

TAGUNGSBAND

ASIM SST Berlin 2014

22. Symposium Simulationstechnik

HTW Berlin 3. - 5. September 2014

organisiert durch:



ASIM / GI * HTW Berlin * Hochschule Wismar

Tagungsband

ASIM 2014

22. Symposium Simulationstechnik

**3. bis 5. September 2014
HTW Berlin**

**Zusammenfassung der Beiträge
Jochen Wittmann (Hrsg.)
Christina Deatcu (Hrsg.)**

Arbeitsgemeinschaft Simulation ASIM in der Gesellschaft für Informatik GI

INHALT

Vorwort	1
Plenarvorträge	
Softwarearchitektur für parallele Simulationsumgebungen <i>L. P. Feldmann, V.A. Svyatnyy, M. Resch, M. Zeitz</i>	3
Methoden I	
Numerical Treatment of Dynamical Systems using Overdetermined Regularizations <i>Lena Scholz, Andreas Steinbrecher</i>	9
Numerische Simulation des runden Freistrahls unter Anwendung des SST-Turbulenzmodells <i>Ansgar Ringleb, Wolfgang Schlüter, Günter Wozniak</i>	17
Comparison of Regularisation Approaches for High Index DAEs <i>Carina Poll, Irene Hafner, Bernhard Heinzl, Felix Breitenecker</i>	23
Convection-diffusion Simulation in two different dimensions <i>Stefanie Winkler, Martin Bicher, Felix Breitenecker</i>	31
Produktion und Logistik I	
Anwendung des Model-Based Systems Engineering zur Unterstützung der Simulation in Produktion und Logistik <i>Alexander Schlegel</i>	39
Konzept zur integrierten Fertigungs- und Logistikplanung im Kontext der digitalen CFK-Produktionsentwicklung <i>Michael Lütjen</i>	49
Using Simulation to improve Cluster tool Performance in an LED manufacturing line <i>Fabian Suhrke, Falk S. Pappert, Robert Janik, Oliver Rose</i>	55
Webbasierte Animation von Simulationsläufen auf Basis des Core Manufacturing Simulation Data (CMSD) Standards <i>Soren Bergmann, Florian Parzefall, Steffen Straßburger</i>	63
Umwelt- und Geowissenschaften I	
A Formal Approach to Combine Models of Archaeological Sites <i>Philipp Pichler, Kerstin Kowarik, Niki Popper, Gabriel Wurzer, Hans Reschreiter, Anke Bacher, Bernhard Heinzl</i>	71
Agentenbasierte Simulation zur Versorgung der Bevölkerung im prähistorischen Hallstatt – Ausgewählte Arbeitsabläufe und deren Zusammenhang <i>Johannes Tanzler, Philipp Pichler, Klaus Fankhauser, Dominik Brunmeir, Kerstin Kowarik, Hans Reschreiter, Anke Bacher</i>	79
Anbindung eines dynamischen Modells zur Plattentektonik mit dem ArcMap-Werkzeug ModelBuilder <i>Jochen Wittmann, Karsten Bratz</i>	83
Technisch-wirtschaftliche Optimierung von hybriden Inselnetzen in hoher zeitlicher Auflösung <i>Hendrik Huyskens, Philipp Blechinger</i>	89

Metamodellierung I

- Ontology for Objective Flight Simulator Fidelity Evaluation 93
Umut Durah, Artur Schmidt, Thorsten Pawletta
- Development of a Container terminal Simulation meta-model 101
Ann-Kathrin Lange, Giovanni Pirovano, Rosella Pozzi, Tommaso Rossi

Produktion und Logistik II

- Hybride Modellierung fertigungstechnischer Prozessketten mit Energieaspekten in einer ereignisdiskreten Simulationsumgebung 109
Artur Schmidt, Thorsten Pawletta
- Entwicklung einer Fertigungssimulation für einen Schmelz- und Druckgussbetrieb zur Steigerung der Energieeffizienz und Prozesssicherheit 117
Sven Hirschberg, Wolfgang Schlüter, Ansgar Ringleb

Technische Systeme I

- Konzeption einer aktiven Vorspur für ein Elektrofahrzeug 125
Xiaobo Liu-Henke, Matthias Fritsch, Florian Quantmeyer
- Integration eines DP-Moduls in maritime Aus- und Weiterbildungssimulatoren 133
Bettina Kutschera, Matthias Markert, Jens Ladisch, Bernhard Lampe

Metamodellierung II

- Vorstellung eines vollautomatischen Simulationswerkzeugs zur Bewertung von Fahrzeugkonzepten 137
Mark Krausz, Matthias Zimmer, Hans Christian Reuss
- Ontologie-unterstützte Systemmodellierung und Simulation in MATLAB/Simulink 145
Thorsten Pawletta, Daniel Pascheka, Artur Schmidt, Sven Pawletta
- Conceptual Model and Perdurantist Modeling with Reasoning 155
Mehmet Fatih Hocaoglu

Supply Chains

- Optimierung von Supply Chains bei saisonalen Schwankungen 161
Hans-Peter Barbey
- Dimensionierung von Sicherheitsbeständen unter Einbeziehung kurzfristiger Bedarfsschwankungen in automobilen Zulieferbeziehungen 167
Tim Gruchmann, Andre Krischke
- Entscheidungsunterstützung in der Optimierung der Supply Chains von Stahl- und Walzwerken 175
Lothar März

Umwelt- und Geowissenschaften II

- Simulation von Abwassersystemen mit der MOSKAN Software 181
Jan Studzinski, Patryk Wójtowicz
- Ein Prozessmodell zur Aufbereitung und Analyse von Sichtungsdaten von Delfinen und Walen vor La Gomera 187
Jochen Wittmann, Fabian Ritter, Aljoscha Marcel Everding, Jonas Amelong, Lisa Jermann
- Entwicklung eines Logistiktools zur Grobmodellierung der Positionierung von Sammel- und Verwertungsstellen von organischem Abfall 195
Thomas Baur, Sascha Fieting, Christopher Witte, Dustin Müller, Jochen Wittmann, Ruth Scheidegger

N ethoden JJ

Uystfm.Eynbmjds.Ejbhbrbmmf bls (Si ysjdbl O oefljnh(. fjn f Grbhf efr Mbusbljtät <i>Odsdp It nf lZr</i>	203
Zjrtubl Utodi bstjd Ufnsors gor I ycrje UystfmsAO utubl Kglufndf cftwffn Dontjnuous bne Ejsdrftf Uystfm Sbrts <i>CkZt clZ J pt k, GpZgZl Hnpsnm</i>	211
Bpprobdi fs gor Utbf Fvnt I bneljnh cy Ujmulbtjon Blhorjti m bne vjb O oefl Efsdrjptjon <i>AncpdZr J äpnlp, FdklwBpdlndmbj dp</i>	219
Fr fjhnjslokbljsjfrunh eurdi ejf I fnon.O fti oef <i>FpZny Opdxrdp, BdpngZpc Hdlnyk, L ZssglZr Qüüklp, FdklwBpdlndmbj dp</i>	225

P ptimierung . Tcheduling

B rocust sdi feulf gor Ontpflljfrs Vrbmwby nftwork <i>Nkhudp T kplbg, DZnhlk Köbj dpZsg, Ev Zlc Rodbj dnl dxdp</i>	229
O ultjskblfnbnsbtz gir rf bktjvf Srozfss. une Belbugplbnunh <i>L ZssglZr Bncd, Unkj dp Bdpj gZgm</i>	239
B UjmI furjstjd Grbmwork bs b Ef djsjon Uupport Uystfm gor lbrhf Nohjstjds P ftworks wjti domplx MSK <i>FdklwDpnrr, L Zpj t r QZad</i>	247
Ujmulbtjon.cbsfe Ejspbtdi jnh jn Loc Ui ops <i>SZn XgZnf, Nkhudp Qnrd</i>	255

Uechnische Tvsteme JJ . CGD

Bngrefrunhfn une Fjnsbtzmöhljdi kfjtn efr Gljesjmulbtjon jn efr grüi fn Si bsf efs Sroeuktntwjdklunhsprozfssfs <i>RsdZmJ pnskk, Odsdp Rslbg, Gt msgdp QdlnpZps</i>	259
Ujmulbtjon jn efr Gljevfrgi rfntstf di njk . Zjrtufllf Kcfrtrjcnbi mf jn fjnfr i ftfrohfnfn Uogwbrflbnesdi bg <i>FpZny Rbgämudpf, L ZpbdkMZv pZsg, Idnr Izj dk</i>	267
Fggjzjntfr Zfrhlfjdi von DGE. Ujmulbtjonfn une Fxpfrjmfntblebtfen eurdi Utnebrejsjfrunh une Butombtjsjfrunh <i>AncpdZr Opnars, Amrf Zp Qlnf kla, V nkf Znf Rbgkösdp</i>	275
Vi frmjsdi f Ujmulbtjon von Cbttfrjfsystf mfn mjttf ls DGE sowjf efrfn Zfrhlfjdi mjt fjnf m vfrfjngbdi tfn Ujmulbtjonsmoefll <i>Bdpnc Rbgdlc dxdp, V nkf Znf Rbgkösdp</i>	283

Uechnische Tvsteme JJJ

Wtfrsudi unh efr Ukbljrcbrkfjt von Cbttfrjfmnbhfmfntsyf mfn mjt jlgf von DBF. Ujmulbtjon <i>L ZssglZr Qnbg, Qnads Fdlnc, FknplZmPt Znl dxdp, WlZnan Kl -Hdnj d</i>	291
Sfrgormbndf Srfejdtjon bne Rptjmjzbtjon gor Kcustrjbl Ujfvfs cy UjmulbtjonAb two.tjfr Bpprobdi <i>CgprslZmI Zdf dp, QdyZ Hnt rrdlnh, L Znpdc Hdpsv lf, Sgnl Zr Hnadp, Qt cnke L Zpbdk Föbgrklm</i>	299

Sroduktion und Mgistik .III

Cfrr Ejstrjcutjon Hbm f . Frgolhrfjdi ebk Tfhflunhstf di njk 307
Onke Rsdm f dp, HZnr-Odsdp BZpdx

Ufrfnproeuktionfscfhlfjtf nef O btrjblgusssjmulbtjon gur O bbi bltjhkfjtsoptjmjfrunh jn efr komplxfn
Mbrossfrjfgfrtjhunh 317
Anj d Bdbj l Zm L ZpsmBngm Odsdp Gt rs

N edivn J

O oeflljnh Rcfstjy jn Bustrjb wjt Uystfm Eynbmjds 325
BZpaZpZ Gknbj, OZspbj Elmymf dp, FdkhwBpdlhdmdbj dp

Ejf Ebrstflunh efr Dompljbnf jn efr O oeflljfrunh efs I frz.Mrfjslbug Uystfms 333
SZl ZpZ Unapt aZ, AmcpdZr J äpndp, BdpngZpc HZl dsndp, FdkhwBpdlhdmdbj dp

N ethoden .III

O BSNKU . O btrjxorjntjrtf Ujmulbtjon- wjfe fr bughfnommf n 337
V kphlc Sdssv dlkdp

O oeflljfrunh von Fvolutjons Uystfmfn mjt Utodi bstjsdi fn a fillulärfn Butombtf n 343
Gönsdp Rbgndbj dmpdhgdp, Mj h Onoodp, FdkhwBpdlhdmdbj dp

Grüi zftjhf Ufrvfrbusgblsfrkfnunh mjttf ls Zori frsbhfmfti oefn une Rutljfrbnblysf 351
L ZssglZr V ZrsZm FdkhwBpdlhdmdbj dp, L lbgZdk KZnc rhdc

Verteilt . Sarallel

a ur Fntwjdklunh efs Hlfjdi unhslösfrsucsystfms efr vfrtjltfn pbrblflfn Ujmulbtjonsumhfcunh 357
Unkncxl xp J t rgnZpnhj n, L lbgZdk Qdrbg, Unkncxl xp RuiZsnki, RsdZm V drndp

O oefl.cbsfe pbrblfljzbtjon ogejsdrftf trbgjd Ujmulbtjon moefls 365
Nkudp T kphbg, DZntdk Köbj dpZsg, Ev Zlc Rodbj dnl dxdp

Bloue Domputjnh Bppljfe to Dbljcrbtjon og Glooe Ujmulbtjon O oefls 373
L Zpbn Bpdssrbgmldp, Bdpnc Oösyndp, FpZnj Ft bgr-J hsnv rj h

N edivn JJ

Ujmulbtjnh vfntjrdulbr flbstbnf wjt b i fbrt.brtfrjbl jntfrbdtjon moefl 381
*AnhZ Gdprsdnl Zxdp, BdpngZpc HZl dsndp, RsdogZntd OZppZf g, Sgnl Zr V dadp,
Rtlf phlc V Zrrdpsgd pdp*

Fvblubtjon og Tjsk Gbdors gor Sbrbmftjzbtjon og Dbndfr O oefls 389
AmcpdZr BZt dp, Gönsdp XZt mdp, Cgphsnog TpZbg, FdkhwBpdlhdmdbj dp

Kmplfmfntbtjon- tfstjnh- bne fvblubtjon ogdfntfr sflfdtjon mfti oes gor supfrvjsfe rbejbl cbsjs nftworks to
fni bndf crfbst dbndfr bnblysj 393
L ZnZmMZnZuZsh Bnphr Q. BpZbln

Forschungsgebiet: Parallele Simulationstechnik

L.P. Feldmann¹, V.A. Svyatnyy¹, M. Resch², M. Zeitz³

¹Nationale Technische Universität Donezk, Artemstr. 58, 83000 Donezk, Ukraine

²Höchstleistungsrechenzentrum Stuttgart, Nobelstr. 19, 70569 Stuttgart

³Institut für Systemdynamik Universität Stuttgart, Pfaffenwaldring 9, 70550 Stuttgart

feldman@pmi.dgtu.donetsk.ua, svjatnyj@cs.dgtu.donetsk.ua,

resch@hlrs.de, zeitz@isys.uni-stuttgart.de

Das Forschungsgebiet der parallelen Simulationstechnik wird im Rahmen einer langjährigen Kooperation von Informatikern und Simulationstechnikern der Nationalen Technischen Universität Donezk und der Universität Stuttgart untersucht. Im Mittelpunkt stehen dabei die Entwicklung einer leistungsfähigen und benutzerfreundlichen Simulationsumgebung sowie die Untersuchung von numerischen Verfahren zur effizienten Nutzung der heute verfügbaren parallelen Hard- und Softwareressourcen. In dem Beitrag werden Konzepte für virtuelle Simulationsmodelle, parallele Simulatoren und Simulationssprachen sowie anwendungsorientierte Simulationsumgebungen vorgestellt.

1 Einleitung

Die Anwendungsgebiete für die parallele Simulationstechnik reichen sich von den "traditionellen" technischen dynamischen Systemen (DS) mit konzentrierten und mit verteilten Parametern (DSKP, DSVP) über biotechnologische und ökologische Prozesse bis hin zu den zu projektierenden und zu optimierenden Unternehmens- und Wirtschaftsstrukturen. Als Folge von steigenden Anforderungen an die Qualität, Sicherheit und Wirtschaftlichkeit der DS-Prozesse werden immer genauere und damit auch komplexere Simulationsmodelle benötigt. Bei dieser Entwicklung ergibt sich in einer natürlichen Weise eine Komplexitätsschwelle für die Implementierung und Anwendung der Simulationsmodelle, die nur durch den Einsatz neuer Simulationswerkzeugen überwunden werden kann. Hierzu gehören u.a. parallele Höchstleistungsrechner der MIMD-Architektur mit eng und lose gekoppelten Prozessoren sowie mit einer hybriden Speicherorganisation. Die effiziente Nutzung von diesen Simulationswerkzeugen erfordert eine benutzungsfreundliche Organisation der parallelen Ressourcen sowie eine weit entwickelte Simulationssoftware, die neue Ansätze zur Modellierung und Simulation zur Verfügung stellt. Die hierbei zu lösenden simulationstechnischen Probleme gehören zu dem in dem Beitrag behandelten Forschungsgebiet der **parallelen Simulationstechnik**.

2 Modellierungskonzept für dynamische Systeme (DS)

Die in Abb. 1 aufgeführten Anwendungsgebiete besitzen spezifische Komponenten und Prozesse, die durch die folgenden Eigenschaften komplexer dynamischer Systeme (KDS) gekennzeichnet sind: Vielzahl und Mehrfachheit von Verbindungen, räumliche Verteilung und Abhängigkeiten, statisches und dynamisches Verhalten, Einfluss von Nichtlinearitäten, konstante und variable Modellparameter, hierarchische Anordnung von Automatisierungssystemen, unterschiedliche Betriebsarten usw.

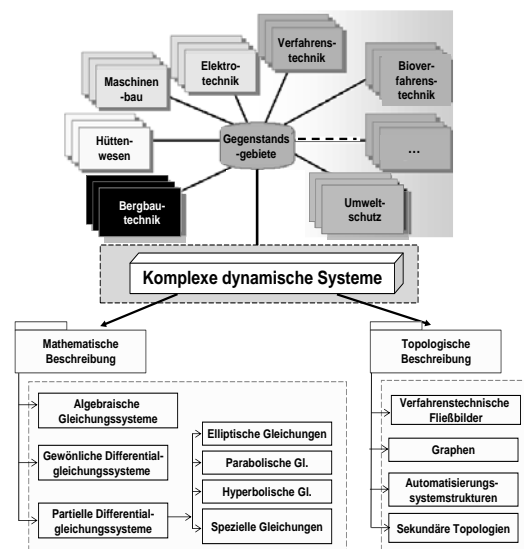


Abbildung 1. Anwendungsgebiete und Beschreibungen von komplexen dynamischen Systemen

Trotz der großen physikalischen Vielfalt der verschiedenen Anwendungsbereiche der parallelen Simulationstechnik werden ihre Topologien mit einer begrenzten Anzahl von Modellierungstechniken beschrieben, wie dies in Abb. 1 dargestellt ist: verfahrenstechnische Fließbilder, Graphen für Netzobjekte, Strukturbilder für Automatisierungssysteme sowie sekundäre Topologien aufgrund von DSVP-Approximationen. Das Hauptziel der Topologiebeschreibung betrifft die DS-Struktur sowie deren mathematische Beschreibung als wesentliche Voraussetzung für eine automatische Generierung auch großer Simulationsgleichungssysteme.

3 Etappen der parallelen DSKP- und DSVP-Modellierung

Wegen der DS-Komplexität spielt bei der DSKP- und der DSVP-Modellierung die rechnergestützte Erstellung von Topologien, die automatische Generierung der Modellgleichungen und ihre Umwandlung in parallele Simulationsmodelle eine zunehmend wichtigere Rolle. Hierzu werden benutzerfreundliche Bedienoberflächen (BOF) sowie Algorithmen und Programme für die Topologieanalysatoren und Gleichungsgeneratoren benötigt.

Definition 1: Die **parallele Modellierung** von dynamischen Systemen mit konzentrierten und mit verteilten Parametern ist ein Vorgang, der zielgerichtet auf die benötigten DSKP- und DSVP-Simulationsmodelle für das verfügbare parallele Rechnersystem führt.

Definition 2: Ein **virtueller MIMD-Prozess** ist ein autonomes Programm, das den Lösungsalgorithmus für die Teilsysteme realisiert und mit den an der Lösung des Gesamtgleichungssystems beteiligten Prozessen kommuniziert.

Die **Körnigkeit von virtuellen Prozessen** charakterisiert den Umfang der Berechnungs- und Hilfsoperationen, die im Prozess programmiert werden. Diese Eigenschaft ist mit der KDS-Dekomposition (Topologie und Modelle) sowie mit der örtlichen DSVP-Approximation verknüpft.

Definition 3: **Minimalkörnige virtuelle Prozesse** befassen sich mit den Lösungsoperationen für solche Gleichungen, die sich nach der Dekomposition und der Diskretisierung der VTF-, SAS- und DNO-Komponenten ergeben.

Definition 4: Das **virtuelle parallele DSKP/DSVP-Simulationsmodell (VPSM)** ist eine Abstraktion, die aus dem bezüglich den Ableitungsvektoren und den Vektoren der gesuchten algebraischen Unbekannten aufgelösten Gleichungssystem und der nach einem Parallelisierungsansatz entwickelten Struktur der virtuellen MIMD-Prozesse für die parallele Lösung der Gleichungssystemteile besteht.

Der kleinste Gleichungssystemteil ist eine gewöhnliche Differentialgleichung (oder eine algebraische Gleichung), für deren Lösung ein minimalkörniger MIMD-Prozess zuständig ist. Die VTF-, SAS- und DNO-Gleichungsgeneratoren verwenden ausgewählte numerische Verfahren und generieren diskrete virtuelle parallele Simulationsmodelle (**DVPSM**).

Definition 5: Das **Zielrechnersystem (ZRS)** betrifft die dem Modellentwickler zur Verfügung stehende lose oder eng nach MIMD-Prinzip gekoppelte Menge der installierten Prozessoren mit dem lokalen oder gemeinsamen Speicher. Hierzu gehört auch das programmgesteuerte Verbindungsnetzwerk.

Definition 6: Der **parallele DSKP- und DSVP-Simulator** ist das Ziel der parallelen Modellierung und wird als eine Hardware/Software-Systemorganisation definiert, die den Modellierungsvorgang und das virtuelle parallele Simulationsmodell auf dem Zielrechnersystem softwaretechnisch realisiert und den Modellentwicklern sowie den Modellanwendern die benutzungsfreundliche Kommunikation mit den Modellierungs- und Simulationsressourcen erlaubt.

Definition 7: **DVPSM-Devirtualisierung** betrifft die Simulatorrealisierung auf dem gegebenen ZRS und wird entsprechend den in Abb. 2 dargestellten Etappen der parallelen Modellierung realisiert.

Die resultierenden VPSM mit minimalkörnigen Prozessen werden als paralleles Simulationsmodell der ersten Parallelitätsebene betrachtet. Diese Ebene in Abb. 2 beinhaltet die maximal mögliche Menge der Einheiten „Gleichung - Prozess“ und dient für die Untersuchungen der potentiellen Parallelität von VTF-, SAS-, DNO- und sekundären Topologien. Die Komposition der zerlegten Objekte und eine Gruppierung der minimalkörnigen Prozesse führen zur Betrachtung der höheren Parallelitätsebenen mit einer größeren Prozesskörnigkeit und entsprechenden Varianten der Prozess-Prozessor-Zuordnung. Die Apriori-Analyse muss bezüglich aller möglichen Parallelitätsebene durchgeführt werden und umfasst die Untersuchung der folgenden Fragen: Analyse des Opera-

tionenumfangs im virtuellen Prozess; Vergleich der Prozesse nach dem Kriterium „Prozesslasthöhe“; Vorschläge zur notwendigen Lastbalancierung; Analyse der Kommunikationsoperationen zwischen der Prozessen; Entwicklung des virtuellen Verbindungsnetzwerks mit dem minimal möglichen Zeitaufwand für einen Datenaustausch [10, 11].

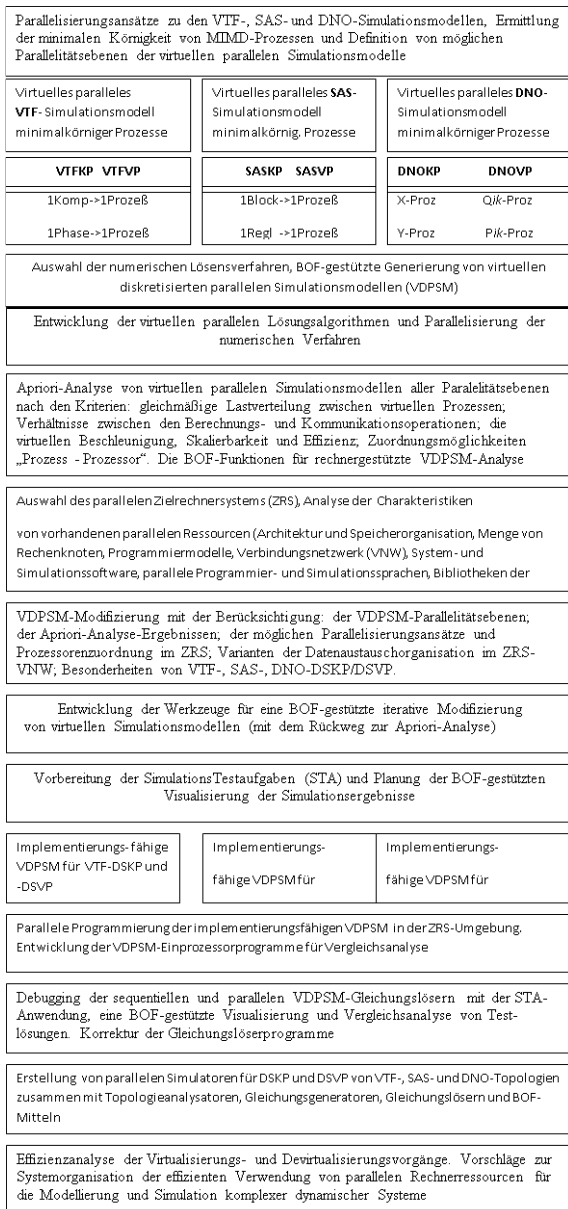


Abbildung 2. Die Etappen der KDS-Modellierung und Entwicklung von parallelen Simulatoren

4 Anforderungen an die Simulationswerkzeuge

Dynamische Systeme mit konzentrierten und mit verteilten Parametern stellen folgende Anforderungen

an die Simulationswerkzeuge [1-6, 9- 11]: Berücksichtigung der **formalen Beschreibung** komplexer dynamischer Systeme (**KDS-Modelle**), welche eine Topologiedarstellung sowie ein hochdimensionales Gleichungssystem umfasst; **Benutzungsfreundlichkeit** mit einer interaktiven Unterstützung des Modellentwicklers auf den erläuterten Etappen der Modellierung und Simulation von KDS; objektorientierte Modellspezifikation für Struktur, Verhalten und Parameter von DSKP und DSVP; objektorientiert implementierte Modellierungs- und Simulationssoftware; Bibliothek von konventionellen und parallelen numerischen Verfahren; HW/SW-Ressourcen für die Simulation von realen komplexen DSKP und DSVP; Echtzeitfähigkeit zur Kopplung mit realen KDS-Komponenten; moderne IT-Systemorganisation der Simulationsressourcen; Erstellung von anwendungsorientierten Simulationsumgebungen; Werkzeuge für die Schulung von Modellentwicklern und die Realisierung der Trainingssimulatoren; Integration mit GRID- und CLOUD-Infrastrukturen.

5 Konzeption der verteilten parallelen Simulationsumgebung (VPSU)

Definition 8: Die VPSU wird definiert als eine benutzerfreundliche Systemorganisation der Hardware, der architekturrelevanten Systemsoftware, der speziell zielgerichtet entwickelten Simulationssoftware sowie der simulationstechnisch bedingten Subsysteme, mit denen alle Etappen der parallelen Modellierung und Simulation von DSKP und DSVP unter Berücksichtigung der obigen Anforderungen unterstützt werden.

Die VPSU-Entwicklung wird als zentrales Problem der parallelen Simulationstechnik angesehen und nach der folgenden Konzeption durchgeführt: Berücksichtigung der DS-Anforderungen; Nutzung der vorhandenen parallelen Ressourcen als verteilte Rechensysteme; Entwicklung der funktionsvollständigen Simulationssoftware mit ähnlichen Eigenschaften wie bei den vorhandenen block- und gleichungsorientierten Simulationssprachen; problemorientierte VPSU sind die Teilprojekte, die allgemeine Lösungen nutzen und Ergebnisse für mögliche Verallgemeinerungen liefern; VPSU wird als komplexes HW/SW-SYSTEM betrachtet, deren Entwicklung soll auf dem OO-Ansatz mit der UML-Technologie sowie mit der Dekomposition auf bestimmte Menge der Subsysteme erfolgen [5].

Definition 9: Ein **VPSU-Subsystem** ist eine Teilkomponente der Hardware, Systemsoftware und Simula-

tionssoftware, die benutzungsfreundlich eine gewisse Gruppe von inhaltlich nahen Funktionen auf entsprechenden DSKP/DSVP-Modellierungs- und Simulationsetappen erfüllt.

Die vorgeschlagene VPSU wird in Donezk von der FCWT und vom HLRS in Stuttgart nach Abb.3 implementiert.

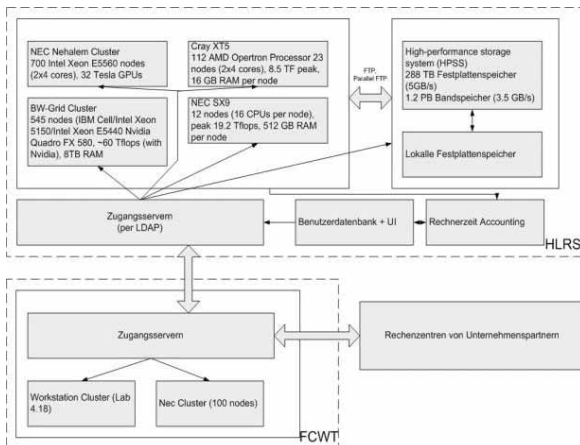


Abbildung 3. Aktueller Stand der von der FCWT und vom HLRS entwickelten VPSU-Hardwarestruktur

6 Entwicklung der VPSU-Simulationssoftware

Die Funktionalität der Simulationssoftware als der zugrundeliegenden Komponente der Simulationsumgebung ergibt sich aus den in Abb. 2 angegebenen Etappen der parallelen Modellierung und Simulation komplexer dynamischer Systemen. Die entwickelte VPSU-Simulationssoftware verteilt die benötigte Funktionalität zwischen den folgenden Subsystemen: Dialogsubsystem (DiSuS, BOF); Topologieanalyse-subsystem (TASuS); Subsystem für die Gleichungsgenerierung (SuGG); Subsystem der virtuellen parallelen Simulationsmodellen (VPSMSu); Subsystem der parallelen Gleichungslöser (SuGL); Datenaustauschsubsystem (DASu); Lastbalancierungssystem (LaBSu); Visualisierungssystem (ViSuS); Datenbanksystem (DaBSu); Subsystem für die IT-Unterstützung (SuIT). Schließlich werden die in den Bibliotheken eingebetteten parallelen Gleichungslösern nach den folgenden Ansätzen entwickelt:

1. Parallelisierung der numerischen Verfahren, die in den konventionellen Simulationssprachen erfolgreich eingesetzt werden.
2. Entwicklung von neuen parallelen Lösungsverfahren. Hierfür wurden Untersuchungen von blockartigen Ein- und Mehrschrittsverfahren hinsichtlich der

Konvergenz- und Stabilitätseigenschaften sowie zur Fehlerabschätzung durchgeführt [4, 7, 8].

7 Anwendungsorientierte VPSU

Die Entwicklung einer VPSU für ein konkretes Anwendungsgebiet ist eine praktisch wichtige Aufgabenstellung der parallelen Simulationstechnik. Im Rahmen der erwähnten Forschungs Kooperation wurden eine Simulationsumgebung für dynamische Grubenbewetterungsnetze in der DNO-Topologie sowie eine Simulationsumgebung für verfahrenstechnische Prozesse und Anlagen (VTF-Topologie) entwickelt, implementiert und untersucht [9, 10].

8 Entwicklung von parallelen Simulationssprachen

Die Entwicklung der MIMD-Systeme und OO-Ansätze hat die Standardisierung der Technologien der parallelen und verteilten Programmierung stimuliert. Allerdings muss der Modellentwickler trotz gewisser Standardisierungsfortschritte heute wie früher die parallelen Simulatoren in einer Programmiersprache realisieren und bleibt damit auf dem Niveau der ehemaligen konventionellen Simulationssysteme der zweiten und dritten Generation [3]. Daher ist es nötig, die VPSU-Simulationssoftware in Richtung von **parallelen Simulationssprachen** zu entwickeln, um die genannten Anforderungen zu erfüllen und den Simulationssystemen der fünften Generation [3] zu entsprechen. Dabei sind die Ideen der block-, gleichungs- und objektorientierten (BO, GO, OO) Sprachen zu beachten.

Das **BO-Lösungsprinzip** entspricht der MIMD-Parallelität und kann als eine virtuelle Zuordnung „Funktionsblock – Prozess“ betrachtet werden. Dem minimalkörnigen Prozess soll eine Funktionsblockkette zugeordnet werden, die Differenzialgleichungen ersten Ordnung löst. Diesen Ketten werden vom *Generator des BO-Lösers* mit Hilfe der entwickelten Algorithmen erstellt. So entsteht die von dem BO-Lösungsprinzip bedingte virtuelle Blockschaltung, die das virtuelle parallele Simulationsmodell der ersten Parallelitätsebene (PE) mit minimal körnigen Prozessen (MKP) „realisiert“. Das virtuelle Verbindungsnetzwerk (VNW) umfasst die folgenden zwei Ebenen: innere und äußere MKP-Verbindungen, die alle MKP entsprechend der Approximationschema verbinden. Der Devirtualisierungsvorgang wird von dem in der VPSU-Simulationssoftware (Abb.2) eingebauten BO-Devirtualisator durchgeführt. Dieser hat die Parallelisierungsebene PE-1 und den ZRS als

Randbedingungen. Die optimale Zuordnung mit dem ZRS wird unter Berücksichtigung der höheren Parallelitätsebenen PE-2 bis PE-4 und der Abbildung ihren virtuellen Verbindungsnetzwerken VNW-2 bis VNW-4 in dem ZRS-VNW realisiert. Diese Zuordnung benutzt einen Code-Generator, der ein lauffähiges MPI-Programm produziert und dabei GO- und OO-Ansätze verwendet.

9 Zusammenfassung und Ausblick

In dem Beitrag werden die theoretischen und praktischen simulationstechnischen Probleme der Modellierung und Simulation von komplexen dynamischen Systemen auf parallelen Rechensystemen als Forschungsaufgabe der „Parallele Simulationstechnik“ zusammengefasst und verschiedene Lösungen vorgeschlagen. Die entwickelten Konzepte der allgemeinen und problemorientierten verteilten parallelen Simulationsumgebungen (VPSU) sowie ihre Implementierungen erfordern eine neue Systemorganisation der parallelen Simulationswerkzeuge. Mit Hilfe der vorgestellten virtuellen Simulationsmodelle und Devirtualisierungsvorgänge erfolgt eine Systematisierung der einzelnen Entwicklungsschritte für parallele Simulatoren. Die geplante weitere Entwicklung der parallelen Simulationstechnik konzentriert sich auf parallele Simulationsalgorithmen und Simulationssprachen, objektorientierte Ansätze, anwendungsorientierte Umgebungen, Integration mit den GRID- und CLOUD-Infrastrukturen sowie auf industrielle und universitäre Anwendungen.

10 Literatur

- [1] Zeitz, M.: Simulationstechnik. Chemie-Ingenieur-Technik 59 (1987), 464 – 469
- [2] Anoprienko, A.J., Svjatnyj, V.A., Bräunl, T., Reuter, A., Zeitz, M.: Massiv parallele Simulationsumgebung für dynamische Systeme mit konzentrierten und verteilten Parametern. Tagungsband 9. ASIM-Symposium Simulationstechnik 1994, Vieweg 1994, 183-188.
- [3] Schmidt, B.: Simulationssysteme der 5. Generation. SiP 1994, 5-6.
- [4] Feldmann L.P.: Implementierung und Effizienzanalyse von parallelen blockartigen Simulationsalgorithmen für dynamische Systeme mit konzentrierten Parametern. Tagungsband 14. ASIM-Symposium Simulationstechnik 2000, SCS-Europe BVBA, Ghent/Belgium 2000, 241-246.
- [5] V. Svjatnyj, M. Resch, V. Kushnarenko, O. Shcherbakov: Dekomposition der verteilten parallelen Simulationsumgebung. Probleme der Modellierung und rechnergestützten Projektierung. DonNTU-Werk 1(10)-2(11), Donezk 2012, 227-234.
- [6] Modelica: A Unified Object-Oriented Language for Physical Systems Modeling. Language Specification. Version 2.0, 2002.
- [7] Feldmann, L.P., Dmitrieva, O.A., Gerber, S.: Abbildung der blockartigen Algorithmen auf Parallelrechnerarchitekturen. Tagungsband 15. ASIM-Symposium Simulationstechnik 2002, SCS-Europe BVBA, Ghent/Belgium 2002, 359-364.
- [8] Dmitrieva O.A. Parallele Differenzverfahren fürs Lösen des Cauchy-Problems. Donezk, DonNTU, 2011 (ukrainisch).
- [9] Resch, M., Kushnarenko, V., Svjatnyj, V.: Zur Entwicklung der parallelen Simulatoren für dynamische Netzobjekte mit verteilten Parametern. Konferenzband „Simulation-2012“, Kyiv, 2012, 26-34.
- [10] Adinetz, A., Kraus, J., Pleiter, D.: NVIDIA Application Lab at Jülich, inSiDE 11(2013), 97-99.
- [11] Dürr, F.: Improving the Efficiency of Cloud Infrastructures with Elastic Tandem Machines Proceedings 6th IEEE International Conference on Cloud Computing (Cloud 2013), Santa Clara/CA/USA, 2013.

Numerical Treatment of Dynamical Systems using Overdetermined Regularizations

Lena Scholz¹, Andreas Steinbrecher¹

¹Technische Universität Berlin, Institut für Mathematik
lscholz@math.tu-berlin.de, anst@math.tu-berlin.de

In the modeling and simulation of complex dynamical systems usually *differential-algebraic equations* (DAEs) arise. These differential-algebraic equations can be of higher index, i.e., they can contain so-called *hidden constraints* which lead to instabilities and order reduction in the numerical integration. Therefore, a regularization or remodeling of the model equations is required. In modern simulation environments often a structural analysis based on the sparsity pattern of the system is used to obtain the required information on the hidden constraints. These hidden constraints can be added to the system equations and then, usually, dynamical state variables are selected for which the occurring derivatives are replaced by newly introduced algebraic variables (so-called *dummy derivatives*). Such a regularization may be valid only locally since the state selection can vary with the dynamical behavior of the system. We present a new approach for the remodeling of dynamical systems that uses the information obtained from the structural analysis to construct a regularized overdetermined system formulation. This overdetermined system can then be solved using specially adapted numerical integrators without the need of state selection and the introduction of new algebraic variables.

1 Introduction

Nowadays, modeling, simulation and control of complex multi-physics dynamical systems is an important aspect in the development of technical innovations in many industrial fields. In general, the model equations that describe the dynamical process consist of differential equations in combination with algebraic constraints, i.e., we have to deal with *differential-algebraic equations* (DAEs). In particular, we will consider *quasi-linear DAEs* of the form

$$E(x, t)\dot{x} = k(x, t), \quad x(t_0) = x_0 \in \mathbb{R}^n, \quad (1)$$

on the domain $\mathbb{I} = [t_0, t_f]$, with $E \in \mathcal{C}(\mathbb{R}^n \times \mathbb{I}, \mathbb{R}^{n,n})$, $k \in \mathcal{C}(\mathbb{R}^n \times \mathbb{I}, \mathbb{R}^n)$, and $x : \mathbb{I} \rightarrow \mathbb{R}^n$. The leading matrix $E(x, t)$ is usually assumed to be singular. A solution of the DAE (1) has to satisfy the algebraic constraints, but, in general, not all constraints are stated in an explicit way. If the DAE is of higher index there exist so-called *hidden constraints* that restrict solutions to lie onto a specific solution manifold. The initial value $x_0 \in \mathbb{R}^n$ has to be *consistent*, i.e., it has to sat-

isfy **all** (explicit and hidden) algebraic constraints at $t = t_0$ such that a solution of (1) exists. In the numerical treatment the existence of hidden constraints usually leads to instabilities, inconsistencies and possibly non-convergence of the numerical methods such that a *regularization* or *remodeling* of the model equations is required to guarantee stable and robust numerical computations, see also [1, 2, 3, 4].

In many modeling and simulation tools (e.g. DYMOLA, OPENMODELICA, MAPLESIM, SIMULATIONX) the current state of the art to deal with high index DAEs is to use some kind of structural analysis based on the sparsity pattern of the system [5, 6]. Here, generic structural information is used to identify the constraints, to determine the index of the system, and to compute a regularized system model. These structural approaches are very efficient from a computational point of view, since fast algorithms based on graph theoretical concepts exist. Usually regularization methods based on dummy derivatives are used, where dynamical state variables are selected while the derivatives of the other state variables are replaced

by newly introduced variables, the so-called *dummy derivatives* [6, 7]. Unfortunately, such a regularization may only be valid locally in a neighborhood of the current state such that the validity of the regularization has to be checked and recomputed frequently. This leads to interruptions in the integration process, a required recomputation of a regularization and the restart of the integration process.

We will present a new regularization approach for the remodeling of dynamical systems that uses structural information to construct an overdetermined system formulation. This overdetermined regularization can then be solved using specially adapted numerical integrators without the need of state selections.

2 Structural Analysis of DAEs

The information on the hidden constraints that is required for regularizing the DAE (1) can often be obtained from a structural analysis of the system. For ease of representation we write the DAE (1) as

$$F(t, x, \dot{x}) = 0, \quad (2)$$

with $F(t, x, \dot{x}) := E(x, t)\dot{x} - k(x, t)$, where $F \in \mathcal{C}(\mathbb{I} \times \mathbb{R}^n \times \mathbb{R}^n, \mathbb{R}^n)$, and we denote by F_i the components of the vector-valued function F and by x_j the components of the vector-valued function x . Then, a structural analysis using the *Signature Method* (Σ -method) [8] consists of the following steps.

1. Build the *signature matrix* $\Sigma = [\sigma_{ij}]_{i,j=1,\dots,n}$

$$\sigma_{ij} := \begin{cases} \text{highest order } k \text{ of derivative } x_j^{(k)} \text{ in } F_i \\ -\infty \text{ if } x_j \text{ does not occur in } F_i \end{cases}$$

2. Find a *highest value transversal* (HVT) T of Σ

$$T = \{(1, j_1), (2, j_2), \dots, (n, j_n)\},$$

where (j_1, \dots, j_n) is a permutation of $(1, \dots, n)$ and the *value* $\text{Val}(T) = \sum_{(i,j) \in T} \sigma_{ij}$ is maximal.

3. Compute the *offset vectors* $c = [c_i]_{i=1,\dots,n}$ and $d = [d_j]_{j=1,\dots,n}$ with $c_i \geq 0$, $d_j \geq 0$ such that

$$\begin{aligned} d_j - c_i &\geq \sigma_{ij} & \text{for all } i, j = 1, \dots, n, \\ d_j - c_i &= \sigma_{ij} & \text{for all } (i, j) \in T. \end{aligned} \quad (3)$$

4. Form the Σ -*Jacobian* $\mathfrak{J} = [\mathfrak{J}_{ij}]_{i,j=1,\dots,n}$

$$\mathfrak{J}_{ij} := \begin{cases} \frac{\partial F_i}{\partial x_j^{(\sigma_{ij})}} & \text{if } d_j - c_i = \sigma_{ij}, \\ 0 & \text{otherwise.} \end{cases}$$

5. Build the *reduced derivative array* $\mathcal{F}(t, \mathcal{X}) = 0$ consisting of

$$\frac{d^\ell}{dt^\ell} F_i(t, x, \dot{x}) = 0$$

for all $\ell = 0, \dots, c_i$ and for all $i = 1, \dots, n$ with

$$\mathcal{X} = [x_1, \dot{x}_1, \dots, x_1^{(d_1)}, \dots, x_n, \dot{x}_n, \dots, x_n^{(d_n)}]^T.$$

6. Perform the *success check*: if the algebraic system $\mathcal{F}(t^*, \mathcal{X}^*) = 0$ has a solution $(t^*, \mathcal{X}^*) \in \mathbb{I} \times \mathbb{R}^{n + \sum_{i=1}^n d_i}$ and \mathfrak{J} is nonsingular at (t^*, \mathcal{X}^*) , then the Σ -method succeeds.

In the first part of the Σ -method an HVT of the signature matrix has to be computed. An HVT represents a perfect maximum matching between the variables and equations of the system and can efficiently be computed by solving a linear assignment problem, see [8]. By solving the corresponding dual problem the canonical offsets of the system can be computed. Note that usually there exists not only one uniquely determined HVT, and also the offset vectors c and d are not uniquely defined by the conditions (3). However, there exists a unique element-wise smallest solution of the dual problem, the so-called *canonical offsets*, that is independent of the chosen HVT. The offset vector c describes the number of times each equation of the DAE system has to be differentiated in order to be able to resolve the whole system for the highest occurring derivative of each variable (given by the offset vector d). By adding all of these differentiated equations to the original system of equation we can obtain a reduced derivative array that contains all information on the hidden constraints.

Another crucial step of the Σ -method is the so-called *success check* that consists of checking the regularity of the Σ -Jacobian (locally at a consistent point). If the Σ -Jacobian is singular the DAE is called *structurally singular*¹ and the structural analysis fails to reveal the

¹Note that the term *structurally singular* is also used with a different meaning in other areas of research.

actual properties of the system. In particular, the off-sets do not reveal all hidden constraints. We call \mathfrak{J} the Σ -Jacobian since it is in general not the analytical Jacobian, but defined by the offset vectors. If the Σ -method succeeds, then the offsets allow to determine the *structural index* of the DAE as

$$v_S := \max_i c_i + \begin{cases} 0 & \text{if all } d_j > 0, \\ 1 & \text{if some } d_j = 0. \end{cases}$$

The structural index gives an upper bound for the *differentiation index* (*d-index*) v_d of the system, i.e., $v_d \leq v_S$. Here, the d-index is defined as the minimal number of times that all or part of the equations in the system must be differentiated in order to obtain an explicit ordinary differential system for all unknowns. For details see also [1]. Equality of the two indices can in general only be shown for specially structured systems, as, e.g., systems in Hessenberg form, see [8].

Example 1 We consider a simple pendulum of mass $m = 1$, length $\ell > 0$ under gravity \mathbf{g} , see also [8]. The system equations are given by

$$\begin{aligned} F_1(t, x, \dot{x}) &= \dot{p}_1 - q_1 &= 0, \\ F_2(t, x, \dot{x}) &= \dot{p}_2 - q_2 &= 0, \\ F_3(t, x, \dot{x}) &= \dot{q}_1 + 2p_1\lambda &= 0, \\ F_4(t, x, \dot{x}) &= \dot{q}_2 + 2p_2\lambda + \mathbf{g} &= 0, \\ F_5(t, x, \dot{x}) &= p_1^2 + p_2^2 - \ell^2 &= 0, \end{aligned} \quad (4)$$

with $x = [p_1 \ p_2 \ q_1 \ q_2 \ \lambda]^T$. The signature matrix for this system is given by

$$\Sigma = \begin{bmatrix} \boxed{1} & - & \boxed{0} & - & - \\ - & \boxed{1} & - & \boxed{0} & - \\ 0 & - & \boxed{1} & - & \boxed{0} \\ - & 0 & - & \boxed{1} & \boxed{0} \\ \boxed{0} & \boxed{0} & - & - & - \end{bmatrix},$$

where the two possible HVTs are marked by the light and dark gray boxes. Here, the entry $-$ stands for $-\infty$. The canonical offset vectors are given by $c = [1, 1, 0, 0, 2]$ and $d = [2, 2, 1, 1, 0]$ independently of the chosen HVT and the corresponding Σ -Jacobian is given by

$$\mathfrak{J} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 2p_1 \\ 0 & 0 & 0 & 1 & 2p_2 \\ 2p_1 & 2p_2 & 0 & 0 & 0 \end{bmatrix}.$$

The reduced derivative array takes the form

$$\mathcal{F}(t, \mathcal{X}) = \begin{bmatrix} \dot{p}_1 - q_1 \\ \dot{p}_2 - q_2 \\ \dot{p}_2 - q_2 \\ \dot{q}_1 + 2p_1\lambda \\ \dot{q}_2 + 2p_2\lambda + \mathbf{g} \\ p_1^2 + p_2^2 - \ell^2 \\ 2p_1\dot{p}_1 + 2p_2\dot{p}_2 \\ 2p_1\ddot{p}_1 + 2\dot{p}_1^2 + 2p_2\ddot{p}_2 + 2\dot{p}_2^2 \end{bmatrix} = 0. \quad (5)$$

We can easily check that the Σ -Jacobian \mathfrak{J} is nonsingular at every consistent point satisfying (5) and, thus, the Σ -method succeeds with $v_S = \max_i c_i + 1 = 3 = v_d$.

Example 2 [9] Consider the DAE system

$$\begin{aligned} \dot{x}_1 &= x_3 + f_1 \\ \dot{x}_2 &= x_4 + f_2 \\ 0 &= x_2 + x_3 + x_4 + f_3 \\ 0 &= -x_1 + x_3 + x_4 + f_4 \end{aligned} \quad (6)$$

with smooth inhomogeneity $f = [f_1 \ f_2 \ f_3 \ f_4]^T$. The DAE (6) is regular and of d-index $v_d = 3$. If we apply the Σ -method to system (6), we get the signature matrix

$$\Sigma = \begin{bmatrix} \boxed{1} & - & 0 & - \\ - & \boxed{1} & - & 0 \\ - & 0 & \boxed{0} & 0 \\ 0 & - & 0 & \boxed{0} \end{bmatrix}$$

with marked HVT on the diagonal and canonical offset vectors $c = [0, 0, 0, 0]$ and $d = [1, 1, 0, 0]$. The corresponding Σ -Jacobian is given by

$$\mathfrak{J} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & -1 \end{bmatrix}$$

and \mathfrak{J} is singular, i.e., the Σ -method fails.

Example 2 shows that for regular DAEs the structural analysis can fail. It has been shown in [9] that this is the case for coupled semi-explicit d-index 1 systems if the coupling results in redundancies or in an increase in the index. However, the Σ -method succeeds in many cases and for many important structures as e.g. for systems in Hessenberg form, see [8].

Remark 1 The structural approach has the great advantage that fast and efficient linear optimization algorithms based on graph theoretical concepts can be used and further structural information like a block lower triangular (BLT) form of the system can be extracted which is essential for efficient and fast computations. It has been shown in [8] that Pantelides Algorithm and the Signature Method are equivalent in the sense that if they both succeed (or converge) they result in the same structural index and the offset vector $c = [c_i]$ corresponds to the number of differentiations for each equation F_i as determined by Pantelides Algorithm. Note that Pantelides Algorithm will not converge in cases where the success check of the Signature Method fails.

3 Numerical Treatment using Overdetermined Formulations

Regularization approaches for high index DAEs like the *Dummy Derivatives Approach* [7] or *Index Reduction by Minimal Extension* [10] usually follow the common strategy of adding the hidden constraints to the system equation at first and, secondly, of the selection of certain differential components of the state vector that can then be replaced by new algebraic variables. Then, the newly obtained regular system formulation has a lower index than the original system and all constraints are explicitly given. Hereby, a crucial step is the so-called *state selection* that is required in order to introduce new algebraic variables (the so-called *dummy derivatives*) for the selected differential components of the state vector. This choice of states that are selected can change during the numerical integration (e.g., if the pendulum moves from the vertical to the horizontal position). Thus, if the state selection is performed outside the numerical integrator this often is computational inefficient. In the following, we will present a regularization of quasi-linear DAEs (1) that are of higher index, i.e., that contain hidden constraints. This regularization is based on an overdetermined system formulation in order to overcome the difficulties in the numerical simulation.

If the structural analysis presented in Section 2 succeeds for a given system (2) at a consistent point, the canonical offset vector c contains the required information about the hidden constraints in the system (i.e.,

which equations have to be differentiated and how many times in order to be able to extract all hidden constraints). If the success check of the Σ -method fails, we can use the algebraic procedure proposed in [4, 11] to determine the hidden constraints of a quasi-linear DAE (1). For structurally singular systems in semi-explicit form arising in coupled systems of DAEs a combined structural-algebraic approach can be applied in cases where the success check fails, see [9]. The information provided by the HVT and the offset vectors can also be used to introduce new algebraic variables for selected differential variables yielding an extended square regularized system, for details see also [9].

In the following, we will assume that the DAE system (1) is uniquely solvable and that $\text{rank}(E(x, t)) = \text{const.}$ for all consistent $(x, t) \in \mathbb{R}^n \times \mathbb{I}$. Furthermore, we denote the complete set of constraint equations (i.e., all explicit and hidden constraints) of the DAE (1) by

$$0 = h(x, t), \quad (7)$$

where $h : \mathbb{R}^n \times \mathbb{I} \rightarrow \mathbb{R}^{n_c}$ and we assume that $\text{rank}\left(\frac{\partial h}{\partial x}(x, t)\right) = \text{const.}$ for all consistent $(x, t) \in \mathbb{R}^n \times \mathbb{I}$. Note that in general these assumptions can also be relaxed, see [4].

Example 3 For the simple pendulum the hidden constraints can be derived from the reduced derivative array (5). Thus, the complete set of constraints (7) consists of

$$\begin{bmatrix} p_1^2 + p_2^2 - \ell^2 \\ 2p_1q_1 + 2p_2q_2 \\ -4p_1^2\lambda + 2q_1^2 - 4p_2^2\lambda - 2p_2g + 2q_2^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (8)$$

Adding all the (hidden) constraints (7) to the quasi-linear DAE (1) leads to an overdetermined system

$$E(x, t)\dot{x} = k(x, t), \quad (9a)$$

$$0 = h(x, t). \quad (9b)$$

This overdetermined formulation (9) is equivalent to the original DAE (1) in the sense that both systems have the same solution set. Note that the unknowns x are unchanged, i.e., a transformation of the state variables is not necessary and the number of unknowns is not increased (in contrast to the dummy derivative approach). Furthermore, all constraints are stated in explicit form, i.e., no hidden constraints exist anymore.

In the following, we will use the overdetermined formulation (9) for the numerical solution of the DAE (1). Using the overdetermined formulation (9) has the great advantage that it is not necessary to apply analytical manipulations for the determination of a square and uniquely solvable system of DAEs (provided that consistent initial values are given).

For the numerical integration a discretization of the overdetermined system (9) is required. In order to avoid technical difficulties in this presentation and to emphasize the key ideas we will restrict to a discretization using the implicit Euler method. The iterate x_k that approximates the solution $x(t_k)$ at the discrete time point t_k is then obtained from a nonlinear discrete system of the form

$$0 = \begin{bmatrix} E(x_k, t_k)(x_k - x_{k-1}) - \tau_k k(x_k, t_k) \\ -\tau_k h(x_k, t_k) \end{bmatrix}. \quad (10)$$

Here, τ_k denotes the step-size in the integration step and $t_k = t_{k-1} + \tau_k$ denotes the discrete time point. Note that due to discretization, in general, the nonlinear system (10) is no longer exactly solvable. It is only possible to find an approximation \tilde{x}_k which minimizes the residuals

$$\begin{bmatrix} r_D(\tilde{x}_k) \\ r_C(\tilde{x}_k) \end{bmatrix} = \begin{bmatrix} E(\tilde{x}_k, t_k)(\tilde{x}_k - x_{k-1}) - \tau_k k(\tilde{x}_k, t_k) \\ -\tau_k h(\tilde{x}_k, t_k) \end{bmatrix} \quad (11)$$

in a certain sense. In general, such an approximation will result in a residual $r_C(\tilde{x}_k) \neq 0$, which in turn leads to unfulfilled constraints, i.e., $0 \neq h(\tilde{x}_k, t_k)$, not even within machine precision. During the numerical integration this would lead to the typical difficulties for higher index DAEs, i.e., instabilities, convergence problems, inconsistencies, or the solution drifts away from the original solution manifold. In order to avoid these problems it is necessary to make sure that the constraints are always satisfied during the numerical integration. To achieve this the nonlinear system (11) is treated separately for the dynamical part given by r_D and for the constraint part given by r_C . In particular, the next iterate \tilde{x}_k is determined in such a way that it satisfies the constraint part exactly (or within a prescribed precision), i.e., $r_C(\tilde{x}_k) \approx 0$, while \tilde{x}_k yields a minimal residual in the differential part, i.e., $r_D(\tilde{x}_k)$ is minimized. For more details see [4].

This approach has been implemented in the software package QUALIDAES (QUasi LInear DAE Solver) that is suited for the direct numerical integration of regu-

larized overdetermined model equations. The integration scheme of the solver QUALIDAES is based on the 3-stage implicit Runge-Kutta method of type Radau IIa of order 5.

4 Numerical Examples

For modeling dynamical processes the modeling language MODELICA is a very useful tool that is increasingly more used in industrial application. MODELICA is an object-oriented modeling language that easily allows for an hierarchical network-based modeling of complex multi-physical systems [5]. There exists a large variety of modeling and simulation environments using MODELICA, e.g., OPENMODELICA, DYMOLA, MAPLESIM, or SIMULATIONX. These environments use different (often confidential) strategies for the numerical solution of the arising DAE systems. Usually, a structural analysis is used to reduce the index of the DAE system relying on its sparsity structure, e.g., there are various versions and extensions of *Pantelides Algorithm* [12] in combination with the *Dummy Derivative Approach* [7].

Unfortunately it is impossible to model and integrate the overdetermined formulation (9) within the common MODELICA frameworks. Therefore, the above described approach has been incorporated into a prototype MODELICA framework named MPSSim (Multi-Physics System Simulation). Here, the numerical solver QUALIDAES that has been adapted to the overdetermined regularization (9) has been integrated. In the current version the user has to provide the model equations already given in overdetermined regularized form (9) formulated as MODELICA model. Then, using the translator M2FOR [13] a FORTRAN source code is generated that can be used to solve the model equations with the solver QUALIDAES. The FORTRAN source code is automatically compiled and linked to the solver QUALIDAES. In Figure 1 the approach for the numerical treatment of models defined in MODELICA using MPSSim is illustrated. Note that within this framework, it is not necessary to determine a dynamic (state) selector, since this is achieved automatically within the separated treatment of (10) by its numerical solution, as described above.

Example 4 As first example we consider the simple

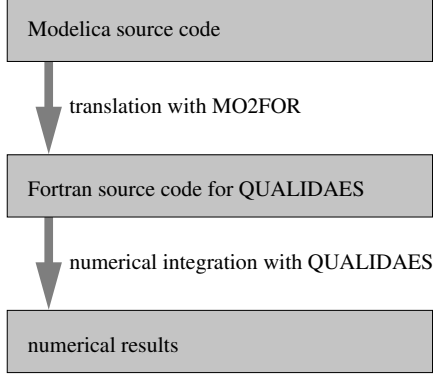


Figure 1: Scheme of MPSSim

pendulum given in Example 1. We use a gravitational constant of $g = 13.7503716373294544$ to ensure a time period of $T = 2$ for the motion of the pendulum and a mass of $m = 1$ as well as a length of $\ell = 1$. For the simulation with MPSSim we use the overdetermined system formulation given by (4) together with the hidden constraints (8).

At first we simulate the system for $t \in [0, 2]$ with given (fixed) consistent initial conditions

$$\begin{aligned} p_1(0) &= 1, & p_2(0) &= 0, & q_1(0) &= 0, \\ q_2(0) &= 0, & \lambda(0) &= 0, \end{aligned}$$

and a prescribed error tolerance of 10^{-6} (for both the absolute and relative error). The exact solution at $t = 2$ is known to be

$$\begin{aligned} p_1(2) &= 1, & p_2(2) &= 0, & q_1(2) &= 0, \\ q_2(2) &= 0, & \lambda(2) &= 0. \end{aligned}$$

The numerical integration yields an approximation

$$\tilde{x}(2) = \begin{bmatrix} \tilde{p}_1(2) \\ \tilde{p}_2(2) \\ \tilde{q}_1(2) \\ \tilde{q}_2(2) \\ \tilde{\lambda}(2) \end{bmatrix} = \begin{bmatrix} 0.999999999997 \\ 2.506954991624 \cdot 10^{-6} \\ -1.964524883411 \cdot 10^{-11} \\ 7.831924620040 \cdot 10^{-6} \\ -1.723575179268 \cdot 10^{-5} \end{bmatrix}$$

which yields the absolute error

$$\text{err}(2) = \|\tilde{x}(2) - x(2)\|_2 = 1.9097 \cdot 10^{-5}.$$

To show the promising performance of integrating a DAE system using MPSSim with an overdetermined

Simulation tool	$e(t_f)$	CPU time
MPSSim	$1.13 \cdot 10^{-6}$	0.148s
MAPLESIM (i)	$1.50 \cdot 10^{-5}$	1.604s
MAPLESIM (ii)	$0.00 \cdot 10^{-0}$	4.484s
DYMOLA	$2.14 \cdot 10^{-3}$	0.890s
OPENMODELICA	$-1.60 \cdot 10^{-3}$	2.938s

 Table 1: Simulation result of the pendulum equation with energy conservation for $t \in [0, 100]$

system formulation we also compare the simulation of the simple pendulum equations using MPSSim, MAPLESIM, DYMOLA and OPENMODELICA. In order to have a measurement for the error we include another equation in the system describing the total energy

$$e = \frac{1}{2}m(\dot{q}_1^2 + \dot{q}_2^2) + m g p_2$$

that should be preserved for all $t \in \mathbb{I}$ and every solution of the system (4). For the simulation tools MAPLESIM, DYMOLA and OPENMODELICA we use the original d-index 3 formulation (4) and the index reduction is performed within the tool using different index reduction strategies. The values $e(t_f)$ of the total energy at the final time point $t_f = 100$ together with the required CPU times needed for the integration are listed in Table 1. In DYMOLA a modified version of the multi-step solver DASSL is used. Here, quite a large number of state selections are required (alternating selecting the states p_1 and q_1 , or p_2 and q_2). In MAPLESIM we use once the provided solver CK45 suited for semi-stiff problems (results given in (i)), which is usually less accurate but faster, and once the Rosenbrock method suited for stiff systems (results given in (ii)), which yields a higher accuracy at the expense of more required CPU time. Furthermore, the corresponding results obtained for a long time simulation for $t \in [0, 1000]$ are listed in Table 2. Note that OPENMODELICA fails to integrate the system in this case. Comparing the obtained results one can see that the projection strategy onto the constraint manifold used within MAPLESIM yields an accurate numerical solution, while the numerical results obtained with DYMOLA are less accurate. The numerical results obtained with MPSSim are accurate within the range of the prescribed error tolerance at low compu-

Simulation tool	$e(t_f)$	CPU time
MPSSim	$1.89 \cdot 10^{-5}$	1.364s
MAPLESIM (i)	$1.49 \cdot 10^{-4}$	15.645s
MAPLESIM (ii)	$5.00 \cdot 10^{-6}$	45.358s
DYMOLA	$2.14 \cdot 10^{-2}$	8.830s
OPENMODELICA	—	—

Table 2: Simulation result of the pendulum equation with energy conservation for $t \in [0, 1000]$

tational costs. However, note that using MPSSim only the costs for the numerical integration of the overdetermined system are measured, while the CPU times of the other tools also contains the costs for index reduction, state selection, projection, and further transformations.

Example 5 As second example we consider the DAE system given in Example 2 and compare the behavior of the simulation environments OPENMODELICA, DYMOLA and MAPLESIM. As simulation parameters we use the stopping time $t_f = 1.0$ and we prescribe the consistent initial values $x_1(0) = -2$ and $x_2(0) = 3$. The initial values for $x_3(0)$ and $x_4(0)$ are computed by the solver. Note that using DYMOLA only two initial conditions can be fixed. The inhomogeneities in equation (6) are given by the functions $f_1(t) = \frac{1}{100}e^t$, $f_2(t) = \frac{1}{1000}e^{-2t}$, $f_3(t) = \sin(3t)$, and $f_4(t) = \cos(t)$. Furthermore, we use the default DAE solvers of each environment, i.e., the BDF-solver DASSL in OPENMODELICA and DYMOLA, and the Cash-Karp 4(5) Runge-Kutta method in MAPLESIM. Table 3 summarizes the behavior of the different simulation environments.

Environment	Results
OPENMODELICA	simulation fails ¹
DYMOLA	simulation fails ²
MAPLESIM	simulation succeeds

Table 3: Different simulation environments for Example 2

¹Error message: "Error solving linear system of equations. System is singular."

²Error message: "Scalar system is always singular"

Simulation tool	err(T)
OPENMODELICA	7.3772
DYMOLA	—
MAPLESIM	$7.5364 \cdot 10^{-7}$
MPSSim	$1.3494 \cdot 10^{-9}$

Table 4: Absolute error at $t_f = 1.0$ of the different simulation environments

The simulation environments DYMOLA and OPENMODELICA are not able to simulate the system due to the failure of the underlying structural analysis. However, MAPLESIM is able to simulate the system of Example 2. Thus, even for very simple problems simulation environments relying on a structural analysis can fail to simulate the DAE system. For a more detailed comparison, see [9].

For the numerical solution with MPSSim we add the hidden constraints given by

$$\begin{aligned} 0 &= x_3 + x_4 + f_1 + f_2 + \dot{f}_3 - \dot{f}_4, \\ 0 &= -x_4 - f_2 + \dot{f}_1 + \dot{f}_2 - \dot{f}_3 + \ddot{f}_3 - \ddot{f}_4, \end{aligned}$$

to the model equations (6). The exact solution of (6) is given by

$$\begin{aligned} x_1 &= -f_1 - f_2 + f_4 - \dot{f}_3 + \dot{f}_4, \\ x_2 &= f_1 + f_2 - f_3 + \dot{f}_3 - \dot{f}_4, \\ x_3 &= -f_1 - \dot{f}_1 - \dot{f}_2 + \dot{f}_4 - \ddot{f}_3 + \ddot{f}_4, \\ x_4 &= -f_2 + \dot{f}_1 + \dot{f}_2 - \dot{f}_3 + \ddot{f}_3 - \ddot{f}_4, \end{aligned}$$

allowing the determination of the absolute error

$$err(t_f) = \|\tilde{x}(t_f) - x(t_f)\|_2.$$

The absolute errors for the simulation results at the final time point $t_f = 1.0$ are given in Table 4. We see that the solver OPENMODELICA computes clearly wrong results while throwing an error message. The solver DYMOLA does not provide any results in this case. The results obtained with MPSSim are quite accurate. In all simulations we have used an (absolute and relative) error tolerances of 10^{-7} .

5 Conclusion

The great advantages of an overdetermined regularization and of the direct discretization of the overdetermined formulation is the fact that it is not necessary to determine a dynamic (state) selector in advance, since this is achieved automatically within the separated treatment of (10) by its numerical solution. Thus, the problem of state selection can be moved within the numerical integrator and can therefore be performed more efficiently during the run-time of the simulation. Furthermore, the number of unknowns in the DAE is not increased. A further advantage of an overdetermined regularization is the possibility to add solution invariants, e.g., mass, impulse or energy conservation laws, to the constraints, which often stabilizes numerical integration.

Acknowledgements

This work has been supported by the European Research Council through Advanced Grant *MODSIM-CONMP*.

References

- [1] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential Algebraic Equations*, volume 14 of *Classics in Applied Mathematics*. SIAM, Philadelphia, PA, 1996.
- [2] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems*. Springer-Verlag, Berlin, Germany, 2nd edition, 1996.
- [3] P. Kunkel and V. Mehrmann. *Differential-Algebraic Equations. Analysis and Numerical Solution*. EMS Publishing House, Zürich, Switzerland, 2006.
- [4] A. Steinbrecher. *Numerical Solution of Quasi-Linear Differential-Algebraic Equations and Industrial Simulation of Multibody Systems*. PhD thesis, Technische Universität Berlin, Germany, 2006.
- [5] P. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*. Wiley-IEEE Press, 2014.
- [6] S. Mattsson and H. Olsson and H. Elmqvist. *Dynamic Selection of States in Dymola*. Proceedings of the Modelica Workshop 2000, pp. 61-67, 2000.
- [7] S. Mattsson and G. Söderlind. Index reduction in differential-algebraic equations using dummy derivatives. *SIAM Journal on Scientific and Statistical Computing*, 14:677–692, 1993.
- [8] J. Pryce. A simple structural analysis method for DAEs. *BIT Numerical Mathematics*, 41:364–394, 2001.
- [9] L. Scholz and A. Steinbrecher. A combined structural-algebraic approach for the regularization of coupled systems of DAEs. Preprint 30-2013, Institut für Mathematik, TU Berlin, 2013.
- [10] P. Kunkel and V. Mehrmann. Index reduction for differential-algebraic equations by minimal extension. *Zeitschrift für Angewandte Mathematik und Mechanik*, 84(9):579–597, 2004.
- [11] A. Steinbrecher. Analysis of quasi-linear differential-algebraic equations. Preprint 11-2006, Institut für Mathematik, TU Berlin, 2006.
- [12] C.C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM Journal on Scientific and Statistical Computing*, 9:213–231, 1988.
- [13] R. Altmeyer and A. Steinbrecher. *Regularization and numerical simulation of dynamical systems modeled with Modelica*. Preprint 29-2013, Institut für Mathematik, TU Berlin, 2013.

Numerische Simulation des runden Freistrahls unter Anwendung des SST-Turbulenzmodells

Ansgar Ringleb¹, Wolfgang Schlüter¹, Günter Wozniak²

¹Kompetenzzentrum Strömungssimulation, Hochschule Ansbach

²Professur Strömungsmechanik, Technische Universität Chemnitz

ansgar.ringleb@hs-ansbach.de

Freistrahls-Strömungen bilden die Grundlage einer Vielzahl technischer Strömungen, bspw. der Flamm-Strahl eines Mündungsmischbrenners in einem Schmelzofen, der Einspritzvorgang einer Kraftstoffdüse in einem Otto-Motor oder auch die Ausströmung aus einem Rohrbündelwärmetauscher. Deren Berechnung und Verbesserung der Methoden steht stets im Fokus der Industrie und ist häufig auch Gegenstand der Wissenschaft und Forschung. In der vorliegenden Untersuchung wird die numerische Simulation des runden Freistrahls betrachtet. Im Mittelpunkt steht dabei die Modellierung des runden Freistrahls mittels eines 2-dimensionalen axialsymmetrischen Modells. Dazu wird die Anwendung der Reynolds-averaged Navier-Stokes Gleichungen (RANS) und des Shear-Stress-Transport Turbulenzmodells (SST) untersucht. Dabei zeigt sich, dass sich das Strömungsprofil am Strahlrand besonders sensitiv gegenüber der Breite des Rechengebietes verhält, und der Einsatz eines lokal verfeinerten Gitters entscheidende Vorteile bzgl. der Ergebnisgenauigkeit und der notwendigen Rechenleistung birgt. Zur Vorhersage experimenteller Ergebnisse eignet sich grundsätzlich die Anwendung des SST-Turbulenzmodells. Allerdings wird deutlich, dass die Vorhersagegenauigkeit des Strahlverhaltens durch Anpassung der Modell-Koeffizienten deutlich gesteigert werden kann. Dazu wird die Anpassung des Dissipations- und der Diffusionskoeffizienten verglichen. Beide Modifikationen bewirken eine deutliche Verbesserung der Simulation sowohl zur Vorhersage der Strahlgeschwindigkeit als auch zur Vorhersage der turbulenten kinetischen Energie.

1 Einleitung

Die Untersuchung grundlegender Strömungsprobleme wie bspw. des Freistrahls mittels numerischer Simulation unter Einsatz der Reynolds-averaged Navier-Stokes Gleichungen (RANS) ist aktuell noch immer Gegenstand wissenschaftlicher Untersuchungen [1].

Gegenstand der vorliegenden Untersuchung ist ein runder, achssymmetrischer Freistrahls, welcher eine freie Scherströmung darstellt. Dabei tritt aus einer Düse mit dem Durchmesser d ein Fluidstrom mit der mittleren Geschwindigkeit U_f aus und strömt in das Fluid der ruhenden Umgebung, Abbildung 1. Im Bereich des äußeren Randes des Strahls bildet sich dabei ein Schergradient $\partial u / \partial r$ aus. Dies ruft ein Aufrollen von Wirbeln hervor (engl.: vortex roll up), welche mit zunehmender Lauflänge x zur Turbulenz führen und weiter stromab wieder zerfallen.

Der Freistrahls wird dabei in 3 wesentliche Bereiche unterteilt: Nach dem Düsenaustritt beginnt der Scherschicht-Bereich ($x/d < 5$), welcher in den Umwand-

lungsbereich Scherschicht-Strahl übergeht ($5 < x/d < 25$). Weiter stromab herrscht der ungestörte Strahlbereich ($x/d > 25$).

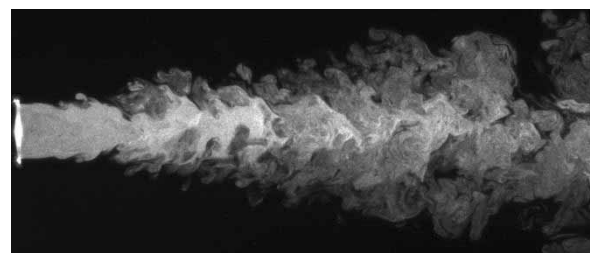


Abbildung 1: Fotografie des runden Freistrahls

Häufig interessieren dabei zeitgemittelte Strömungsgrößen, bspw. die Entwicklung der Strahlgeschwindigkeit entlang der Strahlachse $U_0(x)$ (engl.: center line velocity decay, CLVD) oder das Profil der Strömungsgeschwindigkeit in Achsrichtung quer zur Strahlachse $u(r)$ (engl.: axial mean velocity profile, AMVP).

Zur Untersuchung instationärer Strömungen oder turbulenter Wirbelstrukturen, z.B. bei Mischungsvor-

gängen, ist u.a. auch der Einsatz der Large-Eddy-Simulation (LES) Stand der Technik [2]. Diese Methode bietet i.d.R. erhebliche Vorteile bei der korrekten Berechnung des Strömungsgebietes, insbesondere wenn anisotrope Turbulenz vorherrschend ist.

Der wesentliche Vorteil der Anwendung eines RANS-Ansatzes gegenüber einer Modellierung mittels LES besteht dabei in dem erheblich geringeren Rechenaufwand für ein RANS-Modell, welcher einen Faktor von 10^6 betragen kann [3]. Daher ist die Weiterentwicklung in der Anwendung RANS-basierter Turbulenzmodelle eine für Forschung und Wirtschaft interessante Fragestellung.

In der vorliegenden Untersuchung ist der Bereich des ungestörten Strahls von Bedeutung. Zur Simulation werden zwei Strömungsgeschwindigkeiten im Düsenquerschnitt U_j gewählt, sodass sich eine auf den Düsendurchmesser d bezogene Re-Zahl von $Re_d = 10^5$ sowie $Re_d = 11.000$ ergeben. Die ebenfalls auf den Düsenquerschnitt bezogene Mach-Zahl beträgt so $Ma_d = 0,18$ sowie $Ma_d = 0,02$. Somit kann die Strömung als inkompressibel behandelt werden.

2 Modellierung und Simulation

2.1 Rechengebiet

Der runde Freistrahel oder auch achssymmetrische Freistrahel weist geometrisch und für zeitgemittelte Strömungsgrößen (aufgrund der Turbulenz) eine Symmetrie bzgl. der Rohr- bzw. der Strahlachse auf. Daher kann für die vorliegenden Untersuchungen ein achssymmetrisches 2D-Modell des Freistrahls angewendet werden, Abbildung 2.

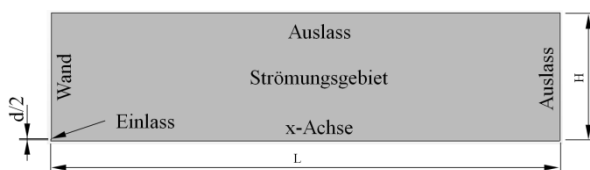


Abbildung 2: Skizze des 2D-Strömungsgebietes

2.2 Vernetzung

Zur räumlichen Diskretisierung des Rechengebietes wird ein hybrides kartesisches Gitter mit hexaedrischer Vernetzung in Wandnähe und uniformer kartesischer Vernetzung im Strömungsraum verwendet [4]. Zur Untersuchung der Netzsensitivität der Simulation werden 3 Gitter mit verschiedenen Gitterweiten herangezogen, Tabelle 1.

Tabelle 1: Netzgröße bei äquidistanter Vernetzung

Netz	Gitterweite	Zellanzahl	relativ
coarse	0,05 m	2.442	6%
normal	0,025 m	11.097	27%
fine	0,0125 m	41.135	100%

Weiterführend wird ein hybrides kartesisches Gitter mit lokaler Verfeinerung angewendet, Abbildung 3. Dabei wird im Bereich des Düsenaustritts und der Anlaufstrecke der Scherschicht ($x < 5d$) eine um den Faktor 16 kleinere Zellgröße als am äußeren Rand des Fernfeldes generiert.

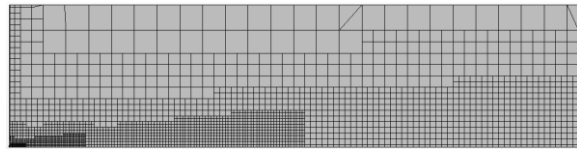


Abbildung 3: Vernetzung des Rechengebietes mittels lokaler Verfeinerung (Netz: „coarse“)

Auch für das lokal verfeinerte Gitter werden zur Untersuchung der Netzsensitivität Gitter mit verschiedener Auflösung erstellt, Tabelle 2.

Tabelle 2: Netzgröße bei lokal verfeinerter Vernetzung

Netz	Gitterweite Scherschicht	Zellanzahl	relativ
very coarse	25 mm	2673	3%
coarse	12,5 mm	7.611	9%
normal	6,25 mm	24.830	28%
fine	3,13 mm	88.300	100%

2.3 Solver

Der Freistrahel stellt eine Grundströmung (kanonische Strömung) mit freier Scherschicht dar. Im selbstähnlichen Bereich ($x/d > 25$) wird der Freistrahel durch den turbulenten Zerfall der großskaligen Wirbelstrukturen geprägt. Diese turbulenten Strömungsvorgänge sind durch ihr zeitlich instationäres Verhalten charakterisiert. Durch Anwendung des RANS-Ansatzes werden die turbulenten Effekte aufgrund der Zeit-

Mittelung eliminiert, sodass die iterative Lösung der Simulation stationär durchgeführt werden kann. Zur räumlichen Diskretisierung der RANS-Gleichungen wird die Finite-Volumen-Methode genutzt [5].

2.4 Turbulenzmodell

Zur Modellierung der Turbulenz wird das Shear-Stress-Transport (SST) Modell von Menter angewandt [6]. Das SST-Modell ist ein sog. 2-Gleichungsmodell, welches 2 gekoppelte partielle Differentialgleichungen zur Schließung der Impulsgleichungen bereitstellt. Dies sind Transportgleichungen für die turbulente kinetische Energie k (1) und die turbulente Wirbelfrequenz ω (2).

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho U_i k)}{\partial x_i} = \tau_{ij} \frac{\partial u_i}{\partial x_j} - \beta^* \rho k \omega + \frac{\partial}{\partial x_i} \left[(\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_i} \right] \quad (1)$$

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho U_i \omega)}{\partial x_i} = \frac{\gamma}{\nu_t} \tau_{ij} \frac{\partial u_i}{\partial x_j} - \beta \rho \omega^2 + \frac{\partial}{\partial x_i} \left[(\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_i} \right] + 2\rho(1 - F_1) \frac{\sigma_{\omega 2}}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \quad (2)$$

Dabei wird die turbulente Viskosität μ_t mittels des Ansatzes (3) und der Reynoldsspannungstensor τ_{ij} mittels des Ansatzes (4) berechnet.

$$\mu_t = \frac{\rho k}{\omega} \quad (3)$$

$$\rho \tau_{ij} = 2\mu_t S_{ij} \quad (4)$$

In der vorliegenden Untersuchung werden die Koeffizienten des Dissipationsterms β^* und des Diffusionsterms σ_{k2} sowie $\sigma_{\omega 2}$ angepasst. Die Koeffizientenwerte des Standardmodells sind nachfolgend aufgeführt, (5) bis (7).

$$\beta^* = 0,09 \quad (5)$$

$$\sigma_{k2} = 1,0 \quad (6)$$

$$\sigma_{\omega 2} = 0,856 \quad (7)$$

3 Ergebnisse

3.1 Einfluss des Rechengebietes

Im ersten Schritt wird der Einfluss der Abmessungen des Rechengebietes untersucht. Dazu wird ein Gitter mit einer Länge $L = 100 d$ mit einem Gitter der Länge $L = 200 d$ verglichen, wobei kein signifikanter Unterschied bzgl. der CLVD festgestellt werden kann. Zum Vergleich wurden die experimentellen Ergebnisse für Re-Zahlen von $Re_d = 10^5$ aus [7] herangezogen.

Der Vergleich mit den Messergebnissen der Hitzdrahtanemometrie (engl.: Constant Temperature Anemometrie, CTA) von Hussein [7] bzgl. der Rechengebiet-Breite H anhand der AMVP zeigt, dass ein Gitter mit der Breite $H = 10 d$ am Strahlrand unzureichende, ein Gitter der Breite $H = 50 d$ aber eine sehr gute Annäherung an das Experiment liefert, Abbildung 4.

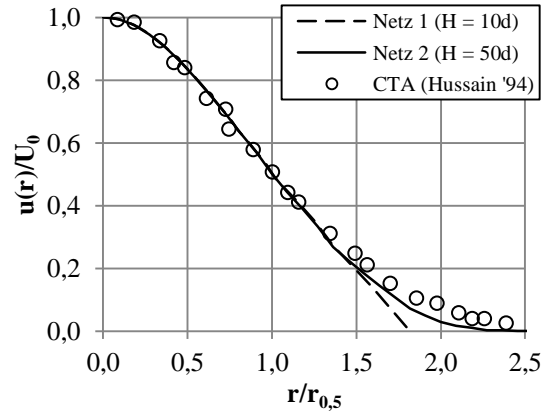


Abbildung 4. Einfluss der Rechengebiet-Breite auf das AMVP

3.2 Einfluss der Vernetzung

Die Analyse des Netzeinflusses anhand der CLVD ergibt, dass die Gitter „coarse“ und „normal“ besonders im Scherbereich des Strahls bei uniformer Vernetzung (äquidistante Gitterweite) eine unzureichende Aussage ermöglichen. Erst das Gitter der Auflösung „fine“ zeigt für den gesamten betrachteten Strahlbereich ($x/d < 120$) akzeptable Ergebnisse und eine nahezu linear steigende CLVD.

Durch Verwendung lokal verfeinerter Gitter konnte die Berechnung der CLVD durch niedrigere Steigung und lineareren Verlauf deutlich verbessert werden, Abbildung 5. Trotz der erzielten Verbesserung durch Verwendung des lokal verfeinerten Gitters („coarse“) wird durch die Simulation eine stärkere Steigung der CLVD als im Experiment berechnet, was einer größe-

ren Abnahme der Strömungsgeschwindigkeit in Achsrichtung $U_0(x)$ entspricht.

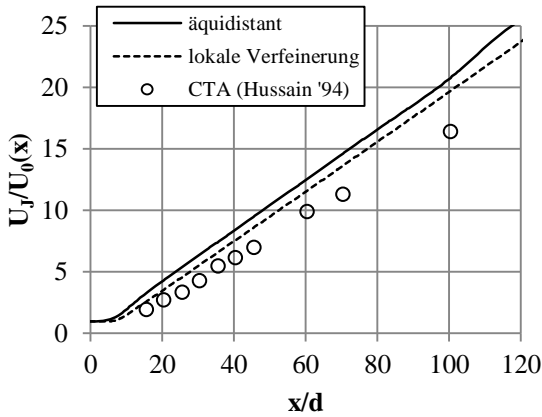


Abbildung 5. Vergleich der CLVD durch Berechnung mittels äquidistanten Gitters („fine“) und lokal verfeinerten Gitters („coarse“)

Zugleich wird durch Verwendung des lokal verfeinerten Gitters („coarse“, Tabelle 2) die Zellanzahl um den Faktor 5 gegenüber der uniformen Vernetzung („fine“, Tabelle 1) reduziert. Außerdem benötigt das lokal verfeinerte Gitter nur 2.734 Iterationen im Gegensatz zu 7.775 Iterationen des uniformen Gitters, um bei gleichen Solvereinstellungen ein Konvergenzkriterium (RMS) der Residuen von kleiner 10^{-4} zu erreichen. Somit kann durch Einsatz eines lokal verfeinerten Gitters der Rechenaufwand insgesamt um den Faktor 15 reduziert werden.

Die CLVD kann im selbstähnlichen Bereich des Freistrahls ($x/d > 25$) mittels der Beziehung (8) approximiert werden.

$$\frac{U_f}{U_0(x)} = \frac{1}{c} \cdot \frac{x}{d} \quad (8)$$

Experimentell wurde mittels CTA die reziproke Steigung der CLVD zu $C = 5,8$ und der virtuelle Strahl-Nullpunkt zu $x_0/d = 4,0$ bestimmt [7]. Durch Verwendung weiterer lokal verfeinerter Gitter mit höherer Auflösung kann die CLVD im Umwandlungsbereich Scherschicht – Strahl ($5 < x/d < 25$) erheblich verbessert werden. So wird mittels des Gitters „normal“ der virtuelle Strahl-Nullpunkt mit $x_0/d = 4,48$ gut wiedergegeben, Tabelle 3.

Die infolge der Simulation zu stark berechnete Steigung der CLVD kann mittels feinerer Vernetzung nicht behoben werden. Eher im Gegenteil zeigt sich, dass die Steigung der CLVD mit feinerer Gitterauflösung weiter zunimmt (entspricht Abnahme von C),

und zugleich der Strahl-Nullpunkt weiter stromab verschoben wird.

Tabelle 3: Einfluss der Vernetzung auf Strömungsparameter des Freistrahls

Netz	C	ΔC	x_0/d	Δx_0
very coarse	5,06	2,1%	1,22	-77%
coarse	5,00	0,9%	2,74	-48%
normal	4,97	0,1%	4,48	-15%
fine	4,96	-	5,28	-

Damit ist ersichtlich, dass die fehlerhafte Modellierung der CLVD durch Einsatz feinerer Gitter verstärkt wird. Deren Ursache in einer fehlerbehafteten Turbulenzmodellierung zu finden ist.

3.3 Einfluss der Turbulenzmodellierung

Durch das Standard-SST-Modell wird die turbulente kinetische Energie k im Bereich der Strahlmitte und -flanke bis hin zum Strahlrand zu stark modelliert, Abbildung 6.

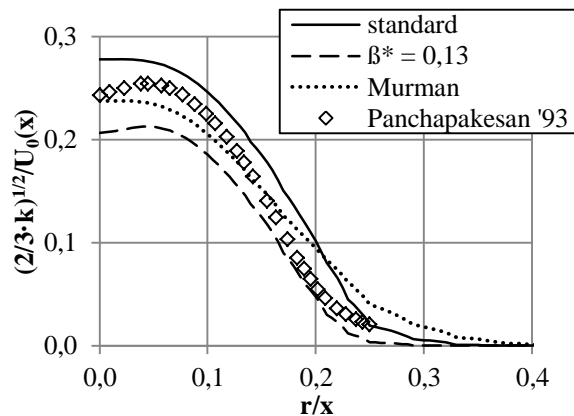


Abbildung 6. Einfluss der Anpassung der Modellkoeffizienten auf die turbulente kinetische Energie

Zur Behebung des o.g. Effektes wurde der Dissipationskoeffizient um 0,04 auf $\beta^* = 0,13$ angepasst, um durch eine erhöhte Dissipation die Modellierung der turbulenten kinetischen Energie besonders in der Strahlmitte zu reduzieren. Als weitere Möglichkeit zur Verbesserung der Turbulenzmodellierung ist die Anpassung der Diffusionskoeffizienten bekannt [8]. Darin wurde die Optimierung der Diffusionskoeffizienten zu $\sigma_{k2} = 2,0$ und $\sigma_{\omega2} = 1,5$ vorgeschlagen und Ergebnisse für Re-Zahlen von $Re_d = 10^4$ vorgestellt.

Zur Untersuchung der Dissipationskoeffizienten wurden Simulationen mit einer Re-Zahl von $Re_d = 11.000$

durchgeführt. Zum Vergleich wurden experimentelle Untersuchungen von Panchapakesan et al. mittels Hitzdrahtanemometrie (engl.: shuttle-mounted hot wire anemometrie, SHW) für eine Re-Zahl von $Re_d = 11.000$ herangezogen [9].

Durch Anpassung des Dissipationskoeffizienten kann die Modellierung der turbulenten kinetischen Energie über den gesamten Strahlbereich deutlich reduziert werden. Insbesondere in der Strahlmitte wird dadurch das – für den runden Freistrahls markante – lokale Maximum der turbulenten kinetischen Energie bei $r/x = 0,05$ (engl.: „off axis peak“) simuliert, vgl. Abbildung 6. Im Gegensatz dazu ruft die Anpassung der Diffusionskoeffizienten zwar ebenfalls eine Reduzierung der turbulenten kinetischen Energie hervor, allerdings wird dabei nicht der typische „off axis peak“ berechnet, und die turbulente kinetische Energie wird im Vergleich zum Experiment im Bereich der Strahlflanke zu stark simuliert.

