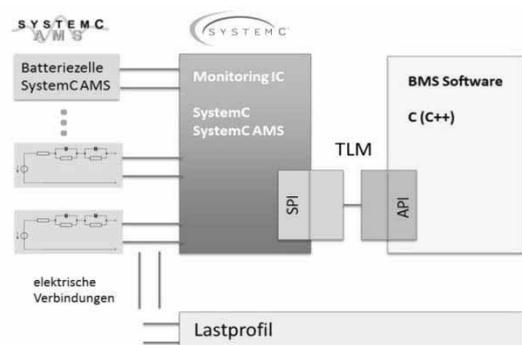


Abb. 1 BMS-Entwurfsumgebung

Die Modellerstellung für die Hardwarekomponenten erfolgt mit den Hardwarebeschreibungssprachen SystemC und SystemC AMS. Dieser Ansatz erlaubt neben der Berücksichtigung der Eigenschaften der Batterien, von Schaltungen zum Ausgleich ungleicher Ladezustände (Charge Balancing) und der Lastprofile eine detaillierte Einbeziehung der Komponenten der Batteriemangement-ICs (BM ICs) hinsichtlich Charakteristiken der A/D-Wandler, Verzögerungszeiten u.ä. sowie der Datenschnittstelle (speziell SPI) zwischen BM ICs und Batteriemangement

ment-Controller in die Untersuchungen mit einer hohen Simulationsgeschwindigkeit. Für die Erstellung der BM-Software wird eine AUTOSAR RTE konforme Programmierschnittstelle (API) mit den benötigten C-Funktionen bereitgestellt.

Die virtuelle Entwurfsplattform wird an Hand eines Funktionsdemonstrators im Vorhaben IKEBA erprobt. Dabei wird zunächst das erwartete Nominalverhalten simuliert, das später mit Messergebnissen verglichen werden soll. Mittels Simulation können unterschiedliche Konzepte zum Betrieb der Batterien mit dem Ziel der Reichweitenerhöhung verglichen werden.

In Zusammenhang mit der Modellierung und Simulation für den Funktionsdemonstrator sind eine Reihe von Modellen unter Verwendung von SystemC und SystemC AMS erstellt worden. Eine Parametrisierung erfolgte für die Komponenten, die im Funktionsdemonstrator Verwendung finden sollen. Tabelle 1 gibt einen Überblick über die derzeit vorhandenen Modelle, die bei Fraunhofer IIS/EAS erstellt worden sind. Durch Austausch und Erweiterung von Modellen kann die beschriebene Lösung leicht an andere Anwendungsfälle angepasst werden.

Tabelle 1 Modelle für virtuelle Entwurfsplattform

Block	Modelle/Modellbibliothek	Beschreibung
Batterie	battery_cell	Tabellenmodell für Batteriezelle
	battery_pack_6s1p	Batteriestapel (6 Batteriezellen in einer Reihe)
BM IC	bmic_eln	Modell des elektrischen Klemmenverhaltens des BM IC zur Messung und Überwachung von Batteriestapeln in hybriden und elektrischen Fahrzeugen
	bmic_simple	Verhaltensmodell für BM IC zur Messung und Überwachung von Batteriestapeln in hybriden und elektrischen Fahrzeugen
Umgebung	passive_balancing_eln_6s1p	Schaltung zur passiven Ladungsbalancierung

	simple_hvcs_eln	Strommesser (High Voltage Current Sensor)
	ntc_thermistor_characteristics	temperaturabhängige NTCWiderstandscharakteristik (NTC-Heißleiter Semitec 104AP-2)
	temperature_measurement_eln	Widerstandsschaltung zur Temperaturmessung
	cell_temp	Modell der Wärmeerzeugung in der Batteriezelle und –austausch mit der Umgebung
	file_in_eln_isource	Stromquelle mit Einlesen eines Stromprofils aus einer Datei
BM Controller	rte	AUTOSAR RTE konforme API des BM Controllers (BM-Software)
TLM	tlm2spi_target	TLM Interface für SPI (für Verbindung des BM IC zum BM Controller)
	tlm_current_sensor	TLM Interface für LIN (für Verbindung des Stromsensors zum BM Controller)
	tlm_controller	TLM-Modell des BM Controllers

3 Beispiel

Die folgende Abbildung zeigt das heterogene Beispiel für ein BMS-Systemmodell, welches im IKEBA-Projekt erarbeitet wurde. Zur Anwendung kam dabei ein generischer Ansatz, wobei einzelne Komponenten ausgetauscht werden können. Ein möglicher Anwendungsfall ist die Untersuchung der Anforderungen an die Genauigkeit der ADUs für die Spannungs-, Strom- und Temperaturmessung. Aufgrund des generischen Aufbaus des virtuellen Prototyps kann darüber hinaus die Funktion der BMS Software bei Austausch der Einzelkomponenten z.B. anderer Batteriezellen oder anderer Monitoring-ICs überprüft werden. Für Untersuchungen auf Systemebene ist dabei eine

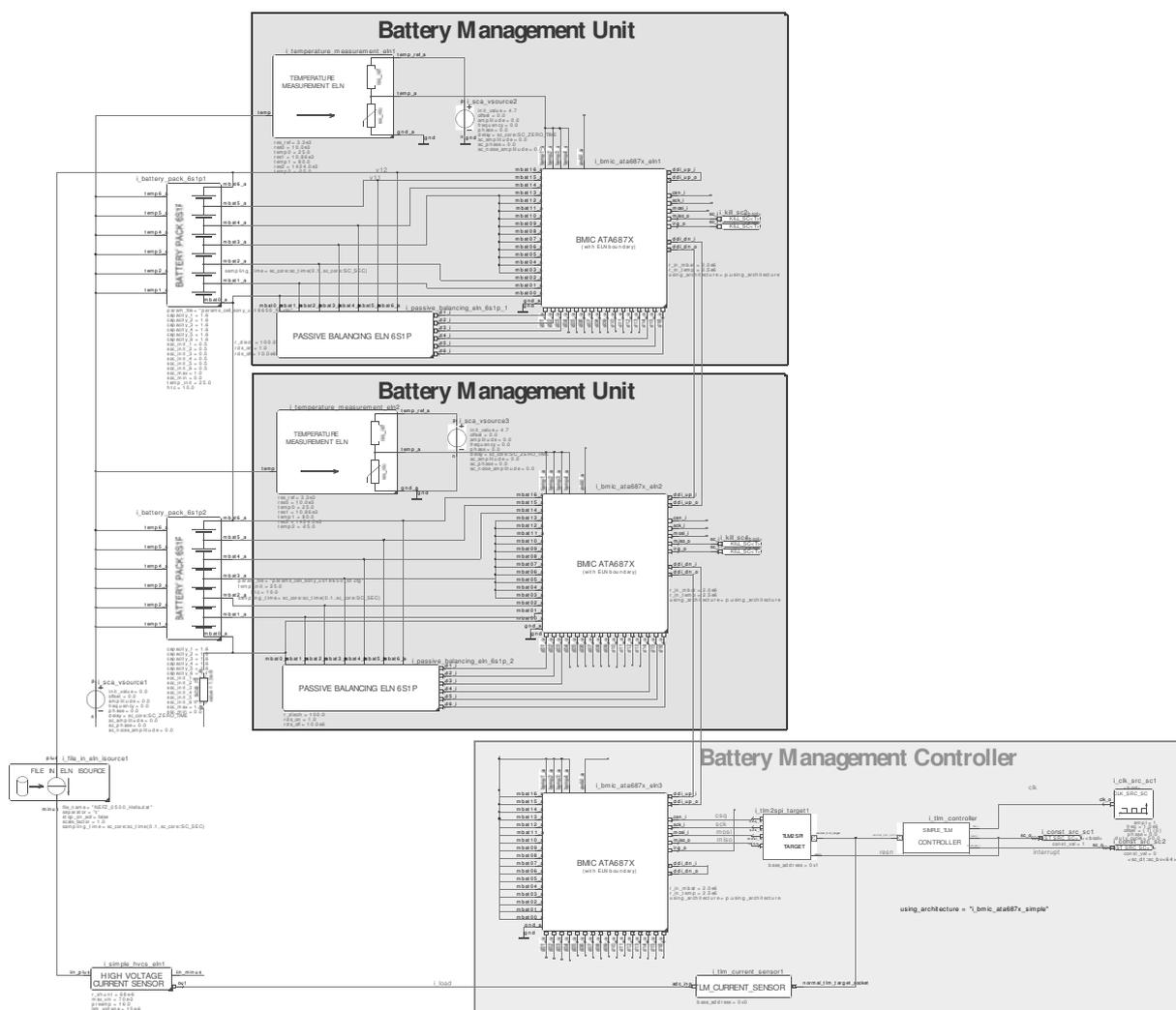


Abb. 2: IKEBA-Toplevel eines BM-Systems Ebenso kann, beispielweise für das Modell des BM-ICs der Abstraktionsgrad ausgewählt werden