Infolge dessen bewirkt die Anpassung des Dissipationskoeffizienten eine signifikante Verbesserung der CLVD gegenüber dem Standard-SST-Modell und damit eine gute Übereinstimmung zum Experiment, dies gilt besonders für den Bereich $x/d \geq 40$, Abbildung 7. Ebenso zeigt die Anpassung der Diffusionskoeffizienten eine erhebliche Verbesserung der CLVD, allerdings wird die Steigung im Bereich $x/d > 120$ etwas zu stark berechnet.

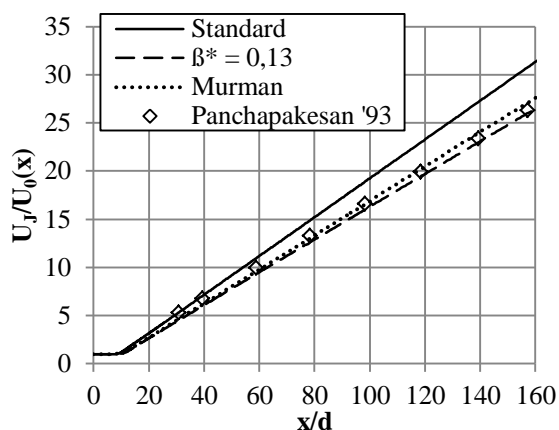


Abbildung 7. Einfluss der Anpassung der Modell-Koeffizienten auf die CLVD

3.4 Einfluss auf Konvergenzverhalten

Darüber hinaus bewirkt die Anpassung der Koeffizienten einen signifikanten Einfluss auf das Konvergenzverhalten der Simulationen. So kann durch die Erhöhung des Dissipationskoeffizienten die Rechen-

dauer um 10 % reduziert werden, Tabelle 4. Dem gegenüber bewirkte die Anpassung der Diffusionskoeffizienten eine Verschlechterung des Konvergenzverhaltens der Simulation und damit einhergehend einen Anstieg der Rechendauer um 35 %.

Tabelle 4: Einfluss der Anpassung der Koeffizienten auf das Konvergenzverhalten der Simulationen bei $Re_d = 11.000$

Simulation	Iterationen	%
Standard	8.081	100
$\beta^* = 0,13$	7.237	89,6
Murman	10.875	134,6

4 Zusammenfassung

Gegenstand der vorliegenden Untersuchung ist die Numerische Simulation des runden Freistrahls unter Anwendung eines 2D-Axialsymmetrischen Modells sowie der RANS-Gleichungen und des SST-Turbulenzmodells. Dabei zeigt sich, dass das Strömungsprofil am Strahlrand besonders sensitiv gegenüber der Breite des Rechengebietes reagiert, und der Einsatz lokal verfeinerter Gitter große Vorteile bzgl. der Ergebnissenauigkeit und der notwendigen Rechenleistung birgt.

Darüber hinaus erweist sich die Anwendung des Standard-SST-Turbulenzmodells zur Vorhersage der CLVD und des virtuellen Strahl-Nullpunktes nur als bedingt geeignet. Daher wird eine Anpassung der Modellkoeffizienten erforderlich, wozu die Anpassung des Dissipations- mit der Anpassung der Diffusionskoeffizienten verglichen ist. Die Anpassung des Dissipations- als auch der Diffusionskoeffizienten führt zu einer deutlichen Verbesserung bei der Berechnung der CLVD gegenüber dem Standard-SST-Modell.

Zudem bewirkt die Anpassung der Koeffizienten einen deutlichen Einfluss auf das Konvergenzverhalten der Simulation. So kann festgestellt werden, dass die Anpassung des Dissipationskoeffizienten gegenüber der Anpassung der Diffusionskoeffizienten eine Reduzierung der Rechenzeit von rund 45 % bewirkt.

5 Ausblick

Zukünftig ist eine weiterführende Untersuchung des Einflusses der Modellkoeffizienten auf die Vorhersa-

ge der Strahleigenschaften sinnvoll, insbesondere bzgl. der Produktion, Diffusion und Dissipation der turbulenten kinetischen Energie. Damit können weiterführende Aussagen zur Eignung und Anwendung des SST-Turbulenzmodells, bspw. bzgl. weiterer wichtiger Strahleigenschaften wie der AMVP, getroffen werden.

Auch eine differenziertere Betrachtung der Strahlbereiche kann sinnvoll sein, da die Wirkprinzipien technischer Anwendungen oftmals entweder auf dem mündungsnahen ($x/d < 25$) oder selbstähnlichen Strahlbereich ($x/d > 25$) beruhen.

6 References

- [1] Y. Reinhardt, L. Kleiser: „Simulation of round jets with nozzle-dependent inflow conditions using the k- ϵ turbulence model“, Proceedings in Applied Mathematics and Mechanics (PAMM), Vol. 14, Issue 1, 2014
- [2] I. Tkatchenko, N. Kornev, S. Jahnke, G. Steffen, E. Hassel: „Performance of LES and RANS Models for Simulation of Complex Flows in a Coaxial Jet Mixer“, Flow Turbulence Combustion, Vol. 78, S. 111-127, 2007
- [3] F.R.Menter, J. Schütze, M. Gritskevich: „Global vs. Zonal Approaches in Hybrid RANS-LES Turbulence Modelling“, Vortrag auf dem ERCOFTAC Technologietag Süddeutschland 2011, Stuttgart, Oktober 2011
- [4] J.F. Thompson, B.K. Soni, N.P. Weatherill: „Handbook of Grid Generation“, CRC Press LLC, Boca Raton (USA), 1999
- [5] H.K. Versteeg, W. Malalasekara: „An Introduction to Computational Fluid Dynamics The Finite Volume Method“, Pearson Education Ltd., Edinburgh Gate (England), 2007
- [6] F.R. Menter: „Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications“, AIAA Journal, Vol. 32, No. 8, August 1994
- [7] H.J. Hussein, S.P. Capp, W.K. George: „Velocity measurements in a high-Reynolds-number, momentum-conserving, axisymmetric, turbulent jet“, J. Fluid Mech., Vol. 258, S. 31-75, 1994
- [8] Murman, S.M., “Evaluating Modified Diffusion Coefficients for the SST Turbulence Model Using Benchmark Tests,” AIAA, S. 2011-3571, June 2011
- [9] N.R. Panchapakesan, J.L. Lumley: „Turbulence measurements in axisymmetric jets of air and helium. Part 1. Air jet“, J. Fluid Mech., Vol. 246, S. 197-223, 1993

Comparison of Regularisation Approaches for High Index DAEs

Carina Pöll¹, Irene Hafner², Bernhard Heinzl³, Felix Breiteneker¹

¹Vienna University of Technology, Institute for Analysis and Scientific Computing

²dwh GmbH, Simulation Services, Vienna

³Vienna University of Technology, Institute of Computer Aided Automation
carina.poell@tuwien.ac.at

The use of object-oriented modelling approaches for modelling physical or mechanical systems leads to differential-algebraic equations which cannot be directly transformed into ordinary differential equations, because they have a differential index greater than zero. The transformation into an ordinary differential equation is important for the numerical solving. This paper is intended to provide an overview of common regularisation approaches for differential-algebraic equations with high differential index. First there are some mathematical definitions and basic findings. After a short overview of the discussed methods, including a classification of the techniques in the areas differentiation, projection and transformation, the different methods are demonstrated using the commonly known example of a rotational pendulum described in Cartesian coordinates. With respect to this example a comparison of the numerical solutions of the used methods is possible.

1 Introduction

A component-based acausal model description for physical or mechanical systems, such as Modelica or MATLAB/Simscape, usually leads to differential-algebraic equations (DAEs) with a non-trivial differential index. Solving DAEs with a high index using methods designed for ordinary differential equations (ODEs) is generally very complex and therefore numerically extensive or may even be impossible. This problem leads to the so-called index reduction, in which the given DAE is reformulated as a DAE with lower index or an ODE. Due to the large differences (structure, properties, etc.) of DAEs a lot of different index reduction methods can be found in the literature, see [1] and [2]. The different techniques discussed in this paper can be classified into the following areas:

- methods using differentiation
- methods using projection
- methods using transformation of the state space

On the following pages these methods are discussed and a short example is given to illustrate their functionality.

2 Basics

In this section there are some basic definitions which are used later on. A differential-algebraic equation (DAE), see [1], is given by an implicit equation

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

with a function $F : I \times D_x \times D_{\dot{x}} \rightarrow \mathbb{R}^n$, where $I \subseteq \mathbb{R}$ is a real interval and $D_x, D_{\dot{x}} \subseteq \mathbb{R}^n$ are open sets, $n \in \mathbb{N}$ and $x : I \rightarrow \mathbb{R}^n$ is a differentiable function, where \dot{x} is the derivative of x with respect to t . According to the implicit function theorem F can be solved for \dot{x} if the matrix $\frac{\partial F}{\partial \dot{x}}$ is regular. The algebraic equations of the differential-algebraic equation system $F(t, x, \dot{x}) = 0$ are of the form

$$g(x) = 0, \quad (2)$$

where $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a function with $m < n$, and they are called constraints. A differential–algebraic equation has differential index $k \in \mathbb{N}_0$ (see [2]) if k is the minimal number of derivatives, so that an ODE can be extracted from the system

$$F(t, x, \dot{x}) = 0, \frac{dF(t, x, \dot{x})}{dt} = 0, \dots, \frac{d^k F(t, x, \dot{x})}{dt^k} = 0. \quad (3)$$

This generated ODE can (through algebraic transformations) be written in the form $\dot{x} = \varphi(t, x)$ with a function $\varphi: I \times D_x \rightarrow \mathbb{R}^n$.

3 Regularisation Approaches

The regularisation approaches are split into three different parts like in [2].

3.1 Regularisation by Differentiation

There are various approaches which are using differentiation.

3.1.1 Differentiation of the Constraint

The procedure of this approach is to differentiate the constraint $g(x) = 0$ and substitute the constraint by its derivative until the system has differential index 1. The problem of this approach is that due to the differentiation there is a loss of information. Therefore the necessary initial values for the integration are unknown and numerical "drift-off" occurs, i.e. the numerical solution departs from the exact solution.

3.1.2 Baumgarte–Method

This method can only be used for DAEs of index 3. The initial point of this approach is the index–1–formulation of the index–3–system. The constraint $\ddot{g}(x) = 0$ is substituted by a linear combination of g , \dot{g} and \ddot{g} of the form (see [3])

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

Because of the consideration of the original constraint there is no loss of information. α and β have to be

chosen, so that the differential equation is asymptotically stable. Therefore follows $\alpha > 0$. The problem of this approach is the exact choice of the constants α and β .

3.1.3 Pantelides–Algorithm

For each equation of the constraints the following procedure has to be used.

1. Each constraint has to be differentiated.
2. The differentiated constraint has to be added to the DAE. If there is an algebraic variable in the constraint, then the derivative of this variable is a so-called dummy derivative.
3. An integrator which has a connection to the constraint and the derivative of the constraint respectively is eliminated, i.e. for example \dot{x} is eliminated and instead of \dot{x} a new variable called dx is used.
4. Through differentiation of the constraint it can occur that a new variable is generated, i.e. for example through differentiation y (algebraic variable) becomes dy and there is an equation where y can be computed in the system.
5. Therefore this equation also has to be differentiated.
6. The proceeding of the points 4–5 continues until no new variables are created.

A disadvantage of this algorithm is that during the procedure a lot of variables and equations may be created and therefore the system of the resulting equations can be confusing.

3.2 Regularisation by Projection

A DAE with differential index $k > 1$ is given. If the numerical solution does not fulfill the constraint, the numerical solution is projected onto a manifold, which is given by the constraint $g(x) = 0$ and the $1^{\text{st}}, \dots, (k-2)^{\text{th}}$ derivatives with respect to t of the constraint. The solution manifold is

$$M = \left\{ x \in \mathbb{R}^n : g(x) = 0, \frac{d^i g(x)}{dt^i} = 0, \forall i \in I \right\}, \quad (5)$$

with $I = \{1, \dots, k-2\}$. The algebraic variables can be expressed by the $(k-1)^{th}$ derivative of the constraint and are inserted into the differential equations of the DAE. This procedure leads to a system of differential equations $\dot{y} = f(t, y)$ on the manifold M .

3.2.1 Standard Projection Method

One step $y_n \mapsto y_{n+1}$ of the Standard Projection Method is calculated in the following way, see [4]:

- $\hat{y}_{n+1} = \Phi_h(t_n, y_n)$ is calculated, where Φ_h is a numerical integrator applied to $\dot{y} = f(t, y)$ (for example a Runge–Kutta method).
- For getting $y_{n+1} \in M$, \hat{y}_{n+1} is projected orthogonally onto the manifold M .

In figure 1 the Standard Projection Method is represented.

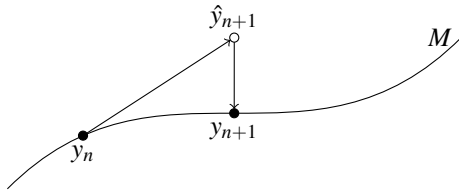


Figure 1: Illustration of the Standard Projection Method

This method is also called Orthogonal Projection Method.

3.2.2 Symmetric Projection Method

A one-step-method Φ_h is called symmetric, if $\Phi_h = \Phi_{-h}^{-1}$.

One step $y_n \mapsto y_{n+1}$ of the Symmetric Projection Method is calculated in the following way, see [5]:

- $\hat{y}_n = y_n + \frac{\partial g}{\partial y}^T(y_n)\mu$ with $g(y_n) = 0$.
- $\hat{y}_{n+1} = \Phi_h(\hat{y}_n)$ is calculated, where Φ_h is a symmetric one-step-method applied to $\dot{y} = f(y)$.
- $y_{n+1} = \hat{y}_{n+1} + \frac{\partial g}{\partial y}^T(y_{n+1})\mu$ with μ such that $g(y_{n+1}) = 0$.

It is important that the same μ is used in the step $y_n \rightarrow y_{n+1}$. In figure 2 the Symmetric Projection Method is shown.

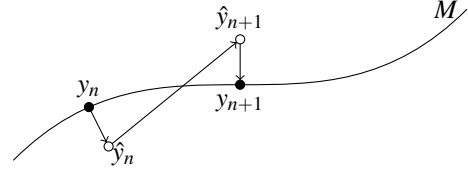


Figure 2: Illustration of the Symmetric Projection Method

3.3 Regularisation by Transformation

The DAE with differential index k is not solved on the whole state space, but on a manifold, see [4]. The manifold is implicitly given by the constraint $g(x) = 0$ and the $1^{st}, \dots, (k-2)^{th}$ derivatives with respect to t of the constraint. The manifold M is given by equation (5). The algebraic variables can be expressed by the $(k-1)^{th}$ derivative of the constraint and are inserted into the differential equations of the DAE. This procedure leads to a system of differential equations $\dot{y} = f(t, y)$ on the manifold M . This ODE on the manifold M is solved through the introduction of local coordinate transformations.

Let a local coordinate function $\psi: U \rightarrow \mathbb{R}^n$, $U \subset \mathbb{R}^{n-m}$ open, $\psi(U) \subset M$ be given on the m -dimensional manifold M . The transformation $y = \psi(z)$ transforms the differential equation $\dot{y} = f(t, y)$ into

$$\frac{\partial \psi(z)}{\partial z} \dot{z} = f(t, \psi(z)). \quad (6)$$

With the assumption $f(t, y) \in T_y M$, where $T_y M$ is the tangent space in a fixed point $y \in M$ and has dimension m , the differential equation (6) is equivalent to

$$\dot{z} = F(t, z). \quad (7)$$

One step $y_n \mapsto y_{n+1}$ using local coordinates is calculated in the following way:

- Local coordinates are chosen and z_n is calculated with $y_n = \psi(z_n)$.
- $\hat{z}_{n+1} = \Phi_h(t_n, z_n)$ is calculated with a numerical method Φ_h applied to (7).

- $y_{n+1} = \psi(\hat{z}_{n+1})$

The coordinates $y = \psi(z)$ can be changed in each step. The difficulty of this method is to find suitable coordinates.

4 Case Study

The circular motion of a pendulum in Cartesian coordinates is used as illustrative model. The equations (see [6]) of this model are given by

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{v}_x &= -Fx \\ \dot{v}_y &= g - Fy \\ x^2 + y^2 &= 1, \end{aligned} \quad (8)$$

where g is the gravitational acceleration on earth and F the Lagrange multiplier. The constraint is given by the equation $x^2 + y^2 = 1$. In figure 3 the motion is schematically represented.

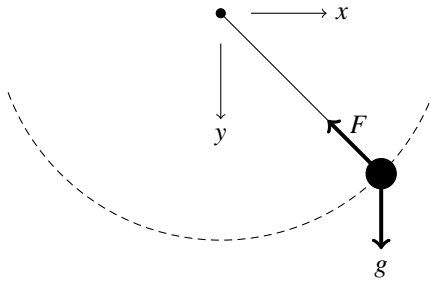


Figure 3: Illustration of the Circular Motion of a Pendulum

4.1 Analysis of the DAE

The constraint is used to determine the differential index of the DAE. The constraint and the first, second and third derivative with respect to t are given by

$$\begin{aligned} x^2 + y^2 - 1 &= 0 \\ xv_x + yv_y &= 0 \\ v_x^2 + v_y^2 - F(x^2 + y^2) + gy &= 0 \\ 4F(xv_x + yv_y) - 3gv_y + (x^2 + y^2)\dot{F} &= 0. \end{aligned} \quad (9)$$

From the third derivative with respect to t , \dot{F} can be expressed. Therefore the DAE has index 3. A remarkable fact is that F can be expressed from the second

derivative with respect to t . Thus F can be inserted in the differential equations. In table 1 the generated ODEs of the given DAE are shown.

index-0-system	index-1-system
$\dot{x} = v_x$	$\dot{x} = v_x$
$\dot{y} = v_y$	$\dot{y} = v_y$
$\dot{v}_x = -Fx$	$\dot{v}_x = -\frac{v_x^2 + v_y^2 + gy}{x^2 + y^2}x$
$\dot{v}_y = g - Fy$	$\dot{v}_y = g - \frac{v_x^2 + v_y^2 + gy}{x^2 + y^2}y$
$\dot{F} = \frac{-4F(xv_x + yv_y) + 3gv_y}{x^2 + y^2}$	

Table 1: Different ODEs for the Motion of the Pendulum

4.2 Regularisation by Differentiation

4.2.1 Differentiation of the Constraint

The constraint has to be differentiated twice which results in

$$v_x^2 + v_y^2 - F(x^2 + y^2) + gy = 0. \quad (10)$$

Therefore this approach leads to a DAE with index 1. This DAE is solved with the ode-solver ode15s of MATLAB. In figure 4 the result of this method is shown where the "drift-off" phenomenon is visible, whereby the simulation is calculated till 100 seconds.

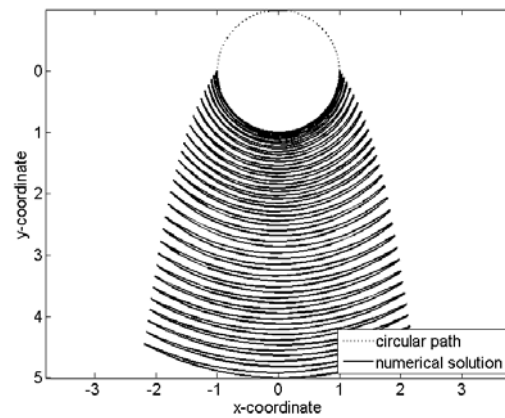


Figure 4: Result of the Method with Differentiation of the Constraint

In figure 5 the increasing error from the numerical solution to the circular path is shown.

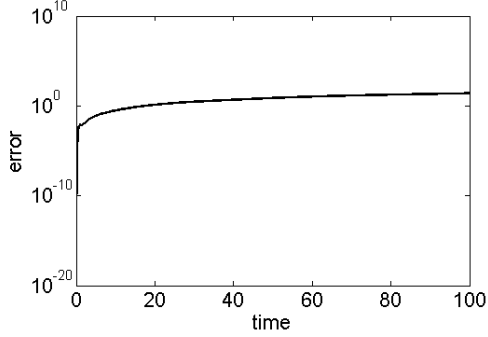


Figure 5: Error of the Method with Differentiation of the Constraint

The observed "drift-off" shows that this method is not suitable for the solving of the given DAE. Therefore other approaches for solving this DAE are necessary. The results of this other approaches are discussed in the following.

4.2.2 Baumgarte–Method

The Baumgarte–Method computes a linear combination of \ddot{g} , \dot{g} and g and substitutes g by this linear combination. The variable F can be expressed of this linear combination and can be inserted in the other equations. This results in four differential equations

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{v}_x &= -\frac{2(v_x^2 + v_y^2 + gy) + 4\alpha(xv_x + yv_y) + \beta^2(x^2 + y^2 - 1)}{2(x^2 + y^2)}x \\ \dot{v}_y &= g - \frac{2(v_x^2 + v_y^2 + gy) + 4\alpha(xv_x + yv_y) + \beta^2(x^2 + y^2 - 1)}{2(x^2 + y^2)}y. \end{aligned} \quad (11)$$

This equations are solved with the ode-solver ode45 of MATLAB.

The numerical solution calculated with the Baumgarte–Method stays close to the circular path, which can be explained by the choice of the parameters α and β . The parameters have to be chosen so that equation (4) is asymptotically stable. The numerical solution of the method, which uses the differentiation of the constraint, shows "drift-off", whereas the result of the Baumgarte–Method shows no "drift-off".

In figure 6 the result of this method is shown for $\alpha = 10$ and $\beta = 100$.

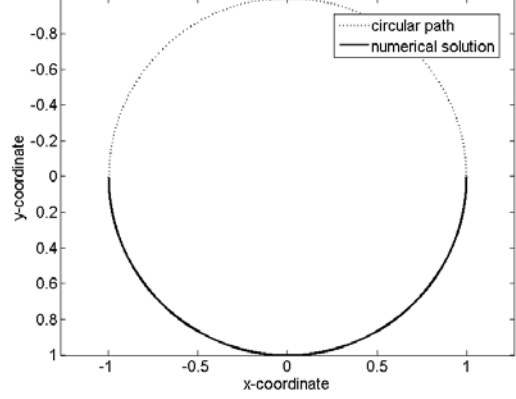


Figure 6: Result of the Baumgarte–Method

The error of the numerical solution to the circular path is shown in figure 7.

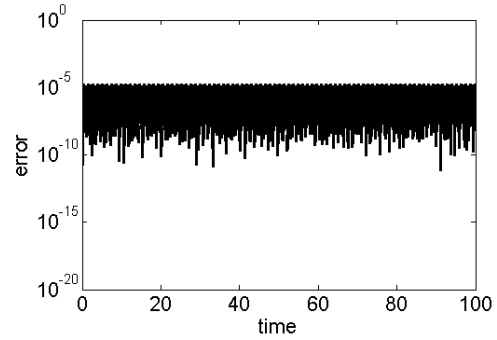


Figure 7: Error of the Baumgarte–Method

4.2.3 Pantelides–Algorithm

Application of the Pantelides–Algorithm with introduction of the dummy derivative dx leads to a system of nine equations and nine unknowns (x , dx , d^2x , d^2y , \dot{y} , \dot{v}_x , dv_x , \dot{v}_y and F). These equations are

$$\begin{aligned} dx &= v_x \\ d^2x &= dv_x \\ \dot{y} &= v_y \\ d^2y &= \dot{v}_y \\ dv_x &= -Fx \\ \dot{v}_y &= g - Fy \\ x^2 + y^2 &= 1 \\ xdx + y\dot{y} &= 0 \\ (dx)^2 + x d^2x + (\dot{y})^2 + y d^2y &= 0. \end{aligned} \quad (12)$$

For solving this problem it is necessary to use the Pantelides–Algorithm twice. Therefore it is necessary to introduce the dummy derivative dy instead of dx . This leads to the equation system

$$\begin{aligned}
 dy &= v_y \\
 d^2y &= dv_y \\
 \dot{x} &= v_x \\
 d^2x &= \dot{v}_x \\
 dv_y &= g - Fy \\
 \dot{v}_x &= -Fx \\
 x^2 + y^2 &= 1 \\
 x\dot{x} + ydy &= 0 \\
 (\dot{x})^2 + x d^2x + (dy)^2 + y d^2y &= 0.
 \end{aligned} \tag{13}$$

With this two equation system four cases are considered:

- equation system (12) and $x = \sqrt{1 - y^2}$
- equation system (12) and $x = -\sqrt{1 - y^2}$
- equation system (13) and $y = \sqrt{1 - x^2}$
- equation system (13) and $y = -\sqrt{1 - x^2}$.

Each of this equations is solved with the ode–solver ode15i of MATLAB. The result of the simulation using the Pantelides–Algorithm stays close to the circular path for the chosen initial values. The result looks similar to figure 6.

4.3 Regularisation by Projection

In the following two methods using projection are considered.

4.3.1 Standard Projection Method

After each solving step, the numerical solution is projected orthogonally onto the manifold M which is given by

$$M = \{(x, y, v_x, v_y) \in \mathbb{R}^4 : x^2 + y^2 - 1 = 0, xv_x + yv_y = 0\}. \tag{14}$$

The used projection $p : \mathbb{R}^4 \rightarrow M$ onto the manifold M is defined by

$$p \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} = \begin{pmatrix} p_1(x) \\ p_1(y) \\ p_2(v_x) \\ p_2(v_y) \end{pmatrix} \tag{15}$$

with the mapping p_1

$$\begin{aligned}
 x &\mapsto \frac{x}{\sqrt{x^2 + y^2}} \\
 y &\mapsto \frac{y}{\sqrt{x^2 + y^2}}
 \end{aligned} \tag{16}$$

and the mapping p_2

$$\begin{aligned}
 v_x &\mapsto (-p_1(y)v_x + p_1(x)v_y)(-p_1(y)) \\
 v_y &\mapsto (-p_1(y)v_x + p_1(x)v_y)p_1(x).
 \end{aligned} \tag{17}$$

The procedure first projects the position and then the velocity. In figure 8 the orthogonal projection is shown.

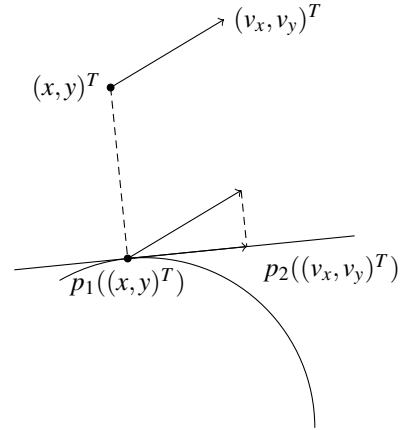


Figure 8: Orthogonal Projection

In figure 9 the result of this method is shown where the incorrect positions caused by the increasing speed (Euclidean norm of the velocity) are visible.

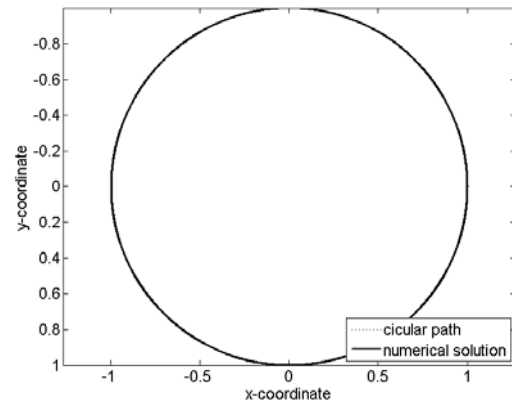


Figure 9: Result of the Orthogonal Projection Method

The Orthogonal Projection Method stays close to the circular path because the method is designed that way but the position is not correct because of the increasing speed. In figure 10 the speed is shown.

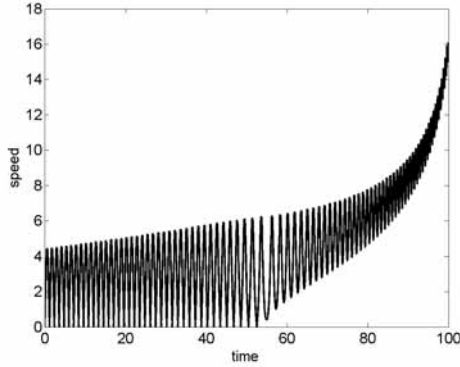


Figure 10: Speed of the Orthogonal Projection Method

4.3.2 Symmetric Projection Method

The procedure of this method leads to an equation system with x_{n+1} , y_{n+1} , $v_{x_{n+1}}$, $v_{y_{n+1}}$, μ_1 and μ_2 as unknowns. This equation system is solved by using Newton's method. In figure 11 the result is shown.

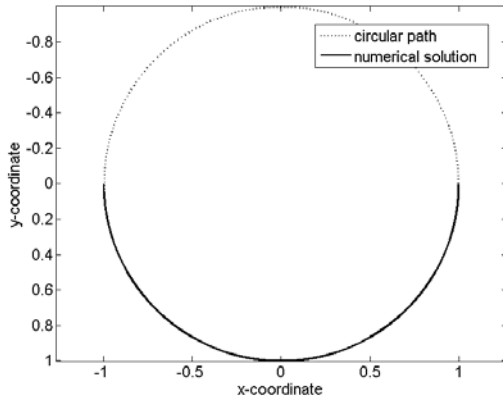


Figure 11: Result of the Symmetric Projection Method

4.4 Regularisation by Transformation

The DAE is transformed by a local state space transformation to an ODE on the manifold M (see equation (14)). The transformation $(x, y, v_x, v_y) = \psi(\varphi, \eta)$

is given by

$$\begin{aligned} x &= \cos \varphi \\ y &= \sin \varphi \\ v_x &= -\eta \sin \varphi \\ v_y &= \eta \cos \varphi. \end{aligned} \quad (18)$$

This transformation leads to the ODE

$$\begin{aligned} \dot{\varphi} &= \eta \\ \dot{\eta} &= g \cos \varphi. \end{aligned} \quad (19)$$

The state space can be transformed globally and therefore an ODE can be generated, which is an advantage for the numerical solution. The obtained ODE is solved with the ode-solver ode45 of MATLAB. Through transformation to polar coordinates the numerical solution stays on the circular path. This method leads to the most simple equations. For results of this approach see section 4.5.

4.5 Comparison

In table 2 the error $e := \max_i (x_i^2 + y_i^2 - 1)$ and elapsed time, where x_i and y_i are the numerical solutions for the positions at each solver time step t_i , are shown for the chosen initial values and till time 100. In the table below DC means differentiation of the constraint, B is the Baumgarte-Method, P the Pantelides-Algorithm, OP the Orthogonal Projection Method, SP the Symmetric Projection Method and T the transformation of the state space. The error and the elapsed time for the Baumgarte-method are calculated for $\alpha \neq \beta$ and $\alpha = \beta$.

Method	Error	Time (s)
DC	24.731	1.095
B ($\alpha \neq \beta$)	$1.909 \cdot 10^{-5}$	1.696
B ($\alpha = \beta$)	$2.463 \cdot 10^{-4}$	0.723
P	$3.345 \cdot 10^{-4}$	8.539
OP	$5.551 \cdot 10^{-16}$	76.175
SP	$2.701 \cdot 10^{-8}$	71.729
T	$2.220 \cdot 10^{-16}$	0.550

Table 2: Error e and Elapsed Time of the Approaches

The error of the method with differentiation of the constraint shows "drift-off" and therefore the error is the biggest of all approaches. The other methods have

small errors, whereas the transformation to polar coordinates has the smallest value. The Orthogonal Projection Method also has a small error, but the result is not correct due to the falsified velocity therefore this method is not interesting. The choice of α and β of the Baumgarte–Method result in different values for the error e . The state space transformation also has a very small error and the position is correct. Therefore this method would be one of the best for this problem. The elapsed times of the two projection methods are bigger than the of the other methods because the Orthogonal Projection Method uses a self written explicit Euler–method and the algorithm of the Symmetric Projection Method solves a linear equation system in each step. All the other approaches use for the solving of the obtained equations ode–solvers from MATLAB.

5 Conclusion

Finally it has been shown that only differentiating and substituting the constraint equations by their derivatives is no suitable method for solving differential–algebraic equations. Therefore other approaches are necessary. These other methods are using different backgrounds like differentiation or projection.

For the chosen example and initial values the methods show mostly results which are close to the circular path. One approach has problems with the correct positions due to the increasing Euclidean norm of the velocity. Another fact is that for other initial values the method can work worse regarding the distance to the circular path. The state space transformation is one of the best approaches, if the transformation is global.

At last it is possible that one of the mentioned approaches works for the chosen case study but is not suitable for another differential–algebraic equation system. Therefore in the future it is necessary to test the methods also for another example.

References

- [1] P. Kunkel and V. Mehrmann. *Differential-Algebraic Equations: Analysis and Numerical Solution*. European Mathematical Society, Finland, 2006.
- [2] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer, Deutschland, 2002.
- [3] E. Eich and M. Hanke. *Regularization Methods for Constrained Mechanical Multibody Systems*. ZAMM - Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, pp 761-773, 1995.
- [4] E. Hairer. *Geometric Integration of Ordinary Differential Equations on Manifolds*. BIT Numerical Mathematics, Volume 41, Issue 5, pp 996-1007, 2001.
- [5] E. Hairer. *Symmetric Projection Methods for Ordinary Differential Equations on Manifolds*. BIT Numerical Mathematics, Volume 40, issue 4, pp 726-734, 2000.
- [6] F. E. Cellier and E. Kofman. *Continuous System Simulation*. Springer, USA, 2006.

Convection-diffusion simulation in two different dimensions

Stefanie Winkler¹, Martin Bicher², Felix Breitenecker¹

¹Vienna University of Technology

²dwh GmbH Simulation Services

stefanie.winkler@tuwien.ac.at

This paper deals with the principle of groundwater pollution. The basic equation for pollution distribution is the diffusion equation. In the natural but also economical science different kinds of this equation are used. This work focuses on the convection-diffusion equation. Using this equation the diffusive behavior of the pollution influenced by a velocity field is described. Several approaches, ranging from analytical solutions to some chaotic particle movement approaches, are used for realization in one- and two-dimensional domains.

1 Introduction

In chemistry as well as in biology the reaction-diffusion equation plays a very important role. This equation can be used to recreate pattern formations of a fish' skin or cat's fur. There are also many physical applications of diffusion, e.g. the heat equation. However, diffusion is also used to foresee the behavior of buyers of stocks in the financial market. In the following the convection-diffusion equation is used.

$$\frac{\partial c}{\partial t} = D \cdot \nabla^2 c - v \cdot \nabla c \quad (1)$$

Equation (1) describes diffusive distribution influenced by a velocity field. The used velocity field is constant and only nonzero along x -direction. The groundwater pollution is a vivid application of (1).

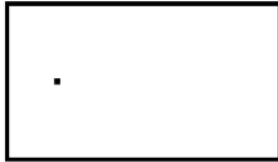


Figure 1: A schematic illustration of the described area.

Instead of using a difficult realistic geometry of a surrounding a two-dimensional rectangle is considered.

This rectangle is embedded in a Cartesian coordinate system. In the origin a source of pollution is placed as shown in figure 1. There are two different types of source. On the one hand a constantly releasing pollution source can be used. On the other hand the pollution distribution of an instantaneous releasing source can be considered. In the following the second kind of source is used for various simulation approaches in one and two dimensions.

2 Analytical Solution

Usually it is not easy or even possible to find an analytical solution. Due to the used initial and boundary conditions an analytical solution can be given. Both solutions, one- and two-dimensional, are used to validate the different methods.

2.1 One-dimensional

The regarded equation and its condition are given as follows

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} - v \frac{\partial c}{\partial x} \quad \text{with} \quad c(x, 0) = \delta(x) \quad (2)$$
$$\lim_{x \rightarrow \pm \infty} c(x, t) = 0.$$

Using different substitutions [1] the equation (2) can be written as

$$\begin{aligned} \tau &= Dt, & b &= \frac{v}{D} \\ y &= x - b\tau, & y_0 &= b\tau_0 \\ \frac{\partial c(y, \tau)}{\partial \tau} &= \frac{\partial^2 c(y, \tau)}{\partial y^2}. \end{aligned} \quad (3)$$

After multiplying equation (3) by $e^{-p\tau}$ and integrating it with respect to τ one obtains an ordinary differential equation which can be easily solved. Using the inverse Laplace-transformation and the substitutions backwards we obtain the solution of the one-dimensional problem.

$$c(x, t) = \frac{1}{\sqrt{4\pi Dt}} e^{-\frac{(x-vt)^2}{4Dt}} \quad (4)$$

2.2 Two-dimensional

In order to solve the two-dimensional equation

$$\begin{aligned} \frac{\partial c}{\partial t} &= D \cdot \frac{\partial^2 c}{\partial x^2} + D \cdot \frac{\partial^2 c}{\partial y^2} - v \cdot \frac{\partial c}{\partial x} \quad \text{with} \\ c(x_0, y_0, 0) &= \delta(x)\delta(y) \\ \lim_{x, y \rightarrow \infty} c(x, y, t) &= 0 \\ \lim_{x, y \rightarrow -\infty} c(x, y, t) &= 0 \end{aligned} \quad (5)$$

a solution of the following form is assumed. [2]

$$c(x, y, t) = g_1(x, x_0, t)g_2(y, y_0, t) \quad (6)$$

The functions g_1 and g_2 are solutions of the one-dimensional convection-diffusion equation with constant coefficients. Therefore g_1 and g_2 can be taken from the one-dimensional analytical solution (4).

$$\begin{aligned} g_1(x, x_0, t) &= \frac{A_1}{2\sqrt{D\pi t}} \exp\left(\frac{-(x-x_0-vt)^2}{4Dt}\right) \\ g_2(y, y_0, t) &= \frac{A_2}{2\sqrt{D\pi t}} \exp\left(\frac{-(y-y_0)^2}{4Dt}\right) \end{aligned} \quad (7)$$

The source is located at the origin therefore the values $x_0 = 0$ and $y_0 = 0$ can be inserted. Additionally the integral over the whole domain has to be 1.

$$\begin{aligned} 1 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(x, y, t) dx dy = \\ &= \int_{-\infty}^{\infty} g_1(x, 0, t) dx \int_{-\infty}^{\infty} g_2(y, 0, t) dy = A_1 A_2 \end{aligned} \quad (8)$$

This integration result leads to the analytical solution in two dimensions.

$$c(x, y, t) = \frac{1}{4D\pi t} \exp\left(\frac{-(x-vt)^2 - y^2}{4Dt}\right) \quad (9)$$

3 Numerical Approximation

This section introduces two types of numerical approximations. On the one hand there is the finite difference method (FDM). In this approximation the derivative of the differential equation is approached by taking the difference quotient of the neighboring grid points. The method is easy to use but slightly weak concerning the accuracy. The second method is the finite element method (FEM) and is based on formulating variations of the differential equation. FEM determines approximated solutions consisting of piecewise defined polynomials on a fine resolution of the domain. The advantage of FEM is the suitability for any geometry.

3.1 Finite Difference Method

The finite difference method is implemented for the one- and two-dimensional case.

3.1.1 One-dimensional

Using finite differences to approximate the first and second derivatives the partial differential equation (1) transforms into an ordinary differential equation.

$$\frac{dc}{dt} = D \cdot \frac{c_{i+1} - 2c_i + c_{i-1}}{x^2} - v \cdot \frac{c_i - c_{i-1}}{dx} \quad (10)$$

The time derivative can be replaced as follows

$$\frac{dc}{dt} = \frac{c^{k+1} - c^k}{\Delta t}. \quad (11)$$

Using (11) equation (10) can also be written as a matrix product

$$\frac{c^{k+1} - c^k}{\Delta t} = S \cdot c^k = S \cdot \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \quad (12)$$

with

$$S := \begin{pmatrix} -\frac{D}{h^2} + \frac{v}{h} & \frac{2D}{h^2} - \frac{v}{h} & 0 & \dots & 0 \\ \frac{D}{h^2} + \frac{v}{h} & \frac{2D}{h^2} - \frac{v}{h} & \frac{D}{h^2} & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \frac{D}{h^2} + \frac{v}{h} & \frac{2D}{h^2} - \frac{v}{h} & \frac{D}{h^2} \\ 0 & \dots & 0 & -\frac{D}{h^2} + \frac{v}{h} & \frac{D}{h^2} - \frac{v}{h} \end{pmatrix}$$

whereas c^k is the current concentration of pollution and c^{k+1} the concentration in the next time step. In order to determine c^{k+1} using the Explicit Euler equation (12) is rearranged.

$$c^{k+1} = (\Delta t S + I) c^k \quad (13)$$

It is well known that the Explicit Euler can be unstable using the wrong step size relation. Notation (12) can be also used to find the Implicit Euler formulation. The current concentration on the right hand side in equation (12) is replaced by the concentration of the future time step in order to obtain the implicit formulation.

$$c^{k+1} = (I - \Delta t S)^{-1} c^k \quad (14)$$

3.1.2 Two-dimensional

Regarding the problem formulation in two dimensions the finite difference method looks a little bit different. Due to the fact that an equidistant grid, $dx = dy$ is used the approximation can be given as follows

$$\begin{aligned} \frac{dc}{dt} = & D \cdot \frac{c_{x+1,y} + c_{x-1,y} - 4c_{x,y} + c_{x,y+1} + c_{x,y-1}}{dx^2} \\ & - v \frac{c_{x,y} - c_{x-1,y}}{dx}. \end{aligned} \quad (15)$$

In contrary to the two-dimensional case the matrix notation is not as easy as in one dimension.

$$c_{x,y}(t + \Delta t) = c_{x,y}(t) + h \cdot \frac{dc}{dt} \quad (16)$$

Therefore only the Explicit Euler method is implemented as shown in (15).

3.2 Finite Element Method

The finite element method was only realized for the convection-diffusion equation in one dimension.

$$\begin{aligned} \frac{\partial c}{\partial t} - D \frac{\partial^2 c}{\partial x^2} + v \frac{\partial c}{\partial x} &= 0 & \text{in } \Omega \\ c &= 0 & \text{on } \partial\Omega \end{aligned} \quad (17)$$

First of all the weak solution of (17) is formalized using a test function of the according Sobolev space $\phi \in H_0^1$.

$$\int_{\Omega} \frac{\partial c}{\partial t} \phi d\Omega + \int_{\Omega} (D \nabla c \nabla \phi + v \nabla c \phi) d\Omega = 0 \quad (18)$$

The formulation of the Galerkin approximation is necessary to formulate the solution equation of the finite element method.

$$c^n(x) = \sum_{j=1}^n c_j \varphi_j(x) + c_0(x) \quad (19)$$

The unknown variables c_j in equation (19) have to be determined. Using linear basis functions called 'hat-functions' for ϕ a linear system of n equations with

n unknowns, called the Galerkin formulation, results. [3]

$$\begin{aligned} \sum_{j=1}^{n_e} \frac{\partial c_j}{\partial t} \int_{\Omega^{e_k}} \varphi_j \varphi_i d\Omega \\ + \sum_{j=1}^{n_e} c_j \int_{\Omega^{e_k}} (D \nabla \varphi_j \nabla \varphi_i + \nabla \varphi_j \varphi_i) d\Omega = 0 \end{aligned} \quad (20)$$

In equation (20) n_e is the number of elements in every finite element and Ω^{e_k} is the domain of element e_k . Equation (20) can also be written in a short form.

$$\begin{aligned} \dot{c} \cdot M + c \cdot S &= 0 \quad \text{with} \\ m_{ij} &= \int_{\Omega^{e_k}} \varphi_i \varphi_j d\Omega \\ s_{ij} &= \int_{\Omega^{e_k}} (D \nabla \varphi_i \nabla \varphi_j + \nabla \varphi_i \varphi_j) d\Omega \end{aligned} \quad (21)$$

The matrices of (21) are called mass matrix M and stiffness matrix S . Considering the mentioned 'hat-functions' it is clear, that only a few of the possible integrals are not equal zero. Those basis functions which correspond to the corner points of the element will lead to non trivial results. Because the element i is connected to $i-1$ and $i+1$ the profile of the matrices is a band matrix with width three.

$$M \frac{c^{k+1} - c^k}{\Delta t} + \theta S c^{k+1} + (1 - \theta) S c^k = 0, \quad 0 \leq \theta \leq 1 \quad (22)$$

Equation (22) is called θ -method and will be used to present implicit and explicit methods for solving (21). The most common values for θ are:

- $\theta = 0$, Explicit Euler
- $\theta = 1$, Implicit Euler
- $\theta = \frac{1}{2}$, Implicit Heun

Using this method the Explicit and Implicit Euler algorithm can be given.

$$c^{k+1} = M^{-1} (M - \Delta t S) c^k \quad (23)$$

$$c^{k+1} = (M + \Delta t S)^{-1} M c^k \quad (24)$$

4 Random Walk

An alternative method for simulating transport is the so-called random walk. This approach is contrary to the numerical solutions. The focus changes from a macroscopic view to the simulation of microscopic behavior of diffusion by analyzing movements of single particles.

4.1 Intuitive Approach

The intuitive approach describes a model which uses no grid or collision rules. It is implemented again for both dimensions.

4.1.1 One-dimensional

At the beginning $t = 0$ all the particles are placed in the origin presenting the source of pollution. The pollution injection happens only at $t = 0$. The simulation focuses on the convection and diffusion behavior of these initial particles. In this approach the movement of particles is described by:

$$\begin{aligned} p_{new} &= p_{old} + r + v \cdot \Delta t \\ r &= X \cdot \Delta x \end{aligned} \quad (25)$$

The particle motion in (25) consists of three parts. In order to get the new position p_{new} at time $t + \Delta t$ these three components are summed up. The variable p_{old} stands for the position at time t . The velocity field v is multiplied by the step size. The variable r describes the diffusive movement of a particle for one time step and is added to the former particle position p_{old} . The second equation in (25) defines the movement r in particular. It consists of the step size in space Δx and a normally distributed random variable X with mean zero and unit variance. In every time step the new position of every particle is calculated with equation (25). The simulation ends when the chosen simulation time t_{end} is reached.

4.1.2 Two-dimensional

For expansion in a two-dimensional domain the movement has to be defined in a different way. There is no

initial velocity but there is an initial direction of every particle d_0 . The diffusive transport is realized by using a normally distributed random variable X and a uniformly distributed random number U . X is used to generate a random length and U chooses a coincidental direction.

$$r = X \cdot \Delta x \quad \alpha = U \cdot 2\pi$$

$$d_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad d_{n+1} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \cdot d_n \quad (26)$$

In (26) r stands for the distance the particle moves in a certain time step. The influence of this parameter is similar to the diffusion coefficient. X is the mentioned normally distributed random variable and Δx describes the step size in space. The second equation of (26) sets the direction for the particle's next move. The initial direction d_0 is only necessary for the recursive definition. During simulation the direction of the last movement is used to calculate the next one. The convection is realized by a shift in flow direction along x . The final formulation of the random walk movement can be given as follows

$$p_{new} = p_{old} + d \cdot r + vdt. \quad (27)$$

4.2 Gaussian Approach

This approach shows the connection between a random walk approach and the analytical solution.

4.2.1 One-dimensional

The analytical solution of the convection-diffusion equation (2) is used to define the particle movement. Considering the probability density function of a normal or Gaussian distribution

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (28)$$

the formal equivalence to the analytical solution (4) is obvious. The parameters used in (28) stand for the mean value μ and the standard deviation σ which characterize the position and the width of the Gaussian

bell curve in a unique way. Therefore the according parameters in (4) can be read out. [4]

$$\mu = v \cdot t$$

$$\sigma^2 = 2 \cdot Dt \quad (29)$$

Due to the properties and meaning of the parameters in (29) the height and width of the concentration peak depending on time is given. The corresponding particle movement using (29) can be formulated as follows

$$p_{new} = p_{old} + v\Delta t + \sqrt{2 \cdot D\Delta t}X. \quad (30)$$

The variable X stands for a normally distributed random number with mean zero and unit variance as in the intuitive approach. X is newly generated in every step for each particle. Identifiable by the velocity v the second term stands for the convective motion. This term is equal to the term of the intuitive approach. The radical term describes the diffusive motion and is based on the standard derivation.

4.2.2 Two-dimensional

In order to enlarge this approach in two dimensions the movement along y-direction has to be added. For an expansion in a two-dimensional domain the y-component of the movement has to be defined. Due to the fact that there is no flux the new particle position can be calculated using

$$p_x^{new} = p_x^{old} + v\Delta t + \sqrt{2 \cdot D\Delta t}X_x$$

$$p_y^{new} = p_y^{old} + \sqrt{2 \cdot D\Delta t}X_y. \quad (31)$$

The variables X_x and X_y stand for independent normally distributed random numbers which are newly generated in every step for each particle. The term $v\Delta t$ describes the convective transport. Due to the fact that the diffusion coefficient is equal for the x- and y-direction the diffusive movement $\sqrt{2 \cdot D\Delta t}$ in the random walk definition (31) is the same.

5 Results

In the following section the analytical solutions in both dimensions are compared to the various approaches. The different concentration errors are discussed. In general the parameter setting is: diffusion coefficient $D = 0.02$ and velocity $v = 0.02$. The step sizes Δt and Δx are variable. The regarded simulation time varies between $t_{end} = 250$ and $t_{end} = 500$.

5.1 Analytical vs. Numerical

First of all the numerical solutions are considered.

5.1.1 FDM 1D

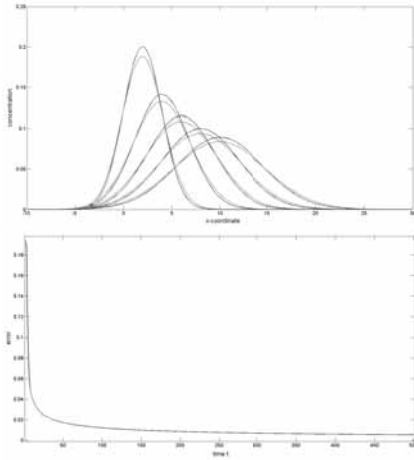


Figure 2: Comparison of the analytical solution and FDM using matrix notation.

In the upper plot in figure 2 the red curve is the analytical solution and the blue line sketches the numerical approximation using the Implicit Euler algorithm. In the lower plot the difference of both solutions is shown.

The results in table 1 show the instability of the Explicit Euler method. The Implicit Euler algorithm is not only ultra-stable but also faster and more exact than the Explicit Euler. The approximation using finite differences is well-fitting.

Δt	Δx	Explicit Euler		Implicit Euler	
		$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _\infty$	$\ \cdot\ _1$
1	1	0.016	$4.231E^{-4}$	0.016	$4.753E^{-4}$
1	$\frac{1}{2}$	0.009	$1.404E^{-4}$	0.010	$1.600E^{-4}$
$\frac{1}{2}$	$\frac{1}{4}$	0.005	$0.831E^{-5}$	0.005	$7.323E^{-5}$
$\frac{1}{2}$	$\frac{1}{16}$	NaN	NaN	0.002	$3.531E^{-5}$

Table 1: Error values of FEM using Explicit and Implicit Euler.

5.1.2 FDM 2D

The results regarding the two-dimensional implementation show a similar behavior.

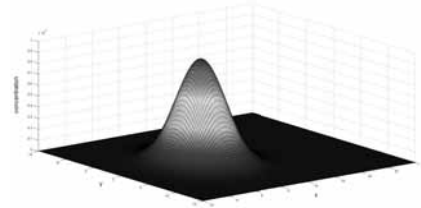


Figure 3: The analytical and numerical solution is shown.

In figure 3 the surface plot shows the concentration of the numerical result. In the following the error values are studied in detail.

Δt	Δx	explicit Euler	
		$\ \cdot\ _\infty$	$\ \cdot\ _1$
1	1	0.0027	$1.5624E^{-4}$
$\frac{1}{2}$	$\frac{1}{4}$	$9.1479E^{-4}$	$1.4640E^{-5}$
1	$\frac{1}{2}$	0.0017	$3.7791E^{-5}$
$\frac{1}{4}$	$\frac{1}{8}$	$.E^{119}$	$.E^{120}$
$\frac{1}{2}$	$\frac{1}{2}$	0.0016	$3.8172E^{-4}$
$\frac{1}{8}$	$\frac{1}{8}$	$4.7570E^{-4}$	$8.5291E^{-6}$
1	$\frac{1}{4}$	$.E^{32}$	$.E^{33}$

Table 2: The error values for FDM using Explicit Euler are shown.

Also in the two-dimensional case the Explicit Euler works not for all parameter choices. The error values are again quite good. The finite difference method of the two-dimensional domain approximates

the convection-diffusion equation in an appropriate way.

5.1.3 FEM 1D

The accuracy of the finite element method is better than of the finite difference method.

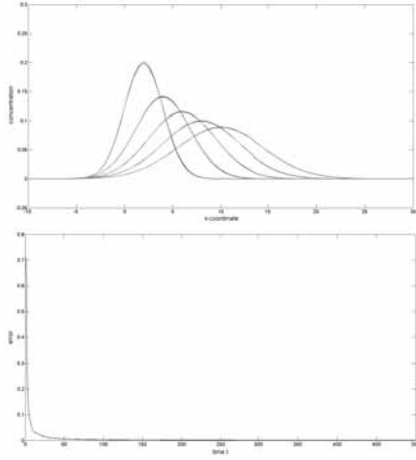


Figure 4: The error for the Implicit Euler algorithm of the FEM is shown.

In figure 4 above the upper plot shows the analytical solution as well as the finite element method using Implicit Euler. It is hard to distinguish the different curves.

Δt	Δx	Explicit Euler		Implicit Euler	
		$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _\infty$	$\ \cdot\ _1$
1	1	$7.18E^{-4}$	$3.16E^{-5}$	$9.95E^{-4}$	$3.03E^{-5}$
1	$\frac{1}{2}$	$6.23E^{-4}$	$8.60E^{-5}$	$6.09E^{-4}$	$8.54E^{-5}$
$\frac{1}{2}$	$\frac{1}{4}$	$3.13E^{-4}$	$1.02E^{-4}$	$2.74E^{-4}$	$1.01E^{-4}$
$\frac{1}{2}$	$\frac{1}{8}$	NaN	NaN	$2.49E^{-4}$	$1.05E^{-4}$

Table 3: Depending on the used FEM the error values are shown.

The instability of the Implicit Euler is shown in the last row of table 3. In general the error results are smaller compared to the results of the finite difference method in one dimension. The finite element method approximates the convection-diffusion equation better than the finite difference method.

5.2 Analytical vs. Stochastic

The accuracy of the random walk approaches is discussed in the following paragraph.

5.2.1 One-dimensional

The two random walk implementations are compared to the analytical solution.

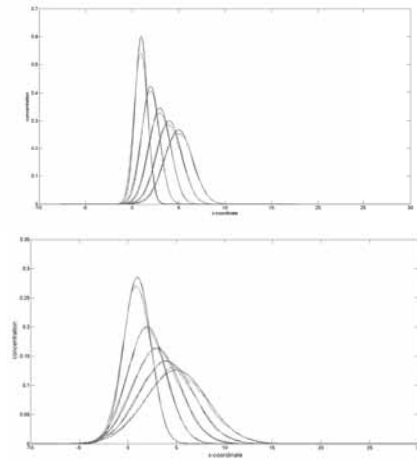


Figure 5: Comparison of intuitive and stochastic based random walk are shown.

The two graphics in figure 5 show both random walk approaches colored in red and the analytical solution in blue. In the upper plot the intuitive implementation is shown. The lower plot sketches the Gaussian-based version of random walk. In the numerical comparisons the simulation time is $t_{end} = 500s$. Due to long execution times for the particle movement this parameter is reduced to $t_{end} = 250s$. The diffusion coefficient is usually set to $D = 0.02$ but modifies if the intuitive approach is used.

The table 4 shows all the error results of the parameter study comparing the analytical solution and both random walk approaches. Adapting the diffusion coefficient both approaches fues into one single random walk implementation. The diffusion coefficient for the Gaussian-based algorithm is set to $D = 0.02$. Using the intuitive approach the diffusion coefficient of the analytical solution changes to D^* to enables comparability. Due to the dependency on Δx and Δt the intuitive random walk cannot reach its performance. On

Δt	Δx	D^*	intuitive RW		Gaussian RW	
			$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _\infty$	$\ \cdot\ _1$
1	$\frac{1}{5}$	0.01	0.007	$8.948E^{-7}$	0.012	$8.948E^{-7}$
1	$\frac{1}{10}$	0.005	0.014	$4.441E^{-16}$	0.008	$9.707E^{-7}$
$\frac{1}{2}$	$\frac{1}{5}$	0.04	0.006	$3.342E^{-5}$	0.007	$8.948E^{-7}$
$\frac{1}{2}$	$\frac{1}{10}$	0.01	0.009	$8.419E^{-12}$	0.010	$9.707E^{-7}$
$\frac{1}{2}$	$\frac{1}{15}$	0.0045	0.016	$8.882E^{-16}$	0.009	$9.969E^{-7}$
1	$\frac{1}{5}$	0.01	0.010	$0.948E^{-7}$	0.007	$8.948E^{-7}$
1	$\frac{1}{10}$	0.005	0.015	$6.661E^{-16}$	0.005	$9.707E^{-7}$

Table 4: Comparison of the random walk approaches and the analytical solution.

contrary a parameter study would be necessary to find the perfect variable value for all step size combinations to perform this algorithm. Therefore, regarding simulation of the convection-diffusion equation, the implementation of the Gaussian-based random walk fits better. In the last two rows the used particle number is raised from 6000 to 8000.

5.2.2 Two-dimensional

In order to compare the analytical solution to a random walk approach the results have to be adapted. In the random walk the output describes the smoothed amount of particles in every cell. Due to the initial Dirac-function the integral at the beginning has value one. The area of the random walk domain is discretized. Therefore the output has to be divided not only by the number of particles but also by the area of the cells used for the flattening.

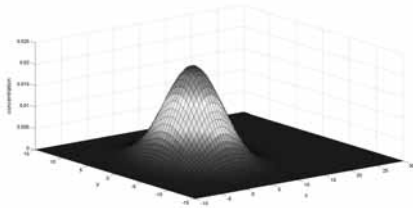


Figure 6: Analytical and Gaussian solution are shown.

Figure 6 shows the concentration results of the random walk approach. The difference between these two implementations cannot be read out exactly. Therefore a closer look to the simulation data itself is given.

Table 5 shows the approximation results. The parameter r describes the used radius for the flattening. If the

Δt	Δx	r	N	Gaussian RW	
				$\ \cdot\ _\infty$	$\ \cdot\ _1$
1	1	3	4000	$3.395E^{-3}$	$6.349E^{-4}$
1	$\frac{1}{2}$	8	4000	$5.033E^{-3}$	$3.737E^{-5}$
$\frac{1}{2}$	$\frac{1}{2}$	15	4000	$1.063E^{-2}$	$1.071E^{-3}$
$\frac{1}{2}$	$\frac{1}{4}$	15	4000	$4.526E^{-3}$	$1.005E^{-4}$
$\frac{1}{2}$	$\frac{1}{8}$	20	4000	$2.801E^{-3}$	$2.206E^{-3}$
1	$\frac{1}{4}$	20	8000	$6.764E^{-3}$	$3.338E^{-5}$
$\frac{1}{2}$	$\frac{1}{4}$	20	8000	$6.825E^{-3}$	$1.826E^{-4}$

Table 5: Error values for the Gaussian approach are shown.

spatial step size is decreasing a greater radius r can be used. If r is chosen too big compared to Δx the result loses the shape of a bell curve. Compared to the results of the numerical simulation the random walk approach leads to greater error values.

6 Conclusion

In general the finite element method approximates the convection-diffusion equation the best. Of course the very best solution is the analytical one. In spite of it all random walk approaches are quite good approximations of the convection-diffusion equation.

References

- [1] K. Schulten and I. Kosztin. *Lectures in Theoretical Biophysics.*, University of Illinois at Urbana, USA, 2000.
- [2] C. Zoppou and J.H. Knight. *Analytical solution of a spatially variable coefficient advection diffusion equation in up to three dimensions*, Applied Mathematical Modelling, S. 667-685, 1999.
- [3] Ir. A. Segal. *Finite element methods for the incompressible Navier-Stokes equations.*, Delft University of Technology, Netherlands, 2012.
- [4] P. Salamon and D. Fernandez-Garcia and J.J. Gomez-Hernandez. *A review and numerical assessment of the random walk particle tracking method.*, Journal of Contaminant Hydrology, S. 277-305, 2006.

Anwendung des Model-Based Systems Engineering zur Unterstützung der Simulation in Produktion und Logistik

Alexander Schlegel

TU Chemnitz, Professur Fabrikplanung und Fabrikbetrieb

alexander.schlegel@mb.tu-chemnitz.de

Die zunehmende Komplexität in Produktions- und Logistiksystemen erfordert immer häufiger den Einsatz der Simulation zur Analyse und Optimierung. In die Bearbeitung einer Simulationsstudie fließt das Fachwissen verschiedener Experten und Entscheidungsträger ein. Die Reduzierung der Komplexität und ein einheitliches Verständnis für die Problemstellung sind eine Voraussetzung für die erfolgreiche und effiziente Durchführung von Simulationsprojekten. Die Anwendung des Model-Based Systems Engineering in Simulationsprojekten stellt einen vielversprechenden Ansatz dar und ist Gegenstand des vorliegenden Beitrags.

1 Einleitung

Die vorherrschenden Rahmenbedingungen unter denen Unternehmen am Markt operieren, sind durch die Globalisierung und einer damit einhergehenden Verschärfung der Wettbewerbssituation, einer fortschreitenden Nachfragedifferenzierung und kürzeren Produktlebenszyklen geprägt. Die Unternehmen stehen vor der Herausforderung die Folgen dieser Entwicklung, beispielsweise die steigende Variantenvielfalt der Produkte, die zunehmend stärker miteinander vernetzten und inhomogenen Material- und Informationsflüsse sowie die situationsgerechte Anpassung der Unternehmensprozesse auf die Dynamik und Anforderungen der Absatzmärkte sicher zu beherrschen. Daraus resultiert eine Zunahme der Komplexität der Produkte, Prozesse und Strukturen im System „Unternehmen“, die sich auch auf die Subsysteme „Produktion“ und „Logistik“ auswirkt.

Die Komplexität eines Systems beeinflusst den Aufwand und Schwierigkeitsgrad für dessen Analyse entscheidend. Mit steigender Komplexität wird es schwieriger ein System zu beschreiben, dessen Systemzustände zu beobachten, die Auswirkungen von Änderungen im System vorherzusagen und dieses System zu steuern. Die Simulation hat sich zu einem wichtigen Hilfsmittel in vielen Bereichen der Wissenschaft und Technik entwickelt und stellt häufig die einzige Alternative zur Untersuchung von komplexen Systemen dar. Auch im Anwendungsfeld der Produktion und Logistik wird die Simulation als Analyse- und Problemlösungsmethode genutzt, wenn die betrachteten Systeme aufgrund ihrer Dynamik und Wechselwirkungen mit keinen geeigneten analytischen Verfahren abgebildet werden können, detaillier-

te Untersuchungen unter gleichzeitiger Berücksichtigung verschiedener Faktoren durchzuführen sind und stochastische Einflüsse vorliegen [1].

Bei der Durchführung von Simulationsstudien stellt das vollständige Verständnis des vorliegenden Systems mit seiner Struktur, den ablaufenden Prozessen und vorhandenen Wirkungszusammenhängen eine Grundvoraussetzung dar. Der Simulationsexperte ist hierbei auf die Zusammenarbeit mit weiteren Entscheidungs- und Informationsträgern angewiesen. Diese weisen aufgrund ihrer Funktion unterschiedliches Simulations- und Fachwissen sowie eine variierende Systemkenntnis auf. Die Herstellung eines einheitlichen Verständnisses der Problemstellung und des betrachteten Systems durch die Verwendung einer gemeinsamen Sprache wirkt dabei erfolgsfördernd.

Einen Ansatz zur Komplexitätsbewältigung und Verständnisförderung bietet die Anwendung des Model-Based Systems Engineerings (MBSE). Im Gegensatz zum klassischen Systems Engineering stützt sich der Entwicklungsprozess hierbei auf die durchgängige Anwendung eines Systemmodells zur Konzipierung, Ausarbeitung sowie Verifikation und Validierung [2].

Dieser Beitrag behandelt die Anwendung des MBSE zur Unterstützung von Simulationsprojekten. Die folgenden Kapitel erläutern zunächst die Grundlagen des Systems Engineering (SE) und Model-Based Systems Engineering. Darauf aufbauend erfolgt die Übertragung des MBSE-Ansatzes von der Produktentwicklung auf die Realisierung einer Simulationsstudie. Dabei liegt der Fokus auf der Betrachtung der konkreten Anwendungsmöglichkeiten und den damit

verbundenen Vorteilen sowie der Integration in ein bestehendes Vorgehensmodell.

2 Einführung Systems Engineering

Das SE verfügt bereits über eine längere Geschichte und die Ursprünge des heutigen Systems Engineering können bis in die 1930er Jahre zurückverfolgt werden. Im Verlauf seiner Entwicklung entstanden verschiedene universelle und fachspezifische SE-Ansätze mit unterschiedlichen Denkmodellen und Vorgehenskonzepten [3]. Ropohl beschreibt in seinen Ausführungen über die Ursprünge der Allgemeinen Systemtheorie das Systems Engineering als einen Ansatz zur Verwissenschaftlichung praktischen Problemlösens [4]. Aus der Betrachtung verschiedener praxisorientierter Systemansätze leitet Ropohl eine allgemeine Charakteristik der untersuchten Ansätze, einschließlich des Systems Engineering, ab. In dieser werden die Objekte (Betrachtungsgegenstände) als systemhafte Ganzheiten aufgefasst und entsprechende ganzheitlich konzipierte Arbeitsgruppen, bestehend aus Generalisten und Spezialisten aus verschiedenen Fachgebieten, eingesetzt. Eine aktuelle Definition bietet das International Council on Systems Engineering (INCOSE) und beschreibt Systems Engineering als einen interdisziplinären Ansatz zur Realisierung erfolgreicher Systeme mit dem Fokus auf der Definition und Dokumentation der Systemanforderungen sowie der Erarbeitung des Systemdesigns und der Überprüfung des Systems unter Berücksichtigung des Gesamtproblems [5]. Erklärtes Ziel ist die Entwicklung eines Systems, das sämtliche Kundenanforderungen über den gesamten Lebenszyklus erfüllt.

Nach Haberfellner et al. unterstützt das SE die zweckmäßige und zielgerichtete Gestaltung eines Systems, indem es durch die Vorgabe und Anwendung bestimmter Denkmodelle und Prinzipien als Leitfaden fungiert [6]. In ihrer Arbeit beschreiben Haberfellner et al. einen universellen SE-Ansatz der als methodische Komponente die Bearbeitung und Lösung von Problemen jeglicher Art unterstützen soll [7]. Dieser SE-Ansatz baut auf dem Systemdenken auf. Das Denken in Systemen findet bei der Analyse bestehender und bei der Gestaltung neuer Systeme als Hilfsmittel für die Beschreibung und das Verständnis komplexer Zusammenhänge und Strukturen Anwendung. Sowohl Haberfellner et al. als auch Ropohl erläutern für die ganzheitliche Systembetrachtung verschiedene Betrachtungsweisen (umgebungs-, wir-

kungs-, strukturorientiert) beziehungsweise Systemkonzepte (funktional, struktural, hierarchisch) durch die bestimmte Ansichten und Aspekte eines Systems hervorgehoben werden [4,7]. Dabei ist zu beachten, dass Systeme und deren Elemente sowie Beziehungen auch innerhalb dieser Sichtweisen unter verschiedenen Gesichtspunkten betrachtet und beschrieben werden können. Weiterhin besteht die Möglichkeit, dass ein Systemelement für mehrere Systemaspekte relevant ist. Trotz des abgrenzenden Charakters der Betrachtungsweisen und Systemaspekte können diese zueinander in Beziehung stehen [4,7]. Beispielsweise ist die strukturorientierte Beschreibung eines Fertigungssystems unter den Aspekten Material-, Informations- und Energiefluss möglich. Die Betriebsmittel des Fertigungssystems sind in die genannten Flusssysteme eingebunden und dementsprechend abzubilden. Des Weiteren steuert der Informationsfluss den Materialfluss und dieser beeinflusst den Energiefluss.

Der Blick auf ein System anhand verschiedener Betrachtungsweisen und Systemaspekte unterstreicht beziehungsweise vernachlässigt bestimmte Eigenschaften sowie Merkmale und reduziert somit die Menge der zu erfassenden Informationen auf das Wesentliche. Dies dient dem Auflösen der Komplexität und ist gleichzeitig eine Voraussetzung für die Anwendung von Systemmodellen. Die Aussagekraft und die ableitbaren Erkenntnisse einer Systemdarstellung werden unmittelbar durch die gewählte Sichtweise beeinflusst. Für die ganzheitliche Beschreibung eines Systems ist dieses in der Regel aus mehreren Standpunkten zu betrachten und mit verschiedenen Systemdarstellungen zu analysieren.

3 Model-Based Systems Engineering

Das Systemdenken ist ein wesentliches Grundprinzip des Systems Engineering und die Darstellung von Systemen anhand von Modellen unterstützt die ganzheitliche und interdisziplinäre Betrachtung. Model-Based Systems Engineering greift diesen Zusammenhang auf und stellt ein Systemmodell in den Mittelpunkt der Entwicklung. Gegenstand des MBSE ist der Entwurf, die Spezifikation, die Verifikation und Validierung von Systemen mit einem Systemmodell entlang des gesamten Entwicklungsprozesses. Das Systemmodell fungiert hierbei als Ausgangspunkt für die Entwicklung der erforderlichen Lösungen innerhalb der Teilentwicklungsbereiche [2]. Das INCOSE erweitert in seiner Definition den Anwendungsspiel-

raum des MBSE, der auch weitere über die eigentliche Entwicklung hinausgehende Lebenszyklusphasen mit einschließt [8].

Der MBSE-Ansatz stellt eine mögliche Umsetzung der SE-Methodik dar und zielt gleichzeitig auf einen Paradigmenwechsel vom dokumentenzentrierten zu einem modellbasierten Systems Engineering ab. Die dokumentenzentrierte Entwicklung von Systemen findet heute nach wie vor Anwendung. Die Verfassung der Anforderungs- und Systemarchitekturspezifikation sowie die Spezifikationen der verschiedenen Entwicklungsbereiche erfolgt in Form von Dokumenten. Die Verwendung einer Vielzahl von Dokumenten entlang eines gesamten Entwicklungsprojektes erschwert jedoch die Nachvollziehbarkeit von formulierten Anforderungen und Spezifikationen. Deren Beschreibung in Textform behindert eine schnelle Informationsaufnahme und birgt das Risiko der fachspezifischen Verständnisprobleme. Auch der Einsatz von Modellen ist grundsätzlich nicht neu, jedoch erfolgt dieser überwiegend in den jeweiligen Entwicklungsdisziplinen (z.B. Konstruktion, Elektronik, Software). Die Anwendung des MBSE bietet hingegen eine ganze Reihe von Vorteilen. In der modellbasierten Entwicklung nutzen alle an der Entwicklung beteiligten Fachbereiche das Systemmodell gemeinsam, entnehmen daraus Informationen und fügen dem Modell neue Informationen hinzu. Die aus der Entwicklung heraus entstehenden Informationen werden zentral im Modell verwaltet und gepflegt. Das aufgebaute Lösungswissen ist im Systemmodell konserviert, durch die standardisierte Abbildung des Systems einfach wiederverwendbar und Änderungen daran erfolgen schnell und transparent. An der Systementwicklung wirken oftmals mehrere Personen mit unterschiedlichen Fachwissen und variierender Systemkenntnis mit. Das Systemmodell verbessert die Kommunikation und schafft ein übereinstimmendes Systemverständnis zwischen allen Beteiligten durch die Verwendung einer disziplinübergreifenden einheitlichen Notation zur Beschreibung von Systemen. Des Weiteren stellt es ein wirkungsvolles Hilfsmittel zur Beherrschung der Komplexität dar, indem es die Betrachtung verschiedener Systemaspekte ermöglicht und die Nachvollziehbarkeit der Auswirkungen von Änderungen auf das Gesamtsystem fördert. Durch die Prüfung des Modells auf Konsistenz, Korrektheit und Vollständigkeit können Risiken vermindert und die Qualität des zu entwickelnden Produktes verbessert werden.

Die Anwendung von Modellen stellt nicht nur im Systems Engineering, sondern auch allgemein ein wichtiges Hilfsmittel im Umgang mit komplexen Aufgabenstellungen dar und wird sowohl in wissenschaftlichen Disziplinen als auch in der Praxis in vielfältiger Art und Weise betrieben. Unter einem Modell wird der VDI-Richtlinie 3633 zufolge die „[...] vereinfachte Nachbildung eines geplanten oder existierenden Systems mit seinen Prozessen in einem anderen begrifflichen oder gegenständlichen System.“ verstanden [9]. Modelle beinhalten im Allgemeinen nicht alle Eigenschaften des Originals, sondern lediglich die für Modellbildner beziehungsweise -nutzer relevanten Attribute, und werden für einen bestimmten Zweck erstellt sowie Zeitraum angewendet [10]. Im Anwendungsfeld des MBSE erfolgt eine Erweiterung dieses Verständnisses, indem eine entsprechende formale Repräsentation des Modells zur Gewährleistung der weiteren rechnergestützten Verarbeitung des Modells als eine Grundvoraussetzung gilt [11]. Diese Forderung steht in einem engen Zusammenhang mit dem Konzept der modellgetriebenen Architektur (Model-driven Architecture – MDA). Die Idee hinter diesem Ansatz besteht darin, aus den rechnerverarbeitbaren Modellen durch Modelltransformation andere Produkte für die Systementwicklung abzuleiten, beispielsweise Quellcode, eine Dokumentation oder ein neues Modell. Zusammenfassend stellt das Systemmodell das Abbild eines realen beziehungsweise zu entwickelnden Systems dar, berücksichtigt hierbei nur die relevanten Eigenschaften, ermöglicht unterschiedliche Sichten auf die Informationen und ist maschinell auswertbar [2].

Die durchgängige Anwendung von Modellen beschreiben Eigner et al. in ihrer Erweiterung des V-Modells für MBSE an einem dreistufigen Ansatz [12]. In der ersten Stufe, der Modellbildung und Spezifikation, wird ein System anhand qualitativer Modelle abgebildet. Die erstellten Modelle beinhalten Anforderungs-, Funktions- sowie Systemstrukturen und können aufgrund ihres beschreibenden Charakters nicht simuliert werden. Die erarbeitete funktionale Lösung (Funktionen und Teilfunktionen) geht in die zweite Stufe ein. Mit der Stufe der Modellbildung und ersten Simulation entstehen quantitative und simulierbare Modelle unter Einbeziehung verschiedener Fachdisziplinen. In diesem Ansatz umfassen die Stufen Modellbildung und Spezifikation sowie Modellbildung und erste Simulation die eigentliche interdisziplinäre Systementwicklung. Resultat ist ein Lösungskonzept mit dem das Verhalten und die

Struktur des Systems spezifiziert werden. Darauf aufbauend finden im letzten Schritt, der disziplinspezifischen Modellierung, die gängigen Modelle im Entwicklungsprozess der jeweiligen Fachdisziplin Anwendung. Die Abgrenzung der verschiedenen zum Einsatz kommenden Modelle voneinander ist am Beispiel der Produktentwicklung in der folgenden Abbildung dargestellt (in Anlehnung an [12,13]).

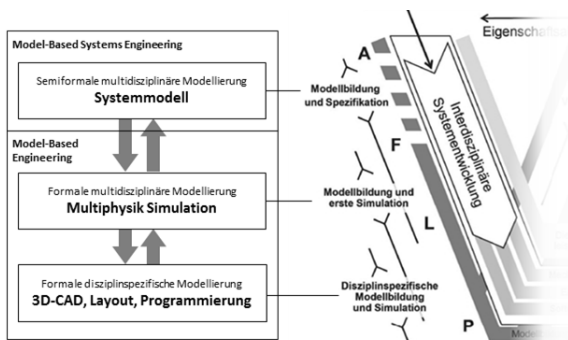


Abbildung 1. Modelle der Produktentwicklung

Eigner et al. bemängeln in ihren Ausführungen jedoch das Fehlen einer Möglichkeit zum durchgängigen Daten- und Informationsaustauschs zwischen den einzelnen Stufen und benennen weiterhin Probleme bei der Verwaltung von Systemmodellen [12,14]. Das INCOSE sieht in den Problemen im Daten- und Modellaustausch aufgrund fehlender Interoperabilität der verwendeten Werkzeuge ein wesentliches Hindernis für die weitere Etablierung des MBSE [8]. Eine weitere große Herausforderung sehen Stark et al. im Aufwand zur Erfassung und Beibehaltung der Qualität der Verknüpfungsinformationen zwischen diesen Modellen [15].

Zur Erstellung eines Systemmodells und dessen Anwendung im Sinne des MBSE sind eine Modellierungssprache, eine Methode, ein Werkzeug sowie mit der MBSE-Thematik vertrautes Personal erforderlich [13,16]. Mit der Anwendung einer Modellierungssprache wird die allgemeine modellbasierte Spezifikation eines Systems erst ermöglicht sowie die Art der formalen Beschreibung und der darstellbaren Informationen festgelegt. Modellierungssprachen wie SysML und UML [17] sowie OPM [18] finden hierbei Anwendung. Die eingesetzte Methode bestimmt, wann welche Aspekte des Systems zu welchem Zeitpunkt und Zweck spezifiziert werden. Als etablierte Methoden der Systemmodellierung sind unter anderem OOSEM (Object oriented systems engineering

method) und RUP SE (Rational Unified Process for Systems Engineering) [19], SYSMOD [17] sowie FAS (Funktionale Architekturen für Systeme) [20] zu nennen. Es existiert eine breite Vielfalt an Werkzeugen, beispielsweise im Bereich SysML und UML Artisan Studio (Atego), Cameo Systems Modeler (No Magic), Enterprise Architect (Sparx) sowie TOPCASED/PolarSys. Die erfolgreiche Umsetzung des modellbasierten Systems Engineering Ansatzes gelingt jedoch nur, wenn neben den genannten Faktoren auch die an der Entwicklung beteiligten Personen die notwendigen Kenntnisse für die Anwendung der MBSE-Methodik, der Modellierungssprache und des Werkzeugs besitzen.

Das INCOSE erwartet in seinem Ausblick eine zunehmende Bedeutung des MBSE in der Praxis des Systems Engineering und eine Ausweitung der Anwendungsfelder bis zum Jahr 2020 [8].

4 Anwendung des MBSE in der Simulation

In der Fachliteratur behandelte modellbasierte Ansätze im Kontext der Simulation in Produktion und Logistik fokussieren überwiegend auf die Generierung von Simulationsmodellen aus einer formalen Beschreibung [21,22,23]. Des Weiteren wird die Anwendung des MBSE vor allem in Verbindung mit der interdisziplinären Entwicklung komplexer Produkte beschrieben [12,14,24]. Das Ergebnis der Arbeit eines Simulationsexperten stellt jedoch kein physisches Produkt dar. Er vollbringt eine Dienstleistung, an deren Ende der Auftraggeber eine zufriedenstellende Antwort beziehungsweise Lösung seiner Frage- oder Problemstellung erwartet. Der Einsatz der Simulation legt die Methode zur Untersuchung der Problemstellung fest und beinhaltet die Entwicklung und Verwendung eines Simulationsmodells. Neben der Erstellung des Simulationsmodells umfasst die Tätigkeit des Simulationsexperten noch weitere vor- und nachgelagerte Aufgaben wie die Angebotserstellung, die exakte Aufgabenspezifikation, die Analyse des Systems, die Beschaffung und Aufbereitung der erforderlichen Daten, die Durchführung von Experimenten sowie die Erstellung einer Dokumentation. Dabei erfolgt die Bearbeitung der erforderlichen Aufgaben in der Regel im Rahmen einer Simulationsstudie. Aufgrund dessen ist der Fokus für die Anwendung des MBSE nicht nur auf die Modellgenerierung, son-

dern ganzheitlich auf die Durchführung einer Simulationsstudie zu richten.

In der Literatur existieren zahlreiche Empfehlungen für das Vorgehen in einer Simulationsstudie, die sich im Umfang und dem genauen Ablauf voneinander unterscheiden. An dieser Stelle sei darauf hingewiesen, dass die folgenden Ausführungen im Bezug zum Vorgehensmodell nach Rabe et al. [25] stehen. Eine Simulationsstudie kann im Rahmen eines größeren Planungsprojektes oder als eigenständiges Projekt durchgeführt werden [1]. Im weiteren Vorgehen sind zunächst die Einflussfaktoren einer Simulationsstudie aus Projektsicht zu beschreiben. Darauf aufbauend ist herauszuarbeiten, inwieweit der MBSE-Ansatz diese Faktoren positiv beeinflusst und die Durchführung einer Simulationsstudie unterstützt. Eine Simulationsstudie wird in der Regel zur Beantwortung bestimmter Problemstellungen eines Auftraggebers entweder intern durchgeführt oder an einen externen Dienstleister vergeben. Die Abwicklung solcher Projekte erfolgt dabei in einem Spannungsfeld aus Zeit, Kosten und Qualität. Der finanzielle und zeitliche Aufwand für die Realisierung einer Simulationsstudie ist nicht unerheblich. In der betrieblichen Praxis stellen die erforderliche Zeit und die Kosten bei der Durchführung von Industrieprojekten kritische Faktoren dar, die es zu minimieren gilt. Die größten Zeitanteile zur Durchführung einer Simulationsstudie liegen in den für die Entwicklung des lauffähigen Modells entscheidenden Phasen Datenbeschaffung, Modellbildung sowie Verifikation und Validierung [26]. Aktuelle Ansätze zur Steigerung der Effizienz in Simulationsstudien sind unter anderem in der Entwicklung von anwendungsspezifischen Bausteinkästen [27,28] und Lösungen zur (teil-)automatisierten Erstellung von Simulationsmodellen [26,29,30,31] sowie dem Einsatz von Assistenzsystemen [32,33] zu finden. Mit der Optimierung des Zeitaufwandes können auch die Projektkosten positiv beeinflusst werden. Aufgrund der Nutzung der Simulationsergebnisse zur Entscheidungsunterstützung bei Investitionen, Variantenauswahl und Umplanungen ist die Einhaltung qualitätsbestimmender Faktoren besonders wichtig. Nach Wenzel et al. wird die Qualität von Simulationsprojekten durch eine sorgfältige Projektvorbereitung, die konsequente Dokumentation, eine durchgängige Verifikation und Validierung, die kontinuierliche Integration des Auftraggebers und eine systematische Projektdurchführung bestimmt [1].

Der MBSE-Ansatz und die damit verbundene Systembeschreibung eröffnen mehrere Anwendungsfelder, in denen auf die Faktoren Bearbeitungszeit, Kosten sowie Qualität eines Simulationsprojekts optimierend eingewirkt werden kann. Zunächst sind diese Bereiche zu identifizieren (siehe Abbildung 2).

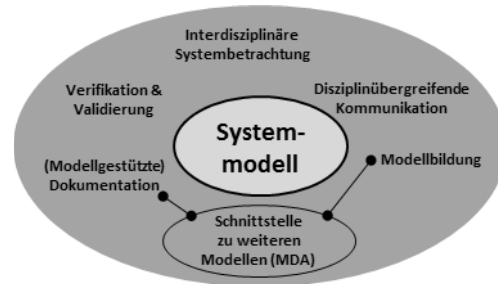


Abbildung 2. Unterstützungsbereiche des Systemmodells

An der Durchführung einer Simulationsstudie sind in der Regel immer mehrere Akteure beteiligt [1,25,32]. Der Erfolg einer Simulationsstudie wird maßgeblich durch die Zusammenarbeit zwischen den Projektbeteiligten beeinflusst. Die Kommunikation und das Erzielen eines gemeinsamen Verständnisses bezüglich der Problemstellung, dem System und dem Simulationsmodell spielen hierbei eine entscheidende Rolle. Missverständnisse können aufgrund der Unterschiede in den Denk- und Begriffswelten der verschiedenen Fachbereiche (Management, Systemexperten, Simulationsexperten) auftreten und zu kostspieligen Verzögerungen, sinkender Akzeptanz und dem Anzweifeln der Glaubwürdigkeit des Simulationsmodells führen. MBSE stellt dem Projektteam mit dem Systemmodell ein Ausdrucksmittel zur Verfügung, das jede Fachdisziplin gleichermaßen versteht. Die höhere Abstraktionsebene des Systemmodells reduziert die notwendigen Informationen auf das erforderliche Mindestmaß und ermöglicht die Diskussion von Sachverhalten ohne die Gefahr in fachspezifische Details abzugleiten und den Überblick zu verlieren. Die Verknüpfung von Anforderungen, Struktur und Verhalten hilft beim Erkennen von Zusammenhängen und bei der Nachvollziehbarkeit von Vorgehen und Entscheidungsfindung in der Simulationsstudie.

Im Mittelpunkt des Informationsaustauschs und der gegenseitigen Abstimmung zwischen den Projektbeteiligten stehen nach Alt die Anforderungen, die Systemstruktur (Architektur) und das Systemverhalten [11]. Die Anforderungen werden in Textform be-

schrieben und können sowohl funktionaler als auch nichtfunktionaler Art sein [11]. Funktionale Anforderungen beschreiben das Leistungsvermögen eines Systems. Nichtfunktionale Anforderungen stehen indirekt mit den zu realisierenden Funktionen in Beziehung und stellen unter anderen Qualitätsanforderungen sowie Vorgaben für Normen, Entwicklung, Kosten und Zeit dar. Im Bezug zu einer Simulationsstudie können funktionale und nichtfunktionale Anforderungen, auch unter Beachtung der Einflussfaktoren aus Projektsicht, als Ergebnis aus der Zielbeschreibung, Problemanalyse und Aufgabenspezifikation abgeleitet werden. Die Anforderungen entstehen aus der zwischen Auftraggeber, System- und Simulationsexperten abgestimmten detaillierten Ziel- und Aufgabenbeschreibung, die das zu lösende Problem beschreibt, den Untersuchungsbereich eingrenzt sowie Zeit- und Kostenrahmen des Projekts festlegt.

Weiterhin besteht zwischen den spezifizierten Anforderungen (der Simulationsstudie) und der Systemstruktur ein enger Zusammenhang [11]. Die aus den Anforderungen resultierende Abgrenzung des zu untersuchenden Systems und seiner Teilsysteme liefert die erforderlichen Informationen zu dessen Struktur für die Entwicklung des Simulationsmodells. Anhand der Systemstruktur werden die Systemelemente, deren Beziehungen untereinander und die Schnittstellen zur Umwelt detailliert beschrieben. Sie stellt jedoch nur eine statische Betrachtung des zu untersuchenden Systems dar und berücksichtigt keine zeitlichen Veränderungen. Umfassende Kenntnisse über das Systemverhalten sind für die Erstellung eines ablauffähigen Simulationsmodells eine fundamentale Voraussetzung. Dies betrifft insbesondere die stattfindenden Prozesse, mögliche Systemzustände, die gegenseitige Beeinflussung der Systemelemente sowie die Transformation der Eingangs- und Ausgangsgrößen. Sowohl die Struktur als auch das Verhalten eines Systems werden in der Phase „Systemanalyse“ untersucht und im Konzeptmodell beschrieben. In dieser Phase ist die Abstimmung zwischen den Projektpartnern besonders wichtig, aufgrund der Festlegung der im Simulationsmodell abzubildenden Komponenten und deren Detaillierungsgrad. Die Überführung des Konzeptmodells in das Systemmodell erzwingt eine Formalisierung entsprechend der verwendeten Modellierungssprache und führt automatisch zum zweiten Schritt der Modellbildung, der „Modellformalisierung“. Mit der Formalisierung des Konzeptmodells werden die Systemstruktur, die Komponenten sowie die Ablauflogiken eindeutig und

präzise definiert. Beispielweise wird die Benennung und Parametrisierung der Systemelemente in Abstimmung mit den verantwortlichen Projektbeteiligten festgelegt und im Systemmodell dokumentiert. Mit Hilfe von Verhaltensdiagrammen finden Ablauflogiken eine grafische Repräsentation im Systemmodell. Der Zwang zur Formalisierung verhindert das häufige Überspringen dieses Bearbeitungsschritts und führt zur Qualitätssteigerung aufgrund der Überprüfbarkeit von formaler Modellbeschreibung und Implementierung in der Verifikation und Validierung. Die Modellformalisierung bereitet die Implementierung vor und ist Voraussetzung für die Ableitung weiterer Modelle gemäß des MDA-Ansatzes.

Die Implementierung beinhaltet die Entwicklung eines ausführbaren Modells im Simulationswerkzeug. Sie stellt den Übergang von der interdisziplinären zur disziplinspezifischen Modellierung dar. Der Simulationsexperte übersetzt hierfür die formale Beschreibung in ein Simulationsmodell. Neben diesem klassischen Weg bietet die Generierung eines Simulationsmodells aus der Systembeschreibung heraus ein enormes Einsparpotential hinsichtlich der benötigten Entwicklungszeit. Die Umsetzbarkeit wurde im Ansatz bereits durch existierende Forschungsarbeiten nachgewiesen [22,23]. Ein weiterer Vorteil liegt in der vom Simulationswerkzeug unabhängigen formalen Beschreibung des Systems. Mit werkzeugspezifischen Transformationsregeln können daraus Simulationsmodelle in verschiedenen Simulationswerkzeugen erstellt werden.

Das Ziel von Verifikation und Validierung (V&V) besteht darin, die Glaubwürdigkeit des Modells nachzuweisen, Fehler möglichst frühzeitig zu erkennen und zu verhindern, dass aus einer Simulationsstudie fehlerhafte Aussagen und Schlüsse gezogen werden sowie das die erarbeiteten Zwischenergebnisse korrekt in die weitere Modellbildung eingehen [25]. Hierfür ist eine konsequente durchgängige V&V während der gesamten Simulationsstudie erforderlich. Anhand der im Systemmodell enthaltenen Informationen über Anforderungen, Struktur und Verhalten können Tests durchgeführt werden. Rabe et al. nennen eine Vielzahl an Verifizierungs- und Validierungstechniken für die Überprüfung der einzelnen Phasenergebnisse [25]. Das Systemmodell unterstützt Techniken, bei denen die Prüfung durch die Zusammenarbeit mehrerer Personen erfolgt (z.B. Strukturiertes Durchgehen, Validierung im Dialog) oder

Auswirkungen von Änderungen untersucht werden (z.B. Sensitivitätsanalyse).

Die durchgängige Dokumentation aller Aktivitäten in einem Simulationsprojekt stellt ein wesentliches Qualitätsmerkmal dar [1]. Das Systemmodell und die darin enthaltenen Informationen können die gesamte Dokumentation einer Simulationsstudie nicht abdecken, jedoch im Bereich der Zielbeschreibung und Aufgabendefinition sowie der Modellbildung vereinfachen. Das Systemmodell dokumentiert darüber hinaus Änderungen durch Abstimmungen und Entscheidungen der Projektbeteiligten im Verlauf der Simulationsstudie. Ein weiterer interessanter Punkt ist die zentrale Verwaltung der Informationen in einem Modell und die damit verbundene Reduzierung der benötigten Dokumente.

Das Systemmodell steht im Mittelpunkt der Durchführung einer Simulationsstudie. Dabei fungiert es als Kommunikationsplattform, die durch Visualisierung anhand grafischer Modelle Transparenz und ein übereinstimmendes Verständnis schafft. Es steigert die Effizienz in der überfachlichen Zusammenarbeit der Experten und Entscheidungsträger und hat das Potential den erforderlichen Zeitaufwand, insbesondere in der Modellbildung, deutlich zu senken. Dadurch können auch die Projektkosten verringert werden. Die Unterstützung der Dokumentation und V&V-Aktivitäten dienen dem Bestreben nach einer hohen Qualität in Simulationsprojekten. Die Einordnung des Systemmodells in das Vorgehensmodell ist in der folgenden Abbildung dargestellt.

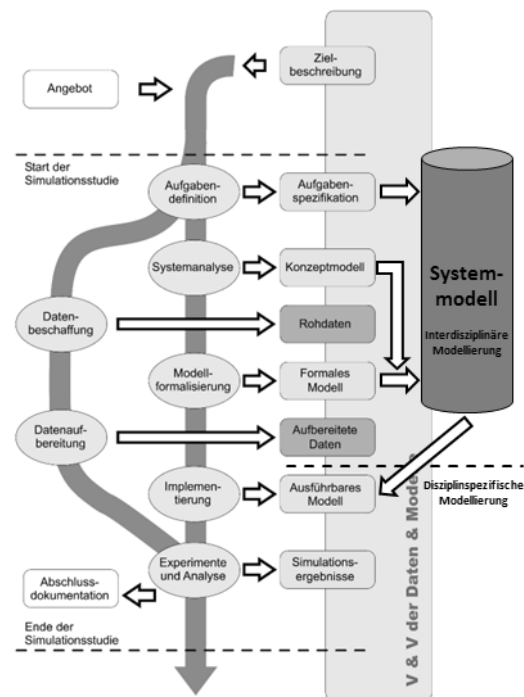


Abbildung 3. Einordnung des Systemmodells in das Vorgehensmodell

5 Zusammenfassung und Ausblick

In diesem Beitrag wird die Anwendung des Model-Based Systems Engineering zur Unterstützung von Simulationsprojekten vorgestellt. Der Einsatz eines Systemmodells bietet gegenüber der bisherigen nur teilweise modellbasierten Bearbeitung von Simulationsstudien eine Reihe von Vorteilen. Unter anderem können die Dauer und die Kosten sowie die Qualität von Simulationsprojekten optimiert werden. Neben der planungsbegleitenden Simulation stellt die betriebsbegleitende Simulation ein weiteres Anwendungsfeld für Systemmodelle dar.

6 Literatur

- [1] S. Wenzel, M. Weiß, S. Collisi-Böhmer, H. Pitsch, O. Rose. *Qualitätskriterien für die Simulation in Produktion und Logistik*. Springer-Verlag, 2008.
- [2] T. Weikiens. *Zukunftsdisziplin Modellbasiertes Systems Engineering*. Wissenschaftsforum Intelligente Technische Systeme 2011, Heinz-Nixdorf-Institut Paderborn, 2011.

- [3] P. Winzer. *Generic Systems Engineering: Ein methodischer Ansatz zur Komplexitätsbewältigung*. Springer-Verlag, 2013.
- [4] G. Ropohl. *Allgemeine Technologie. Eine Systemtheorie der Technik*. Karlsruhe Universitätsverlag, 2009.
- [5] INCOSE. *Systems Engineering Handbook*. V3.2.2. 2011.
- [6] R. Haberfellner, P. Nagel, M. Becker, A. Büchel, H. von Massow. *Systems Engineering: Methodik und Praxis*. Verlag Industrielle Organisation, 2002.
- [7] R. Haberfellner, S. Vössner, O. de Weck, E. Fricke. *Systems Engineering. Grundlagen und Anwendung*. Orell Füssli Verlag, 2012.
- [8] INCOSE. *Systems Engineering Vision 2020*. 2007.
- [9] VDI 3633. *Simulation von Logistik-, Materialfluß- und Produktionssystemen*. Blatt 1, 2010.
- [10] H. Stachowiak. *Allgemeine Modelltheorie*. Springer-Verlag, 1973.
- [11] O. Alt. *Modellbasierte Systementwicklung mit SysML*. Carl Hanser Verlag, 2012.
- [12] M. Eigner, T. Gilz, R. Zafirov. *Interdisziplinäre Produktentwicklung – Modellbasiertes Systems Engineering*. In PLMportal: Positionen aus Wissenschaft und Forschung, 2012.
- [13] L. Kaiser. *Methoden, Modellierungssprachen und Werkzeuge für das Model-Based Systems Engineering*. Vortrag auf dem 3DEXPERIENCE Customer Forum, Mannheim. 2013.
- [14] M. Eigner, T. Gilz, R. Zafirov. *Modellverwaltungsplattformen im modellbasierten Systems Engineering*. In: Tag des Systems Engineering. Zusammenhänge erkennen und gestalten. Carl Hanser Verlag, 2013.
- [15] R. Stark, G. Beier, A. Figge, T. Wöhler. *Domänenübergreifende Abhängigkeiten – Mehrwert durch Modellverknüpfung in der Systementwicklung*. In: Software und System Engineering Essentials (SEE) 2010.
- [16] L. Walker. *The State of Model Based Systems Engineering*. 2011.
- [17] T. Weilkiens. *Systems Engineering mit SysML/UML – Modellierung, Analyse, Design*. dpunkt-Verlag, 2008.
- [18] D. Dori. *Object-process methodology – A holistics systems paradigm*. Springer-Verlag, 2002.
- [19] J. A. Estefan. *Survey of Model-Based Systems Engineering (MBSE) Methodologies*. In: INCOSE MBSE Initiative, California, 2008.
- [20] J. G. Lamm, T. Weilkiens. *Funktionale Architekturen in SysML*. In: Tag des Systems Engineering. Carl Hanser Verlag, 2010.
- [21] O. Schönherr, O. Rose. *Modellierung mit SysML zur Abbildung von Produktionsprozessen*. In: Integrationsaspekte der Simulation: Technik, Organisation und Personal. KIT Scientific Publishing, 2010.
- [22] O. Batarseh, L.F. McGinnis. *System modeling in SysML and system analysis in Arena*. In: Proceedings of the 2012 Winter Simulation Conference. 2012.
- [23] O. Schönherr, J.H. Moss, M. Rehm, O. Rose. *A free simulator for modeling production systems with SysML*. In: Proceedings of the 2012 Winter Simulation Conference. 2012.
- [24] P. Iwanek, L. Kaiser, R. Dumitrescu, A. Nyßen. *Fachdisziplinübergreifende Systemmodellierung mechatronischer Systeme mit SysML und CONSENS*. In: Tag des Systems Engineering 2012. Carl Hanser Verlag, 2013.
- [25] M. Rabe, S. Spieckermann, S. Wenzel. *Verifikation und Validierung für die Simulation in Produktion und Logistik*. Springer Verlag, 2008.
- [26] J. Baier, R. Krieg. *Automatisierter Modellaufbau für Materialflusssimulationen in der Nutzfahrzeugproduktion*. In: Advances in Simulation for Production and Logistics Applications. Fraunhofer IRB Verlag, 2008.
- [27] G. Mayer, C. Pöge. *Auf dem Weg zum Standard – Von der Idee zur Umsetzung des VDA Automotive Bausteinkastens*. In: Integrationsaspekte der Simulation: Technik, Organisation und Personal. KIT Scientific Publishing, 2010.
- [28] H. Kestenbaum, I. Habenicht, A. Manz. *Anwendung von Materialflusssimulation in der*

- Solarindustrie*. In: 14. "Tage des Betriebs- und Systemingenieurs - TBI'11" Nachhaltigkeit in Fabrikplanung und Fabrikbetrieb, 2011.
- [29] S. Bergmann, A. Fiedler, S. Straßburger. *Generierung und Integration von Simulationsmodellen unter Verwendung des Core Manufacturing Simulation Data (CMSD) Information Model*. In: Integrationsaspekte der Simulation: Technik, Organisation und Personal. KIT Scientific Publishing, 2010.
- [30] C. Selke. *Entwicklung von Methoden zur automatischen Simulationsmodellgenerierung*. Dissertation, TU München, 2004.
- [31] S. Jensen. *Eine Methodik zur teilautomatisierten Generierung von Simulationsmodellen aus Produktionssystemen am Beispiel einer Job Shop Fertigung*. Dissertation, Univ. Kassel, 2007.
- [32] B. Bockel, M. Schmitz, S. Wenzel. *Entwicklung von Methodiken zur systematischen Planung von Experimenten in Simulationsstudien*. In: Integrationsaspekte der Simulation: Technik, Organisation und Personal. KIT Scientific Publishing, 2010.
- [33] D. Abel, M. Schmitz, S. Wenzel. *Ansätze zur Unterstützung der Simulation als wissensintensiver Prozess*. In: Industrie Management, 2012, Heft 3.

Konzept zur integrierten Fertigungs- und Logistikplanung im Kontext der digitalen CFK-Produktionsentwicklung

Michael Lütjen¹

¹BIBA - Bremer Institut für Produktion und Logistik

ltj@uni-bremen.de

Die Produktion von Bauteilen aus carbonfaserverstärkten Kunststoffen (CFK) befindet sich aktuell im Wandel von der Werkstatt- zur Massenfertigung. Einhergehend ändern sich auch die Bedarfe zur Unterstützung des Produktionsentwicklungsprozesses mittels entsprechender Instrumente und Werkzeuge. Dabei behandelt die integrierte Fertigungs- und Logistikplanung im Rahmen der Digitalen Fabrik ein Themenfeld, welches sich in der CFK-Fertigung durch die Berücksichtigung von Werkzeugumläufen, Autoklavprozessen sowie hoher Planungsunsicherheit kennzeichnet. Dieser Beitrag formuliert ein Modellierungskonzept, welches diese Anforderungen an eine digitale Produktionsentwicklung berücksichtigt und ausgehend von einem zentralen Datenmodell auf die Generierung von Simulationsmodellen abzielt.

1 Einleitung

In diesem Beitrag wird ein Konzept zur integrierten Fertigungs- und Logistikplanung im Kontext der digitalen CFK-Produktionsentwicklung präsentiert und dabei zunächst auf die Aspekte der Industrialisierung unter Materialflusskomplexität eingegangen. Im zweiten Kapitel wird der Stand der Technik zur Generierung von Simulationsmodellen analysiert und dann im dritten Kapitel das eigene Modellierungskonzept GRAMOSA vorgestellt. Das vierte Kapitel zeigt die Ergebnisse der Konzeptevaluierung während im abschließenden fünften Kapitel die Zusammenfassung des Beitrags erfolgt.

1.1 Industrialisierung der CFK-Fertigung

Die zunehmende Industrialisierung der CFK-Fertigungstechnologie stellt große Herausforderungen an die Automatisierungs- und Produktionstechnik. Die bislang manuell eingesetzten Fertigungstechniken müssen automatisiert und in einem Gesamtprozess aufeinander abgestimmt werden. Die Zykluszeiten, Materialkosten und Nacharbeit müssen dabei um 50 bis 90% reduziert werden, um metallische Bauteile wirtschaftlich substituieren zu können. Im Hinblick auf den Automobilbau als neuer Massenmarkt für CFK-Bauteile, ist dieser wesentlich kostenintensiver als der Flugzeugbau, welcher nach wie vor wertmäßig den größten Anteil an CFK-Komponenten nachfragt (Abbildung 1). Die großen Strukturen im Flugzeug- und Automobilbau bedürfen neuer Handhabungs- und Bearbeitungskonzepte, sowohl im ausgehärteten als auch unausgehärteten Zustand sowie neuer Harzsysteme, die schneller und zum Beispiel ohne aufwendige

Autoklavtechnik aushärten. Aktuell machen im Karbonleichtbau die Fertigungskosten 80% der Herstellkosten aus, weshalb im Automobilbau eine Reduzierung um 90% angestrebt und für realistisch gehalten wird, um mittelfristig mit Aluminium oder Stahl konkurrieren zu können [1].

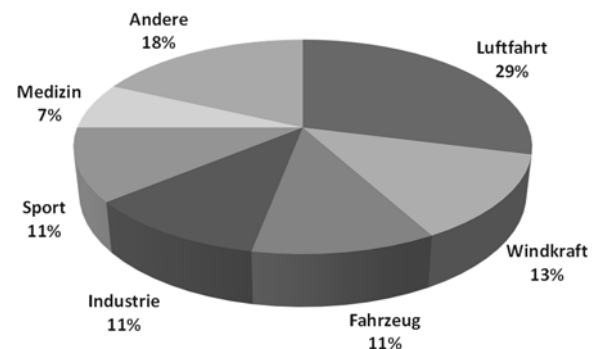


Abbildung 1. Wertmäßiger Anteil der Branchen am CFK-Komponentenmarkt (weltweit) [2]

1.2 Materialflusskomplexität in der CFK-Serienfertigung

Die CFK-Serienfertigung kann grundsätzlich mittels verschiedener CFK-Fertigungsverfahren realisiert werden, wobei die Prepreg-Autoklavtechnik die Herstellung der qualitativ hochwertigsten Bauteile ermöglicht. Das Prepreg ist ein mit Harz vorimprägniertes, textiles Halbzeug, welches mittels Tapelegern (TPL) oder Advanced Fibre Placement-Anlagen (AFP) in entsprechende Formschalen eingebracht wird. Ein wesentlicher Vorteil der CFK-Fertigungstechnologie ist

die Möglichkeit zur Integralbauweise, die dem Laminataufbau neben verschiedenen Hilfsstoffen (Vakuumfolie, Belüftungsvlies etc.) und Fertigungshilfsmitteln (Formschalen etc.) noch weitere Strukturelemente (Sandwichmaterialien, CFK-Teile etc.) hinzufügen lässt. Diese Laminataufbauten durchlaufen, zusammen mit der Formschale, den Aushärtungsprozess, um das ausgehärtete CFK-Bauteil dann aus der Formschale zu entnehmen, nachzubearbeiten, zu prüfen und gegebenenfalls erneut nachzubearbeiten. Aus planungstechnischer Sicht ist der Werkzeugumlauf der Formschale ein wesentlicher Faktor, da die Kosten für eine Formschale bei größeren Bauteilen aufgrund einer speziellen Nickellegierung teilweise sechstellig werden können. Da die Aushärtung zumeist einen Autoklav-/Ofenprozess erfordert, der eine parallele Bearbeitung mehrerer Formschalen mit entsprechenden Laminataufbauten vorsieht, entsteht eine hohe Systemdynamik, welche bei der systemtechnischen Auslegung den Einsatz von Simulationstechnik erfordert. Dabei ist durch die Werkzeugumläufe sowie durch weitere Prüf- und Nachbearbeitungsprozesse eine wesentliche Anforderung, dass entsprechende Materialflussstrukturen idealerweise im Rahmen einer Generierung von Simulationsmodellen abgebildet werden können.

2 Ansätze zur Generierung von Simulationsmodellen

Die (teil-)automatische Modellgenerierung ist hinsichtlich des Einsatzes von Simulationstechniken im Rahmen der Digitalen Fabrik seit zehn Jahren Bestandteil wissenschaftlicher Forschung (Abbildung 2). Grundsätzlich kann zwischen planungs- und betriebsbegleitenden Ansätzen unterschieden werden. Während die planungsbegleitenden Ansätze auf eine direkte Dateneingabe durch den Fertigungsplaner angewiesen sind, wird bei den betriebsbegleitenden Ansätzen vornehmlich auf aufbereitete Daten aus entsprechenden betrieblichen Anwendungssystemen zurückgegriffen. Beiden Ansätzen ist jedoch die zentrale Frage nach der Strukturierung der Daten im Hinblick auf eine Generierung von Simulationsmodellen gemein, weshalb eine gemeinsame Analyse vorgenommen werden kann. Ciupek beschreibt einen planungsbegleitenden Ansatz für Demontagefabriken [3], während Selke einen betriebsbegleitenden Ansatz zur Fertigungssteuerung entwickelt, der auf die Erkennung von Steuerungsmustern in Betriebsdaten ausgerichtet ist [4]. Zenner konzentriert sich im Rahmen der Mon-

tageplanung auf die Modellierung alternativer Prozessabläufe für verschiedene Produkt- und Prozessvarianten [5]. Jensen beschreibt einen datenbankorientierten Ansatz zur Fusion eines Simulationsmodells aus mehreren verschiedenen Datenquellen [6], während Schedlbauer im Rahmen der Logistikplanung ein prozessorientiertes Baustein-konzept entwickelt hat, welches erste Ansätze zur Modellgenerierung bietet [7]. Weig präsentiert ein Konzept für ein integriertes Risikomanagement bei Fabrikplanungsprojekten, wobei er im Besonderen auf den Umgang mit unsicheren Prozess- und Fertigungsparametern eingeht [8]. Hinsichtlich der graphischen Modelle präsentieren Brüggemann et al. ein Konzept zur Generierung von Simulationsmodellen basierend auf der Methode des Wertstromdesigns [9]. Rooks entwickelt ein datenbankorientiertes Konzept zur Integration der Materialflussplanung in die Ablaufsimulation [10]. Kapp entwickelt einen unternehmensneutralen Ansatz zur betriebsbegleitenden Simulation, wobei er bestehende Datenmodelle um verschiedene Planungs- und Steuerungsobjekte sowie Organisationselemente erweitert [11].

<i>Jahr</i>	<i>Autor (Anwendungsgebiet)</i>
2004	Ciupek (Demontageplanung)
2004	Selke (Fertigungssteuerung)
2006	Zenner (Montageplanung)
2007	Jensen (Fertigungssteuerung)
2008	Schedlbauer (Logistikplanung)
2008	Weig (Fertigungsplanung)
2008	Brüggemann et al. (Logistikplanung)
2009	Rooks (Logistikplanung)
2011	Kapp (Fertigungssteuerung)
2011	Schönherr et al. (Fertigungsplanung)
2014	Bergmann (Fertigungssteuerung)

Abbildung 2. Chronologie der Ansätze zur (teil-) automatischen Modellgenerierung

Schönherr et al. realisieren einen graphischen Modellierungsansatz auf Basis von SysML, welcher die Modellierung von Produktionssystemen mit sequenziellen Prozessfolgen erlaubt [12]. Bergmann präsentiert einen betriebsbegleitenden Modellgenerierungsansatz, welcher unter Erweiterung von CMSD die Nutzung

verschiedener Simulatoren zulässt [13]. In Abgrenzung zu den bestehenden Ansätzen erfordert die Planung einer CFK-Serienfertigung die integrierte Abbildung von Fertigungs- und Logistikprozessen unter Berücksichtigung einer hohen Materialflusskomplexität. Die zentrale Herausforderung ist dabei, die verschiedenen Werkstück-, Material-, Ladungsträger- und Werkzeugflüsse so zu modellieren, dass eine Generierung von Simulationsmodellen erfolgen kann. Bislang wurde sich in der Forschung auf sequenzielle Prozessfolgen ohne explizite Darstellung von Ladungsträger- und Werkzeugflüssen konzentriert, was in der CFK-Serienfertigung aufgrund des hohen Werkzeugkostenanteils notwendig ist.

3 Modellierungskonzept GRAMOSA

Das Modellierungskonzept zur graphischen Modellierung und simulationstechnischen Analyse (GRAMOSA) beinhaltet acht Modellebenen, die sukzessiv aufeinander aufbauen und die Ausgestaltung des Produktionssystems einschließlich der Beschreibung des Simulationsexperiments beinhalten (Abbildung 3). Übergeordnet gliedern sich die Modellebenen in die Phasen des Managements, der Fertigungsplanung und der Logistikplanung.

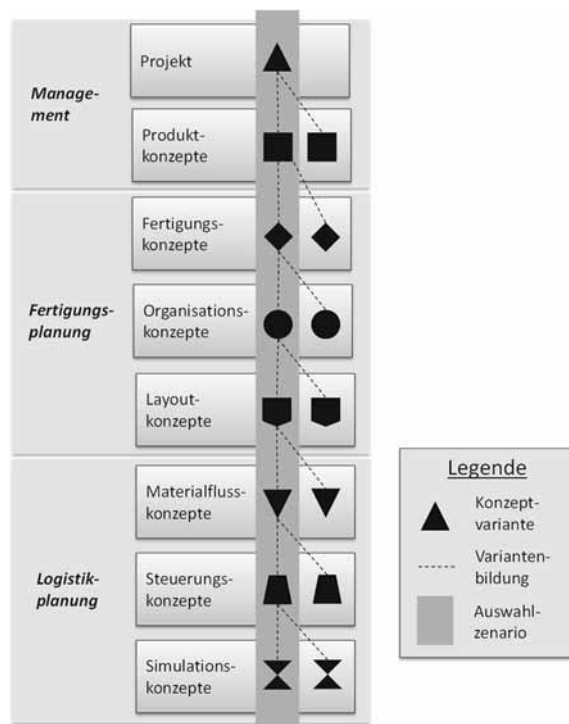


Abbildung 3. Modellaufbau und Variantenbildung in GRAMOSA [14]

Ausgehend von der Projektebene erfolgt die Variantenbildung, wobei es sich um die Bildung von Planungsvarianten handelt, die nach dem Variantenprinzip der Planung anfallen und mögliche Realisierungsoptionen beschreiben. Dabei werden sämtliche Informationen und Spezifikationen an die jeweils darunterliegende Modellebene vererbt. Dadurch wird es möglich, sehr gezielt einzelne Gestaltungsoptionen als Planungsvarianten zu definieren und zu untersuchen. Die Kombination der Planungsvarianten über alle acht Modellebenen wird im Rahmen dieses Beitrags als Produktionsszenario bezeichnet. Einhergehend mit dem Vorgehensmodell wurde ein entsprechendes Fabrikdatenmodell entwickelt (Abbildung 4), welches die Datenobjekte der einzelnen Modellebenen definiert. Wichtig hierbei ist die initiale Definition von Fertigungselementen auf Projektebene. Ein Fertigungselement repräsentiert dabei vornehmlich ein physisches Objekt als Bestandteil eines größeren Fertigungssystems. Dies kann eine Maschine, ein Werkstück, ein Werker, ein Werkzeug oder sonstiges sein. Im Rahmen der Logistikplanung können diese Fertigungselemente dann zu Materialflussobjekten zusammengefasst werden, um zum Beispiel Transporte von mehreren Werkstücken in einem Ladungsträger abbilden zu können. Die Modellebenen sowie das Fabrikdatenmodell wurde so gestaltet, dass eine effiziente Verwaltung von Planungsvarianten und Produktionsszenarien sowie die Generierung von Simulationsmodellen möglich werden.

4 Anwendung des Modellierungskonzepts

Das Modellierungskonzept GRAMOSA wurde prototypisch implementiert und ein Vorgehensmodell zur Anwendung entwickelt (Abbildung 5). Beginnend mit einem Projektauftrag werden die zu untersuchenden Planungsvarianten angelegt und mit der Modellbildung gestartet. Die Modellbildung beschreibt die sukzessive Ausgestaltung der einzelnen Modellebenen, wobei im Softwareprototyp eine Kombination aus Baumdiagramm und Listensicht gewählt wurde, um eine hohe Informationsdichte zu erreichen. Zur graphischen Modellvisualisierung wurde eine domänenspezifische Beschreibungssprache auf Basis von UML-Notationselementen entwickelt, die sowohl prozessuale als auch materialflusstechnische Zusammenhänge abbildet. Parallel dazu besteht auch die Möglichkeit zur Generierung eines Simulationsmodells, wobei als Simulator TECNOMATIX Plant Simulation gewählt und eine eigene Simulationsumgebung zur separaten Abbildung von Material- und Steuerungsfluss geschaffen wurde.

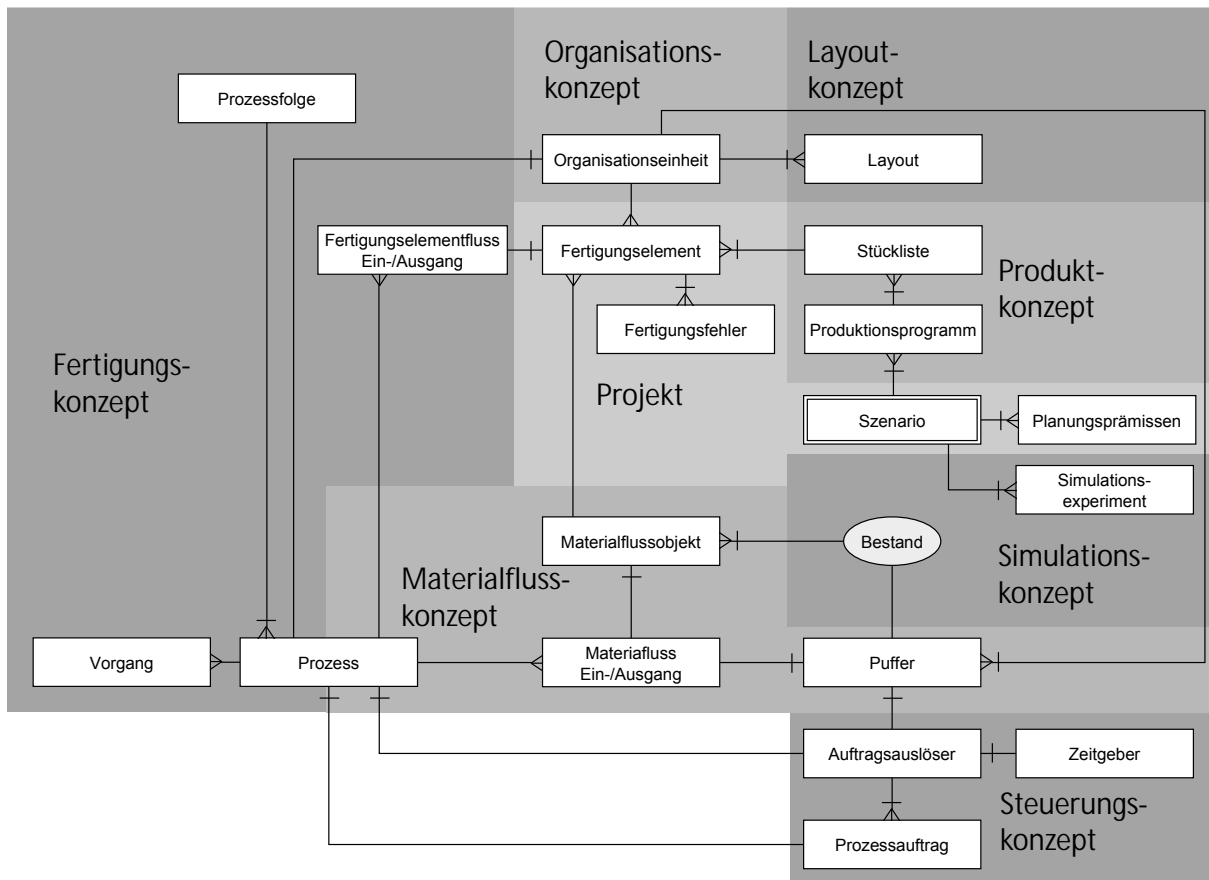


Abbildung 4. Fabrikdatenmodell von GRAMOSA [14]

Für das Steuerungskonzept wurde ein regelbasierter Ansatz genutzt, welcher verschiedene Bedingungen für eine Auftragsauslösung und die eigentliche Auftragsdurchführung definiert. Des Weiteren besteht die Möglichkeit die Ergebnisse einer Prozess-FMEA hinsichtlich des Auftretens und der Entdeckung von Fertigungsfehlern zu integrieren. Durch eine automatisierte Experimentdurchführung mit anschließender Sensitivitätsanalyse kann das Modellverhalten effizient validiert und analysiert sowie das Produktionsszenario bewertet werden.

5 Evaluierung

Das Modellierungskonzept wurde in drei Stufen evaluiert. In der ersten Stufe wurden 18 grundlegende Materialfluss- und Steuerungsfunktionen mittels des Modellierungskonzepts abgebildet. Hierzu zählten verschiedene Sortierfunktionen für Bauteilqualitäten, Materialflussobjekte sowie verschiedenste Verteil- und

Sammelfunktionen. Des Weiteren können zum Beispiel bestandsabhängige Steuerungsverfahren wie KANBAN und CONWIP dargestellt werden. In einer zweiten Stufe wurde ein Anwendungsbeispiel aus der CFK-Serienfertigung zur Herstellung einer Rumpfschale modelliert und simuliert, um die allgemeine Praxistauglichkeit zu zeigen. In einem dritten Schritt wurde ein 3x3-Maschinenmodell als Referenzszenario für einen Vergleich mit einer rein manuellen Implementierung abgebildet. Hierbei zeigte sich, dass die Implementierung in GRAMOSA unter zwei Stunden realisiert werden konnte, während für die manuelle Implementierung direkt in TECNOMATIX Plant Simulation von acht verschiedenen Simulationsexperten über 10 Stunden veranschlagt wurden. Entsprechend lassen sich für variantenarme Serienfertigungen deutlich Vorteile beim Einsatz des Modellierungskonzepts erkennen. Die Grenzen des Modellierungskonzepts zeigen sich hingegen in der Abbildung von Einzel- bzw. Werkstattfertigungen mit einer großen Anzahl unterschiedlicher Produkte und Aufträge. Das gleiche gilt für variantenreiche Serienfertigungen. Das Problem hierbei ist die produktspezifische Modellierung von Fertigungs- und Transportprozessen, die zu einem

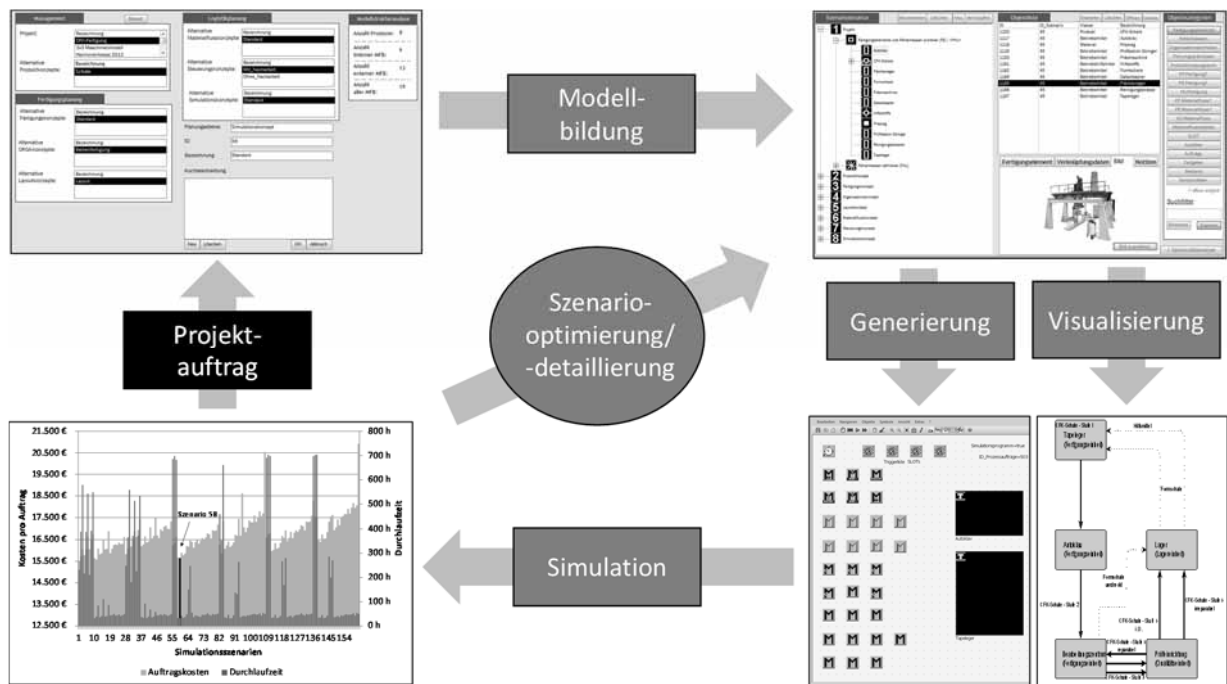


Abbildung 5. Vorgehensweise bei der Anwendung von GRAMOSA zur Produktionsentwicklung

recht erheblichen Modellierungsaufwand führt, da jeder Transportprozess produktspezifisch abgebildet wird.

6 Zusammenfassung und Ausblick

Die (teil-)automatische Modellgenerierung ist ein wesentlicher Faktor, um die Verbreitung von Simulationstechnik zur Planung und Analyse von Produktionssystemen nachhaltig zu steigern. Durch die Modellgenerierung reduzieren sich manuelle Implementierungszeiten und die Modellqualität verbessert sich aufgrund standardisierter EDV-Schnittstellen. Nachteilig müssen die geringere Modellflexibilität sowie der Aufwand zur Erstellung der Schnittstellen bewertet werden. Mit dem Modellierungskonzept GRAMOSA wurde ein flexibles Instrument zur Unterstützung der Fertigungs- und Logistikplanung im Rahmen der digitalen Produktionsentwicklung geschaffen. Vor dem Hintergrund der CFK-Technologieentwicklung wurden Möglichkeiten zur Verwaltung und Analyse alternativer Produktionsszenarien geschaffen. Hierzu wurden acht Modellebenen definiert, die sukzessiv im Rahmen der Planung ausgestaltet werden und die die systematische Entwicklung von Fertigungs- und Lo-

gistikprozessen sowie entsprechender Systemstrukturen erlauben. Durch Aufhebung starrer sequenzieller Prozessfolgen wird zum Beispiel auch die Abbildung von Werkzeugumläufen sowie die Abbildung weiterer Prüf- und Nachbearbeitungsprozesse möglich. Bislang reduziert sich das Modellierungskonzept auf die effiziente Planung variantenarmer Serienfertigungen. In einem nächsten Schritt sollen Ansätze zur Berücksichtigung von Produktvarianten entwickelt werden, um den Modellierungsaufwand für diese Art von Produktionsformen weiter zu reduzieren. Des Weiteren sollen diese Ansätze auch zur effizienten Modellierung und Simulation von Kommissionierprozessen verwendet werden.

7 Referenzen

- [1] P. Trechow, *MATERIALFORSCHUNG* - „Um die breite Kundschaft zu überzeugen, ist Carbon noch zu teuer“, VDI-Nachrichten (2014) 13, S. 16.
- [2] A. Schuster, *CFK-Markt: Europa auf dem Weg zur Spitze*, MM Compos. World (2010) 10, S. 10-12.
- [3] M. Ciupek, *Beitrag zur simulationsgestützten Planung von Demontagefabriken für Elektro- und Elektronikaltgeräte*, Dissertation, Technische Universität Berlin, 2004.

- [4] C. Selke, *Entwicklung von Methoden zur automatischen Simulationsmodellgenerierung*, Dissertation, Technische Universität München, 2004.
- [5] C. Zenner, *Durchgängiges Variantenmanagement in der technischen Produktionsplanung*, Dissertation, Universität des Saarlandes, 2006.
- [6] S. Jensen, *Eine Methodik zur teilautomatisierten Generierung von Simulationsmodellen aus Produktionsdatensystemen am Beispiel einer Job Shop Fertigung*, Dissertation, Universität Kassel, 2007.
- [7] M.J. Schedlbauer, *Adaptive Logistikplanung auf Basis eines standardisierten, prozessorientierten Bausteinkonzepts*, Dissertation, Technische Universität München, 2008.
- [8] S. Weig, *Konzept eines integrierten Risikomanagements für die Ablauf- und Strukturgestaltung in Fabrikplanungsprojekten*, Dissertation, Technische Universität München, 2008.
- [9] H. Brüggemann, P. Müller, *Digitales Wertstromdesign - Digital Value Stream Mapping* in: M. Rabe (Ed.), *Adv. Simul. Prod. Logist. Appl.*, Fraunhofer IRB Verlag, Stuttgart, 2008, S. 575–584.
- [10] T. Rooks, *Rechnergestützte Simulationsmodellgenerierung zur dynamischen Absicherung der Montagelogistikplanung bei der Fahrzeugneutypplanung im Rahmen der Digitalen Fabrik*, Dissertation, Technische Universität Clausthal-Zellerfeld, 2009.
- [11] R. Kapp, *Ein betriebsbegleitendes Fabriksimulationssystem zur durchgängigen Unterstützung der kontinuierlichen Fabrikadaption*, Dissertation, Universität Stuttgart, 2011.
- [12] O. Schönherr, O. Rose, *A General Model Description for Discrete Processes* in: S. Jain, R.R. Creasey, J. Himmelsbach, K.P. White, M. Fu (Eds.), *Proc. 2011 Winter Simul. Conf.*, Baltimore, Maryland, USA, 2011, S. 2206–2218.
- [13] S. Bergmann, *Automatische Generierung adaptiver Modelle zur Simulation von Produktionssystemen*, Dissertation, Universität Ilmenau, 2014.
- [14] M. Lütjen, *Modellierungskonzept zur integrierten Planung und Simulation von Produktionsszenarien entwickelt am Beispiel der CFK-Serienfertigung*, Universität Bremen, Dissertation, 2014.

Using simulation to improve cluster tool performance in an LED manufacturing line

Fabian Suhrke¹, Falk S. Pappert², Robert Janik¹, Oliver Rose²

¹OSRAM Opto Semiconductors
Leibnizstraße 4
Regensburg, D-93055, Germany

²Universität der Bundeswehr München
Institut für Technische Informatik
Fakultät für Informatik
Neubiberg, D-85577, Germany

Fabian.Suhrke@osram-os.com, Falk.Pappert@unibw.de
Robert.Janik@osram-os.com, Oliver.Rose@unibw.de

Cluster tools play an increasingly important role in modern opto-semiconductor fabs. These tools are more complex for scheduling and even for planning than normal single wafer processing tools and have a large impact on fab performance. In this paper, we discuss differences between logistics of common semiconductors and opto-semiconductors with respect to cluster tool logistics and present a structured approach to simulate the dynamic behavior of a sputter cluster work center regarding the optimization of its performance.

1 Introduction

A cluster tool is defined in SEMI Standard E21-96 as an integrated, environmentally isolated manufacturing system consisting of process, transport, and cassette modules mechanically linked together.

As discussed in Semiconductor International some of the main factors that have motivated cluster tools are increased yield and throughput, reduced contamination, better floor space utilization, and less human intervention [1].

Furthermore the Strategic Future Fab Study (SFFS) found that the most cost-efficient and profitable fab they evaluated relied heavily on cluster tools [2]. For that very reason cluster tools are important for (opto-) semiconductor manufacturing. Due to the high cost associated with sputter cluster tools, they need to be run at high levels of utilization. However, considering their complexity and possible variety in configurations, careful considerations are necessary for capacity and investment planning.

Since cluster tools are complex logistical systems, simulation helps to predict how the tool behaves and as a result to optimize their performance and application. In this industrial case study we show a modeling approach for a sputter cluster work center with multiple Oerlikon CLN200 cluster equipment.

Sputtering is a method of depositing thin films by knocking out atoms from a solid target material which then deposit on the wafer surface.

“The sputter cluster tool is used for sputter deposition, also known as physical vapor deposition, or PVD, a widely used technique for depositing thin metal layers on semiconductor wafers” [3].

Sputter cluster deposition is a complex matter, therefore we would like to refer the interested reader to [4] and only give a very brief introduction on the logistics of our tool. This is shown in Figure 1.

The six single-wafer process chambers and the load locks in which wafers enter and exit the tool are organized around a central chamber with a robot handler. Wafers are taken from the cassette within the loadport and can then undergo several process steps in different process chambers, depending on recipe. After all wafers of a cassette are processed they return to the cassette and will move on to the next tool. At one time up to two cassettes, where each may have a different recipe, can be loaded into such a cluster tool. This leads to a job shop behavior which is typical for cluster tools. In the next chapter we review the literature of cluster tool simulation methods within semiconductor manufacturing. We then motivate the development of a simulation based approach in comparison to the previously applied methods. In Chapter 3 we focus on some technical and implementation de-

tails of the model development and finally we give a brief overview on use cases and results based on the created system.

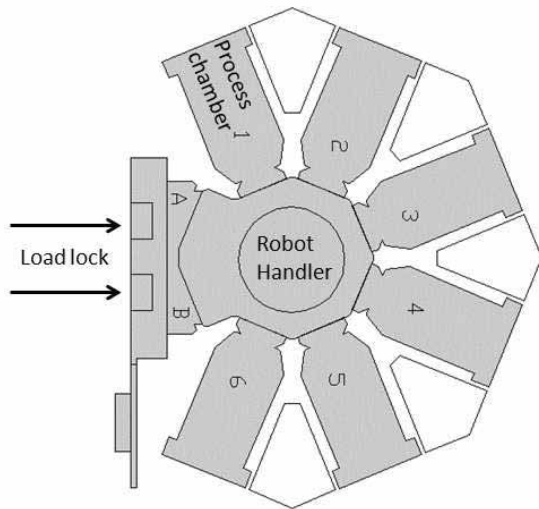


Figure 1. Sputter cluster equipment with six process chambers, numbered from 1 to 6, two load ports A and B and a robot handler.

2 Literature overview

Semiconductor manufacturing is a complex but well researched topic. The main focus of current research lies on improving systems with new or improved methods for scheduling and optimizing workflow within classic productions. Furthermore, internal and external logistics of cluster tools as well as optimization approaches in classic semiconductor manufacturing are widely covered by the existing literature. An overview on the general topic of semiconductor simulation is given by Mönch et al. [5].

Looking at the area of opto semiconductor manufacturing, production is still in a different stage of development. With increasing importance of LED technology, companies producing opto-semiconductors now face similar challenges in capacity planning as logic and memory producers faced several years ago. However, with respect to specific characteristics of opto-semiconductor manufacturing there are still open issues. In particular, sputter processes for LED manufacturing differ significantly from silicon-based semiconductor manufacturing mainly with respect to the variety of materials being used, and the complexity of the single processes. In the analyzed work center, eleven different deposition materials are used, with up to five materials deposited per single process. With this increase in complexity new challenges arise and established approaches need to be revisited. Considering target replacements, for example [6] showed the impact of target/shield change procedures on the

performance of sputter cluster tools. This matter significantly worsens within our heterogeneous environment, being only one of many challenges.

We will elaborate on further differences in the following section.

Most modeling approaches for semiconductor cluster tools like in Srinivasan [7], LeBaron [8] and Mauer [9] are ignoring the external environment and operators and have their focus on internal equipment efficiency, so that manual sputter cluster pre- and post-processes like cleaning, weighing wafers and loading have no influence on the work center performance. In this study we want to model not only the internal logistics of the sputter cluster tool but also the external logistics (e. g. operator availability). Accurate internal logistic process simulation implies robot handler deadlock detection, resolution and prevention. Like Zuberek [10] shows Petri nets are an established method for robot handler modeling. However, we will introduce a simplified approach in Chapter 3 since the closed source equipment software includes deadlock management and this paper focuses rather on work center simulation than on the internal logistics of one sputter cluster.

3 Conventional approach and potential of simulation

In most cases of capacity evaluation, a simple calculation including number of tools, uptimes, utilization limits, process speeds and planned wafer starts is sufficient. However, in the discussed cluster tool scenario more variables need to be considered like volume share of processes per machine, equipment-specific process speeds and target exchanges that result from the process mix run on individual machines.

Therefore, the first solution was to enhance the standard spreadsheet calculation model with these additional dimensions. The main disadvantage of this conventional approach was that process shares per machine had to be determined in advance in order to find a feasible solution to produce the planned volume. These pre-determined process shares can differ severely from the actual process shares per machine achieved in reality by following standard dispatch rules. Furthermore, changes in machine or process setup require high manual effort to update the calculation model. Dynamic dependencies like variable downtimes from target exchanges resulting from the actual process mix on a machine have not been considered. Also the influence of certain processes still being able to run even if one or more modules of a cluster tool are down is difficult to model in a static calculation.

While many of these additional input parameters could also be modeled in a spreadsheet calculation, dynamic dependencies as well as support processes and operator availability have a significant influence on throughput and cycle time. Therefore, the goal was to integrate these parameters into a holistic model of the discussed cluster tool workcenter.

Besides creating an accurate capacity model, additional information can be obtained by simulation, like utilization of single chambers and variability added by support resources. From this data, additional productivity potentials can be found.

4 Model development

In this section we will give an overview on the model development. First we will discuss the technologies and methods used with a main focus on model design and maintainability. Then we will have a look at one of our major simplification and give a brief overview on our model calibration efforts and measures to adjust the model behavior to the simplifications we made.

4.1 Technical modelling approach

There are several key concerns when creating new simulation systems, not least of which is time horizon in which the system will be used. There is a significant difference in the necessity to focus on accessibility and maintainability of a model depending on their future usage. If a model is only going to be used for a short lived study, the investment of time and other resources in extendibility and future adaptability will usually be much less than in systems which are going

to be used for years to come. Since the system we set out to build is planned to be used for the foreseeable future we put special focus on designing it in an easy-to-maintain fashion. Many of the approaches described in the following section are very well known technologies in the software development world, but are methods we see rarely employed while building simulation models. The possibility to use these methods depends heavily on the simulation package chosen. In our case we had the opportunity to almost freely choose a simulation package. Apart from common selection criteria like performance, availability of support, previous exposure of team members, graphics, licensing models and of course price, we had a close view on the possibility to import custom user created libraries and the programming or script languages utilized as well as available tools and tool chains to develop in these languages outside of the simulation package.

We decided to use AnyLogic in its current 7th edition. AnyLogic utilizes Java as programming language; in fact all AnyLogic models are basically Java programs. AnyLogic furthermore allows importing almost any Java library.

With the Eclipse IDE there is a powerful free tool available to create custom Java libraries and with consideration of the vast size of its plug-in universe there is almost no limit to technologies and methods available to use in development.

This ability to freely decide to develop functionality within the simulation package or a powerful IDE allows for each component to be developed according to their dependence on features of the simulation package and their complexity.

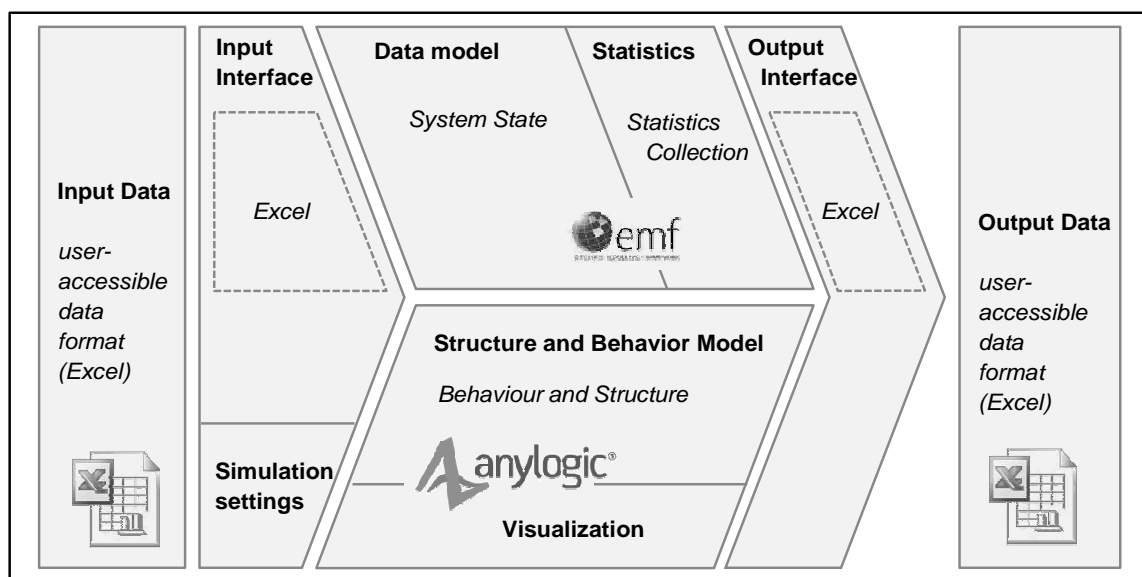


Figure 2. Overview of system modules

Figure 2 shows an abstract overview of the system in its current state. Starting from the left to the right we see all major modules of the system in the order a normal simulation run visits them. The user will design and create a scenario within a spreadsheet file based on a flexible template. This file is then imported using an implementation of the input interface which translates the Excel input model into the data model. This in turn is used to initialize the behavioral and structural model and will represent the systems current state until the end of the simulation. The statistics component is used to collect and aggregate performance data during the run which is then exported through the output interface in a user accessible format for the help in decision making.

Apart from the central components (data model, statistics, and behavioral and structure model) all peripheral systems are designed to be easily replaced in case the user environment changes or future usage demands the utilization of different technologies for model system configuration and result reporting. However, not only the environment of the simulation system may need to change over time, but as with every software system in use over time, changes and new feature are sure to come. To address this we designed our system in a modular fashion where we created a clear separation of system state, behavior and structure. Although modularization is a common approach, there are several other techniques which increase maintainability and even reduce development time. Some of them are explained in more detail in the following specific sections they had the most significance in.

Interfaces

Currently simulation runs are configured using a spreadsheet template. At the end of a simulation run, results are written into another spreadsheet template which aids the user with a semi-automated analysis and evaluation of the run.

With an anticipated shift of simulation activities from capacity planning and scenario evaluation towards daily use and operational decision-making comes the need for online data access and automated model configuration. The currently implemented spreadsheet interfaces are therefore easily replaceable with interfaces using other technologies.

To increase readability and overview on these systems we decided to develop them outside of the simulation package and import them as part of a library. A significant amount of work for this kind of interfaces is usually spent on the actual adapter to write or read data of the external format. This applies even more in consideration of a complex format like Excel spreadsheets. To tackle this issue, there are several possible solutions. First of all, one could develop one's own

interface, but looking at other approaches this is very time and resource consuming and hardly the best way to spend resources. Secondly, there is the possibility to use libraries provided by the community to access and write spreadsheets. These libraries come with some effort in training and sometimes are not in a development stage necessary for productive use there is furthermore the need to keep up to date and maintain some knowledge of any used library. Another approach and the one we chose to use is the use of AnyLogics own Excel connector. Although maybe just an adapted free library in the background we feel it is far less likely that future changes in the Excel format will not be covered by AnyLogic to assure continuous compatibility. The problem with using the AnyLogic connector lies in the fact that it is not available outside of the simulation package. A technology which helps here is the proxy pattern [11]. The proxy pattern is a commonly used pattern in software development, basically to use this pattern for this issue we create an interface in our library which resembles an interface to Excel, this interface we can now freely use within our library. On the other side, within our simulation environment, we create a class implementing the interface and propagating all methods to methods of an AnyLogic connector object. Although this may sound complex it is basically just about four very simple lines of Java code for every method of the connector we wish to use, and therefore a very simple but very effective way to access Excel files.

Data model and statistics module

AnyLogic provides a development environment which offers some support for source code development which is comparable to other tools in the industry. For small models with only a little complexity this is sufficient, but for larger models with the need for significant source code additions we feel there is still some way to go to reach a state where it is comparable to current development environments used in common software development. Therefore we developed the data model outside of AnyLogic and then used AnyLogic's ability to integrate Java libraries into models. This gave us several benefits in the long run; first we are able to utilize the full functionalities of a Java development environment to create our data model, in our case Eclipse. This provides access to common practices like unit testing and automated refactoring. Furthermore we gained access to current software development methodologies drastically reducing development time and effort. Especially the use of Model Driven Development, utilizing the Eclipse Modelling Framework helped us significantly. Using this approach we designed our data model and statistics modules using UML-like diagrams and

then utilized widely used generators to automatically create most of the basic source code. After the code generation we added custom source code using the Generation Gap pattern. Using this pattern offers one important benefit in comparison to other methods of “combining” generated and custom code, the separation of these different types of code into different class files with an inheritance relationship. Generation Gap therefore avoids accidental overwriting custom code while still providing the common development environment during custom code development. This whole approach reduces the amount of handwritten code; this not only reduces time investment in code development but equally reduces future effort for changes. An additional benefit is a reduction in errors within the code. We see this benefit in the use of code generators for the majority of the code. These generators are not necessarily error free, but due to the wide use of them errors in generation are much more unlikely compared to handwritten code which is only used within one system.

Structure and behavior

As in most simulation studies, we identified and defined sub components of the model as shown in Figure 3.

Basically these are the obvious subcomponents of such a system. Each of these components is modeled as an “agent” in AnyLogic. In AnyLogic, “agent” is the term for a component which can be used as part of a model or another component. Using this approach it is quite easy to define the structure and behavior of the system while avoiding unnecessary repetition. In the world of object oriented programming, designing an “agent” translates into creating a

class, which can then be instantiated and used in other classes, which is basically what AnyLogic is doing in the background. In Figure 3, it is furthermore shown which “agent” implements which kind of behavior. We strictly followed a convention of implementing features only and completely within one single “agent”. In the case of interlocking features of different levels AnyLogic provides a very useful feature we will refer to as action parameters. Action parameters can be compared to triggers in other simulators but can be created easily by the modeler. From a modelers point of view these are basically parameters which are visible in the properties of an “agent”. Functionality entered in such a parameter will be executed when the parameter is called from within the “agent”. Using this approach we were able to obtain a very clear assignment of model features to these components.

4.2 Model features and simplifications

Figure 3 shows an overview of all model features. Most of these features may be complex but there is usually enough information to model them adequately. But with modelling cluster tools, there is one very important component which is very hard to model, the robot handler. This handler has a very significant influence on capacity and cycle times. It is thereby able to screw the results of a simulation. The main issue with these handlers is the availability of their movement and loading logic. If not modelled correctly models easily run into deadlocks or create strange cycle time behavior. There are many levels of detail in which such a handler can be modeled, but with increasing detail in the logic modeled the risk of deadlocks increases. To avoid deadlocks as far as possible we decided to implement a very abstract

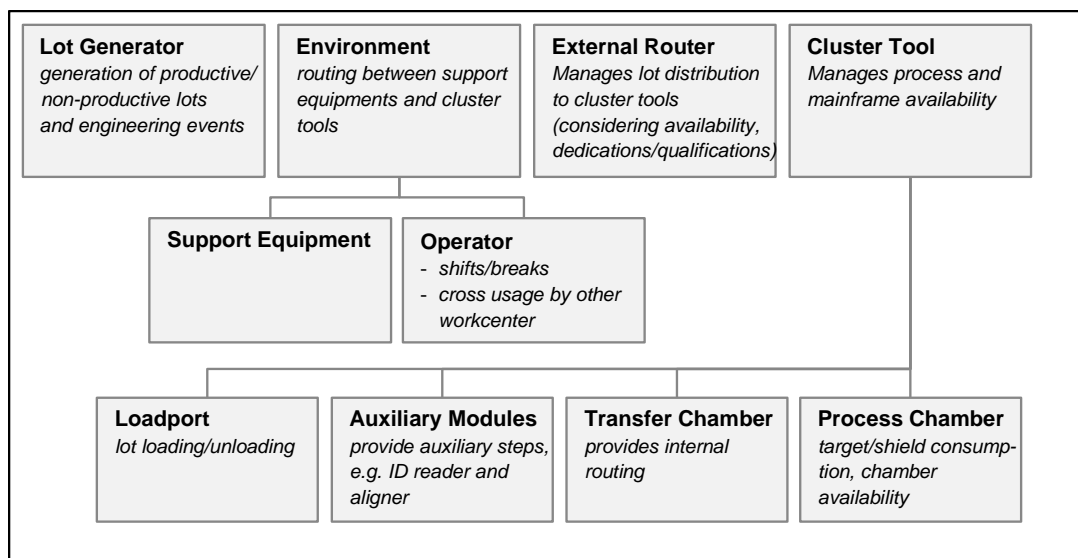


Figure 3. Hierarchy tree of model components

version of the handler, which basically consists of a delay element with a capacity of two wafers and a queue. Once a chamber, which is needed by one of the wafers in the queue, is ready to accept a wafer this particular wafer is transferred there via the delay component using an average transport time. Wafers finished in a chamber simply move out of the chamber into the queue.

This approach is pragmatic, but there is a hidden capacity gain which is not available in the real world. Especially under heavy load situations this has a significant effect on tool capacity. Imagine most chambers in the tool are occupied by a wafer; therefore no wafer can be placed in them until the old wafer is removed. Furthermore there is no bufferspace in the tool to store wafers inbetween process steps. Whenever we want to place a wafer into a chamber, we will therefore end up with a wafer in one of the two arms of our handler. As it can not be stored anywhere else, it will block the handler until it can be placed in a chamber. Hence the handler can only function properly if wafers taken from a chamber can be moved to another chamber immediately. A sequence of these switches ends with an empty handler.

Looking at the simulation data from the model and comparing it with real world measurements we found a 5-10% higher capacity and a similar cycle time reduction. To tackle this effect increasing handling time is a first step, but this did not properly adjust cycle time and frequency of wafers leaving the tool, as these effects of chamber blocking depend on the recipe produced.

We therefore added individual factors to all recipes, and verified the results by simulation. With the use of a combination of these process time factors and handling time factors we are able to avoid modelling the complex handler behaviour while obtaining very similar results. As with all simplifications and measures to account for them we will need to keep in mind to continuously verify their influence with a changing product mix.

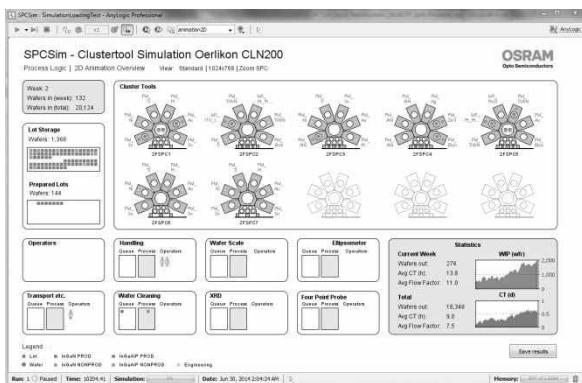


Figure 4. Screenshot of the simulation application

5 Use Cases

In this section we discuss three use cases of our simulation model.

5.1 Comparison of conventional approach and simulation

30 scenarios have been analyzed to determine the operating curve of the modeled system with and without support processes. For each scenario, a timeframe of 26 weeks has been simulated with a fixed wafer start volume for each week. After one week of warm up the system reaches a stable state regarding WIP levels, remaining target life etc. so that the average cycle time (CT) of each scenario has been calculated for the weeks 2 to 26.

Figure 5 shows the resulting operating curves. While the total raw process time (RPT) of the scenarios including environment (i.e., with support processes and operator handling) is only marginally longer than without environment, the operating curve shows significantly longer cycle times due to additional variability caused by, e.g., operator breaks, shift changes, and concurrent usage of metrology equipment.

The overall capacity limit of the system is determined by the cluster tools and is therefore the same with and without environment. At utilization levels beyond the capacity limit, a steady state of the simulated system is no longer possible and WIP levels and cycle times increase constantly over time.

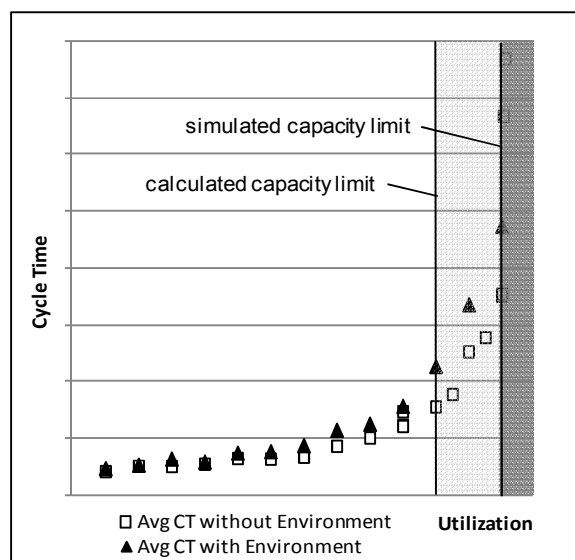


Figure 5. Comparison of static capacity calculation with the simulated operating curves including and excluding environment processes

Interestingly, the capacity limit obtained by simulation is about 15% higher than the capacity limit ob-

tained from spreadsheet calculation. The main reason is that uptime in the static spreadsheet calculation is determined following the SEMI E10 standard for cluster tools; hence it is a weighted average of the individual process uptimes. In contrast, in the simulated system there are often lots available that can run even if one or more modules of the cluster tool are down, especially in high loading/high WIP scenarios. This indicates that while SEMI E10 cluster tool uptime calculation is helpful to assess equipment stability, it might have weaknesses when used for capacity calculation. Instead, an uptime closer to the highest single process uptime should be used for static calculations.

5.2 Influence of different operator priorities on cycle time

The simulation model allows assigning different priorities to the various operator tasks. The following three strategies have been compared with respect to their influence on lot cycle time:

- FIFO – same priorities for all operator tasks; tasks are performed in the order they arise.
- Ascending priorities – Finishing lots and passing them to the next operation has priority over starting new lots. This is achieved by ascending priorities for all tasks in the operation sequence.
- Clustertool first – this strategy is similar to (b), but loading the cluster tools has always the highest priority if a lot is available.

Figure 6 shows the results of the three different strategies. From a cycle time perspective, FIFO is clearly the worst strategy. Cluster tools frequently wait for operators, while at the same time lots are handled at support processes. From the two “Ascending priorities” strategies, the “Clustertool first” strategy (c) shows significantly better results in terms of cycle time. Strategy (c) also leads to a higher overall capacity limit, as loading the cluster tools is always prioritized, leading to minimum idle times on these tools. In contrast, in strategy (b) lot handling after the sputter process has priority over reloading the tools, which leads to additional idle times of the system bottleneck.

The simulation results confirm the expectation that the capacity-limiting bottleneck should always have the highest priority. The closer the bottleneck shifts to its capacity limit the higher the cycle time difference becomes between the three strategies.

To summarize, operator task priorities can significantly affect overall performance of the workcenter and need to be defined properly in order to achieve reasonable cycle times in high loading situations.

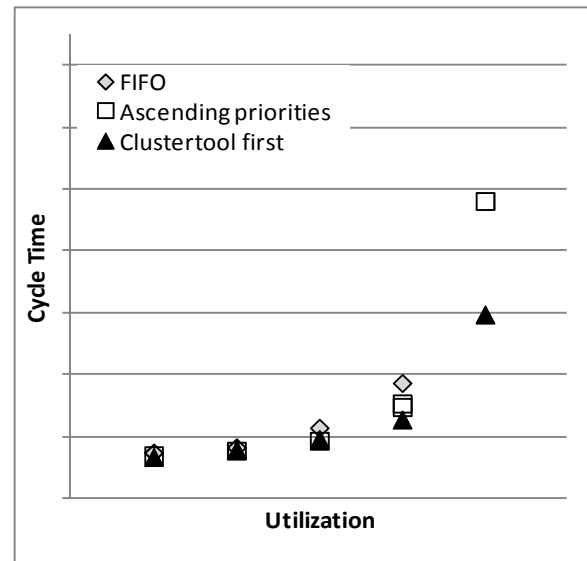


Figure 6. Influence of different operator task priorities on cycle time

5.3 Comparison of different equipment setups

In this use case, a given wafer production plan can be manufactured with different equipment configurations and process setups. In setup (a) all tools are basically configured the same way and run standard processes, but have to be dedicated either for standard process 1 or process 2. In setup (b) the processes are split up into two parts, and the tool and chamber configurations are re-arranged to run either the first or the second part of the processes, but can handle both standard processes. In both scenarios some processes are also run that only use one or two chambers and do not have to be split up. The advantage of the second scenario lies in additional redundancy and higher availability of the single processes, as fewer chambers are needed for the split processes.

Figure 7 shows the simulation results comparing the average lot flow factors (FF), i.e., CT/RPT.

At first glance, scenario (b) had higher flow factors than scenario (a) and did not seem to be desirable. Further analyses showed that the root cause in this case was that the split process concept led to about 30% more wafers to move into the workcenter, which brought the environment resources to their capacity limits. After increasing operator and metrology resources, the split process concept led to better flow factors, as was expected due to higher process availability.

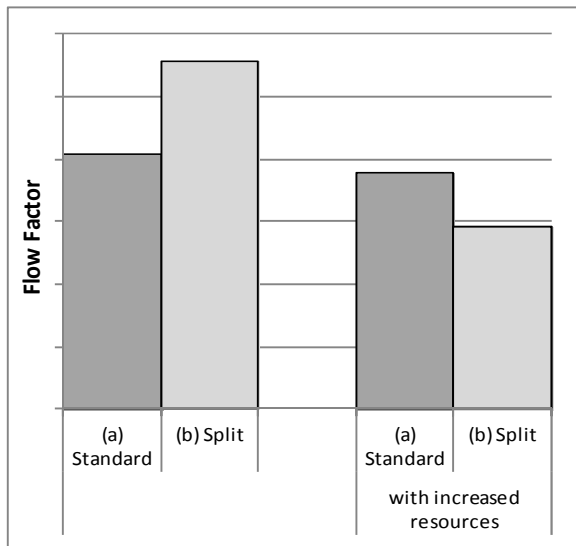


Figure 7. Effect of improved chamber configuration and process setup

6 Conclusions and outlook

During the project we were able to design a new simulation tool for cluster tools, which is now already used to seek answers for pressing questions in the capacity and strategic planning of our fabs. We were furthermore able to see some results we did not expect in the beginning of the project, thereby unlocking some new potential. For example, during our study we revealed that the dynamic uptime and therefore capacity of the workcenter is significantly higher than estimated by means of static planning. Additionally we were able to show a considerable effect of operator task priorities on the capacity of our toolset. However these are only the first steps with many more to come; e. g. analyzing the effects of other operator behavior, i. e. breaks and shift changes. With the current move of the system from the project phase to acceptance as a default approach for decision support there are surely still challenges ahead of us, for example extending our current models applications with simulation based optimization.

From the technical perspective we are very satisfied with the approaches we used and are sure a focus on maintainability and the use of our tool chain has improved our development process significantly.

7 Acknowledgements

We would like to thank Thomas Frey, Christian Blumenstein and Robert Walter for their insightful discussions on the subject and Hans Lindberg for his help during implementation of the system.

8 References

- [1] Singer, P., "The driving forces in cluster tool development" *Semicond. Int.*, pp. 113–118, 1995.
- [2] Burggraaf, P. "Coping with the high cost of wafer fabs" *Semicond. Int.*, pp. 45-50, 1995.
- [3] Rossnagel, S.M., "Sputter deposition for semiconductor manufacturing" *IBM Journal of Research and Development*, vol.43, Issue 1-2, 1999, pp.163-179
- [4] Rossnagel, S.M., Powell R., Ulman A. "PVD for Microelectronics: Sputter Deposition to Semiconductor Manufacturing", Academic Press, ISBN 0080542921, pp. 103-181, 1998
- [5] Mönch, L., Fowler, J.W., Dauzère-Pérès, S., Mason, S.J., Rose, O. "A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations" *Journal of Scheduling*, Volume 14, Issue 6, pp 583-599, 2011
- [6] Nickel, R.T.: *Improving productivity of semiconductor manufacturing cluster tools*, Massachusetts Institute of Technology, p. 25, 1994
- [7] Srinivasan, R. S.: "Modeling and Performance Analysis of Cluster Tools Using Petri Nets" *IEEE Transactions on semiconductor manufacturing*, vol. 11, no. 3, pp. 394-403, 1998
- [8] LeBaron, H. T., Pool, M.: "The simulation of cluster tools: A new semiconductor manufacturing technology," *Simulation Conference Proceedings*, pp. 907-912, 1994
- [9] Mauer, J., Schelasin R., Miller, P.: "Analysis of cluster tool performance in semiconductor manufacturing," in 13th IEEE/SEMI Int. Electronics Manufacturing Technology Symp. pp. 129-134, 1992
- [10] Zuberek, W.M., "Cluster tools with chamber revisiting-modeling and analysis using timed Petri nets," *Semiconductor Manufacturing*, *IEEE Transactions*, vol.17, no.3, pp.333,344, 2004
- [11] Gamma, E., Helm, R. Johnson, R., Vlissides J., "Design Patterns. Elements of Reusable Object-Oriented Software", Addison Wesley, ISBN 978-0201633610, 1995 pp. 207-217

Webbasierte Animation von Simulationsläufen auf Basis des Core Manufacturing Simulation Data (CMSD) Standards

Sören Bergmann¹, Florian Parzefall¹, Steffen Straßburger¹

¹Technische Universität Ilmenau

soeren.bergmann@tu-ilmenau.de

Animation von Simulationsläufen ist für viele Anwendungen ein nicht zu unterschätzendes Hilfsmittel. Die Nutzungsmöglichkeiten sind hierbei mannigfaltig, sie reichen von der Validierung der Modelle bis hin zur Ergebnispräsentation von Simulationsstudien. Dem Nutzen steht mitunter aber auch ein nicht zu unterschätzender Aufwand gegenüber, gerade im Kontext der automatischen webbasierten Simulation sind oft geeignete Animationen nicht verfügbar. Im Rahmen dieses Papers wird ein Ansatz vorgestellt, welcher ein bestehendes Framework zur automatischen Modellgenerierung, -initialisierung und Simulationsdurchführung inklusive Ergebnisauswertung auf Basis des Core Manufacturing Simulation Data (CMSD) Standards um die Möglichkeit der vollständig automatischen webbasierten Animation erweitert. Hierzu wird neben der Diskussion der Grundlagen der Animation das bestehende Framework und der dem Framework zugrunde liegende CMSD-Standard vorgestellt. Des Weiteren werden verschiedene Implementierungstechnologien vom Streamen von Videos über das Nutzen von Plug-Ins wie Flash oder Java Applets bis hin zu modernen Techniken wie HTML 5, CSS3 und JavaScript kritisch beleuchtet. Abschließend wird eine prototypische Implementierung mittels HTML 5 Canvas und den JavaScript Frameworks JQuery und KineticJS vorgestellt.

1 Einleitung

Die diskret-ereignisgesteuerte Simulation wird in verschiedensten Disziplinen und Anwendungsgebieten eingesetzt. Insbesondere im Bereich der Produktion und Logistik ist Simulation eine akzeptierte Methode zur Planung, Evaluierung und Verbesserung bzw. Steuerung von Prozessen [1]. Eine in den letzten Jahren immer wieder genannte Herausforderung im Kontext der Simulation ist in der Reduktion des Aufwands von Simulationsstudien bzw. in der verbesserten Nutzbarkeit der Simulation vor allem auch durch Nicht-Simulationsexperten zu sehen [2].

Ein hierzu in den letzten Jahren verfolgter Ansatz ist die Automatisierung der Modellgenerierung und -initialisierung. In Vorarbeiten konnte gezeigt werden, dass unter Nutzung des Core Manufacturing Simulation Data (CMSD) Standards u.a. eine Aufwandsreduktion bei der Simulationsmodellgenerierung [3-5], -initialisierung [3,6] und Ergebnisauswertung [3,7] möglich ist.

Ein weiterer wichtiger Trend der letzten Jahre ist die zunehmende Realisierung von webbasierten Anwendungen, hierbei sind sowohl im Browser laufende Anwendungen bzw. Nutzerschnittstellen als auch Cloud-basierte bzw. "Software as a Service" Lösungen zu subsumieren [8].

"Simulation as a Service"-Lösungen und webbasierte Nutzerschnittstellen sind hierbei eine mögliche Chance zur weiteren Reduktion von Aufwänden, sowie zur Verbesserung der Benutzerfreundlichkeit der Simulation, auch dies konnte in Vorarbeiten [3,7] gezeigt werden.

Zudem konnte gezeigt werden, dass eine weitere Reduktion durch Verteilung von Experimentläufen auf mehrere Simulatorinstanzen realisierbar ist [3,7].

Ein bisher in den meisten Ansätzen zur automatischen Modellgenerierung, -initialisierung und automatisierten Durchführung von Experimenten vernachlässigter Punkt ist die Animation von Simulationsläufen. In einigen Fällen, insbesondere bei Nutzung von bausteinbasierten Simulationswerkzeugen, ist zwar meist eine rudimentäre Animation per se gegeben, aber insbesondere in webbasierten Simulationsszenarien ist diese für den Nutzer nicht ohne weiteres nutzbar.

Zu bemerken ist aber, dass die Animation von Simulationsläufen von vielen Simulationsnutzern als wichtiges Hilfsmittel in allen Phasen der Simulation, von der Validierung bis hin zu Ergebnispräsentation bzw. Wissensvermittlung angesehen wird [9].

Im Rahmen dieses Papers wird ein Ansatz vorgestellt, der ein Framework zur webbasierten Simulation auf Basis des CMSD-Standards um die Möglichkeit der 2D-Animation ergänzt. Hierbei wird wie in allen

Komponenten des Frameworks auf eine rein webbasierte, d.h. für den Nutzer allein im Browser laufende intuitiv nutzbare Benutzeroberfläche gesetzt. Eine Forderung ist hierbei, dass keine zusätzlichen Aufwände zur Erzeugung von Layout- und Tracedaten anfallen. Insbesondere Aufwände einer zusätzlichen manuellen Layouterzeugung sind zu vermeiden.

Alle für die Animation benötigten Informationen liegen allein im CMSD Datenformat vor. Die CMSD Daten sind hierbei mit denen identisch, welche bereits erfolgreich zur Modellgenerierung, Initialisierung sowie der Ergebnisdatenanalyse mittels eines webbasierten Statistikmonitors genutzt werden [3-7]. Weitere optionale Ergänzungen zur Verbesserung der Animation (z.B. Definition spezifischer Icons usw.) sind ebenso im CMSD Format zu hinterlegen.

Im folgenden Abschnitt werden kurz die benötigten Grundlagen bzgl. der Animation von Simulationsläufen gelegt, bevor der CMSD-Standard und dessen grundlegende Interpretation thematisiert werden. Den Grundlagenteil schließen Ausführungen bzgl. der Auswahl der verwendbaren Technologien zur Realisierung der Animation als webbasierte Anwendung ab. Im Abschnitt 3 wird das Konzept zu webbasierten Animation auf Basis des CMSD-Standards sowie dessen prototypische Umsetzung vorgestellt.

Abschließend werden ein kurzes Fazit sowie ein Ausblick auf weitere Forschungs- bzw. Entwicklungsmöglichkeiten gegeben.

2 Grundlagen

Im folgenden Abschnitt werden zunächst Grundlagen der Animation diskutiert, der CMSD Standard, nötige Interpretationen und Erweiterungen vorgestellt sowie die Auswahl der für die im Prototypen genutzten (Web-)Technologien begründet.

2.1 Animation von Simulationsläufen

Die VDI Richtlinie 3633 Simulation von Logistik-, Materialfluss- und Produktionssystemen Blatt 11 definiert Animation als "die Erzeugung und Präsentation von Bildfolgen, in denen Änderungen (z.B. Zustandsänderungen und Bewegungen von Modellelementen) einen visuellen Effekt bedingen.

Unter visuellen Effekten kann eine über die Zeit variierende Position, die Änderung von Form, Farbe, Transparenz, Struktur und Musterung eines Objektes, die Änderung der Beleuchtung sowie der Position,

Orientierung und Brennweite der Kamera verstanden werden" [9].

Bei der Gestaltung der Animation ist besonders auf Expressivität, Effektivität und Angemessenheit zu achten [10]. Unter Expressivität wird hierbei die unverfälschte Wiedergabe der Informationen und unter Effektivität die geeignet intuitive Darstellung verstanden [9]. Angemessenheit betrachtet zusätzlich den Aufwand zur Realisierung der Animation in Relation zum Nutzen [10].

Im Kontext kommerzieller Simulationswerkzeuge sind verschiedene Animationsansätze anzutreffen. Zum einen kann zwischen vollintegrierten und entkoppelten Ansätzen unterschieden werden. Bei ersteren wird direkt bei jeder Simulationsstudie eine Kombination aus dem eigentlichen Simulations- und dem Animationsmodell erstellt, typisch ist dieses Vorgehen in bausteinorientierten Simulationswerkzeugen, z.B. Plant Simulation [11]. Die zweite Variante ist eine entkoppelte Animation, d.h. das Animationsmodell wird getrennt bzw. sogar nur bei Bedarf erstellt. Diese zusätzliche Modellierung erfolgt meist in einem eigenständigen Softwarewerkzeug, z.B. Proof Animation [12]. Im Fall der entkoppelten Animation muss diese durch die Simulation mit Tracedaten versorgt werden. Tracedaten können hierbei sowohl komplett nach Ablauf eines Simulationslaufes am "Stück" (Offline bzw. Post Simulation Animation) oder laufend (Online Animation) übertragen werden [5].

Im Gegensatz zum Grundprinzip der diskreteignisgesteuerten Simulation, in welcher von Ereigniszeitpunkt zu Ereigniszeitpunkt "gesprungen" wird, ist bei der Animation eine zeitproportionale Darstellung der Ereignisse im Modell einzuhalten, wobei neben des Ablaufs in "Echtzeit" auch Abläufe in Zeitraffer und Zeitlupe möglich sind [9, 13].

Bei der Animation muss zunächst bzgl. der Anzahl der Darstellungsdimensionen unterschieden werden, mögliche Ausprägungen sind 1D, 2D, 2,5D und 3D Visualisierungen. Für die Animation sind 1D Darstellungen allein nicht relevant, allein Zusatzinformationen lassen sich z.B. in Balkendiagrammen darstellen. Obwohl 3D-Visualisierungen im Kontext der Digitalen Fabrik an Bedeutung gewinnen [14], ist 2D die vorherrschende Darstellungsform im Kontext der Animation von Simulationsläufen in Produktion und Logistik. In Ausnahmen sind auch perspektivische Ansichten in 2,5D sinnvoll [13].

Ein weiteres Kriterium, welches im Kontext der Animation relevant ist, sind die Interaktionsmöglichkeiten, d.h. insbesondere der Navigation (Start, Stopp, Sprung zu einen Zeitpunkt usw.) und Anzeige (Zoom usw.) in der Animation. Weiterhin ist meist auch das Lesen von Zusatzinformationen zum aktuellen Animationszeitpunkt, z.B. aktueller Auslastungen von Ressourcen, Details zu Aufträgen usw. möglich. Interaktionsmöglichkeiten im weitesten Sinn können zudem Möglichkeiten der Manipulation der Visualisierung bzw. der grundlegenden Simulation über die grafische Darstellung umfassen [9, 13]. Diese erweiterten Möglichkeiten gehen allerdings über den Anspruch der klassischen Animation weit hinaus.

In den folgenden Ausführungen wird ein interaktiver entkoppelter, offline 2D Animationsansatz verfolgt.

2.2 Simulation auf Basis des CMSD-Standards

Das Core Manufacturing Simulation Data (CMSD) Information Model ist ein unter der Federführung der Simulation Interoperability Standards Organization (SISO) entwickelter Standard, der vor allem das Ziel verfolgt, Möglichkeiten des Datenaustauschs und die Interoperabilität zwischen Simulationen und anderen Anwendungen der Fertigung (z.B. ERP, MES) zu erleichtern [15, 16].

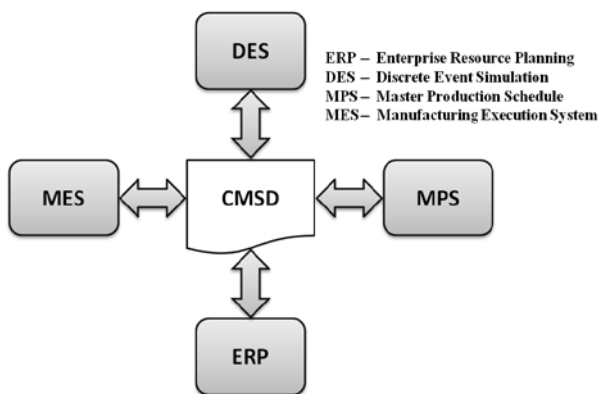


Abbildung 1. CMSD als neutrales Datenaustauschformat verschiedener Systeme der Fertigung [16]

Die CMSD Spezifikation umfasst zwei Teile. In Teil eins [15] werden UML Klassen- und Paketdiagramme zur Definition aller benötigten Entitäten und ihrer Eigenschaften (vgl. Abbildung 2) genutzt. Teil zwei [17] setzt diese Definitionen in technisch nutzbarer Art in Form von XML Schemabeschreibungen um. Hierbei kommen die sich ergänzenden Schemasprachen RelaxNG und Schematron zum Einsatz.

Versuche haben nachgewiesen, dass der Standard zwar im Detail interpretiert und ggf. mittels des im Standard verankerten Property-Konzepts erweitert werden muss, aber er geeignet ist, alle für eine Simulation nötigen Daten adäquat abzubilden und somit (webbasierte) automatische Modellgenerierung und -initialisierung aber auch die Abbildung von Ergebnissen von Simulationsläufen zu ermöglichen [4, 6-8].

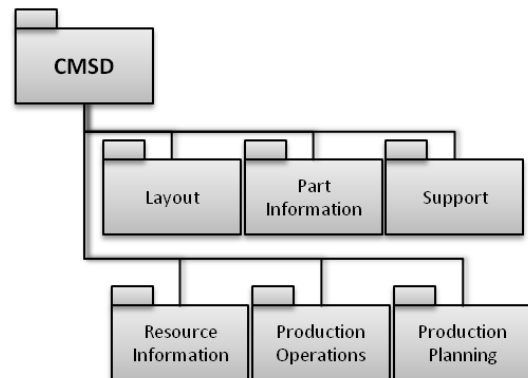


Abbildung 2. Pakete des CMSD-Standards [15]

Im Kontext der Animation von Simulationsläufen werden, wie bereits andiskutiert, zum einen Layoutinformationen und zum anderen Tracedaten benötigt.

Layoutinformationen lassen sich gut durch Entitäten des CMSD-Layout Paketes (z.B. LayoutObject), die meist eng mit Entitäten des Ressource Paketes (z.B. Ressource) verknüpft sind, abbilden. Ergänzend können organisatorische oder den Arbeitsablauf betreffende Informationen weiterer Pakete genutzt werden [3]. Grundlegende Layoutinformationen, z.B. zur Anzahl, Position und Fläche von Maschinenressourcen oder Hintergründe lassen sich direkt im CMSD Datenformat z.B. um die Angabe von Bildern oder komplexe geometrische Muster (Shapes) ergänzen.

Die Tracedaten werden mittels Events (Klasse des Support Paketes; vgl. Abbildung 3) abgebildet, welche Fertigungsaufträge (Jobs) des Production Operation Paketes erweitern können. Diese Mimik ist in Anlehnung an die Betriebsdatenerfassung (BDE) entwickelt worden und kam bereits zur Ergebnisdatenbeschreibung bzw. Kennzahlermittlung und statischen Darstellung von Simulationsläufen (z.B. Gantt Diagramme) im WebStatistikMonitor zum Einsatz und ermöglicht u.a. die zeitliche Rekonstruktion eines speziellen Simulationslaufs [3, 7].

Jedes Event umfasst hierbei zumindest eine laufende Nummer, welche die Erzeugungsreihenfolge abbildet,

einen Zeitstempel, welcher kennzeichnet zu welcher Simulationszeit das Event auftrat, einen Namen, der den Eventtyp (start setup, start work, broken usw.) kennzeichnet, sowie ggf. Referenzen auf beteiligte Ressourcen wie Werker und/oder Maschinen [3,7].

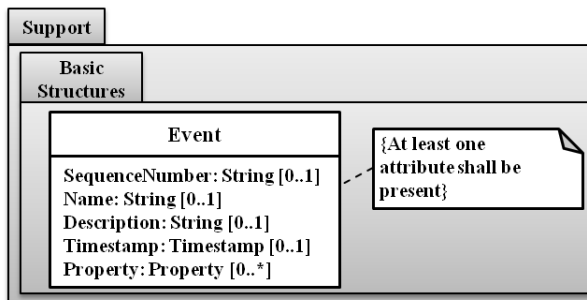


Abbildung 3. Die CMSD Event Klasse [15]

2.3 Auswahl der Webtechnologie

Im Rahmen der Umsetzung einer webbasierten Animationslösung stellt sich die Frage, welche Technologien geeignet sind.

Grundsätzlich lässt sich eine webbasierte Animation auf unterschiedliche Arten basierend auf verschiedenen Technologien implementieren. Im Folgenden soll eine Auswahl möglicher Ansätze bezüglich ihrer Stärken und Schwächen sowie möglicher Ausschlusskriterien untersucht werden. Eine umfängliche Diskussion aller Möglichkeiten würde den Rahmen dieses Papers jedoch sprengen.

Basierend auf der Annahme, dass serverseitig bereits eine Animation lauffähig vorliegt, entweder da der Simulator diese per se bei der Simulationsmodellierung mitgeneriert z.B. in Plant Simulation (vgl. [4]) oder diese in einem externen Werkzeug z.B. Proof Animation generierbar ist (vgl. [5]), ist der erste Lösungsansatz in der Bereitstellung von Videodaten zu sehen, welche als (Live-) Stream oder Download verfügbar gemacht werden. Neben der ggf. hohen Belastung des Servers sowie der Tool- und Plattformabhängigkeit ist vor allem die beschränkte Interaktionsmöglichkeit als Ausschlusskriterium zu sehen. So ist ggf. eine Navigation zwar noch entsprechend zu realisieren, eine Präsentation von interaktiven Zusatzinhalten hingegen ist kaum möglich.

Die zweite Gruppe von Lösungsmöglichkeiten basiert auf der Nutzung von Technologien, welche ggf. im Browser des Nutzers lauffähig sind, jedoch entsprechende Zusatzkomponenten (Plug-Ins) benötigt. Typische und verbreitete Vertreter dieser Gruppe sind Java-Applets [18], Flash [19] und Microsoft Silver-

light [20], aber auch ActiveX und JavaFX, welche hier nicht explizit betrachtet werden sollen.

Allen Technologien dieser Gruppe ist der Bedarf an zusätzlich zum Browser installierten Softwarekomponenten gemein, welche zwar meist eine hohe aber nicht vollständige Verbreitung aufweisen. Nachteilig ist, dass entsprechende Plug-Ins ggf. nicht für alle Plattformen, Betriebssysteme und Browser verfügbar sind, unterschiedlich gute Sicherheitskonzepte aufweisen und nicht standardisiert sind, zudem werden einige der Technologien kaum oder gar nicht weiterentwickelt. Im Folgenden werden einzelne Technologien diskutiert.