hohe Performance bei der Simulation zwingend, um mit vertretbarem zeitlichen Aufwand zu aussagekräftigen Ergebnissen zu gelangen. Um dies gewährleisten zu können, werden effiziente Modellierungsmethodiken wie etwa eine Abstraktion der Bussysteme auf Transaktionsebene (TLM) angewendet. Bei diesem Ansatz liegt der Fokus weniger auf der physikalischen Umsetzung der Schnittstelle, dem realen Signalspiel, sondern einzig auf der Transaktion der Daten selbst. Dies unterstützt damit einen generischen Modellierungsansatz, wo Änderungen am Bussystem, etwa bezüglich dem Anschluss weiterer Komponenten oder der Anwendung von Systemkom-

ponenten anderer Hersteller auf einfache Weise realisiert werden können. Bei herkömmlichen Modellierungsmethodiken werden hingegen bei zu erwartenden Änderungen der Systemkonfiguration oder Änderungen der Schnittstellenspezifikationen ständige Aktualisierung bzw. Variation z.B. des Controllermodells nach sich ziehen. Ein generischer Ansatz wäre so jedoch nicht mehr gegeben. Darüber hinaus werden gerade in der Automobilelektronik, eine Vielzahl von unterschiedlichsten Schnittstellen für die Anbindung von Controllern untereinander oder zu anderen Schaltkreisen im System angewendet. Zudem liegen meist applikationsbedingte Variationen dieser Schnitt-

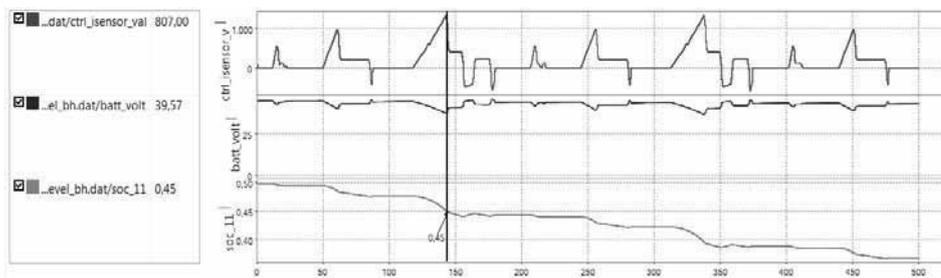


Abb. 3 Batteriespannung und Ladezustand bei vorgegebener Last

stellen vor. Für eine funktionale Systemuntersuchung spielt die genaue Anbindung der Buskomponenten jedoch meist nur eine untergeordnete Rolle. Zum anderen bietet die TLM-Methodik in Bezug auf die Simulationsperformance eine hohe Effizienz, da das Signalspiel nicht nachgebildet werden muss. Diese Effizienz bildet damit auch eine wesentlich Voraussetzung, um statistische Untersuchungen von Parameterstreuungen durchführen zu können. Im Anschluss wird detaillierter auf das verwendete Batteriemodell und die Einbeziehung der BM-Software in die Systembeschreibung eingegangen. Das nachfolgende Diagramm zeigt erste Simulationsergebnisse für den Batteriespannungs- und SOC-Verlauf in Abhängigkeit eines vorgegebenen Lastszenarios.

4 Ausgewählte Modelle

4.1 Modell für eine Batteriezelle

Die Batteriezellen sind Lithium-Ionen-Zellen. Die Einzelzelle wurde in SystemC AMS modelliert. Das Modell besteht – wie in den meisten elektrischen Modellen – aus einer Leerlaufspannungsquelle V_{oc} , einem Innenwiderstand R_{IN} sowie zwei RC-Gliedern $R_1||C_1$ und $R_2||C_2$, siehe Abbildung 4:

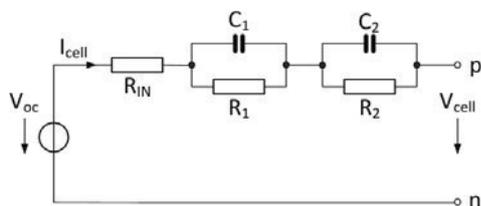


Abb. 4 Struktur des Modells einer Batteriezelle

Dabei steht R_{IN} für den generellen Innenwiderstand, das RC-Glied $R_1||C_1$ für eine Regenerierung nach Lade-/Entladestromänderung mit einer Zeitkonstante von etwa einer Minute und das RC-Glied $R_2||C_2$ für eine AC-Impedanz mit einer Zeitkonstante von etwa 1 ms. Die Elemente des Ersatzschaltbildes hängen

vom Ladezustand der Batterie Zelle (SOC – State of Charge) und der Temperatur der Batteriezelle ab. Die Änderung des Ladezustandes ergibt sich aus den Lade- und Entladeströmen der Zelle und der Zellkapazität. Letztere ist abhängig von der Temperatur.

Die skizzierten Abhängigkeiten können aus Ergebnissen von Messungen beim Laden und Entladen ausgewählter Zellen mit Stromimpulsen (CIT – Current Interruption Technique) ermittelt werden [3]. Zur Modellierung der Zellen können analytische Modelle wie beispielsweise das von Gao et al. [4] genutzte verwendet werden. In der beschriebenen Simulationsumgebung werden die Abhängigkeiten auch mit Tabellenmodellen erfasst. Für ausgewählte Werte von SOC und Temperatur werden die zugehörigen Parameter des Ersatzschaltbildes bestimmt. Durch Interpolation zwischen den gegebenen Stützstellen können während der Simulation die konkreten Parameter ermittelt werden. Da die Änderungen von SOC und Temperatur „langsam“ erfolgen, werden zur Beschleunigung der Simulation bei der Parameterbestimmung im implementierten Modell SOC- und Temperaturwerte zum jeweils letzten Simulationszeitpunkt verwendet.

4.2 Interface von BM IC und BM Controller

Die Schnittstellen von BM ICs und Strommesser zum Batteriemanager-Controller wurden mit Transaction-Level-Modellen (TLM) beschrieben. Folgende Funktionen werden durch die Transaction-Level-Modelle unterstützt

- Initialisierung BM ICs
- Initialisierung Strommesser
- Ermittlung von Zellspannungen unter Verwendung der BM ICs, falls erforderlich getrennte Funktionen für
 - Start der Messungen
 - Lesen der Werte

- Ermittlung von Zelltemperaturen unter Verwendung der BM ICs, falls erforderlich getrennte Funktionen für
 - Start der Messungen
 - Lesen der Werte
- Ermittlung von Strangströmen unter Verwendung der Strommesser, falls erforderlich getrennte Funktionen für
 - Start der Messungen
 - Lesen der Werte
- Setzen von Steuersignalen der BM ICs für das Ansteuern der Ladungsausgleichsschaltungen

4.3 Schnittstelle zur BM Anwendersoftware

Die BMS-Anwendersoftware kann auf Werte, die über die BM ICs und die Strommesser bereitgestellt werden, über eine AUTOSAR RTE konforme Programmierschnittstelle (API) zugreifen.

4.4 Einbeziehung der BM - Software

Um die Software des Batteriemangementssystems in die Systemsimulation einbeziehen zu können, muss zunächst ein Controllermodell existieren, welches die Verbindung zu den Hardwarekomponenten des Systems herstellt. Hierzu existiert ein generisches Controllermodell, welches die Softwareapplikationen über einen Schedulerprozess aufruft, der seinerseits sensitiv auf externe Takt-, Reset- und Interruptsignale reagiert.

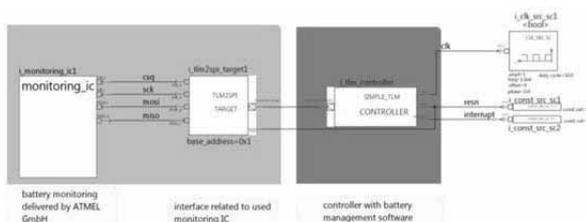


Abb. 5 Bussystem mit Controller und Monitoring IC

Zusätzlich werden den Softwarekomponenten Zugriffsfunktionen, auf der sogenannten Hardware-Abstraktionsebene, bereitgestellt. Bei Aufruf einer solchen Funktion durch eine Softwareapplikation wird eine entsprechende TLM-Transaktion durch das Controller-Modell ausgelöst und damit der Zugriff auf eine periphere Hardware realisiert. Dies schafft die Möglichkeit für performante Zugriffe unabhängig von der tatsächlich physikalischen Schnittstelle. Ebenso ist damit gewährleistet, dass die Software auch für verschiedene Architekturen und periphere Konfigurationen verwendet werden kann, sofern die

angewählte Hardwareadresse unterlegt und ansprechbar ist.

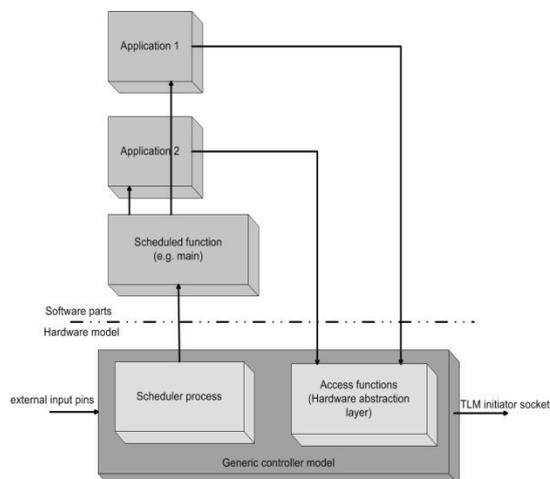


Abb. 6 generisches Controllermodell und Einbindung von Softwareapplikationen

Damit existiert eine Lösung, die eine saubere Trennung der einzelnen Komponenten ermöglicht, welche für sich konsistent weiterentwickelt werden können. Falls das angesprochene Hardwaremodell, z.B. des Monitoring ICs, über keine TLM-Schnittstelle verfügt, muss gegebenenfalls eine Konvertierung in reale Schnittstellensignale erfolgen, welche dann konkret an die Schnittstellenspezifikation angepasst werden können. Ein solcher Konverter ist damit Bestandteil der Entwicklung des Modells des Monitoring-ICs und könnte mit relativ einfachen Mitteln automatisiert generiert werden, da die zugrundeliegenden Beschreibungen entweder aus grundlegenden Basisbibliotheken (z.B. TLM Sockets) oder aus der Spezifikation des konkreten ICs (z.B. SPI-Frames) entnommen werden können.

4.5 Hardware – In – The – Loop – Anwendung (HiL)

Das vorgestellte Systemmodell kann im weiteren Entwicklungsverlauf des Gesamtsystems für HiL (Hardware-In-The-Loop) Simulationen verwendet werden, da es auf Basis von SystemC AMS einen hohen Grad an Abstraktion besitzt [5]. Dieser gewährleistet eine schnelle Ausführbarkeit mit hinreichender Genauigkeit, was Grundvoraussetzung an das Modell für den Echtzeiteinsatz auf einem HiL System ist. Dabei werden einzelne Module des Modells durch real existierende Hardwarekomponenten ersetzt, wohingegen andere virtuell verbleiben. Mögliche Vorteile dieser Trennung, sind die Verkürzung von Rechenzeit bei Auslagerung komplexer Modell-

komponenten in gut verstandene und etablierte Hardware, der frühzeitige Test eigener Hardwarekomponenten, im realen System weit vor dessen vollständiger physischer Verfügbarkeit, sowie der Test der verschiedenen spezifizierten Schnittstellen innerhalb des Systems. Dabei spielt es im Allgemeinen, mit Ausnahme von Punkt eins, keine Rolle ob es sich bei der Hardware um ausgereifte Produkte oder um eine prototypische Implementierung handelt. Weitere Vorteile entstehen durch einen vereinfachten Teststand durch das HiL, wenn es sich bei den ersetzten Komponenten um gefährliche, große oder teure Baugruppen handelt.

Eine sinnvolle Trennung von realen und virtuellen Komponenten ist im vorliegenden Projekt daher zwischen der eigentlichen Batterie und dem Sensorchip mit angeschlossenem BMS zu realisieren. Die Batterie wird dabei im HiL abgebildet, wohingegen der Rest in realer Hardware angeschlossen wird. Hierbei werden am HiL spezielle hochpräzise Spannungsquellen angeschlossen, welche gestapelt die gewünschte Gesamtspannung des Zellenverbands zur Verfügung stellen und somit die elektrischen Eigenschaften der spezifizierten Schnittstelle nachbilden. Die Entscheidung, die Batterie zu simulieren, basiert auf folgenden grundlegenden Vorteilen. Eine Batterie beinhaltet komplexe chemisch-physikalische Vorgänge, welche nicht eindeutig reproduzierbar sind und Zeit benötigen, um sie einzustellen. Genannt seien hier die gezielte Ladung und Entladung zu einem definierten Punkt sowie das Einstellen einer definierten Temperatur mithilfe eines Klimaschranks bzw. Kalorimeters. Des Weiteren reagiert die Batterie sensibel auf Fehler und sollte daher unter hohen Sicherheitsvorkehrungen und erst im fortgeschrittenen Entwicklungsstadium des BMS an dieses angeschlossen werden. Im Fall des HiLs ist dies aber ohne Bedeutung, wodurch das BMS inklusive der vorgesehenen Schutzmechanismen für Tiefenentladung bzw. Überladung, sowie für die thermische Überlastung frühzeitig getestet werden kann. Auch das Verhalten bei Fehlern innerhalb des Batterieblocks wie Leitungunterbrechungen, Kurzschlüsse oder Wackelkontakte können innerhalb des HiLs einfach realisiert werden.

Ausblick: Entscheidend für die Qualität eines Systems ist neben vielen anderen Kriterien auch die Fähigkeit auf fehlerhafte oder unerwartete Zustände adäquat reagieren zu können, sowie robust gegenüber Störgrößen zu sein. Um Aussagen hierzu treffen zu können, müssen Untersuchungen durchgeführt wer-

den, welche Fehlerzustände und Störungen provozieren bzw. beinhalten. Gegenwärtig wird im Rahmen des Projektes eine Bibliothek zur Fehlerinjektion erstellt, welche generische Strukturen enthält, um auf einfache Art und Weise fehlerhaftes Verhalten nachzubilden zu können. Das Besondere hierbei wird sein, dass die Fehlerbeschreibungen nicht Teil des Designs unter Test (DUT) sind, sondern Fehlerstrukturen manipulativ nur für den jeweiligen Testfall eingebunden werden. Damit wird eine saubere Trennung zwischen Systemmodell und Testumgebung erzielt. Um, desweiteren die Robustheit des Systems gegenüber Störgrößen, Parametertoleranzen usw. untersuchen zu können, müssen statistische Analyse durchgeführt werden. Die hohe Performance von Modellbeschreibungen auf Basis von SystemC AMS ermöglicht dies auch auf Systemebene mit hoher Effizienz. Im Rahmen dieses Projektes werden statistische Untersuchungen zu einem späteren Zeitpunkt realisiert. Auf Bibliotheken und Erfahrungen aus anderen Projekten kann dabei zurückgegriffen werden [6].

5 Zusammenfassung

Komplexe heterogene BM-Systeme können mit den vorgestellten Mitteln effizient entworfen, analysiert und die unterschiedlichen Komponenten auf einander abgestimmt werden. Ebenso können damit neue Anwendungsstrategien, etwa für eine Reichweitenerhöhung entwickelt werden.

6 Referenzen

- [1] Integrierte Komponenten und integrierter Entwurf energieeffizienter Batteriesysteme (Vorhaben IKEBA). URL:<http://www.iam.kit.edu/awp/ikeba/>
- [2] COSIDE: The Design Environment for Heterogeneous System. URL:http://www.eas.iis.fraunhofer.de/content/dam/eas/de/documents/broschueren/fraunhofer-eas_coside.pdf
- [3] Lei, B.; Melcher, A. et al: Parametrisierung einer Ersatzschaltung für Li-Ionenzellen. Posterabstracts. Batterieforum D 2015. MS11-VP19.2
- [4] Gao, L. et al: Dynamic Lithium-Ion Battery Model for System Simulation. DOI: 10.1109/TCAPT.2002.803653
- [5] Einwich K.; Arndt, T. (2012): SystemC-AMS basierte Modellierung, Simulation und HiL Echtzeitsimulation für Anwendungen der Automobilelektronik. In: ASIM-Konferenz STS/GMMS 2012. Vienna: ARGESIM-Verl, S. 203–208.
- [6] Clauß, C.; Haase, J.; Markwirth, T.: Statistische Analyse mit dem SAE Standard J2748; ASIM-Treffen; Wismar 2008

Die diesem Beitrag zugrundeliegenden Arbeiten wurden mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen FKZ 16N12440 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Determination of Radiation Models Based on Measured Electromagnetic Field Distribution

Zongyi Chen, Denis Rinas, Stephan Frei

On-board System Lab
TU Dortmund University
Dortmund, Germany
zongyi.chen@tu-dortmund.de

Abstract

In EMC far-field measurements are commonly used to detect emission problems of electronic systems. Numerous standards have defined far-field thresholds. Far field measurements require large and expensive test facilities. Using more simple to realize near field measurements and Huygens principle for the calculation of the far-field can be a problem solution. The method is quite robust for higher frequencies. For lower frequencies the model needs to be further improved. A method that combines measurement data with simulation data, gathered from a standardized measurement setup, can provide quite accurate field estimations and is presented here.