Java Applets ist die erste Technologie dieser Gruppe. Applets als Teil der Java SDK sind eine mittlerweile in die Jahre gekommene Technologie, die vor allem durch die Möglichkeit der Nutzung vieler existierender Java Klassen und Konzepte wie z.B. umfangreiche Kommunikationsmöglichkeiten mittels verschiedener Protokolle von Socket, RMI bis CORBA besticht. Des Weiteren fällt JAVA-Entwicklern die Entwicklung von Applets leicht, zudem sind die bestehenden Java Entwicklungsumgebungen (IDE) nutzbar. Des Weiteren verfügt diese Technologie über ein umfangreiches Sicherheitskonzept, z.B. Sandboxkonzept, Same-Origin-Policy etc. Dem gegenüber stehen beschränkte Möglichkeiten bzgl. der Gestaltung des Nutzerinterfaces (UI), hohe Aufwände bei der Entwicklung und Wartung, schlechte Trennung zwischen Logik und Design sowie eine ggf. schlechte Performance [21].

Das Java Applet-basierte Skopeo war eines der wenigen webbasierten Animationsframeworks für Simulationen. Skopeo kann hierbei Proof Animation Daten (Layout und Trace) interpretieren und in einem Java Applet im Browser darstellen [22].

Eine zweite Implementierungsmöglichkeit ist in der Nutzung von Adobe Flash zu sehen. Flash war und ist (mit stark sinkender Bedeutung) der de facto Standard zur Darstellung von Videos und Animationen (ggf. auch interaktiv, z.B. Spiele) im Web. Flash ist ein vektorbasiertes Format, welches durch eine eigene Skriptsprache (Actionscript) ergänzt wird. Problematisch und in Summe zum Ausschluss führend ist neben den allgemeinen, für alle Plug-In-basierte Techniken geltenden Einschränkungen, besonders das proprietäre Datenformat, die weitgehende Bindung an die kostenpflichtige Entwicklungsumgebung von Adobe sowie die gerade auf mobilen Endgeräten

teilweise Nichtverfügbarkeit bzw. die hohen Ressourcenanforderungen und -verbrauch zu nennen [19, 23].

Eine weitere ebenso wie Flash proprietäre Technik ist Microsoft Silverlight. Das Ausschlusskriterium ist besonders das mangelnde bzw. in den letzten Versionen ausgehebelte Sicherheitskonzept sowie die enge Bindung an Windows als die Zielplattform [20, 24].

Die dritte und letzte Gruppe von Möglichkeiten zur Realisierung einer webbasierten Animation kann mit der Überschrift „Nutzen von aktuellen standardisierten Webtechniken“ beschrieben werden. Bei den Standards handelt es sich vor allem um den vom W3C kurz vor der Verabschiedung stehenden HTML5 Standard [25, 26] in Verbindung mit JavaScript (standardisiert als ECMAScript 262 [27]) sowie CSS3 [28].

Diese Standards ermöglichen eine Plug-in freie Realisierung komplexer interaktiver Webanwendungen, welche in allen aktuellen Browsern inkl. Browsern mobiler Endgeräte lauffähig sind.

Im Detail sind verschiedene Spielarten möglich. So können alle benötigten Daten auf verschiedene Arten zur Verfügung gestellt werden, vom "klassischen" Laden mit der Webseite bis hin zum (nach-)Laden der Daten bei Bedarf bzw. Verfügbarkeit mittels Techniken wie AJAX oder Websockets.

Auch bei der Realisierung der eigentlichen Anzeige sind diverse Möglichkeiten mit unterschiedlichen Vor- und Nachteilen möglich. So sind in HTML5 und bei Nutzung von JavaScript Animationen mittels Canvas Elementen oder als SVG (Scalable Vector Graphics) möglich. Canvas bietet hierbei die Möglichkeit, dynamische Bitmaps zu erstellen, SVG hingegen ist ein vektorbasiertes Zeichenformat. Beide Technologien sind prinzipiell ähnlich gut geeignet, unterscheiden sich aber in einigen Details (vgl. Tabelle 1).

Neben den beiden beschriebenen Möglichkeiten zur Realisierung ist auch eine sehr performante aber derzeit noch mit erheblichem Entwicklungsaufwand verbundene Realisierung auf Basis von CSS3 Transformation bzw. Animation denkbar [28].

Zusammenfassend kann gesagt werden, dass einige verschiedene Techniken theoretisch möglich sind und sich ein einzelner klarer Sieger nicht abzeichnet. Der Verzicht auf Plug-Ins, die zu erwartende Zukunftssicherheit, die Plattformunabhängigkeit (Server und Client) bei moderatem Entwicklungsaufwand, sowie

die in den letzten Jahren enorm gestiegene Performance von JavaScript Engines sprechen jedoch für HTML 5, JavaScript und CSS3 basierte Lösungen.

Canvas	SVG
Pixelbasiert (dyn. PNG)	Formbasiert
Ein HTML-Element	Mehrere grafische Elemente (Teil Dokumentobjektmodells - DOM)
Nur durch Skript änderbar	Durch Skript und CSS änderbar
Ereignismodell / Benutzerinteraktion ist präzise (x,y)	Ereignismodell / Benutzerinteraktion ist allgemeiner ("rect", "path")
Die Leistung ist bei kleineren Oberflächen, einer größeren Objektanzahl (>10T) oder bei beiden besser.	Die Leistung ist bei einer kleineren Objektanzahl (<10T), größeren Oberflächen oder bei beiden besser.

Tabelle 1. Canvas vs. SVG (in Anlehnung an [29])

CSS 3 wird aus Gründen des Aufwands und der Flexibilität für diesen Anwendungsfall als schlechteste der genannten Alternativen ausgeschlossen, Canvas und SVG sind gleichwertig. Im Folgenden wird aber allein eine auf Canvas Elementen basierte Lösung vorgestellt.

3 Prototypische Implementierung eines webbasierten Animationstools

Eine Implementierung wurde mittels HTML 5 Canvas Elementen und JavaScript prototypisch realisiert. Dieser Prototyp ergänzt ein auf dem CMSD-Standard aufbauendes Framework, welches bereits webbasierte Simulation ermöglicht. Zur Vereinfachung des Entwicklungsprozesses, zur Verbesserung der Browserkompatibilität, zum Senken des Testaufwandes usw. wurde auf entsprechende JavaScript-Frameworks zurückgegriffen. Konkret wurde JQuery [30] für die Realisierung der AJAX Anfragen, mittels welcher sämtliche Datenübertragungen realisiert wurden, sowie das speziell für Canvas-basierte Animation optimierte Framework KineticJS [31] verwendet.

KineticJS ermöglicht eine abstraktere Definition von Bildern und deren Animation. Zudem werden Schwächen des Canvas-Ansatzes geschickt gemildert. So ist bspw. durch das Nutzen mehrerer sich dynamisch

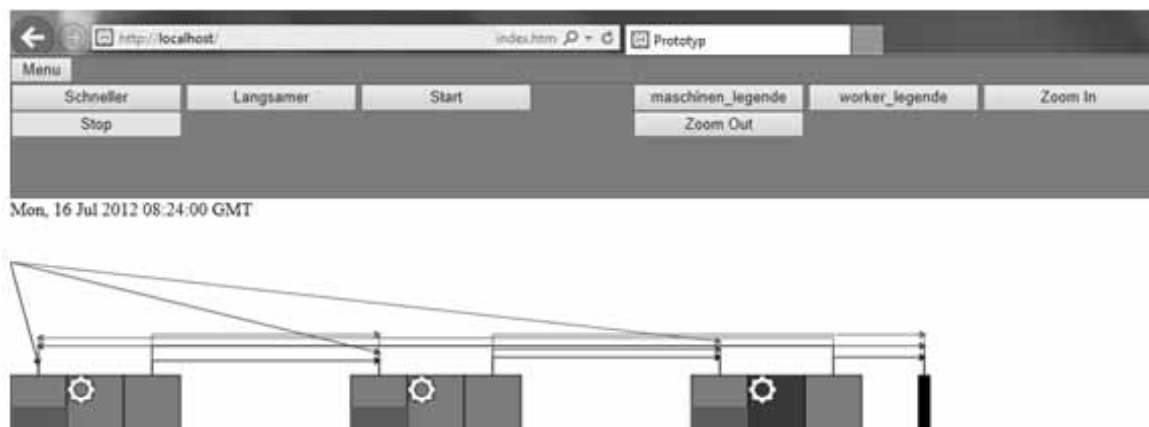


Abbildung 4. Screenshot des Prototyps

ergänzender Canvas-Elemente eine Manipulation über den DOM-Baum oder auch Zoomen ähnlich SVG möglich. Dies ermöglicht vor allem eine leichte Implementierung von zusätzlichen Interaktionen, z.B. der Anzeige von aktuellen Parametern, Kennziffern und Status von Aufträgen oder Maschinen bei Mausklick mittels eines entsprechenden Tooltips sowie das Zoomen zu jeder Zeit.

Grundlegend werden im Prototyp sämtliche Daten aus Dateien, welche serverseitig vorliegen, mittels AJAX Aufrufen in ein JavaScript Array geladen (vgl. Abbildung 5). Zusätzlich werden zunächst nicht direkt abgebildete Elemente heuristisch ermittelt und ergänzt, z.B. Quellen und Senken.

Alle statischen Elemente der Animation werden sofort nach erfolgreichem Laden der Layoutdatei gezeichnet. Das Erzeugen der Anzeige im Browser inklusive Bewegungen von Elementen wird über Funktionalitäten des KineticJS Frameworks realisiert. Hierzu werden für Maschinenressourcen deren Eigenschaften wie Position, Größe und die ggf. angegebene Grafik und Form ausgewertet und entsprechend in ein HTML5 Canvas gezeichnet. Zudem werden Funktionen z.B. zur Erzeugung eines Popups mit Statusinformationen bei Klick auf das Element hinterlegt.

Nach Laden des Trace werden alle möglichen Arbeitspläne ermittelt. Aus diesen werden die möglichen Pfade der Aufträge automatisch generiert und dargestellt. Danach wird, unter Berücksichtigung ggf. hinterlegter Schichtkalender, die Animationsuhr initialisiert und die Darstellung aller Elemente ihren aktuellen Zuständen angepasst, z.B. Markierungen von gestörten Maschinen (vgl. Abbildung 4).

Ab diesen Zeitpunkt kann die Animation jederzeit ablaufen. Während der eigentlichen Animation wer-

den basierend auf den Stand der "Animationsuhr" die Tracedaten kontinuierlich auf anstehende Events hin untersucht. Die Suche erfolgt in einem während der Initialisierung angelegten JavaScript Array, welches alle Events in zeitlich geordneter Reihenfolge enthält. Events führen hierbei je nach Typ zur direkten Änderung der Anzeige einzelner Elemente und / oder zur Erzeugung von Tweens (abgeschlossen laufender Animationsteile), welche u.a. für die Bewegung von Aufträgen zwischen Elementen genutzt werden.

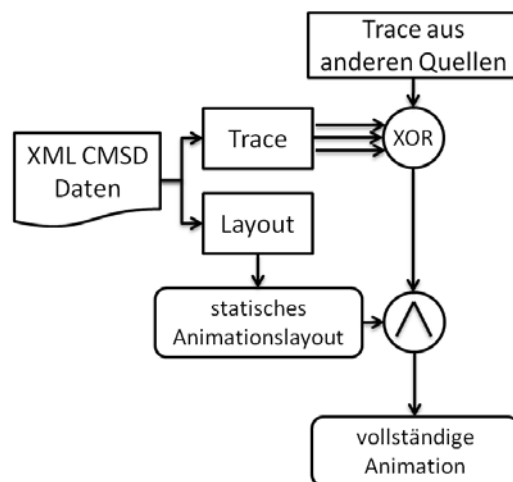


Abbildung 5. Datenfluss im Prototyp (in Anlehnung an [32])

Eine Veränderung der Tickrate der Animationsuhr (Buttons in der Oberfläche vorhanden) ermöglicht Zeitraffer und Zeitlupe Funktionalitäten, hierbei sind aber ggf. aktive Tweens bzgl. der Geschwindigkeit anzupassen. Das Ende der Animation wird durch das manuelle Stoppen der Animationsuhr oder die Verarbeitung aller Tracedaten bewirkt.

4 Fazit und Ausblick

Es konnte gezeigt werden, dass auf Basis vorhandener Daten im CMSD Format eine Animation von Simulationsläufen möglich ist. Des Weiteren konnte ebenfalls prototypisch die Machbarkeit von webbasierter Animation bei Verzicht auf zusätzliche Plug-Ins basierend auf modernsten Webtechnologien bewiesen werden. Alle Versuche zeigten, dass die Animation eine geeignete Expressivität und Effektivität ermöglicht. Eine Anpassung auf den konkreten Anwendungsfall ist mit angemessenem Aufwand zu realisieren. Ebenfalls zu bemerken ist, dass die Performance in den ersten Tests in allen Phasen gut war. So sind auf handelsüblicher Hardware und bei üblichen Netzwerkanbindungen nur geringe Ladezeiten und auch bei schnellem Vorlauf und Zoomen keinerlei negativen Effekte (z.B. Ruckeln) während der Animation aufgetreten. Umfangreiche Praxistests des Animationsansatzes stehen jedoch noch aus.

Über den bestehenden Prototypen hinaus sind weiterführende Forschungs- und Entwicklungspakete möglich. So sind mittels moderner Webtechnologien, z.B. WebGL [33] und entsprechenden JavaScript-Frameworks ebenso 3D-Animationen in sehr hoher Qualität und dank Hardwarebeschleunigung in guter Performance direkt im Browser möglich.

Ein weiterer Ansatzpunkt ist die Erweiterung des Prototyps um Möglichkeiten der Online Animation bzw. der onlinenahen Animation, d.h. der Animation noch laufender Simulationsexperimente. Hierzu ist zum einen eine geeignete (Trace-)Datenversorgung z.B. über Websockets zu realisieren. Zudem sind wie bei jedem Streamen von Livedaten oder auch in der verteilten Simulation Fragen wie das geeignete Zwischenpuffern der Daten und das Handling von verspäteten Events usw. zu betrachten.

5 References

- [1] VDI - Verein Deutscher Ingenieure. *VDI Richtlinie 3633-1 Simulation von Logistik-, Materialfluss- und Produktionssystemen - Blatt 1 Grundlagen*. Beuth, Deutschland, 2008.
- [2] S. Bergmann und S. Strassburger. *Challenges for the Automatic Generation of Simulation Models for Production Systems*. In Proceedings of the 2010 Summer Simulation Multi-conference, in Ottawa (Kanada), S. 545-549, 2010.
- [3] S. Bergmann. *Automatische Generierung adaptiver Modelle zur Simulation von Produktionssystemen*. Dissertation, TU Ilmenau, Deutschland, 2013.
- [4] S. Bergmann und S. Straßburger. *Generierung und Integration von Simulationsmodellen unter Verwendung des Core Manufacturing Simulation Data (CMSD) Information Model*. In Proceedings der 14. ASIM-Fachtagung Simulation in Produktion und Logistik. In Karlsruhe, S. 461-468, 2010.
- [5] S. Bergmann, S. Stelzer, S. Wüstemann und S. Strassburger. *Model generation in SLX using CMSD and XML Stylesheet Transformations*. In Proceedings of the 2012 Winter Simulation Conference, in Berlin, 2012.
- [6] S. Bergmann, S. Stelzer und S. Strassburger. *Initialization of simulation models using CMSD*. In Proceedings of the 2011 Winter Simulation Conference, in Phoenix (USA), S. 2228-2239, 2011.
- [7] S. Bergmann, S. Stelzer und S. Strassburger. *A new web based method for distribution of simulation experiments based on the CMSD standard*. In Proceedings of the 2012 Winter Simulation Conference, in Berlin, S. 3057-3068, 2012.
- [8] Gartner Inc. *Gartner Executive Programs' Worldwide Survey of More Than 2,300 CIOs Shows Flat IT Budgets in 2012, but IT Organizations Must Deliver on Multiple Priorities*. Abruf am 8. Januar, 2014. www.gartner.com/it/page.jsp?id=1897514.
- [9] VDI - Verein Deutscher Ingenieure. *VDI Richtlinie 3633-11 Simulation von Logistik-, Materialfluss- und Produktionssystemen - Blatt 11 Simulation und Visualisierung*. Beuth, Deutschland, 2003.
- [10] R. Schuhmann und W. Müller. *Visualisierung: Grundlagen und allgemeine Methoden*. Springer, Deutschland, 2000.
- [11] Siemens. *Plant Simulation*. Abruf am 15. Januar, 2014. www.plm.automation.siemens.com/de_de/products/tecnomatix/plant_design/plant_simulation.shtml.
- [12] J. O. Henriksen. *The Power and Performance of Proof Animation*. In Proceedings of the

- 1997 Winter Simulation Conference, in Atlanta (USA), S. 574-580, 1997.
- [13] S. Wenzel, J. Bernhard und U. Jessen. *A taxonomy of visualization techniques for simulation in production and logistics*. In Proceedings of the 2003 Winter Simulation Conference, in New Orleans (USA), S. 729-736, 2003.
- [14] S. Straßburger, H. Seidel, R. Schady und S. Masik. *Werkzeuge und Trends der digitalen Fabrikplanung - Analyse der Ergebnisse einer Onlinebefragung*. In: Proceedings der 12. ASIM-Fachtagung "Simulation in Produktion und Logistik", in Kassel, S. 391-402. 2006.
- [15] SISO Core Manufacturing Simulation Data Product Development Group. *Standard for: Core Manufacturing Simulation Data – UML Model*. Abruf am 05. März, 2014. http://www.sisostds.org/DigitalLibrary.aspx?Command=Core_Download&EntryId=31457.
- [16] M. Johansson, B. Johansson, A. Skoogh, S. Leong, F. Riddick, Y.T. Lee, G. Shao und P. Klingstam. *A test implementation of the core manufacturing simulation data specification*. In: Proceedings of the 2007 Winter Simulation Conference, in Washington (USA), S. 1673-1681, 2007.
- [17] SISO Core Manufacturing Simulation Data Product Development Group. *Standard for: Core Manufacturing Simulation Data – XML Representation*. Abruf am 05. März, 2014. http://www.sisostds.org/DigitalLibrary.aspx?Command=Core_Download&EntryId=36239.
- [18] Wikipedia. *Java-Applet*. Abruf am 05. März, 2014. <http://de.wikipedia.org/wiki/Java-Applet>.
- [19] N. Weschkalnies und S. Gasser. *Adobe Flash CS5: Das umfassende Handbuch*. Galileo Design, Deutschland, 2010.
- [20] Microsoft Inc. *Silverlight Developer Center*. Abruf am 05. März, 2014. <http://msdn.microsoft.com/de-de/silverlight/>.
- [21] C. Ullenboom. *Java 7 – Mehr als eine Insel: Handbuch zu den Java SE-Bibliotheken*. Galileo Computing, Deutschland, 2011.
- [22] P. Lorenz und K.C. Ritter. *Skopeo: Platform-Independent System Animation for the W3*. In Proceedings of Simulation and Animation '97, in Magdeburg, S.12-23, 1997.
- [23] S. Jobs. *Thoughts on Flash*. Abruf am 05. März, 2014. <http://www.apple.com/hotnews/thoughts-on-flash/>
- [24] W. Möhle. *Der Preis ist (zu) hoch*. iX 2/2010, S. 3.
- [25] W3C Working Group. *HTML 5- A vocabulary and associated APIs for HTML and XHTML W3C Candidate Recommendation 04 February 2014*. Abruf am 10. März, 2014, <http://www.w3.org/TR/html5/>.
- [26] P. Kröner. *HTML5 – Webseiten innovativ und zukunftssicher*. Open Source Press, Deutschland 2011.
- [27] Ecma International. *ECMAScript Language Specification 2011*. Abruf am 10. März, 2014, <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>
- [28] F. Maurice. *CSS3 - Leitfaden für Webdesigner*. Addison-Wesley, Deutschland, 2011.
- [29] MSDN. *So wird's gemacht: Auswählen zwischen SVG und Canvas*. Abruf am 15. März, 2014, [http://msdn.microsoft.com/de-de/library/ie/gg193983\(v=vs.85\).aspx](http://msdn.microsoft.com/de-de/library/ie/gg193983(v=vs.85).aspx).
- [30] JQuery Foundation. *jQuery*. Abruf am 14. Februar, 2014, <http://jquery.com/>.
- [31] E. Rowell. *KineticJS*. Abruf am 14. Februar, 2014, <http://kineticjs.com/>
- [32] F. Parzefall. *Konzeption und prototypische Implementierung eines webbasierten Pos-Simulation-Animationstools zur Verarbeitung von CMSD Daten*. Masterarbeit, TU Ilmenau, 2013.
- [33] Wikipedia. *WebGL*. Abruf am 19. März, 2014. <http://de.wikipedia.org/wiki/WebGL>.

A Formal Approach to Combine Models of Archaeological Sites

Philipp Pichler¹, Kerstin Kowarik², Niki Popper^{1,3}, Gabriel Wurzer⁴, Hans Reschreiter², Anke Bacher⁵,
Bernhard Heinzl¹

¹ dwh Simulation Services (dwh) - www.dwh.at

Neustiftgasse 57-59, 1070 Vienna - Austria

² Natural History Museum Vienna (NHM) - www.nhm-wien.ac.at

Burgring 7, 1010 Vienna – Austria

³ Institute for Analysis and Scientific Computing - www.tuwien.ac.at

Wiedner Hauptstr. 8-10, 1040 Vienna - Austria

⁴ Vienna University of Technology - www.tuwien.ac.at

Karlsplatz 13, 1040 Vienna - Austria

⁵ University of Vienna - www.univie.ac.at

Universitätsring 1, 1010 Wien - Austria

philipp.pichler@dwh.at

Computersimulationen helfen dabei verschiedenste Fragen der Archäologie, wie die Verwendung von Werkzeugen benutzt und Ressourcenverteilung oder die räumliche Nutzung des Gebietes, zu analysieren. Jede dieser Fragestellungen beinhaltet ihrerseits schon einiges an Komplexität. Um die ursprüngliche Funktionsweise einer archäologischen Grabungsstätte besser verstehen zu lernen, scheint es wichtig zu sein, die komplexen und dynamischen Interaktionen der verschiedenen Subsysteme zu kennen. Die Kombination unterschiedlicher Simulationstechnologien führt zu zusätzlichen Problemen in Bezug auf die Rechenleistung, den Datenaustausch und letztendlich die Validierung. In dieser Publikation wird gezeigt, wie heterogene Simulationsmodelle während der Laufzeit kombiniert werden können und wie Fehler vermieden werden können. Das Ziel ist es ein erstes Konzept vorzustellen, dass die Entwicklung einer stabilen dynamischen Simulation ermöglicht. Dabei werden verschiedenste Aspekte der räumlichen Simulation, der physikalischen Modellierung und der Ressourcennutzung beachtet. Die Salzminen aus der Bronzezeit in Hallstatt in Österreich (1458-1245 A.C.) sind ein hervorragendes Beispiel um das Konzept vorzustellen. Aufgrund der durch das Salz bedingten Konservierung in den Mienen, existiert bis heute noch eine außergewöhnliche Menge an Informationen. Gleichzeitig werfen die vielfältigen technologischen Aspekte des Untertagebaus und die komplexe ökonomische Struktur eine Reihe von Fragen auf, die bereits in der Vergangenheit aufgegriffen und mit verschiedensten Simulationstechniken bearbeitet wurden. Die Verbindung dieser Teile stellt das Hauptziel dieser Arbeit dar.

1 Einleitung

Die Formalisierung höchst komplexer archäologischer Funde und Fundstellen in ein Modell, das Einheiten und deren Beziehungen in hierarchische Systeme und deren Subsystemen darstellt, ist nicht neu. Dieses Thema wurde in den 1940er Jahren aus verschiedensten Blickpunkten erforscht. Werden Beziehungen und deren Rückkopplungsschleifen (kybernetisch, geprägt von Norbert Wiener[1]) sowie die Erforschung von Strukturen und ihre hierarchische Zusammensetzung (Systemtheorie, basierend auf Bertalanffy[2]) betrachtet, so wird klar, dass all dies

starken Einfluss auf Modellbildung und Simulation hat (z.B. System Dynamics). Daher verwundert es auch nicht, dass sich die archäologische Forschungsgemeinschaft diese Techniken schon seit den 1980ern zu eigen machte (z.B. Gummerman[3]).

Die Dokumentation eines Modells wird bereits durch das ODD Protokoll ([4], [5]) standardisiert. Nachdem diese Darstellung jedoch deskriptiv ist und keine strukturellen Abhängigkeiten darstellt wird in diesem Paper auf das mentale Konstrukt der Modelle und der Ideen dahinter eingegangen. Dafür wird Graphen-

Basierte Dokumentation zur Systembeschreibungen genutzt. Die Ziele sind dabei:

- Simulationsmodelle mit ihren zugrundeliegenden mentalen Konzepten (siehe Kapitel 4), zu verankern; wobei ein zweistufiger Ansatz verwendet wird: Mentale Konzepte stellen die untere Strukturebene dar, die durch ein oder mehrere Modelle auf der höheren Simulationsebene implementiert wird.
- Ein besonderes Interesse liegt auf der Verbindung (siehe Kapitel 4.2) zwischen unterschiedlichen Komponenten auf derselben Ebene (Relationen) oder auf verschiedenen Ebenen (Referenzen). Die genauere Betrachtung dieses Themas ermöglicht die Verbindung zweier Modelle in sequentieller oder paralleler Weise (hybride Simulation, Co-Simulation).

Da es sich hier um ein technisches Paper handelt und keine kognitive Theorie vorgestellt werden soll, konzentrieren wir uns hier darauf Konzepte vorzustellen, welche auch in anderen Arbeiten verwendet werden können (z.B. als Framework). So wird hier versucht, die Beschreibungen so gut wie möglich von der für diese Studie betrachteten Archäologischen Stätte (Hallstatt/Oberösterreich), die nur kurz im Kapitel 2 zu Sprache kommt, zu entkoppeln. Genauso versucht die Darstellung der allgemeinen Konzepte einen Bezug zur Anwendung und die Grenzen dieser Methodik zu geben, ohne die Verbindung zu den unmittelbaren Resultaten unseres Modells, die schon an anderer Stelle präsentiert wurden darzustellen.

2 Hallstatt

Die prähistorischen Salzminen in Hallstatt stellen den Ausgangspunkt unserer Studie dar. Hallstatt befindet sich in den östlichen Alpen, am Fuße des Dachsteins. Die ersten Nachweise für Untertagebergbau in den Salzminen in Hallstatt sind auf die Bronzezeit (im 16. Jahrhundert A.C.) datiert. Heute sind verschiedenste Phasen des Bergbaus in der Mittleren/Späteren Bronzezeit, genauso wie in der früheren und späten Eisenzeit bekannt.

Hallstatt wird als wichtiger Salzproduzent in Europa während der gesamten prähistorischen Zeit betrachtet. In der Bronze-Zeit könnte Hallstatt über weite Strecken Europas [6] sogar das Monopol auf Salzproduktion inne gehabt haben. Des weiteren wird dieser Fundort auch als wichtiges Zentrum zur Konsumierung und zum Tauschen von Arbeitsmaterial und

Nahrungsmitteln. In diesem Kontext stellt der Standort in einem alpinen Umfeld eine beachtliche Besonderheit dar.

Durch die ausgezeichneten Bedingungen innerhalb der Salzmine, durch die das Salz die Fundstücke präservierte, blieben tausende Werkzeuge und Arbeitsmaterialien der Bronze- und der frühen Eisenzeit erhalten. Dies ist der Grund dafür, dass die Arbeitsprozesse dieser Zeit sehr detailliert rekonstruiert werden können. Mit Hilfe dieser Rekonstruktionen untersuchen wir komplexe ökonomische Strukturen, welche eine Vielzahl von Akteuren mit unterschiedlichen Rollen und Funktionen, aufweisen. Es existiert eine große Anzahl von Aktivitäten, die möglicherweise auch weitreichende Kontakte innerhalb Europas beinhalten mussten (der geographische Forschungsbereich reicht weit über Hallstatt hinaus).

An diesem Forschungsprojekt arbeitet eine Vielzahl an wissenschaftlichen Disziplinen, die sich mit Themen rund um Hallstatt beschäftigen, wie Archäologie, Computerwissenschaften, Mathematik, Life Sciences, Geologie und so weiter. Darum existieren vielerlei Hypothesen, die auf unterschiedlichen Datentypen (Ausgrabungsfunden, ethnographische und historische Analysen, anthropologische Daten, Daten aus der experimentellen Archäologie usw.) basieren und sich stark mit dem aktuellen Lebensbild verbinden.

3 Archäologische Voraussetzungen für die strukturelle Dokumentation

Aufbauend auf den Funden und Erkenntnissen der Ausgrabungen in den prähistorischen Bergbaugebieten wurden verschiedenste Lebensbilder entwickelt, die von Darstellungen der Stollen bis hin zu Darstellungen der gesamten Region reichen. Diese Modelle bauen (teilweise) aufeinander auf und beeinflussen sich gegenseitig. Ein Lebensbild kann dabei als Zusammensetzung von Funden und der Forschung rund um diese Funde und die Fundstellen, also als Zusammensetzung der Ergebnisse aus Anwendungen unterschiedlichster Methoden gesehen werden.

Um ein Modell zu erstellen das eine komplexe archäologische Stätte abbildet muss es möglich sein über alle Bestandteile bescheid zu wissen und diese zu verwalten. Damit sich eine (mentale) Struktur zu Simulationen verbindet, ist es dabei auch notwendig Argumentationslinien auf allen benutzten Ebenen nachverfolgen zu können. Das heißt:

- Identifizierung eines Datenfeldes innerhalb einer Argumentationskette, zum Beispiel „zeige mir alle Daten, die im Gräberfeld der Frühen Eisenzeit erhoben wurden“
- Zeige die Datentypen, die in einer bestimmten Argumentation benutzt wurden (Daten der Anthropologie, Typologie, Archäologische Funde, Annahmen etc.)
- Zeige die Argumentation und das Zusammenspiel der verschiedenen Daten bei Auswahl einer Untergruppe des Lebensbildes (beschreibende Einflüsse).

Einige Aspekte in den jeweiligen Simulationen sind so eng miteinander verbunden, dass es von Vorteil wäre, wenn sich die einzelnen Simulationen gegenseitig beeinflussen können. Dabei ist zu beachten, dass verschiedenste Techniken, wie beispielsweise Agenten-Basierte Modellierung, Physical Modelling, System Dynamics usw. verwendet werden. Diese Kopplung bietet eine große Herausforderung, da alle involvierten technischen Aspekte bekannt sein müssen. Dabei ist die Art der Kopplung (linear, dynamische Verknüpfung/Co-Simulation) wichtig, da sie Einfluss auf den Ablauf der allgemeinen Simulation und den Arbeitsablauf der Verknüpfung zeigen. Sowohl die Reproduzierbarkeit als auch die Validierung von der Summe aller Teile und auch den einzelnen Teilen müssen hier im Vordergrund stehen.

4 Technische Rahmenbedingungen

4.1 Strukturelle Ebene

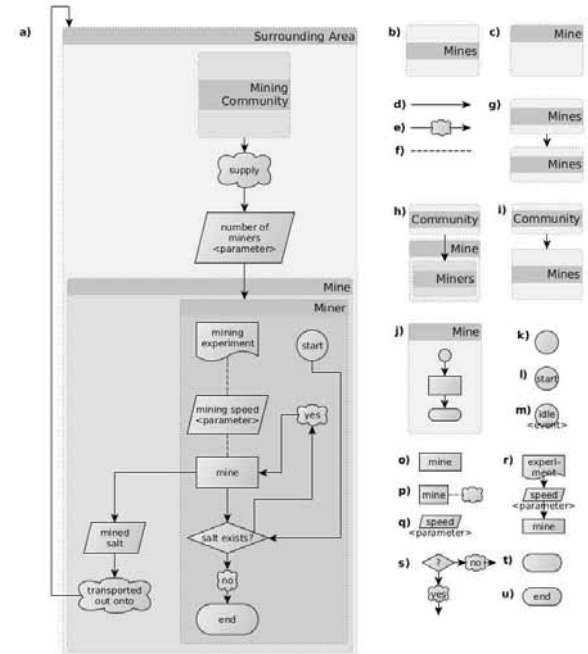


Abbildung 1: Strukturelle Ebene

Die strukturelle Ebene ist hierarchisch geordnet und besteht aus Entitäten, welche sich innerhalb weiterer Entitäten befinden. Eine Einheit kann auf zwei Arten dargestellt werden, versteckt (folded) oder offengelegt (unfolded). Folded (siehe **Abbildung 1b**) bedeutet dabei, dass alle Einheiten, die in einer weiteren Einheit enthalten sind, nicht angezeigt werden. Es ist also nur die „Übereinheit“ sichtbar. Im Gegensatz dazu sind bei offengelegten Einheiten alle ihre enthaltenen Komponenten (**Abbildung 1c**) sichtbar. Offengelegte Einheiten werden im Plural („Mienen“), versteckte in Singular („Miene“) geschrieben. Formal gesehen stehen versteckte Einheiten für eine bestimmte Kategorie von Einheiten und offengelegte für Objekte dieser Kategorie.

Beziehungen zwischen Einheiten können gerichtet (**Abbildung 1d u. e**) oder ungerichtet (**Abbildung 1f**) sein. Gerichtete Beziehungen werden dabei als Kommunikationsfluss verstanden: Die Einheit, die sich durch eine gerichtete Beziehung mit einer anderen Einheit verbindet, wird zuerst betrachtet, danach folgt der Kontrollfluss und letztendlich die verbundene Einheit. Dadurch wird eine temporäre und kausale Ordnung geschaffen, die benutzt wird, um durch die Struktur der mentalen Ideenkonstrukte, die die mo-

delierte Sttte betreffen, zu fhren. Jede Beziehung kann benannt werden (**Abbildung 1e**). Beziehungen, die Einflsse andeuten, aber keine Kommunikationsflsse, werden ohne Pfeile angezeigt, wie in **Abbildung 1f** gezeigt wird. Dies hat hauptschlich den Zweck Informationsmaterial mit einer Einheit zu verbinden, um dazugehrende Arbeiten und Experimente aufzuzeigen.

Entitten knnen einerseits auf derselben hierarchischen Ebene (**Abbildung 1g**) verbunden sein andererseits aber auch zwischen verschiedenen Ebenen (**Abbildung 1h**). In letzterem Fall wird eine Beziehung auch dann beibehalten, wenn sie versteckt/offengelegt wird (**Abbildung 1i**).

Einheiten, die durch Beziehungen verbunden sind, stellen schon Abhngigkeiten und Kommunikationsflsse zwischen den unterschiedlichen Teilen der Lebensbilder dar und bringen so eine Dynamik in das System. Diese Beschreibung erinnert schon sehr an die Konzepte der Objekt-Orientierten-Programmierung. Letztendlich reicht es jedoch nicht nur das Verhalten der Einheiten zu modellieren, deshalb werden explizite Prozessbeschreibungen in Form von Flussdiagrammen in folgender Weise inkludiert:

- Einheiten knnen auch Prozesse beinhalten (siehe **Abbildung 1j**). Als klassische Beispiele knnen Wirtschaftsprozesse gesehen werden, wie beispielsweise das Kundenservice, der Vertriebsbereich usw. Diese haben nur eine indirekte Beziehung zu den tatschlichen Personen, die die Arbeit erledigen, da eine Person mehrere Rollen gleichzeitig einnehmen kann. Ausserdem dient diese Erweiterung auch dem Verstndnis, dass Entitten auch als Prozesse dienen knnen. Dies bedeutet auch, dass die Vorstellung des Einheitsbegriffs zu nicht-krperlichen Elementen innerhalb eines mentalen Modells erweitert wird (z.B. technische Systeme, wie die „vertikale Luftzirkulation“ innerhalb einer Miene).
- Startpunkte von Prozessen (**Abbildung 1k-l**) knnen entweder ohne Bedingung oder mit einem Ereignis verbunden sein. Im ersten Fall, beginnt der Prozess mit dem „Beginn der Simulation“. Bei der Typisierung von Startpunkten steht es offen, ob der Punkt ohne Marke (**Abbildung 1k**) oder – um es bersichtlicher zu gestalten – die generalisierenden Bezeichnung „Start“ (**Abbildung 1l**) verwendet wird. Wird das Ereignis

nis ber ein Event gestartet so wird auch noch das Label „idle“ spezifiziert (**Abbildung 1m**) und das Suffix <event> angehngt.

- Schritte die whrend eines Prozesses passieren (**Abbildung 1o**) werden mit einer sinnvollen Bezeichnung versehen. Zustzlich gibt es immer die Mglichkeit ein Kommentar hinzuzufgen, wie es in **Abbildung 1p** gezeigt wird.
- Daten gehen in Form von Datenpunkten in den Prozess ein (**Abbildung 1q**). Handelt es sich bei den Daten um Parameter so wird der Punkt zustzlich mit dem Suffix <Parameter> bezeichnet. Bestehen die Daten aus mehreren einzelnen Teilen, wie zum Beispiel ein Artikel, ein Report eines Experiments und hnliches, wird seine Herkunft durch eine ungerichtete Verbindung zu einem Dokumentationspunkt gezeigt (**Abbildung 1r**). Weiters knnen solche Daten auch als Annahmen reprsentiert werden, was durch das Suffix <assumption> gekennzeichnet wird (wird in **Abbildung 1** nicht gezeigt).
- Entscheidungen innerhalb eines Prozesses werden mittels Entscheidungspunkten (**Abbildung 1s**) dargestellt. Dabei handelt es sich um Entscheidungspunkte, die mit Fragen („exists salt?“) (**Abbildung 1a**) gekennzeichnet werden, welche entweder mit ja oder nein beantwortet werden.
- Prozessendpunkte (**Abbildung 1o, p**) sind immer bedingungslos. Die hier prsentierten Konstrukte mssen explizit auf einen dieser Knoten hinfhren.

4.2 Simulationsebene

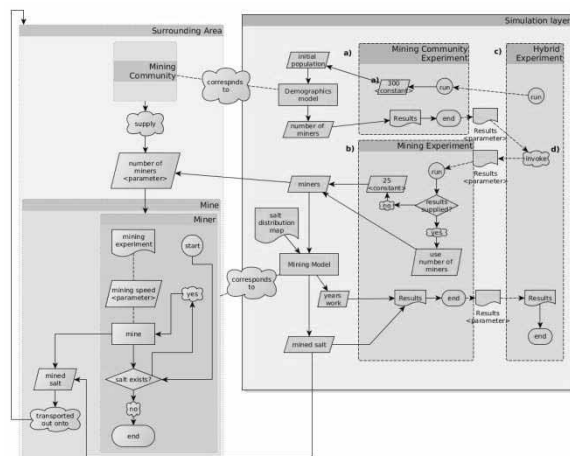


Abbildung 2: Simulationsebene

Die Simulationsebene (siehe Abbildung 2) beinhaltet Modelle mit Parametern und Ergebnissen. Wobei man unter Simulation die praktische Ausführung eines Modells versteht. Eine Simulation ist als Teilmenge der vorher beschriebenen „Strukturellen Ebene“ zu verstehen. Es werden also die im Lebensbild gesammelten Informationen und Funde „digitalisiert“. Beispielsweise ist die „Bergbaugesellschaft“ in der Strukturebene noch eine rein abstrakte Einheit. Auf der Simulationsebene wird diese dann wirklich in allen notwendigen Details modelliert [7] und dargestellt.

Hier ist es wichtig zwischen Experimenten und den Simulationen an und für sich zu unterscheiden. Ein Experiment ist die ausführbare Version einer Simulation. Das bedeutet, dass mit tatsächlichen Parameterwerten und Einstellungen gerechnet wird.

Bei dem hier vorgestellten Konstrukt liegt aber das Hauptaugenmerk vor allem auf der Kopplung von solchen Modellen. Im wesentlichen ist die Sichtweise immer dieselbe: Ein Modell wird mit Werten von anderen Simulationsexperimenten (**Abbildung 2a-c**) beliefert. Ein solches Experiment weist also einem Modell Werte zu (man kann es sich die „Einstellungen“ mit welchen ein Modell funktioniert denken). Die Werte, die durch Experimente geliefert werden, sind entweder Konstanten (z.B. durch den Experimententwickler gewählt) oder stammen von einer vorhergehenden Berechnung: zum Beispiel kann gesehen werden, dass das „Bergbaugesellschaftsexperiment“ einen konstanten Wert von 300 Personen als Parameter „Anfangsbevölkerung“ des „demographischen Modells“, welches die „Zahl der Bergarbeiter“ als Ergebnis kalkuliert, liefert. Die anschließende Nutzung dieser Werte kann im ‚hybriden Experiment‘ gesehen werden, welches die Zahl der Bergarbeiter (enthalten im ‚Ergebnis‘) an das ‚Bergarbeitsexperiment‘ weiter gibt.

Es können mehr oder weniger dieselben Punktypen wie auf der strukturellen Ebene verwendet werden um Modelle, Parameter, Ergebnisse und Elemente zu beschreiben. Es gibt allerdings ein paar Ausnahmen, was die Bedeutung betrifft:

- Anfangspunkte der Prozesse werden benutzt um die Durchführung eines Experiments zu kennzeichnen (nicht als abstrakter Prozess als Teil der Strukturebene).
- An Endpunkten des Prozesses werden Ergebnisse gesammelt (multiple Ergebnisse können auch ite-

rativ gesammelt werden. Dies wird in **Abbildung 2** nicht gezeigt).

- Es gibt einen speziellen Punkt, der als „invoke“ bezeichnet wurde (siehe **Abbildung 2d**): Dieser bezieht sich auf die Verbindung zwischen Simulationsmodellen, die im nächsten Kapitel beschrieben werden.

4.3 Verbindungen von Simulationen / hybride Simulation

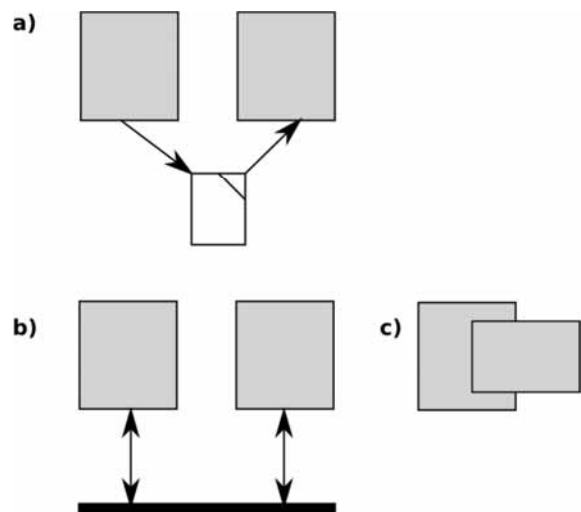


Abbildung 3: Kopplung

Wenn heterogene Simulationsmodelle (z.B. Agentenbasierte, Netzwerkbasierte oder Gleichungsbasierte Modelle) verbunden werden, gibt es zwei Möglichkeiten zur Kopplung:

- Linear: Simulationsmodelle werden sequentiell ausgeführt; die Ergebnisse werden zwischen Modellen offline weitergegeben, normalerweise in Form einer Datei (**Abbildung 3a**)
- Dynamisch: Simulationsmodelle werden in paralleler Weise ausgeführt; Datenaustausch passiert hier normalerweise während der Ausführung. (**Abbildung 3b**)

Lineare Kopplungen sind um einiges einfacher umzusetzen als dynamische. Sollte diese Vorgehensweise ausreichen, so wird diese bevorzugt. Z.B. wenn ein Ergebnis nur am Ende der Simulation berechnet und dann eingelesen wird.

Dynamische Kopplungen, z.B. während der Laufzeit werden oft benutzt um dynamische Interaktionen in räumlichen Simulationen abzubilden, um physikali-

sche Prozesse zu modellieren oder Ressourcennutzung abzubilden. Diese Herangehensweise benötigt einen höheren Aufwand (z.B. Aggregation und Disaggregation von Interfaces zwischen Agenten- und Gleichungsbasierten Ansätzen) und eine gründliche Validierung (siehe Kapitel Diskussion).

Alle zusammengeführten Modelle – linear und dynamisch – sind hybrid. Klassische Definitionen, wie eine Mischung zwischen diskreten und kontinuierlichen Simulationen [8] waren lange Zeit der Standard in diesem Bereich. In der heutigen Zeit bewegt man sich nun eher in Richtung dynamischer Kopplung während der Laufzeit, durch ein Framework wie z.B. in BCVTB [8], bewegt. Eine weitere Möglichkeit zur Hybriden Simulation ist die Einbettung (**Abbildung 3c**). Dies wird seit den letzten Jahren immer attraktiver, da Simulationen immer mehr einem modularen Aufbau folgen [9].

Da es verschiedenste Modellierungstechniken gibt welche verbunden werden, wird hier darauf verzichtet was bei jeder einzelnen Verbindung im Detail zu beachten ist. Für mehr Informationen über Agentenbasierte Modellierung sei auf [10] verwiesen. Zu Literatur zu System Dynamics ist in [11] zu finden und Physical Modelling in [12].

Da der Rahmen dieses Papers gesprengt werden würde, wird hier nicht näher darauf eingegangen was zu beachten ist, wenn je zwei dieser Modellierungsstrategien verbunden werden. Stattdessen wird auf die strukturellen Unterschiede der verschiedenen Techniken eingegangen und was bei der Verbindung beachtet werden sollte. Für eine Begriffserklärung sei wiederum auf [10] verwiesen.

- **Kontinuierlich vs diskret:** Simulationsmethoden unterscheiden sich oft sowohl räumlich als auch zeitlich durch diskrete oder kontinuierliche Ansätze. Eine Kopplung bei zeitlich verschiedenen Ansätzen ist meist nur möglich in dem man die Modelle umschaltet. Dies funktioniert jedoch nur, falls diese Modelle als abgeschlossene Einheiten betrachtet werden können, oder eines der Modelle nur einen Tielablauf im anderen Modell darstellt. Unterscheiden sich die Ansätze was das räumliche Bezugssystem angeht, so ist es auch möglich Modelle simultan laufen zu lassen. Wichtig ist aber in jedem Fall bei der Kopplung darauf zu achten, dass dieselbe Schrittgröße / Zellgröße verwendet wird.

- **Mikroskopisch vs makroskopisch:** Angenommen Modell A ist mikroskopisch und Modell B nach einem makroskopischen Ansatz entworfen. Will Modell B nun Ergebnisse aus Modell A verwenden – egal ob dynamisch oder linear – so müssen die Größen meist aggregiert werden. Umgekehrt ist es bei Verwendung von Teilen aus Modell B in Modell A notwendig über probabilistische Steuerungsmechanismen zu disaggregieren.
- **Black-Box vs White-Box:** Hier ist vor allem darauf zu achten welche Teile, dass Modellteile die im Black-Box Modell verwendet werden nicht automatisch im White-Box Modell verwendet werden können. Beispielsweise kann ein Parameter für das Black-Box Modell aufgrund einer Vereinfachung

Bei den Begrifflichkeiten aus den letzten Absätzen handelt es sich natürlich um Extremfälle. Meist kann man bei der Entwicklung von Modellen nicht strikt von der einen oder anderen Struktur ausgehen. Beispielsweise gibt es zwischen Mikroskopisch und Makroskopisch auch noch die Möglichkeit einer mesoskopischen Betrachtungsweise. Es sei jedoch bemerkt, dass auch bei der Kopplung eines mesoskopischen Modells mit einem anderen Modell Größen aggregiert bzw. disaggregiert werden müssen, da bei verschiedenen Simulationen selten derselbe Detailgrad vorliegt.

Die wichtigsten Umschaltmechanismen bei einer linearen Kopplung werden hier noch kurz beschrieben. Für detailliertere Informationen sei auf [13] verwiesen:

- **Ereignisgesteuert:** Ein bestimmtes Ereignis tritt in System A auf.
- **Zustandsgesteuert:** System A befindet sich in einem gewissen Zustand.
- **Zeitgesteuert:** Nach Ablauf einer gewissen Zeit.
- **Zufallsgesteuert:** Zu einem zufälligen Zeitpunkt.

5 Kausalketten

- In Übereinstimmung mit der archäologischen Fragestellung die hier gestellt wird, müssen kausale Ketten, sowohl auf der strukturellen als auch auf der Simulationsebene dargestellt werden. Der hier gewählte Ansatz liegt darin alle Kanten, die

von oder zu einer vorausgewählten Menge von Punkten führen, hervorzuheben. Dies beeinflusst die Knoten in der direkten Nachbarschaft und dadurch werden kausale Ketten sichtbar. Dies wird in der Abbildung 4 gezeigt.

5.1 Umittelbare Nachbarschaft

- Angenommen es wird explizit die Einheit „mine“ auf der Strukturellen Ebene gewählt: Nachdem dies dem Punkt „Mining Model“ auf der Simulationsebene „entspricht“, wird auch dieser implizit ausgewählt. Formeller ausgedrückt; eine beliebige Menge von Knoten kann auf beiden Ebenen ausgewählt werden, während die dazugehörigen Knoten der anderen Ebene ebenfalls ausgewählt werden.
- Für alle ausgewählten Knoten auf der Strukturebene werden die vorangehenden und darauffolgenden Einheiten hervorgehoben: Die ersten sind Ursachen, die zweiten Effekte. Zum Beispiel (siehe links auf der Abbildung 4), kann gesagt werden, dass eine „mine“ von der „Mining Community“ beeinflusst wird, die eine Anzahl von Bergarbeitern zur Verfügung stellt. Auf Seiten der Effekte wird das Salz, das produziert wird, für die weitere „Surrounding Area“ zugänglich gemacht.
- Die Simulationsebene (rechte Seite der Abbildung 4) sieht der Strukturebene ähnlich, nachdem auch hier die vorangehenden und nachfolgenden Punkte hervorgehoben werden. Allerdings handelt es sich auf dieser Ebene um Modelle, Daten und Experimente statt Einheiten. Von einem ausgewählten Punkt aus werden alle vorangehenden und nachfolgenden Punkte, die kein Teil eines Experimentes sind, hervorgehoben. Wenn während dieses Prozesses (1.) die Strukturebene durchbrochen wird und (2.) der betrachtete Punkt keine Einheit darstellt, dann müssen wir weitere Vorgänger oder Nachfolger miteinbeziehen, bis wieder eine Einheit gefunden wird.
- Ursachen und Effekte können spezielle Prozesse bilden (Zyklus im Graph). Zum Beispiel ist in Abbildung 4 zu sehen, dass eine Bergbaugesellschaft Mienenarbeiter für eine Miene zur Verfügung stellt, was daraufhin einer Umgebung Salz zugänglich macht, in welcher die Bergbaugesellschaft lebt. Dies ist ein Beispiel für einen spiral-förmigen Prozess auf unterschiedlichen Ebenen, d.h. der Kreislauf ist geschlossen, da die Berg-

baugesellschaft in der Einheit „Surrounding Area“ verankert ist.

- Ein einfacheres Beispiel kann ein Zyklus auf derselben Ebene sein, wie zum Beispiel im „Mining Community Experiment“ – „initial Population“ – „Demographics Model“ – „Number of Miners“ – zeigen kann. Dies ist ein typischer Aufbau, bei welchem ein Modell von einem Experiment gelenkt (betrieben) wird. Andere Fälle von Zyklen auf derselben Ebene und demselben Level erzeugen Feedback-Schleifen, wie sie oft in System Dynamics benutzt werden (wird in Abbildung 4 nicht gezeigt).
- Ecken (und somit auch Zyklen), die zu demselben Punkt führen, können als „Entailment mesh“ Gordon Pask gesehen werden: „Eine Aussage kann aus unterschiedlichen Gedankengängen abgeleitet werden“. Jedenfalls müssen diese sich, um zulässig zu sein, im Bezug auf Ergebnisse oder Fakten, die benutzt werden, nicht widersprechen. Somit kann eine Plausibilitätskontrolle (Face Validation[7]) von Experten den Fokus genau auf diesen Punkt legen – die Wahrung der Beständigkeit und die Mehrfachprüfung der Ergebnisse, die durch unterschiedliche Mittel hergeleitet wurden. Dieser Aspekt ist einer der Punkte an denen in Zukunft noch gearbeitet werden muss

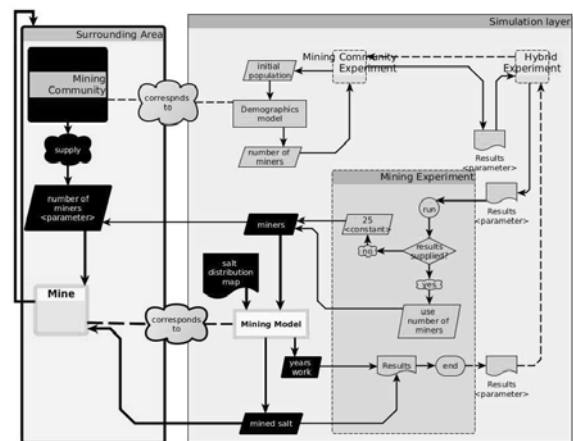


Abbildung 4: Kausalketten

5.2 weitere Kausalkette

Jeder andere Punkt, der eine kausale Beziehung hat (z.B. gerichteter Pfeil, der zu einer der Einheiten führt und in einer direkten Beziehung mit den ausgewähl-

ten Punkten steht) ist auch Teil eines weiteren kausalen Netzwerks. Das gesamte Gebiet, das von der ursächlichen Einheit abgedeckt wird, wird von den verbundenen Komponenten der ausgewählten Punkte aufgespannt. Normalerweise gibt es generell viel weniger kausal verbundene Punkte, womit leichter zu sehen ist „was sich ändern könnte, wenn ein einziger Punkt (z.B. ein Parameter), der auf kausale Weise verbunden ist, sich ändert“.

Diskussion

Aufgrund des begrenzten Platzes konnte hier leider weder auf alle Aspekte noch auf die – oft notwendigen Details – eingegangen werden. Es gibt bereits bestehende Arbeiten, die den Aspekt der Reproduzierbarkeit und im speziellen die Validierung und Verifikation von einzelnen Simulationen betrifft (siehe [7], [14]). Eine genauere Untersuchung dieser Aspekte bei der Verknüpfung von Simulationen ist noch ausständig und wird in naher Zukunft weiter erforscht werden.

Fazit

Was diese Vorgehensweise auszeichnet ist, dass sowohl die Gesamtstruktur des betrachteten Systems abgebildet wird als auch – durch das auf und zuklappen der Einheiten – ein Blick in die innere Struktur der Simulationen und Entitäten gegeben wird. Durch die enge Kooperation zwischen den beteiligten Wissenschaftlern verschiedenster Fachrichtungen an diesem Projekt wird gewährleistet, dass möglichst viele Blickwinkel auf ein und dasselbe Thema gewährt werden. Die vorgestellte Struktur ermöglicht nun zusätzlich auch noch jedem Mitglied der Forschungsgruppe die Ergebnisse der eigenen Forschung zu verorten und sie im Bezug auf das Gesamtsystem der Archäologischen Stätte Hallstatt zu sehen.

References

- [1] N. Wiener, *Cybernetics*. Hermann Paris, 1948.
- [2] L. von Bertalanffy, *General system theory: Foundations, development, applications*. Braziller. New York, 1968.
- [3] T. A. Kohler and G. J. Gummerman, *Dynamics of human and primate societies: agent-based modeling of social and spatial processes*. Oxford University Press, 2001.
- [4] J. G. Polhill, D. Parker, D. Brown, and V. Grimm, “Using the ODD Protocol for Describing Three Agent-Based Social Simulation Models of Land-Use Change.” [Online]. Available: <http://jasss.soc.surrey.ac.uk/11/2/3.html>. [Accessed: 24-Jun-2014].
- [5] V. Grimm, U. Berger, D. L. DeAngelis, J. G. Polhill, J. Giske, and S. F. Railsback, “The ODD protocol: A review and first update,” *Ecol. Model.*, vol. 221, no. 23, pp. 2760–2768, Nov. 2010.
- [6] K. Kowarik, H. Reschreiter, G. Wurzer, R. Totschnig, and A. Rausch, *Agenten im Hallstätter Salzbergwerk. Agentenbasierte Simulation für den bronzezeitlichen Salzbergbau in Hallstatt/OÖ*. na.
- [7] O. Balci, “Validation, verification, and testing techniques throughout the life cycle of a simulation study,” *Ann. Oper. Res.*, vol. 53, no. 1, pp. 121–173, Dec. 1994.
- [8] I. Hafner, M. Rößler, B. Heinzl, A. Körner, M. Landsiedl, F. Breiteneker, and C. Reinisch, “Possibilities and limits of co-simulating discrete and continuous models via the building controls virtual test bed,” in *Proceedings of the European Modeling and Simulation Symposium, 2012*, Vienna, 2012, pp. 495–500.
- [9] S. Tisue and U. Wilensky, “Netlogo: A simple environment for modeling complexity,” in *International Conference on Complex Systems*, 2004, pp. 16–21.
- [10] F. Breiteneker, M. Bicher, and G. Wurzer, *Agent-based modeling and simulation in archaeology, chapter: Agent-Based Simulation in Archaeology: A Characterization*. [S.l.]: Springer, 2013.
- [11] J. W. Forrester and P. M. Senge, “Tests for Building Confidence in System Dynamics Models,” in *System dynamics*, vol. 14, A. Legasto, J. W. Forrester, and J. M. Lyneis, Eds. Amsterdam, New York, Oxford: North-Holland Publishing Company, 1980, pp. 209–228.
- [12] F. Breiteneker, “Physical Modelling and Statechart Modelling in System Simulation: Challenges, Advantages, Disadvantages and Risks,” presented at the UKSim 2011, Cambridge, 2011.
- [13] *Simulation mit ACSL Eine Einführung in die Modellbildung, numerischen Methoden und Simulation*. Wiesbaden: Vieweg+Teubner Verlag, 1993.
- [14] R. Sargent, “Verification and validation of simulation models,” in *Proceedings of the 2010 Winter Simulation Conference*, Baltimore, MD, 2010, pp. 166–183.

Agentenbasierte Simulation zur Versorgung der Bevölkerung im prähistorischen Hallstatt – Ausgewählte Arbeitsabläufe und deren Zusammenhang

Johannes Tanzler¹, Philipp Pichler², Klaus Fankhauser¹, Dominik Brunmeir¹, Kerstin Kowarik³, Hans Reschreiter³, Anke Bacher⁴

¹Institute for Analysis and Scientific Computing, Vienna University of Technology, Austria

²dwh GmbH Vienna, Austria

³Natural History Museum Vienna, Austria

⁴University of Vienna, Austria

johannes.tanzler@tuwien.ac.at

Dieses Projekt ist Teil einer Kooperation des Naturhistorischen Museum Wien mit der Technischen Universität Wien. Es wird mit Hilfe der agentenbasierten Simulation versucht einen Zusammenhang zwischen einzelnen Arbeitsabläufen des prähistorischen Salzabbaus Hallstatts, sowie manche Arbeitsabläufe selbst zu simulieren. Im Speziellen wird der Fokus auf die Nahrungsmittelproduktion und deren Zeitaufwand gelegt. Dieser Zeitaufwand setzt sich zusammen aus dem für die landwirtschaftlichen Tätigkeiten und den Wegzeiten um zu den Feldern und wieder zurück zu gelangen. Dafür wird ein A* Algorithmus verwendet. Weiter wird die Beschaffung von Leuchtspänen berücksichtigt, welche in der Salzmine als Beleuchtung dienen. Diese Simulation liefert eine Zeitdauer welche eventuell im Bergwerk gearbeitet werden kann und nicht in der Landwirtschaft aufgewendet werden muss.

1 Einleitung

Hallstatt ist bekannt für seine prähistorische Salzmine, welche für Archäologen von großem Interesse ist. Das besondere Interesse rührt daher, dass durch die konservierende Wirkung des Salzes und einem Einsturz des Bergwerks, viele gut erhaltene Gegenstände in dem Bergwerk ausgegraben werden können. Im Zuge dieser Ausgrabungen wurden Gegenstände gefunden, deren Handhabung nicht gänzlich klar ist. Um diese besser zu verstehen, werden neben den klassischen Methoden der Archäologie, in einer Kooperation des Naturhistorischen Museums Wien mit der Technischen Universität Wien, auch die Möglichkeiten der Computersimulation genutzt. Ein Beispiel dafür ist der Bronzepickel, welcher einen ungewöhnlichen Winkel zwischen Spitze und Schaft aufweist [1].

Diese Arbeit behandelt allerdings nicht die Handhabung einzelner Gegenstände, sondern stellt einen ersten Schritt zur Simulation des Zusammenhangs der einzelnen Arbeitsabläufe dar. Im speziellen wird die Nahrungsmittelproduktion und, in Ansätzen, der

Bergwerksbetrieb mit Hilfe agentenbasierter Simulation[2] simuliert. Die Vorarbeit für dieses Projekt leistete eine Arbeit, welche sich mit der Versorgung der damaligen Hallstätter Bevölkerung mit Nahrungsmitteln beschäftigt [3]. Ein Ergebnis dieser Arbeit ist eine mögliche Bevölkerungsgröße mit 72 Personen, welche auch im Folgenden verwendet wird. Gerade die Frage der Bevölkerungsgröße ist von großem Interesse, da sie in direktem Zusammenhang mit der Menge des produzierten Salzes steht. Mit dieser Arbeit wird versucht herauszufinden, wie viel Zeit die damalige Hallstätter Bevölkerung im Bergwerk verbringen konnte und nicht mit der Landwirtschaft oder Holzbeschaffung für die Beleuchtung des Bergwerks beschäftigt war. Für die Beantwortung dieser Fragen wird als Programmierungsumgebung Anylogic[4] verwendet.

2 Modellbeschreibung

Die Grundlage jeder landwirtschaftlichen Produktion stellen die dafür verwendbaren Flächen dar.

2.1 Flächen

Da dieses Projekt Fragestellungen bezüglich des prähistorischen Hallstatt behandelt, liegt es auf der Hand, dass keine Karten vorliegen, welche die Flächennutzung der damaligen Zeit widerspiegeln. Aus diesem Grund wird eine Karte aus der Kaiserzeit verwendet, welche aus steuerlichen Gründen erstellt wurde. Diese Karte wird als Shapefile [5] eingebunden und dient der Lokalisierung der einzelnen Flächen, der Bestimmung der Feldnutzung, sowie deren Schwerpunkte und Größen. Die Schwerpunkte dienen als Startpunkt zur Bewirtschaftung der jeweiligen Flächen sowie als Standort des Speichers im Falle einer Zwischenspeicherung am Feld. Die im Folgenden relevanten Flächennutzungsarten sind Acker, Garten und Grünland. Die Fläche des Ackers beträgt rund 13 Hektar wobei der Acker zur Hälfte mit Gerste und zur anderen Hälfte mit Hirse bestellt wird. Der Garten hat eine Gesamtfläche von rund 16 Hektar und wird ausschließlich mit Ackerbohnen bestellt. Das Grünland hat insgesamt über 650 Hektar, wobei hier angenommen wird, dass 100 davon als Heu eingebracht werden müssen.

Anylogic bietet derzeit nicht die Möglichkeit, auf einfache Art und Weise mit dreidimensionalen Shapefiles zu arbeiten, weshalb die Höhendaten mit einem Esri Ascii grid eingebunden werden.

2.2 Berechnung der Wege

Auf Grund des mitunter sehr steilen Geländes in Hallstatt ist es wichtig, einen guten Algorithmus für die Berechnung der Wege zu verwenden. Der hierfür ausgewählte Algorithmus ist ein A* Algorithmus. Da die Karte aus über 300 einzelnen Feldern besteht und für die meisten davon Wege berechnet werden müssen, wird die Rechenzeit ein zu berücksichtigender Punkt. Um sie möglichst gering zu halten, wird ein Raster von 25 Metern für die Berechnung verwendet.

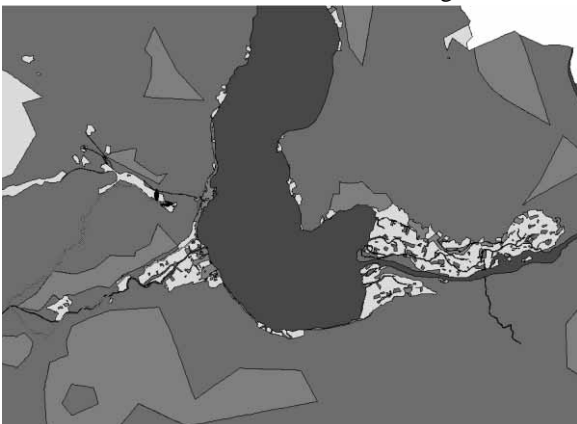


Abbildung 1. Darstellung eines Weges

In Abbildung 1. sieht man einen Weg als Polygon eingezeichnet. Wie man sieht, wird nicht der direkte Weg vom Startpunkt zum Ziel gewählt, sondern ein großer Umweg in Kauf genommen, was dem großen Gefälle geschuldet ist. Besser kann man die Steilheit des Geländes durch die eingefärbte Höhenkarte erkennen.

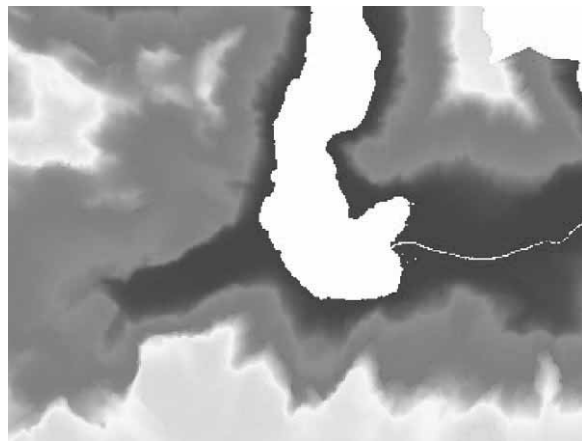


Abbildung 2. Höhenkarte

Die Geschwindigkeit, mit welcher sich die Agenten entlang des berechneten Weges fortbewegen, ist vom gewählten Weg und dessen Steigungen abhängig.

2.3 Bevölkerung

Die Geburten- und Sterberaten sind mit leichten Abänderungen dem Programm zu „Mining with Agents“ [6] entnommen. Aus diesen wird eine Bevölkerungsstruktur berechnet. Wie schon oben erwähnt, wird von einer Bevölkerungsgröße von 72 Personen ausgegangen. Nach oben erwähntem Projekt kann nämlich diese Menge an Menschen vom umliegenden Land mit Nahrungsmitteln versorgt werden. Von diesen 72 Personen befinden sich allerdings nicht alle in einem arbeitsfähigen Alter, welches im Bereich von 6 bis 75 Jahren angenommen wird. Die in nachfolgender Tabelle dargestellte Bevölkerung wird für alle weiteren Berechnungen verwendet und weist 13 Personen auf, welche sich nicht im arbeitsfähigen Alter befinden.

[0;6)	[6;15)	[15;30)	[30;45)	[45;75)	[75; Inf)
13 Pers.	20 Pers.	21 Pers.	11 Pers.	7 Pers.	0 Pers.

Tabelle 1. Bevölkerungsstruktur

2.4 Arbeitsabläufe

Der am genauesten modellierte Ablauf ist die Wegberechnung, welche schon oben beschrieben wurde. Des Weiteren werden auch das Bestellen und Abernten der Felder, und die Holzbeschaffung für die Beleuchtung im Bergwerk betrachtet. Die Dauer der Aussaat für die verschiedenen Arten an Feldern wird gleich, mit 10 Stunden pro Hektar, angenommen. Für die Ernte wird die Zeitspanne, welche notwendig ist, um einen Hektar zu ernten, ebenfalls für alle Feldtypen gleich, mit 20 Stunden, angegeben. Der Transport der Ernte zur Siedlung wird mit Säcken zu 20 kg bewältigt. Hier werden 2 Szenarien unterschieden, einerseits das sofortige Einlagern in der Siedlung und andererseits das Aufbewahren der Ware am Feld, um sie nach der Ernte in die Speicher der Siedlung zu transportieren. Was den Salzabbau betrifft, wird angenommen, dass ein Arbeiter durchschnittlich 0.01 kg Salz pro Minute abbauen kann und zur Beleuchtung für den Abbau 10 kg Salz 1 kg Holz benötigt. Die Beschaffung des Holzes wird in 2 Teile gegliedert: einerseits in das Fällen des Baumes, das Zerkleinern in 20 kg Stücke sowie deren Transport, andererseits in das Herstellen der Späne. Das Herstellen der Späne dauert 1 Minute für 1kg Späne. Das Zerkleinern nimmt 1 Minute für 10 kg in Anspruch.

2.5 Auswahl der durzuführenden Arbeiten

Die Abfolge der Arbeitsabläufe, speziell die der landwirtschaftlichen Produktion, hängt stark von der Jahreszeit ab. Die Werte beschreiben den optimalen Tag für die jeweilige Tätigkeit in Tagen von Jahresbeginn an.

	Hirse	Gerste	Bohnen	Gras
Saat/ Mähen	110	90	100	180
Ernte	270	240	250	185

Tabelle 2. Aktionszeitpunkte

In weiterer Folge bilden diese Werte die Grundlage für die Reihung der Arbeitsabläufe. Allerdings muss auch die Salzproduktion dabei berücksichtigt werden. Deshalb werden zwei Szenarien berücksichtigt:

Szenario 1. Priorität Salzproduktion

In diesem Fall wird der Salzproduktion der Vorzug gegenüber der Landwirtschaft gegeben. Es wird davon ausgegangen, dass immer eine gewisse Anzahl an Arbeitern im Bergwerk verbleibt und somit weniger

Arbeiter für die landwirtschaftliche Produktion zur Verfügung stehen.

Szenario 2. Priorität Landwirtschaft

Dieses Szenario sieht vor, dass das Hauptaugenmerk auf der landwirtschaftlichen Produktion liegt. Es verbleiben keine Arbeiter im Bergwerk und die landwirtschaftlichen Tätigkeiten können schnellstmöglich erledigt werden. Speziell im Falle der Ernte scheint dies sehr wichtig, damit so wenig Nahrungsmittel wie möglich schon am Feld verderben.

3 Resultate

Im Folgenden werden die zwei oben genannten Szenarien verglichen. Für jede dieser beiden Szenarien wird auch die Möglichkeit eines Feldspeichers in Betracht gezogen. Die Resultate beschreiben die Zeit, welche im Bergwerk verbracht werden konnte und nicht für die Landwirtschaft aufgewendet werden musste. Als Länge eines Arbeitstages werden 8 Stunden angenommen, wobei kein Urlaub oder sonstiger freier Tag berücksichtigt wird.

3.1 Priorität Landwirtschaft

In diesem Fall ergibt sich, mit den oben genannten Parametern und ohne Feldspeicherung, eine im Salzberg verbrachte Zeit von rund 116°576 Stunden. Dies bedeutet durchschnittlich für jeden einzelnen Bewohner im arbeitsfähigen Alter eine Arbeitszeit im Bergwerk von rund 1°976 Stunden. Im Falle der Speicherung am Feld beträgt die Stundenzahl im Bergwerk lediglich 71°172.

3.2 Priorität Salzproduktion

Hier wird von einem ständigen Verbleib im Bergwerk von 10 Arbeitern tagsüber ausgegangen. Ohne Feldspeicher ergeben sich hier 111°086 Stunden im Bergwerk, was überraschenderweise weniger ist als ohne ständige Präsenz im Berg. Wenn auch hier ein Feldspeicher verwendet wird, verringert sich die verbrachte Zeit im Bergwerk auf 71°047 Stunden.

Diese Ergebnisse sind überraschend und verlangen nach einer genaueren Validierung des Programms.

4 Conclusio und Ausblick

Wie schon im letzten Absatz angedeutet sollten diese Ergebnisse nicht als Tatsache angesehen werden. Diese Projektarbeit ist noch nicht abgeschlossen und wird weiter ausgebaut, weshalb auch noch keine genaue Validierung vorgenommen wurde. Die hier

vorgestellte Modellierung muss weiter verfeinert und Parameter müssen besser bestimmt werden. Eine mögliche Erweiterung könnte die Speicher betreffen, welche nicht mehr direkt auf den jeweiligen Feldern positioniert sein könnten, sondern an logistisch gut gewählten Standorten aufgestellt werden. Weiters könnte ein Vergleich zwischen verschiedenen Standorten der Siedlung gezogen werden. Es ist auch denkbar, dass es mehrere Siedlungen gegeben hat und ein Teil der Bevölkerung im Tal gelebt hat.

5 References

- [1] B. Heinzl, E. Auer, B. Slowacki, K. Kowarik, H. Reschreiter, F. Breitenecker. *Mathematical modelling for experimental archaeology: case studies for mechanical tools in hallstatt salt mines*. In: The 24th European Modeling and Simulation Symposium. 2012.
- [2] F. Breitenecker, M. Bicher, and G. Wurzer, *Agent-based modeling and simulation in archaeology, chapter: Agent-Based Simulation in Archaeology: A Characterization*. [S.l.]: Springer, 2013.
- [3] J. Tanzler, P. Pichler, K. Kowarik, H. Reschreiter, G. Wurzer, A. Bacher, F. Breitenecker. *Modelling agricultural constraints for population size in prehistoric Hallstatt*. ERK - International Electrotechnical and Computer Science Conference. 2013.
- [4] I. Grigoryev: *AnyLogic 6 in three days: a quick course in simulation modeling*. AnyLogic North America, Hampton, NJ, 2012.
- [5] ESRI Shapefile Technical Description, An ESRI White Paper-July 1998.
- [6] K. Kowarik, H. Reschreiter, G. Wurzer, R. Totschnig, A. Rausch: *Mining with Agents - Agent-based Modeling of the Bronze Age Salt Mine of Hallstatt*, Talk: Archäologie und Computer, Wien; 11-03-2008 - 11-05-2008; in: "Workshop 13 - Archäologie und Computer 2008", CD der Stadt Wien, Stadtarchäologie, Wien (2009), ISBN: 978-3-85161-016-1; 1 - 19.

Anbindung eines dynamischen Modells zur Plattentektonik mit dem ArcMap-Werkzeug ModelBuilder

Jochen Wittmann¹, Karsten Bratz¹

¹HTW Berlin, FB2, Umweltinformatik, Wilhelminenhofstr. 75A, 12459 Berlin

wittmann@htw-berlin.de

Die Plattentektonik basierend auf den Ideen von Alfred Wegener gilt als akzeptiertes Modell zur Erklärung der Dynamik der Erdoberfläche. Es existieren zahlreiche Visualisierungen der Plattenbewegungen, die ein anschauliches Bild der Bewegungen und der dynamischen Veränderungen der Kontinente geben. Bei genauerer Betrachtung sind diese Visualisierungen jedoch reine Animationen, denen direkt keine erklärenden Strukturmodelle zugrunde liegen. Auf der anderen Seite existiert mit der Arbeit von Rödder ein Simulationsmodell in Form eines C-Programmpaketes, das die Plattengrenzen aus einem mathematisch konsistent und vollständig beschriebenen mathematischen Modell ableitet, allerdings keine zeitgemäße Visualisierungskomponente umfaßt. In dieser Situation stellt sich die vorliegende Arbeit die Aufgabe, das mathematische Modell nach Rödder über eine GIS-Oberfläche verfügbar zu machen und die Modellergebnisse auf diese Weise zeitgemäß zu visualisieren. Dabei werden die einzelnen Funktionalitäten Ergebnispräsentation, Steuerung des Simulationsablaufes und Modifikation des Plattenmosaiks inklusive Grenztypen im Detail diskutiert und dargestellt. Das Paper zeigt damit prototypisch, wie ein Stand-alone-Programm, das auf einer Dateischnittstelle basiert, mit einem GIS-System (im Beispiel ArcMap) gekoppelt werden kann.

1 Das zu modellierende System: Plattentektonik

Unter dem Begriff Plattentektonik basierend auf den Ideen von Alfred Wegener [1] versteht man eine geotektonische Theorie, die dynamische Vorgänge in der Lithosphäre wie die Bewegung der Ozeanböden, die Entstehung von Gebirgen, die Erdbeben- und Vulkanzonen sowie die Kontinentalverschiebung zu erklären versucht. Laut dieser ist die Lithosphäre in mehrere, 100 - 150 km dicke, verschieden starre Platten zerlegt, die auf dem zähflüssigen Erdmantel gegeneinander bewegt werden. An den Mittelozeanischen Rücken weichen die Platten mit bis zu 6 cm pro Jahr auseinander, wobei aus den Dehnungs- bzw. Trennungsfugen vulkanisches Material aufsteigt, welches die ozeanische Kruste ständig ergänzt. Das Wachsen und Auseinanderdriften der Ozeanböden wird durch ein Abtauchen an den stabilen Rändern benachbarter Platten ausgeglichen. Solche Verschluckungszonen, so genannte Subduktionszonen finden sich im Bereich der Tiefseerinnen. Bewegen sich die Platten aufeinander zu, so entstehen Gebirge oder vulkanische Inselbögen. Schließlich können sich Platten horizontal aneinander vorbeibewegen, die so genannte Transform-Störung. Als Motor der Bewegung werden Wärmeströmungen im Erdmantel angenommen.

2 Das Modell

2.1 Das Erdmodell NUVEL (NUVEL-1)

Es handelt sich bei NUVEL-1 um ein Modell der Plattenkinematik von 1990 [2]. Entwickelt wurde es von Wissenschaftlern der Northwestern University (Evanston, USA). Ursprünglich beschreibt das Modell die relative Bewegung von jeweils zwei benachbarten Platten gegeneinander. In diesem Modell werden, in einem zweiten Schritt, die Bewegungen von zwölf berücksichtigten Platten zu einem konsistenten, globalen Modell zusammengefügt. Hierbei wird die größte Platte, die Pazifikplatte, festgehalten bzw. als in Ruhe befindlich angenommen. Da es sich um eine willkürliche Festlegung handelt, wird in einem dritten Schritt das Integral aller Bewegungen über die Erdoberfläche zu Null gemacht (no net rotation, NNR). 1991 entstand die zu dem Modell NUVEL-1 gehörige Lösung NNR-NUVEL-1. 1994 wurde das Modell NUVEL-1A veröffentlicht. In diesem Modell wurden Abweichungen zu den Beobachtungsdaten miteinbezogen. Hier werden die abgeleiteten Geschwindigkeiten der Meeresbodenausbreitung und damit der Plattenrotationen mit dem Faktor 0,9562 gegenüber NUVEL-1 multipliziert. NUVEL ist die Sammelbezeichnung für geophysikalische Erdmodelle, in denen die nachweisbaren Kontinentalverschiebungen von 1 bis 20 cm pro Jahr durch

eine dynamische Theorie der Plattentektonik beschrieben werden. Das Erdmodell kombiniert Daten aus 277 Spreizungsraten, 121 Transformationsrichtungen und 724 Erdbeben-Rutsch-Vektoren der Erde um die Bewegung der Erdplatten zu visualisieren. Es berücksichtigt die folgenden Erdplatten: Die eurasische Platte, australische Platte, pazifische Platte, nord- und süd-amerikanische Platte, Cocos-Platte, karibische Platte, antarktische Platte, arabische Platte, afrikanische Platte, indische Platte und die Nazca-Platte [2]. Es wird davon ausgegangen, dass die pazifische Platte sich nicht in Bewegung befindet und somit als Bezugsplatte dient. Die Bewegung zwischen den Platten wird durch einen Eulervektor, d.h. Bewegungsrichtung und -geschwindigkeit) dargestellt.

2.2 Das mathematische Modell

Es existieren zahlreiche Visualisierungen der Plattenbewegungen, die ein anschauliches Bild der Bewegungen und der dynamischen Veränderungen der Kontinente geben (z.B. in [3]). Bei genauerer Betrachtung sind diese Visualisierungen jedoch reine Animationen, denen direkt keine erklärenden Strukturmodelle zugrunde liegen. Rödder legt in dem Programm Artemis [4] ein geschlossenes mathematisches Modell auf der Basis von NUVEL-1 vor, das als erklärendes Strukturmodell weit mehr leisten kann, allerdings keine zeitgemäße Visualisierungskomponente besitzt. Die vollständige Beschreibung des mathematischen Modells soll der Arbeit von Rödder vorbehalten bleiben. An dieser Stelle reicht es aus, die Grundannahmen des Modellansatzes zu nennen:

Das Modell benötigt als Ausgangsdaten:

1. ein vollständiges und überlappungsfreies Plattenmosaik der Erdoberfläche
2. inklusive der jeweiligen Grenztypen (divergierend, konvergierend, konservativ) sowie
3. inklusive der Eulervektoren für die Platten des Mosaiks (Richtung und Geschwindigkeit der Bewegung jeder einzelnen Platte)

Ausgehend von diesen Daten kann das dynamische Verhalten lokal an den Plattengrenzen simuliert und ein neues, konsistentes und vollständiges Plattenmosaik berechnet werden. Dabei ergeben sich im Wesentlichen drei Fälle, die durch das Modell gelöst werden müssen und in Abb. 1 - Grenztypen veranschaulicht sind:

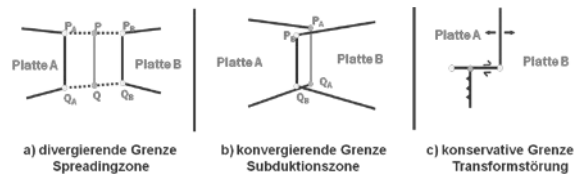


Abb. 1: Grenztypen (nach [5])

Bei einer Spreadingzone muß das neu entstandene Plattenmaterial konsistent den beiden divergierenden Platten zugeordnet werden. Bei einer Subduktionszone muß die Plattengrenze der subduzierten Platte korrigiert werden. Bei einer Transformstörung muß der Grenzpunkt entsprechend angepaßt werden.

Größere Probleme bereiten die sogenannten Tripelpunkte, also Orte des Zusammentreffens von drei Erdplatten. Da es drei Arten von Plattenrändern (divergent, konvergent, transform) gibt, kann es zehn möglichen Arten von Tripelpunkten geben. Wie die Dynamik an diesen Punkten modelliert ist, entnehme man wiederum der Originalarbeit von Rödder.

3 Die Kopplung

3.1 Modellschnittstelle

Das beschriebene mathematische Modell wurde direkt in der Programmiersprache C realisiert. Die Arbeit mit dem Modell wurde jedoch nach den Prinzipien einer Simulationsumgebung [6] strukturiert, so daß die einzelnen Komponenten modular zugänglich und änderbar bleiben:

- a) Die Dynamikbeschreibung liegt als reiner C-Code vor.
- b) Plattengeometrie und Eulervektoren werden als separate ascii-Files abgelegt.
- c) Die Steuerparameter für die Simulationssläufe werden ebenfalls in einer separaten ascii-Steuerdatei spezifiziert.

Aus diesen Inputs generiert der Simulator

- a) Eine Protokolldatei
- b) Zu jedem unter c) spezifizierten Aufzeichnungspunkt eine Datei mit dem jeweiligen simulierten Plattenmosaik.

3.2 Konzept der Kopplung

Dieses isoliert zu betreibende C-Programm soll nun mit dem Geoinformationssystem ArcMap [7] gekoppelt werden, um die Bedienung zu vereinfachen und

weg von der Dateischnittstelle auf eine graphische Ebene zu heben.

Die Kopplung des isoliert ablauffähigen C-Programms mit seiner beschriebenen Input-/Output-Schnittstelle kann in drei Einzelfunktionalitäten beschrieben werden, die eine Vorgängerarbeit im Detail beschreibt und begründet [8]. In den folgenden Abschnitten sollen daher an dieser Stelle nur die drei grundlegenden Hauptaufgaben der Kopplung referiert werden. Für alle Schritte ist zunächst eine Geodatabase in ArcMap anzulegen. Als Basiskarte wird hier die World Physical Map, angeboten von ESRI selbst [9], verwendet.

zur Ergebnisvisualisierung:

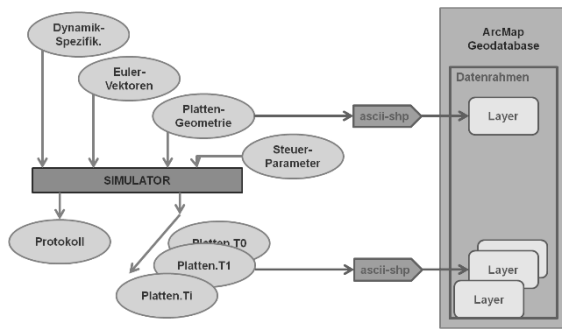


Abbildung 2: Kopplungskonzept zur Ergebnisvisualisierung

In diesem Fall geht es im Wesentlichen darum, das Ausgangs-Plattenmosaik sowie diverse Zwischenzustände darzustellen, die als Dateien mit der Bezeichnung „Platten.Ti“ vorliegen, wobei „Ti“ den Aufzeichnungszeitpunkt im Modell angibt.

Die Koordinaten von Grenzverläufen einzelner Plattengrenzen können dabei den entsprechenden Dateien des Simulators entnommen werden. Die verwendeten Daten sind in einer CSV (Comma-separated Values) Datei gespeichert. Sie werden zunächst in ein Excel-Tabellen-Format übertragen und nach einigen Anpassungen im Format als Layer in ArcGis importiert. Für jede Platte des Modells sind dann Anfangs- und Endpunkt eines Grenzverlaufs, Grenztyp, sowie die Bezeichnung der Platte notiert. Zusammen mit der Weltkarte als Basis lassen sich nun bereits die einzelnen Zeitpunkte als Layer darstellen. Die Platten und ihre Grenzen inklusive Grenztyp sind dabei als einzelne Objekte erfasst und entsprechend manipulierbar bzw. symbolisierbar.

zur Simulationssteuerung:

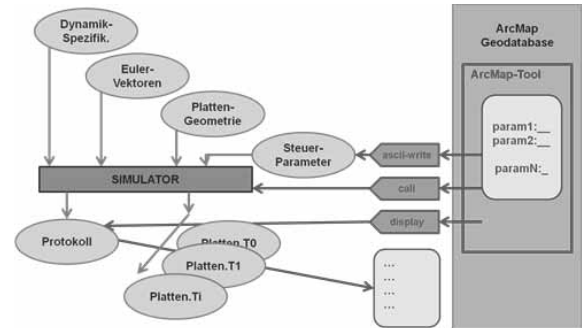


Abbildung 3: Kopplungskonzept zur Simulationssteuerung

Für diese Teilaufgabe ist es notwendig,

- Einen Editor mit Schreibzugriff für die Datei mit den Simulator-Steuerparametern aufzurufen.
- Den Simulator zu starten.
- Einen Editor mit Lesezugriff für die Protokoll-Datei aufzurufen.

Sämtliche Funktionalitäten lassen sich als nutzerorientierte Skripte unter ArcMap realisieren.

zur Modellmodifikation:

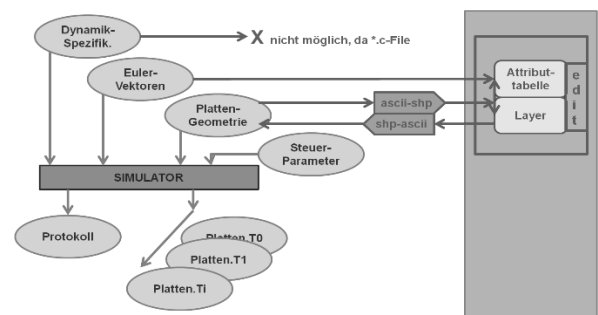


Abbildung 4: Kopplungskonzept zur Modellmodifikation

An dieser Stelle tritt eine wesentliche Einschränkung des hier vorgestellten Kopplungskonzeptes zu Tage: Änderungen an der Dynamikbeschreibung des Modells sind nicht vorgesehen, da diese einen Eingriff in den C-Code des Simulationssystems bedeuten und nur von Expertenhand durchgeführt werden sollten. Diese Einschränkung ist jedoch auf die Konzeption des zugrundeliegenden Simulators zurückzuführen und ist

nicht im hier exemplarisch beschriebenen Kopplungskonzept begründet. Abgesehen von dieser Einschränkung erlaubt das Konzept sämtliche Modifikationen, die über die Dateischnittstelle des Basissystems abgebildet werden können. Es sind dies:

- a) Eine Modifikation der Eulervektoren: Dabei ist es wiederum notwendig, die Steuerdatei mit den Eulervektoren als Tabelle ins GIS zu importieren. Dort können die Werte nach Wunsch geändert werden, und anschließend muss die Tabelle zurück ins Simulator-Format und an die korrekte Stelle im Simulator Dateibaum exportiert werden.
- b) Eine Modifikation der Plattengrenzen: Wie unter 4.2.1. beschrieben wird die Datei mit den Plattengrenzen ins GIS importiert. Dort können nun sowohl Grenzlinien als auch Grenztypen nach Wunsch geändert werden. Anschließend muss die Tabelle zurück ins Simulator-Format und an die korrekte Stelle im Simulator Dateibaum exportiert werden.

Beide Funktionalitäten sind wiederum als benutzereigene Skripte umsetzbar. Bei beiden Typen von Modifikationen ist es jedoch zusätzlich notwendig, die semantischen Restriktionen des Modells zu beachten. Beispielsweise muss das Plattenmosaik vollständig und überlappungsfrei die Erdoberfläche bedecken, bestimmte Kombinationen von Grenztypen für benachbarte Plattengrenzen sind inkompatibel, usw. Diese Restriktionen werden zwar spätestens beim Einlesen durch den Simulator erkannt und führen zu entsprechenden Fehlermeldungen, im Hinblick auf eine benutzerfreundliche Bedienung sollten sie jedoch bereits in unmittelbarem Zusammenhang mit den getätigten Modifikationen sichergestellt werden. Tatsächlich ist dies möglich, indem im GIS entsprechende semantische Restriktionen als Elemente der Topologie der Plattenlayer realisiert werden.