1 Introduction

In EMC far field measurements are commonly used to detect emission problems of electronic systems. Numerous standards have defined far-field thresholds. Far field measurements require large and expensive test facilities. Using more simple to realize near-field measurements and Huygens principle for the calculation of the far field can be a problem solution. Alternatively, the measured electric and magnetic near-fields can be used for the reconstruction of the sources by a set of equivalent electric and/or magnetic dipoles [1]. However, using dipole method for a complex radiating structure, like a several layers-PCB, the radiation model can work with acceptable accuracy only if geometry information is known and many near-field scan points are taken. Huygens' principle [2-3] is independent from knowledge on the internal structures and requires only the near field distribution.

A Huygens surface model can be constructed that does not consider the complexity of the radiating structure. For this reason, this research is based on Huygens principle.

2 Theory

According to Huygens principle description [4], normally a closed fictitious Huygens surface is chosen to

enclose the radiating structure. As shown in **Figure 1**, using electric and magnetic fields on the Huygens surface, equivalent currents can be calculated and considered as a new radiating source that replaces the radiating structure and emulates fields in the exterior region of the Huygens surface. Inside of the surface the field is zero.

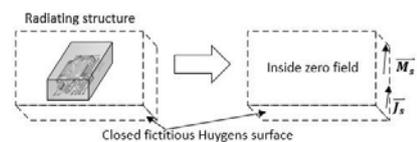


Figure 1 Representation of the Huygens principle

The new radiating sources are equivalent electric current \vec{J}_s and magnetic current \vec{M}_s , which are given by:

$$\vec{J}_s = \vec{n} \times \vec{H}_s \quad (1)$$

$$\vec{M}_s = -\vec{n} \times \vec{E}_s \quad (2)$$

where \vec{n} : Normal vector of Huygens surface (pointing to outer side of Huygens surface in **Figure 1**),

\vec{E}_s, \vec{H}_s : Fields on Huygens surface.

As illustrated in **Figure 2**, if the proper Huygens surface S has been chosen, and both electric currents and

magnetic currents on Huygens surface are known, the radiation model can be created.

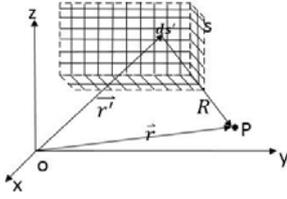


Figure 2 Field calculation in Cartesian coordinate system

By integrating all sources the field at an observation point P can be expressed as:

$$\vec{E}(\vec{r}) = \iint (-j\omega\mu\vec{J}_s G(\vec{r}, \vec{r}') + \vec{M}_s \times \nabla' G(\vec{r}, \vec{r}') - \frac{1}{j\omega} (\nabla' \cdot \vec{J}_s) \nabla' G(\vec{r}, \vec{r}')) ds' \quad (3)$$

Where $G(\vec{r}, \vec{r}')$ is free space Green's function.

3 Simulations and Measurements

3.1 Comparison of Huygen's surface calculation with full wave simulation of the complete structure with Method of Moment (MoM)

A dipole antenna structure in free space, as represented in **Figure 3**, has the length of 15 cm and the radius of 1 mm. An excitation of 1 V in the middle of the antenna is assumed. Observation point is [0.25 m, 0.25 m, 0.25 m]. Size of Huygens surface is 4 cm (length) x 4 cm (width) x 19 cm (height). The electric field of from Huygens surface and calculation with MoM of the physical structure matches well for all three components. MoM based software Concept-II [5] is used in this research.

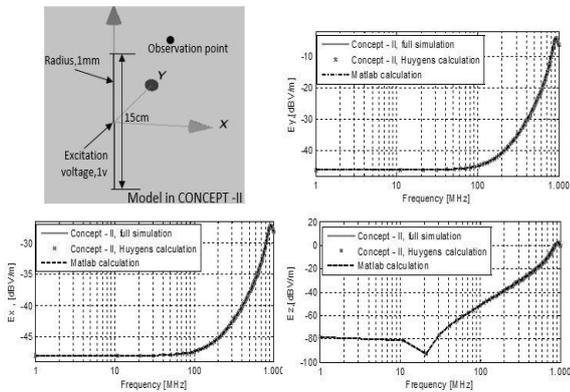


Figure 3 Comparison of electric field from theoretical calculation with MoM using a dipole antenna

3.2 Huygens surface size and discretization size

Error caused by different Huygens surface and discretization sizes has been investigated by observing the error of field magnitude at the observation point. As shown in **Figure 4**, a 1.5 m wire placed at 5 cm above a metal plate (2 m x 1 m) is taken as a wire loop structure, the structure is above infinite ground 0.9 m. The wire loop is excited with 1 V and terminated with 50 Ohm. Observation point is at [0.75 m, -1 m, 0.95 m], which is 1 m distant to center of wire, and with the same height as wire. As represented in **Figure 5**, upper figure shows that when discretization size is 2.5 cm (mesh size), error caused by different distances (Distance is the height between real radiating structure and each side of Huygens surface, different distance results in different size of closed Huygens surface). The magnitude of horizontal electric field has error of less than 1.5% for non-resonance frequencies. The lower figure shows when each side of Huygens surface is 5 cm distant to radiating structure, discretization sizes (mesh sizes) have been varied from 1 cm to 3 cm, the error of non-resonances are less than 0.5%. Both subfigures show that the error is larger, maximal up to 5.5% only for resonances.

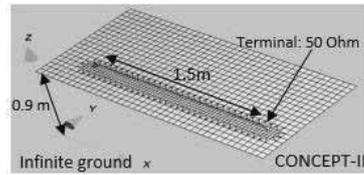


Figure 4 Structure used for investigation of Huygens surface size and discretization size

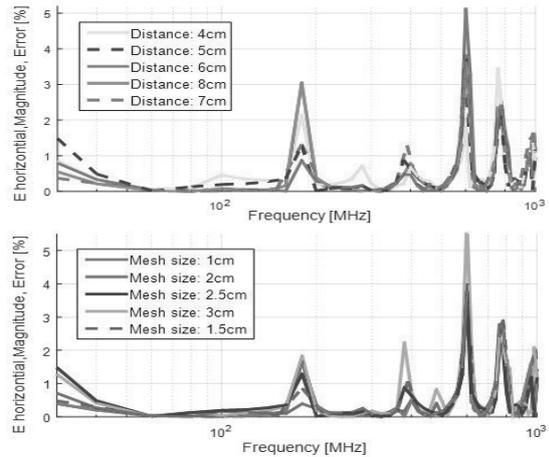


Figure 5 Field magnitudes resulting from Huygens surface sizes (upper) and discretization sizes (lower)

3.3 Near-field measurements

The structure shown in **Figure 6** has been used for near-field measurements, it's a wire (8 cm) above a metal plane (10 cm x 5 cm) with height 0.25 cm. The excitation is 2 V with 50 Ohm internal resistor. Each side of the Huygens surface has 1 cm distance to the structure. Firstly, using simulated fields from Concept-II, Gaussian white noise with different SNR have been added to complex field data (both electric and magnetic fields). The predicted electric field at 3 m (illustrated in **Figure 7**) shows that radiation model created based on Huygens principle can work quite robustly. For lower frequencies, the deviation becomes larger. As a verification, **Figure 8** shows comparison of the near-field measurement on top side of Huygens surface for magnetic field y-component at 31 MHz. It can be seen from the figure, both magnitude and phase are very noisy, which cause problem for radiation model creation, the near-field measurement result (in **Figure 8**) shows a good agreement with the simulation result (in **Figure 7**).