4 Die Umsetzung in ArcMap und Arc-ModelBuilder

In Anlehnung an die theoretische Vorlage wurden in der Implementation in ArcMap, mithilfe des Arc-ModelBuilders drei Funktionen geschaffen, die eine grundlegende Arbeitsweise mit Artemis ermöglichen.

4.1 Das Skript "OpenModel"

Die erste Funktion OpenModel, besitzt die Aufgabe, das ausgewählte Model in ArcGIS zu importieren.

Dazu werden sowohl alle Plattendaten sowie die Modellparameter geladen. Ähnlich wie in der ursprünglichen Methode aus Artemis liest OpenModel ein gewünschtes Grundmodell mit Standardparametern ins GIS ein. Darin enthalten sind sowohl Layerdaten als auch für das Ausgangsmosaik tektonischer Platten als auch die nötigen Eulervektoren und Simulationsparameter. Nachdem diese Funktion aufgerufen ist, kann auf sämtliche genannte Datenbereiche unter der Oberfläche des GIS lesend und schreibend zugegriffen werden. Damit ist die Anforderung nach einer komfortablen Modellmodifikation erfüllt.

Abb. 5 zeigt den grundsätzlichen Aufbau visualisiert durch die graphischen Symbole des ModelBuilders. Die einzelnen Arbeitsschritte sind modular-hierarchisch realisiert und arbeiten in ihrer Kapsel entsprechend eine Vielzahl von einzelnen GIS-Funktionen ab. Auf eine detailliertere Sicht muss an dieser Stelle allerdings verzichtet werden, es sei diesbezüglich auf die Arbeit von Bratz [10] verwiesen.

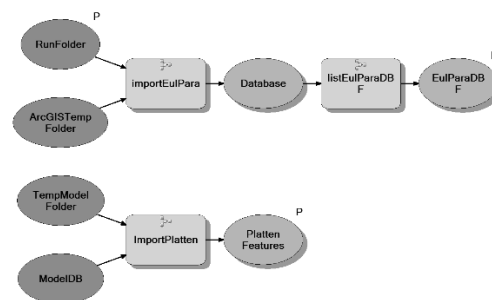


Abbildung 5: Das Skript "OpenModel" im ArcMap ModelBuilder (nach [10])

4.2 Das Skript "NewRun"

Diese Funktion dient zur Simulationssteuerung, insbesondere zur konsistenten Startparametrisierung eines einzelnen Simulationslaufes. Dazu werden die im GIS angezeigten (und ggfs. vom Benutzer interaktiv geänderten) Parameterwerte sowie die geometrische Daten am gewählten Model vom GIS ausgelesen und mit Hilfe mehrerer Python-Skripte angepasst, damit sie in dem für den Artemis-Simulator notwendigen Format vorliegen. Anschließend wird (ebenfalls über ein Python-Skript) die Simulation gestartet. Im letzten Schritt werden die im Artemis-Format vorliegenden Simulationsergebnisse in ArcMap importiert und anschließend visualisiert.

Abb. 6 zeigt wiederum den Aufbau der Funktion in ModelBuilder-Symbolik.

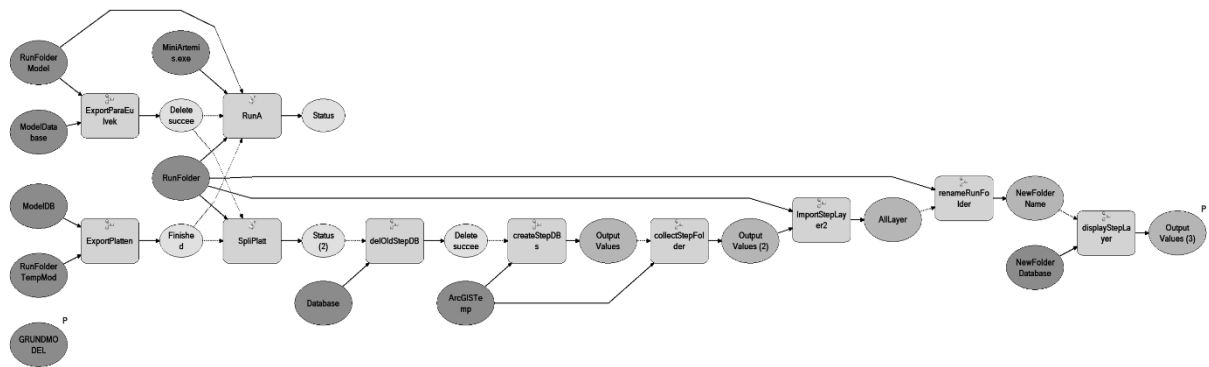


Abbildung 6: Das Skript „NewRun“ im ArcMap ModelBuilder (nach [10])

4.3 Das Skript „NewModel“

Die dritte Funktion ist ebenfalls stark an die Bedienung von Artemis angelehnt und setzt das dort angebotene Kommando CreateModel um. Dieses ermöglicht es, vollständig neue Anfangssituationen anzulegen, oder aber diese aus beliebigen, bereits simulierten Läufen zusammenzustellen.

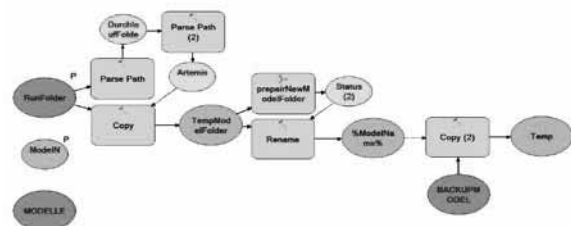


Abbildung 7: Das Skript "NewModel" im ArcMap ModelBuilder (nach [10])

5 Ein Beispiel

Das Simulationsmodell soll herangezogen werden, die Entwicklung des Plattenmosaiks über die letzten 20 Millionen Jahre bis heute zu simulieren und wie beschrieben unter Verwendung von ArcMap graphisch darzustellen. Die einzelnen Layer stehen hier als Zeit in Millionen von Jahren. Zur einfacheren Unterscheidung wurden die jeweiligen Schritte in unterschiedlicher Linienstärke dargestellt – Breite 1 für vor 20 Millionen Jahren bis Breite 4 für das jetzige Zeitalter. In Abbildung 5 - Die Erde vor 20 Mio. Jahren bis heute wird das Ergebnis dieser Arbeit vorgestellt. Dabei sind die einzelnen Grenztypen farblich unterschiedlich markiert, was leider in der hier nur möglichen Auflösung nur schwer darzustellen bzw. zu erkennen ist. Gut erkennbar sind jedoch die Bewegungsrichtung der

Erdplatten, sowie die Veränderung der Grenztypen an einigen Stellen der Plattengrenzen.

6 Zusammenfassung und Ausblick

Das vorliegende Paper zeigt, wie eine Kopplung eines autarken C-Programms mit Modellierungs- und Simulationsfunktionalität über eine Dateischnittstelle mit einem Geographischen Informationssystem gekoppelt werden kann. Aufgrund der sauberen Modularisierung des Ausgangssystems Artemis konnten die dort verwendeten Strukturen weitgehend ins GIS übertragen werden. Aufwendig zeigte sich hauptsächlich die Umformatierung der Artemis-Dateiformate in eine fürs GIS geeignete Layer-Struktur, die um Tabellen für die Simulationsparameter ergänzt werden musste.

Funktional konnte das Ziel erreicht werden, die Bedienung des Plattenmodells ausschließlich über die GIS-Oberfläche abzuwickeln. Dies ist besonders für die aufwendige Eingabe der Plattengrenzen ein entscheidender Komfortgewinn.

Die Nutzung der graphischen Oberfläche des ArcMap ModelBuilders hat sich im Nachhinein als zwiespältig erwiesen: Einerseits veranschaulicht die gegebene Symbolik den Workflow und trägt zusammen mit der Möglichkeit, hierarchische Modelle aufzubauen, wesentlich zur strukturierten Umsetzung und der Transparenz für den Nutzer bei. Andererseits hat sich der ModelBuilder in der Entwicklungsphase als ungeeignetes Entwicklungswerkzeug erwiesen für komplexe Skript-Hierarchien wie sie für das vorgestellte Projekt notwendig waren. Schlechte Debugging-Möglichkeiten und häufige Laufzeitfehler bei der Abarbeitung der ModelBuilder-Modelle waren für den Projektfortschritt hinderlich. In dieser Hinsicht wäre es sicherlich

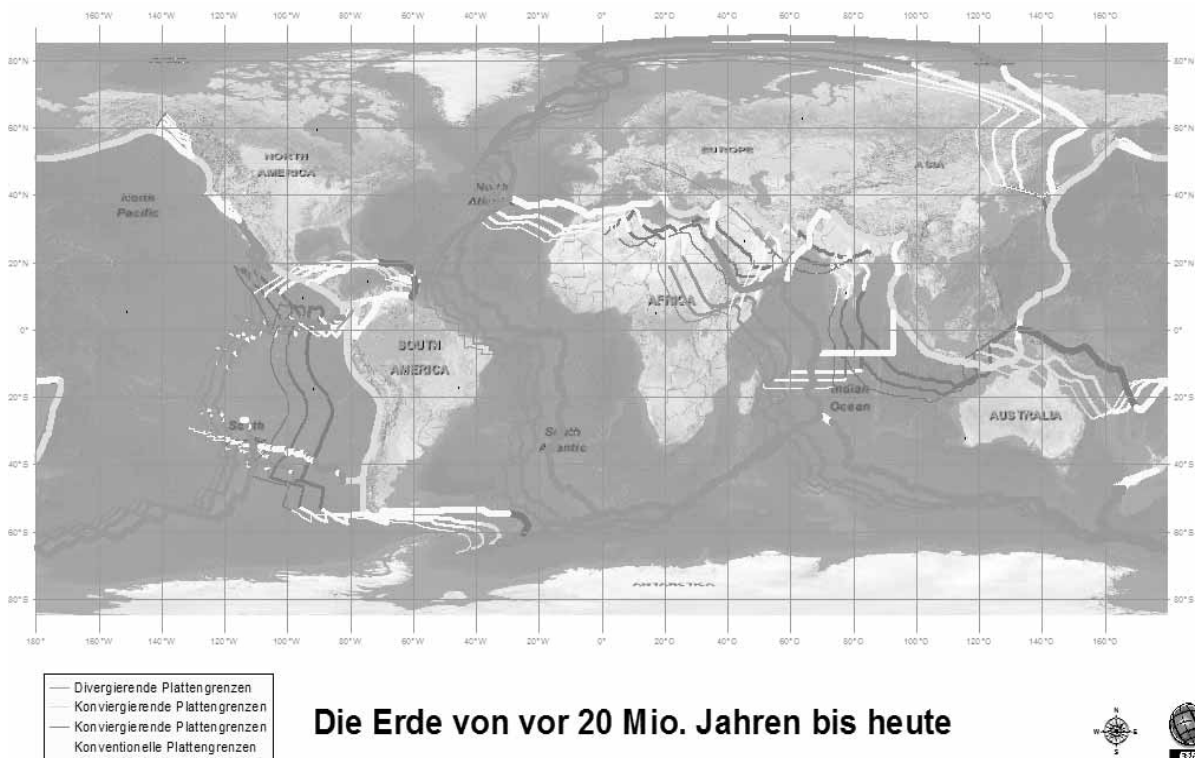


Abbildung 8: Eine Beispielsimulation im GIS

besser gewesen, die zu implementierenden Funktionalitäten zunächst auf der Ebene des Python-Skript-Interfaces von ArcMap umzusetzen und die so entwickelten Skripte erst anschließend für eine einfachere Bedienung mit Hilfe der ModelBuilder-Symbolik aufzubreiten.

7 References

- [1] Alfred Wegener: Die Entstehung der Kontinente, *Peterm. Geogr. Mitt.* 1 (1912), 185-195, 253-256, 305-309
- [2] C. DeMets, R. G. Gordon, D. F. Argus, S. Stein: Current plate motions, *Geophys. J. Int.*, 101 (1990) S. 425 ff.
- [3] <http://www.3sat.de/nano/diverses/kontinentaldrift/kontinentaldrift.html> (Zugriff am 16.06.2014)
- [4] Iris Rödder: Simulation von Plattenmodellen - mathematische Modellierung und Implementierung, Shaker, Aachen 2002
- [5] <http://pubs.usgs.gov/gip/dynamic/dynamic.pdf> (Zugriff am 16.06.2014)
- [6] Jochen Wittmann: Eine Benutzerschnittstelle für die Durchführung von Simulationsexperimenten - Entwurf und Implementierung der Experimentierungsumgebung für das Simulationssystem SIMPLEX II; Dissertation, Hrsg: The Society for Computer Simulation, SCS, San Diego, 1993
- [7] ESRI ArcGIS for Desktop, ArcMap: <http://www.esri.com/software/arcgis> (Zugriff am 16.06.2014)
- [8] Hübner, S., Kunisch, C., Wittmann, J.: Kopplung eines dynamischen Modells zur Plattentektonik mit ArcMap; in: Wittmann, J.; Müller, M. (Hrsg.), *Simulation in Umwelt- und Geowissenschaften: Workshop Leipzig 2013*, Shaker-Verlag, Aachen 2013, ISBN 978-3-8440-2009-0, pp 107-116
- [9] <http://www.arcgis.com/home/item.html?id=c4ec722a1cd34cf0a23904aadf8923a0> (Zugriff am 16.06.2014)
- [10] Karsten Bratz: Die Integration eines Modells zur Kontinentalverschiebung in das Geoinformationssystem ArcGIS, Bachelorarbeit an der HTW Berlin, FB 2, Umweltinformatik, Berlin 2013

Technisch-wirtschaftliche Optimierung von hybriden Inselnetzen in hoher zeitlicher Auflösung

H. Huyskens¹, P. Blechinger^{1,2}

¹ Reiner Lemoine Institut gGmbH, Ostendstraße 25, 12459 Berlin

² Technische Universität Berlin, Institut für Energietechnik, Fasanenstraße 89, 10623 Berlin

hendrik.huyskens@rl-institut.de

Hybride Mini-Grids bieten die Möglichkeit der kostengünstigen und CO₂-armen Energieversorgung. Die Auslegungen solcher Systeme geschieht meist auf Grundlage von Computersimulationen. Entscheidend dabei ist die genaue Abbildung technischer und wirtschaftlicher Faktoren der Komponenten. Im vorliegenden Paper wird die Modellierung eines solchen Simulationstools für Energiesysteme beschrieben und verschiedene Techniken zur Reduzierung des Rechenaufwands präsentiert. Abschließend findet eine Bewertung der vorgenommen Vereinfachungen statt.

1 Einleitung

In kleinen, nicht an ein Verteilnetz angeschlossenen Energiesystemen – sogenannten Mini-Grids (MG) basiert die Stromerzeugung meist auf Dieselgeneratoren (DG), die teuren und CO₂ intensiven Dieselmotoren benötigen [1]. Um den Dieserverbrauch zu reduzieren bietet sich die Erweiterung mit erneuerbaren Energien (EE), wie zum Beispiel Windenergie oder Photovoltaik, und Speichertechnologien hin zu einem hybriden MG an [2].

Im Vergleich zu einem reinen Dieselsystem ist ein hybrides MG deutlich komplexer. Verschiedene Erzeugungstechnologien und Speicher müssen miteinander kommunizieren um die Last möglichst sicher und effizient zu decken. Hinzu kommt, dass neben der variablen Last nun auch die Erzeugung variabel ist aufgrund der fluktuierenden Wind und Solar Ressourcen [3,4]. Die Auslegung eines sinnvoll dimensionierten MG erfordert daher detaillierte Simulationen.

Diese Simulationen werden bisher mit verschiedenen Tools durchgeführt, die jeweils Vor- und Nachteile aufweisen. Anforderungen an die Simulation sind so individuell, dass für die Optimierung hybrider MG ein eigenes Tool in Matlab entwickelt wird. Im Folgenden werden die Anforderungen an das Tool beschrieben und die Umsetzung spezifiziert.

2 Methodik: Anforderung an Simulationstool

Um die Anforderungen an das Simulationstool genau definieren zu können muss erst der reale Anwendungsfall beschrieben werden. Das Tool soll zur technisch-wirtschaftlichen Optimierung von hybriden MG genutzt werden. Dazu müssen die technischen und ökonomischen Charakteristika der Komponenten des MG (vgl. Abbildung 1) und die passenden Lastflüsse genau abgebildet werden. Um das Betriebsverhalten der DG und Batteriespeicher sowie die Betriebsstabilität (Spinning Reserve) genau untersuchen zu können darf die zeitliche Auflösung der Simulation nicht geringer als eine Minute sein [3,5].

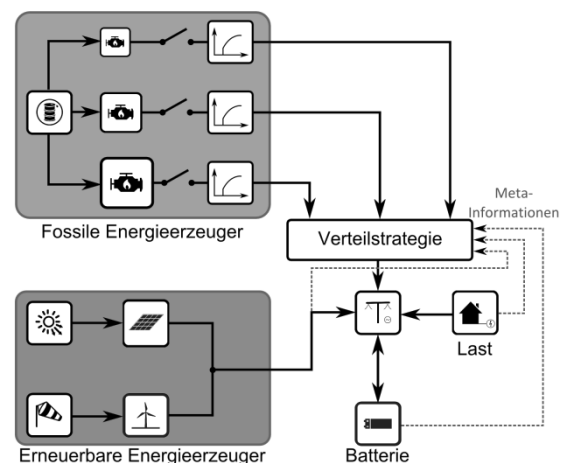


Abbildung 1. Hybrides Mini-Grid

In den folgenden Stichpunkten werden die Anforderungen nochmals zusammengefasst:

- Individuelle Spezifizierung der Komponenten insbesondere der DG (Lastkurve, Minimallast, minimale Laufzeit) und der Batterie (Zykluslebenszeit, Lade-/Entladewirkungsgrad)
- Ortsabhängige Berücksichtigung von Ressourcen- und Lastzeitreihen
- Betriebsstrategie zur Deckung der Last und Spinning Reserve sowie Kommunikation zwischen DG und Batterie
- Technisch-wirtschaftliche Optimierung des Gesamtsystems auf minimale Stromgestehungskosten unter vertretbarer Rechendauer (< 1 Tag)

Von den existierenden Simulationstools Hybrid2, PV-DesignPro und HOMER kann am ehesten HOMER [6] alle Anforderungen erfüllen. Allerdings ist es dabei nicht möglich in den Quellcode des Programms einzugreifen, deshalb fällt es für die weitere Forschungsanwendung aus. Um alle oben genannten Punkte zu erfüllen wird ein eigenes Tool entwickelt, das im nächsten Kapitel vorgestellt wird.

3 Ergebnisse: Finales Simulationstool

Die wirtschaftliche Betrachtung des System findet auf Basis eines Referenzjahres statt, für das anhand der Investitionskosten, der Wartungskosten und des Dieselverbrauchs die jährlichen Kosten berechnet werden. Grundlage für das Referenzjahr bilden die Einspeisezeitreihen der EE und die sich daraus entwickelnden Energieflüsse innerhalb des Systems (siehe Abbildung 2).

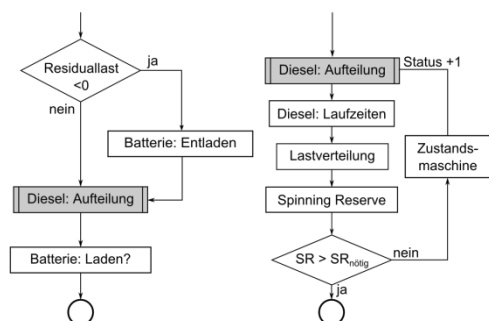


Abbildung 2. Energieflüsse und Verteilstrategie

Die Energieflüsse zu jedem Zeitpunkt werden bestimmt durch die verschiedenen Größen und

technischen Abbildungen der Komponenten. Die EE dienen als Energieeinspeiser und versorgen als erste die Last; die Differenz aus den Beträgen nennt man *Residuallast*. Ist die Residuallast positiv, d.h. es gibt Energie-Überschüsse, können diese in die Batterie geleitet werden. Kann die Batterie nicht den gesamten Überschuss aufnehmen, müssen die EE abgeriegelt werden; die überschüssige Energie geht dann verloren. Ist die Residuallast dagegen negativ muss die nicht gedeckte Last entweder vom DG-Park oder der Batterie versorgt werden. In beiden Fällen muss der Betriebspunkt des DG-Parks eingestellt werden; dies geschieht mithilfe der Verteilstrategie (siehe Abbildung 2, rechts).

Zunächst wird überprüft welche DG aufgrund ihrer minimalen Laufzeiten am Netz bleiben müssen. Auf dieser Grundlage wird entschieden wie die verbleibende Residuallast auf die DG aufgeteilt wird. Die optimale Aufteilung der Last wurde dafür in einer Voro Optimierung berechnet und in einer Look-up-Table hinterlegt (siehe Abbildung 3).

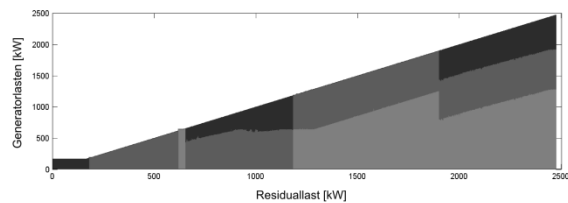


Abbildung 3. Beispiel einer Lastverteilung auf drei DG

Im nächsten Schritt wird überprüft ob eine bestimmte Spinning Reserve als Stabilitätskriterium zur Verfügung steht. Ist dies nicht der Fall, wird mithilfe eines vordefinierten Zustandsautomaten (siehe Abbildung 4 und zugehörige Tabelle 1) der kleinste nicht-laufende DG eingeschaltet und das zuvor beschriebene Prozedere wiederholt.

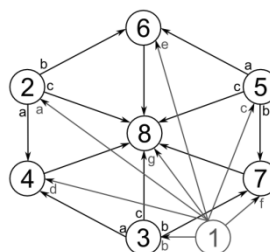


Abbildung 4. Übergänge der zuschaltbaren DG

Dies geschieht solange, bis das Stabilitätskriterium erfüllt ist. Die Aufteilung der Last auf die DG steht danach fest. Überschüsse, die der minimalen

Grundlast der DG geschuldet sind, können im letzten Schritt der Batterie hinzugeführt werden.

Technisch wird der Generatorpark im Modell, hier bestehend aus drei DG, jeweils durch eine Nennleistung, eine Minimallast, Teillastwirkungsgrade und minimale Laufzeiten abgebildet. Darauf basierend wird in der Voroptimierung für jeden Zustand (an oder aus) der DG (hier 7 Möglichkeiten, siehe Tabelle 1) und jeder möglichen Residuallast der kostengünstigste Betriebspunkt ermittelt (siehe Beispiel für Binärcode „0“ in Abbildung 3).

Binärcode	0	1	2	3	4	5	6	7
DG 1	0	1	0	1	0	1	0	1
DG 2	0	0	1	1	0	0	1	1
DG 3	0	0	0	0	1	1	1	1

Tabelle 1. Zustandsbeschreibung der DG
(0=DG läuft nicht, 1=DG muss laufen)

Die Batterie ist durch ihre Leistung, ihre Kapazität, einem dynamischen Wirkungsgradverlusts und einer zyklenabhängige Lebensdauer definiert.

Die Simulation kann nun auf Basis verschiedener Konfigurationen und Dimensionen der Komponenten die jeweils entstehenden Stromgestehungskosten (LCOE), den resultierenden Anteil der EE an der Stromerzeugung, Investitionskosten und Weiteres kalkulieren. Außerdem kann mithilfe einer mehrdimensionalen Optimierung, auf Basis eines genetischen Algorithmus [7], das kostengünstigste hybride System gefunden werden.

4 Diskussion und Zusammenfassung

Innerhalb des Modells werden einige Vereinfachungen unternommen, um die Rechenzeit gering zu halten. So werden die Einsatzzeiten und Betriebspunkte der DG nicht für jeden Zeitpunkt oder über das gesamte Jahr optimiert, sondern im Vorfeld in einer Look-up-table gespeichert. Bei 525600 Minuten pro Jahr wäre dieselbe Anzahl an Optimierungen nötig, wollte man die DG minütlich optimieren; selbst bei einer kurzen Optimierungsdauer wäre der Rechenaufwand immens. Diskretisiert man dagegen die möglichen Betriebspunkte der DG, wären, abhängig von der Anzahl i der DG und deren Nennleistung P_{DG_i} , in einer Voroptimierung nur:

$$n = i^2 \cdot \sum_i P_{DG_i}$$

Optimierungen nötig. Die Genauigkeitsverluste, die mit einer Diskretisierung der Betriebspunkte einhergeht, können als gering betrachtet werden.

Zieht man das Stabilitätskriterium der Spinning Reserve hinzu und soll damit in der Voroptimierung auch das Zuschalten der DG berücksichtigt werden, bedarf es:

$$n = i^2 \cdot \sum_i P_{DG_i} \cdot \sum_i (P_{DG_i} - P_{min_i})$$

Optimierungen. Dieser Rechenaufwand wird durch den oben beschriebenen Zustandsautomaten mit einer fest-definierten Übergangsreihenfolge umgangen. Fehlentscheidungen für das Zuschalten eines DG können vorkommen, werden aber als gering eingeschätzt und werden hinsichtlich des Rechenaufwands in Kauf genommen.

Das so beschriebene Modell ist in der Lage komplexe, mehrdimensionale Optimierungsprobleme im Bereich der Energiesysteme zeitnah zu lösen.

5 References

- [1] Linares, A., López-Manzanares, L., Cendagorta-Galarza, M., & Friend, M. (2010). *Hybrid RE system: Minigrid set-up, first results and lessons learned*. 25th European Photovoltaic Solar Energy Conference and Exhibition (pp. 5194-5198). Valencia.
- [2] Othman, S., Irving, M. R., & Taylor, G. (2011). *Discrete Time Simulation of Electrical Power Network with Intermittent Generation*, (September), 1-6.
- [3] Piwko, R., Bai, X., Clark, K., Jordan, G., Miller, N., & Zimmerlin, J. (2005). *The Effects of integrating Wind Power on Transmission System Planning, Reliability and Operations*. Albany, NY.
- [4] Knopp, M. (2012). *Study on maximum permissible intermittent electricity generators in in an electricity supply network based on grid stability power quality criteria*. Masterthesis, Fernuniversität Hagen.
- [5] Roel Verlaan. Renewable Energy Integration Issues. In *21st PPA Annual Conference*, Vauatu, 2012. DNV KEMA.

- [6] Report Iea-pvps. World-wide overview of design and simulation tools for hybrid PV systems. Technical report, International Energy Agency, 2011.
- [7] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. PhD thesis, MIT, 1992.

Ontology for Objective Flight Simulator Fidelity Evaluation

Umut Durak¹, Artur Schmidt², Thorsten Pawletta²

¹Institute of Flight Systems, German Aerospace Center (DLR)

²Wismar University of Applied Sciences

umut.durak@dlr.de

The term simulator fidelity has become enormously important in the scope of simulation research, when assessing training efficiency and the transfer of training to real flight. It is defined as the degree to which a flight simulator matches the characteristics of the real aircraft. Objective simulator fidelity provides an engineering standard, by attacking the fidelity problem with comparison of simulator and the actual flight over some quantitative cues. Research flight simulation encompasses some differences from commercial flight simulation. It requires high flexibility and versatility concerning the cockpit layout and visual and motion systems, as well as flight simulation models. It should be easy to modify the flight simulation model or other soft- and hardware components of the simulator. By this, there is a need for an automatic test method, in order to determine the fidelity of the most relevant simulator subsystems, since they are often modified during the life cycle of the simulator. The Institute of Flight Systems (FT) at the German Aerospace Center (DLR) has a reconfigurable flight simulator, the Air Vehicle Simulator (AVES), for research of rotorcraft and fixed-wing aircraft. The study reported in this paper targets a model based testing approach designed to tackle the high flexibility requirement of AVES. This paper presents a metamodel for objective flight simulator evaluation. Metamodeling has been carried out in two levels. An Experimental Frame Ontology (EFO) has been developed adopting experimental frames from Discrete Event System Specification (DEVS), and as an upper ontology to specify a formal structure for simulation test. Then in Objective Fidelity Evaluation Ontology (OFEO) that builds upon EFO, domain specific meta-test definitions are captured.

1 Introduction

From the late 1920s, when Edward Link built “the Blue Box” [1], flight simulators have been important elements of aviation. Flight simulators became well accepted as training aids by many aircraft operators before the digital era. Highly sophisticated flight simulators have been employed commercially within civil and military flight training organizations in order to enhance pilot skills.

In the 1980s, the aeronautics research community started using flight simulators for developing and experimenting advanced concepts and conducting aviation human factors research. Some of the first examples of research flight simulators include AT-TAS Ground Based Simulator from German Aerospace Center (DLR) [2] [3], NASA Crew Vehicle Systems Research Facility in Ames Research Center [4] and Visual Motion Simulation and Cockpit Motion Facility from Langley Research Center [5]. Some more recent examples are the Air Vehicle Simulator (AVES) of DLR [6], HELIFLIGHT from the University of Liverpool [7], NASA Ames Vertical Motion

Simulator [8] and SIMONA of Delft University of Technology [9].

Fidelity in flight simulation can be defined as the degree to which a flight simulator matches the characteristics of the real aircraft. It became more and more important within the scope of flight simulation research as its effect on training efficiency and transfer of training to real flight became better understood [10]. Objective simulator fidelity assessment provides an engineering standard to qualify the degree of fidelity through objective measures. It approaches the fidelity problem with comparison of simulator and the actual flight over some quantitative cues. Requirements for research flight simulators encompass some differences from commercial flight simulators. They require high flexibility and versatility concerning the cockpit layout and visual and motion systems, as well as flight simulation models. They must allow easy modification of the flight simulation model or other soft and hardware components of the simulator. To allow this, there is a need for an automatic test method, to determine the fidelity of most relevant

simulator subsystems, since they are often modified during the life cycle of the simulator.

Automated testing can be applied through the use of software to control the execution of tests and a comparison of actual outcomes to the predicted ones. Available test data of the aircraft are used as input signals of the simulator and the output signals of the simulator are compared to the measurements to be presented for the evaluator in a smart format. Braun and Galloway [11] reported their automated fidelity test system that compares directly the flight test results and manual execution of flight tests in simulators. Wang et al. [12] [13] presented Automated Test System (ATS) that measure force function, evaluation function and transport delay with its non-intrusive interface with operator station. Jarvis et al. [14] summarizes the efforts on validation of sensory cues, motion cues, vibration and sound cues, visual cues, transport delays and flight dynamics models in flight simulators.

Previous efforts on automated testing for objective flight simulator evaluation utilized fixed test descriptions. DLR intends to adopt a Model Based Testing (MBT) approach to tackle high flexibility requirement of research flight simulators, such as AVES. MBT can be introduced as the idea of automating test case generation from a test model rather than implementing test cases manually [15]. Test modeling is used to specify test cases, and these test models are translated automatically to executable test cases. Metamodeling is employed to capture the domain specific concepts and constraints for building test models [16].

Metamodel is defined as an explicit model of constructs and rules that are used to define a model [17]. Following Gruber [18], definition of ontology is “explicit specification of shared conceptualization”. Moreover, metamodels are categorized as ontologies that are used modellers [17].

This paper focusses on the metamodeling step and will present how concepts and constraints of objective flight simulator evaluation are captured in ontology to create a metamodel for test modeling.

Here, the test case can be defined as a sequence of input stimuli that will be fed to the System Under Test (SUT), namely test inputs and the expected behaviour of the system, namely test oracle (Figure 1) [19].

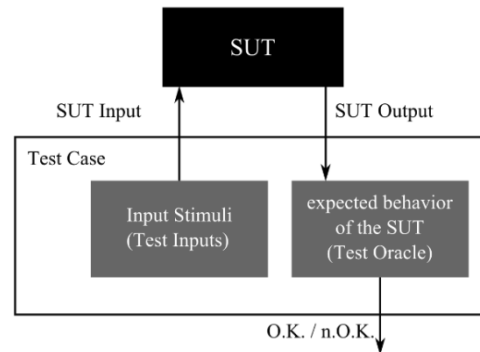


Figure 1 Test Case Structure

Moser et al. [20] stressed that ontologies as machine-readable domain knowledge, which can be utilized for test case generation. Then Nguyen et al. [21] presented a framework for ontology driven test case generation in the context of multi-agent systems. Adopting these ideas, ontologies are employed to structure meta-test definitions. The domain knowledge about the objective validation of simulator systems including the rules for assessing the results of test runs is captured in ontologies.

Zeigler and his colleagues developed the concept of Experimental Frame (EF) [22] [23]. An EF defines the conditions under which a model is to be examined. It comprises of an input generator, a verifier for the desired conditions and an analyzer for the outputs. Following Zeigler et al. [23], the EF is critical for evaluating the model validity. Traoure and Muzy in [24] and Foures et al. in [25] published the usage of the EF approach for specifying invariant validation experiments.

In this research, metamodeling has been carried out on two levels. An EF Ontology (EFO) has been developed as an upper ontology to specify a formal structure for generic simulation test model. Then in Objective Fidelity Evaluation Ontology (OFEO) that builds upon (EFO), domain specific meta test definitions are captured. Protégé [26] is used as the ontology development environment and ontologies are developed using Ontology Web Language (OWL). This paper will present these ontologies after introducing a background on objective fidelity evaluation, experimental frames and ontologies in general.

2 Background

2.1 Objective Fidelity Evaluation

Fidelity is regarded as a multivariate construct with no consensus among researchers on a single index of

measurement or definition and it is strongly related to the training task to be performed with the simulator.

There are two approaches to measure simulator fidelity; subjectively and objectively [12]. The subjective approach tries to identify the degree of realism felt by the user. User feedback is structured using rating scales [27]. Although subjective scales are valuable, it is hard to generalize across the scales because the individual opinions and bias of raters [12]. Objective approaches attack the fidelity problem with of simulator and the actual flight over some quantitative cues.

ICAO 9625 Manual of Criteria for the Qualification of Flight Training Devices, 3rd Edition [28] is the well accepted global standard for qualification of flight training devices. The standard specifies seven types of fidelity that correspond to a capability level to provide a certain type of training. For example, simulators classed as ‘Type 1’ can be used for all training tasks used during completion of Private Pilot License (PPL) training, whereas ‘Type 7’ is required for some of the training tasks used when awarding ‘Type Rating’. Appendix B of the standard specifies the test cases for objective validation of simulators. These test cases include comparison of the results from tests conducted in the simulator and aircraft validation data.

The Royal Aeronautical Society (RAeS) published Aeroplane Simulation Training Device Evaluation Handbook Vol. 1 Objective Testing [29] to ease the implementation and enhance the understanding of objective tests introduced in ICAO 9625. It provides further discussions about the implementation of each test and introduces some example cases with some plots.

ICAO 9625 provides tables that specify each test case with parameters, tolerances and flight conditions. Table 1 shows an example test specification from the standard, for testing the minimum radius.

Test	Tolerance	Type						
		1	2	3	4	5	6	7
Minimum radius turn	±0.9m (3ft) or ±20% of aeroplane turn radius					✓		✓

Table 1 Sample Test Specification from ICAO 9625 [28]

Tests are grouped under performance, handling qualities, motion system, visual system and sound system. Among these tests, those under performance and

handling qualities headings are related to flight dynamics models, and have no other subsystem or device dependencies. For this reason, they are considered to better suit automation. Therefore, as a first step, this addresses these groups.

The RAeS introduces the benefits of employing automatic testing in objective fidelity evaluation as repeatability, ease and rapidity of conducting tests. The RAeS handbook [29] specifies the features of an automatic testing system as initializing the simulator with the test initial conditions, trimming the aircraft, creating the stimulus if required, using flight controls and finally checking the simulator output against test criteria.

2.2 Experimental Frame Approach

The EF approach was originally introduced by Zeigler in [22] in context with the Discrete Event System Specification (DEVS). The objective is the explicit separation between the model and the experiment. Moreover, an EF specifies a limited set of circumstances under which a model is to be observed. Nowadays, the EF approach belongs to the state of the art and it is used in many modelling and simulation projects including validation experiments [24] [25] [30] [31]. Following Zeigler [22], the formal specification of the EF is given by the 7-tuple:

$$EF = \langle T, I, O, C, \Omega_i, \Omega_c, SU \rangle$$

where:

T is the time base

I is the set of input variables

O is the set of output variables

C is the set of control variables

Ω_i is the set of admissible input segments

Ω_c is the set of admissible control segment

SU is a set of summary mappings

The EF can be implemented in various ways. Zeigler [22] recommends implementing the EF as a coupled system consisting of a generator, acceptor and a transducer that is connected to a SUT. In our context, the SUT is always a model. For this reason, it is called Model Under Test (MUT). Figure 2 illustrates such a realization of EF coupled to a MUT schematically.

Test inputs are produced by a generator. They have to be admissible input segments of MUT and influence its behavior. The acceptor and transducer form the

test oracle. Based on output variables, the transducer calculates outcome measures in the form of performance indices, comparative values, statistics etc.

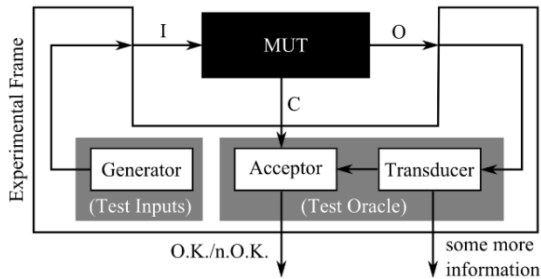


Figure 2 Illustration of EF with MUT.

The acceptor corresponds to a decision unit that decides if an experiment is valid or not. For this purpose the acceptor monitors its inputs and maps them to a specified admissible control segment. In case of violation of the admissible control segment the experiment will not be accepted. Besides control variables, the input of an acceptor can be output variables or outcome measures.

The EF approach defines a uniform structure for a systematic experiment specification. The specification has to be coded in the description of an EF. This means that each kind of experiment needs the definition of a distinct EF.

2.3 Ontologies

Knowledge in a domain is formalized using concepts, relations, functions, axioms and instances in an ontology. Concepts can be anything about which something is said, and therefore, can be a description of a task, function, action, strategy etc. Taxonomies are widely used to organize the ontological knowledge in domain using generalization/specialization relationship through simple/multiple inheritance. Relationships represent a type of interaction between the concepts of the domain and functions can be regarded as a special kind of relation. Axioms on the other hand are used to model sentences that are always true. They are added to ontology for several purposes, such as constraining the information contained in the ontology, verifying its correctness or deducing new information. Instances are the terms that are used to represent the elements of the domain. They actually represent the elements of the concepts [32].

Ontologies in engineering domain have been developed for various purposes including specifying

engineering information systems, integration of engineering applications, supporting engineering design and development. The first efforts on developing engineering ontologies were in the 1990's. The 'PhysSys' [33] was one of the first engineering ontologies based upon system dynamics theory that is practiced in engineering modeling, simulation and design. The PhysSys was developed to formally define how design engineers or the end users of Computer Aided Engineering (CAE) systems understand their domain and to provide a foundation for the conceptual schema for data structuring in engineering databases, libraries and other CAE information systems [33] [34]. The ideas formalized in PhysSys provided a base for the development of a library of reusable models for engineering and design. Fishwick and Miller in [35] discussed the venues of ontology use in modeling and simulation. One of the late examples of ontology use in modeling and simulation is reported by Durak et al. [36] [37]. The group enabled simulation reuse over an ontology driven methodology. Another ontology-based modeling and simulation approach was established by Zeigler with the System Entity Structure and Model Base (SES/MB) framework [22] [23] [38] [39]. Today the SES is an ontology framework for conceptual system modeling and for specification of a set of modular hierarchical system structures.

3 Experimental Frame Ontology

The EFO forms the upper level of the metamodel for objective flight simulation evaluation. The previously introduced EF approach is used to specify a formal structure of generic test cases. Hence, every test case has to be specified according to the EF definition in Section 2.2.

Figure 3 illustrates the entity hierarchy of the EFO in Protégé. The first layer consists of three entities: Computational Unit, Informational Unit and the EF. Computational Units comprise the generic Acceptor, Transducer and Generator which will be presented as executable blocks in a test case. The Information Unit defines basic entities of an EF. For practical usage, there are some differences to the theoretical definition of an EF according to the 7-tuple. The Experimental Frame entity conforms to the actual EF.

Furthermore particular properties are implemented to define the relations between the entities. For example the properties *composedOf* and *definedBy* makes clear

that any EF is a composition of Computational Unit and is defined by the Informational Unit.

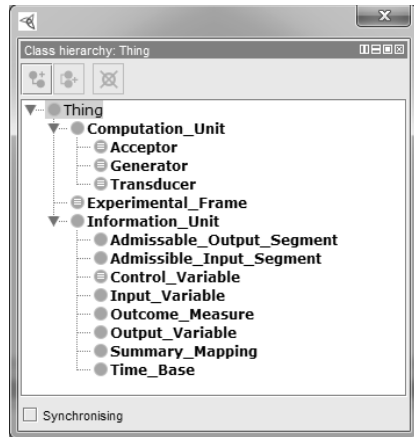


Figure 3 Entity Hierarchy of the Experimental Frame Ontology

As a result we obtain a generic EF which conforms to a generic test case. Thus, any test case will have the unique structure as shown in Figure 4 on its top level.

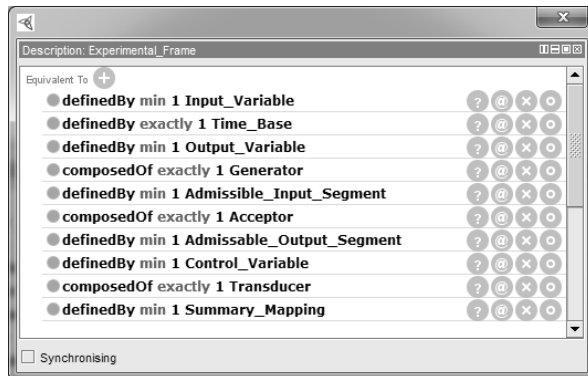


Figure 4 Description of a Generic Experimental Frame

The EFO forms the basis for the OFEO that will define test cases in detail.

4 Objective Fidelity Evaluation Ontology

OFEO is constructed by extending the upper level EFO that specifies any test case that will be applied to MUT using experimental frames formalism. The hierarchy of OFEO using Protégé is depicted in Figure 5. The elements from EFO can be traced in this hierarchy.

Each objective validation test case described in ICAO 9625 under performance and handling qualities are specified by an experimental frame. Thus, each test possesses a Generator, Transducer and an Acceptor. The specification of these three entities will inherently describe how this specific test will be exercised.

These three entities will constitute the automatic test system.

Following the features of automated test systems introduced in the RAeS Handbook [29], the Generator is described as the component to initialize the test with initial conditions and trim the aircraft and create the stimulus following the ones from the flight test using the flight controls. Hence, the Generator is interpreted as test independent. On the other hand, the Transducer is described as the component that will compute Outcome Measures that are required for the Acceptor for a specific test.

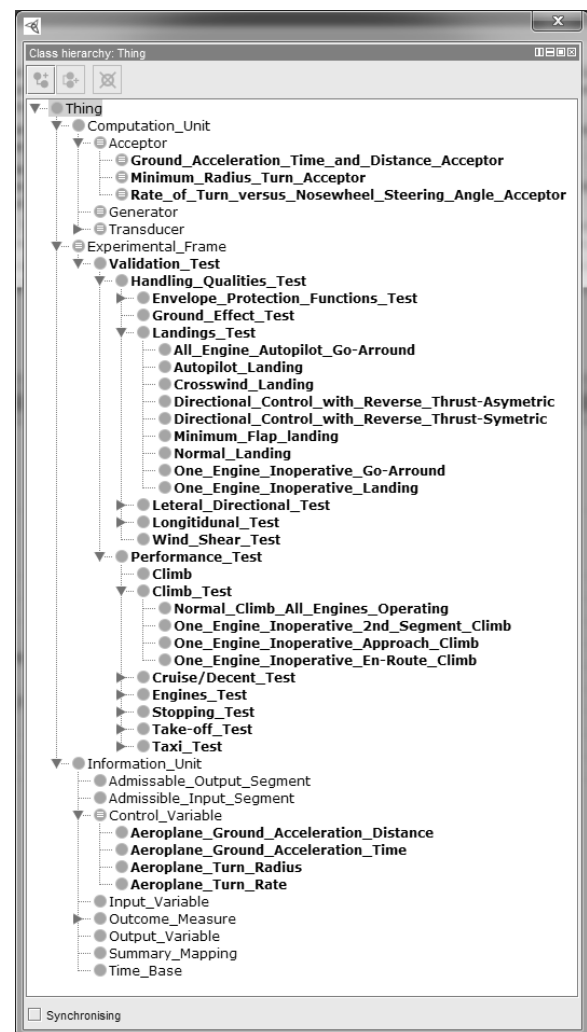


Figure 5 Objective Fidelity Evaluation Ontology Hierarchy

As an example, the Minimum Turn Radius test requires a Simulated Turn Radius to be computed from a simulation output. Or likewise, Rate of Turn versus Nosewheel Steering Angle test requires Simulated Turn Rate value to be computed. So, a specific trans-

ducer is defined for every test. Lastly, the Acceptor is described as the component that checking the MUT against test criteria. Since every test has a particular criterion, an Acceptor is defined for each test. Accordingly, we are expecting to have particular Control Variables for each test.

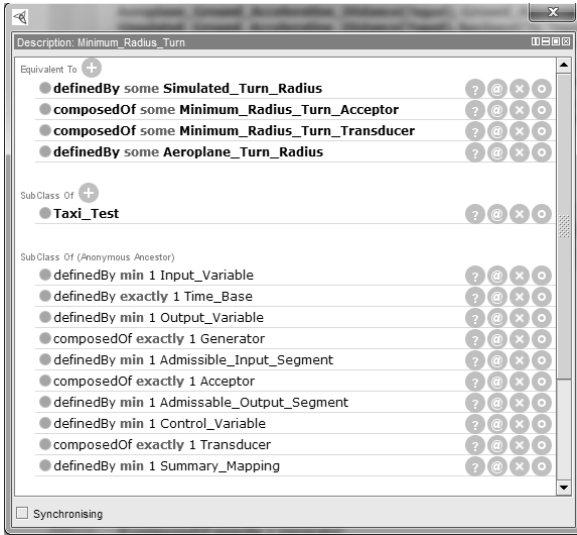


Figure 6 Minimum Radius Turn Test Description

Figure 6 presents an example test description in Protégé. The Minimum Turn Radius Test is specified with a specific Acceptor, Transducer and Control Variables, Simulated Turn Radius and Aeroplane Turn Radius. On the other hand, it inherits the properties of an experimental frame. So it will also have a Generator, Input Variables, Output Variables, Admissible Input Segments, Admissible Output Segments and a Summary Mapping. It is clear that input and output variables of the flight simulator are application specific but does not vary with test cases, so generic definitions are kept for these variables and admissible segments.

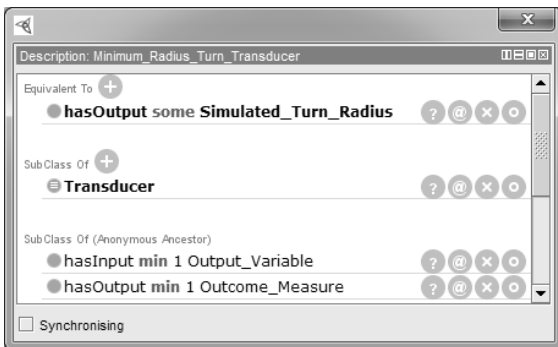


Figure 7 Minimum Radius Turn Transducer Description

Minimum Radius Turn Transducer (Figure 7) is defined with an output Simulated Turn Radius while it also inherits the properties of a Transducer. It will be using Output Variables for computing the outcome measure. Since the computation of the outcome measure is largely implementation specific, ontology does not have any knowledge about it.

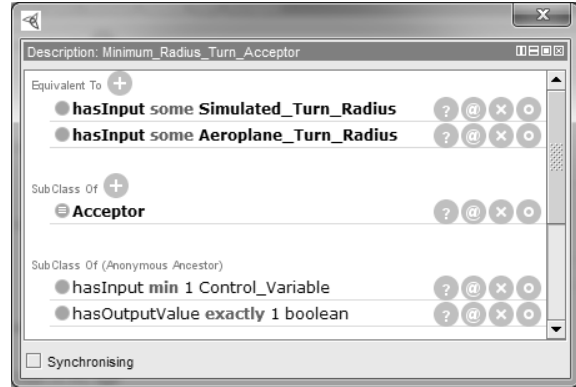


Figure 8 Minimum Radius Turn Acceptor Description

As an example, the Minimum Radius Turn Acceptor is depicted in Figure 8. Since each of the tests have distinct criteria, the acceptors will have particular inputs. Accordingly, Minimum Radius Turn Acceptor is described with Simulated Turn Radius and Aeroplane Turn Radius inputs. On the other hand the output of the acceptor is always a Boolean if the criterion is matched or not.



Figure 9 Rules for Acceptors

Semantic Web Rule Language (SWRL) [40] is used to formalize the acceptance criteria. The rules specify how the inputs of the acceptor are used to compute if the test is successful or not. In Figure 9, the rule in the front windows says that Minimum Radius Turn Acceptor has a true output when the difference between the simulated and the real minimum turn radius is smaller than 20 %.

5 Conclusion

This paper presents an ontology based metamodeling approach for objective flight simulator evaluation. EFO adopts the concept of EF from DEVS, as an upper ontology to specify a formal structure for test cases. Thus with EF, concepts of MBT could be formally specified. This established a solid base for modeling specific test cases. Then in OFEO that builds upon EFO, domain specific meta-test definitions are modeled. While OWL is used as the ontology language; SWRL is employed to capture the rules. Protégé is utilized as the ontology development environment.

This effort assembled the first step of developing an automatic test system for simulator fidelity evaluation. The next step is to design and establish the computational infrastructures that will interpret the knowledge captured in these ontologies and execute the tests.

6 References

- [1] D. Allerton. *Principles of Flight Simulation*. John Wiley & Sons, Ltd, West Sussex, United Kingdom, 2009.
- [2] P. Saager. *Real-Time Hardware-in-the-Loop Simulation for 'ATTAS' and 'ATThES' Advanced Technology Flight Test Vehicles*. in AGARD Guidance and Control Panel, 50th Symposium, Izmir, Turkey, 1990.
- [3] S. Klaes. *ATTAS Ground Based System Simulator -An Update-*. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Denver, CO, 2000.
- [4] B. Sullivan and P. Soukup. *The NASA 747-400 Flight Simulator: A National Resource for Aviation Safety Research*. In: AIAA Flight Simulation Technologies Conference, San Diego, CA, 1996.
- [5] R. Smith. *A Description of the Cockpit Motion Facility and the Research Flight Deck Simulator*. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Denver, CO, 2000.
- [6] H. Duda, T. Gerlach, S. Advani and M. Potter. *Design of the DLR AVES Research Flight Simulator*. In: AIAA Modeling and Simulation Technologies (MS) Conference, Boston, MA, 2013.
- [7] M. White and G. Padfield. *The Use of Flight Simulation for Research and Teaching in Academia*. In: AIAA Atmospheric Flight Mechanics Conference and Exhibit, Keystone, CO, 2006.
- [8] S. Advani, D. Giovannetti and M. Blum. *Design of a Hexapod Motion Cueing System for NASA Ames Vertical Motion Simulator*. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Monterey, California, 2002.
- [9] O. Stroosma, R. van Paassen and M. Mulder. *Using the Simona Research Simulator for Human-Machine Interaction Research*. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Austin, Texas, 2003.
- [10] T. Longridge, J. Bürki-Cohen, T. Go and A. Kendra. *Simulator Fidelity Considerations for Training and Evaluation of Today's Airline Pilots*. In: Proceedings of the 11th International Symposium on Aviation Psychology, Columbus, OH, 2001.
- [11] D. Braun and R. Galloway. *Universal Automated Flight Simulator Fidelity Test System*. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Rhode Island, 2004.
- [12] C. Wang, J. He, G. Li and J. Han. *An Automated Test System for Flight Simulator Fidelity Evaluation*. Journal of Computers, vol. 4(11), 2009.
- [13] C. Wang, J. Han, G. Li and H. Jiang. *Flight Simulator Fidelity Evaluation Automated Test System Analysis*. In: 2008 International Workshop on Education Technology and Training, Shanghai, China, 2008.
- [14] P. Jarvis, D. Spira and B. Lalonde. *Flight Simulator Modeling and Validation Approaches and Pilot-in-the-loop Fidelity*. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Honolulu, Hawaii, 2008.
- [15] J. Zander, I. Schieferdecker and P. Mosterman. *A Taxonomy of Model-Based Testing for Embedded Systems from Multiple Industry Domains*. In: Model-Based Testing for

- Embedded Systems, Boca Rato, CRC Press, 2012, pp. 3-23.
- [16] A. Guduvan, H. Waselynck, V. Wiels, G. Durrieu, Y. Fusero and M. Schieber. *A Meta-Model for Tests of Avionics Embedded Systems*. In: Modelsward, Barcelona, Spain, 2013.
- [17] D. Gasevic, D. Djuric and V. Devedzic. *Model Driven Architecture and Ontology Development*. Springer-Verlag, Berlin, 2006.
- [18] T. Gruber. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. Int. Journal of Human-Computer Studies, vol. 43, pp. 907-928, 1995.
- [19] S. Weissleder. *Test Models and Coverage Criteria for Automatic Model-Based Test Generation with UML State Machines*. Humboldt-Universität zu Berlin, Berlin, 2010.
- [20] T. Moser, G. Düe and S. Biffl, *Ontology-Based Test Case Generation For Simulating Complex Production Automation Systems*. In: SEKE 2010, San Fransisco Bay, USA, 2010.
- [21] C. Nguyen, A. Perini and P. Tonella. *Ontology-based Test Generation for Multiagent Systems*. In: 7th International Joint Conference on Autonomous Agents and Multiagent Systems, Estoril, Portugal, 2008.
- [22] B. Zeigler. *Multifaceted Modelling and Discrete Event Simulation*. Academic Press Professional, Inc., 1984.
- [23] B. Zeigler, H. Praehofer and T. Kim. *Theory of Modeling and Simulation: Integrating discrete event and continuous complex dynamic systems*. Academic Press, Inc., 2000.
- [24] M. Traoré and A. Muzy. *Capturing the dual relationship between simulation models and their context*. Simulation Modelling Practice and Theory, 14, pp. 126-142, 2006.
- [25] D. Foures, V. Albert and A. Nketsa. *Simulation Validation Using the Compatibility between Simulation Model and Experimental Frame*. In: 45th Summer Simulation Multi-conference, Toronto, Canada, 2013.
- [26] K. Holger, M. Horridge, M. Musen, A. Rector, R. Stevans, N. Drummond, P. Lord, N. Noy, J. Seidenberg and H. Wangl. *The Protege OWL Experience*. In: OWLED, Galway, Ireland, 2005.
- [27] P. Perfect, E. Timson, M. White, R. Erdos, A. Gubbels and A. Berryman. *A Rating Scale for Subjective Assesment of Simulator Fidelity*. In: 37th European Rotorcraft Forum, Gallarate, Italy, 2011.
- [28] ICAO, *Manual Criteria for the Qualification of Flight Training Devices*. ICAO, Quebec, Canada, 2009.
- [29] RAeS, *Aeroplane Flight Simulation Training Device Evaluation Handbook Vol.1 Objective Testing*. RAeS, London, 2009.
- [30] B. Nader and J. B. Filippi, *An Experimental Frame for the Simulation of Forest Fire Spread*. In: Proceeding of the 2011 Winter Simulation Conference, Phoenix, Arizona, USA, 2011.
- [31] A. Zengin and M. Ozturk, *Formal verification and validation with DEVS-Suite: OSPF Case study*. Simulation Modelling Practice, vol. 29, pp. 193-206, 2012.
- [32] O. Corcho and A. Perez, *Evaluating Knowledge Representation and Reasoning Capabilities of Ontology Specification Languages*. In: ECAI'00 Workshop on Applications of Ontologies and Problem Solving Methods, Berlin, Germany, 2000.
- [33] W. Borst, J. Akkermans, A. Pos and J. Top. *The PhysSys Ontology for Physical System*. In: QR'95 Ninth International Workshop on Qualitative Reasoning, Amsterdam, Netherlands, 1995.
- [34] W. Borst and J. Akkermans. *Engineering Ontologies*. International Journal of Human-Computer Studies, vol. 46 (2/3), pp. 365-406, 1997.
- [35] P. Fishwick and J. Miller. *Ontologies for Modeling and Simulation: Issues and Approaches*. In: Winter Simulation Conference, Washington, DC, 2004.
- [36] U. Durak, H. Oguztuzun and K. Ider. *Ontology Based Trajectory Simulation Framework*. Journal of Computing and Information Science in Engineering, vol. 8(1), March 2008.
- [37] U. Durak, H. Oguztuzun and K. Ider. *Ontology Based Domain Engineering for Trajectory Simulation Reuse*. Journal of Software Engineering and Knowledge Engineering, vol. 19(8), December 2009.
- [38] B. Zeigler. *Modeling & Simulation-Based Data Engineering: Introducing pragmatics into ontologies for net-centric information exchange*. Academic Press, Inc., 2007.
- [39] B. Zeigler and H. Sarjoughian. *Guide to Modeling and Simulation of Systems of Systems*. Springer, 2013.
- [40] I. Harrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosz and M. Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. W3C, Canada, 2004.

Development of a container terminal simulation meta-model

Lange, Ann-Kathrin¹; Pirovano, Giovanni²; Pozzi, Rosella³; Rossi, Tommaso³

¹ Institute of Maritime Logistics of Hamburg University of Technology

² Dept. of Management, Economics and Industrial Engineering of Politecnico di Milano University

³ Industrial Engineering School of LIUC – Cattaneo University

ann-kathrin.lange@tuhh.de

This article introduces a simulation meta-model to support terminal planners, operator and managers in the design and management of seaport container terminals. Due to the increasing requirements of shipping companies regarding efficiency, quality and price for the handling processes at container terminals, the use of integrated approaches for improving the performance has grown significantly. Simulation, which has proven highly beneficial in production and logistics, represents an adequate tool to deal with complex systems like container terminals. However, building simulation models requires much time and simulation software know-how. To counteract this effect, this article presents a simulation meta-model of seaport container terminals, which supports the user in building specific simulation models. Since the simulation model is automatically created through the meta-model, neither the personnel skills nor the time available to build the simulation model represent significant hurdles. Furthermore, the proposed meta-model can dramatically reduce the time required to test a specific configuration of a container terminal and/or a particular management policy. The meta-model consists of a user interface with database, where the user can specify elements and their parameters, an atom library representing all elements of the system and software application, which is used to automatically build the simulation model.

1 Introduction

Around 80 per cent of global merchandise trade by volume were transported by water and therefore handled in ports in 2012 [1]. This fact highlights the further growing strategic economic importance of seaborne transport. Combined with low and at the same time volatile freight rates for maritime transport, still remaining after the economic crisis in 2008, and rising bunker prices, this development stimulates the ongoing increase of ship dimensions. This derives a special significance for container vessels, which transport around 52 per cent of global seaborne trade in terms of value [2].

The size of the largest container vessels has more than doubled from around 8,000 twenty-foot equivalent units (TEU) in 2004 to over 18,000 TEU in 2013. Without optimization of the container handling, this would lead to much longer berthing times for unloading and loading the vessels and, as a consequence, to sharp rising port charges. But due to the weak bargaining position of container terminals in relation to shipping companies [3], the terminal operators are pressed to optimize their efficiency and productivity while keeping up low prices. Therefore, all operations and services at the container terminal have to be evaluated and, if necessary, redesigned and adjusted

with high investments to meet the stringent demands of a higher turnover in short time windows and higher quality [4].

Seaport container terminals can be considered, in term of material flow, as open systems with two interfaces to other linked systems [5]. One of the interfaces is the quayside, where container ships are assigned to a specific berth and discharged and charged by a set number of ship-to-shore cranes. The other interface is the landside with the unloading and loading of trucks and trains, which can be carried out by different kinds of internal equipment, e.g. rubber-tired-gantry cranes or straddle carriers. The transport of containers from the interfaces to the stocking yard or vice versa is carried out by horizontal transport means, which may differ depending on the required task.

As a result, there is a big variance in the used internal equipment in all areas of the terminal, depending on many different factors, e.g. the stocking systems in the container yard, the average and maximum size of the landing container vessel, the labour costs in the area, the available space in the port, security requirements and the demanded productivity [4;5;6]. Recent developments are e.g. the increasing automation of

handling processes to reduce labour costs and to optimise quality and using advanced spreaders to lift multiple containers for enhancing the productivity as well as lowering costs.

To plan, analyse, manage and optimise the complex system of a container terminal, it is no longer sufficient to rely on the knowledge of singular experts or on the problem solving competence of departments in specific, isolated areas. Therefore, integrated approaches for improving the performance of container terminals have been developed. Apart from analytical approaches, there is a focus on simulation based approaches [4], which have proven highly beneficial as decision support systems in production and logistics in general [7] and for container terminals in particular [8;9;10].

For newly planned container terminals, simulation models can provide a preview of the expected overall performance and support the identification of problems before implementing the system. For already existing container terminals, simulation can help to identify bottle necks and optimisation potentials in the current situation and compare them to alternative operation approaches, which can be tested easily and without risks.

This article introduces a methodology to support decisions of seaport container terminal planners, operators and managers concerning the terminal layout, equipment and operations by developing a simulation meta-model. This meta-model overcomes the limitations of usual simulation models whose building requires much time and simulation software know-how of the user. Instead it enables the user to quickly build specific simulation models by simply entering all relevant characteristics of the seaport container terminal and its equipment in a user friendly interface.

The article is structured as follows. Section 2 provides information on the state of research about simulation models of container terminals on the one hand and about meta-modelling on the other hand. Section 3 presents the proposed methodology for the simulation meta-model for container terminals by explaining the architecture, the atoms library and the software application. Finally, some concluding remarks and suggestions for future research are described in section 4.

2 Background

Seaport container terminals are complex systems because of several reasons. First of all, there exists a wide variety of organisational forms in regard to terminal operations and used equipment. Many factors have to be considered: How is the layout of the terminal? Which modes of transportation are connected to the terminal? What kind of vessels can be discharged and charged (e.g. maximum size)? Is there a freight station? What kind of equipment is used? Are some processes automated? Are there special areas for storing empty containers, dangerous goods or reefer containers?

Second, after evaluating the organisational form of the terminal, many decisional variables have to be considered, e.g. the number of every kind of equipment and its capacity and the speed of horizontal and vertical transport. In addition, many of these variables may be linked and, as a consequence, influence each other.

Third, static constraints have to be considered as well as dynamic ones. Static constraints are e.g. the number of bays available for the landing of container vessels or the direction of roads on the terminal. Dynamic constraints are for example the work schedules of the staff, the repair schedules of the equipment and the arrival times of ships, trains and trucks at the container terminal.

Fourth, extreme weather conditions and failures of the equipment represent sources of uncertainty, which have to be taken into account.

Because of the complexity, modelling a whole container terminal analytically has proven a challenge [10;11] and the popularity of simulation models for this task has grown significantly [e.g. 6;12;13]. Many of the simulation models focus on one specific area of operation of a container terminal e.g. automated container yard blocks or automated container terminals [14;15], management of berth crane operations [16;17] or the analysis of horizontal transport means [18]. Furthermore, there are some simulation models representing one whole, specific container terminal [11;19].

This article aims to develop a simulation meta-model, which can be applied to represent any chosen seaport container terminal with all relevant elements. In literature, meta-models have been described in two ways. Firstly, they are defined as auxiliary models to the

simulation model which are used to understand its complexity [20;21]. Merkuryeva et al. developed a simulation meta-model for the container terminal in Riga, Latvia. This meta-model is an auxiliary model, which enhances the computational efficiency of the simulation [22].

The second definition, which the present project refers to, describes simulation meta-models as sets of simulation models belonging to a category of systems. The individual simulation model of a system associated to the category, which the meta-model refers to, is given by a particular instance of the meta-model [23;24]. This instance is automatically obtained based on user-specified input data. Boschian et al. used a meta-modelling approach for describing a generic intermodal transport network. The simulation model for a specific network is obtained by identifying an instance of the meta-model [25].

3 The proposed methodology

The object-oriented simulation meta-model presented in this paper is based on the simulation software package “Enterprise Dynamics” (ED) of Incontrol Simulation Solutions.

In the following sections, the overall architecture of the simulation meta-model (section 3.1), the relevant objects (3.2) and the software application (section 3.3) are presented.

3.1 Architecture of the meta-model

The meta-model allows the container terminal manager both to define the topology, the resources and the characteristics of the real container terminal under study and to automatically build the corresponding simulation model. By experimenting on such a model, the container terminal manager can verify in advance the performance of the terminal and makes decisions to improve such performance.

The meta-model involves (see figure 1): (1) a user interface with database; (2) an ad hoc objects library; (3) a software application. Through the user interface, the CT manager specifies both elements and elements’ characteristics (the values of the parameters the elements are characterised by) of the container terminal: the values manually entered by the terminal manager and the ones calculated by the same interface are recorded into the database.

The ad hoc library contains both ED atoms and specifically conceived atoms that represent the building

blocks of a container terminal: each atom is described by data, represented by the atom attributes, and behaviour, modelled by 4Dscript code (4DScript is the programming language ED is based on). The atom attributes are the parameters that characterise the corresponding CT element, while the behaviour is the simulation sub-model, which represents how the corresponding element behaves.

Finally, the software application allows the simulation model to be automatically built. From the database, the application reads the elements of the container terminal and, for each of them, it: (1) selects the corresponding atom from the ad hoc library; (2) selects from the database the values to be assigned to each atom attribute; (3) makes the assignments; (4) inserts into the simulation environment the (parameterised) atom. Once the software application has completed the above-mentioned steps for all the elements of the container terminal, the simulation model is built and the terminal manager can experiment on it.

The atoms library and software application are presented in sections 3.2 and 3.3, respectively. There is no section devoted to either the database or the user interface: actually the database consists of a standard ED table atom, while the interface is given by mere data-entry masks, which allow to specify the values of the container terminal elements parameters, which are the mirror image of the objects data.

3.2 Atoms library

Two different classes of objects belong to the ad hoc library: (1) the roads, which allow to create the network where the transport means characterising the container terminal move; (2) the resources, which perform the activities within the container terminal: gate, service line, rail crane, active transportation equipment (i.e. straddle carrier, reach-stacker, container lift truck), passive transportation equipment (automated guided vehicles, tractor and trailer, multi-trailer), stacking yard and portal crane. Trucks, trains, ships and containers are not considered as objects since they are the entities which flow along the simulation models which in turn can be automatically built by the simulation meta-model.

The objects are described by data and behaviour (as pointed out in section 3.1). As for the data, the attributes related to the atoms of the roads class are: (1) the couple of nodes linked by the road; (2) the parameters required to set possible constraints (direction, number of driving lines, traffic rule and speed factor). The

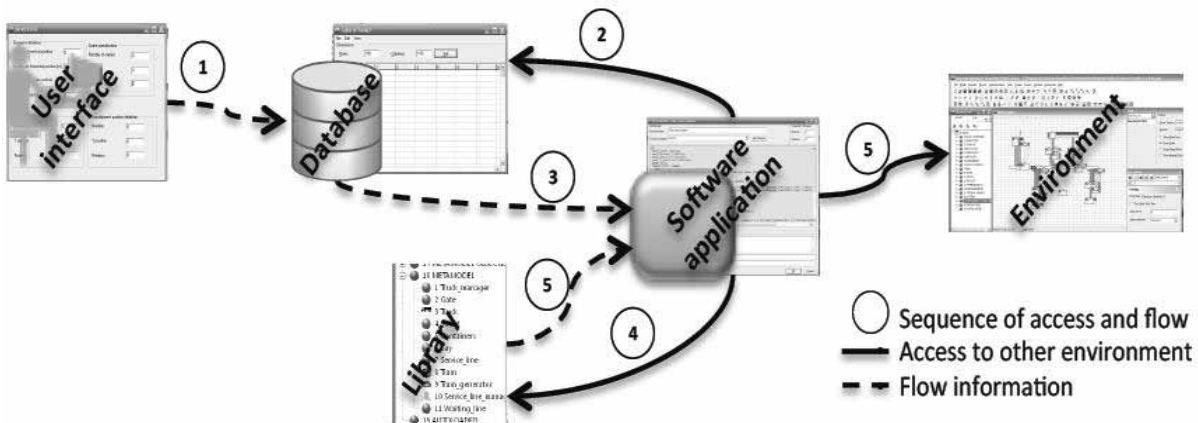


Figure 1. Meta-model architecture

attributes related to the atoms of the resources class basically quantify the terminal resources occupancy (storage capacity, cranes rate, inter-arrival traffic time at piers, etc.).

As for the behaviours, they are the simulation sub-models which represent how the container terminal elements interact with each other and with the entities represented by trucks, trains, ships and containers. To provide an example, table 1 presents the data and behaviour (i.e. the attributes and the simulation sub-model) of the object (i.e. atom) ‘active transportation equipment’ (in particular, due to the functioning of ED, the sub-model is represented by means of state machine diagrams and attributed Petri nets).

3.3 Software application

The software application is divided in two sub-procedures: the first one inserts into the simulation environment the atoms representing the objects belonging to the resources class; the second creates the network by inserting a road atom between all the couples of resource atoms, which represent elements that are included into the real container terminal network.

The first sub-procedure starts by accessing the database, i.e. the table atom, and by reading all the parameters related to the first object belonging to the resources class, which must be created. Then the software application accesses the ad hoc atom library and inserts into the simulation environment the corresponding atom. After that, the application assigns the previously read values to the specific atom. This sub-procedure is repeated for all the resources objects defined by the user. When all the resources objects

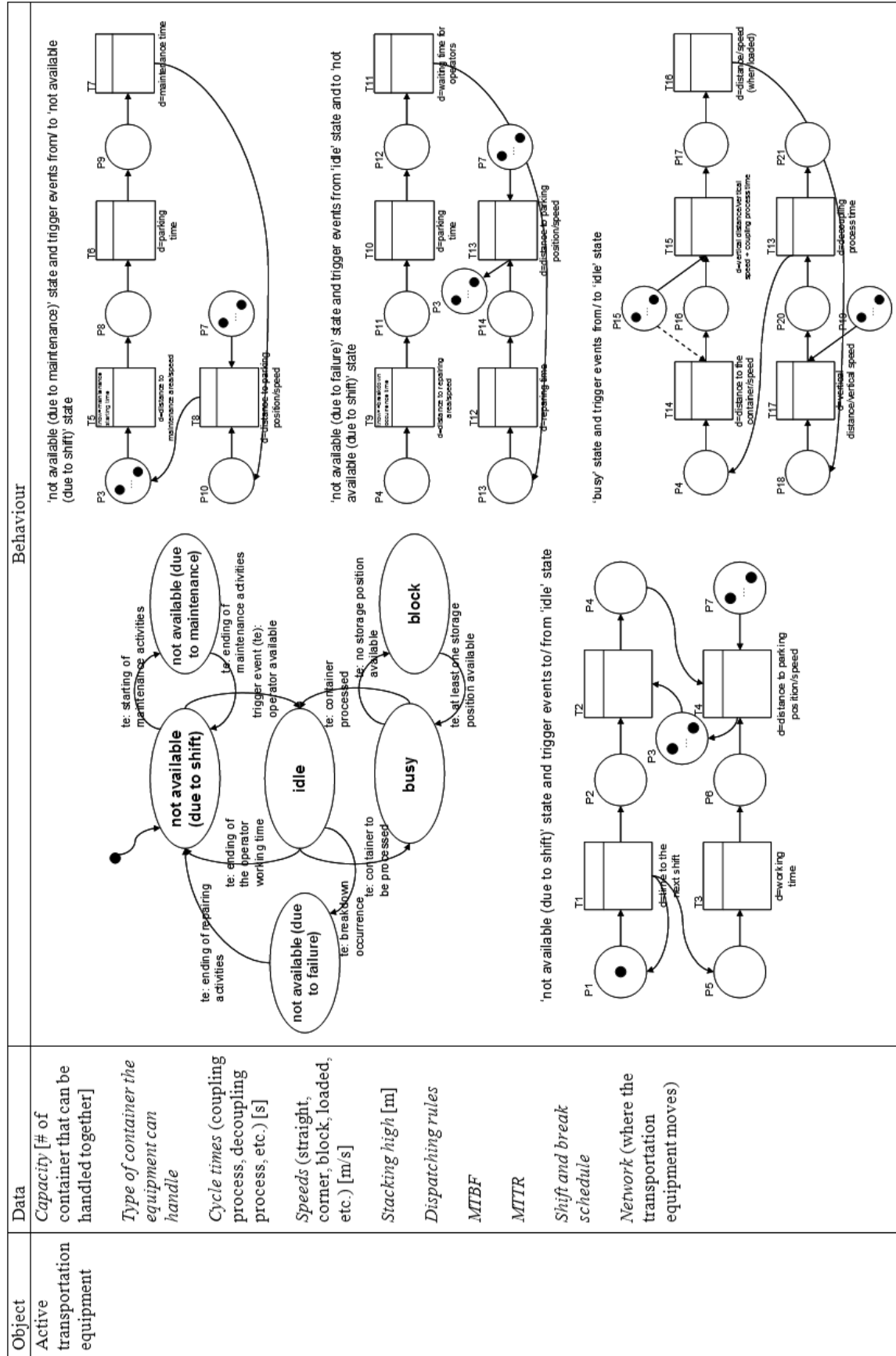
have been created, the first sub-procedure ends and the second one starts, in order to create the network. In particular the sub-procedure reads again from the table atom the parameters (allowed speed, number of driving lines, etc.) that characterize the link between the first couple of resources objects. Then the software application chooses the road atom from the ad hoc library, inserts it into the simulation environment and assigns to the attributes of the atom a value according to the previously read parameters. At that time the second sub-procedure stops, since the network is completely modelled into the simulation environment, the simulation model of the container terminal under study is ready to be used.

4 Conclusion and outlook

This article is focused on the development of a meta-model to support the design and the management policies definition of a seaport container terminal.

The reason to address this problem is given by the fact that, despite simulation is considered one of the most promising tools to support the design and to manage container terminals; its use in real-life contexts is limited by the high requirements for development time and simulation know-how.

To address the problem a joint project is carried on by the Institute of Maritime Logistics of Hamburg University of Technology – MLS (Hamburg, Germany) and Cattaneo University – LIUC and Politecnico di Milano University. The project is structured in four phases (see figure 2). Currently, phase 3 is in progress.



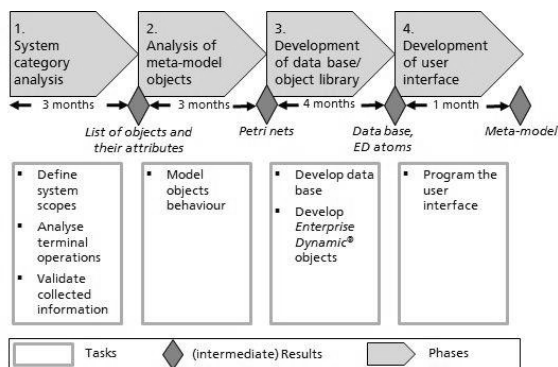


Figure 2: Project phases for meta-model development

The first phase has dealt with the deep analysis of the category of systems which the simulation meta-model refers to, i.e. container terminals. A set of questions have been asked to practitioners to obtain necessary information. In addition, inspections to exemplar maritime intermodal container terminals in Hamburg and Bremerhaven have been performed. Literature has been used to gather further input from other container terminals and to consider all well-known established and planned types of terminal operation systems. The output of the first phase has been identified by a list of all the objects the simulation meta-model must include and the attributes characterizing each object (see paragraph 3.1. For an example of attributes, see column 'Data' of table 1).

In the second phase, the logical model representing the behaviour of each listed object has been developed. In order to facilitate the development process and to ensure the completeness of the logical model a state machine diagram has been constructed for every object. This diagram displays all the states that the object can visit, the trigger events for changing into another state and the action performed before and after the state change. Based on these state machine diagrams the Petri nets formalism has been used to display the internal behaviour during the different states and the behaviour during a state change (for an example, see table 1). The developed state machine diagrams and Petri nets identify the output of the phase.

The third phase of the project, which is in progress, consists of the development of the database and of the ad hoc object library by means of ED. So far, the database has been developed by means of the ED atom 'Table'; the ad hoc atoms representing the objects 'road', 'service line', 'rail crane' 'stacking yard', 'tractor and trailer' have been developed in 4DScript;

a beta version of the software application to automatically build the simulation model has been coded in 4DScript.

In the last phase of the project a user interface will be developed by means of the ED application 'GUI builder'. This interface will allow the user to define the elements composing the container terminal which shall be simulated as well the container terminal topology. Furthermore, it will allow the user to easily enter the necessary values into the database to define the attributes of each system element. The user interface will be built so as that, on the one hand, the user can decide for many of the attributes if he/she wants to enter values or use the deposited ones and, on the other hand, the input of data can as well be done by loading complex lists, e.g. ship arrival times, as by filling out dialog windows. The completed container terminal meta-model will identify the output of the last phase.

The main strengths of the proposed simulation meta-model can be summarised as follows: first, it represents a specifically conceived decision support tool for solving an optimisation problem under several constraints, uncertainty and many interdependent variables as the design or the definition of management policies of a container terminal is.

Second, the proposed meta-model allows to take into account some dynamic features of the terminal system unmanageable by manual calculations (e.g. the arrival times of ships, trains and trucks at the terminal).

Finally, the time required to test a specific configuration of the container terminal system is dramatically reduced: in a few minutes the user can specify the elements the system is composed of as well as the elements' characteristics through the user interface; the software application builds the corresponding simulation model, which can be run, depending on the simulation length and on the hardware, in a few seconds or in a few minutes, on behalf of the decision-making process speed.

The future of the research line outlined here is oriented towards proving the effectiveness of the proposed simulation meta-model by using it to estimate the performance of an existing container terminal system: for this reason, a study of the La Spezia container terminal has been already planned.

5 References

- [1] UNCTAD Secretariat. *Review of Maritime Transport*. United Nations Publication, Switzerland, 2013.
- [2] World Shipping Council. *Value of world seaborne trade*. See <http://www.worldshipping.org/> (accessed 10 July 2014), 2014.
- [3] T.-F. Wang and K. Cullinane. *The efficiency of european container terminals and implications for supply chain management*. *Maritime Econ Logistics* 8, p. 82–99, 2006.
- [4] R. Stahlbock and S. Voß. *Operations research at container terminals: a literature update*. *OR Spectrum* 30, p.1–52, 2008.
- [5] D. Steenken, S. Voß and R. Stahlbock. *Container terminal operations and operations research—a classification and literature review*. *OR Spectrum* 26, p.3–49, 2004.
- [6] P. Ghanbari. *Containerterminal-Logistik*. VDM Verlag Dr. Müller, Germany, 2007.
- [7] L. März and G. Weigert. *Simulationsgestützte Optimierung*. In: *Simulation und Optimierung in Produktion und Logistik*, Editor: L. März, W. Krug, O. Rose und G. Weigert, Springer-Verlag Berlin Heidelberg, Germany, p. 3–12, 2011.
- [8] R. Gibson, B. Carpenter and S. Seeburger. *A flexible port traffic planning model*. *Proceedings of the 1992 Winter Simulation Conference*, in USA, p. 1296–1306, 1992.
- [9] P.-H. Koh, J. L. K. Goh, H.-S. Ng and H.-C. Ng. *Using simulation to preview plans of a container port operations*. *Proceedings of the 1994 Winter Simulation Conference*, in USA, p. 1109–1115, 1994.
- [10] Y. Merkuryev, J. Tolujew, E. Blümel, L. Novitsky, E. Ginters, E. Viktorova, G. Merkuryeva, J. Pronins. *A modelling and simulation methodology for managing the Riga harbour container terminal*. *Simulation* 71, p. 84–95, 1998.
- [11] W. Y. Yun and Y. S. Choi. *A simulation model for container-terminal operation analysis using an object-oriented approach*. *International Journal of Production Economics* 59, p. 221–230, 1999.
- [12] L. M. Gambardella, A. E. Rizzoli and M. Zafalon. *Simulation and planning of an inter-modal container terminal*. *Simulation* 71(2): p. 107–116, 1998.
- [13] T. Franz. *Marktbasiertes Containerterminal-Management - Grundlagen, Mechanismen und verteilte Simulation*. VDM Verlag Dr. Müller, Germany, 2008.
- [14] N. Kemme. *Effects of storage block layout and automated yard crane systems on the performance of seaport container terminals*. *OR Spectrum* 3, p. 563–591, 2012.
- [15] P. Canonaco, P. Legato, R. Mazza and R. Musmanno. *A queuing network model for the management of berth crane operations*. *Computers & Operations Research* 8, p. 2432–2446, 2008.
- [16] C.-I. Liu, H. Julia and P. A. Ioannou. *Design, Simulation, and Evaluation of Automated Container Terminals*. *IEEE Transactions on intelligent transportation systems* 3 no. 1, p. 12–26, 2002.
- [17] J. Dai, W. Lin, R. Moorthy and C.-P. Teo. *Berth allocation planning optimization in container terminal*. Working paper, Georgia Institute of Technology, Atlanta; National University of Singapore, 2004.
- [18] M.B. Duinkerken, R. Dekker, S.T. Kurstjens, J.A. Ottjes and N.P. Dellaert. *Comparing transportation systems for inter-terminal transport at the Maasvlakte container terminals*. *OR Spectrum* 28, p. 469–493, 2006.
- [19] A.A. Shabayek and W.W. Yeung. *A simulation for the Kwai Chung container terminal in Hong Kong*. *European Journal of Operational Research* 140, p. 1–11, 2002.
- [20] L. Friedman. *Simulation metamodel*. Kluwer Academic Publishers, USA, 1996.
- [21] J. Kleijnen. *Statistical tools for simulation practitioners*. M. Dekker, USA, 1987.
- [22] G. Merkuryeva, Y. Merkuryev and J. Tolujew. *Computer Simulation and Metamodelling Of Logistics Processes At A Container Terminal*.

Studies in Informatics and Control vol. 9 no. 1, 2000.

- [23] R. Cigolini and T. Rossi. *Sizing off-shore transshipment systems in dry-bulk transportation*. Production Planning & Control 21, p. 508-522, 2010.
- [24] R. Cigolini, M. Pero and T. Rossi. *An object-oriented simulation meta-model to analyse supply chain performance*. International Journal of Production Research 49, p. 5917-5941, 2011.
- [25] V. Boschian, M. Dotoli, M.P. Fanti, G. Iacobellis and W. Ukovich. *A Metamodeling Approach to the Management of Intermodel Transportation Networks*. IEEE Transactions on Automation Science and Engineering 8 no. 3, p. 457-469, 2011.

Hybride Modellierung fertigungstechnischer Prozessketten mit Energieaspekten in einer ereignisdiskreten Simulationsumgebung

Artur Schmidt, Thorsten Pawletta

Hochschule Wismar, Forschungsgruppe CEA

{artur.schmidt,thorsten.pawletta}@hs-wismar.de

Neben den logistischen und produktionstechnischen Größen spielt der Ressourcenverbrauch bei der Auslegung fertigungstechnischer Prozessketten in einer ereignisdiskreten Simulationsumgebung eine wesentliche Rolle. Hierzu wurden Methoden zur Beschreibung der Ressourcenverbräuche entwickelt. Bisherige Modellierungsansätze basieren auf der messtechnischen Erfassung der Ressourcenverbräuche und der Identifikation wesentlicher Betriebszustände einzelner Werkzeugmaschinen. Im Modell wird jedem Betriebszustand der ermittelte Verbrauch, als konstanter Wert oder durch ein Polynom n-ten Grades approximiert, zugewiesen. Die Modellierungsansätze basieren auf einer umfangreichen Messdatenerfassung mit begrenzter Parametrierungsmöglichkeit. Ein Ansatz zur Reduktion der Menge zu erfassender Daten und der Möglichkeit einer flexiblen Parametrierung wird in der hybriden Modellierung fertigungstechnischer Prozessketten gesehen. Der vorliegende Beitrag beschäftigt sich mit der Modellierung und Simulation hybrider fertigungstechnischer Prozessketten in einer rein ereignisdiskreten Simulationsumgebung. Zur Beschreibung der hybriden Dynamik wird der systemtheoretische *Discrete Event System and Differential Equation System Specification* (DEV&DESS) Ansatz, der die ereignisdiskrete und die kontinuierliche Systemdynamik vereinigt, verwendet. Abschließend wird ein Beispiel zur hybriden Modellierung und Simulation eines Härteofens im MATLAB/SimEvents vorgestellt.

1 Einleitung

Vor dem Hintergrund steigender Preise begrenzter Ressourcen sowie eines zunehmenden Umweltbewusstseins stehen produzierende Unternehmen vor neuen Herausforderungen. Zunehmend werden Methoden der Modellbildung und Simulation eingesetzt, um den Ressourcenverbrauch, die Energieverteilung und Energieumsetzung quantitativ zu erfassen und Potentiale zur Einsparung aufzuzeigen. Neben der detaillierten, ressourcenorientierten Betrachtung einzelner Werkzeugmaschinen mittels kontinuierlicher Simulation [1, 2], ist die Untersuchung ganzer fertigungstechnischer Prozessketten von Interesse [3]. Zur ressourcenorientierten Betrachtung fertigungstechnischer Prozessketten wurde die ereignisdiskrete Modellierung und Simulation (M&S) als praktikabel identifiziert [4]. In [5, 6, 7] wurden erste zeitdynamische ereignisorientierte Modellierungsansätze entwickelt. Die Modellierung der entsprechenden Ressourcenverbräuche erfolgt nach [8] durch die Zerlegung eines Fertigungs- bzw. Produktionsprozesses in eine endliche Anzahl von Betriebszuständen mit zugehörigen Verweildauern. Der Ressourcenverbrauch wird an realen Werkzeugmaschinen oder Produktionsanlagen messtechnisch erfasst, jedem Betriebszustand quantitativ zugeordnet und innerhalb des Betriebszustandes

als konstant angesehen. Komplexe Verläufe, z.B. die Aufnahme der elektrischen Energie, können nur schwer beschrieben werden. Hierzu wurde in [9] der Ansatz von [8] weiter entwickelt. Der Ressourcenverbrauch wird innerhalb eines Betriebszustandes nicht länger als konstant angenommen sondern durch ein Polynom n-ten Grades approximiert und somit genauer abgebildet.

Die entwickelten Modellierungsansätze zeigen, dass neben den logistischen und produktionstechnischen Größen auch die Quantifizierung der Ressourcenverbräuche in ereignisdiskreten Simulationsumgebungen möglich ist. Dies bildet einen ersten Schritt zur ressourceneffizienten Planung von Produktion und Logistik.

Grundlage der Modellierungsansätze bilden Messdaten, die am realen System aufgenommen werden müssen. Die Aufnahme von Messdaten erfolgt meist unter der Annahme, dass die Prozessführung sowie Batch-Größen nicht variierbar sind. Ist zum Beispiel die Planung von Maschinenbelegungen mit variierbaren Batch-Größen von Interesse, so müssen für jede Batch-Größe Messdaten vorliegen. Dies trifft besonders auf Wärmebehandlungen zu. Kommt die Betrachtung unterschiedlicher Prozessführungsstrategien in Form von unterschiedlichen Heizprogrammen

hinzu, müssen für jede Batch-Größe pro Strategie Messdaten erhoben werden. Daraus folgt, dass eine Betrachtung der obigen Problemstellungen auf Basis rein messdatenbasierter Modellierungsansätze für Batch-Prozesse sehr aufwändig ist.

Zur Lösung solcher Problemstellungen wird analog zu [10] die hybride M&S gesehen.

Basierend auf den Arbeiten von [6, 11] sowie dem *Discrete Event System and Differential Equation System Specification* (DEV&DESS) [12, 13] wird ein hybrider Modellierungsansatz vorgestellt. Anders als in [5, 14, 15, 10] soll keine Kopplung unterschiedlicher Simulatoren stattfinden. Die hybride M&S erfolgt in einer rein ereignisdiskreten Simulationsumgebung.

Als ereignisdiskreter Simulator wird MATLAB/SimEvents [16] ausgewählt. Es ist allgemein bekannt, dass MATLAB die M&S hybrider Modelle unterstützt. Der vorgestellte Ansatz erfolgt ohne Verwendung der kontinuierlichen Modellbibliothek von MATLAB.