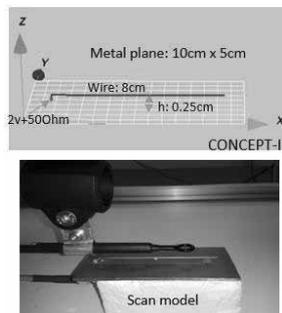


Figure 6 structure used for near-field measurements

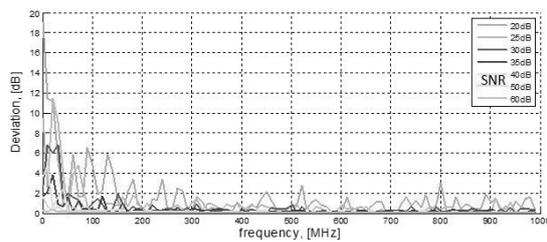


Figure 7 Electric field at 3 m with different SNR added to complex electric and magnetic field on Huygens surface

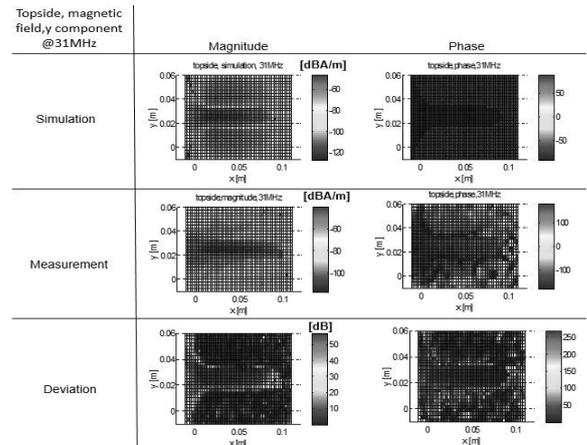


Figure 8 Comparison of simulated field with measured field

4 Conclusion and future work

In the research calculation of electric far-field using ideal electric and magnetic field distribution on Huygens surface has been compared with the simulated field from the physical structure model. Furthermore, investigations on discretization and size of Huygens surface were carried out such that an accurate radiation model could be determined for meaningful far-field prediction. Besides, this research has investigated how accuracy of radiation model is influenced by noisy field data using different SNR, and has shown deviations between field data that obtained from near-field measurement test bench and simulation. It can be concluded that noisy electric and magnetic field distributions on Huygens surface cause much more problems at lower frequencies than at higher frequencies for determining the far-field. Future work will focus on how to improve radiation model for lower frequencies.

5 References

- [1] T.Xin, D. W. P. Thomas, A. Nothofer, P. Sewell, and C. Christopoulos. "Modeling Electromagnetic Emissions From Printed Circuit Boards in Closed Environments Using Equivalent Dipoles," *IEEE Trans. Electromagn. Compat.*, vol. 52, no.2, pp.462-70, May.2010.
- [2] A.Taaghool, T. K. Sarkar, "Near-Field to Near/Far-Field Transformation for Arbitrary Near-Field Geometry, Utilizing an Equivalent Magnetic Current," *IEEE Trans. Electromagn. Compat.*, vol. 38, no.3, pp.536-542, Aug.1996.
- [3] P. Barri`ere, J. Laurin, Y.Goussard, "Mapping of Equivalent Currents on High-Speed Digital Printed

Circuit Boards Based on Near-Field Measurements,” *IEEE Trans. Electromagn. Compat.*, vol. 51, No. 3, pp. 649–658, Aug. 2009.

- [4] C. A. Balanis, *Advanced Engineering Electromagnetics*. New York: JohnWiley & Sons, 1989.
- [5] CONCEPT-II-12.0, Technische Universitaet Hamburg-Harburg. (2015). [Online]. Available: <http://www.tet.tuhh.de/en/concept/>.

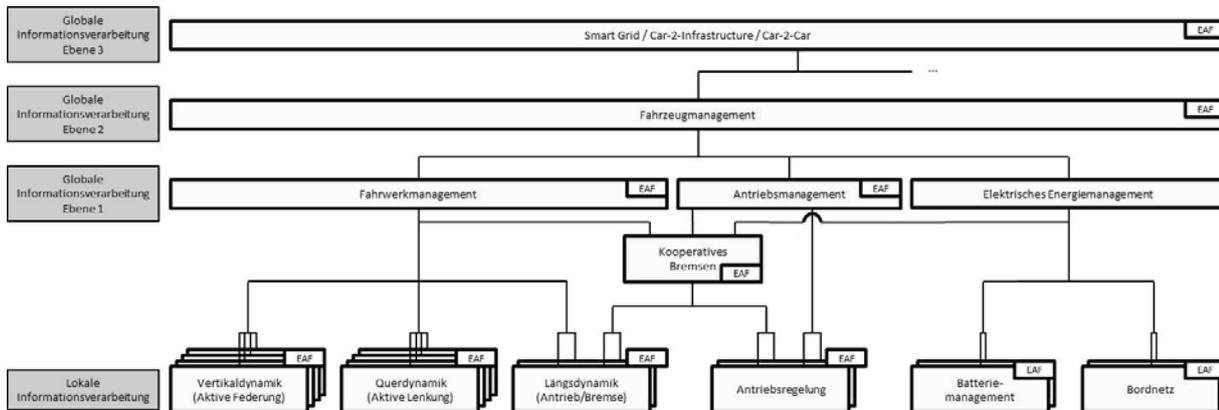


Abbildung 2. Mechatronische Strukturierung des Forschungsfahrzeugs

Nach der Strukturierung erfolgt die Auslegung und Verifikation der einzelnen Funktionsgruppen mittels Model-in-the-Loop, Software-in-the-Loop und Hardware-in-the-Loop. Hierbei werden von Beginn an Modelle aller mechatronischen Komponenten in der Simulation berücksichtigt, sodass die Validierung und Optimierung der Reglerfunktion möglichst realitätsnah erfolgen kann.

Der folgende Beitrag behandelt gezielt die aktive Fahrzeugfederung.

2 Aktive Federung

Bei der Auslegung passiver Fahrzeugfederungen muss stets der Zielkonflikt, einen möglichst hohen Fahrkomfort (geringe Aufbaubeschleunigung) bei gleichzeitig hoher Fahrsicherheit (geringe Radlastschwankungen) zu erreichen, gelöst werden. Dies ist jedoch mit passiven Federungen trotz Optimierung nur bedingt möglich. Weiterhin wird die Reichweite elektrisch angetriebener Fahrzeuge maßgeblich durch die noch geringe Energiedichte der Akkumulatoren begrenzt. Zur Reduzierung von Wirkungsgradverlusten werden radnahe Direktantriebe favorisiert. Durch diese steigt jedoch die ungefederte Radmasse stark an, was den Fahrkomfort bei konventionellen Federungssystemen deutlich verringern würde [1]. Diese Entwicklung verschärft den beschriebenen Zielkonflikt, weshalb die Notwendigkeit aktiver Federungssysteme steigt.

Um diesen Zielkonflikt zu lösen, werden aktive Elemente zwischen Rad und Aufbaumassee integriert, sodass das Führungs- bzw. Störverhalten durch externe Energiezufuhr aktiv verbessert werden kann. In vorliegendem Forschungsvorhaben sind hierfür me-

chatronische Module, u. a. vier Aktuatoren zwischen den einzelnen Rädern und der Aufbaumassee, vorgesehen.

In vorangegangenen Arbeiten wurde bereits ein Hardware-in-the-Loop-Prüfstand konzipiert und eine Regelung anhand eines linearen Viertelfahrzeugmodells ausgelegt [2]. In diesem Beitrag wird auf die Optimierung der aktiven Federung durch Erweiterung des Modells eingegangen. Hierbei wird die Modellgültigkeit vergrößert und die Wirksamkeit der Regelung erhöht.

3 Modellbildung des McPherson Federbeins

Grundlage des mechatronischen Entwicklungsprozesses ist ein hinreichend genaues Modell des zu regelnden Systems. Für die Vertikaldynamik wird häufig das lineare Viertelfahrzeugmodell als gute Näherung der Realität zur Reglerauslegung verwendet [3]. Hierbei werden die Bewegungen der gefederten und ungefederten Masse als reine Vertikalbewegung modelliert. Beim McPherson Federbein führt die ungefederte Masse jedoch zusätzlich zur linearen Bewegung eine überlagerte Rotationsbewegung aus, welche signifikanten Einfluss auf das Systemverhalten ausübt [4], [5].

Das im Folgenden vorgestellte Modell berücksichtigt in Abgrenzung zu den bereits veröffentlichten Modellen die Aktorik der Vertikaldynamikregelung und ermöglicht somit eine bessere Analyse der Regelstrecke sowie eine einfachere Reglersynthese.