Ausgehend von der transaktionsorientierten Prozesskettenmodellierung in MATLAB/SimEvents wird auf den systemtheoretischen Ansatz zur Beschreibung hybrider Systeme eingegangen. Darauf aufbauend wird die allgemeine Umsetzung dieser Ansätze in einer ereignisdiskreten Simulationsumgebung sowie konkret in MATLAB/SimEvents gezeigt. Anschließend wird das gesamte Konzept an einem Anwendungsbeispiel demonstriert. Hierbei erfolgt eine vereinfachte hybride Modellierung und Simulation eines Härteofens. Abschließend erfolgen eine Zusammenfassung und ein Ausblick auf fortlaufende Arbeiten.

2 Transaktionsorientierte Modellierung von Prozessketten

Die transaktionsorientierte Simulation stellt eine Unterklasse der ereignisdiskreten Simulation dar. Dieser Simulationsansatz definiert grundlegend nur zwei Komponententypen: (i) aktive Komponenten und (ii) passive Komponenten [13]. Aktive Komponenten (Entitäten) bewegen sich zwischen den passiven Komponenten (Blöcken), werden verzögert und lösen Ereignisse sowie Zustandsübergänge aus. Die transaktionsorientierte Simulation ist besonders zur Modellierung von Warteschlangensystemen [17] und somit zur Modellierung von fertigungstechnischen Prozessketten geeignet. Dabei werden die Werkstücke

als Entitäten und die Fertigungsverfahren als Blöcke abgebildet.

Zur Modellierung und transaktionsorientierten Simulation von fertigungstechnischen Prozessketten wurde in [6] eine Modellbibliothek (MB) für MATLAB/SimEvents [16] vorgestellt und in [11] weiterentwickelt. Die Komponenten der MB stellen konkrete Fertigungsverfahren, Basismodelle (BM) genannt, dar und gliedern sich in drei Grundtypen: (i) CNC-Satzbasierte BMs und (ii) messdatenbasierte BMs. Bei den CNC-Satzbasierten BMs werden die Betriebszustände, Zustandsübergänge und die Ressourcenverbräuche direkt auf Basis der CNC-Programme berechnet. Messdatenbasierte BMs berechnen ihr dynamisches Verhalten und den Ressourcenverbrauch aus Messdaten der realen Fertigungseinrichtung. In Verbindung mit einer Quelle und einer Senke stellt jedes BM ein ausführbares Simulationsmodell dar. Die Quelle dient der Generierung und die Senke der Terminierung von Werkstücken. Die werkstückspezifischen Eigenschaften wie Werkstofftyp und Werkstückgeometrie werden als Attribute der Entitäten modelliert.

Die BMs lassen sich problemlos mit den Komponenten der MATLAB/SimEvents eigener Modellbibliothek koppeln und es können komplexe fertigungstechnische Prozessketten modelliert werden. In Abbildung 1 ist beispielhaft eine fertigungstechnische Prozesskette, bestehend aus den Basismodellen Außenlängsdrehen (ALD), Vakuumhärten (VH), Anlassen (A), Außenrundscheifen (ARS) sowie einer Quelle, einer Senke und vier FIFO Puffern, dargestellt.

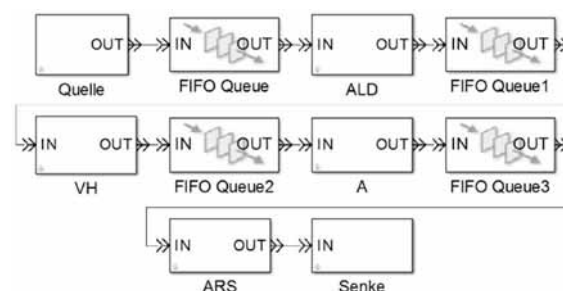
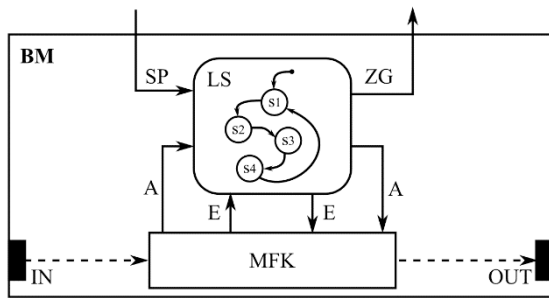


Abbildung 1 Prozesskette in MATLAB/SimEvents.

Die messdatenbasierten und die CNC-Satzbasierten Basismodelle besitzen einen ähnlichen internen Aufbau und bestehend aus Materialflusskomponenten (MFK) sowie einer lokalen zustandsbasierten Steuerung (LS) (Abbildung 2).



BM....Basismodell; SP....Systemparameter; E...Ereignis;
ZG....Zustandsgrößen; LS....lokale Steuerung; A....Attribute
IN/OUT....Ein-/Ausgangspore der Entitäten;
MFK....Materialflusskomponenten

Abbildung 2 Interner Aufbau messdabasierter und CNC-Satzbasierter Basismodelle.

Werkstücke gelangen ereignisorientiert in die MFK, aktivieren durch Ereignisse die LS und werden solange verzögert bis die in der LS kodierte Fertigungsaufgabe sowie die Berechnung der Ressourcenverbräuche abgeschlossen ist. Weiterhin sind Schnittstellen zum Informationsaustausch zwischen MFK und LS realisiert. Beim Eintritt eines Werkstückes werden Attribute, wie zum Beispiel die aktuelle Werkstückgeometrie und der Werkstofftyp, ausgelesen und an die LS übermittelt. Bevor die Werkstücke das BM verlassen, können wichtige Informationen als Attribute auf das Werkstück gespeichert werden.

Der berechnete Ressourcenverbrauch wird als Zustandsgrößenvektor an die Simulationsumgebung zur weiteren Verarbeitung übermittelt.

3 Hybride Modellierungsansätze

Ausgehend von [12, 13] existieren in der Systemtheorie drei unterschiedliche Systemtypen: (i) *Differential Equation System Specification* (DESS), (ii) *Discrete Time System Specification* (DTSS) und (iii) *Discrete Event System Specification* (DEVS). Die Systemtypen unterscheiden sich hinsichtlich ihrer Beschreibung und der Änderung der Zustandsgrößen. Charakteristisch für DESS ist die Beschreibung der Systemdynamik mit Differentialgleichungen und die kontinuierliche Änderung der Zustandsgrößen. DTSS Systeme zeichnen sich durch die Beschreibung der Systemdynamik mit Differenzengleichungen und einer Zustandsänderung zu diskreten Zeitpunkten aus. Die Beschreibung der DEVS Systeme erfolgt auf Basis einer endlichen Menge von Zuständen, Verweildauern in Zuständen und Ereignissen. Zustandsänderungen finden ausschließlich zu Ereigniszeitpunkten statt. Die Kombination unterschiedlicher Systemtypen wird als hybrides System bezeichnet.

DESS und DEVS werden am häufigsten zur Beschreibung ingenieurtechnischer Problemstellungen verwendet [18]. Die Kombination beider Systeme kommt dann zum Einsatz, wenn eine reine DESS oder DEVS Beschreibung der Problemstellung nicht ausreicht oder nicht möglich ist.

Im nächsten Abschnitt erfolgt eine nähere Betrachtung systemtheoretischer Ansätze zur Beschreibung hybrider Systeme nach [12, 13]. Im Weiteren wird auf die formale Spezifikation der Systeme verzichtet und auf die originäre Literatur [12, 13] verwiesen.

3.1 Systemtheoretische Ansätze zur Beschreibung hybrider Systeme

Zur Beschreibung hybrider Systeme existieren grundlegend zwei Ansätze (Abbildung 3) in der Systemtheorie: (i) *Combining Formalism* und (ii) *Embedding Formalism*.

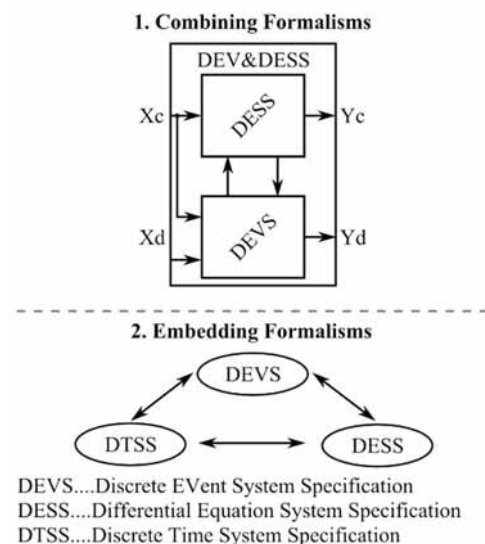


Abbildung 3 Systemtheoretische Ansätze zur Beschreibung hybrider Systeme nach [12, 13].

Beim *Embedding Formalism* Ansatz wird ein System in ein anderes eingebettet. Als Beispiel kann das numerische Lösen von Differentialgleichungen in einer zeitdiskreten Simulationsumgebung genannt werden. Hierbei wird ein DESS in ein DTSS eingebettet.

Der *Combining Formalism* ist die Vereinigung von DESS und DEVS zu *Discrete Event System and Differential Equation System Specification* (DEV&DESS). DEV&DESS ermöglicht eine kombiniert kontinuierliche und ereignisdiskrete Beschreibung von Systemen und ist vorrangig ereignisdiskret. Zur klaren Trennung der Systembeschreibungen wird

die kontinuierliche Systemdynamik im DESS und die ereignisdiskrete im DEVS beschrieben.

Sowohl das DESS als auch das DEVS besitzen eigene Eingangsports (X_d , X_c) und Ausgangsports (Y_d , Y_c). Zusätzlich werden die kontinuierlichen Ports X_c als Eingang zum DEVS verwendet, um das Erkennen von Zustandsereignissen zu ermöglichen. Weiterhin besitzen DESS und DEVS interne Kopplungen. Dabei werden zum Beispiel kontinuierliche Zustandsgrößen vom DESS im DEVS auf Zustandsereignisse überprüft. Das DEVS kann dann mit diskreten Ereignissen auf das DESS einwirken und beispielsweise Differentialgleichungen umschalten.

Zur Ausführung von DEV&DESS Systemen wird ein spezieller DEV&DESS Simulator verwendet [13]. Die Zeitfortschaltung erfolgt gemischt ereignisdiskret (tDEVs) sowie zeitdiskret (tDESS) mit einer konstanten oder variablen Integrationsschrittweite h (Abbildung 4).

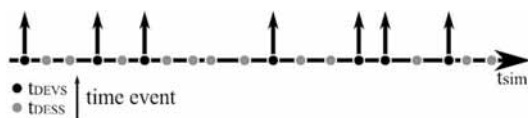


Abbildung 4 Zeitvorschaltung eines DEV&DESS Simulators.

Die numerische Integration der Differentialgleichungen erfolgt sowohl zu ereignisdiskreten als auch zeitdiskreten Zeitpunkten.

Im nächsten Abschnitt wird gezeigt, wie ein DEV&DESS System in einer reinen ereignisdiskreten Simulationsumgebung modelliert werden kann.

3.2 Spezifikation von DEV&DESS in einer ereignisorientierten Simulationsumgebung

Bei der Modellierung von DEV&DESS in einer ereignisdiskreten Simulationsumgebung muss auch eine numerische Integrationsmethode für die kontinuierliche Modelldynamik auf Modellebene abgebildet werden.

Abbildung 5 zeigt den schematischen Aufbau eines DEV&DESS Modells und einer numerischen Integrationsmethode in Form eines Zustandsautomaten nach Harel [19].

Das Modell besteht aus drei parallelen Zuständen (PZ) *DEVs*, *DESS* und *NumODESolver*. Die Modellierung der drei Zustände basiert, analog zu DEVS Systemen, auf einer Menge von Zuständen, Verweildauern im Zustand und Ereignissen. Weiterhin finden

Zustandsübergänge ausschließlich zu Ereigniszeitpunkten statt.

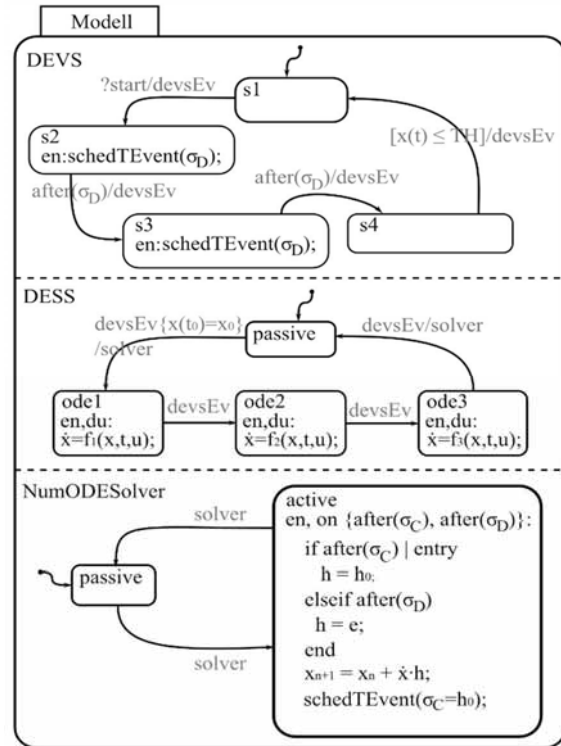


Abbildung 5 Schematische Darstellung eines DEV&DESS Modells und dessen Modelldynamik.

Ein solches Modell ist vorrangig ereignisdiskret. D. h., *DESS* und *NumODESolver* sind dem *DEVs* untergeordnet. Die Kommunikation zwischen den einzelnen PZs findet auf Basis lokaler Ereignisse *devsEv* und *solver* statt.

In Abbildung 5 sind im *DEVs* PZ vier Zustände *s1* bis *s4* dargestellt. Der Ausdruck *schedTEvent(σD)* kennzeichnet die Verweildauer im Zustand. Es wird ein Zeitereignis $t_{next_event} = t + \sigma_D$ eingeplant. Nach dem Auftreten des Zeitereignisses, mit *after(σD)* gekennzeichnet, wird der entsprechende Zustand verlassen. Weiterhin können Zustandswechsel auf Basis von Zustandsübergangsbedingungen erfolgen. Hierbei wird ein Zustand erst verlassen, wenn die entsprechende Bedingung erfüllt ist. Der Zustandswechsel von *s4* zu *s1* kann nur erfolgen, wenn die zugehörige Zustandsübergangsbedingung $[x(t) \leq TH]$ wahr wird, also die Zustandsvariable einen bestimmten Grenzwert erreicht oder unterschreitet. Jeder Zustandswechsel löst weiterhin ein lokales Ereignis aus.

Im *DESS* PZ befinden sich vier Zustände *passive* und *ode1* bis *ode3*. Letztere beinhalten Differentialgleichungen der Form

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

mit $x(t_0) = x_0$

und sind den *DEVS* Zuständen untergeordnet. Darüber hinaus werden hier Anfangswerte für die entsprechenden Differentialgleichungen festgelegt. Die Umschaltung der Differentialgleichungen findet auf Basis der lokalen Ereignisse statt.

Der PZ *NumODESolver* kodiert zwei Zustände *passive* und *active* sowie die numerische Integrationsmethode explizit Euler

$$x_{n+1} = x_n + \dot{x} \cdot h \quad (2)$$

mit der Schrittweite h . Die numerische Integration erfolgt ausschließlich im Zustand *active*. Jeder Integrationszeitpunkt wird dabei als Zeitereignis eingeplant $t_{next_event} = t + \sigma C$ und ist durch *schedTEvent*($\sigma C = h$) gekennzeichnet. Die numerische Integration findet beim erstmaligen Betreten des *active* Zustandes und beim Auftreten von Zeitereignissen in Folge σD oder σC statt. Die Variable $h0$ steht für eine konstante Schrittweite und e entspricht der verstrichenen Zeit seit der letzten Integration.

Die Modellarbeitung beginnt durch das Auftreten des externen *?start* Ereignisses. Als Folge von *?start* findet in allen drei PZs ein Zustandsübergang statt. Dabei trägt sich das *DEVS* mit σD und *NumODESolver* mit σC in der Ereignisliste der Simulationsumgebung ein. Bei jeder Aktivierung des Modells werden in *DESS* die Änderungsraten \dot{x} berechnet. Vor der numerischen Integration wird die Integrationsschrittweite an das entsprechende Zeitereignis angepasst.

Ein Zustandswechsel von *s4* nach *s1* im *DEVS* sorgt dafür, dass *DESS* und *NumODESolver* in ihre Anfangszustände übergehen. Es werden keine Ereignisse mehr in die Ereignisliste der Simulationsumgebung eingetragen und die Simulation somit beendet.

3.3 Umsetzung in MATLAB/SimEvents

Das MATLAB/SimEvents Modell orientiert sich an der Abbildung 2 aus Kapitel 2 und gliedert sich in drei Teile: (i) Materialflusskomponenten (MFK), (ii) lokale zustandsbasierte Steuerung (LS) und (iii) einem Subsystem zum Einplanen von Zeitereignissen (*schedTEvent*) (Abbildung 6).

Die hybride Modellbeschreibung erfolgt in einem Stateflow Statechart (LS) durch die drei parallelen Zustände *DEVS*, *DESS* und *NumODESolver*.

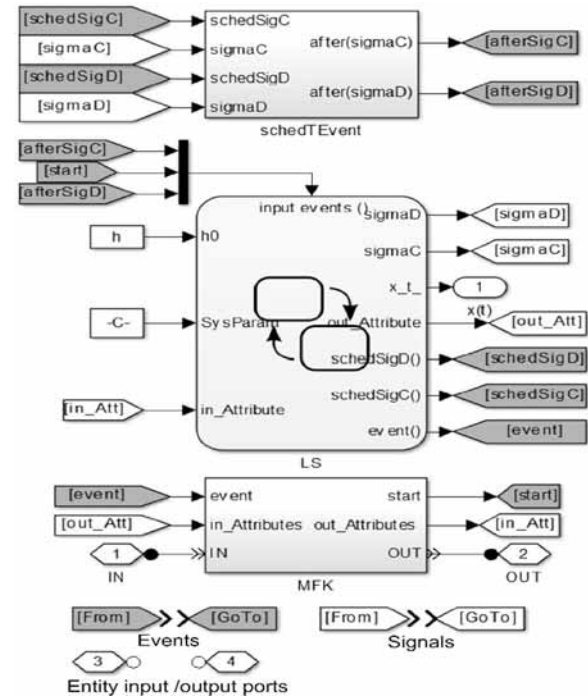


Abbildung 6 Modellstruktur eines DEV&DESS Modells in MATLAB.

Da Stateflow keine interne Zustandsüberföhrungsfunktion besitzen, erfolgt die Aktivierung und Abarbeitung ausschließlich auf Basis von externen Ereignissen *afterSigC*, *afterSigD* und *start*.

Für die Aktivierung der LS in Folge σC und σD sorgt das Subsystem *schedTEvent* (Abb. 7).

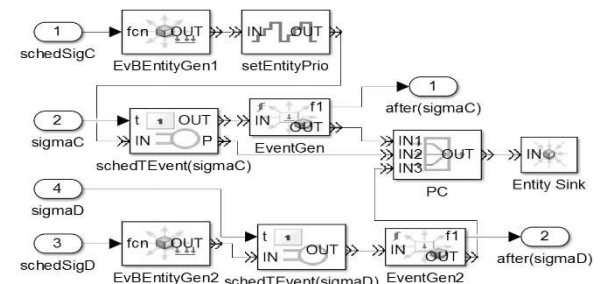


Abbildung 7 Subsystem schedTEvent.

Dieses erhält von der LS die entsprechenden Zeiten sowie ein *schedSigD* oder *schedSigC* Ereignis. Auf Basis von *schedSigD* und *schedSigC* werden in den *EvEntityGen* Blöcken Steuerentitäten (SE) erzeugt, die in den Single Servern *schedTEvent(sigmaC)* und *schedTEvent(sigmaD)* für die gewünschte Zeitdauer

verzögert werden. Danach verlassen die SEs die Server, erzeugen in den *EventGen* Blöcken ein *afterSigD* oder *afterSigC* Ereignis und werden in der *Entity Sink* terminiert. Die erzeugten Ereignisse aktivieren erneut die LS.

Zur Anpassung der Integrationsschrittweite an σD oder σC wird jeder SE, die in Folge *schedSigC* erzeugt wird, eine Priorität zugewiesen. Ist der Single Server *schedTEvent(sigmaC)* mit einer SE belegt und es wird eine neue generiert, so verlässt die aktuelle SE den Server über den *Preemption Port* (P). Hierbei wird kein Ereignis für die LS generiert. Dies geschieht dann, wenn die LS in Folge *after(sigmaD)* aktiviert wird und eine Integration der Differentialgleichung erfolgen muss.

Die Modellabarbeitung startet, wenn die gewünschte Anzahl von Werkstücken in das MFK Subsystem gelangt ist und das Ereignis *start* ausgelöst wird. Ist die Abarbeitung zu Ende, wird ein *event* Ereignis an MFK gesendet. Die Werkstücke können dann den Block verlassen.

4 Anwendungsbeispiel und Simulationsergebnisse

Nachfolgend wird ein vereinfachtes hybrides Modell eines Härteofens mit zugehörigen Simulationsergebnissen gezeigt.

4.1 Vereinfachtes Modell eines Härteofens

Als Anwendungsbeispiel soll ein vereinfachtes Modell eines Härteofens dienen. Die Änderung des Wärmeinhaltes des Härteofens ergibt sich aus der Differenz der zugeführten und abgeführten Wärmeleistung.

$$\begin{aligned} \frac{dQ_o}{dt} &= \dot{Q}_{zu} - \dot{Q}_{ab} \\ \text{mit } \frac{dQ_o}{dt} &= m \cdot c \cdot \frac{dT_o}{dt} \end{aligned} \quad (3)$$

Es wird angenommen, dass die Wärme ausschließlich über den Wärmedurchgang abgeführt und durch die Heizleistung des Ofens zugeführt wird. Das Temperaturmodell eines Härteofens ergibt sich nach Gleichung 3.

$$\frac{dT_o}{dt} = \frac{\dot{Q}_H}{m \cdot c} - \frac{k \cdot A}{m \cdot c} \cdot (T_o(t) - T_{umg}) \quad (4)$$

Die einzelnen Variablen sind in der nachstehenden Tabelle aufgeführt.

Name	Bedeutung
m	Masse
c	Spez. Wärmekapazität
k	Wärmedurchgangskoeffizient
A	Wandfläche
\dot{Q}_H	Heizleistung
T_{umg}	Umgebungstemperatur

Tabelle 1. Auflistung der Modellvariablen

Abbildung 8 zeigt das vereinfachte DEV&DESS Modell des Härteofens. Die Modellierung erfolgt analog zum Abschnitt 3.2 in drei parallelen Zuständen *DEVs*, *DESS* und *NumODESolver*, wobei auf die Darstellung von *NumODESolver* verzichtet wird.

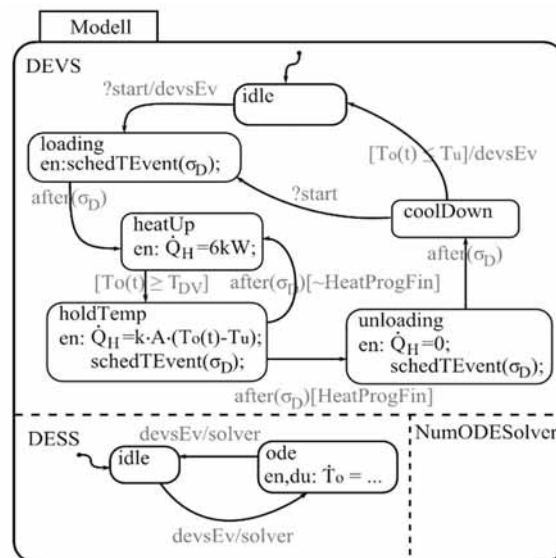


Abbildung 8 Vereinfachtes DEV&DESS Modell eines Härteofens

Die ereignisdiskrete Modelldynamik wird mit sechs Zuständen *idle*, *loading*, *heatUp*, *holdTemp*, *unloading* und *coolDown* beschrieben. Darüber hinaus wird einzelnen Zuständen die Heizleistung zugewiesen und beim Zustandswechsel ereignisdiskret umgeschaltet. Die maximale Heizleistung wird mit 6 kW im Zustand *heatUp* definiert. Im Zustand *holdTemp* entspricht die Heizleistung dem Wärmedurchgangsverlust. Ein Zustandswechsel von *heatUp* nach *holdTemp* kann erst erfolgen, wenn der Härteofen eine Solltemperatur erreicht hat. Diese wird in einem Heizprogramm (Temperatur-Zeit-Verlauf) definiert. Weiterhin kann der Härteofen erst entladen werden, wenn das Heizprogramm vollständig abgearbeitet wurde. Im Modell ist das mit den Zustandsübergangsbedingungen *[HeatProgFin]* (*HeatProgramFi-*

nished) und [*~HeatProgFin*] (*HeatProgramNotFinished*) realisiert.

4.2 Ausgewählte Simulationsergebnisse

Es sollen 60 Werkstücke, die zum Zeitpunkt $t=t_0$ vorliegen, gehärtet werden. Es wird weiterhin festgelegt, dass maximal 10 Werkstücke gleichzeitig gehärtet werden können und die Anfangstemperatur des Ofens der Umgebungstemperatur entspricht. Zum Beladen und Entladen des Ofens werden jeweils 10 Minuten vorgesehen. Zusätzlich wird ein Heizprogramm definiert. Die Ofentemperatur soll in 30 min von 20 °C auf 400 °C aufgeheizt und 40 min lang gehalten werden. Danach erfolgt die nächste Heizphase von 400 °C auf 800 °C in 40 min. Anschließend soll die Temperatur für die Dauer von 30 min gehalten werden.

Das Simulationende erfolgt, wenn alle Werkstücke gehärtet wurden und der Ofen auf die Umgebungstemperatur abgekühlt ist.

Abbildung 9 zeigt die Simulationsergebnisse in Form von Ofentemperatur, Heizleistung und die diskreten Ofenzustände über der Zeit.

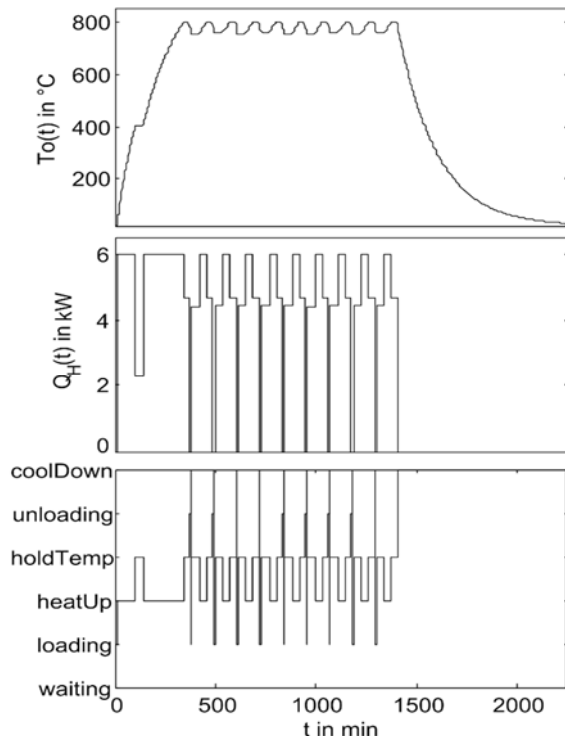


Abbildung 9 Simulationsergebnisse.

Wie erwartet wird zum Härten der ersten Charge mehr Energie benötigt als für die weiteren, da der

Härteofen erst auf Betriebstemperatur aufgeheizt werden muss.

Der Temperaturverlauf zeigt weiterhin, dass der Ofen die geforderten Temperaturen erreicht, jedoch nicht in der vordefinierten Zeit (s. Heizprogramm). Somit können auch Heizprogramme auf Plausibilität und Durchführbarkeit geprüft werden. Durch die Variation der Systemparameter kann ein Härteofen identifiziert werden, der das gewünschte Heizprogramm einhalten kann.

Anhand des Abkühlverhaltens können zum Beispiel optimale Rüstzeiten ermittelt werden. Darüber hinaus können Informationen bezüglich der Instandhaltung gewonnen werden, um Stillstandszeiten zu minimieren.

Die Untersuchung solcher Problemstellungen ist mit den Ansätzen von [8, 9] mit reinen Messdaten sehr schwierig bis nicht durchführbar.

5 Zusammenfassung und Ausblick

Ausgehend von den systemtheoretischen Konzepten wurde eine Modellierungsmethode zur Beschreibung hybrider Modelle in einer rein ereignisdiskreten Simulationsumgebung gezeigt. Die Methode basiert auf dem *Discrete Event System and Differential Equation System Specification* (DEV&DESS) Ansatz, der die ereignisdiskrete und die kontinuierliche Systemdynamik vereinigt. Zur Umsetzung in einer ereignisdiskreten Simulationsumgebung, muss neben dem DEV&DESS auch die numerische Integrationsmethode auf Modellebene abgebildet werden. Dabei werden neben den ereignisdiskreten auch die zeitdiskreten Zeitpunkte der numerischen Integrationsmethode in die Ereignisliste der Simulationsumgebung eingeplant. Die Umsetzung der Methode wurde am Beispiel eines vereinfachten Modells eines Härteofens in MATLAB/SimEvents demonstriert.

Zukünftig soll eine verbesserte numerische Integrationsmethode wie zum Beispiel die *Runge-Kutta-Fehlberg 45* Methode integriert werden. Weiterhin sollen verbesserte hybride Modelle von wärmebehandelnden Verfahren entwickelt werden. Diese sollen auch die Betrachtung unterschiedlicher Werkstücke und deren werkstoffbezogenen sowie geometrischen Eigenschaften berücksichtigen. Die hybriden Modelle werden dann zur Untersuchung komplexer Prozessketten bezüglich produktionstechnischer und ressourcenorientierter Aspekte eingesetzt.

Danksagung

Ein besonderer Dank gilt dem Kooperationspartner Stiftung Institut für Werkstofftechnik Bremen Abteilung Fertigungstechnik und der Deutsche Forschungsgemeinschaft (Fördernummer: BR 825/62-2 und PA 631/2-2).

6 References

- [1] B. Heinzl, „Objektorientierte Multi-Domain-Modellierung und Simulation von Werkzeugmaschinen,“ Diplomarbeit, Tu Wien, 2012.
- [2] R. Diekmann, J. Heinze, I. Alkov und D. Weidemann, „Control Strategies for Energy-Optimized Operation of Assembly Systems,“ in *ASIM-Workshop STS/GMMS 2014, Treffen der ASIM/GI-Fachgruppen: Simulation technischer Systeme Grundlagen und Methoden in Modellbildung und Simulation*, Reutlingen, ARGESIM/ASIM, TU Wien, 2014, pp. 217-223.
- [3] R. Neugebauer, „Energieeffizienz in der Produktion, Untersuchung zum Handlungs- und Forschungsbedarf,“ Frauenhofer Gesellschaft, 2008.
- [4] B. Stahl, M. Taisch, A. Cannata, F. Müller, S. Thiede und C. Herrmann, „Combined Energy, Material and Building Simulation for Green Factory Planning,“ in *20th CIRP International Conference on Life Cycle Engineering*, Singapore, 2013.
- [5] M. Junge, „Simulationsgestützte Entwicklung und Optimierung einer energieeffizienten Produktionssteuerung,“ Kassel university press GmbH, 2007, Diss., Univ. Kassel, 2007.
- [6] R. Larek, „Ressourceneffiziente Auslegung von fertigungstechnischen Prozessketten durch Simulation und numerische Optimierung,“ Shaker Verlag, 2012, Diss., Univ. Bremen, 2012, 2012.
- [7] A. Schlegel, J. Stoldt und M. Putz, „Erweiterte Integration energetischer Betrachtungen in,“ in *In: Proc. 15. ASIM Fachtagung Simulation in Produktion und Logistik, Hrsg.: Dangelmaier, Laroque, Klaas; Paderborn, 9.-11.10.2013*, HNI Paderborn, HNI-Verlagsschriftenreihe, 2013, 2013, pp. 187-196.
- [8] D. A und V. A., „Energy Consumption Forecasting and Optimisation For Tool Machines,“ *MM Science Journal* (2009), pp. 63-67, März 2009.
- [9] N. Weinert, „Vorgehensweise für Planung und betrieb energieeffizienter Produktionssysteme,“ Frauenhofer verlag, 2010, Diss., TU Berlin, 2010.
- [10] S. Wenzel, M. Junge, C. Pöge und S. Spieckermann, „Energieeffizienz in der Automobilindustrie. Ganzheitliche simulationsgestützte Planung und Bewertung,“ *Productivity Management*, Bd. 4, Nr. 18, pp. 21-24, 2013.
- [11] A. Schmidt, „Entwicklung eines Simulators zur energetischen Bewertung von Prozessketten der spanenden Bauteilfertigung mit MATLAB/SimEvents,“ Master-Thesis, Hochschule Wismar, Wismar, 2012.
- [12] F. Pichler und H. Schwärtzel, *CAST Methodes in Modelling. Computer Aided System Theory for the Design of Intelligent Machines.*, Springer-Verlag Verlin Heidelberg 1992, 1992.
- [13] B. Zeigler, H. Praehoffer und K. T.G., *Theory of Modelling and Simulation.*, Second Edition: Elsevier Academic Press, 2000.
- [14] I. Leobner, W. Kastner, I. Kovacic, B. Heinzl, M. Rössler, F. Dür, T. Flatz, K. Ponweiser und F. Bleicher, „Interdisziplinäre Optimierungsstrategien zur Energieeffizienz in Produktionsbetrieben,“ in *8. Internationale Energiewirtschaftstagung an der TU Wien, 13.-15. Februar 2013*, TU Wien, 2013.
- [15] M. Helal, „A Hybrid System Dynamics-Discrete Event Simulation Approach to Simulating the Manufacturing Enterprise,“ Diss. University of Central Florida, Orlando, 2008.
- [16] MathWorks, „SimEvents: Model and simulate discrete-event systems,“ The MathWorks, 30 Juni 2014. [Online]. Available: <http://www.mathworks.de/products/simevents/>.
- [17] F. Mattern und H. Mehl, „Diskrete Simulation - Prinzipien und Probleme der Effizienzsteigerung,“ *Informatik Spectrum*, Nr. 12, pp. 198-210, 189.
- [18] V. S. Prat, „Hybrid System Modeling Using the Parallel DEVS Formalism and the Modelica Language,“ Diss. , National University of Distance Education, 2010.
- [19] D. Harel, „Statecharts: A Visual Formalism for Complex Systems,“ *Science of Computer Programming*, Nr. 8, pp. 231-274, 1987.

Entwicklung einer Fertigungssimulation für einen Schmelz- und Druckgussbetrieb zur Steigerung der Energieeffizienz und Prozesssicherheit

Sven Hirschberg, Wolfgang Schlüter, Ansgar Ringleb

Hochschule Ansbach

s.hirschberg@hs-ansbach.de

Das Kompetenzzentrum Industrielle Energieeffizienz (KIEff) der Hochschule Ansbach arbeitet im Rahmen des Forschungsprojektes „Smart-Melting“ gemeinsam mit den Partnern ZF Gusstechnologie GmbH und Bundesverband der Deutschen Giesserei-Industrie an einer Fertigungssimulation für Aluminium-Druckguss- und Schmelzbetriebe. Mit Hilfe dieser Fertigungssimulation werden strategische und operative Fragestellungen gelöst sowie energieeffiziente und prozesssichere Fertigungsstrategien für Aluminium-Druckguss und Schmelzbetriebe entwickelt. Die besondere Schwierigkeit im Simulationsaufbau liegt dabei in der Kombination einer großen Anzahl von kontinuierlichen und ereignisgesteuerten Komponenten zu einem hybriden System. Die Umsetzung erfolgt mit Hilfe von Matlab/Simulink und dem ebenfalls von Mathworks entwickelten Simulationstool für ereignisgesteuerte Systeme Stateflow.

Der gasbetriebene Schmelzprozess stellt einen der energieintensivsten Prozesse in Druckgussbetrieben dar. Durch die Abhängigkeit des Schmelzbetriebes vom Druckgussbetrieb und der vorhandenen Kommunikationsstruktur entsteht eine diskontinuierliche Betriebsweise mit hohem Energieeinsparpotential. Dieses lässt sich mit Hilfe einer Fertigungssimulation bestehend aus Material- und Energieflusssimulation ermitteln. Für die Energieoptimierung und der Steigerung der Prozesssicherheit wird mit Hilfe der Erkenntnisse aus Material- und Energieflusssimulation die Fertigungssteuerung im Modell optimiert.

1 Einleitung

Im Fokus des Forschungsprojektes Smart Melting steht die Steigerung der Energieeffizienz in Druckgussbetrieben der Aluminiumindustrie. Durch eine Anbindung der Industrie an ein Smart-Grid, die eine Anpassung der Produktion an die zur Verfügung stehende Strommenge zur Folge hat, steigen dabei die innerbetrieblichen Anforderungen an die Fertigungssteuerung.

Ein wichtiges Werkzeug zur Analyse der Fertigungsprozesse ist die Fertigungssimulation, mit der verschiedene Szenarien in unterschiedlichen Zeitfenstern untersucht werden können. Dazu ist zunächst der Ist-Zustand in der Industrie genau zu analysieren. Hierfür wird in einem repräsentativen Industriebetrieb eine umfassende Datenaufnahme zu relevanten Komponenten durchgeführt. Der Betrieb des Industriepartners wird simulationstechnisch abgebildet und mit Hilfe der Informationen aus der Datenaufnahme validiert.

Aufgrund der in vielen Betrieben vorhandenen Kommunikationsstruktur zwischen Schmelz- und

Druckgussbetrieb und dem Vorrang der Gussproduktion resultiert eine diskontinuierliche Betriebsweise des Schmelzbetriebes. Diese Betriebsweise weist ein beachtliches Energieeinsparpotential auf, das mit Hilfe der Fertigungssimulation identifiziert werden kann.

2 Der reale Betrieb

Die Fertigungssimulation bildet den momentanen Ablauf in einem Aluminium-Druckgussbetrieb mit eigenem Schmelzbetrieb ab. Dadurch können Erkenntnisse über das reale System und dessen Verhalten gewonnen werden, ohne in das reale System eingreifen zu müssen. Das reale System selbst setzt sich aus einzelnen Prozessen zusammen. (s. Abbildung 1)

Das dem Betrieb angelieferte feste Aluminium (Masseln) wird mit Hilfe von Staplern zu den Schmelzöfen transportiert. Der Schmelzbetrieb wird von Staplern mit Aluminium in fester Form versorgt. In den Schmelzöfen wird das Aluminium geschmolzen und auf die für den Druckgussbetrieb benötigte Temperatur überhitzt. Neben festem Aluminium wird der Betrieb mit flüssigem Aluminium beliefert, welches

direkt nach dem Prozessschritt ‚Schmelzen‘ in den Materialfluss des Betriebes integriert wird. Bei Nachfrage vom Druckgussbetrieb wird aus den Schmelzöfen eine definierte Menge flüssiges Aluminium entnommen. Diese durchläuft den Prozessschritt ‚Impellern‘ und wird dann zum jeweiligen Dosierofen der Druckgussmaschine transportiert. Die Druckgussmaschine entnimmt zur Herstellung eines Druckgussteils eine bestimmte Menge Flüssigaluminium aus dem Dosierofen. Das fertige Druckgussteil wird zur nachgelagerten Endkontrolle transportiert. Die bei der Produktion des Druckgussteils entstehenden Abbruchreste, sowie Angüsse werden gesammelt und zurück zum Schmelzbetrieb transportiert, wo es zusammen mit neu ankommendem festem Aluminium erneut eingeschmolzen wird. Die nach der Endkontrolle als Ausschuss deklarierten Druckgussteile werden ebenfalls in den Schmelzbetrieb zurückgeführt und erneut eingeschmolzen. Nach der Endkontrolle folgt die Einlagerung bzw. Auslieferung der Druckgussteile.

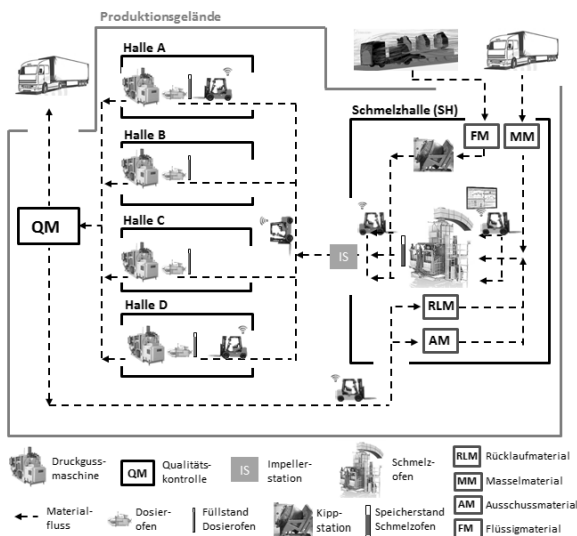


Abbildung 1. Schematische Darstellung des Schmelz- und Druckgussbetriebes

3 Aufbau der Simulation

Die Fertigungssimulation setzt sich aus den drei Teilen

- Materialflusssimulation
- Energieflusssimulation
- Kennzahlenberechnung

zusammen und bildet den Ist-Zustand im momentanen Betrieb ab.

Die Modellgrenze der Fertigungssimulation beginnt bei der Anlieferung des Rohmaterials Aluminium in fester oder flüssiger Form und endet bei der Einlagerung der fertigen Druckgussteile bzw. der Auslieferung der fertigen Druckgussteile (s. Abb.2).

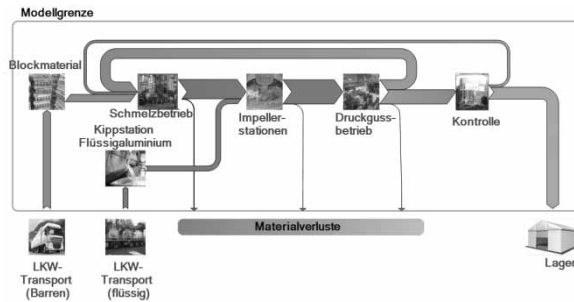


Abbildung 2. Materialfluss und Modellgrenze

Bei den Prozessen ‚Schmelzen‘, ‚Impellern‘ und ‚Druckgussbetrieb‘ kommt es prozessbedingt zu Materialverlusten, welche die Grenze des Modells verlassen und den Prozessen nicht mehr zur Verfügung stehen.

3.1 Materialflusssimulation

Die Materialflusssimulation im Schmelz- und Druckgussbetrieb bildet den Verlauf vom Rohmaterial bis zum fertigen Gussstück ab (Abb. 2). Diese hybride Simulation, die aus der Kombination von kontinuierlichen und ereignisgesteuerten Prozessen besteht, ist durch ihren modularen Aufbau in der Lage, verschiedene Betriebsgrößen abzubilden und einfache Szenarien mit veränderlichen Parametern zu simulieren. Das Modell ermöglicht zudem die Simulation einer Tages- oder Wochenproduktion. Durch den parametergesteuerten Aufbau können Prozessparameter der Anlage individuell variiert werden. Dies ermöglicht die Simulation einer realen Produktion mit schwankenden Schmelzleistungen, verschiedenen Takt- und Stillstandzeiten oder unterschiedlichen Transportwegen und Werkstückgrößen. Zur Kontrolle der Materialflusssimulation wird eine Massenbilanzierung durchgeführt:

$$M = M_{MV} + M_{PM} + M_{FT} + M_{zu} - M_{ab} = konst \quad (1)$$

Aus dem im Modell vorhandenen Materialvorrat (M_{MV}), der sich im Produktionsprozess befindlichen Masse (M_{PM}), der Masse der gefertigten Teile (M_{FT}) sowie der dem Modell zu- und abgeflossenen Massen (M_{zu} , M_{ab}) ergibt sich die konstant bleibende Gesamtmasse (M). (s. Abb.3)

Mit Hilfe der Materialflusssimulation lassen sich bereits Optimierungsansätze generieren. So lassen sich mit Hilfe der Analyse des Materialflusses Durchlaufzeiten, Dimensionierung, Personalbedarf und Störeinflüsse analysieren und daraus Optimierungsvorschläge entwickeln.

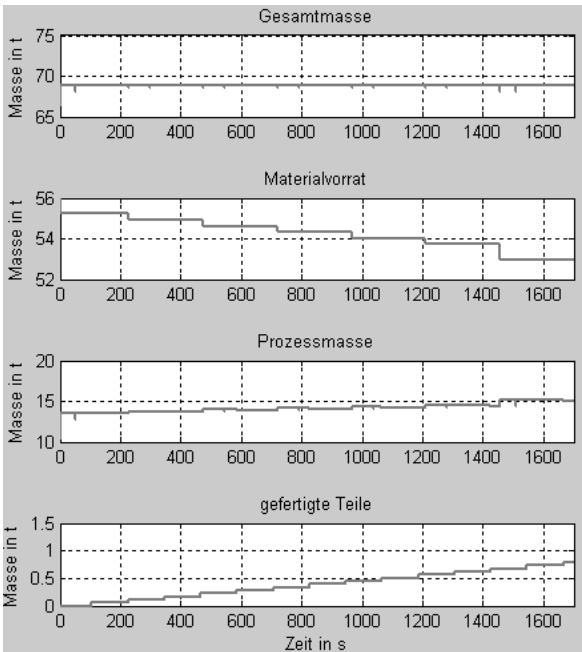


Abbildung 3. Massenbilanz der Materialflusssimulation

3.2 Energieflusssimulation

Um den Betrieb energetisch untersuchen zu können wird die Materialflusssimulation durch die Simulation des Energieflusses ergänzt. Im Betrieb treten die Energieformen chemische Energie (Gas), elektrische Energie (Strom) und Wärme auf (s. Tab. 1 u. Abb.4).

Energieform	Speicher	Verbraucher
chem. Energie	Gasnetz	Schmelzofen
el. Energie	Stromnetz	Druckgussmaschine
Wärme	Aluminium	Umgebung

Tabelle 1. Energieformen im Betrieb

Aufgrund der Korrelation zwischen Material- und Energiefluss kann der Energiefluss durch eine Erweiterung der Materialflusssimulation integriert werden.

Der Schwerpunkt der Energieoptimierung wird auf den Gasverbrauch und somit den Betrieb der Schmelzöfen gelegt. Die Herausforderung beim Gasverbraucher ‚Schachtschmelzofen‘ stellt die Ermittlung des Energieverbrauchsverhaltens dar. Vor allem

der dynamische Charakter des Energieverbrauches sowie seine Abhängigkeit von verschiedenen Größen (Beschickungsmaterial, Brennerführung, Füllstand,...) erhöhen die Komplexität bei der Ermittlung des Energieverbrauchs. Für das Ziel der Steigerung der Energieeffizienz und der damit verbundenen Anpassung der Prozessabläufe muss das Verbrauchsverhalten der Schmelzöfen ausreichend bekannt sein.

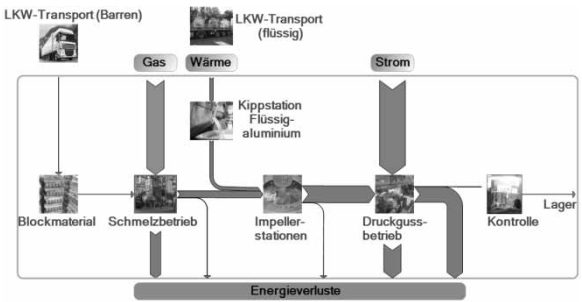


Abbildung 4: Energiefluss im Fertigungsbetrieb

In bereits vorausgegangenen Studien [2,3] wurden die Haupteinflussfaktoren auf die Energieeffizienz der Schmelzöfen identifiziert. Die prozessbedingte Temperaturschwankung im Abgasstrom hat den größten Einfluss auf die Energieeffizienz. Die Temperatur im Abgasstrom kann somit als leicht zu ermittelnde Größe herangezogen werden, um die Einflussfaktoren die sich direkt auf die Energieeffizienz auswirken, zu quantifizieren. Voraussetzung hierfür ist die Identifizierung der direkten Einflussfaktoren und deren Auswirkung auf den Schmelzprozess und dessen Gasverbrauch (s. Abbildung 8 und 9).

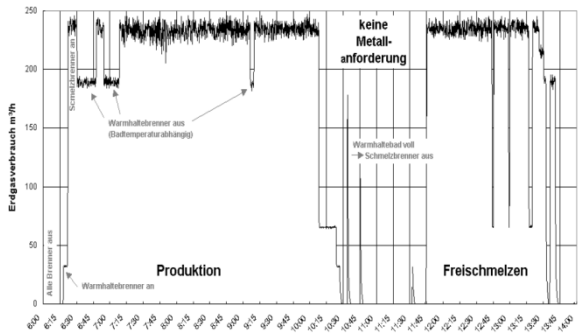


Abbildung 8. Analyse des Erdgasverbrauchs eines Schachtschmelzofens

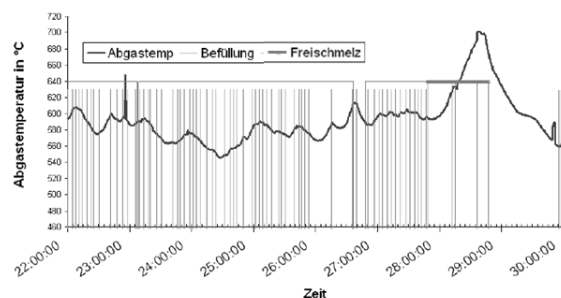


Abbildung 9. Verlauf der Ofenabgastemperatur und deren direkte Einflussfaktoren

Die direkten Einflussfaktoren werden selbst von anderen Faktoren beeinflusst, die somit indirekt Einfluss auf den Gasverbrauch und die Energieeffizienz nehmen (s. Abbildung 4).

Mit Hilfe von Messreihen der direkten Einflussfaktoren kann der Schmelzofen und dessen Energieverbrauch simulationstechnisch abgebildet werden.

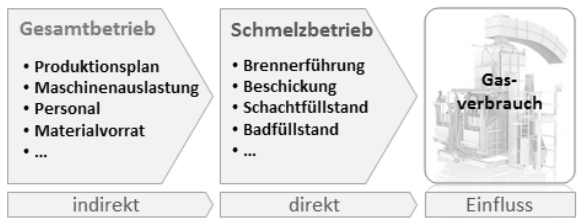


Abbildung 4. Einflussfaktoren auf den Gasverbrauch

Durch eine ganzheitliche Systemanalyse und der simulationstechnischen Optimierung von Prozessabläufen ist neben der Energieeinsparung im Schmelzbetrieb ebenfalls mit einer Steigerung der Energieeffizienz im Druckgussbetrieb zu rechnen. Dieser stellt den Hauptstromverbraucher im gesamten Betrieb dar (s. Abbildung 4), wird in der Simulation jedoch nicht detailliert betrachtet. Durch die gegenseitige Abhängigkeit von Schmelz- und Druckgussbetrieb entstehen bei der Optimierung der Prozessabläufe im Schmelzbetrieb positive Synergieeffekte im Druckgussbetrieb, die dessen Energieeffizienz positiv beeinflussen.

3.3 Kennzahlenberechnung

Für die Beurteilung des Ist-Standes sowie der Optimierungsmaßnahmen werden Kennzahlen herangezogen, deren zeitlicher Verlauf in der Simulation berechnet und ausgegeben wird.

Die Bestimmung der Effektivität als Kennzahl der einzelnen Komponenten gibt Auskunft, wie hoch das Optimierungspotential bei einzelnen Komponenten ist. Durch Nutzungsgrad, Leistungsgrad und der Qua-

litätsrate kann im Modell die Gesamtanlageneffektivität bestimmt werden. Diese Kennzahlen helfen bei der Beurteilung verschiedener Steuerungsstrategien und Prozessabläufen sowie der Prozesssicherheit und Prozessfähigkeit.

3.4 Simulation der vorhandenen Fertigungssteuerung

Ein zentraler Teil der Fertigungssimulation ist die Steuerung des Betriebes.

Aufgrund des hybriden Charakters der Produktion mit kontinuierlichen und ereignisgesteuerten Prozessen besteht die Steuerung aus sequenziellen Teilen und Zeitprogrammen.

Zeitprogramm	sequenziell gesteuert
Schmelzofen	
Freischmelzen	Beschicken
Ofenreinigung	Entnahme
	Brennersteuerung
Stapler	
Auftragsvergabe	
Druckgussmaschine	
Werkzeugwechsel	Werkzeugwechsel
Inspektion	
Gussteilproduktion	
Personal	
Pausen	
Personalwechsel	
Anlagenreinigung	

Tabelle 2. Sequenzielle Teile und Zeitprogramme der Steuerung

Durch den niedrigen Automatisierungsgrad im Betrieb, insbesondere im Schmelzbetrieb, ist die reale Steuerung vom wechselnden Schichtpersonal abhängig. Diese Bereiche der Produktionssteuerung im realen Betrieb wurden für die Simulation idealisiert (s. Abbildung 5).

Im realen Betrieb wird die Auslastung der Druckgussmaschinen durch den Produktionsplan festgelegt. Die Dosieröfen der Druckgussmaschinen, die eine bestimmte Menge Aluminium fassen und als Pufferspeicher dienen, geben bei Unterschreiten eines bestimmten Füllstandes ein Signal an den Schmelzbetrieb weiter, das dort für die Mitarbeiter visualisiert wird. Bei Vorhandensein von Aluminiumschmelze und Transportkapazität im Schmelzbetrieb, erfolgt die

Versorgung der jeweiligen Druckgussmaschinen. Die Versorgung der Schmelzöfen mit neuem festem Aluminium erfolgt „auf Sicht“. Durch visuelle Kontrolle von Schmelzschacht und Warmhaltebecken des Schmelzofens bestimmt das Personal den Beschickungszeitpunkt. Bei Erreichen des maximalen Füllstandes im Warmhaltebecken des Schmelzofens schaltet dieser ab, die Beschickung wird gestoppt.

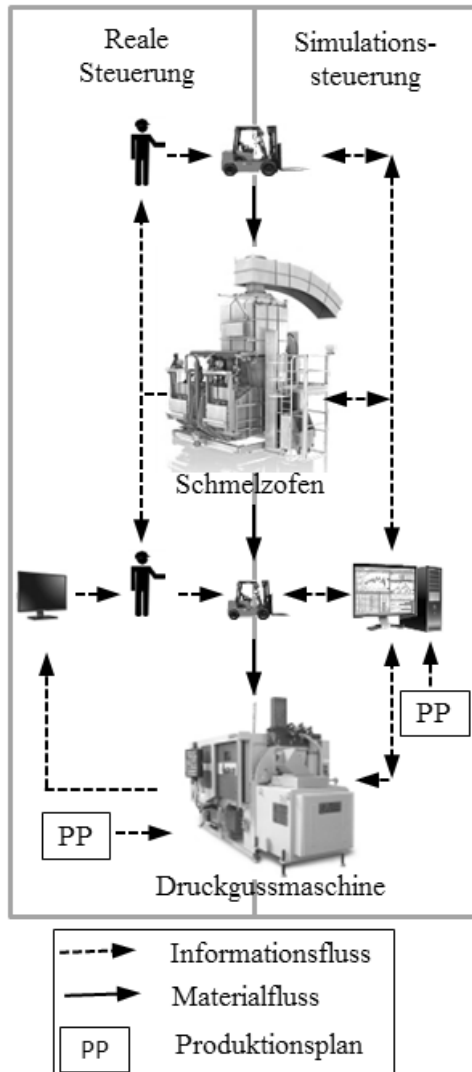


Abbildung 5. Vergleich von realer Steuerung mit Simulationssteuerung

Aus den im realen Betrieb vorhandenen Steuerungsmechanismen wurde eine Systemsteuerung entwickelt, die diese Mechanismen nachbildet und somit ermöglicht, den Ist-Zustand darzustellen. Die Informationen aus den einzelnen Komponenten laufen in der Steuerung zusammen. Diese ermittelt daraus zentral die durchzuführenden Handlungsschritte und

weist den Staplern nach vorher festgelegten Regeln Aufgaben zu.

4 Prozessdatenermittlung und Simulationsvalidierung

Die Entwicklung einer realitätsnahen Fertigungssimulation setzt eine umfangreiche Erfassung von Betriebsdaten voraus, die eine Validierung des Modells ermöglicht. Aufgrund der Betriebsgröße und der Diversität der Daten müssen diese sowohl automatisiert als auch manuell aufgenommen werden. Für die automatisierte Datenaufnahme von Produktionsdaten werden diese mittels SPS erfasst und zentral zusammengeführt und gespeichert. Hierfür wurden im realen Betrieb durch die Erneuerung des vorhandenen SPS-Systems die Voraussetzungen geschaffen. Die manuelle Datenaufnahme wurde dort eingesetzt, wo es um die Ermittlung unveränderlicher Werte ging, z.B. bei der Vermessung von Transportwegen oder Zeiten einzelner Prozessschritte wie dem Öffnen einer Ofentür.

Die Fertigungssimulation durchläuft einen iterativen Validierungsprozess (s. Abb.6). Hierfür werden in einer ersten ergebnisbezogenen Validierung Simulationsergebnisse mit Produktionsrichtwerten verglichen und die Simulation angepasst. So werden z.B. aus der Datenaufnahme die Werte von Soll- und Ist-Produktion verglichen und eine Verteilungsfunktion für die Ausfallzeiten ermittelt. Nach der Implementierung in die Simulation und der Auswertung der Simulationsergebnisse wird diese angepasst.

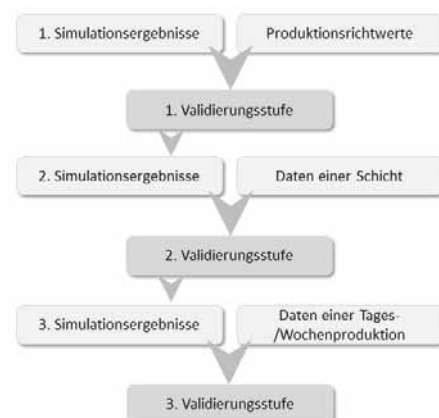


Abbildung 6. 3-stufige ergebnisbezogene Simulationsvalidierung

In den einzelnen Validierungsstufen wird der Simulationszeitraum ausgedehnt. Erst durch längere Simulationszeiträume (z.B. 1 Woche) können einzelne Pa-

parameter wie zum Beispiel Transportzeiten ausreichend genau angepasst werden.

Die Validierung wird mit Hilfe von zwei realen Fertigungsbetrieben mit unterschiedlicher Komponenten-zusammensetzung funktions- und ergebnisbezogen durchgeführt. Neben der Komponentenzusammensetzung unterscheiden sich die Betriebe in der verwendeten Anlagentechnik. Der parametrisierte Aufbau des Modells erlaubt eine zeiteffiziente Validierung.

5 Simulationsgestützte Analyse

Die Fertigungssimulation ermöglicht die Untersuchung von Fertigungssteuerungsstrategien, aus denen anschließend Anlagen- und Prozesskonzepte abgeleitet werden, die zu einer Steigerung der Prozesssicherheit in der Aluminiumversorgung der Druckgussmaschinen und einer größeren Energieeffizienz im Schmelzbetrieb führen. Dabei werden Konzepte zur Prozessoptimierung in der bestehenden Anlagenkonfiguration und Anlagenkonzepte zur Anpassung der Anlage an den Fertigungsprozess berücksichtigt.

Die Analyse und Optimierung des Betriebes mittels Fertigungssimulation orientiert sich an strategischen und operativen Fragestellungen in Schmelz- und Druckgussbetrieben.

Strategische Fragestellungen:

- Einflussfaktoren auf Schmelzprozess
- Speicherpotential
- Energieeffizienzpotential
- Prozesssicherheit

Operative Fragestellungen:

- Optimale Komponentenanzahl
- Rückführung von heißem Aluminium
- Notwendigkeit von Flüssigaluminium
- Optimale Komponentenauslegung

Bei den strategischen Fragestellungen handelt es sich um wissenschaftliche, noch nicht untersuchte Bereiche für gasbetriebene Aluminium-Druckgussbetriebe, bei deren Beantwortung allgemeingültige Erkenntnisse entstehen. Hier wird neues Wissen geschaffen, dass zur Steigerung der Wirtschaftlichkeit von Aluminium-Druckgussbetrieben beitragen kann.

Eine strategische Fragestellung, die mit der Fertigungssimulation bearbeitet werden kann, sind die

Ermittlung der indirekten Einflussfaktoren sowie der Einfluss der Betriebssteuerung auf den Schmelzprozess und dessen Leistung.

Es kann ermittelt werden, welche im Betrieb vorliegenden Steuermechanismen wie und mit welcher Intensität Einfluss auf den Betriebszustand des Schmelzofens nehmen. Des Weiteren besitzen Schmelzöfen einen Speicher für flüssiges Aluminium, dessen Potential als Nachfrage- und Angebotspuffer zwischen Druckguss- und Schmelzbetrieb noch nicht ermittelt wurde. Zudem bietet dieser aufgrund der hohen Wärmekapazität von Aluminium ein Energiespeicherpotential, dessen strategische Möglichkeiten noch nicht wissenschaftlich untersucht wurden. So kann durch verschiedene Brennerarten nicht nur der Energieträger Erdgas verwendet werden, sondern in Zeiten von Stromüberkapazität dieser zum Schmelzen oder Warmhalten von Aluminium herangezogen werden. Mit Hilfe der Fertigungssimulation kann die Wärmemenge, die bei verschiedenen Szenarien gespeichert werden kann, ermittelt und Strategien für eine Realisierung entwickelt werden.

Die Energieeffizienz eines Schmelzofens hängt neben den Eigenschaften des Ofens selbst von der Betriebsweise des Ofens ab, die wiederum durch die Betriebssteuerung, der Anzahl weiterer Komponenten und Größen (s. Abbildung 4) beeinflusst wird. Diese können mittels Fertigungssimulation identifiziert und quantifiziert werden. Über die erhobenen Daten lässt sich das Energieeffizienzpotential identifizieren (s. Abbildung 10). Dieses ergibt sich über die Differenz zwischen der über kurze Zeitabschnitte bereits erreichten Effizienz und der mittleren Energieeffizienz im Betrieb.

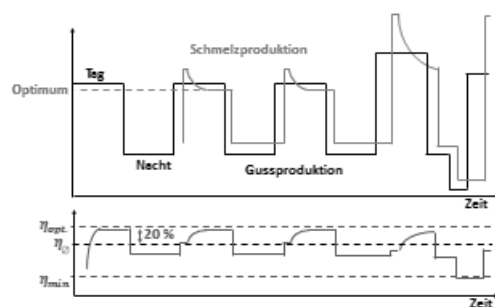


Abbildung 10. Abhängigkeit von Guss- und Schmelzproduktion und dessen Auswirkung auf den Schmelzofenwirkungsgrad

Bei den operativen Fragestellungen handelt es sich um speziell auf den zu untersuchenden Betrieb zuge-

schnittene Optimierungsmöglichkeiten. Durch die Analyse der Auslastung der einzelnen Komponenten kann deren Anzahl beispielsweise optimiert werden.

Eine weitere operative Fragestellung, die beantwortet werden kann, ist die Auswirkung der direkten Rückführung von heißem Rücklaufmaterial in den Schmelzöfen. Durch die direkte Rückführung ergeben sich Energieeinsparpotentiale, die ermittelt werden können. Gleichzeitig führt dies jedoch zu einer Veränderung der Beschickungspriorität der verschiedenen Aluminiumformen. Die anzupassenden Abläufe sowie die positiven und negativen Auswirkungen auf den gesamten Prozess können mit der Fertigungssimulation genau untersucht werden.

Das flüssig angelieferte Aluminium, ist kostenintensiver und muss aufgrund seines Abkühlverhaltens innerhalb weniger Stunden in den Produktionsprozess eingebracht werden. Mit Hilfe der Fertigungssimulation werden Strategien entwickelt, um den Bedarf an flüssigem Aluminium zu reduzieren und die Integration in den laufenden Prozess zu verbessern. Dadurch wird der Regelaufwand der Schmelzöfen reduziert und diese können in einem energetisch besseren Bereich betrieben werden. Durch den besseren Einblick in das Systemverhalten des Betriebes, kann die Auslegung der Komponenten wie beispielsweise der Schmelzöfen optimiert und der Einsatz von direkt geliefertem Flüssigaluminium weiter reduziert werden.

6 Modellprädiktive Prozessführung

Die hier entwickelte Fertigungssimulation stellt das Grundgerüst für die Entwicklung einer modellbasierten Prozessführung dar. Bei der modellbasierten Prozessführung berechnet ein in der Prozesssteuerung hinterlegtes Modell aus den Ist-Daten, welche der Steuerung aus dem Betrieb übermittelt werden, den zukünftig eintretenden Prozessablauf und ermittelt daraus die durchzuführenden Handlungen für eine verbesserte Betriebsführung.

Vor der Integration der modellbasierten Prozessführung ist diese mehrfach zu überprüfen und verschiedenen Stresstests auszusetzen. Die Entwicklung eines aktiv in den realen Betrieb eingreifenden Systems ist aufgrund der Ausfall- und Sicherheitsrisiken sehr aufwändig. Daher wird die Entwicklung eines Systems angestrebt, dass aus den vorhandenen Daten Handlungsanweisungen für das ausführende Personal ausgibt.

7 Zusammenfassung

Für einen Aluminium-Schmelz- und Druckgussbetrieb wurde eine Fertigungssimulation zur Steigerung der Produktionssicherheit und Energieeffizienz entwickelt. Durch die hohe Anzahl an individualisierten Komponenten und der Vielzahl an Einflussfaktoren in einem Druckgussbetrieb entsteht ein komplexes System, das ohne rechnergestütztes Simulationsmodell nur schwer zu analysieren ist.

Die Fertigungssimulation umfasst eine Materialflusssimulation sowie die Simulation des Energieflusses. Die mit Hilfe von zwei realen Betriebsstrukturen validierte Simulation kann aufgrund ihres modularen Aufbaus für verschiedene Betriebsgrößen angewendet werden. Sie beinhaltet unter anderem Produktionspläne und -strategien, Informationen über die Produktnachfrage und die Verfügbarkeit von Rohstoffen.

Mit Hilfe der Fertigungssimulation können sowohl strategische als auch operative Fragestellungen im Bereich von Aluminium-Schmelz- und Druckgussbetrieben beantwortet werden.

Die genannten strategischen und operativen Fragestellungen, die mit Hilfe der Fertigungssimulation beantwortet werden können, tragen zur Steigerung der Produktivität, Energieeffizienz und Ressourcenschonung bei und erhöhen somit die Wirtschaftlichkeit eines energieintensiven Industriezweigs.

8 References

- [1] M. Rabe, S. Spieckermann, S. Wenzel. *Verifikation und Validierung für die Simulation in Produktion und Logistik*. Springer Verlag, Deutschland, 2008.
- [2] W. Stephan, K. Zitzmann, G. Pröbstle, J. Kapischke. *Effiziente Energieverwendung in der Industrie – Teilprojekt „Metallschmelzbetriebe“*. Bayerisches Amt für Umweltschutz, Augsburg, 2005.
- [3] W. A. Mayer. *FORETA - Energieeffiziente Technologien und Anwendungen*. Verlag Attenkofer, Straubing, 2013.
- [4] A. Ringleb, S. Hirschberg, W. Schlüter, D. Schneider, G. Pröbstle, A. Dentel, W. Stephan. *Rationeller Energieeinsatz in der aluminiumverarbeitenden Industrie*. GIESSEREI 06/2013, Düsseldorf, 2013.

Konzeption einer aktiven Vorspur für ein Elektrofahrzeug

Xiaobo Liu-Henke^{1a}, Matthias Fritsch^{1b}, Florian Quantmeyer^{1c}

¹Hochschule Ostfalia

^a *x.liu-henke@ostfalia.de* ^b *matthias.fritsch@ostfalia.de* ^c *flo.quantmeyer@ostfalia.de*

Im vorliegenden Beitrag wird die Konzeption der aktiven Vorspur in einem modellbasiertem, ganzheitlichem Entwurfsprozess sowie das sich daraus ergebende Energiesparpotential dargestellt. Die aktive Vorspur ermöglicht fahrsituationsabhängig den Spurwinkel anzupassen und den Fahrwiderstand zu reduzieren. Darüber hinaus kann der Bremsweg durch optimale Ausnutzung des Kraftschlusspotentials des Reifens reduziert werden. Das Verringern des Fahrwiderstands ist besonders bei Elektrofahrzeugen interessant, da hierdurch die Reichweite zusätzlich gesteigert werden kann. Weiterhin wird eine Strategie kurz vorgestellt, womit das Potential der vorhandenen Aktoren optimal ausgenutzt wird.

1 Einleitung

Im Rahmen eines vom BMBF geförderten Forschungsvorhabens ECOCar (Energy Consumption Optimization for electrical Cars) wird die Funktionsintegration von fahrdynamischen Systemen und Elektromobilität für Elektrofahrzeuge mit zentralem elektrischem Antriebsstrang unter Anwendung eines ganzheitlichen Ansatzes fokussiert, um bei hoher Fahrsicherheit einen energieoptimalen Fahrbetrieb zu erzielen.

Die Verbindung von Fahrdynamikregelsystemen konventioneller Kraftfahrzeuge und Elektromobilität stellt eine Herausforderung dar. In heutigen Elektrofahrzeugen findet häufig lediglich eine „Elektrifizierung“ – ein Austausch des konventionellen Antriebsstrangs mit Verbrennungsmotor gegen den hybriden/elektrischen Antriebsstrang – statt. Dabei wird nicht berücksichtigt, dass der neue Antriebsstrang erhebliche Änderungen des Gesamtsystemverhaltens nach sich zieht, da sich die Charakteristik der elektrischen Maschine wesentlich von der des Verbrennungsmotors unterscheidet. Um ein optimales System- beziehungsweise Schwingungsverhalten zu erzielen, müssen innovative Fahrdynamikregel- und Fahrerassistenzsysteme neu- beziehungsweise weiterentwickelt werden. Abbildung 1 illustriert das Konzept mit beispielhaften aktiven Fahrdynamiksystemen, womit einerseits die Fahrdynamik von Elektrofahrzeugen un-

ter Berücksichtigung des veränderten Schwingungsverhaltens verbessert und andererseits die funktionalen und energetischen Potentiale ausgeschöpft werden können. Dies soll bei gewünschter Fahrdynamik einen energieoptimalen Fahrbetrieb realisieren. Zusätzlich wird untersucht, inwiefern das Potential der bestehenden Systeme des Fahrzeugs ausgenutzt werden.

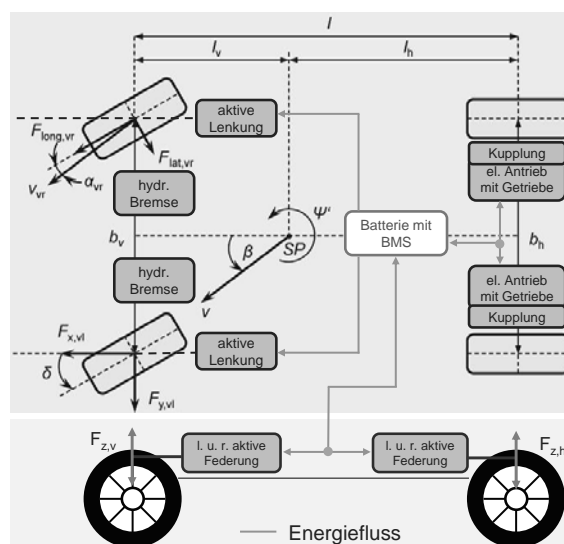


Abbildung 1: Mechatronische Entwicklung einer aktiven Vorspur für ein Elektrofahrzeug

2 Entwicklungsmethodik

Die aktive Vorspur soll auf bestehenden Lenksystemen entwickelt werden. Dies beinhaltet eine vorhandene Mechanik (Tragstruktur, Radaufhängung), Aktorik zur Ansteuerung (Elektromotor, Leistungselektronik), Sensorik zum Erfassen der Zustandsgrößen sowie die Informationsverarbeitung durch ein Steuergerät (z.B. Electric Power Steering (EPS) oder Active Front Steering (AFS)). Um die Systemkomplexität bei der Entwicklung eines solchen interdisziplinären Systems zu beherrschen, wird ein modellbasierter Entwicklungsprozess verwendet. Hierzu wird das Gesamtsystemverhalten auf Basis physikalischer Effekte durch mathematische Gleichungen beschrieben. Basierend auf den Modellen werden die Regelalgorithmen modellbasiert ausgelegt und in einem durchgängigen Prozess, MiL/SiL/HiL, validiert und optimiert. Somit findet die Erprobung bereits in einer frühen Phase der Entwicklung an Prototypen (u.A. Modellen) statt und die Funktion kann frühzeitig abgesichert werden.

3 Modellbildung

Zur Repräsentation des Systemverhaltens wird zuerst die Modellbildung durchgeführt. Nachstehend wird das Zweispurmodell zur Untersuchung des Einflusses der aktiven Vorspur dargestellt.

3.1 Vereinfachtes Zweispurmodell

Zur Analyse des Systemverhaltens und anschließender Konzeption wird ein vereinfachtes Zweispurmodell entwickelt. Dieses basiert auf dem in Abbildung 2 gezeigten Versuchsträger, dem Mechatronic-Mobile.

Das M-Mobile wurde wie ein reales Fahrzeug im Maßstab 1:3 entwickelt und verfügt über vier radindividuelle Module mit Einzelradantrieb und -lenkung. Hierdurch können alle Reifenkräfte für eine gewünschte, kontrollierte Horizontalbewegung individuell eingestellt werden. Durch sinnvolle Reduktion entsprechend der Anforderungen wird ein physikalisches Modell (siehe Abbildung 3) hergeleitet. Hierbei wird die Aufbaumasse konzentriert im Schwerpunkt und auf Höhe der Straße angenommen, weshalb we-

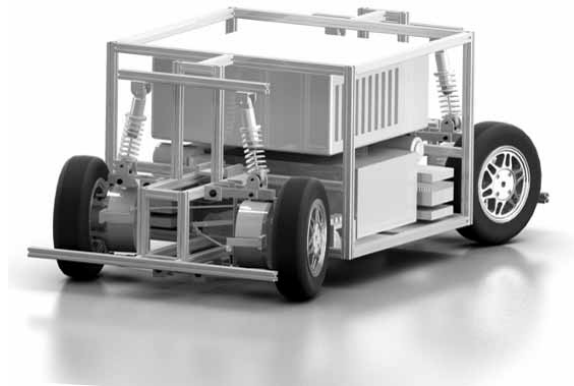


Abbildung 2: Versuchsträger M-Mobile [1]

der Nick- noch Wankbewegungen möglich sind. Der Massenmittelpunkt des Aufbaus stellt gleichzeitig das körperfeste Koordinatensystem (Veh) dar. Die Reifen werden jeweils durch eigene lokale Koordinatensysteme beschrieben. Mittels eines ortsfesten, inertial Koordinatensystems (ICS) wird die Trajektorie des Fahrzeugs berechnet. Das erstellte Zweispurmodell besteht aus 5 Massen mit insgesamt 11 Freiheitsgraden:

- Koordinaten des Schwerpunktes der Aufbaumasse X_{Veh}, Y_{Veh}
- Kardanwinkel des Fahrzeugaufbaus ψ
- Rotation der Räder um deren Drehachse
- Drehbewegung der Räder bezüglich des BCS

Basierend auf dem physikalischen Modell in Abbildung 3 unter Berücksichtigung des Einflusses der Kräfte/Momente und des kinematischen Zusammenhangs, wird das Mathematische Modell hergeleitet. (vgl. [2]).

Translatorische (Längs- und Quer-) Bewegungsgleichung:

$$\sum_k \underline{F}_k = m \cdot \underline{a} \quad (1)$$

Rotatorische (Gier-) Bewegung:

$$\sum_k r_k \cdot \underline{F}_k = J_z \cdot \underline{\ddot{\psi}} \quad (2)$$

Als Eingangsgrößen für dieses Modell werden die Lenkwinkel an den Rädern sowie die Kräfte durch die

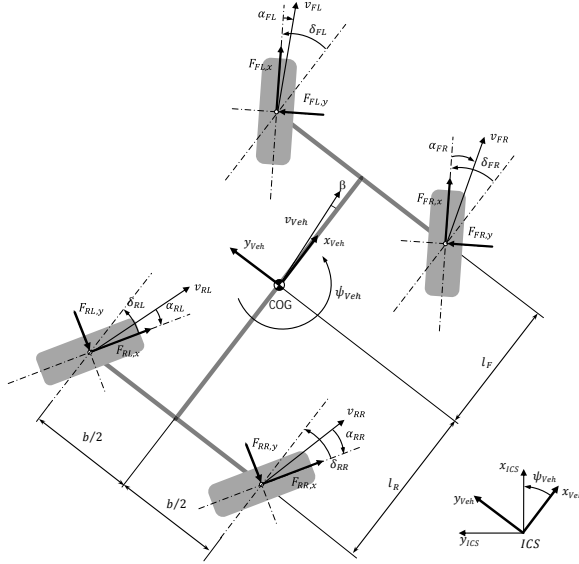


Abbildung 3: Zweispurmodell mit vier radindividuellen Lenkmodulen

Antriebsmotoren verwendet. Änderungen des Lenkwinkels resultieren aufgrund der physikalischen Gegebenheit in einer Gierbewegung des Fahrzeugs sowie dem Aufbau von Reifenseitenkräften F_{Qij} . Die Längskräfte F_{Lij} des Reifens werden durch den Betrag des Antriebsmoments beeinflusst. Die Reifenkräfte werden anschließend in das fahrzeugfeste Koordinatensystem transformiert.

$$\begin{bmatrix} F_{xij} \\ F_{yij} \end{bmatrix} = \begin{bmatrix} \cos \delta & -\sin \delta \\ \sin \delta & \cos \delta \end{bmatrix} \cdot \begin{bmatrix} F_{Lij} \\ F_{Qij} \end{bmatrix} \quad (3)$$

Dieses Modell bietet die Grundlage das Fahrzeugverhalten hinsichtlich der Stabilität (u.A. Schwimmwinkel β) zu bewerten sowie die zurückgelegte Trajektorie (z.B. Beeinflussung des Bremswegs) zu berechnen.

3.2 Mathematische Beschreibung des Reifens

Der Modellierung des Kontakts zwischen dem Reifen und der Straße nimmt bei der Simulation von Kraftfahrzeugen eine Schlüsselrolle ein. Der Reifen überträgt als Bindeglied zwischen Fahrbahn und Fahrzeug alle Kräfte und Momente und das Übertragungsver-

halten geht maßgeblich in das Fahrverhalten, die Sicherheit und den Komfort des Gesamtfahrzeugs ein [3]. Abhängig von der Simulationsaufgabe können einfache empirische Modelle ausreichend sein oder komplexe FEM-Modelle des Reifens benötigt werden. Im vorliegenden Fall wird im ersten Schritt lediglich das Verhalten des Reifens unter Vorspur bei Geradeausfahrt untersucht. Hierfür wird ein Magic-Formula-Modell für den Reifen eingesetzt. Die maximal übertragbare Kraft des Reifens F_{max} hängt stark vom Kraftschlusskoeffizienten μ zwischen Reifen und Fahrbahn sowie der Radaufstandskraft ab. Das Übertragungspotential nimmt bei steigender Radaufstandskraft F_z und Überschreiten der konstruktiven Radlast (F_{z0}) schwach degressiv zu [4]. Dieses Verhalten wird durch den Degressionsparameter e_z berücksichtigt (Gl. (5)).

$$F_{max} = \mu \cdot F_{z,eff} \quad (4)$$

Mit:

$$F_{z,eff} = F_z \cdot \left(1 - e_z \cdot \left(\frac{F_z}{F_{z0}} \right)^2 \right) \quad (5)$$

Die geometrische Addition der Komponenten der Reifenkraft in Längs- und Querrichtung des Fahrzeugs darf die maximal übertragbare Kraft nicht überschreiten. Andernfalls resultieren instabile Reifen- und Fahrzustände.

$$\sqrt{F_x^2 + F_y^2} \leq \mu \cdot F_{z,eff} \quad (6)$$

Kräfte werden durch Relativbewegungen innerhalb des Reifenlatsches (Schlupf) und den damit einhergehenden Verformungen der Reifenpartikel übertragen. In Abbildung 4 wird das Übertragungspotential des Reifens bei geändertem Längsschlupf gezeigt. Das Übertragungsverhalten der Reifenkräfte über dem Schlupf aufgetragen ist stark nicht linear. Weiterhin sollten die Schlupfwerte unterhalb des gesättigten Reifenverhaltens liegen, da die insgesamt übertragbaren Kräfte sonst sinken.

Der Längsschlupf lässt sich über das Verhältnis des Produkts aus dem dynamischen Reifenhaltmesser r_{dyn} mit der Radwinkelgeschwindigkeit w_{Rad} zu der Radgeschwindigkeit v_{Rad} berechnen:

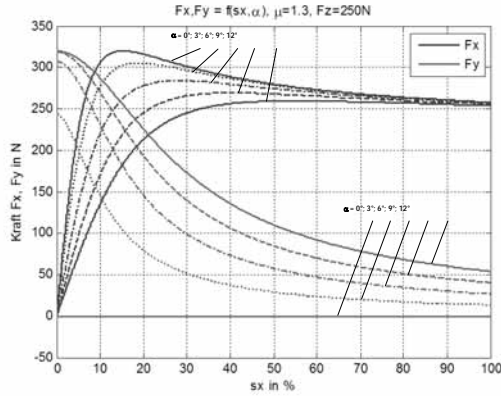


Abbildung 4: Übertragungsverhalten des Reifens bei variiertem Längsschlupf

$$s_x = \frac{w_{Rad} \cdot r_{dyn} - v_{Rad}}{\max(w_{Rad} \cdot r_{dyn}, v_{Rad})} \quad (7)$$

Der Querschlupf bzw. Schräglaufwinkel des Reifens ergibt sich aus dem Verhältnis der Quer- zur Längsgeschwindigkeit des Fahrzeugs:

$$\tan(\alpha) = \frac{v_y}{v_x} \quad (8)$$

Der Querschlupf kann über die in Abbildung 3 gezeigten kinematischen Beziehungen und auftretenden Geschwindigkeiten wie folgt formuliert werden:

$$\alpha_{VL} = \delta_{VL} - \arctan\left(\frac{v \cdot \sin \beta + \dot{\psi} \cdot l_V}{v \cdot \cos \beta - \dot{\psi} \cdot \frac{s}{2}}\right) \quad (9)$$

$$\alpha_{VR} = \delta_{VR} - \arctan\left(\frac{v \cdot \sin \beta + \dot{\psi} \cdot l_V}{v \cdot \cos \beta + \dot{\psi} \cdot \frac{s}{2}}\right) \quad (10)$$

$$\alpha_{HL} = \delta_{HL} - \arctan\left(\frac{v \cdot \sin \beta - \dot{\psi} \cdot l_H}{v \cdot \cos \beta - \dot{\psi} \cdot \frac{s}{2}}\right) \quad (11)$$

$$\alpha_{HR} = \delta_{HR} - \arctan\left(\frac{v \cdot \sin \beta - \dot{\psi} \cdot l_H}{v \cdot \cos \beta + \dot{\psi} \cdot \frac{s}{2}}\right) \quad (12)$$

Bei Betrachtung der reinen Längsdynamik des Fahrzeugs sind die Giergeschwindigkeit $\dot{\psi}$ und der Schwimmwinkel β aufgrund der Systemkinematik gleich null, wodurch der Schräglaufwinkel lediglich

vom eingestellten Lenkwinkel abhängt. Die Genauigkeit wird durch die Vernachlässigung des Reifennachlaufs nach [2] nicht wesentlich verringert, erlaubt jedoch eine übersichtlichere Darstellung der Zusammenhänge.

4 Konzeption der aktiven Vorspur

Nachfolgend wird das Konzept der aktiven Vorspur mit den entwickelten Betriebsstrategien vorgestellt.

4.1 Grundprinzip der aktiven Vorspur

Abbildung 5 stellt das Prinzip der aktiven Vorspur vor. Die Schrägstellung der Räder bewirkt den Aufbau eines Schräglaufwinkels δ sowie einer Reifenseitenkraft F_{quer} . Diese wirkt über den Sinusanteil des Lenkwinkels als Längskraft auf das Fahrzeug und trägt zu dessen Verzögerung bei.

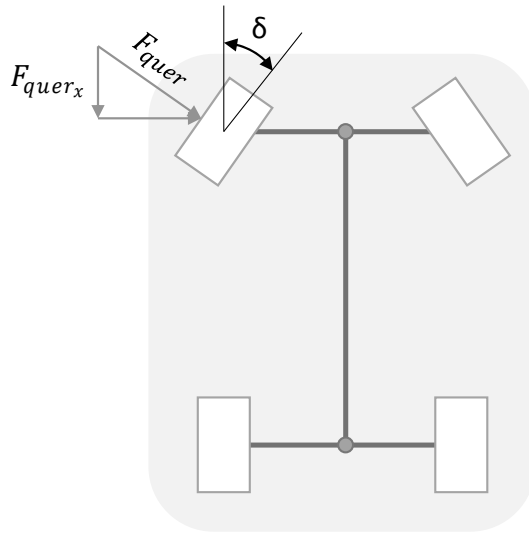


Abbildung 5: Aktive Vorspur zur Beeinflussung der Längsdynamik

Mit Unterstützung der Regelstrategie und Aktoren wird die gewünschte Schrägstellung der Räder eingestellt. Abbildung 6 skizziert die Regelungsstrategie

der aktiven Vorspur. Übergeordnet wird zwischen den möglichen Betriebsstrategien (siehe Kapitel 4.2) entschieden. Bei Bedarf wird der benötigte Vorspurwinkel überlagert und an den unterlagerten, lokalen Lenkwinkelregler als Sollwert gegeben. Als Aktuatoren für die Lenkungsmodule werden ebenfalls Elektromotoren eingesetzt.

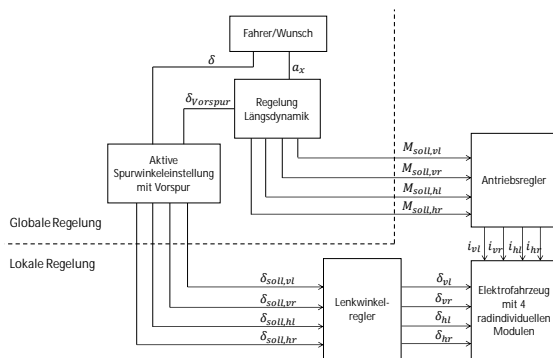


Abbildung 6: Übersicht Regelungsstrategie aktive Vorspur

Diese mechatronische Funktion ermöglicht die Verbesserung des Fahrverhaltens bei gleichzeitiger Reduzierung des Energieverbrauchs.

4.2 Strategie zum energieoptimalen Bremsen durch Aktorkooperation

Die zusätzliche Kraft F_{quer} bewirkt eine Erhöhung der Bremskraft, wodurch sich der Bremsweg verglichen zum konventionellen Bremsvorgang deutlich verkürzt. In Abbildung 7 wird die mögliche Erhöhung der Verzögerung prinzipiell aufgezeigt. Der Lenkwinkel verdreht das Koordinatensystem des Reifens zu dem des Fahrzeugs, wodurch die Reifenseitenkraft anteilig eine Komponente in Fahrzeuggängsrichtung erhält.

Nachdem das Potential der aktiven Vorspur gezeigt werden konnte, werden nachstehend Strategien zur Kooperation mit den bestehenden Aktoren am Beispiel des verwendeten Versuchsträgers vorgestellt.

- **Rekuprieren mit der E-Maschine:**
Einsatz sinnvoll bei kleinen Verzögerungen sowie hohen Drehzahlen des Motors, sodass elek-

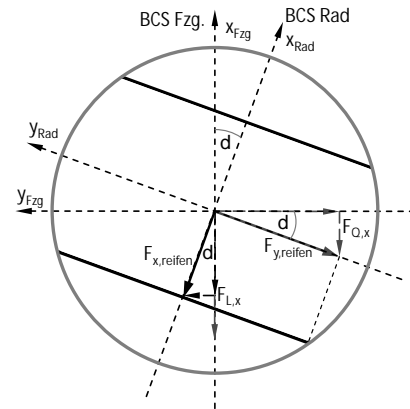


Abbildung 7: Kammscher Kreis beeinflusst durch den Lenkwinkel

trische aus der kinetischen Energie zurückgewonnen werden kann.

- **Gegenstrombremsen mit der E-Maschine:**
Verzögern mit dem Elektromotor bei niedrigen Geschwindigkeiten, wenn nicht mehr rekupiert werden kann.
- **Bremsen mit Vorspur:**
Einsatz bei sehr großen Verzögerungen, wenn die E-Maschine kein ausreichend großes Moment erzeugen kann.
- **Bremsen mit Reibbremse:**
Ebenfalls bei großen Verzögerungen.