3.1 Aufstellung der Bewegungsgleichungen

Das verwendete physikalische Ersatzmodell des Viertelfahrzeugs ist in Abbildung 3 dargestellt. Das System besitzt zwei Freiheitsgrade: die vertikale Aufbau- bewegung z_A und den Drehwinkel des Querlenkers θ sowie zwei Eingangsgrößen: die Straßenanregung z_E sowie die Auslenkung des Federfußpunktes z_p . Die Nulllage des Querlenkerwinkels liegt in Flucht mit der y-Achse. Bei Einfederung ist der Querlenker um den Winkel θ_0 verdreht. Die Massen des Querlenkers und des Reifens werden zur ungefederten Masse mit dem Schwerpunkt C zusammengefasst. Das Feder- bein ist in den Punkten A und B an den Aufbau bzw. den Querlenker angebunden. Die wichtigsten ge- troffenen Vereinfachungen sind:

- Das Federbein ist masselos
- Es tritt nur viskose Reibung auf
- Der Reifen wird als gefederte Masse modelliert, Nichtlinearitäten und Dämpfung werden vernach- lässigt
- Die Aufbaudämpfung wirkt parallel zur Federfuß- punktverstellung

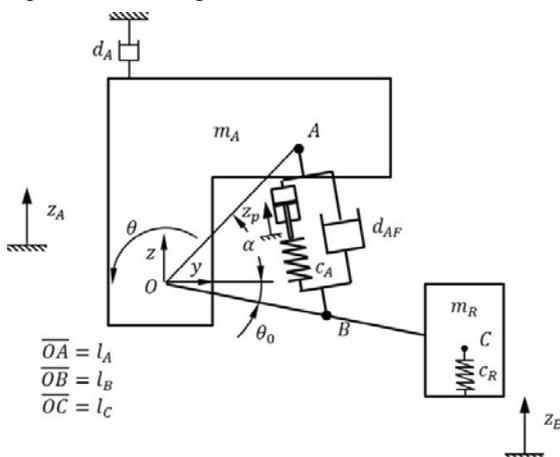


Abbildung 3. Physikalisches Ersatzmodell des McPherson Federbeins

Die Bewegungsgleichungen des Systems werden nach dem Lagrange-Ansatz 2. Art bestimmt. Die generalisierten Koordinaten werden zu $q = [z_A \ \theta]$ gewählt. Aus dem Energiegleichgewicht zwischen kinetischer Energie E_{kin} und potenzieller Energie E_{pot} , welches durch die Lagrangefunktion L beschrieben wird, ergeben sich die Bewegungsgleichungen.

$$L = E_{kin} - E_{pot} \quad (1)$$

$$0 = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} + \frac{\partial P_{dis}}{\partial \dot{q}_i} \quad (2)$$

Wobei sich die kinetische und potenzielle Energie sowie die Dissipationsleistung P_{dis} nach folgenden Gleichungen ergeben:

$$E_{kin} = \frac{1}{2} m_A \dot{z}_A^2 + \frac{1}{2} m_R (\dot{y}_C^2 + \dot{z}_C^2) \quad (3)$$

$$E_{pot} = \frac{c_A}{2} \Delta l_F^2 + \frac{c_R}{2} (z_C - z_E)^2 \quad (4)$$

$$P_{dis} = \frac{1}{2} d_{AF} \Delta \dot{l}_D^2 + \frac{1}{2} d_A \dot{z}_A^2 \quad (5)$$

Die Position und Geschwindigkeit des Punktes C ergeben sich aus der Kinematikfunktion, die aus den Freiheitsgraden und der geometrischen Beziehungen durch eine geeignete Koordinatentransformation bestimmt werden kann.

3.2 Parameteridentifikation und Modellvalidierung

Um die Parameter des Modells im Frequenzbereich zu identifizieren und für den später verwendeten Zustandsbeobachter müssen die nichtlinearen Bewegungsgleichungen linearisiert werden. Die Darstellung erfolgt in Form eines linearen Zustandsraummodells:

$$\begin{aligned} \dot{\underline{x}} &= \underline{A} \underline{x} + \underline{B} u \\ \underline{y} &= \underline{C} \underline{x} + \underline{D} u \end{aligned} \quad (6)$$

Als Zustandsgrößen werden die Aufbau- position und -geschwindigkeit sowie der Querlenkerwinkel und dessen Geschwindigkeit gewählt, sodass sich der Zustandsvektor wie folgt ergibt:

$$\underline{x} = [z_A \ \dot{z}_A \ \theta \ \dot{\theta}]^T \quad (7)$$

Als Eingangsgrößen werden die Straßenanregung sowie die Zylinderfußpunktposition verwendet:

$$\underline{u} = [z_E \ z_p]^T \quad (8)$$

Für die Modellidentifikation wird die Aufbau- position als Ausgangsgröße gewählt. Die Dynamikmatrix \underline{A} sowie die Eingangsmatrix \underline{B} ergeben sich durch partielle Ableitung der Bewegungsgleichungen nach den Zustands- bzw. Eingangsgrößen. Der Ausgang ergibt sich direkt aus den Zustandsgrößen. Daher sind keine Ableitungen zur Bestimmung der Ausgangsmatrix \underline{C} notwendig. Der Durchgriff \underline{D} ergibt sich zu null.

Die Parameteridentifikation erfolgt durch eine Fre- quenzganganpassung (Abbildung 4). Anschließend wird zur Validierung der Nichtlinearitäten das Simu-

lationsergebnis im Zeitbereich mit Messungen verglichen (Abbildung 5).

Abbildung 4 stellt den Frequenzgang des theoretischen und gemessenen Modells gegenüber. Hieraus wird ersichtlich, dass eine gute Übereinstimmung, insbesondere im Amplitudengang, vorliegt.

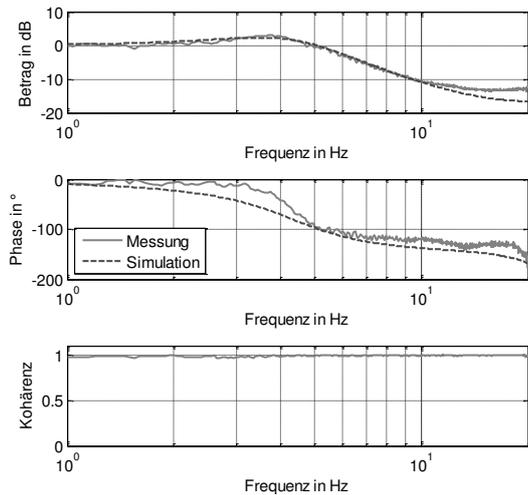


Abbildung 4. Gegenüberstellung von gemessenem und modelliertem Frequenzgang

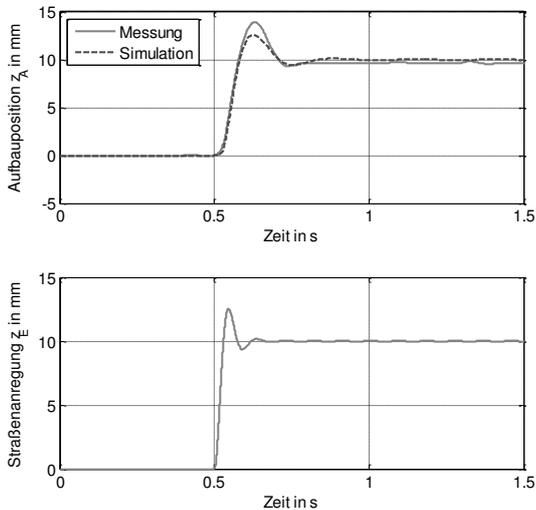


Abbildung 5. Gegenüberstellung der gemessenen und simulierten Sprungantwort bei Anregung durch die Straße im Zeitbereich

In Abbildung 5 sind die Ergebnisse im Zeitbereich bei einer sprungförmigen Änderung der Straßenanregung auf 10 mm dargestellt. Wie bereits im Frequenzbereich stimmt das theoretische Modell gut mit dem realen System überein. Die Ergebnisse (Abbildung 6) bei Anregung durch den Fahrwerkszylinder bestäti-

gen ebenfalls die gute Übereinstimmung. Das erstellte Modell kann somit das kinematische und dynamische Verhalten des realen Systems gut repräsentieren.

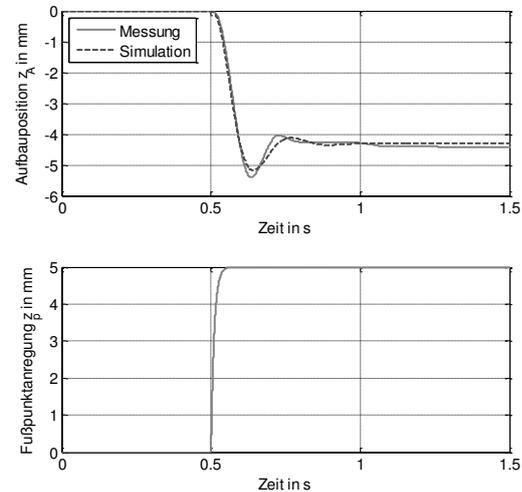


Abbildung 6. Gegenüberstellung der gemessenen und simulierten Sprungantwort bei Anregung durch den Federungsaktor im Zeitbereich.