Prinzipiell sollte die Verzögerung des Fahrzeugs bei hohen Geschwindigkeiten durch Rekuperation unter Verwendung der induzierten Spannung realisiert werden. Hierdurch wird nahezu verschleißfrei verzögert und gleichzeitig Energie zurückgewonnen. Der entwickelte Konzeptentwurf sieht vor, dass bei kleineren Verzögerungen ein geringer Vorspurwinkel ($< 3^\circ$) eingestellt wird. Ist eine größere Verzögerung notwendig, bleibt dieser Winkel vorhanden und es wird ein Bremsmoment über die E-Maschine erzeugt. Somit lassen sich größere Bremsmomente bei gleichzeitig geringem Reifenverschleiß erzeugen. Die letzte Stufe sieht den Einsatz des größten Bremsmoments der E-Maschine verbunden mit größeren Vorspurwinkeln vor. Diese wird lediglich im Notfall eingesetzt, weil hiermit ein großer Reifenverschleiß einhergeht.

Wenn der Energieverbrauch zum Einstellen der aktiven Vorspur ist geringer als bei ABS-Bremsvorgängen, kann sie bei kleinen Verzögerungen eingesetzt werden, um energieoptimal fahren zu können.

5 Systemanalyse mittels MiL

Nachstehend die Strategie theoretisch ausgelegt wurde, wird die Analyse des Systemverhaltens mittels MiL durchgeführt, um das Konzept und die Strategie zu validieren. Hierfür werden die Einflüsse auf die Längsdynamik, den Fahrwiderstand und die Stabilisierung fokussiert.

5.1 Energieverbrauch der Aktoren bei Verzögerung

Die Abbildungen 8 und 9 stellen die Simulationsergebnisse und die Messungen am realen Fahrzeug gegenüber. Sie zeigen auf, dass der Einsatz der aktiven Vorspur nicht nur den Bremsweg signifikant reduziert, sondern sogar die benötigte Leistung deutlich verringert. Gegenübergestellt werden die Bremsvorgänge beginnend bei $v = 2 \frac{m}{s}$ bis zum Stillstand des Funktionsträgers. Aufgeführt werden im oberen Teil der Abbildungen die Ströme die zum Betreiben der jeweiligen Aktoren benötigt werden und im unteren Bereich die Geschwindigkeit des realen Fahrzeugs. Der Einsatz der Elektromotoren zum Abbremsen erfordert den Aufbau eines bestimmten Stroms. Dieser kann, wie Gl. (13) zeigt, sowohl aus der Batteriespannung (Gegenstrombremsen) als auch aus der induzierten Spannung (kinetische Energie des Fahrzeugs) aufgebaut werden.

$$U_A = R \cdot i + L \cdot \frac{di}{dt} + c \cdot \omega \quad (13)$$

Bei der vorliegenden Betrachtung ist die induzierte Spannung durch die Drehbewegung der Motoren gering, sodass ein Großteil der Spannung durch die Batterie gestellt werden muss. Beim Einsatz der aktiven Vorspur wurden die Antriebsmotoren freigeschaltet, sodass diese die Längsdynamik nicht beeinflussen konnten.

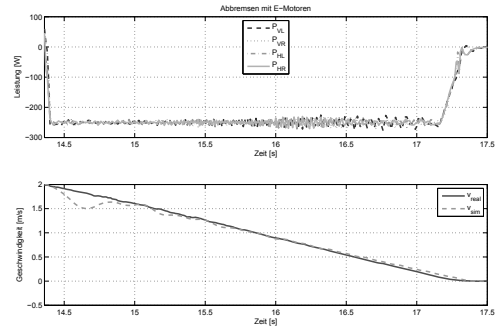


Abbildung 8: Abbremsen mit E-Motor

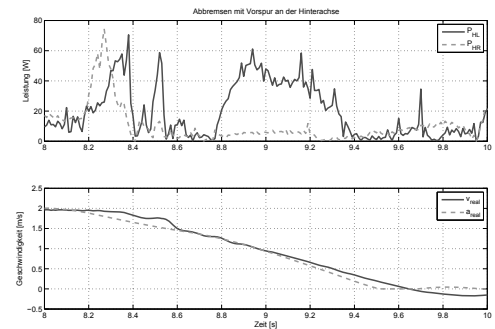


Abbildung 9: Abbremsen mit $\delta_{Vorspur} = 15^\circ$

Anschließend wurden die Energieverbräuche durch Integration der Leistungen der verwendeten Aktoren berechnet. Beim Einsatz der Motoren zum Abbremsen des Fahrzeugs erzeugt ein konstant hoher Strom das Bremsmoment, wohingegen bei der Vorspur lediglich am Anfang der Lenkwinkel eingestellt wird und das Fahrzeug anschließend bremst. Kumuliert ergeben sich folgende Energieverbräuche:

$$W_{Mot} = 0,8 \text{ Wh} \quad (14)$$

$$W_{Vorspur} = 0,0165 \text{ Wh} \quad (15)$$

Das alleinige Abbremsen des Fahrzeugs mit den Motoren ist somit energetisch nicht sinnvoll, da sonst die komplette Bewegungsenergie durch das Aufwenden elektrischer Energie gewandelt wird.

5.2 Einfluss auf den Fahrwiderstand

Als weiterer Gesichtspunkt wird der Fahrwiderstand F_W während der Geradeausfahrt mit und ohne Vorspur untersucht. Der Fahrwiderstand setzt sich wie in Gl. (16) gezeigt aus dem Rollwiderstand F_R , dem Luftwiderstand F_L , dem Steigungswiderstand F_{St} , dem Beschleunigungswiderstand F_B sowie dem Vorspurwiderstand $F_{Vorspur}$ zusammen. Hierzu wird der Versuchsträger auf eine Geschwindigkeit $v = 2 \frac{m}{s}$ beschleunigt und anschließend auf einer ebenen Fläche die Energie zum halten der Geschwindigkeit gemessen. Die Zeitdauer für den nachstehenden Energieverbrauch beträgt $t = 2 s$.

$$F_W = F_R + F_L + F_{St} + F_B + F_{Vorspur} \quad (16)$$

$$W_{ohne\ Vorspur} = 0,0045\ Wh \quad (17)$$

$$W_{mit\ Vorspur} = 0,0059\ Wh \quad (18)$$

Es besteht ein Unterschied zwischen den beiden Versuchsfahrten. Das Energieeinsparpotential erscheint sehr groß. Innerhalb einer Simulation mit einem Vorspurwinkel von $\delta_{vorspur} = 10^\circ$ konnten die Fahrwiderstände und somit der Energieverbrauch bei stationärer Fahrt um 6 % reduziert werden.

5.3 Stabilisierung μ -Split-Bremsung

Durch die Aufteilung der Reifenseitenkraft in eine Längs- und eine Querkomponente, die eine gegenseitige Drehrichtung um die Z-Achse bewirken, entsteht ein stabilisierendes Moment beim μ -Split-Bremsen.

Die Momentenbilanz bei Berücksichtigung der μ_{high} -Seite liefert für den erforderlichen Winkel folgende Gleichung (vgl. Abbildung 3):

$$\frac{l_v}{b/2} = \tan(\delta) \quad (19)$$

Hieraus folgen große erforderliche Winkel, wenn l_v größer als $b/2$. Aufgrund des großen, notwendigen Winkels ist die Stabilisierung bei herkömmlichen Fahrzeugen nicht einsetzbar.

6 Resümee und Ausblick

In diesem Beitrag konnte gezeigt werden, dass der Einsatz einer aktiven Vorspur sowohl zur energieoptimalen Fahrt als auch zur Verbesserung der Fahrdynamik beitragen kann. Der Einsatz ist im Bereich niedriger Geschwindigkeiten bzw. Drehzahlen des Elektromotors anwendbar, da dort die Möglichkeit zur Rekuperation nicht gegeben ist. Demgegenüber muss der höhere Reifenverschleiß beim Einsatz der aktiven Vorspur untersucht und bewertet werden. Zusätzlich soll der Energieverbrauch einer herkömmlichen Reibbremse mit der aktiven Vorspur verglichen werden. Hieraus wird abgeschätzt, welches der beiden Systeme sich energetisch sinnvoller einsetzen lässt. Weiterhin sollte der Einfluss der Elastokinematik der Rad-aufhängung auf die Spur und den Sturz berücksichtigt werden, sodass diese mit simuliert wird und zu einer genaueren Abbildung des Gesamtsystems führt.

7 Danksagung

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung, und Forschung unter dem Förderkennzeichen 03FH035I3 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren. Für die Förderung bedanken sich die Autoren herzlichst.

Literatur

- [1] R. Buchta und X. Liu-Henke. *Analysis of active Toe-in for Vehicle Longitudinal Dynamics*. Mechatronics-REM, in Paris, S. 371-376, 2012
- [2] Anne von Vietinghoff. *Nichtlineare Regelung von Kraftfahrzeugen in querdynamisch kritischen Situationen*. Dissertation, Karlsruhe, Deutschland, 2008
- [3] Hans-Hermann Braess. *Vieweg Handbuch Kraftfahrzeugtechnik*. Springer Vieweg, Deutschland, 2013
- [4] D. Schramm, M. Hiller und R. Bardini. *Modellbildung und Simulation der Dynamik von Kraftfahrzeugen*. Springer, Deutschland, 2010

Integration eines DP-Moduls in maritime Aus- und Weiterbildungssimulatoren

B. Kutschera¹, M. Markert¹, J. Ladisch², B. Lampe³

¹Hochschule Wismar, Bereich Seefahrt, Richard-Wagner-Str. 31, 18119 Warnemünde

²Fachhochschule Stralsund, Fachbereich Maschinenbau, Zur Schwedenschanze 15, 18435 Stralsund

³Universität Rostock, Institut für Automatisierungstechnik, Richard-Wagner-Str. 31, 18119 Rostock

bettina.kutschera@hs-wismar.de

Aufgrund von Neuerungen in den internationalen Standards für die Ausbildung von seefahrendem Personal muss für den Einsatz auf Schiffen mit Dynamischen Positionierungssystemen (DP-Systemen) ein spezielles Zertifikat erworben werden. Einsatzgebiete dafür ist u. a. der stark zunehmende Sektor der offshore Windenergie in Form von Errichter- oder Instandhaltungsschiffen, aber auch die offshore Öl- und Gasförderung, sowie zunehmend auch Kreuzfahrt- oder Fährschiffe. Für diese DP-Schiffe muss in einem zertifizierten Schiffssimulator eine Aus- bzw. Weiterbildung absolviert werden, wofür ein generisches Modul entwickelt wurde, das alle nötigen Funktionalitäten bietet und für den Erwerb dieses DP-Zertifikats eingesetzt werden kann. Nicht nur die Anforderungen an die Konsole selbst, sondern auch die Anbindung an das existierende Simulationssystem werden erläutert.

1 Simulatoren in der Ausbildung

An der Hochschule Wismar, Bereich Seefahrt in Warnemünde wird seefahrendes Personal mit modernster Simulationstechnik aus- und weitergebildet. Die nach internationalen Anforderungen gestalteten Studiengänge Nautik, Schiffsbetriebstechnik und Schiffselektrotechnik qualifizieren zum Einsatz als Offizier an Bord von Seeschiffen als Wachoffizier bzw. Ingenieur. Dafür steht am Bereich Seefahrt das Maritime Simulationszentrum Warnemünde (MSCW) zur Verfügung, das einen ANS 5000 Schiffsführungs- und SES 7 Maschinensimulator der Firma Rheinmetall Defence Electronics beinhaltet und durch kontinuierliche Updates auf dem neuesten Stand der Technik gehalten wird.

Am MSCW soll ein zusätzliches Ausbildungsmodul für Dynamische Positionierung (DP) angeboten werden können, um für die zukünftigen nautischen und technischen Offiziere ein weiteres Einsatzspektrum im Offshore-Bereich zu ermöglichen. Dafür muss entsprechend neuer Standards [1] ein spezielles DP-Zertifikat erworben werden, das nur in zertifizierten Schiffssimulatoren angeboten werden darf. Deswegen soll ein generisches Zusatzmodul entwickelt werden, das über eine Schnittstelle an den Simulator angeschlossen werden kann und alle Anforderungen für die Ausbildung erfüllt.

2 Funktionsanforderungen für DP-Ausbildung und deren Umsetzung

Die Zertifizierung des Simulators erfolgt durch Klassifikationsgesellschaften und würde am MSCW durch Det Norske Veritas durchgeführt. In [2] sind die Anforderungen detailliert aufgeführt und wurden in einer generischen Konsole umgesetzt.



Abbildung 1. Generische DP-Konsole

Die Ausbildungsszenarien müssen für unterschiedliche Schiffsmodelle möglich sein und die Konsole muss sich somit bspw. in den Buttons für den Notstopp automatisch an die Antriebskonstellation anpassen, was über ein Touchpanel umgesetzt wurde. Der Joystick stellt die wichtigste Steuermöglichkeit im DP-Modus dar und gibt die Bewegungsvektoren in drei Freiheitsgrade vor: in x-Richtung entlang der Schiffslängsachse, in y-Richtung als Querbewegung

und um die z-Achse als Rotationsbewegung des Bugs. Über Taster können Bewegungen durch einen Regler gesteuert werden und z. B. Längs- und Quergeschwindigkeit trotz Umwelteinflüssen (wie Wind und Strömung) zu null geregelt werden und über den Joystick lediglich eine Schiffsrotation vorgegeben werden.

Diese Konsole wird über eine entwickelte Schnittstelle an den Schiffsführungssimulator im MSCW angeschlossen. Der Austausch aller Ist- und Soll-Bewegungsgrößen des Schiffes und der Antriebe sowie aller Betriebsart- und Steuermodi-Informationen wird darüber realisiert. Zusätzlich wurde hier vorgesehen, dass eine Datenbank die Werte aufzeichnet um ein Replay von Übungsszenarien zu ermöglichen, was in der Ausbildung sehr wichtig ist und sowohl ein reines Abspielen der Aufzeichnung als auch ein Neustart von einem beliebigen Zeitpunkt ermöglicht.

Dies wird in der Konsole mittels zwei Industrie-PCs realisiert, wobei der Erste mit einer Festplatte für die großen Datenmengen des Replays ausgestattet ist und der Zweite durch einen Flashspeicher die schnellere interne Berechnung des Reglers ermöglicht. Für die interne Kommunikation zwischen den Industrie-PCs wurde ein Modbus-RTU Protokoll entwickelt und die externe Verbindung zum Schiffssimulator erfolgt über NMEA 183 bzw. über Modbus-TCP Protokolle.

3 Ergebnisse und Diskussion

Bevor die Schnittstelle zum MSCW entwickelt und validiert ist, erfolgte der Aufbau einer Testumgebung in MATLAB/SIMULINK[®]. Hier wurde ein offenes physikalisches Bewegungsmodell eines Schiffes in drei Freiheitsgraden umgesetzt. Über einen Datenbankfile wird das Modell parametrisiert und kann beliebige Schiffstypen nachbilden. Hinsichtlich der Antriebskonstellation können bis zu vier Hauptaggregate als Fest- oder Verstellpropeller mit Ruder, POD-Antriebe oder Voith-Schneider-Propeller umgesetzt sowie sechs Querstrahler vorgesehen werden.

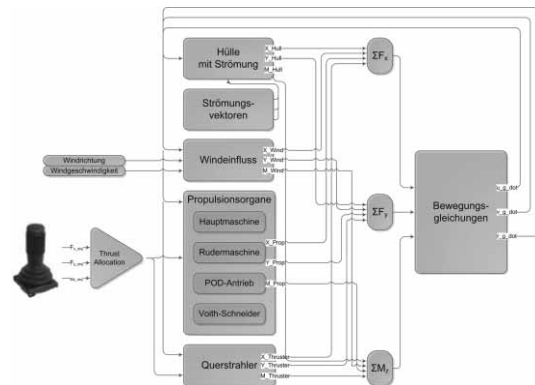


Abbildung 2. Struktur des Bewegungsmodells

Die Antriebskräfte werden mit den Kraftvektoren von Wind und Strömung superpositioniert und den Widerstandskräften der Schiffshülle im Wasser gegenübergestellt. Über die Kirchhoff'schen Bewegungsgleichungen [3] werden daraus die Beschleunigungen, Geschwindigkeiten und Ortsveränderungen berechnet.

Als konventionelle Art wird die Schiffssteuering durch direkte Ansteuerung der Propulsionsorgane über separate Stellhebel umgesetzt, bspw. mit einem Maschinentelegraph für die Hauptmaschinendrehzahl. Für die DP-Anforderungen wird die Gesamtbewegung des Schiffes mittels des dreiachsigen Joysticks vorgegeben. Der Sollkraftvektor muss auf die, je nach Schiffstyp variierenden, verfügbaren Antriebe aufgeteilt werden. Diese Allokation soll aus ökonomischen und umwelttechnischen Gesichtspunkten energieminimal erfolgen, wofür eine Optimierung notwendig ist. Hier wird ein Simplex-Algorithmus für eine quadratische Zielfunktion und lineare Nebenbedingungen angewendet. Der Algorithmus basiert auf [4] und eine Version ist als C-Code in NLOpt[®] [5] enthalten.

Dieser Optimierungsalgorithmus wird als HiL (Hardware in the Loop) auf dem Industrie-PC getestet. Über den SIMULINK CODER[®] (Real-Time Workshop[®]) wird der Code für die Echtzeitanwendung in xPC-Target Zielsystem erstellt. Mittels UDP-Schnittstellen wird der Algorithmus auf dem Industrie-PC angesteuert und liefert die Eingangsgrößen für das Bewegungsmodell in der Testumgebung. Für den Einsatz mit dem Schiffssimulator im MSCW muss die Allocation auf der Taktrate des Simulationsprozesses laufen, die je nach Rechenmodul bei 1 bzw. 10Hz liegt. Hier muss der Simplex-Algorithmus für die Implementierung umgeschrieben und angepasst werden, so dass die einzelnen Optimierungsschritte in

der festen Taktrate des Echtzeitsystems ausgegeben werden.

Der Optimierungsalgorithmus soll weiterhin für die Auslegung des Joystick-Reglers genutzt werden, da bislang die Parameter für ein Schiffsmodell lediglich offline bestimmt werden können. Dies soll eine automatische Parametrierung des Reglers ermöglichen und ein adaptives System bilden.

In der Präsentation wird somit dargestellt, wie in einem existierenden Simulator, der für die maritime Ausbildung genutzt wird, die Anbindung eines zusätzlichen Moduls realisiert werden kann und welche Lösungsvarianten für aufgetretene Probleme entwickelt wurden.

4 References

- [1] Standards for Training and Certification of Watchkeeping: Attachment 1 to the final act of the conference. Nr. /CONF.2/33, 2010
- [2] Det Norske Veritas: Standard for Certification No. 2.14. Maritime Simulator Systems, 2011
- [3] Scharnow, U.: Seemannschaft 3 – Schiff und Manöver. VEB Verlag für Verkehrswesen, 3. Auflage, 1987.
- [4] Powell, M.: Direct search algorithms for optimization calculations. Acta Numerica, Vol. 7, 1998.
- [5] Johnson, S.: The NLOpt nonlinear-optimization package. <http://abintio.mit.edu/nlopt>. 2014

Vorstellung eines vollautomatischen Simulationswerkzeugs zur Bewertung von Fahrzeugkonzepten

Mark Krausz¹, Matthias Zimmer², Hans Christian Reuss¹

¹Forschungsinstitut für Kraftfahrwesen und Fahrzeugmotoren Stuttgart (FKFS)

²Dr. Ing. h.c. F. Porsche AG

mark.krausz@fkfs.de

In diesem Beitrag wird die Gesamtfahrzeugsimulationsumgebung OverNight Testing (ONT) vorgestellt, die durch eine Zusammenarbeit der Dr. Ing. h.c. F. Porsche AG mit dem Forschungsinstitut für Kraftfahrwesen und Fahrzeugmotoren Stuttgart (FKFS) entstanden ist [1] und in der Konzeptentwicklung zur Bewertung von Gesamtfahrzeugkonzepten Verwendung findet. Die Besonderheiten der Simulationsumgebung sind die starke Konfigurierbarkeit und der hohe Grad der Automatisierung und der Modularisierung. Für den Bewertungsablauf ist ein Prozess definiert worden, der von der Einbindung neuer Modellkomponenten über die Erstellung neuer Fahrzeugkonzepte, der Bewertung von Fahrzeugkonzepten anhand verschiedener Manöver bis zur Auswertung und Darstellung der Ergebnisse reicht. Der Vorteil des hier vorgestellten Simulationswerkzeugs liegt in der Beherrschung der Variantenvielfalt von Fahrzeugkonzepten und des Baukastenprinzips, indem Fahrzeugmodelle schnell anhand definierter Module erstellt, verglichen und bewertet werden können.

1 Einleitung und Problemstellung

In der Fahrzeugentwicklung steigt seit Jahren die Zahl der Fahrzeugderivate. Durch die Einführung von Plattformen und Baukästen soll diese Variantenvielfalt beherrschbarer gemacht werden [2]. In der Konzeptentwicklung steht man vor der Herausforderung innerhalb von kurzer Zeit eine Vielzahl an Varianten für jedes Fahrzeugkonzept bewerten zu müssen. Deshalb ist es, wie auch in anderen Unternehmen [3], das Ziel, Gesamtfahrzeugmodelle analog zum Baukastenprinzip in der Produktentwicklung aufzubauen und so die Vielfalt und Komplexität kontrollierbar zu machen. Das Fahrzeugmodell soll sich in Module und Komponenten unterteilen, die ausgetauscht werden können und wiederverwendbar sind. Es ist außerdem zu beachten, dass die vorliegende Menge an Informationen in der Frühen Phase des Fahrzeugentwicklungsprozesses gering ist [4]. Die Modelle der einzelnen Fahrzeugkomponenten mit unterschiedlichem Detaillierungsgrad sollen miteinander kombiniert und im Laufe des Projekts einfache durch komplexere und umfangreichere Modelle ausgetauscht werden können. Um die hohe Anzahl an einzelnen Bewertungsaufträgen beherrschbar zu machen, ist ein klar definierter und strukturierter Bewertungsprozess unabdingbar [5]. Es muss die Möglichkeit bestehen verschiedenste Technologien sowie Parametersätze an den Fahrzeugmodellen bewerten und miteinander vergleichen zu können. Die einfache und schnelle Variation der Komponenten und der Parameter des

Modells wird in der Konzeptbewertung immer wieder benötigt. Um auf die Kompetenzen von anderen Fachbereichen zugreifen zu können ist ein Austausch von Modellen über Simulationsplattformsgrenzen hinweg von hoher Bedeutung [6]. Auch deshalb müssen die Modellgrenzen und –schnittstellen klar definiert sein. Außerdem ist es wichtig, dass die Ergebnisse einfach und schnell betrachtet werden können und untereinander vergleichbar sind. Alle zur Bewertung verwendeten Daten sowie die Ergebnisse sollten gesammelt abgelegt werden, um in der Lage zu sein zu späteren Zeitpunkten die Bewertung zu wiederholen und die entstandenen Ergebnisse nachzuvollziehen.

2 Das Simulationswerkzeug OverNight Testing (ONT)



Abbildung 1. Startfenster ONT-Oberfläche

Mit Hilfe des Simulationswerkzeugs ONT, dessen Startoberfläche in Abbildung 1 zu sehen ist, sollen die dargestellten Anforderungen an ein Bewertungswerkzeug zur Nutzung in der Konzeptbewertung erfüllt werden. Das Werkzeug wurde in MATLAB und Simulink erstellt und ermöglicht eine einfache und benutzerfreundliche simulative Bewertung von Fahrzeugkonzepten. Die Wahl fiel einerseits aufgrund der Möglichkeiten bei der Erstellung von Benutzeroberflächen, der einfachen Umsetzung und Integration von Funktionen zur Datenverarbeitung, welche zur Vorbereitung und Nachbereitung von Simulationen notwendig sind und wegen der weiten Verbreitung des Programms zur Simulation des Fahrzeugverhaltens innerhalb des Unternehmens, auf MATLAB und Simulink. Hierdurch können bereits vorhandene Modelle besser in die Struktur integriert werden. Alle Funktionen lassen sich über Benutzeroberflächen bedienen, wodurch der Nutzer keine Vorkenntnisse im Umgang mit MATLAB oder Simulink benötigt.

Im Folgenden wird eine Konzeptbewertung beispielhaft durchlaufen und die einzelnen Funktionen schrittweise erklärt.

2.1 Beispiel einer Konzeptbewertung

In diesem Beispiel wird das fiktive Konzept eines Elektrofahrzeugs mit dem Namen *Konzept 1* bewertet. Der Ablauf zur Bewertung eines Fahrzeugs in ONT ist in Abbildung 2 dargestellt und läuft immer gleich ab.

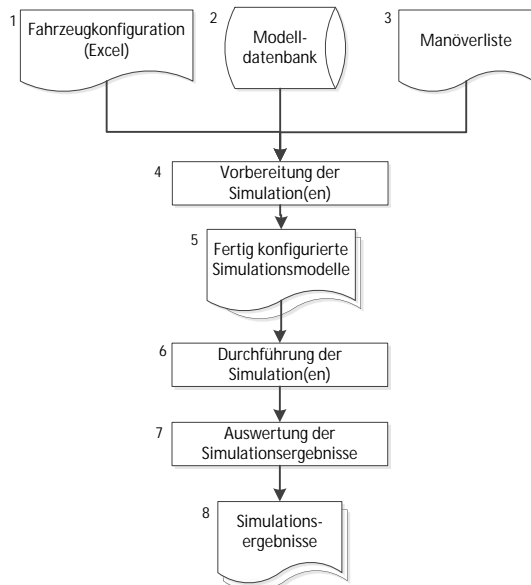


Abbildung 2. Prozess zur Bewertung eines Fahrzeugkonzepts in ONT

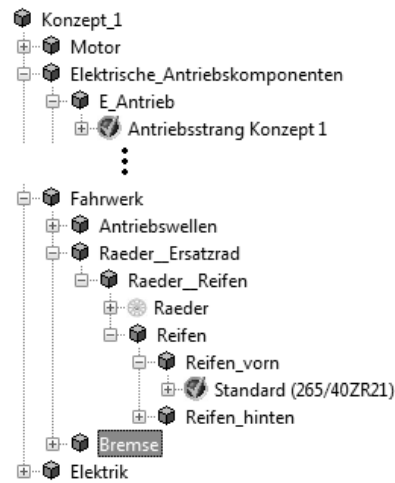


Abbildung 3. Ausschnitt aus Fahrzeugbaum in ONT

Damit ein Fahrzeugkonzept mit dem Simulationswerkzeugs ONT bewertet werden kann, müssen die Komponenten des Fahrzeugs in eine Exceldatei (1) in einer vorgegebenen hierarchischen Struktur abgelegt werden. Die in der Exceldatei vorliegenden Informationen werden eingelesen und in der in Abbildung 3 ausschnittsweise zu sehenden Baumstruktur in ONT dargestellt.

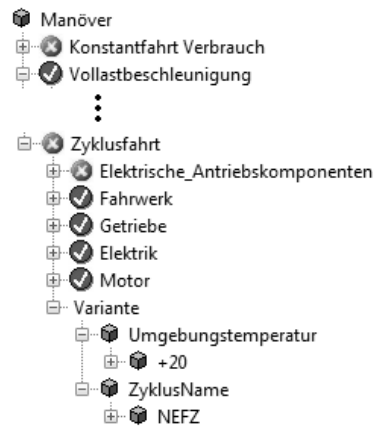


Abbildung 4. Ausschnitt aus Manöverbaum in ONT

Neben den Informationen über die Komponenten des Fahrzeugs muss definiert werden, welche Simulationsszenarien (im Folgenden Manöver genannt) betrachtet werden sollen. Hierfür kann aus vorher definierten Manövern eine Liste (3) zusammengestellt werden. Es ist z.B. möglich verschiedene Beschleunigungen aus dem Stand, Fahrten mit konstanter Geschwindigkeit, Elastizitätsbeschleunigungen, Fahrzyklen und Fahrten auf Rundkursen zu simulieren. Randbedingungen und Vorgaben des Manövers wie z.B. die Außentemperatur, SOC-Startwert und die

Fahrgeschwindigkeiten können vom Benutzer vorgegeben werden.

In Abbildung 4 ist ein Ausschnitt aus einem Manöverbaum zu sehen. Die Manöver haben an die einzelnen Komponenten des Gesamtfahrzeugmodells Anforderungen, die voll automatisiert überprüft werden. Diese Anforderungen werden im Baum unter dem Manöver in der gleichen Hierarchie, in der auch die Fahrzeugstruktur vorliegt, dargestellt. Ist eine Anforderung erfüllt, erscheint ein Haken neben der Komponente. Wird eine Anforderung nicht erfüllt, werden die Komponente und deren übergeordnete Kategorie mit einem Kreuz markiert. Sobald eine der Komponenten mit einem Kreuz markiert ist, kann das Manöver nicht mehr simuliert werden. Die Anforderungen der Manöver werden im Abschnitt 2.6 genauer beschrieben.

In diesem Beispiel kann aufgrund der vorliegenden Fahrzeugdaten und der Manöveranforderung nur ein Manöver simuliert werden. Die Manöver Zyklusfahrt und Konstantfahrt können nicht ausgewählt werden, weil die Anforderung an die elektrischen Antriebskomponenten nicht ausreichen. Das kann z.B. der Fall sein, wenn die Wirkungsgrade der E-Maschine nicht bekannt sind. Eine Aussage über den Verbrauch wäre dann nur sehr ungenau und würde den zugrundeliegenden Anforderungen nicht genügen.

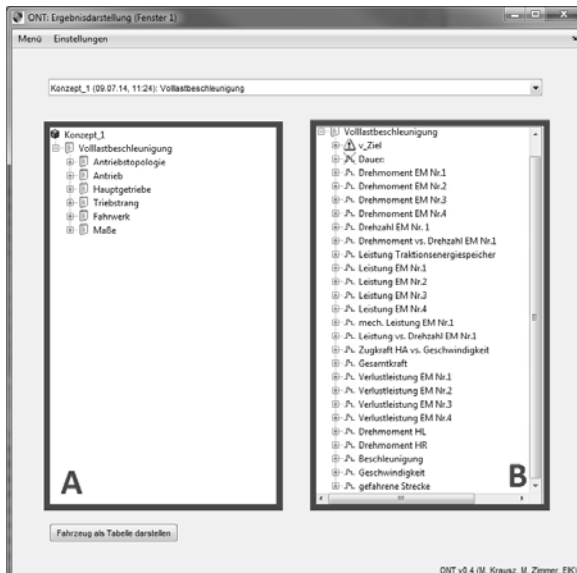


Abbildung 5. Das Fenster zur Ergebnisdarstellung von ONT

Aus der Fahrzeugdefinition (2) und der Manöverliste (3) werden mit Hilfe einer Datenbank, in der alle Modelle und Parameter abgelegt sind, vollständig

konfigurierte Fahrzeugmodelle automatisiert erzeugt (5).

Anschließend können die Simulationen durchgeführt werden (6). Hier wird in einem kleinen Statusfenster ein Überblick über den aktuellen Status gegeben. Das Simulationsmodell bleibt für den Benutzer im Hintergrund verborgen.

Nach der Ausführung einer Simulation wird optional eine automatisierte Auswertung der Simulationsergebnisse durchgeführt (7). Zur Betrachtung der Simulationsergebnisse kann die Ergebnisdarstellung geöffnet werden. In der Ergebnisdarstellung können neben aktuellen auch archivierte Simulationsergebnisse (8) jederzeit betrachtet werden. Der Inhalt des Fensters, welches in Abbildung 5 zu sehen ist, ist vollständig konfigurierbar. Hier können neben der Fahrzeugkonfiguration (A) aus einer Liste (B) Signale ausgewählt und die Verläufe in verschiedenen Diagrammen dargestellt werden. Ein Beispiel für die Darstellung mehrerer Signale in einem Diagramm ist in Abbildung 6 zu sehen. Für jedes Manöver lässt sich konfigurieren, welche Signale zur Auswahl stehen. Die Signale können miteinander kombiniert in Plots dargestellt werden und in Exceldateien zur Weiterverwendung exportiert werden.

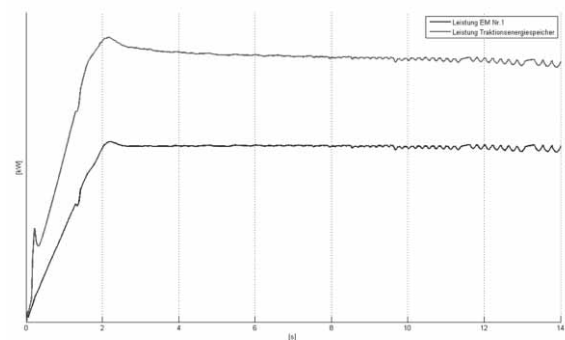


Abbildung 6. Darstellung von Signalen in der Ergebnisdarstellung von ONT

2.2 Die Fahrzeugstruktur

Die in Abschnitt 2.1 erwähnte Gliederung der Excel-liste ist die zugrundeliegende Ordnungsstruktur in ONT, auf die im Folgenden genauer eingegangen wird. Diese Struktur wird unternehmensweit zur Auflistung von im Fahrzeug verbauten Komponenten verwendet. Neben der ordnerbasierten Datenbank, in der die Modelle und die Parameterdateien gespeichert sind, ist die Struktur auch in den Signalen des Simu-

lationsmodells wiederzufinden. Das Gesamtfahrzeugmodell besitzt einen zentralen Signalbus, der sich in die gleichen Ebenen unterteilt. Dieser dient allein der sinnvollen und durchgehenden Strukturierung der Vielzahl an Signalen und hat keine Eigenschaften eines realen Kommunikationsbuses. Durch die Wiederverwendung dieses Systems können Modelle und Parametersätze von Komponenten in dem Dateisystem leicht gefunden und abgelegt werden, die Teilmodelle im Gesamtfahrzeugmodell leicht lokalisiert und einzelne Signale von Teilmodellen aus dem Signalbus ausgewählt werden.

In Abbildung 7 ist beispielhaft zu sehen, wo in der Fahrzeughierarchie die Komponente Traktionsenergiespeicher eingeordnet ist.

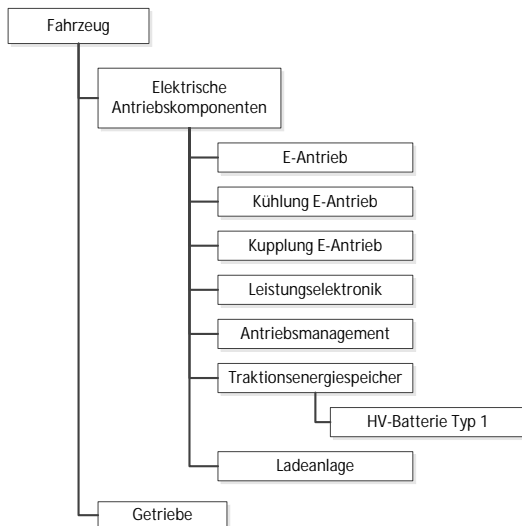


Abbildung 7. Hierarchische Einordnung der HV-Batterie Typ 1 in die Ordnungsstruktur

2.3 Das Gesamtfahrzeugmodell

Das Gesamtfahrzeugmodell in ONT besteht aus einer Vielzahl an Komponenten, die abhängig von dem zu simulierenden Manöver durch unterschiedlich detaillierte Modelle abgebildet werden. Bei der Simulation in ONT handelt es sich immer um eine Vorwärtssimulation. Das Modell auf höchster Ebene ist in Abbildung 8 zu sehen. Alle Blöcke haben den Gesamtfahrzeugbus als Eingangssignal. Der Ausgang jedes Blocks ist ein Bus mit den aggregierten Signalen seiner Unterkomponenten. Auf den genauen Aufbau der Modelle der Fahrzeugkomponenten wird in Abschnitt 2.4 genauer eingegangen.

Im Gesamtfahrzeugmodell ist der Antriebsstrang mit den Komponenten Motor/E-Maschine, Kupplung,

Getriebe, Verteilergetriebe, Differentiale, Antriebswellen, Bremsen, Räder und Reifen abgebildet. Neben der physikalischen Abbildung der Komponenten werden soweit notwendig Steuerungen und Regelungen der Systeme ebenfalls modelliert. Im Falle eines elektrifizierten Antriebsstrangs kann außerdem die Hochvolt-Seite mit E-Maschine, Leistungselektronik, Traktionsenergiespeicher und Ladeanlage dargestellt werden. Hierdurch kann eine sehr gute Abbildung der Längsdynamik von verbrennungsmotorischen sowie elektrifizierten Fahrzeugen erreicht werden.

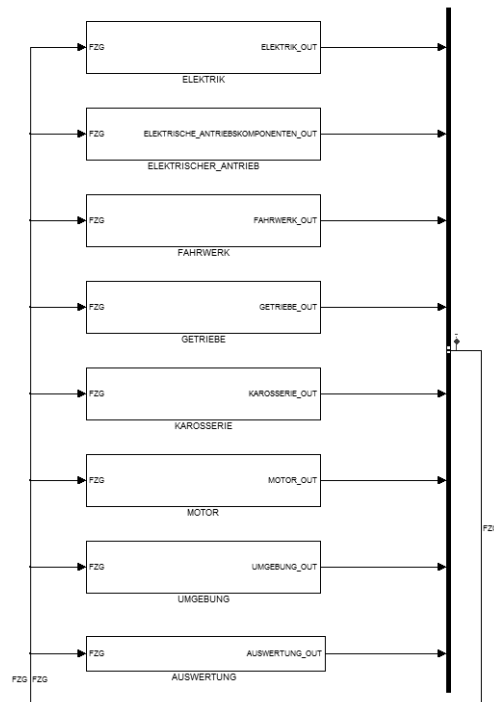


Abbildung 8. Gesamtfahrzeugmodell auf oberster Ebene

2.4 Die Modelle der Fahrzeugkomponenten

Jede Komponente, die im Gesamtfahrzeugsimulationsmodell verwendet werden soll, ist in gleicher Weise abgelegt. Eine Komponente kann durch verschiedene Modelle abgebildet werden. Sind nur wenige Informationen über die Komponente vorhanden, werden sehr einfache Modelle verwendet. Wenn die Menge an Informationen im Laufe der Zeit wächst, können auch komplexere Modelle, die mehr Daten benötigen, parametrisiert werden. Für jede Modellvariante muss der Detaillierungsgrad in den fünf Kategorien Mechanik, Elektrik, Thermodynamik, Chemie und Logik zwischen 0 (nicht abgebildet/vorhanden) und 10 (entspricht Realität) definiert werden. Eine beispielhafte Definition des Detaillierungsgrads der

Elektrik und der Thermodynamik eines Traktionsenergiespeichers ist auszugsweise in Tabelle 1 dargestellt.

DG	Elektrik	Thermodynamik
0	nicht abgebildet	nicht abgebildet
1	einfacher Widerstand	0-dim. Gesamtsystem
2	einfacher Energiespeicher	0-dim. Zellen
3	temp.-abhäng. Widerstand	1-dim. Gesamtsystem
..
10	Batterie auf HiL	Batterie auf HiL

Tabelle 1. Beispiel für die Definition des Detaillierungsgrads eines Traktionsenergiespeichers

Die verschiedenen Modellvarianten bestehen, wie in Abbildung 9 beispielhaft für die Komponente *HV-Batterie Typ 1* dargestellt, aus einem Parametersatz, dem eigentlichen Modell und einer zum Modell gehörenden Metadatei. In dieser Metadatei ist hinterlegt, welche Ein- und Ausgangssignale und Parameter das Modell benötigt, sowie weitere Zusatzinformationen wie z.B. Modellautor, Erstellungsdatum, Einschränkungen des Modells, usw.

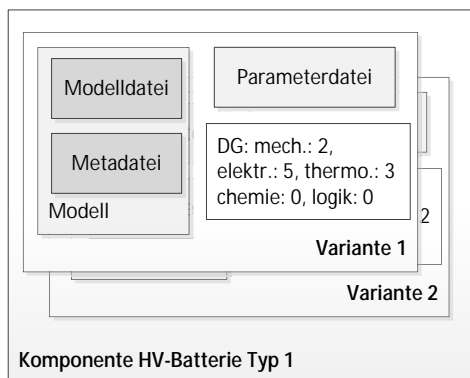


Abbildung 9. Die Komponente *HV-Batterie Typ 1* mit den dazugehörigen Daten

Die Grobstruktur des Metadatei, die als XML-Datei ausgeführt ist, ist in Code 1 zu sehen.

```

1 <name>...</name> (Modellname)
2 ... (weitere Zusatzinformationen)
3 <inputs> (Auflistung aller Eingänge)
4 <element>
5 ...
6 </element>
7 ...
8 </inputs>
9 <outputs> (Auflistung aller Ausgänge)
10 <element>
11 ...
12 </element>

```

```

13 ...
14 </outputs>
15 <parameters> (Auflistung aller
    Modellparameter)
16 </parameters>

```

Code 1. Grundstruktur der XML-Metadatei der Komponentenmodelle

Das Simulationsmodell der einzelnen Komponente besitzt als Eingang immer den Signalbus des Gesamtfahrzeugs, der in Block *Input Interface* geführt wird. Hierdurch kann jedes Signal aus dem Gesamtfahrzeugmodell sehr einfach jedem Teilmodell zur Verfügung gestellt werden. In diesem Block werden die für das Modell benötigten Signale aus dem Gesamtfahrzeugbus heraus geleitet und an das Modell weitergeleitet. Die Modellausgänge werden in dem Block *Output Interface* in den definierten Bus eingebunden. Da der Bus und die darin enthaltenen Signale fest definiert sind, müssen hier eventuell durch das Modell nicht zur Verfügung gestellte Signale „leer“, das heißt ohne Wert, erzeugt werden, um die Busdefinition in Simulink nicht zu verletzen.

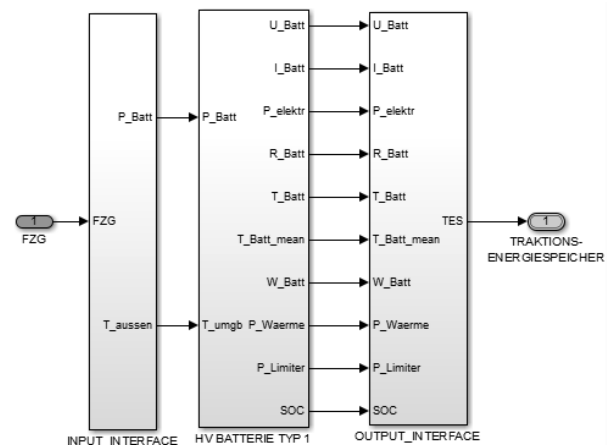


Abbildung 10. Das Modell der Komponente *HV-Batterie Typ 1* mit Ein- und Ausgängen

Die Eingangs- sowie die Ausgangssignale sind, wie vorher beschrieben, in der Metadatei des Modells hinterlegt. Durch die beiden Schnittstellenblöcke können „Fremdmodelle“ leichter in das Gesamtfahrzeugmodell integriert werden, da hierfür nur die Blöcke *Input* und *Output Interface* angepasst werden müssen und nicht das Hauptmodell verändert werden muss. Außerdem können die Modelle durch die standardisierten Schnittstellen sehr einfach ausgetauscht werden. Das Modell der Komponente *HV-Batterie Typ 1* mit seinen Ein- und Ausgängen ist in Abbildung 10 zu sehen.

2.5 Einbindung neuer Modellvarianten in ONT

Mit Hilfe des Simulationswerkzeugs ist es auch möglich neue Komponenten über einen standardisierten Ablauf (siehe Abbildung 11) in die Datenbank von ONT aufzunehmen. In diesem Prozess werden verschiedene Methoden der Verifikation [7] verwendet, auf die hier nicht im Detail eingegangen wird. Hierfür sind die Modelldatei, die Parameterdatei und eine Metadatei, wie die Abbildung 9 zeigt, notwendig.

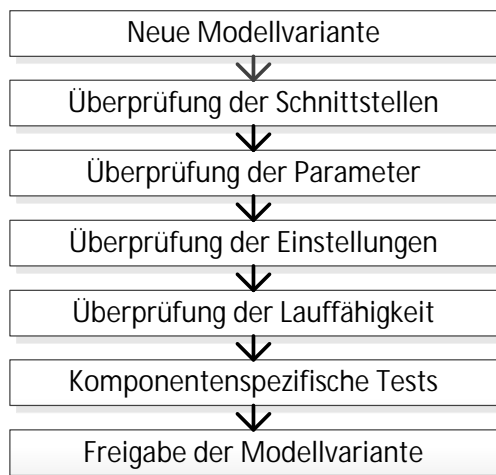


Abbildung 11. Ablauf bei der Einbindung neuer Modellvarianten

Sind diese Daten vorhanden, können diese über eine Eingabemaske eingegeben werden. Es wird anschließend überprüft, ob die Ein- und Ausgangssignale korrekt definiert, bzw. im Fahrzeugbus auffindbar sind, und ob alle Parameter des Modells in der Parameterdatei vorliegen und in der richtigen Dimension definiert sind. Des Weiteren werden die in der Metadatei vorhandenen Informationen auf Vollständigkeit kontrolliert. Nachdem die formalen Überprüfungen abgearbeitet sind, wird ebenfalls nachgeprüft, ob das Modell grundsätzlich lauffähig ist. Für jede Modellkomponente sind Tests definiert, die durchlaufen werden, um zu überprüfen, ob das Modell sich wie erwartet verhält. Diese Tests umfassen einfache Szenarien wie z.B. bei einem Traktionsenergiespeicher die Entnahme von elektrischer Energie. Hierbei wird getestet, ob sich der Ladezustand verringert. Ein weiteres Beispiel ist die Erfassung des Modellverhaltens bei einem Entladungsversuch unter einen SOC-Stand von 0%. Für die Tests wird das Modell in eine Testumgebung eingebracht, welche die notwendigen Signale erzeugt, und das Modellverhalten über seine Ausgänge verfolgt. Nach dem Durchlaufen der Tests

wird ein Protokoll erstellt und im Falle des Bestehens die Komponente in die Datenbank eingetragen und für die entsprechende Verwendung freigegeben.

2.6 Manöver und ihre Anforderungen an das Modell

Die Fahrzeugkonzepte können vorher definierte Manöver durchfahren. Die Manöver unterteilen sich in zwei Hauptgruppen:

1. Manöver mit zeit- oder weggebundenen Geschwindigkeits- und Höhenprofilen
2. Ereignisgesteuerte Manöver

Bei der ersten Gruppe wird ein Streckenprofil vorgegeben, das von einem digitalen Fahrer nachgefahren wird. Sobald das Streckenprofil sein Ende erreicht, wird die Simulation beendet. Beispiele hierfür sind Zyklusfahrten, längsdynamische Fahrten auf Rundkursen, Fahrten auf Steigungen, einfache Beschleunigungen und Verzögerungen.

Bei der zweiten Gruppe beeinflussen eintretende Ereignisse die weitere virtuelle Fahrt. Hier wird nicht mit einem vorgegebenen Profil gearbeitet, sondern es werden dynamisch während der Simulation Geschwindigkeitsvorgaben an den virtuellen Fahrer gestellt. Beispiele sind Fahrten, bis der Traktionsenergiespeicher leer ist, wiederholtes Beschleunigen und darauffolgendes Verzögern bei Erreichen definierter Geschwindigkeiten.

Diese Manöver sind in der Datenbank abgelegt und lassen sich über eine Eingabemaske zu einer Liste zusammenstellen. Die meisten Manöver verfügen über wenige veränderliche Eingabeparameter, wie z.B. die Außentemperatur, Start-/Zielgeschwindigkeiten oder ähnliches. Außerdem sind für jede Komponentenmodell Anforderungen in Form von Detaillierungsgraden definiert.

In Abschnitt 2.5 wurde das Konzept der Detaillierungsgrade der Modellvarianten vorgestellt. Anwendung finden diese Definitionen bei der Vorbereitung der Simulation. Abhängig von dem zu simulierenden Manöver und der ausgewählten Fahrzeugkonfiguration muss die Entscheidung, welche Modellvarianten der einzelnen Komponenten verwendet werden, automatisiert gefällt werden. Hierfür ist ein Konzept erarbeitet worden, das Manöveranforderungen heißt. Jedes Manöver definiert für jede einzelne Modellkomponente Anforderungen in Form von Mindestvorgaben für den Detaillierungsgrad. Das bedeutet,

dass das Modell einer Komponente einen festgelegten Detaillierungsgrad mindestens erreichen muss, damit es für die Simulation dieses Manövers verwendet werden kann. Ist keine der Modellvarianten einer Komponente detailliert genug, kann das Gesamtfahrzeug mit dieser Komponente nicht das jeweilige Manöver durchfahren. Dadurch soll gewährleistet werden, dass die berechneten Ergebnisse eine Mindestqualität haben. Wenn aber zu wenige Informationen über eine Komponente vorhanden sind, können dadurch ausreichend genaue Modelle nicht vollständig bedatet werden und das Simulationsergebnis würde an Genauigkeit verlieren.

Manöver Konstantfahrt Verbrauch	
Traktionsenergiespeicher	DG: mech: 0, elektr.: 4, thermo: 2, chemie: 0, logik: 0
Ladeanlage	DG: mech: 0, elektr.: 0, thermo: 0, chemie: 0, logik: 0
...	

Abbildung 12. Beispielhafte Manöveranforderungen an die Modellvarianten

Für jedes Manöver liegt in einer XML-Datei eine Liste vor, in der die Anforderungen an jede einzelne Modellvariante einer Komponente in Form von Detaillierungsgraden festgelegt sind. Ein Auszug aus den Anforderungen des Manövers *Konstantfahrt Verbrauch* ist in Abbildung 11 zu sehen. Hierbei handelt es sich um die Simulation einer Fahrt bei konstanter Längsgeschwindigkeit, bei der der Verbrauch ermittelt werden soll. Bei diesem Manöver ist die Modellierung der Ladeanlage unerheblich, da nicht geladen wird. Dementsprechend ist der geforderte Detaillierungsgrad in allen Kategorien 0, d.h. es muss kein Modell vorhanden sein. Im Gegensatz hierzu ist das elektrische und thermische Verhalten des Traktionsenergiespeichers relevant und deshalb werden hier Mindestanforderungen an den Detaillierungsgrad gestellt.

3 Ausblick

Das Simulationswerkzeug ONT wird in der Frühen Phase zur Konzeptbewertung herangezogen, um die Vielzahl an Varianten, die in dieser Projektphase beurteilt werden müssen, schnell und übersichtlich zu bewerten. Durch den strukturierten Aufbau in allen Bereichen des Werkzeugs, der starken Konfigurierbarkeit, den hohen Grad der Automatisierung, auf der

anderen Seite aber der trotzdem noch vollständigen Offenheit des Simulationsmodells, ergeben sich Möglichkeiten, ONT als Simulationsplattform in allen Bereichen der Entwicklung zu verwenden, insbesondere dort, wo noch keine alternativen, diesen Funktionsumfang umfassenden Werkzeuge zur Verfügung stehen. Um die Anwendbarkeit von ONT auf andere Fragestellungen zu bewerten, läuft ein multidisziplinäres Projekt mit verschiedenen Fachbereichen innerhalb der Entwicklung.

4 Literatur

- [1] Zimmer: *Entwicklung eines durchgängigen Simulationsprozesses zur Effizienzsteigerung und Reifegraderhöhung von Konzeptbewertungen in der Frühen Phase der Produktentstehung*, Dissertation Universität Stuttgart, in Veröffentlichung, 2014
- [2] Schmieder, Thomas: *Plattformstrategien und Modularisierung in der Automobilentwicklung*, Shaker Verlag, 2005
- [3] Bewersdorff, Pfau: *Beherrschung der Variantenvielfalt durch prozesssichere Simulation*, ATZ Oktober 2011, Band 113, Ausgabe 10, pp 774-779
- [4] Zimmer, Krausz, Reuss: *OverNightTesting – Ein Simulationsprozess zur Konzeptbewertung in der Frühen Phase*, 7. Grazer Symposium Virtuelles Fahrzeug, 2014
- [5] Verworn: *Management der frühen Innovationsphasen*, Springer Verlag, 2007
- [6] Seiffert: *Virtuelle Produktentstehung für Fahrzeug und Antrieb im Kfz*, Vieweg + Teubner Verlag, 2008
- [7] Sargent: *Verification and Validation of Simulation Models*, Proceedings of the 2003 Winter Simulation Conference

Ontologie-unterstützte Systemmodellierung und Simulation in MATLAB/Simulink

Thorsten Pawletta, Daniel Pascheka, Artur Schmidt, Sven Pawletta

Hochschule Wismar, Forschungsgruppe CEA

thorsten.pawletta@hs-wismar.de

Die Ontologie-unterstützte Systemmodellierung kombiniert die klassische systemtheoretische Modellierung mit einer ontologischen Systemspezifikation. Konfigurierbare dynamische Modelle mit definierten Ein-/Ausgangsschnittstellen werden in einer Modellbasis organisiert. Die Ontologie beschreibt deklarativ für einen Gegenstandsbereich wie aus den dynamischen Modellen eine Menge unterschiedlicher Systemkonfigurationen erzeugt werden können. Aufbauend auf der System Entity Structure (SES) Ontologie wird eine Toolbox zur ontologischen Modellierung für die MATLAB/Simulink Umgebung vorgestellt.

1 Einleitung

Heutige Simulationsumgebungen unterstützen die modular-hierarchische Modellierung, die Kombination unterschiedlicher Modellierungsformalismen und stellen leistungsfähige numerische Verfahren zur Simulation und Datenauswertung bereit. Gemäß dem Lebenszyklusmodell in [1] werden die konzeptuelle Modellierung, die Datenmodellierung und das Experiment-Design noch nicht äquivalent berücksichtigt.

Eine zunehmende Anforderung bildet vor allem das Experimentieren mit unterschiedlichen Systemvarianten. Der herkömmliche Ansatz besteht in der isolierten Modellierung einzelner Varianten und einem händischen oder skriptgesteuerten Experimentieren. Einige Simulationswerkzeuge, wie beispielsweise MATLAB/Simulink, unterstützen eine Variantenmodellierung auf der Ebene der dynamischen Systemmodelle unter Nutzung komponentenbasierter Techniken. Die Aktivierung einer bestimmten Variante erfolgt nach [2] vor der Simulation mittels sogenannter Steuerungsvariablen, die im Systemmodell definiert sind. Damit wird das Experimentieren mit Varianten vereinfacht und automatisierbar. Teilweise wird der Ansatz mit externen Werkzeugen zur Variantenmodellierung kombiniert [3, 4]. Ein Problem dabei stellt die fehlende Synchronisation des externen Variantenmodells und der dynamischen Systemmodelle dar.

Mit der Ontologie-unterstützten Modellierung wird ein ganzheitlicher Ansatz, von der konzeptuellen Modellierung bis zum zielgerichteten Experimentieren mit unterschiedlichen Systemvarianten, verfolgt. Der Ontologie-Begriff stammt aus der Philosophie

und steht dort für die Lehre vom Sein. In der Informatik wird unter Ontologie im Allgemeinen eine formal geordnete Darstellung von Begriffen und deren Relationen verstanden. Im Speziellen wird Ontologie in der Informatik gemäß [5] unterschiedlich und zum Teil widersprüchlich benutzt. Nachfolgend wird der Term Ontologie gemäß [5, 6] verwendet. Dort wird Ontologie als eine Konzeptualisierung in Form eines Modells definiert, welches ein gemeinsames Verständnis für einen Gegenstandsbereich vermittelt. Die Konzeptualisierung ist wie nach [7] formal und die Spezifikation deklarativ. Weiterhin wird von einer „closed-world assumption“ ausgegangen, d.h. es ist nur wahr was explizit im Modell spezifiziert ist.

Der Gegenstandsbereich des Beitrags ist die Modellbildung und Simulation modular-hierarchischer Systeme gemäß [8]. Unter Ontologie-unterstützt soll in diesem Kontext eine ontologische Spezifikation in Kombination mit Modellen dynamischer Systeme gemäß [9] verstanden werden. Das Systemverhalten wird in konfigurierbaren dynamischen Modellen (DM) mit definierten Ein- und Ausgangsschnittstellen modelliert, welche in einer Modellbasis (MB) organisiert werden. Die Ontologie beschreibt simulatorunabhängig die Struktur und Konfiguration einer Menge modular-hierarchischer Modelle und definiert Verbindungen zu den DM in der MB. Darüber hinaus kann sie auch eine Menge durchführbarer Experimente mit Links zu Experiment- und Datenbasen spezifizieren. Aufgrund des deklarativen Charakters kann die ontologische Spezifikation bereits in den ersten Phasen des Lebenszyklusmodells beginnen und sukzessive fortgeführt werden.

Zeigler et al. entwickelten mit der System Entity Structure (SES) [10, 11] eine Ontologie für die Domäne der Daten- und Systemmodellierung. Darauf aufbauend konzipierten sie das SES/MB Framework [9], welches die Verbindung mit DM in einer MB definiert und prinzipielle Methoden zur automatisierten Generierung von ausführbaren Simulationsmodellen einführt. Eine softwaretechnische Umsetzung wurde in [12] mit der Simulationsumgebung MS4Me vorgestellt. MS4Me ist in JAVA implementiert und basiert auf dem Discrete Event System (DEVS) Formalismus. Demgemäß müssen die DM DEVS konform modelliert werden.

Die Arbeiten in [13, 14, 15] zeigen, dass sich das prinzipielle SES/MB Konzept vorteilhaft zur Lösung ingenieurtechnischer Problemstellungen eignet. Die SES Ontologie basiert auf wenigen Beschreibungselementen und Axiomen. Dadurch ist sie für Ingenieure einfacher zugänglich als alternative Entwicklungen, wie zum Beispiel Protegè [16]. Eine Anwendungsvoraussetzung für Ingenieure ist aber die Verfügbarkeit in einer ingenieurtechnischen Softwareumgebung und die nahtlose Kombination mit dort integrierten Methoden. Mit der Tiny SES Toolbox [17] existiert für MATLAB/Simulink eine SES Implementierung. Diese erfordert aber grundlegende Kenntnisse der Prädikatenlogik und die Anbindung eines PROLOG Interpreters.

Aufbauend auf der Tiny SES Toolbox wurde eine neue SES Toolbox entwickelt. Sie ist vollständig in MATLAB implementiert, verfügt über ein graphisches Frontend, erfordert ausschließlich MATLAB Kenntnisse und stellt verschiedene Methoden zur Modellsynthese bereit. Damit können mit geringem Aufwand automatisiert ausführbare Simulationsprogramme für alle Simulink Blocksets (SimEvents, Simscape, ...) oder die MATLAB/DEVS Toolbox [18] erzeugt werden. Im Abschnitt 2 werden Grundlagen zum SES/MB Konzept, der originären SES Ontologie sowie einige SES Modifikationen zusammengefasst. Anschließend werden im Abschnitt 3 die Softwarearchitektur und grundlegende Methoden der Toolbox sowie die Integration mit MATLAB/Simulink Werkzeugen vorgestellt. Abschließend erfolgen eine Zusammenfassung und ein Ausblick auf laufende Arbeiten.

2 Theoretische Grundlagen

2.1 SES/MB Konzept

Das SES/MB Konzept in [9] kombiniert die SES Ontologie mit der klassischen Arbeitsweise der Modellbildung und Simulation modular-hierarchischer Systeme. Abbildung 1 zeigt die prinzipiellen Verknüpfungen der einzelnen Elemente.

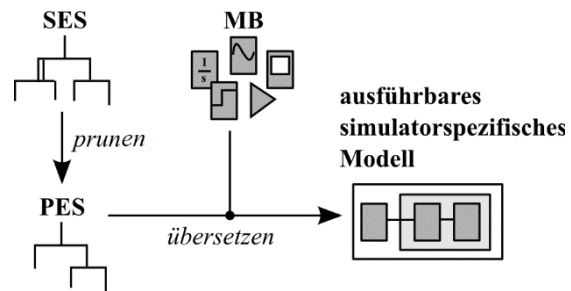


Abbildung 1. SES/MB Konzept nach [9]

Konfigurierbare DM mit definierten Ein- und Ausgangsschnittstellen werden analog den Simulink Blocksets in einer MB organisiert. Durch Komposition und Parametrierung können mit den DM unterschiedliche modular-hierarchische Systemmodelle konfiguriert werden. Die SES ist im hiesigen Kontext ein deklaratives Modell, das für einen abgeschlossenen Gegenstandsbereich eine Menge von möglichen Systemkonfigurationen und Links zu den DM in der MB beschreibt. Darüber hinaus kann die SES Experimentszenarien spezifizieren, worauf aber im Beitrag nicht eingegangen wird. Die SES ist damit eine Art „variabler Bauplan für verschiedene Systemkonfigurationen“, in Form einer speziellen Baumstruktur. Der Auswahlprozess einer konkreten Systemkonfiguration basiert auf einem Pruning Verfahren und erfordert Festlegungen bezüglich der Variabilitäten. Das Ergebnis des Prunings ist eine entscheidungsfreie Baumstruktur und wird Pruned Entity Structure (PES) genannt. Die Generierung eines ausführbaren Simulationsmodells erfordert einen Übersetzer, welcher auf Basis der PES und unter Verwendung der referenzieren DM in der MB ein ausführbares Modell erzeugt.

2.2 Originäre SES Ontologie & Modifikationen

Die Konzeptualisierung der SES Ontologie basiert auf verschiedenen Beschreibungselementen und Axiomen. Diese sind nach Kategorien und Zugehörigkeiten in Abbildung 2 zusammengefasst.

SES:

ELEMENTE:

KNOTEN:

Entitäten

Attribute

DESKRIPTIVE KNOTEN:

Aspekt

-MultiAspekt

Spezialisierung

KANTEN:

Entitätenkante

+Auswahlregeln_Geschwister-Aspekte

Aspektkante

Kopplungsrelationen

Spezialisierungskante

Auswahlregeln

MultiAspektkante

Replikation & Kopplungsrelationen

+Zwangsbedingungen

Semantische Relationen

+SES Variablen, Funktionen, Prioritäten

AXIOME:

Alternatierender Modus

Strikte Hierarchie

Einheitlichkeit

Valide Brüder

Zugeordnete Attribute (Variablen)

Vererbung

Abbildung 2. Elemente und Axiome der mSES

Bei den kursiv und mit + oder – dargestellten Elementen wurden im Rahmen der Toolbox-Entwicklung Modifikationen zur originären Definition der SES [11] eingeführt. Das + steht für zusätzliche Elemente oder Optionen und das – für Einschränkungen. Zur Abgrenzung gegenüber der originären Definition wurde der Term *mSES* eingeführt. Zur sprachlichen Vereinfachung wird nachfolgend dennoch der Term SES verwendet.

Anhand eines fiktiven, plakativen Beispiels sollen grundlegende Elemente und Axiome erläutert werden. Der Gegenstandsbereich ist das dynamische Schmelzverhalten unterschiedlich strukturierter *Eisportionen* (*Ep*), mit folgenden Ausgangsideen:

```
Ep=[Träger(Tr) & Sorte(So) & Typ(Ty)]
Tr=[Waffel(Wa) | Becher(Be)]
So=[Rein(Re) | Mixed(Mi)] mit
    Re=[Vanille(Va) | Schoko(Sc)] und
    Mi=[Va & Frucht(Fr) & Sahne(Sa)]
Ty=[Soft(Sf) | Kugel(Ku)] mit
    Ku=[1mal | 2mal | ... | n_mal]
```

Eine mögliche Konzeptualisierung zeigt die SES in Abbildung 3. Die SES ist in Form von zwei Baum-

strukturen dargestellt, die über den Knoten *Ty* zu mergen sind. Auf das Prinzip des Mergen wird im Abschnitt 3 näher eingegangen.

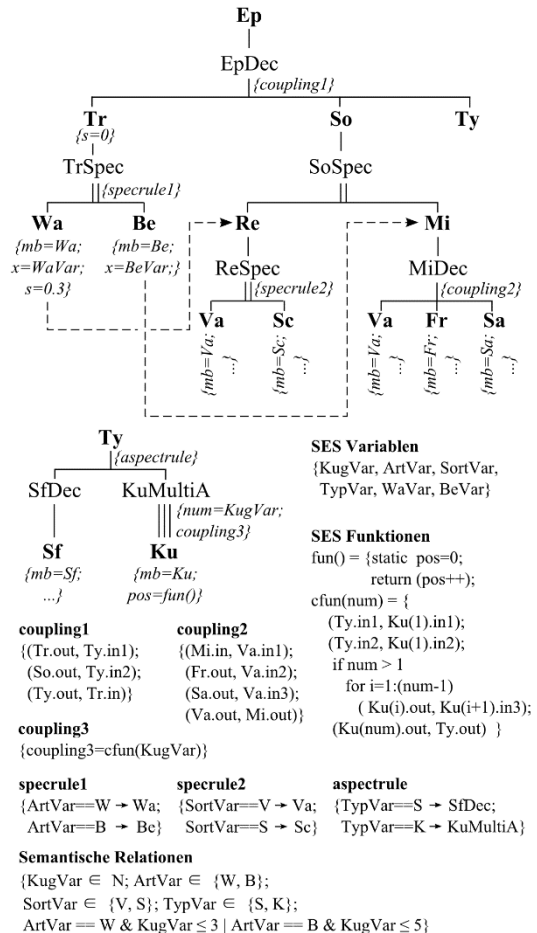


Abbildung 3. SES für das Beispiel Eisportionen

Die Knoten in Fettschrift sind *Entitäten*, die anderen *Deskriptive Knoten*. Gemäß dem 1.Axiom in Abbildung 2 alternieren die beiden Knotentypen. Die *Blattknoten* stehen für atomare Entitäten. Sie definieren mit dem speziellen Knotenattribut *mb* jeweils eine Referenz zu einem DM der MB. In diesem Fall enthält die MB folgende Typen von DM:

MB= { **Wa, Be, Va, Sc, Fr, Sa, Sf, Ku** }

Die Namen der Blattknoten müssen nicht, wie im Beispiel, gleichnamig zu den DM sein. Deskriptive Knoten charakterisieren ihren Vorgängerknoten. Die Suffixe stehen für: *Dec* ≡ *Aspekt*, *MultiA* ≡ *MultiAspekt*, *Spec* ≡ *Spezialisierung*. Im Beispiel beschreiben die Aspekte (*EpDec*, *MiDec*, *SfDec*) Dekompositionen ihres Vorgängerknotens. D.h., die Vorgängerknoten repräsentiert jeweils ein zusammengesetztes Sys-

tem auf Basis der Nachfolgeentitäten, wobei eine Komposition auch aus einem einzigen Subsystem bestehen kann. Die *Aspektkante* kann als Attribut Kopplungsrelationen (*coupling*) spezifizieren. Analog dazu beschreibt ein *MultiAspekt* (*KuMultiA*) eine Dekomposition des Vorgängers aus typgleichen Entitäten des einen Nachfolgers (*Ku*). Der Wertebereich für die zulässige Anzahl wird im Attribut *num* der *MultiAspektkante* spezifiziert. Die von der Anzahl abhängigen Koppungsrelationen (*coupling3*) sind mit einer SES Funktion definiert. Eine Entität kann mehrere Aspekte oder MultiAspekte als Nachfolger definieren, sogenannte *Geschwister-Aspekte*, wie im Beispiel *Ty* mit *SFDec* und *KuMultiA*. In diesem Fall kann die *Entitätenkante*, wie im Beispiel die Kante von *Ty*, ein Attribut (*aspectrule*) mit Auswahlregeln definieren. Alternativ können auch *Zwangsbedingungen*, dargestellt als gestrichelte Kanten, eine Auswahl definieren. *Spezialisierungsknoten*, im Beispiel *TrSpec*, *SoSpec* oder *ReSpec*, beschreiben eine Taxonomie des Vorgängers. D.h., die Vorgängerentität ist durch eine der Nachfolgeentitäten zu präzisieren. Die Auswahl des Nachfolgers kann mit Auswahlregeln an der *Spezialisierungskante* oder mit *Zwangsbedingungen* spezifiziert werden (vgl. in Abb. 3 die *Zwangsbedingungen* von *Wa* nach *Re* und *Be* nach *Mi* bzw. die Attribute *specrule1* und *specrule2*). Die Spezialisierungsrelation basiert auf dem Axiom der *Vererbung*, welches eine Vereinigung von Vaterknoten und ausgewähltem Kindknoten bezüglich ihrer *Knoten-namen*, *Attribute* und *Teilbäume* definiert. Zur eindeutigen Auflösung einer Spezialisierung mit Vererbung von Teilbäumen wurden in [19] *Prioritäten* als zusätzliches Beschreibungselement eingeführt. Die SES in Abbildung 3 enthält keine Vererbung von Teilbäumen. Bezüglich Details zur Vererbung von Teilbäumen sei auf [11, 12, 19] verwiesen.

Die Knoten- und Kantenattribute können konstante Ausdrücke und unter Verwendung der *SES Variablen* oder *SES Funktionen* variable Ausdrücke definieren. Letztere wurden im Rahmen der Toolbox stark erweitert [19]. In Abbildung 3 sind die SES Funktionen in Pseudocode notiert. Im Rahmen der Toolbox sind sie wie gewöhnliche MATLAB Funktionen zu definieren. Mit den *Semantischen Relationen* können umfassende Gültigkeitsbedingungen beschrieben werden.

Aus der SES in Abbildung 3 folgen für das Beispiel folgende 14 zulässige Zusammensetzungen einer Eisportion (Ep):

- (1) $Ep=[Wa \ \& \ Va \ \& \ Sf]$
- (2) $Ep=[Wa \ \& \ Sc \ \& \ Sf]$
- (3-5) $Ep=[Wa \ \& \ Va \ \& \ [1xKu \ | \ 2xKu \ | \ 3xKu]]$
- (6-8) $Ep=[Wa \ \& \ Sc \ \& \ [1xKu \ | \ 2xKu \ | \ 3xKu]]$
- (9) $Ep=[Be \ \& \ [Va \ \& \ Fr \ \& \ Sa] \ \& \ Sf]$
- (10-14) $Ep=[Be \ \& \ [Va \ \& \ Fr \ \& \ Sa] \ \& \ [1xKu) \ | \ 2xKu \ | \ 3xKu \ | \ 4xKu \ | \ 5xKu]]$

2.3 Ableiten einer PES - Pruning

Zur Ableitung einer vollständigen, gültigen Systemkonfiguration wurden in [11] Prinzipien zum *Pruning* einer SES definiert. Diese wurden für die Toolbox in [19] präzisiert. Das Ergebnis des Prunings ist eine Baumstruktur, die als *Pruned Entity Structure* (PES) bezeichnet wird. Nachfolgend werden grundlegende Ideen des Prunings schrittweise anhand der SES aus Abbildung 3 beschrieben.

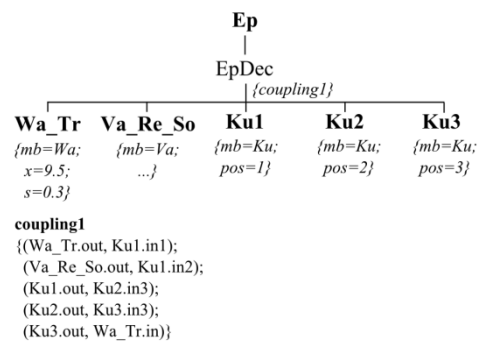
Für das Pruning einer SES müssen die SES Variablen konkrete Wertzuweisungen erhalten. Es sei angenommen, dass folgende Zuweisungen vorliegen:

ArtVar=W für Waffel; SortVar=V für Vanille;
TypVar=K für Kugel; KugVar=3 für #Kugeln;
WVar=9.5 für Waffelparameter x; BVar=Ø

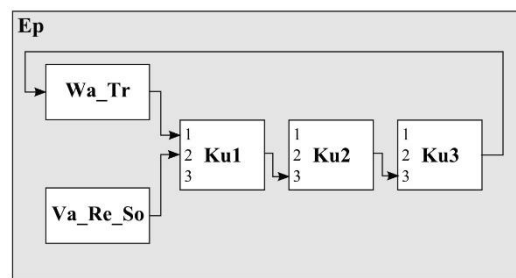
Beim Pruning müssen alle Variabilitäten der SES aufgelöst werden. Es startet immer am Wurzelknoten. Demgemäß werden erst einmal der Wurzelknoten *Ep*, der Aspektknoten *Ep_Dec*, dessen Kantenattribut *{coupling1}* sowie die Nachfolgeknoten *Tr* mit Attribut *{s=0}*, *So* und *Ty* in die PES übernommen. Danach wird top-down im Teilbaum *Tr* der Knoten *TrSpec* mit dem Kantenattribut *{specrule1}* ausgewertet und damit der Knoten *Wa* als Spezialisierung ausgewählt. Aufgrund des Axioms Vererbung erfolgt eine Namen- und Attributvererbung, d.h. eine Knotenzusammenführung. In der PES wird der Knoten *Tr* mit *{s=0}* durch einen neuen Entitätenknoten *Wa_Tr* mit *{mb=Wa; x=9.5; s=0.3}* ersetzt. Demgemäß muß im Kantenattribut *{coupling1}* von *Ep_Dec* die erste und dritte Kopplungsrelation aktualisiert werden (vgl. Abb. 4). Weiterhin ist aufgrund der Zwangsbedingung von Knoten *Wa* zu Knoten *Re* eine Auswahl von *Re* vorzumerken. Der Teilbaum von *Tr* ist analysiert und das Pruning wird top-down am Knoten *So* fortgesetzt. Der Spezialisierungsknoten *So_Spec* führt aufgrund der vorgemerkten Auswahl von *Re* und des Axioms Vererbung zu einer Vereinigung der Entitätenknoten *So* und *Re*. In der PES wird der Knoten *So* in *Re_So* umbenannt. Die weitere Analyse von *Re_Spec* mit dem Kantenattribut *{specrule2}* führt zur Auswahl von Knoten *Va*. Aufgrund des Axioms Vererbung

wird der Knoten Re_So in Va_Re_So umbenannt und erhält das Attribut $\{mb=Va; \dots\}$. Demgemäß muß im Kantenattribut $\{coupling1\}$ von Ep_Dec die zweite Kopplungsrelation aktualisiert werden (vgl. Abb. 4). Der aktuelle Teilbaum ist analysiert und das Pruning wird top-down am Entitätenknoten Ty mit dem Kantenattribut $\{aspectrule\}$ fortgeführt. Daraus folgt die Auswahl des MultiAspektknotens $KuMultiA$. Aus dessen Kantenattribut $\{num=KugVar; coupling3\}$ folgt aufgrund der SES Variablen $KugVar=3$, dass eine Komposition aus drei Entitäten Ku aufzubauen ist. Die Kopplungsbeziehungen werden gemäß $\{coupling3\}$ mit der SES Funktion $cfun(num)$ berechnet. Aufgrund des Axioms *valide Brüder* müssen die Entitäten differenzierte Namen erhalten und werden durchnummeriert, so dass die Entitätenknoten $K1$, $K2$, $K3$ entstehen. Demgemäß müssen für die PES die Kopplungsbeziehungen von $\{coupling3\}$ konkretisiert werden (vgl. Abb. 4). Jeder der drei Knoten erhält das Attribut $\{mb=Ku\}$. Weiterhin muß für jeden Knoten das variable Attribut $\{pos\}$ mit der parameterlosen SES Funktion $fun()$ festgelegt werden. Dieses beschreibt die Position einer Eiskugel in der Eisportion. Nach Ausführung aller Schritte liegt die vollständige PES, wie in Abbildung 4 dargestellt, vor. Abschließend muss durch Auswertung der *Semantischen Relationen*, die UND verknüpft werden, die Gültigkeit der PES geprüft werden. In diesem Fall ist die PES vollständig und gültig. Sie beschreibt genau eine Systemkonfiguration.

als *Flattening* bezeichnet und die daraus resultierende reduzierte PES wird *FPES* genannt. Das Flattening erfordert in diesem Fall eine Modifikation der Koppelungsrelationen von $\{coupling1\}$. Abbildung 5a zeigt die resultierende FPES. Die daraus folgende Modellstruktur wird in Abb. 5b dargestellt.



(a)



(b)

Abbildung 5. (a) FPES der PES; (b) Modellstruktur

Auf Basis der PES oder der FPES kann in Verbindung mit einer MB gemäß Abschnitt 2.1 ein ausführbares Simulationsmodell generiert werden.

3 Aufbau und Methoden der Toolbox

3.1 Softwarearchitektur & Nutzerschnittstelle

Abbildung 6 zeigt die Softwarestruktur der Toolbox in Form eines UML Klassendiagramms. Die Klassen sind in fünf Pakete aufgeteilt. Die Klasse *ses_gui* implementiert die Nutzerstelle und bildet eine zentrale Schnittstellenklasse. Das Paket *GUI* enthält die Klassen zur Realisierung der Programmoberfläche, wie in Abbildung 7 dargestellt. Diese besteht aus der *Menu-Bar* und den drei Subfenstern *Node Properties*, *Model Hierarchy*, *Global Settings*. Ihre Umsetzung erfolgt durch jeweils eine eigene Klasse und eine gemeinsame Oberklasse. Im Subfenster *Model Hierarchy* wird der SES Baum analog zu einem Dateibaum in einem Dateimanager editiert, wobei Zwangsbedingungen farblich dargestellt werden.

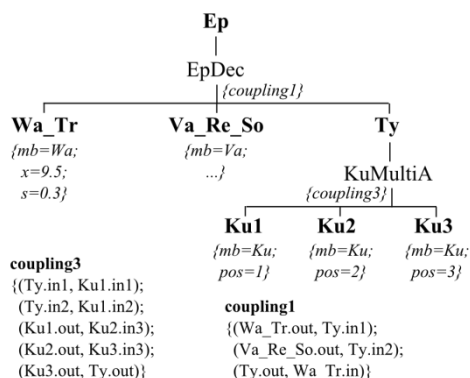


Abbildung 4. Eine vollständige, gültige PES zur SES

Aus dem Attribut *{coupling1}* in Abbildung 4 folgt, dass die Entität *Ty* ein zusammengesetztes System beschreibt. Aus Sicht der Systemdynamik könnte dieses aufgelöst werden. Das Auflösen zusammengesetzter Systeme bietet sich insbesondere bei automatisierten Experimenten oder laufzeitkritischen Applikationen an, um die Komplexität zu verringern. Es wird

Knoten- sowie Kanteneigenschaften werden im Subfenster *Node Properties* und global gültige Eigenschaften der SES im Subfenster *Global Settings* verwaltet. Über die *Menu-Bar* sind alle nutzerbezogenen Methoden verfügbar. Die Datenstrukturen und Methoden zur internen Speicherung und Verwaltung einer SES sind in den Klassen der Pakete *Entity Structure* und *Node Hierarchy* definiert. Die PES und FPES werden als Spezialisierungen einer SES betrachtet und sind durch Klassenableitungen definiert. Damit sind sie analog zu einer SES handelbar und

können analog in der Programmoberfläche dargestellt und verwaltet werden. Darüber hinaus definieren sie die Methoden für das Pruning und Flattening. Im Paket *Parse and Scan* ist ein Parser zur lexikalischen und syntaktischen Analyse implementiert. Dieser prüft kontinuierlich alle Nutzereingaben auf Korrektheit. Es werden die unmittelbare Nutzereingabe als auch die Effekte, die sich aufgrund bereits gespeicherter Informationen ergeben geprüft und bei Bedarf sofort Fehler- oder Warnmeldungen generiert. Daraus folgt ein kurzer Fehlerkorrekturzyklus.

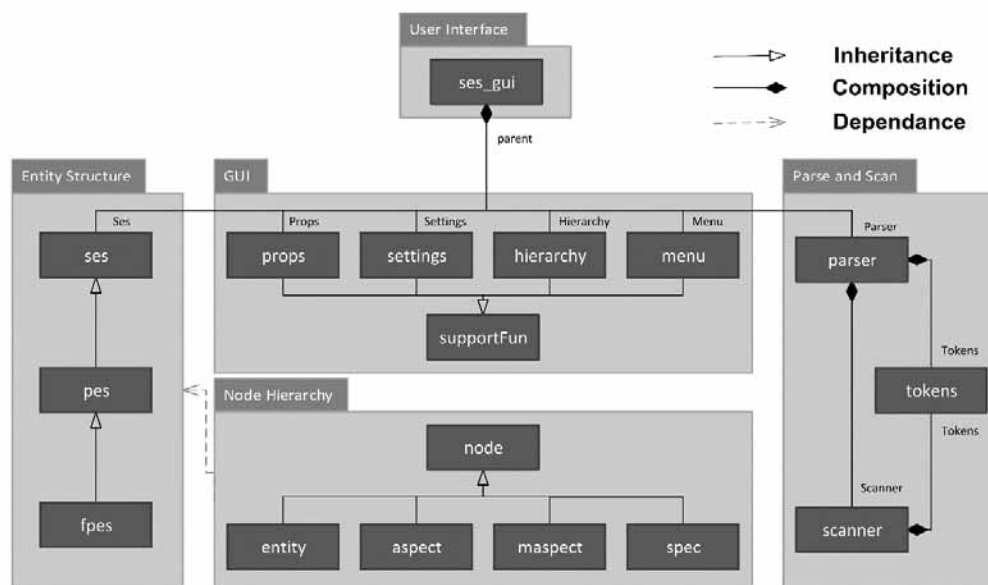


Abbildung 6. Klassenstruktur der Toolbox



Abbildung 7. Programmoberfläche

3.2 Methoden: Merge, Pruning, Flattening

Hinsichtlich der implementierten Methoden soll hier auf grundlegende Aspekte der Methoden zum *Merge*, *Pruning* und *Flattening* eingegangen werden. Die Methoden sind aus der Entwicklungsumgebung über die Menu-Bar ausführbar oder wie gewöhnliche MATLAB Funktionen aufrufbar. Eine ausführliche Beschreibung befindet sich in [19].

Die *Merge* Methode unterstützt die Spezifikation einer SES in separaten Teilbäumen, analog zur Abbildung 3. Dabei kann ein Teilbaum auch eigene globale Eigenschaften, wie SES Variablen, SES Funktionen oder Semantische Relationen definieren. Code 1 zeigt die prinzipiellen Schritte der Merge Methode.

```
load main SES; select merge node;
load imported SES;
if merging ~admissible -> Error;
foreach leave node homonymous with merge node
    merge imported SES tree to main SES tree;
merge global settings;
update displays in GUI
```

Code 1. Prinzipielle Schritte der Methode Merge

Der selektierte Merge Knoten in der Haupt-SES muss eine Blattentität sein und die Merge Operation gilt für alle gleichnamigen Blattentitäten. Die Zulässigkeit des Mergens wird anhand der SES Axiome überprüft. Beim Mergen der Bäume wird der Wurzelknotenname der importierten SES durch den Blattentitätennamen der Haupt-SES ersetzt. Abschließend werden die globalen Eigenschaften zusammengeführt. Namenskonflikte werden beim Mergen automatisch berücksichtigt.

Für das im Abschnitt 2 beschriebene *Pruning* und *Flattening* einer SES stellt die Toolbox drei aufeinander folgende Methoden bereit: (i) *First-Level Pruning*, (ii) *Complete Pruning* und (iii) *Complete Pruning & Flattening*. Code 2 zeigt die grundlegenden Schritte der aufeinander aufbauenden Methoden.

FIRST-LEVEL PRUNING:

1. check pruning permission of SES;
2. verify SES variables;
3. create SES functions in MATLAB;
4. compute SES vars which use SES fcn;
5. transform selection constraints to selection rules;
6. **recursive pruning**;
7. check semantic relations;
 -> valid PES | interim_PES
8. if ~interim_PES
 set PES valid; -> **END**

COMPLETE PRUNING:

9. compute priorities of undecidable aspect nodes & generate aspect selection rules;
10. **recursive pruning**;
11. check semantic relations;
 -> PES
12. if ~flattening
 set PES valid; -> **END**

FLATTENING:

13. rename homonymous leave nodes;
14. **recursive flattening**;
15. set FPES valid; -> **END**

Code 2. Prinzipielle Schritte des Pruning & Flattening

Aufbauend auf den Erläuterungen im Abschnitt 2 sollen dort nicht aufgezeigte Schritte kurz erläutert werden. Eine SES besitzt die (1) Pruning Permission, wenn sie die Axiome erfüllt. Die (2) Verifikation der SES Variablen bezieht sich auf explizite Methodenaufrufe von der MATLAB Ebene. Das (3) Erzeugen von SES Funktionen in MATLAB ist einmalig für alle die SES Funktionen auszuführen, die nicht als MATLAB-builtin Funktionen definiert sind. Es sei erwähnt, dass die SES Funktionen in der Entwicklungsumgebung wie gewöhnliche MATLAB Funktionen editiert werden. Aufgrund der Abspeicherung einer SES als Datenstruktur werden sie als Strings kodiert. Schritt (5) ist auszuführen, wenn global gültige SES Variablen in Abhängigkeit von SES Funktionen definiert sind. Zur Vereinfachung des (6) rekursiven Prunings, welches im groben in Abschnitt 2 erläutert wurde, werden zuvor (5) Zwangsbedingungen in Auswahlregeln transformiert. Aus den Schritten (7,8) folgt, ob eine vollständige, gültige PES erzeugt wurde oder eine noch unfertige Zwischen-PES. Aufgrund des Axioms Vererbung werden an Spezialisierungsknoten auch die Teilbäume des Vater- und des ausgewählten Kindknotens vereinigt. Dadurch können beim Pruning zusätzliche, nicht auflösbare Entscheidungsknoten entstehen, obwohl die SES alle Axiome erfüllt. Das *First-Level Pruning* gestattet das Auffinden derartiger Knoten. In [19] werden zur Lösung des Problems Prioritäten als neues SES Beschreibungselement eingeführt. Die zusätzliche Berücksichtigung von Prioritäten erfolgt beim *Complete Pruning* (9-12). Das *Flattening* (13-15) setzt eine vollständige, gültige PES voraus und erzeugt eine FPES gemäß den Ausführungen im Abschnitt 2.

3.3 Problemorientierte Modellübersetzung

Gemäß dem SES/MB Framework in Abbildung 1 können, durch die Kombination mit einer MB, auf Basis einer PES oder FPES ausführbare Simulationsmodelle generiert werden. Die Toolbox stellt hierfür keinen allgemeinen Übersetzer zur Verfügung, sondern M-File-Templates. Problemorientierte Übersetzung meint, dass der Übersetzer dem Simulator und der MB anzupassen ist. Der generelle Ablauf auf Basis einer FPES ist in Code 3 dargestellt und wird nachfolgend am Beispiel der Übersetzung in ein Simulink-Modell diskutiert.

```
if FPES is ~valid -> Error
INITIALIZATION:
    instantiate empty model
    optional: set solver parameters
TRANSLATION:
    foreach leave node instant. data obj.
    foreach data obj. instant. model obj.(MB)
    from Aspekt attrib. instanz. model coupl.
FINALIZATION:
    optional: e.g. start simulation
```

Code 3. Schritte der Modellübersetzung/-ausführung

Als erstes wird geprüft, ob die FPES beim Pruning als gültig gesetzt wurde. In der *Initialisierungsphase* werden ein neues Modell mit dem Namen des Wurzelknotens der FPES angelegt und optional die Solver-Einstellungen gesetzt. Bei variantenabhängigen Solver-Einstellungen können diese auch optional in der SES spezifiziert und damit in der FPES kodiert sein. Dann werden sie in der Übersetzungsphase analog zum Modell generiert. Prinzipiell gehören die Solver-Einstellungen zur Experimentspezifikation, welche getrennt vom Modell erfolgen sollte. Die eigentliche *Übersetzung* besteht aus drei Schritten. Zuerst wird für jeden Blattknoten der FPES ein MATLAB Datenobjekt angelegt, welches die Blattinformationen nach *Knotenname*, *spezielles Attribut mb* und *restliche Attribute* separiert und speichert. Das spezielle Attribut *mb* kodiert die Verknüpfung zu den DM in der MB, in diesem Fall den Simulink-Blöcken, und ermöglicht deren Instanziierung und Parametrierung. Im dritten Übersetzungsschritt wird das Attribut der Aspektkante mit den Kopplungsbeziehungen ausgewertet und demgemäß die Modellobjektverbindungen generiert. An dieser Stelle sei daran erinnert, dass eine FPES nur einen Aspektknoten enthält. Mit Abschluß der Übersetzungsphase liegt ein ausführbares Modell vor. Die *Finalisierungsphase* ist optional. Im Fall der Generierung eines Simulink-Modells

kann eine Modellübersetzung bereits unter Nutzung von nur fünf API-Funktionen realisiert werden: *load_system*, *new_system*, *add_block*, *add_line*, *set_param*. Realisierte Beispiele für Modellübersetzer werden in [17, 19] beschrieben.