4 Modellbasierte Reglersynthese

Anhand des aufgestellten Modells erfolgt im folgenden Abschnitt die Reglersynthese. Es wird zunächst das Regelungskonzept vorgestellt und anschließend die Bestimmung der Reglerparameter beschrieben.

4.1 Reglerstruktur

Abbildung 7 illustriert die hierarchische Reglerstruktur der Vertikaldynamikregelung. Die globale Aufbauregelung besteht hauptsächlich aus den drei Funktionsmodulen *Entkopplung*, *Abstimmung* und *Verkopplung*. Die *Entkopplung* stellt die zu regelnden Zustandsgrößen bereit. Hierzu wird ein Extended Kalmanfilter verwendet. Im Funktionsmodul *Abstimmung* wird die Modalkraft, welche auf den Aufbau wirken soll, nach Skyhook-Prinzip berechnet:

$$F_{A,soll} = c_{A,v}\Delta z_A + d_{A,v}\Delta \dot{z}_A + c_{\theta,v}\Delta \theta + d_{\theta,v}\Delta \dot{\theta} \quad (9)$$

Abbildung 8 veranschaulicht den Skyhook-Algorithmus mittels zwei virtueller Feder- / Dämpferelemente: ein lineares für die Aufbaubewegung und ein rotatives für die Querlenkerbewegung. Beide Elemente wirken der jeweiligen Bewegungsrichtung entgegen. Die benötigte Ableitung der Führungsgrößen erfolgt durch DT1-Glieder mit einer Zeitkonstante von $T = 0,001$ s.

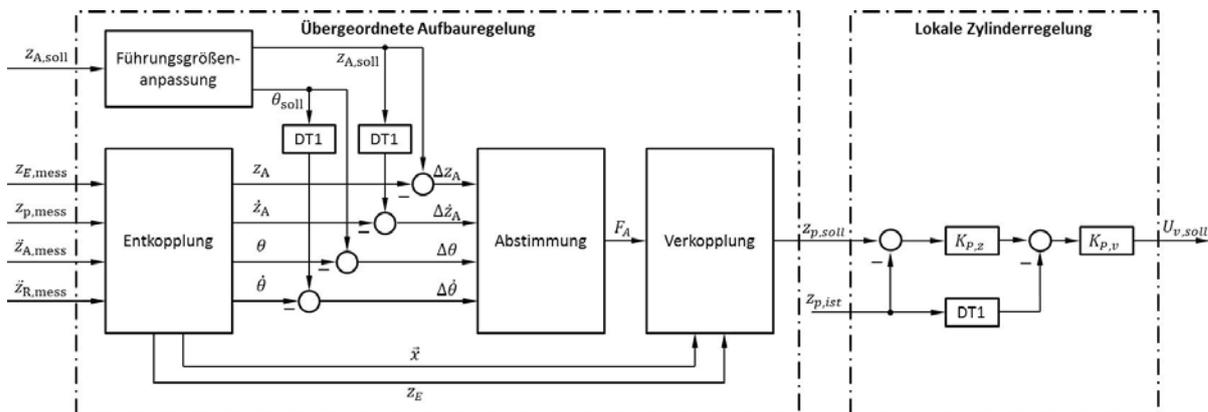


Abbildung 7. Reglerstruktur der aktiven Fahrzeugfederung

Aus der Modalkraft $F_{A,soll}$ wird im Funktionsmodul *Verkopplung* durch ein inverses Fahrzeugmodell eine Sollposition $z_{p,soll}$ für die bereitgestellt.

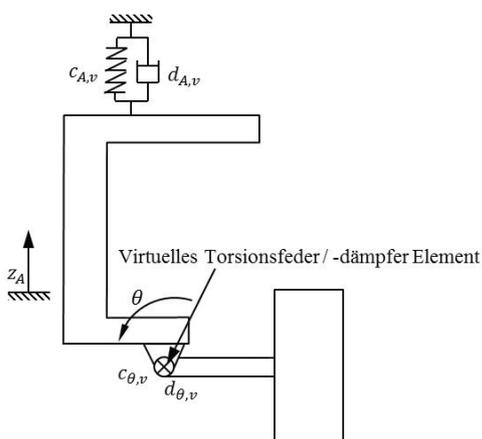


Abbildung 8. Veranschaulichung des aktiven Federungsalgorithmus

Der Globalregelung unterlagert ist die lokale Regelung der Aktorsysteme angeordnet. Hierbei handelt es sich um eine zweifache Kaskadenregelung aus Zylinder-geschwindigkeit- und Position.

4.2 Auslegung der Reglerparameter

Die Reglerparameter werden durch Polvorgabe der Aufbauabewegung bei Führungsanregung bestimmt. Als Referenzmodell dient ein PT2-Glied $G_A(s)$ mit einer Eigenfrequenz von $f_A = 1 \text{ Hz}$ und einem Lehrschen Dämpfungsmaß von $\zeta_A = 0,7$. Die Reglerparameter werden überschlägig nach Gleichung 10 bestimmt [6]:

$$\begin{aligned} c_{A,v} &= 0 \\ d_{A,v} &= 4\pi f_A \zeta_A m_A \\ c_{\theta,v} &= -(2\pi f_A)^2 m_A l_C \\ d_{\theta,v} &= 0 \end{aligned} \quad (10)$$

Nach der überschlägigen Ermittlung der Parameter werden diese anhand des nichtlinearen Modells optimiert. Hierbei wird folgende Zielfunktion minimiert:

$$e = \int_0^T (z_{A,soll} * G_A(s) - z_A)^2 dt \quad (11)$$

Das Ergebnis der Optimierung sowie der Verlauf der Zielfunktion ist in Abbildung 9 dargestellt. $z_{A,soll}$ ändert sich zum Zeitpunkt $t = 0$ sprunghaft auf 5 mm und zum Zeitpunkt $t = 3 \text{ s}$ auf 0 mm . Der Fehler ist vernachlässigbar gering. Als Ergebnis der Optimierung erhält man die geeigneten Reglerparameter.

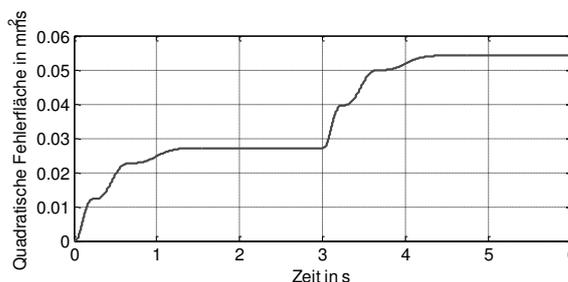
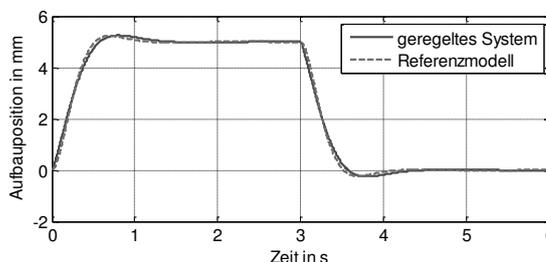


Abbildung 9. Vergleich zwischen PT2 Referenzmodell und geregelter Aufbauabewegung

5 Analyse des Systems unter Echtzeitbedingungen

Der am Modell ausgelegte Algorithmus der Vertikaldynamikregelung wird am vorhandenen HiL-

Prüfstand implementiert und in Betrieb genommen. Nach der Analyse des Systemverhaltens kann der Regelalgorithmus unter Echtzeitbedingungen weitergehend erprobt und optimiert werden.

Die Führungssprungantwort des geschlossenen Regelkreises ist zusammen mit der Antwort des zur Auslegung verwendeten PT2-Systems (Referenz) in Abbildung 10 dargestellt. Die gemessene Aufbauposition stimmt gute mit dem Referenzmodell überein.

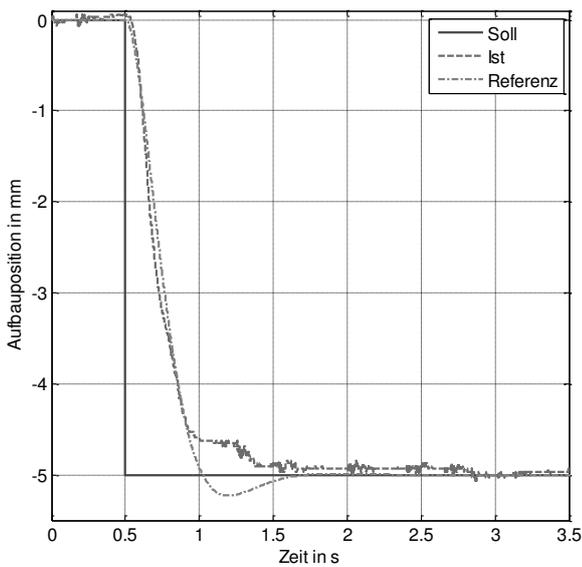


Abbildung 10. Führungssprungantwort des geregelten Systems

Als Störgröße für die Vertikaldynamik ist unter anderem die Straßenunebenheit zu identifizieren. Zur Analyse des Störverhaltens im Zeitbereich wird ein Sollwertsprung des Anregungszyinders auf 10 mm verwendet. Abbildung 11 stellt das Ergebnis dar. Zu erkennen ist eine deutliche Reduzierung der Schwingungen durch die aktive Federung.

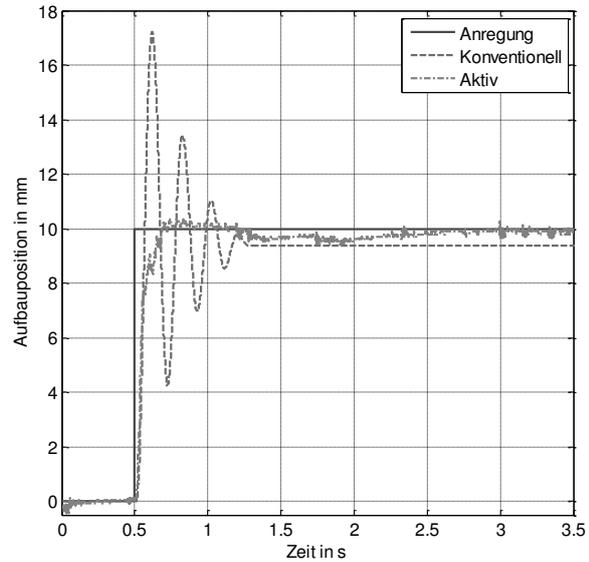


Abbildung 11. Sprungantworten des unregulierten und geregelten Systems bei Störanregung durch die Straße

Auch die Gegenüberstellung des gemessenen Frequenzgangs des konventionellen Fahrwerks und des aktiven Fahrwerks (Abbildung 12) bestätigt die Kenntnisse aus dem Zeitbereich. Die Resonanzüberhöhung des konventionellen Fahrwerks wird komplett ausgeregelt. Das bedeutet, dass das dynamische Verhalten des Systems durch die ausgelegte Regelstruktur wesentlich verbessert wurde.

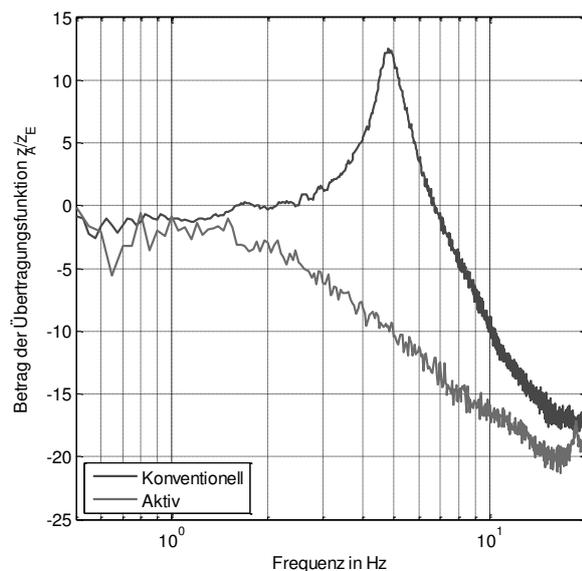


Abbildung 12. Störübertragungsfunktionen des unregulierten und geregelten Systems

Ein anderes Ziel der Regelung ist es, den Fahrkomfort bei guter Dynamik durch die Regelung zu verbessern. Als Maß für den Fahrkomfort gilt der nach [7] bewertete Betrag der Aufbaubeschleunigung. Je größer die Fläche unter der Betragskennlinie ist, desto schlechter ist der Fahrkomfort. Abbildung 13 stellt die Messergebnisse für das konventionelle sowie das aktive Fahrwerk dar.

Es ist zu erkennen, dass durch die Vertikaldynamikregelung eine signifikante Verbesserung des Fahrkomforts um 60% erzielt wird.

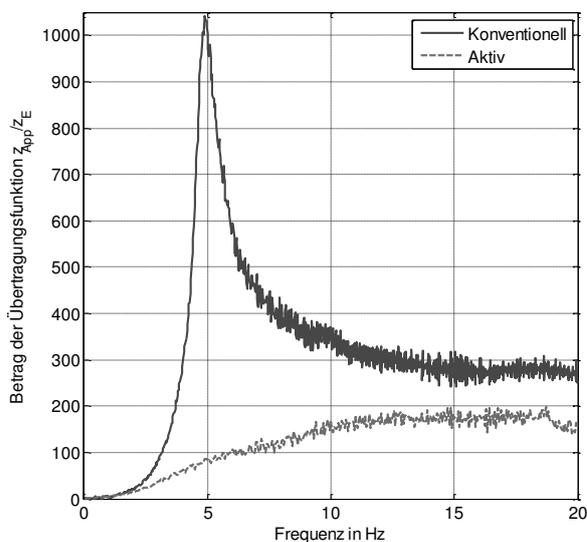


Abbildung 13. Gegenüberstellung des Betrags der Aufbaubeschleunigung

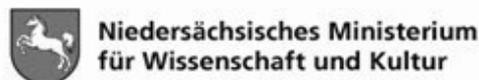
6 Zusammenfassung

In diesem Beitrag konnte gezeigt werden, dass die Modellierungstiefe des verwendeten Fahrzeugmodells zur modellbasierten Reglerauslegung eine wichtige Rolle spielt. Durch weitergehende Berücksichtigung der Fahrphysik wurde ein nichtlineares Fahrzeugmodell zur Abbildung eines McPherson-Federbeins erstellt, welches zu einer erheblichen Übereinstimmung zwischen Modell und Realität führt. Messtechnische Analysen am HiL-Prüfstand im Zeit- als auch im Frequenzbereich zeigten, dass die Fahrdynamik und der Fahrkomfort durch die optimierten Regelalgorithmen wesentlich verbessert wurden.

7 Danksagung

Diese Arbeit wurde im Rahmen der Forschungsprofessur (FH!) von Frau Prof. Dr.-Ing. Xiaobo Liu-Henke durch das Niedersächsische Ministerium für Wissenschaft und Kultur und der Volkswagen Stif-

tung unter dem Förderkennzeichen VWZN2811 gefördert. Die Verantwortung für den Inhalt liegt bei den Autoren. Für die Förderung bedanken sich die Autoren herzlichst.



Niedersächsisches Ministerium
für Wissenschaft und Kultur



VolkswagenStiftung

8 Literatur

- [1] Gysen, B.; Paulides, J.; Janssen, J.; Lomonova, E.: *Active Electromagnetic Suspension System for Improved Vehicle Dynamics*, in Transaction on vehicular Technology, vol. 59. IEEE, Ed., pp. 1156–1163.
- [2] Scheele, M.; Jacobitz, S.; Liu-Henke, X.: *Mechatronische Entwicklung eines HiL - Prüfstands zur Erprobung einer aktiven Federung*, in ASIM STS/GMMS Workshop, vol. 2014, 2014.
- [3] Mitschke, M.; Wallentowitz, H.: *Dynamik der Kraftfahrzeuge*. Wiesbaden. Springer Vieweg, 2014.
- [4] Hong, K.-S.; Jeon, D.-S.; Sohn, H.-C.: *A new modeling of the Macpherson suspension system and its optimal pole-placement control*, Proceedings of the 7th Mediterranean conference on control and automation, 1999, pp. 559–579.
- [5] Fallah, M. S.; Bhat, R.; Xie, W. F.: *New nonlinear model of macpherson suspension system for ride control applications*, American Control Conference, IEEE, Ed., 2008, pp. 3921–3926.
- [6] Liu-Henke, X.: *Mechatronische Entwicklung der aktiven Feder-/Neigetechnik für das Schienenfahrzeug RailCab*. VDI- Fortschritt-Berichte, Reihe 12, Nr. 589, VDI-Verlage, Düsseldorf, 2004
- [7] DIN EN ISO 8041:2006-06, *Human response to vibration*.