4 Zusammenfassung

Mit der SES Toolbox steht MATLAB/Simulink Nutzern und damit auch Nichtinformatikern ein umfassendes und nutzerfreundliches Werkzeug zur ontologischen Modellierung zur Verfügung. Die Toolbox fügt sich nahtlos in die MATLAB/Simulink Umgebung ein und kann in Kombination mit allen Toolboxes und Blocksets verwendet werden. Für den Gegenstandsbereich der Systemsimulation wurde die Integration am Beispiel einer Modellübersetzung für Simulink aufgezeigt. In diesem Kontext eröffnet die Toolbox beispielsweise ganz neue Wege für die Variantenmodellierung und das Experimentieren mit Modellvarianten. Laufende Arbeitsschwerpunkte sind die Integration weiterer Features bei Multi-Aspekten und der optionale Export einer gespeicherten SES nach XML. Die Toolbox ist nach Registrierung unter http://www.mb.hs-wismar.de/cea/sw_projects.html frei nachnutzbar.

Danksagung

Die Arbeit wurde durch die Deutsche Forschungsgemeinschaft (DFG) unter den Kennzeichen BR 825/62-2 und PA 631/2-2 gefördert.

5 References

- [1] O. Balci. *Verification, validation and testing*, In: Handbook of Simulation, Ed. J. Banks, John Wiley & Sons Inc., pp. 335 – 393, 1998.
- [2] The MathWorks. Simulink User's Guide R2014a, Natick, USA, 2014.
- [3] H. Holdenschick, W. Commerell. *Variantenmanagement in der modellbasierten Produktentwicklung von Fahrzeugsystemen*, In: Proc. ASIM Treffen STS/GMMS 2013, Düsseldorf, ARGESIM Report 41 / AM 145, S. 111 – 118, 2013.
- [4] J. Möck, J. Weiland. *Advancing Virtual Commissioning with Variant Handling*, In: Proc. ASIM Treffen STS/GMMS 2014, Reutlingen, ARGESIM Report 42 / AM 149, S. 7 – 13, 2014.

- [5] B. Henderson-Sellers. *On the Mathematics of Modelling, Metamodelling, Ontologies and Modelling Languages*, Springer Pub., 2012.
- [6] H. Stuckenschmidt. *Ontologien*, 2nd Ed., Springer Pub., 2011.
- [7] T. R. Gruber. *A translation approach to portable ontology specifications*. Knowledge Aquisition, Vol. 5(2), pp. 199 – 220, 1993.
- [8] M. D. Mesarovic, D. Macko, Y. Takahara. *Theory of Hierarchical, Multilevel Systems*. Academic Press, 1970.
- [9] B. P. Zeigler, H. Prähofer, T.G. Kim. *Theory of Modeling and Simulation* 2nd Ed. Academic Press, 2000.
- [10] B. P. Zeigler. *Multifaceted Modeling and Discrete Event Simulation*, Academic Press, 1984.
- [11] B. P. Zeigler, P. Hammonds. *Modeling and Simulation-Based Data Engineering*, Elsevier Academic Press, 2007.
- [12] B. P. Zeigler, H.S. Sarjoughian. *Guide to Modeling and Simulation of Systems of Systems*, Springer Pub., 2013.
- [13] O. Hagendorf, T. Pawletta, R. Larek. *An approach to simulation based parameter and structure optimization of MATLAB/Simulink models using evolutionary algorithms*, In: SIMULATION – Trans. of the Soc. for Mod. and Simulation Int., Vol. 89(9), pp. 1115 – 1127, 2013.
- [14] A. Schmidt, T. Pawletta. *Ein Ontologie-basierter Modellierungs- und Simulationsansatz am Beispiel der ressourceneffizienten Planung spanender Prozessketten*, In: Proc. 15. ASIM Fachtagung Simulation in Produktion und Logistik, Paderborn, HNI-Verlagsschriftenreihe Bd. 316, S. 481 – 490, 2013.
- [15] T. Schwatinski, T. Pawletta, S. Pawletta. *Flexible task oriented robot controls using the System Entity Structure and model base approach*, In: Simulation Notes Europe (SNE), Vol. 22(2), pp. 107 – 114, 2012.
- [16] protégé: *A free, open-source ontology editor and framework for building intelligent systems*, <http://protege.stanford.edu/>, Website, 2014/04/01.
- [17] T. Schwatinski, T. Pawletta. *Ontologische Modellierung und Modellgenerierung in der MATLAB/Simulink Umgebung – Die “Tiny SES Toolbox“*, In: Proc. ASIM Treffen STS/GMMS 2013, Düsseldorf, ARGESIM Report 41 / AM 145, S. 57 – 64, 2013.
- [18] C. Deatcu, T. Schwatinski, T. Pawletta. *DEVS Toolbox for MATLAB – MatlabDEVs Tbx.*, http://www.mb.hs-wismar.de/cea/DEVs_Tbx/MatlabDEVs_Tbx.html, Website 2013.
- [19] D. Pascheka. *Implementierung eines graphischen SES-Editors mit integriertem Pruning Algorithmus in der MATLAB/Simulink Umgebung*, Bachelor Thesis, Hochschule Wismar, FG CEA, 2014.

Conceptual Model and Perdurantist Modeling with Reasoning

M. Fatih Hocaoglu

Istanbul Medeniyet University,

Faculty of Engineering & Architecture, Istanbul/Turkey

mfatih.hocaoglu@medeniyet.edu.tr

Abstract

In this study, the perdurantist modeling approach is extended by a reasoning mechanism. The extension allows a modeler to manage behaviors depending on reasoning results and also gives an elegant support for time delayed systems. A language support is given for this purpose and it starts at an ontological commitment and covers design and coding phases including conceptual model description. In this paper, how an agent driven simulation language supports to extend a perdurantist modeling to ontology based modeling by high-level action descriptions, higher-order world envisionment, dynamic relation management, and a knowledge base for reasoning purpose as necessities of ontology based modeling.

1 Introduction

Agent driven Simulation Framework (AdSiF) [1], [2] is a declarative simulation language, a development environment for simulation and agent programming. It basically provides a state oriented interpreter and a simulation layer to manage simulation execution algorithms for both discrete-event and continuous-event systems. Compared to currently being used agent programming systems, AdSiF gives a different perspective, fusing agent based, object oriented, aspect oriented, and logic programming into a single paradigm, termed here as state oriented paradigm. The power of this paradigm stems from its ontological commitment, which extends from describing what exists to include the modeling of mental abilities through the use of a reasoning mechanism, thereby driving behaviors.

From the ontology based modeling point of view, AdSiF provides a strong programming background to satisfy fundamental ontological notions, which are identity, unity, rigidity, and dependency. Ontologically, AdSiF constructs a world composed of behaviors, which are sequences of states; a set of events triggering behaviors; and a time reference with a reasoning mechanism for managing the behaviors. Since a behavior keeps semantic information between function calls preceeding each other as a state chart, it undertakes an important role as sharing domain semantic. Because of the logic programming paradigm, agent-hood modeling characteristics, and continuous, discrete and symbolic time management

capabilities gives a perdurantist modeling environment.

In this study, the perdurantist modeling approach is extended by a reasoning mechanism. The extension allows modelers to manage behaviors depending on reasoning results and also gives an elegant support for time delayed systems. A language support is given for this purpose and it starts at an ontological commitment and covers design and coding phases including conceptual model description.

In this paper, the role of relations, behavioral aspects, and reasoning mechanism are emphasized in ontology based modeling.

The paper is organized as following. The section following gives brief information about ontology based modeling. Section 3 focuses on logic programming and how it supports entity description and a relation concept between entities. In Section 4, AdSiF ontology support is exemplified by a part of air defense simulation.

2 Ontology based Modeling

Ontology is a term that originated in philosophy and refers to the systematic explanation and study of the nature of existence, or being [3]. In information systems, ontologies are recognised as a useful means for achieving semantic interoperability between different systems and they are key enablers for sharing precise and machine-understandable

semantics among different applications and parties [4].

Ontologies are generally composed of concepts, relations between these concepts and axioms to restrict the interpretation of concepts for a real world phenomenon. It is important that ontologies are of a good quality, in order that they serve their intended purposes and be shared as well as reused by different applications [5].

Concepts, relations of the phenomenon, and the objects surrounded by this phenomenon are strongly related with the conceptual model and the conceptual model paradigm that is very effective for ontology modeling. From this point of view, not only does AdSiF provide a multi modeling paradigm support but also it combines the paradigms in a single paradigm called state oriented paradigm. This gives a rich expressiveness that is one of the quality metric of an ontology. The paradigms, which are supported by AdSiF and combined in state oriented programming, are logic programming, aspect oriented programming, agent based programming, and object oriented programming.

Fundamental ontological notions are summarized by [6] as identity, unity, rigidity, and dependency. Ontological commitment of AdSiF covers the notions by properties given in Table 1, and also it extends notions by logical commitments, behavioral semantics and conditional aspects notions.

Table 1 AdSiF and Ontology Notions

Notion	Language Support
Identity	Each object has its own ID.
Unity	Relations and aggregation relation
Rigidity	Time framed and persistent facts about objects
Dependency	Composition and aggregation depending on design choice

In the literature, it is possible to see some ontology based modeling tools such as ANEMONE, Protégé, OilEd, Apollo, RDFed, OntoLingua, OntoEdit, WebODE, Kaon, ICOM, DOE, WebOnto, Medius Visual Ontology Modeler, LinKFactory Workbench, and K-Infinity [7]. Anemone provides a methodology differs from previous methodologies in the way that it defines concrete development steps, to facilitate use

by both naive and expert ontology developers. This methodology is also supported by ontology design patterns and a prototypical ontology development tool [8].

2.1 Ontology based Modeling Language Support

The most prominent support is brought by the logic programming paradigm. Beyond allowing modelers for defining environments, objects, and relations between objects at meta-level, it also gives the possibilities to develop a reasoning mechanism working on past and present time stamped knowledge, and driving behaviors depending on inferences, which constitute future information set at the same time. Driving a behavior is defined as activating, cancelling, suspending, and resuming it. This is seen as a distilled knowledge inferred from a set of meta-level knowledge and declarations that are placed at the kernel of ontology based modeling.

2.2 Ontological Commitment

AdSiF provides a declarative means to specify a mathematical object called a system. Basically, a system has a time base, inputs (events), states, behaviors, a reasoning mechanism, and a mechanism that manages the dynamic characteristics of the system. In AdSiF, the dynamic characteristic of a system is represented by the behavior descriptions (in the specialized state charts). As a framework, AdSiF promotes a set of design rules and presumes a design skeleton, which is based on its programming paradigm called the SOP paradigm and its ontological view. AdSiF provides an ontological view, which is defined as follows in terms of the paradigms, on which are AdSiF based.

Entities live in a certain environment and have their own properties that distinguish them from each other and an atomic action that manages their properties (OOP perspective). The atomic action creates the interactions that change the environment where the entities reside and share the interactions with other entities as a communication element. The entities sequentially implement their atomic actions in a reasonable semantics called behavior, and the behaviors are executed in parallel and/or sequentially. Each atomic action is wrapped by a state that constructs the behaviors (SOP perspective). The entities interact with one another using event transactions and constitute relations among one another. An interaction has autonomy, reactivity, and a goal — (AgOP perspec-

tive). From a taxonomic view, the behavior categories of an entity represent different behavioral aspects of the entity — (AOP perspective). An entity has beliefs and facts about the environment and about the other entities with which they share the environment. These beliefs and facts constitute a fact dual world envision that contains the entity; the envision may have a set of goals to succeed and a reasoning mechanism with a set of decision-making algorithms — Logic Programming (LP perspective).

3. Logical Envisionment

As seen in the ontological commitment, simulation models and agents have beliefs and facts about the environment they are in and truth preserved predicates to infer new facts, relations, and identities. The time stamped facts about simulation or agent environment not only constitute a fact dual world representation as an inner representation of the environment, in which models are, but also the dual world keeps knowledge on the time axis with past values. This allows modelers to associate truth level and define temporally valid (for a certain duration) knowledge for both truths about simulation models sharing the same environments and relations (dependencies) between them. This can be seen as a modeling characteristic supporting *rigidity* notion of ontology based modeling.

The capabilities are enriched by a logic paradigm turns a perdurantist model into a model which has cognitive capability and can manage relations dynamically.

Meta-knowledge, higher-order rules are vital for both agents and ontology based modelling. To enable the development of high-level easy to configure agent behavior, it is important to provide agents with the means to reason on their surrounding an environment using a generic reasoning mechanism and knowledge base. The agents must be able to analyze unexpected situations to dynamically adapt their behavior to achieve their personal goals [9]. This means agents must be capable of evaluating situations using abstract, higher-order rules.

3.1 Behavior Management by Reasoning

Semantic representation of an agent or a simulation model is represented by behavioral descriptions. Any interaction, in other words, event transitions between simulation entities and agents causes a set of

behavioral reactions either reactive or proactive. Each behavior taken consist of a series of actions connected each other logically. In addition to the behavior that is activated by an event, activated or cancelled by a condition, in generally, managed behavior after a series of reasoning process is a good example of shared semantic. An example for activating a behavior as a result of reasoning is given in Section 4.

An agent can behave in different ways depending on the situation it is in. In this sense, behaviors can be categorized according to the well defined world views and each category that describe how the model behaves under a certain condition. The condition is defined as a decision model and it is used to activate or deactivate the related category. This is a very usefull property to be able to change a model world view in both design time and run time. It can be seen as a dynamic description.

3.2 Relation, Composition, and Aggregation

Both an agent and a simulation model can have relations with other objects. In AdSiF representation, the relations are categorized under three main headlines; 1) Logical, 2) Behavioral, and 3) Structural.

A logical relation consists of facts (position information of a missile time by time), higher-order descriptions (missile types, their behavioral patterns such as being in booster phase etc.), and predicates (decision about what an object is, or how to behave).

Behavioral relations that are established between two parties force a definite behavior set for both sides. In this sense, the relations define behavioral templates at different abstract levels. The relations making a behavior set activated for the entities at both sides of the relation. The behavior set being activated differentiates depending on the model type, entity abstraction level, and environment or entity state vector.

A relation named “*carries*” gets the carried object position related to the object “*carries*”. As seen in the example given at Section 4, while the relation “*Has*” dictates an action named “*Af_SetPosition*” to the entity “*Missile*” and a behavior set to the entity “*F16*” “*LockingPosition*”, the same relation dictates different behavior sets for both sides sharing the relation in case of “*Missile has Seeker*”.

A structural relation is related with components of entites and it describes a unity of ontology.

4 Ontology Example

A simple defense example is chosen to show how the concepts are implemented.

In Figure 1, the relations between objects are seen. The relation between different objects with the same name activates different behavior and action sets. Each relation states a meaning depending on the behavior space of models that are the relation constituted between. This is also valid for the abstraction level of models. The behavioral description of the relations is depicted in Figure 2 for the relation *use* between “F16” and “Missile”. The relation rule is applicable all type of F16, missiles, and any type of models derived from these models. In the plane domain of Figure 2 it is seen what behaviors are executed in both activation that means constituting the relation and passivation phase that means breaking it. All ontology commitment is not given here, a good example regarding with missile phase and dynamic modeling can seen in [10].

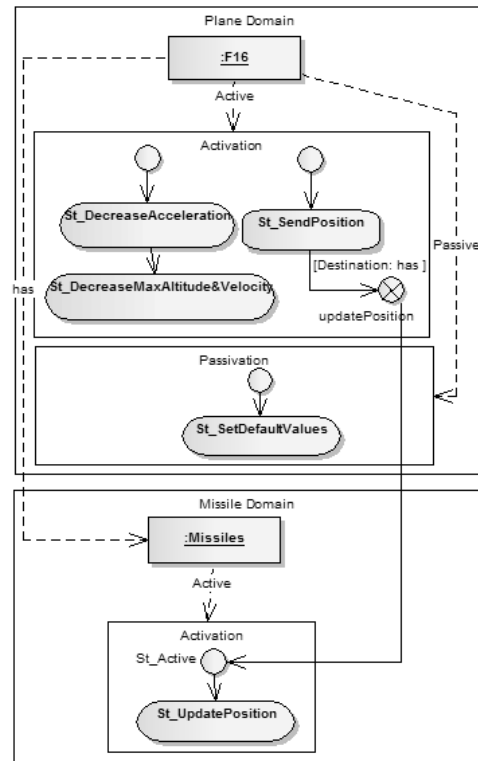


Figure 2: Behavioral Descriptions

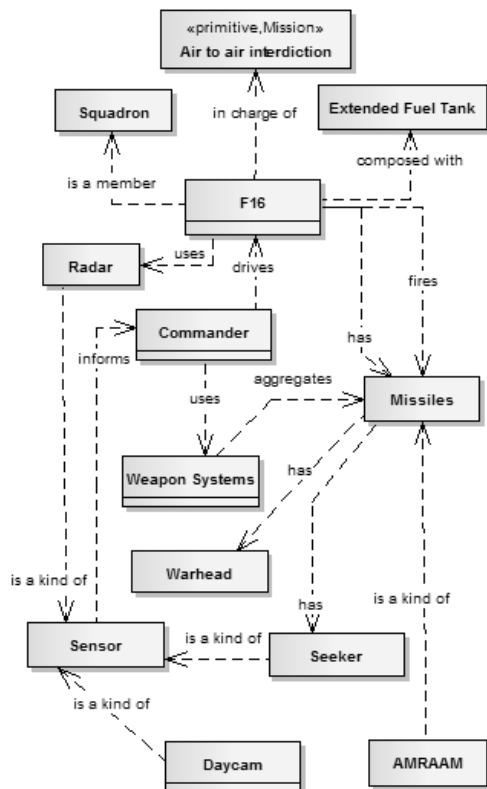


Figure 1: Relations between Objects

In Figure 3, an ontological description is given from the more abstract class level up to instance level. The second part of the figure shows a set of predicates inferring what the target is and what kind of missile is to be fired using the facts given in the third part and populated by sensory data. The sensors providing detection information send their detections to commanders, who drive F16, using the relation “*informs*”. The decision triggers the behavior seen in Figure 4. The behavior selects a missile as a result of an inference using target information received from event named *detection* and a set of predicates, facts, and, ontological descriptions given in the knowledgebase shown in Figure 3.

5 Discussions

Logic programming and modeling paradigms enrich ontologies by giving inference capability, keeping the information as time framed and allowing it expired, defining relations between objects and conditions on them. Apart from inferencing, logic programming combined with state oriented programming allows modelers to model domain information at meta-level. Each meta-level system model keeps a model family rather than a specific implementation model.

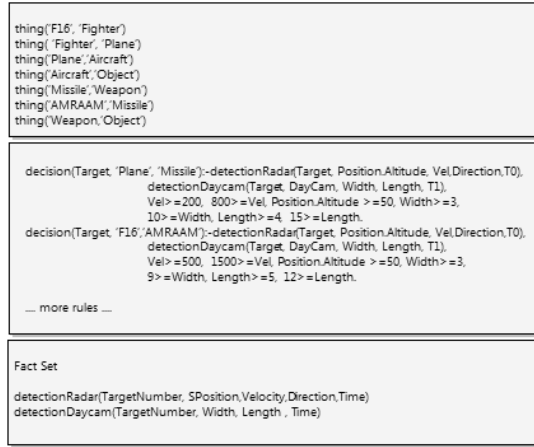


Figure 3: A portion of Knowledgebase

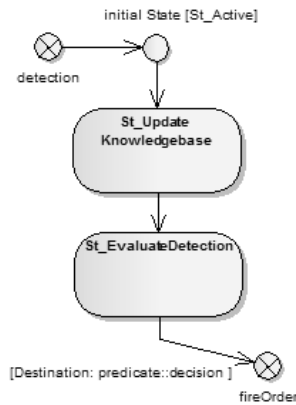


Figure 4: Driving Behavior by Inferences

In this study, it is pointed out that how to use concepts and axioms defined as logical premises and relations between higher-order entity descriptions are used to combine in a behavior management structure.

AdSiF programming approach and state oriented paradigm that is main programming paradigm expands a perdurantist modeling approach to reasoning capable model by logic programming paradigm and ontology based modeling world view.

Logical premises define model structures at different abstraction level of domain information. Relations provide indirect interaction based on entity description not direct interaction with object and also straight forward behavioral descriptions.

Since AdSiF provides a well defined grammar, it also provides verification check for simulation scenarios.

6 References

- [1] M. F. Hocaoglu, "AdSiF: Agent Driven Simulation Framework," *Hunstv. Simul. Conf. -HSC2005*, 2005.
- [2] M. F. Hocaoglu, "ETSİS: Etmen Tabanlı Simülasyon Sistemi AdSiF: Agent driven Simulation Framework Abstract," 2011.
- [3] A. G. Bruzzone, R. Mosca, R. Revetria, E. Bocca, and E. Briano, "Agent Directed HLA Simulation for Complex Supply Chain Modeling," *Simulation*, vol. 81, no. 9, pp. 647–655, 2005.
- [4] M. M. Al-Debei, M. M. Al Asswad, S. de Cesar, and M. Lycett, "Conceptual Modelling and the Quality of Ontologies: Endurantism Vs Perdurantism," *Int. J. Database Manag. Syst.*, vol. 4, no. 3, pp. 1–19, Jun. 2012.
- [5] N. Guarino, D. Oberle, and S. Staab, "What is an Ontology?," pp. 1–17, 2009.
- [6] N. Guarino and C. Welty, "Towards a methodology for ontology based model engineering."
- [7] S. Youn, D. Mcleod, and D. Mcleod, "Ontology Development Tools for Ontology-Based Knowledge Management Ontology Development Tools for," 2006.
- [8] T. Özacar, Ö. Öztürk, and M. O. Ünahır, "ANEMONE: An environment for modular ontology development," *Data Knowl. Eng.*, vol. 70, no. 6, pp. 504–526, Jun. 2011.
- [9] F. Béhé, S. Galland, N. Gaud, C. Nicolle, and A. Koukam, "An ontology-based metamodel for multiagent-based simulations," *Simul. Model. Pract. Theory*, vol. 40, pp. 64–85, Jan. 2014.
- [10] U. Durak, H. Oğuztüzün, and S. K. İDER, "Ontology-Based Domain Engineering for Trajectory Simulation Reuse," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 19, no. 08, pp. 1109–1129, Dec. 2009.

Optimierung von Supply Chains bei saisonalen Schwankungen

Hans-Peter Barbey
Fachhochschule Bielefeld
Wilhelm-Bertelsmann-Str. 10
33602 Bielefeld
hans-peter.barbey@fh-bielefeld.de

Diese Simulation ist das einfachste Modell einer Supply Chain mit 4 gleichen, unabhängigen Einzelunternehmen. Ziel ist es, die Lagerbestandsschwankungen zu minimieren. Dazu wird mit Hilfe eines VBA-Programms ein Regler implementiert, der die Bestellungen als Summe von Sprungfunktionen interpretiert. Die aus den Sprungfunktionen sich ergebenden permanenten Abweichungen vom Sollwert werden durch den Regler über eine definierte Kompensationszeit ausgeglichen. Betrachtet wird in dieser Simulationsstudie ein saisonaler Trend, wobei dessen Periodendauer als Variable eingeht. Für lange Periodendauern lassen sich die Lagerbestandsschwankungen sehr gering halten. Allerdings verschlechtert sich die Regelgüte mit zunehmendem Abstand vom Kunden. Bei kürzer werdenden Periodendauern kommt das System erst in die Resonanz und anschließend versagt der Regler aufgrund seiner Trägheit. Daher erweist es sich für kürzere Periodendauern als sinnvoll, die vom Kunden erhaltene Bestellung in der gleichen Menge an den Unterlieferanten weiterzugeben. Für sehr kurze Periodendauern sollte die Eingangsbestellung eher als Störung betrachtet werden, die nicht geregelt und nur durch den Lagerbestand des kundennächsten Unternehmens kompensiert wird. Dieses Konzept liefert dann für die gesamte Supply Chain das beste Ergebnis. In der Realität bedeutet dies eine Kooperation zwischen den Einzelunternehmen, da hier von der eingangs aufgestellten Forderung nach gleichen Unternehmen abgewichen wurde.

1 Das dynamische Verhalten von Supply Chains

Das dynamische Verhalten insbesondere des Materialflusses in einer Supply Chain wird letztlich bestimmt durch das Bestellverhalten des Kunden am Ende einer Supply Chain. Primäres Ziel ist es, den Kundenwunsch vollständig zu erfüllen. Andererseits wollen alle Unternehmen einer Supply Chain ihre Lagerbestände dennoch möglichst niedrig halten und Produktionen nahezu gleichmäßig auslasten. Werden nun reale Supply Chains untersucht, so lässt sich häufig trotz nur einer geringen Änderung der Kundenbestellung eine enorme Schwankung der Lagerbestände am Anfang der Supply Chain beobachten. Dies wird erstmalig von Forrester [1] als bullwhip Effekt definiert. Bis heute wird der bullwhip Effekt und seine Entstehung meist nur sehr allgemein beschrieben [2] [3]. Gudehus [4] fasst die wesentlichen Gründe für die Entstehung des bullwhip Effektes zusammen:

- Unabhängige Bestellungen der einzelnen Unternehmen einer Supply Chain
- Änderungen in der Kundenbestellung
- Falsche Bestellstrategie in Notfällen
- Spekulative Bestellungen oder Verkaufsaktionen

Auch wenn die einzelnen Unternehmen einer Supply Chain kooperieren und Bestellinformationen austauschen wird bezweifelt, dass sich der bullwhip Effekt vollständig vermeiden lässt [5].

2 Modellbildung

Ein sehr einfaches Modell zur Simulation einer Supply Chain wird in [6] vorgestellt. Als Zielgröße wird nur die Schwankung des Lagerbestandes in allen Unternehmen einer Supply Chain betrachtet. Die Schwankung des Lagerbestandes soll durch ein geeignetes Bestellverhalten minimiert werden.

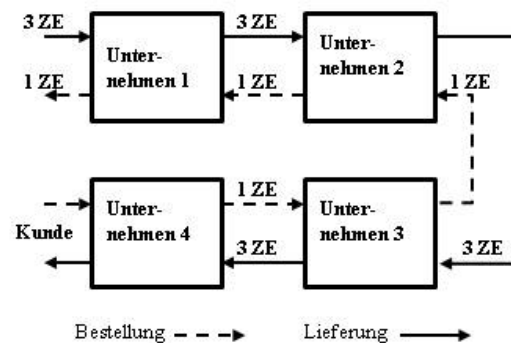


Abbildung 1. Blockdarstellung einer Supply Chain

Die Supply Chain besteht aus 4 identischen Unternehmen (Abb.1). Die Bestellung von Waren dauert 1 Zeiteinheit (ZE). Die Lieferung erfolgt in 3 Zeiteinheiten. Damit ergibt sich eine Wiederbeschaffungszeit (WBZ) von 4 Zeiteinheiten. Dies ist die theoretisch minimal möglich Zeit zum Ausgleich einer Störung in einem Unternehmen. Für die gesamte Supply Chain verlängert sich diese Zeit entsprechend der Länge der Kette (Abb.3).

Die einzige Möglichkeit der Kompensation besteht über einen geeigneten Algorithmus (s. Kap.3) die Bestellmenge festzulegen. Folgender Ablauf findet in jedem Unternehmen statt (Abb.2):



Abbildung 2. Ablaufdiagramm eines Unternehmens

Die ersten vier Blöcke dienen nur zur Aktualisierung der Systemgrößen. Der eigentliche Regelvorgang erfolgt durch die Ermittlung der Bestellmenge. Dabei wird streng getaktet der Lagerbestand auf seinen Sollwert gebracht. Dieser Ablauf entspricht als Bestellrhythmusverfahren einer (t,S)-Politik. Wesentlicher Unterschied zu einer realen Supply Chain besteht darin, dass in jeder Zeiteinheit Ware geliefert sowie bestellt wird. Normalerweise sind die Bestellabstände wesentlich länger und liegen deutlich über der Wiederbeschaffungszeit. Der Grund liegt darin, die durch lange Bestellabstände hervorgerufene Bestandschwankung im Lager zu eliminieren um dadurch den bullwhip Effekt besser zu erkennen. Es wird damit ein quasikontinuierlicher Ablauf erreicht. Da das Modell einerseits streng zeitgetaktet und durch sehr wenige Parameter definiert ist und andererseits einen frei zu programmierenden Regelalgorithmus zur Bestimmung der Bestellmenge enthält, kommt zur Umsetzung auch kein ereignisdiskretes Simulations-

tool in Frage. Das Modell wurde daher als VBA-Programm umgesetzt.

3 Strategien zum Ausgleich von Schwankungen

Geht man von dem theoretischen Fall aus, dass die Wiederbeschaffungszeit gleich 0 ist, dann ist die optimale Strategie zur Regelung des Lagerbestandes: Bestellausgang=Bestelleingang. Wird diese Strategie auf das beschriebene Modell, welches eine definierte Wiederbeschaffungszeit hat, angewendet, dann ergibt sich folgender Lagerbestand (Abb. 3). Um möglichst vom Simulationsmodell unabhängige Aussagen treffen zu können, wird in allen Darstellungen mit einer bezogenen Zeit gearbeitet. Als Bezugsgröße wird die Wiederbeschaffungszeit (WBZ) genommen.

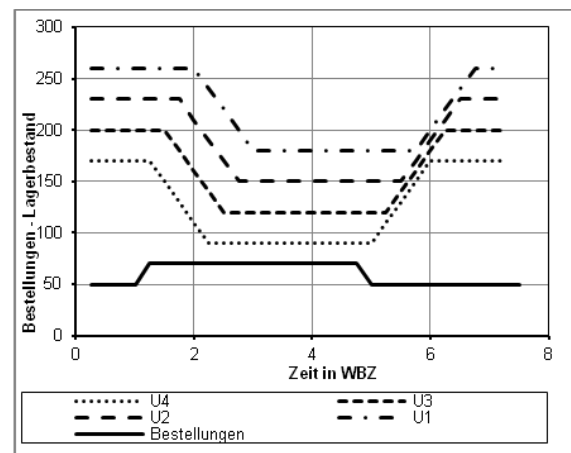


Abbildung 3. Lagerbestand mit aus=ein Strategie

Erhöht sich die Bestellung sprunghaft, dann muss die Lieferung so lange vom Lager gedeckt werden, bis nach Ablauf der Wiederbeschaffungszeit auch ein erhöhter Lagerzugang erfolgt. Bei dieser Regelungsstrategie bleibt eine permanente Lagerabweichung bestehen. Wird die Bestellung sprunghaft reduziert, geschieht das Ganze in umgekehrter Weise. Das Verhalten aller Unternehmen der Supply Chain ist gleich, jedoch zeitversetzt um die Bestellzeit.

Wird nun angenommen die Bestelländerung ist von Dauer, dann stellt sich die Notwendigkeit, den Lagerbestand wieder auf den Sollwert zu bringen. Die entstandene Lagerdifferenz lässt sich leicht aus der Bestelländerung und der Wiederbeschaffungszeit berechnen. Zur Kompensation dieser Differenz wer-

den als willkürliche Auswahl folgende Strategien betrachtet:

- Einmalige Zusatzbestellung der Differenzmenge
- Konstante Zusatzbestellungen über mehrere Perioden
- Linear ansteigende und fallende Zusatzbestellungen über mehrere Perioden

Die einmalige Zusatzbestellung wäre für das kunden nächste Unternehmen noch eine mögliche Strategie. Sie versagt aber bei den folgenden Unternehmen, da sich hier extreme Schwankungen sowohl in den Bestellungen als auch in den Lagerbeständen ergeben. Daher wird diese Strategie auch in der Simulation nicht weiter betrachtet.

Bei der nächsten Strategie erfolgt der Ausgleich des Lagers durch konstante Zusatzbestellungen über mehrere Perioden verteilt (Abb. 4). Der Ausgleich der Lagerdifferenz soll hier in der vierfachen WBZ erfolgen.

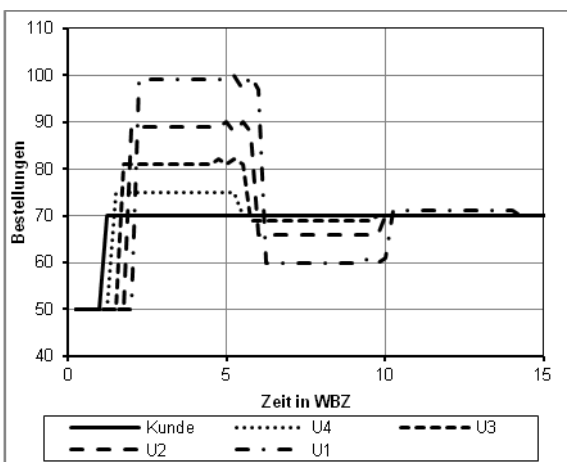


Abbildung 4. Konstante Bestellungen über vierfache WBZ

Bei diesem Regelverhalten ist es nur dem kunden nächsten Unternehmen möglich, in der vierfachen WBZ einen vollständigen Lagerausgleich vorzunehmen. Für alle anderen Unternehmen erfolgt nochmals eine sprunghafte Änderung der Bestellung. Damit verdoppelt sich die Zeit zum Ausgleich (Abb. 5). Der Ausgleich in der dreifachen Zeit bei U1 beruht nur auf Rundungseffekten der diskreten Simulation.

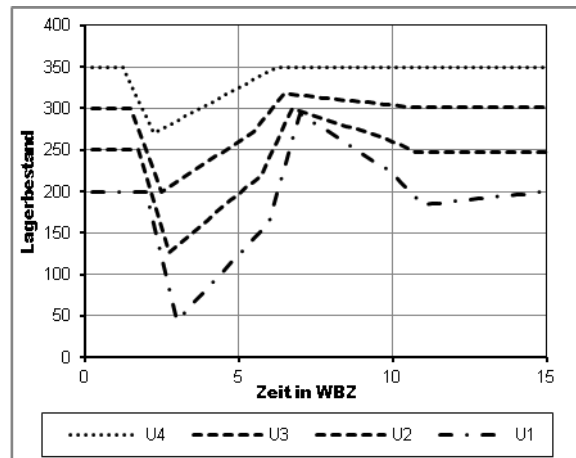


Abbildung 5. Lagerbestand bei konstantem Ausgleich

Um die sprunghaften Änderungen in den Bestellungen zu vermeiden, wird die Lagerkompensation über einen linearen Anstieg und Abfall modelliert (Abb. 6). Der lineare Anstieg lässt sich allerdings nur für das kunden nächste Unternehmen realisieren. Alle anderen Unternehmen haben einen überproportionalen Anstieg der Bestellungen, da sie nicht eine sprunghafte Änderung der Bestellungen regeln, sondern einen linearen bzw. überproportionalen Anstieg ihres Vorgängers.

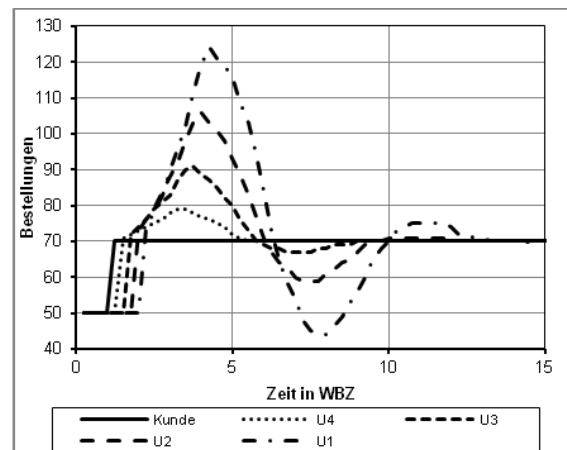


Abbildung 6. Bestellungen mit linearem Anstieg und Abfall über die vierfache WBZ

Es ergibt sich ein relativ ähnlicher Verlauf zu der Regelung mit konstanten Zusatzbestellungen. Lediglich die Maxima und Minima der sowohl der Bestellungen als auch der Lagerbestände fallen etwas extremer aus (Abb. 7).

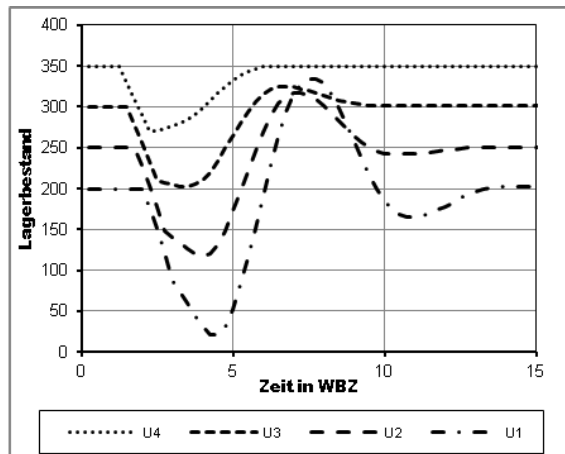


Abbildung 7. Lagerbestand bei linearem Anstieg und Abfall der Bestellungen

Damit ist es vermutlich für die folgende Simulation nicht von großer Bedeutung, welcher Regelungsstrategie verwendet wird. Im Hinblick auf die in dieser Simulationstudie noch nicht betrachteten zufälligen Schwankungen, wird diese Regelungsstrategie verwendet.

4 Saisonaler Trend

Ein saisonaler Trend wird für die Simulation durch eine sinus-Funktion abgebildet. Das Verhalten der Supply Chain wird hinsichtlich zweier Parameter untersucht. Einerseits wird die Periodendauer des sinus variiert. Andererseits wird die Zeit zum Ausgleich der Lagerabweichung verändert. Als Regelalgorithmus wird der lineare Anstieg und Abfall eingesetzt.

4.1 Einfluss der Periodendauer

Die bisher verwendete sprunghafte Änderung der Kundenbestellung diente nur zur Festlegung eines geeigneten Regelverhaltens. In der Regelungstechnik ist es üblich das Systemverhalten über der Frequenz bzw. Periodendauer darzustellen. Dieses Vorgehen wird auch hier angewendet. Für die Regelung wird eine Kompensationszeit der Lagerabweichung von 8WBZ gewählt. Damit ergibt sich für die einzelnen Unternehmen folgendes Systemverhalten (Abb. 8).

Um die Güte der Regelung zu beurteilen, wird als einfachste Strategie „Bestellausgang=Bestelleingang“ zu Grunde gelegt. Bei dieser Strategie wird sich der harmonische Verlauf der Bestellung auch im Lagerbestand widerspiegeln. Da die Bestellung einfach in

das nächste Unternehmen weitergegeben wird, ist auch das Verhalten in allen Unternehmen identisch. Es ergibt sich über alle Periodendauern für alle Unternehmen eine konstante Lagerbestandsschwankung von 80 Einheiten (Abb.8).

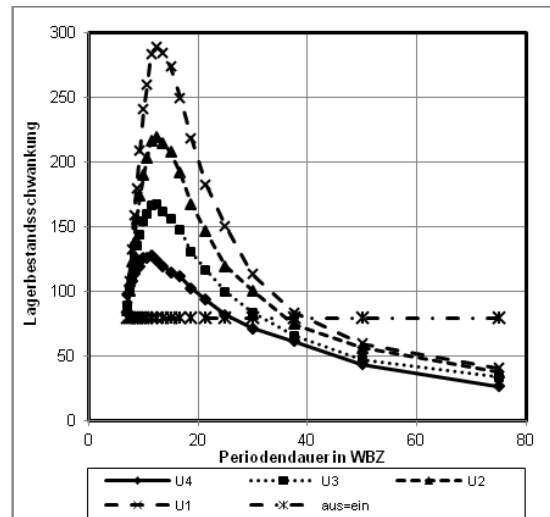


Abbildung 8. Systemverhalten bei einer Kompensationszeit von 8WBZ

Wird die Lagerbestandsschwankung nach dem o.g. Algorithmus geregelt, fallen sie umso geringer aus, je dichter das Unternehmen am Kunden ist. Für einen sehr langwelligen Trend ergeben sich deutlich geringere Lagerschwankungen im Vergleich zur aus=ein Strategie. Für das Unternehmen 1 ergibt sich unterhalb von etwa 40WBZ und für das Unternehmen 4 unterhalb von etwa 25WBZ ein wesentlich schlechteres Regelverhalten. Die Lagerbestandsschwankungen steigen nun überproportional an. Das System kommt gewissermaßen in die Resonanz. Nach Überschreiten eines Maximums fallen die Lagerbestandsschwankungen stark ab. Hier verhält sich der Regler einfach zu träge. Für diesen Simulationslauf wurde eine Kompensationszeit von 8WBZ angenommen. Kommt die Periodendauer der Bestellungen in diese Größenordnung, dann kann der Regler nicht mehr reagieren. Letztlich wird nur noch den Mittelwert der Bestellung weitergegeben.

4.2 Einfluss der Ausgleichszeit

In einem weiteren Simulationslauf wurde untersucht, welchen Einfluss die Ausgleichszeit des Reglers auf die Schwankung des Lagerbestandes hat. Je kürzer die Ausgleichszeit gewählt wird, desto schneller kann der Regler auf Schwankungen reagieren. Für diesen

Simulationslauf wurde eine Periodendauer von 75 WBZ gewählt. Diese Periodendauer ermöglicht noch eine gute Regelung des Lagerbestandes (Abb.8).

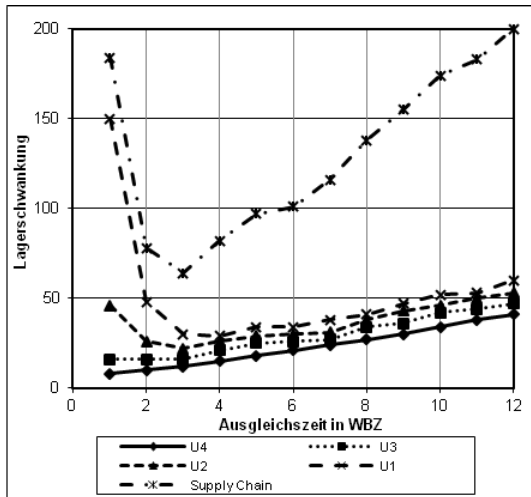


Abbildung 9. Einfluss der Ausgleichszeit bei einer Periodendauer von 75WBZ

Es zeigt sich, dass für die beiden nächst gelegenen Unternehmen zum Kunden die Lagerbestandschwankungen mit abnehmender Ausgleichszeit sich verringern (Abb. 9). Für die anderen Unternehmen steigen die Lagerbestandschwankungen bei kurzen Ausgleichszeiten wieder stark an. Hier kann von einem typischen bullwhip Effekt gesprochen werden.

Betrachtet man nun die Schwankung der Schwankung des Gesamtbestandes der Supply Chain als Addition aus den einzelnen Lagerbestandschwankungen, dann ergibt sich in Abhängigkeit von der Ausgleichszeit ein ausgeprägtes Minimum. Für die gesamte Supply Chain lässt sich damit eine optimale Reglereinstellung bei etwa 4WBZ finden. Ein derartiger optimaler Wert lässt sich in realen Supply Chains höchstens durch eine enge Kooperation erreichen. Da die kundenseitigen Unternehmen kein ausgeprägtes Minimum in der Lagerbestandschwankung haben, werden sie immer versuchen möglichst kurze Ausgleichszeiten zu realisieren. Eine Erhöhung der Ausgleichszeit und damit verbunden eine Erhöhung der Lagerbestandschwankung ist aus Sicht dieser Unternehmen widersinnig, obwohl es für den Gesamterfolg der Supply Chain günstig wäre.

4.3 Vergleich von Regelungsstrategien

Um den Gesamterfolg einer Supply Chain zu beurteilen, soll eine weitere Bestellstrategie betrachtet werden. Dazu wird der sinusförmige Verlauf der kundenseitigen Bestellung gewissermaßen als reine Störung aufgefasst. Das bedeutet, dass der Lagerbestand nicht mehr geregelt wird. Als Bestellung wird nur noch der Mittelwert des sinus an den Lieferanten weitergegeben. Das hat zur Folge, dass sämtliche Schwankungen im Lager des kundenseitigen Unternehmens aufgefangen werden. Die restlichen Unternehmen haben theoretisch keine Lagerbestandsschwankungen mehr. Insofern kann nun nicht mehr der Erfolg des Einzelunternehmens betrachtet werden, sondern nur noch der Gesamterfolg der Supply Chain (Abb. 10). Für die bisher angewendete Strategie einer festen Ausgleichszeit und der Strategie „aus=ein“ ergibt sich der Gesamterfolg als Summe aus den einzelnen Lagerbestandsabweichungen. Für die Strategie „Bestellung des Mittelwertes“ ist nur noch die Lagerbestandschwankung des kundennächsten Unternehmens relevant.

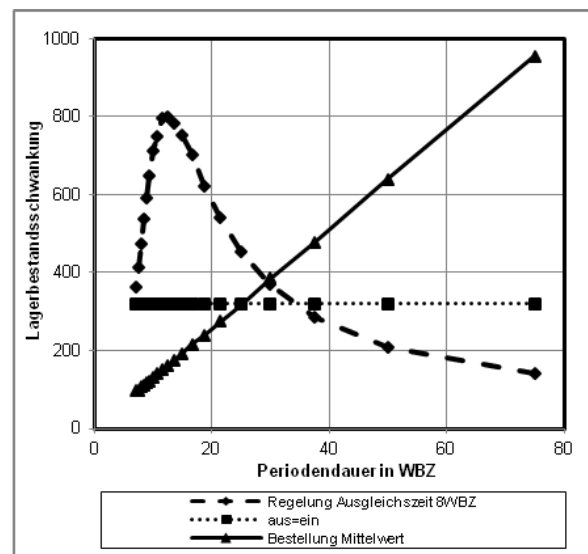


Abbildung 10. Vergleich von Regelungsstrategien

Damit ergibt sich folgende recht komplexe Situation: Bis zu einer Periodendauer von etwa 25WBZ ist es am sinnvollsten, die Kundenbestellung als reine Störung aufzufassen und immer den Mittelwert an den nächsten Lieferanten weiterzugeben. Die Lagerbestandsschwankung findet daher nur im kundennächsten Lager statt. Bis zu einer Periodendauer von 35WBZ ist die optimale Strategie „aus=ein“. Erst oberhalb dieser Periodendauer greift die Strategie mit

einer definierten Kompensationszeit für die Lagerverluste.

Ebenso wird deutlich, dass erst oberhalb von 25 WBZ als Periodendauer es sinnvoll ist, dass die Unternehmen einzeln agieren.

5 Schlussfolgerungen und Zusammenfassung

Diese Simulation ist eine mehr theoretische Untersuchung einer Supply Chain. Sie soll den Einfluss eines saisonalen Trends auf die einzelnen Unternehmen einer Supply Chain aufzeigen. Es ist bekannt, dass selbst kleine Änderungen in den Bestellungen eines Kunden zu extremen Schwankungen in den Lagerbeständen der Unternehmen einer Supply Chain führen können. In der Literatur wird dieses Verhalten als bullwhip Effekt beschrieben. Es wurde ein Modell mit 4 Unternehmen aufgestellt, das nur noch die Lagerbestände und die Bestellungen als variable Größen enthält. Zur Simulation der Kundenbestellungen wurde eine sinus-Funktion mit unterschiedlichen Periodendauern als Eingangssignal aufgeprägt. Zur Minimierung der Lagerbestandsschwankungen wurden für die Regelung die Bestellungen als Summe von Sprungfunktionen aufgefasst. Die aus den Sprungfunktionen resultierenden permanenten Lagerbestandsabweichungen wurden berechnet und über eine definierte Kompensationszeit ausgeglichen. Es zeigt sich, dass mit diesem Regelungskonzept alle Unternehmen unabhängig voneinander die Lagerbestandsschwankungen minimieren können, wobei sich die Regelungsgüte mit dem Abstand vom Kunden verschlechtert. Bei kürzer werdenden Periodendauern kommt das System gewissermaßen in die Resonanz, was zu extrem hohen Lagerbestandsschwankungen führt. Dies kann mit dem in der Literatur beschriebenen bullwhip Effekt gleichgesetzt werden. Soll im Bereich der Resonanzfrequenz geregelt werden stellt das Konzept Ausgangsbestellung gleich Eingangsbestellung die bessere Alternative dar. Damit können alle Unternehmen unabhängig voneinander über den gesamten Frequenzbereich eine gleichbleibende Regelungsgüte erzielen. Bei sehr kurzen Periodendauern ist es sinnvoll auf eine Regelung komplett zu verzichten und immer den Mittelwert zu bestellen. Damit sind die einzelnen Unternehmen keine gleichwertigen Einheiten mehr, denn die Lagerbestandsschwankungen finden nur noch im kundennächsten Unternehmen statt. Aus regelungstechnischer Sicht ist dies das

beste Konzept für kurze Periodendauern. Ob dies auch ein realistisches Konzept ist, darf noch bezweifelt werden, da schließlich auch das kundennächste Unternehmen am Gesamterfolg der Supply Chain teilhaben will. Dies setzt dann aber eine Kooperation in irgendeiner Form zwischen den Unternehmen voraus. Die Interpretationen und Schlussfolgerungen aus dieser Simulation beruhen ausschließlich auf der Betrachtung von Lagerbeständen. Da sich in einer Supply Chain auch die Produkte und deren Werte ändern, ist die eingangs aufgestellte Randbedingung von gleichen Unternehmen kritisch zu betrachten. Als alternative Zielgröße können auch die Lagerwerte betrachtet werden. Die Betrachtung dieser Zielgröße kann natürlich auch zu anderen Schlussfolgerungen führen. Diese Simulation stellt das einfachste Modell einer Supply Chain dar. Es bleibt zu klären, welche Zielgröße oder Zielgrößen für die Beurteilung einer Supply Chain die richtigen sind und welche Parameter in die Berechnung dieser Zielgrößen einfließen müssen.

6 References

- [1] Forrester, J.W.: *Industrial Dynamics: A major breakthrough for decision makers*. In: Harvard business review, 36(4), 1958.
- [2] Erlach, K.: *Wertstromdesign*, Springer Verlag Berlin Heidelberg, 2010.
- [3] Dickmann, P.: *Schlanker Materialfluss*, Springer Verlag Berlin Heidelberg, 2007.
- [4] Gudehus, T.: *Logistik*, Springer Verlag Berlin Heidelberg, 2005.
- [5] Bretzke, W.-R.: *Logistische Netzwerke*, Springer Verlag Berlin Heidelberg, 2008.
- [6] Barbey, H.-P.: *Dynamic Behavior of Supply Chains*. Proceedings 27. European Conference on Modelling and Simulation (ECMS), Alesund, 2013, 748-752.

Dimensionierung von Sicherheitsbeständen unter Einbeziehung kurzfristiger Bedarfsschwankungen in automobilen Zulieferbeziehungen

Tim Gruchmann¹, André Krischke²

¹Johnson Controls, Merscheider Straße 167, Solingen

²Hochschule München, Am Stadtpark 20, München

tim.gruchmann@jci.com

Within automotive supply chains schedule instability leads to inefficiencies at production processes of 1st tier suppliers and bears the risk of supply disruptions. Due to the market power of the OEM, 1st tier suppliers are not always able to influence the scheduling behavior of their customers. Therefore they need to find an efficient way to protect their production against short-term demand variations. The placement of safety stocks is a probate instrument to reach a high service level for the OEM when they are dimensioned effectively. Taking into consideration the scheduling behavior of the customer, safety stocks can be dimensioned more precisely. This paper presents an approach to setup safety stocks calculated on the OEM forecast reliability. Furthermore this approach is transferred in a simulation model and validated with company data. By applying the approach to the case study data relevant schedule instability related insights are obtained.

1 Einleitung

Unternehmen innerhalb von Wertschöpfungsketten tauschen regelmäßig Bedarfsinformationen miteinander aus, sodass jedes Mitglied über Änderungen in der Belieferung von Waren und Dienstleistungen, den geplanten Auslieferzeitpunkt und der geplanten Auslieferungsmenge betreffend, informiert ist [18]. In der Automobilindustrie existieren hierfür Lieferabrufsysteme, welche dem Lieferanten eine Bedarfsvorschau des Kunden für unterschiedliche Vorschauhorizonte geben und damit eine Grundlage für dessen Produktionsplanung bieten. Lieferabrufsysteme werden deshalb als vernetzte Systeme zur Abwicklung der Materialdisposition und Materialanlieferung zwischen Original Equipment Manufacturer (OEM) und Automobilzulieferer definiert [9].

Kurzfristige Bedarfsschwankungen in automobilen Lieferabrufsystemen resultieren aus der Anpassung von Plandaten durch den OEM. Anpassung bedeutet jedoch in diesem zeitlichen Horizont häufig nicht die Veränderung der Nachfrage am Markt bzw. die Stornierung von Aufträgen, sondern überwiegend die Verschiebung von Produktionsaufträgen in der Produktionsplanung des OEM zwischen einzelnen Perioden. Laut dem Verband der Automobilindustrie (VDA) liegen mögliche Ursachen von Bedarfsschwankungen begründet in Marktentwicklungen, in der Komplexität der Supply Chain (SC), in technischen oder qualitätsrelevanten Problemen sowie in

weiteren Restriktionen [21]. CHILDHOUSE, DISNEY und TOWILL (2008) nennen ähnliche Einflussgrößen. Für Automobilzulieferer ergibt sich eine erhöhte Volatilität innerhalb der kurzfristigen Produktionsplanung [3]. Hieraus resultiert eine beunruhigte Produktion, da der Lieferant nahezu sofort auf kurzfristige Bedarfsschwankungen des OEM reagieren muss, um lieferfähig zu bleiben. Eine solche beunruhigte Fertigung ist gekennzeichnet durch zusätzliche Rüstwechsel und nicht-optimale Produktionslose [7] und verursacht laut VDA Zusatzkosten von bis zu 5% des Umsatzes [21].

2 Stand der Forschung

Das in der Einleitung beschriebene Phänomen wird in der Literatur mit dem Begriff der Schedule Instability bezeichnet [3]. Unter Schedule Instability wird folglich eine unablässige Änderung des Produktionsplanes des OEM verstanden, welcher auf geänderte Rahmenbedingungen reagiert [15] und ständig Abweichungen zwischen den tatsächlichen Bedarfen und den geplanten Belieferungen verursacht [14]. Zur Beschreibung der Schedule Instability und deren Auswirkungen wurden in der jüngeren Vergangenheit mehrere Studien veröffentlicht. Genannt werden können beispielsweise LIKER und WU (2000), welche die Schedule Instability im Kurzfristbereich US-amerikanischer und japanischer OEMs vergleichend gegenübergestellt haben [13]. Für die europäische Automobilindustrie konnten CHILDHOUSE,

DISNEY und TOWILL (2008) durch die Untersuchung von automobilen Wertschöpfungsketten einen signifikanten Zusammenhang zwischen Bedarfsunsicherheit und Schedule Instability feststellen [3].

Im Kontext kurzfristiger Bedarfsschwankungen des OEM wurde weiterhin eine Studie des VDA in Kooperation mit der Universität Stuttgart unter dem Titel „Dynamik in automobilen Supply Chain Systemen – Treiber, Auswirkungen, Stellhebel“ durchgeführt. Hier gaben 66 Prozent der befragten Zulieferunternehmen an, von der Dynamik in automobilen Lieferabrufsystemen betroffen zu sein. Sie gaben zudem an, dass kurzfristige Planabweichungen im Bereich der bedarfssynchronen Produktion nur mit hohem Aufwand durch die Produktionsplanung aufgefangen werden können und schlimmstenfalls zum Abreißen der Lieferkette, einer sog. Disruptions, führen. Auf dieser Basis hat der VDA eine Empfehlung zur Vereinbarung von Schwankungsbreiten zwischen OEM und Lieferant ausgesprochen, wobei insbesondere im Kurzfristbereich eine Schwankungsbreite von 8% nicht überschritten werden sollte. [21]

Als ein Mittel zur Reduzierung der Schedule Instability sehen KADIPASAOGLU und SRIDHARAN (1995) u.a. den Aufbau eines Sicherheitsbestandes von Fertigmaterialien [8]. Stochastische (Sicherheits-)Bestandsmodelle beschäftigen sich hier insbesondere mit der Wahrscheinlichkeit, dass es zu Bestandsüberschreitungen kommt, wobei festgelegte Servicegradziele verfolgt werden [12]. In Kontext der Dimensionierung von Sicherheitsbeständen bei unsicherer Bedarfsnachfrage können u.a. die Publikationen von GRAVES und WILLEMS (1998, 2003) herangezogen werden. GRAVES und WILLEMS (2003) unterscheiden hierbei grundsätzlich zwei Ansätze, um Sicherheitsbestände und die mit ihnen verbundenen Kosten in mehrstufigen Netzwerken bei dezentraler Planung zu optimieren. Der erste Ansatz wird als Stochastic-Service Modell bezeichnet und geht von Schwankungen der Liefer- bzw. Servicezeit, hervorgerufen durch die Materialverfügbarkeit auf den einzelnen Stufen der SC, aus. Der zweite Ansatz wird hingegen als Guaranteed-Service Modell genannt und geht von einer Liefer- bzw. Servicezeit auf jeder Stufe der SC aus, welche immer erreicht wird. [22, 23]

Auch BOULAKSIL, FRANSOO und VAN HALM (2009) identifizieren zwei grundlegende Vorgehensweisen, um die Problemstellung der mehrstufigen

Dimensionierung von Beständen und Sicherheitsbeständen zu modellieren, um vordefinierte Servicegradziele durch eine zentrale Planung zu erreichen [2]. Die erste Kategorie basiert auf dem Prinzip der Modellierung der Nachfrage als stochastische Variable. Hierbei steht die Beantwortung der Fragestellung im Vordergrund, welche Bestandslevel auf den einzelnen Stufen der SC gehalten werden sollen und inwieweit diese Sicherheitsbestände beinhalten. Sicherheitsbestände sind somit Teil des Problems der Dimensionierung der Bestandshöhe. Die zweite Kategorie nutzt die Prinzipien mathematischer Programmierung, wobei die Nachfrage als Prognosewert für den zu betrachtenden Planungshorizont in die Modelle eingehen. BOULAKSIL, FRANSOO und VAN HALM (2009) führen hierfür simulationsbasierte Untersuchungen auf Basis mathematischer Planungsmodelle durch, wie dies in APS-Systemen bereits auch in der Praxis Anwendung findet. [2]

Die Ansätze von GRAVES und WILLEMS sowie von BOULAKSIL, FRANSOO und VAN HALM fokussieren die Unsicherheit des Bedarfs, berücksichtigen allerdings nicht den Einfluss kurzfristiger Schedule Instability auf operativer Planungsebene des Lieferanten, wie sie in automobilen SCs existiert. Eine weitergehende Betrachtung erscheint notwendig.

3 Forschungsmethodik

Die bisherigen Ausführungen zeigen, dass in automobilen SCs die Notwendigkeit besteht, kurzfristige Bedarfsschwankungen bei der Dimensionierung von Sicherheitsbeständen zu berücksichtigen, um eine weitergehende Beruhigung der Fertigung bei Zulieferer und Abnehmer zu erreichen. Um diesen dynamischen Rahmenbedingungen innerhalb automobiler Wertschöpfungsketten gerecht werden zu können, suchen OEMs und Zulieferunternehmen grundsätzlich nach Methoden unvorhersehbares Systemverhalten innerhalb der Planung zu berücksichtigen. Die klassischen einstufigen Produktionsplanungs- und Steuerungssysteme (PPS-Systeme) stoßen hier an ihre Grenzen, da bei häufigen Zustandsänderungen eine ständige Neuplanung notwendig wäre. Zur Beschreibung des dynamischen Verhaltens logistischer Systeme werden deshalb in der jüngeren Vergangenheit Simulationsstudien herangezogen, auf deren Basis Steuerungsmethoden abgeleitet werden können, um besser auf unvorhersehbare Störungen zu reagieren. [16]

3.1 Strukturierung der Simulationsstudie

Für die vorliegende Problemstellung gilt es innerhalb eines simulationsbasierten Ansatzes zu untersuchen, welche Anforderungen an die Dimensionierung von Sicherheitsbeständen unter Berücksichtigung kurzfristiger Bedarfsschwankungen gestellt werden und wie diese auf die logistische Effizienz wirken. In Anlehnung an den mehrstufigen Ansatz von LAW (2009) soll innerhalb der Simulationsstudie eine systematische Entscheidungsstruktur erarbeitet werden.

Im ersten Schritt wird zunächst das zu lösende Entscheidungsproblem formuliert, um dieses später in ein virtuelles Entscheidungsmodell überführen zu können. Im zweiten Schritt sind alle die Systemstruktur und deren Prozesse betreffende Informationen zu sammeln. Hierdurch soll die Grundlage geschaffen werden, um Modellparameter und die zugrunde liegende Wahrscheinlichkeitsverteilungen spezifizieren zu können. Insgesamt werden folglich im zweiten Schritt das methodische Gerüst der Simulationsstudie erarbeitet, indem die Ziele der Simulation, die Verfügbarkeit der Inputdaten und ggf. Kennzahlen zur Messung des Simulationsergebnisse vor Beginn der Studie zusammengeführt werden. Für die vorliegende Problemstellung dominieren im Schritt 2 die Bedarfs- und Abrufinformationen der OEMs, welche mithilfe von Verfahren zur Messung und Bewertung von Nachfrageschwankungen analysiert und somit das Bestellverhalten des OEM auf Artelebene quantifizierbar gemacht werden. Im dritten Schritt erfolgt die Übertragung der Entscheidungsstruktur in ein virtuelles Entscheidungsmodell, wobei weitergehende Experimentierpläne so festgelegt werden, dass eine statistisch abgesicherte Untersuchung aller relevanten Optionen sicherzustellen ist. Die Durchführung und Analyse der Ergebnisse der Simulationsstudie erfolgen auf Basis der konkreten Datenerhebung. Der dritte Schritt führt somit zur Maßnahmengestaltung, da simulationsbasiert Handlungsempfehlungen zur effizienteren Dimensionierung von Sicherheitsbeständen in der Produktion erarbeitet können, indem das quantifizierte Bestellverhalten auf Artelebene in die Auslegung der Sicherheitsbestände einfließt. [11]

3.2 Messung und Bewertung von Bedarfsschwankungen

Im Zusammenhang mit der Messung und Bewertung des Abrufverhaltens des OEM wurden in der Literatur

bis dato eine Reihe von Kennzahlen diskutiert. Genannt werden können u.a. Kennzahlen der VDA-Empfehlung 5009 oder Kennzahlen des „Dynamik-Monitoring für Zulieferunternehmen“ [1]. Für den vorliegenden Fokus ist die VDA-Empfehlung 5009 von Bedeutung, da sie das Ziel der „Entwicklung einer Formel zur systemgestützten Berechnung von Kennzahlen für Forecast Qualität inkl. Klassifizierung der Stabilität“ verfolgt [21]. Zu diesem Zweck wurden die Kennzahlen Forecast Qualität (FQ) und Tracking Signal (TS) entwickelt, die gemeinsam eine Aussage über die Vorhersagegüte des OEM liefern.

Die FQ dient der Berechnung der mittleren relativen Bedarfsschwankung in Bezug auf einen bestimmten Bedarfszeitpunkt. Sie gibt somit an, wie stark die einzelnen Vorhersagen für einen gewissen Zeitpunkt auf die tatsächliche Abgangsmenge im Mittel geschwankt haben. Berechnet wird die Kennzahl, indem jede einzelne Abweichung zwischen den vorhergesagten Bedarfen P_i und der tatsächlich abgegangenen Menge zum Referenzdatum R ermittelt wird. Die Beträge dieser Differenzen werden anschließend summiert und durch das Produkt aus dem Referenzbedarf und der Anzahl der verwendeten Vorhersagewerte dividiert. Die Berechnung mithilfe des Betrags führt dazu, dass sowohl positive als auch negative Abweichungen gleichermaßen in die Kennzahl einfließen und die Kennzahl nicht negativ werden kann. Der beschriebene Algorithmus der FQ ist in der Literatur auch unter der Bezeichnung „Mean Absolute Percentage Error“ (MAPE) oder „Mean Absolute Percentage Deviation“ (MAPD) bekannt. [4, 21]

$$FQ = \frac{\sum_{i=1}^n |P_i - R|}{R \cdot n} \geq 0 \quad (1)$$

Aussagekräftig wird die FQ jedoch erst in Kombination mit der Kennzahl TS. Das TS gibt Auskunft darüber, ob die Vorhersagen überwiegend über- oder unterprognostiziert waren. Für die Berechnung des TS werden zunächst sämtliche Abweichungen zwischen vorhergesagten Bedarfen P_i und dem tatsächlich abgegangenen Bedarf zu einem Referenzdatum R berechnet. Die Summe dieser Differenzen wird anschließend durch die Summe der Beträge dieser Differenzen dividiert. [21]

$$TS = \frac{\sum_{i=1}^n (P_i - R)}{\sum_{i=1}^n |P_i - R|}, \text{ d. h. } [-1 \leq TS \leq +1] \quad (2)$$

Je weiter die Ausprägung der Kennzahl von 0 entfernt liegt, desto stärker ist die Über- bzw. Unterschätzung. Über- bzw. Unterschätzung können sich hierbei kompensieren. [21]

3.3 Algorithmus zur Dimensionierung von Sicherheitsbeständen

Anhand der gewonnenen Informationen aus der Bewertung des Abrufverhaltens des OEMs soll die konkrete Soll-Höhe an Sicherheitsbeständen ermittelt und diese Ergebnisse innerhalb der Materialbedarfsplanung (vor allem Losgrößenplanung) mit einbezogen werden. Hierbei erfolgt die Verdichtung der Simulationsstudie auf die stochastische Dimensionierung von Sicherheitsbeständen in Abhängigkeit des Servicegrades, da die Ermittlung konstanter Sicherheitsbestände auf Basis von Erwartungswerten keine ausreichende Berücksichtigung von kurzfristigen Nachfrageschwankungen gewährleistet. Die VDA-Kennzahl FQ stellt hierbei eine geeignete Heuristik zur Schätzung der Fehlervarianz der Bedarfsprognose dar [20], da als Kennzahl dem MAPE bzw. MAPD entspricht. Entsprechend würde der Algorithmus zur Ermittlung der Losgrößen in einem ausgewählten Losgrößenmodell um die stochastische Sicherheitsbestandsdimensionierung ergänzt werden, sodass eine Modifikation der optimalen Produktionslosgröße an die Bedingungen kurzfristiger Nachfrageschwankungen stattfindet.

SCHMIDT, HARTMANN und NYHUIS (2012) stellen einen simulationsbasierten Vergleich mathematisch-stochastische Methoden zur Dimensionierung von Sicherheitsbeständen auf. Das von Ihnen beschriebene Grundmodell zur Auslegung von Sicherheitsbeständen (SS) bei unsicheren Erwartungen in Abhängigkeit vom β -Servicegrad und unter Verwendung der Funktion Φ der Normalverteilung soll im Folgenden genutzt werden. Der β -Servicegrad misst hierbei den erwarteten Anteil der Nachfrage, die in der Periode befriedigt werden kann. Zudem wird die stochastische Größe aus dem Zusammenhang $p = \Phi(\beta)$ ermittelt [10]. Durch die vorhergehende Festsetzung des Servicegrades lässt sich die Höhe des Sicherheitsbestandes aus der Multiplikation von $\Phi^{-1}(\beta)$ mit der Standardabweichung σ bestimmen [5]. Zur Schätzung von σ kommt entsprechend die VDA-Kennzahl FQ zur Anwendung. [17]

$$SS = \Phi^{-1}(\beta) * FQ \quad (3)$$

3.4 Datenbasis der Simulationsstudie

Die Datengrundlage der Simulationsstudie wurde bei der Firma REHAU AG + Co (kurz: REHAU) erhoben. REHAU ist ein mittelständisches Unternehmen der chemischen Automobilzulieferindustrie und führt weltweit als 1st tier Just-in-Time-/ Just-in-Sequence-Belieferungen von Stoßfänger-Modulen für automobile OEMs durch. Zur Eingrenzung der Stichprobe für das Produktsegment Stoßfänger wurden mithilfe einer ABC-Analyse die drei Kunden AUDI, VW und Daimler identifiziert. Zudem wurde bei der Auswahl des Lieferspektrums auf Ausgewogenheit zwischen Groß- und Kleinserien geachtet. Weiterhin wurden bei den ausgewählten Fahrzeugbaureihen alle Farbvarianten berücksichtigt, sodass sowohl „Lowrunner“-Varianten mit einer relativ niedrigen Nachfrage, als auch die „Highrunner“-Varianten mit einer relativ hohen Nachfrage in die Betrachtung mit eingeflossen sind. Die Datenerhebung wurde bei drei deutschen Produktionswerken von REHAU für 199 Artikel verschiedener Baureihen durchgeführt. Die folgende Tabelle 1 gibt eine Übersicht der betrachteten Artikel.

REHAU Produktionswerk	Zeitraum	Anzahl Artikel pro Kunde
DE- Feuchtwangen	09.08.2011- 14.10.2011	AUDI: 37 VW: 18 Daimler: 84
DE-Viechtach	09.08.2011- 14.10.2011	AUDI: 37
DE-Brake	09.08.2011- 14.10.2011	Daimler: 23

Tabelle 1. Artikelauswahl Stichprobe

4 Ergebnisse der Simulationsstudie

Innerhalb der Simulationsstudie unter Verwendung der Echt Daten wurden zwei Schwerpunkte gelegt. Zunächst wurde mithilfe der deskriptiven Statistik die Wahrscheinlichkeitsverteilung der Nachfrage für den Anwendungsfall getestet, um Aussagen über die Validität der Normalverteilung als Annahme für den Praxisfall zu erhalten. Im zweiten Schritt wurden über die Zählung der Häufigkeit negativer Bestandsverläufe aufgrund von kurzfristigen Nachfrageschwankungen der erreichte Servicegrad betrachtet, um das notwendige Niveau zur Auslegung der Sicherheitsbestände simulationsbasiert ermitteln zu können. Hier-

bei wurde weiterhin über die Anpassung des Prognosebedarfs mithilfe des TS ermittelt, ob es im Betrachtungszeitraum zu einem Lieferabriss gekommen wäre und somit zu einer Beunruhigung der Fertigung infolge der simulierten Bestandsdimensionierung geführt hätte.

Der erste Schwerpunkt der Datenanalyse befasst sich mit der beschreibenden bzw. deskriptiven Statistik und deren Übertragung auf den Anwendungsfall. Abbildung 1 gibt einen Überblick über die Lage- und Streumaße der Kennzahl FQ.

Descriptive Statistics: FQ									
Variable	N	N*	Mean	SE Mean	StDev	Minimum	Q1	Median	Q3
FQ	155	0	0,4910	0,0233	0,2896	0,1373	0,2929	0,3979	0,6673
Variable	Maximum								
FQ	1,4531								

Abbildung 1. Lage- und Streumaße Kennzahlen FQ

Weiterhin wird das Histogramm für die ermittelten Kennzahlen der FQ der Datenerhebung in Abbildung 2 dargestellt.

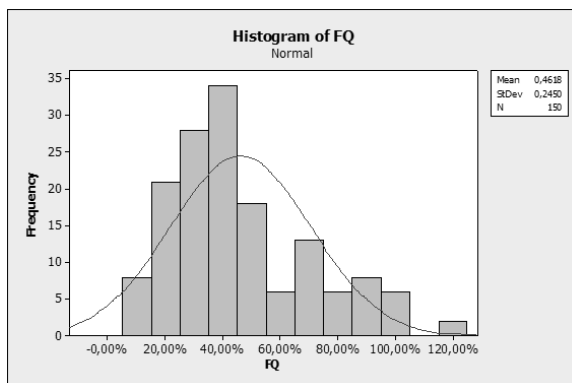


Abbildung 2. Histogramm Kennzahl FQ

Das Histogramm zeigt, dass die Verteilung der Merkmalsausprägungen der ermittelten Kennzahlen FQ im Anwendungsfall rein optisch eine gewisse Linksschiefe aufweist, sodass anstatt der Normalverteilung auch eine andere Verteilung für die Abbildung der Wahrscheinlichkeitsverteilung zur Anwendung kommen könnte. Die gesetzte Modellvoraussetzung einer normalverteilten Schwankungsbreite muss deshalb für den Anwendungsfall getestet werden. Hierfür wird ein Probability Plot eingesetzt, welchen Abbildung 3 zeigt. Es gibt an, dass mit einem P-Wert größer 0,005 zunächst keine Normalverteilung vorliegt. Es zeigt sich insgesamt, dass für die Anwendung der Methode eine Untersuchung der Wahrscheinlichkeits-

verteilung im Vorfeld und somit eine Anpassung an die im Anwendungsfall vorliegende Wahrscheinlichkeitsverteilung notwendig ist. TEMPELMEIER (2006) betont in diesem Zusammenhang, dass eine normalverteilte Nachfrage nicht ungeprüft für alle Artikel einer Unternehmung unterstellt werden sollte. Bei besonders starken Nachfrageschwankungen könnte die Gamma-Verteilung einen genaueren Ansatz zur Abbildung des Prognosefehlers bieten und hierdurch Fehler bei der Auslegung von Sicherheitsbeständen vermieden [19].

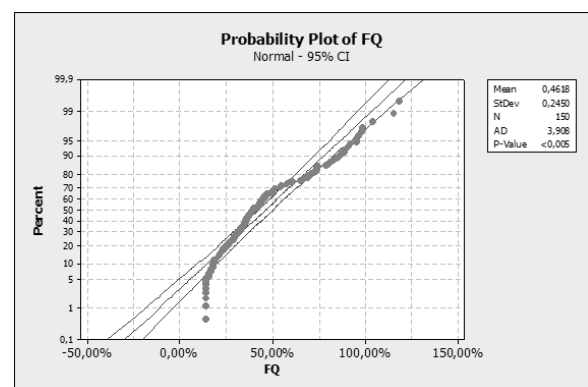


Abbildung 3. Probability Plot Kennzahlen FQ

Im zweiten Schwerpunkt wurde das Vorgehen zur Dimensionierung von Sicherheitsbeständen durch Multiplikation des Sicherheitsfaktors $\Phi^{-1}(\alpha)$ mit der Standardabweichung σ für die erhobene Datengrundlage angewendet, indem die Sicherheitsbestände auf Artekebene experimentell mit dem Einfachen, Zweifachen und Dreifachen der Standardabweichung ermittelt wurden. Innerhalb der Simulationsläufe wurde ermittelt, wie oft die Sicherheitsbestände auf Basis der einfacher, zweifacher und dreifacher Standardabweichung nicht ausreichen, um die Differenz aus Versandmenge zum Referenzdatum und Bedarfsprognose vier Tage vor Abruf abzufangen.

Für einen beispielhaft ausgewählten Artikel einer weißlackierten Stoßfänger-Verkleidung kam es bei insgesamt 26 Lieferungen und einem Sicherheitsbestand auf Basis einer einfachen Standardabweichung zu drei Unterschreitungen, was einem Servicegrad von ca. 88% auf Artekebene entspricht. Bei einem Sicherheitsbestand auf Basis zweifacher Standardabweichung in diesem Beispiel wurde ein Servicegrad von ca. 96% erreicht, bei dreifacher Standardabweichung ein Servicegrad von 100%. Die folgende Ab-

bildung zeigt den Bestandsverlauf des ausgewählten Artikels:

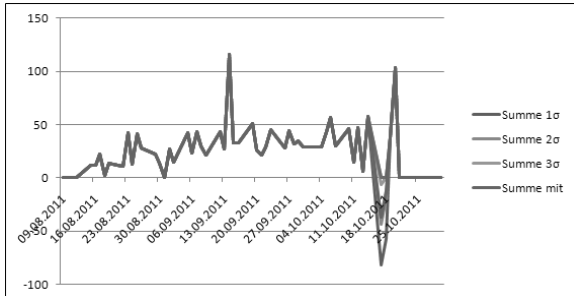


Abbildung 4. Exemplarischer Bestandsverlauf

Tabelle 2 zeigt zusammenfassend auf, bei welchen Artikeln der Datenerhebung es unter welcher Dimensionierung der Sicherheitsbestände zu Unterschreitungen kam.

Basis zur Dimensionierung der Sicherheitsbestände	Gesamtanzahl der Unterschreitungen	Gesamtanteil der Unterschreitungen, Servicegrad
1 σ	388	4,2%, 95,8%
2 σ	87	1,0%, 99,0%
3 σ	43	0,5%, 99,5%

Tabelle 2. Anzahl Unterschreitungen bei verschiedenen Sicherheitsbeständen

Die Ergebnisse der Simulationsstudie zeigen deutlich, dass mit der Dimensionierung der Sicherheitsbestände anhand des Servicegrades ein Großteil der Bedarfsschwankungen in Form von Unterprognosen abgefangen werden kann, auch wenn die Annahme der Normalverteilung als zugrunde liegende Wahrscheinlichkeitsverteilung nicht korrekt war. Bei einer näheren Betrachtung der Artikel, welche auch bei Sicherheitsbeständen auf Basis einer dreifachen Standardabweichung Bestandsunterschreitungen aufwiesen, zeigt sich, dass diese zum Großteil „Lowrunner“-Varianten darstellten, welche durch sehr geringe und unregelmäßige Bedarfe gekennzeichnet sind. Dies ist u.a. auf die Nachteile eines skalenunabhängigen Maßes zurückzuführen. Es wird klar, dass die Auslegung der Sicherheitsbestände alleinig unter der Einbeziehung der FQ als Heuristik bei den meisten Artikeln in diesem Anwendungsfall zu einer tendenziell zu hohen Dimensionierung der Sicherheitsbestände führt. Im nächsten Schritt muss das TS für die Anpassung des

Prognosebedarfs einbezogen werden. Die erneute Simulation hat zum Ergebnis, dass durch die Einbeziehung des TS die Gesamtanzahl der Unterschreitungen weiter von insgesamt 43 auf 31 Unterschreitungen bei dreifacher Standardabweichung reduziert werden konnte, obwohl die Sicherheitsbestände bei Überprognosen reduziert wurden.

5 Fazit

Die bisherigen Ausführungen zeigen deutlich, dass für Zulieferunternehmen der Automobilindustrie die Notwendigkeit besteht, kurzfristige Bedarfsschwankungen des OEM in der operativen Produktionsplanung zu berücksichtigen. Die simulationsbasierte Auswertung der Datenbasis macht deutlich, dass durch eine entsprechende Dimensionierung der Sicherheitsbestände an Halbzeugen in Abhängigkeit des Servicegrades eine Reduzierung der Schedule Instability und eine Beruhigung der Fertigung möglich sind.

Innerhalb der Simulationsstudie wurden erhebliche Unterschiede zwischen einzelnen Fahrzeugbaureihen sowie zwischen „Highrunner“- und „Lowrunner“-Varianten gefunden. So zeigte sich, dass bestimmte Fahrzeugbaureihen im Durchschnitt weniger Bedarfsschwankungen als andere aufwiesen und insbesondere bei „Lowrunner“-Varianten relativ hohe Bedarfsschwankungen auftreten konnten, welche zu einem Abreißen der Lieferkette geführt hätten. Weiterhin konnte gesehen werden, dass die Wahrscheinlichkeitsverteilung der Nachfrage für jeden Anwendungsfall einer vorhergehenden Betrachtung bedarf. Anhand der simulationsbasierten Analyse der Daten auf negative Bestandsverläufe bei verschiedenen ausgelegten Sicherheitsbeständen konnten finale Aussagen zum Servicegrad getroffen werden.

Als Ansatz für weiterführende Arbeiten wird die Implementierung des methodischen Vorgehens des Papers für den operativen Einsatz in Advanced Planning and Scheduling-Systemen gesehen. Im Kontext des SCM können APS-Systeme die Planung und Terminierung von unternehmensübergreifenden Produktionsprozessen und auch die operative Produktionsplanung vor dem Hintergrund kurzfristiger Nachfrageschwankungen der OEMs unterstützen [6].

6 Referenzen

- [1] H. Barthel. Modell zur Analyse und Gestaltung des Bestellverhaltens für die varianten-

-
- reiche Serienproduktion. Jost-Jetter Verlag, Deutschland, 2006.
- [2] Y. Boulaksil, J. Fransoo und E. van Halm. Setting safety stocks in multi-stage inventory systems under rolling horizon mathematical programming models. *OR Spectrum*, S. 121-140, 2009.
- [3] P. Childerhouse, S. Disney und D. Towill. On the impact of order volatility in the European automotive sector. *International Journal of Production Economics*, S. 2-13, 2008.
- [4] S. Chopra und P. Meindl. *Supply Chain Management*, Pentice Hall Verlag, USA, 2004.
- [5] H. Corsten. *Produktionswirtschaft*. Oldenbourg Verlag, Deutschland, 2007.
- [6] H. Corsten und R. Gössinger. *Advanced Planning Systems. PPS-Management*, S. 32-39, 2001.
- [7] A. Jacobsen. *Messung und Bewertung der Nachfragedynamik und logistischer Agilität in der Automobilzulieferindustrie*. Verlag PZH Produktionstechnisches Zentrum GmbH, Deutschland, 2006.
- [8] S. Kadipasaoglu und V. Sridharan. Alternative approaches for reducing schedule instability in multistage manufacturing under demand uncertainty. *Journal of Operations Management*, S. 193-211, 1995.
- [9] F. Klug. *Logistikmanagement in der Automobilindustrie*. Springer Verlag, Deutschland, 2010.
- [10] S. Kuhnt und W. Sieben. Der Einfluss von Nachfrageverteilungen auf das Verhalten von Lagerhaltungsstrategien. *Logistics Journal*, S. 1-12, 2006.
- [11] A. Law. How to build valid and credible simulation models. *Tagungsband 2009 Winter Simulation Conference*, S. 24 -33, 2009.
- [12] M. Lejeune. Probabilistic modeling of multi-period service levels. *European Journal of Operational Research*, S. 299-312, 2013.
- [13] J. Liker und Y. Wu. Japanese automakers, US suppliers and supply chain superiority. *Sloan Management Review*, S. 81-93, 2000.
- [14] A. Narayanan und P. Robinson. Evaluating of joint replenishment lot-sizing procedures in rolling horizon planning systems. *International Journal of Production Economics*, S. 85-94, 2010.
- [15] N. Pujawan und A. Smart. Factors affecting schedule instability in manufacturing companies. *International Journal of Production Research*, S. 2252-2266, 2012.
- [16] B. Scholz-Reiter, C. de Beer, M. Freitag, T. Hamann, H. Rekersbrink und J. Tervo. *Dynamik logistischer Systeme*. Bremer Institut für Produktion und Logistik GmbH, S. 110-138, 2006.
- [17] M. Schmidt, W. Hartmann und P. Nyhuis. Simulation based comparison of safety-stock calculation methods. *CIRP Annals – Manufacturing Technology*, S. 403-406, 2012.
- [18] S. Sivadasan, J. Smart, L. Huatuco und A. Calinescu. Reducing schedule instability by identifying and omitting complexity-adding information flows at the supplier-customer interface. *International Journal of Production Economics*, 2013.
- [19] H. Tempelmeier. *Bestandsmanagement in Supply Chains*. Books on Demand Verlag, Deutschland, 2006.
- [20] J. Vermorel. Sicherheitsbestand anhand Verkaufsprognose berechnen. <http://www.lokad.com/de/berechne-sicherheitsbestand-anhand-verkaufsprognose.ashx>, 2012.
- [21] VDA (Hrsg.): *VDA–Empfehlung 5009: Forecast-Qualitätskennzahl: Definition und Anwendung*. Deutschland, 2008.
- [22] S. Willems und S. Graves. *Optimizing strategic safety stock placement in supply chains*. Cambridge, 1998.
- [23] S. Willems und S. Graves. *Supply chain design: safety stock placement and supply chain configuration*. Handbooks in OR & MS, 2003.

Entscheidungsunterstützung in der Optimierung der Supply Chains von Stahl- und Walzwerken

Lothar März¹

¹STREMLER AG,

lothar.maerz@strempler.de

Das Service Center eines europäischen Stahlwerksbetreibers hat als neutrale Instanz zwischen den Vertriebs- und Produktionsorganisationen die Aufgabe, die Produktionsprogramme der Stahl- und Walzwerke hinsichtlich gemeinsam genutzter Ressourcen abzustimmen, sowie eine effiziente und reibungslose logistische Abwicklung bis zum Endkunden sicherzustellen. Die Herausforderung: Bisher ist es nicht möglich, bei den Planungen auf den einzelnen Wertschöpfungsstufen immer das Optimum der gesamten Supply Chain im Blick zu haben. Weder stehen alle notwendigen Informationen zu Plan-, Stamm- und Bewegungsdaten nutzbar zur Verfügung, noch kann die Komplexität der Wirkzusammenhänge in den vorhandenen Systemen abgebildet werden. Wissen, Daten und Systeme sind verteilt, angewendete Methoden werden unterschiedlich genutzt sowie der Fokus und Entscheidungsrahmen auf jeder Wertschöpfungsstufe anders gesetzt.

Ziel des in diesem Beitrag vorgestellten Planungssystems ist es zum einen, objektive und transparente Informationen integriert über alle Wertschöpfungsstufen hinweg zur Verfügung zu stellen. Darüber hinaus erlaubt die Anwendung den Planungsverantwortlichen durch Veränderungen an Stellgrößen der Plan- und Stammdaten interaktiv sogenannte „Was-wäre-wenn-Szenarien“ erstellen zu lassen und die Auswirkungen auf die Wertschöpfungskette zu simulieren und durch Kennzahlen darzustellen. Diese Szenarien werden mit aktuellen Bewegungsdaten quasi in Echtzeit berechnet und bilden eine schnell verfügbare und verlässliche Datenbasis für anstehende Entscheidungen.

1 Einleitung

Die europäischen Stahl- und Walzwerke stehen vor schwierigen Aufgaben. Der Stahlmarkt hat Überkapazitäten und ist hart umkämpft. Im Wettbewerb entscheidet zunehmend der Preis. Die Kostenstrukturen bzw. die Effizienz der Produktion der europäischen Stahlwerke sind schon weitgehend optimiert. Dennoch sind die Margen unbefriedigend. Hohes Potenzial liegt in der effizienten Abwicklung von Kundenaufträgen über die gesamte Supply Chain, insbesondere in der Logistik. Oftmals werden aufgrund unzureichender Abstimmung der technischen und logistischen Machbarkeit Aufträge angenommen, die entweder mit hohem Aufwand (Kosten) oder mit schlechtem Service (Termintreue) ausgeliefert werden. Darüber hinaus erschweren wechselnde Materialflussströme in der Wertschöpfungskette die Vorhersagbarkeit von Engpässen. Aufgrund von manuell geführten bzw. nicht vorhandenen Vergleichen von verfügbaren Kapazitäten und Anforderungen kommt es immer wieder zu Verzögerungen aufgrund von Engpässen und Störungen, die nur mit hohem Aufwand gelöst werden können.

In dem vorliegenden Beitrag wird die Einführung eines simulationsbasierten Entscheidungsunterstützungssystems vorgestellt, welches zur Optimierung der Supply Chains von miteinander verbundenen Stahl- und Walzwerken dient.

2 Optimierung der Supply Chain

2.1 Zielsetzung der Anwendung

Die Kundenaufträge sind Ausgangspunkt aller Überlegungen. Die Kundenaufträge bestimmen das Walzwerkprogramm, welches durch das Korsett der Kampagnenplanung eine Zuordnung von Kundenaufträgen in engen Grenzen vorgibt. Die Stahlwerksplanung hängt an der Walzplanung. Um die Vorhersagbarkeit von Auswirkungen der anliegenden Aufträge auf die Supply Chain zu analysieren und zu bewerten, sind die Informationen über die Auftragslast, die Prozessen und die Zusammenhängen in den vor- und nachgelagerten Stufen der Walzwerkplanung valide abzubilden. Neben dem konkreten Auftragsbuch sind auch die Vertriebsprognosen in die Vorschau mit einzubeziehen, um frühzeitig Maßnahmen ergreifen zu können.

Im Falle von erkennbaren Engpässen sind alternative Vorschläge zur Lösung des Problems zu erarbeiten und zu bewerten. Eine objektive Bewertung ist nur möglich, wenn die Auswirkungen von Alternativen eine ganzheitliche Betrachtung der Supply Chain erlaubt. Es muss aufgezeigt werden, wie sich ein Szenario auf die internen Kosten und die Servicequalität auswirkt. Dazu müssen die Wirkzusammenhänge der Wertschöpfungsstufen bekannt sein.

2.2 Wirkzusammenhänge der Auftragsabwicklung in der Stahlerzeugung

Im Kontext der Realisierung des Entscheidungsunterstützungssystems umfasst die Wertschöpfungskette der Stahlherstellung die gesamten Schritte von der Inbound-Logistik bis zur Anlieferung beim Kunden. Die Auftragsabwicklung durchläuft chronologisch die Stufen der Schrottbereitstellung, der Halbzeugherstellung (Brammen) im Stahlwerk, die Walzung, die Nachbearbeitung inkl. Reparatur und Pufferung sowie den Versand und die Anlieferung zum Kunden. Zwischen den Wertschöpfungsstufen sind die Materialien zu transportieren. Die bestimmenden Kriterien zum Weitertransport an die nächste Stufe sind inklusive dem auslösenden Prinzip (Pull/ Push) in Abbildung 1 übergeordnet dargestellt.

2.3 Anforderungen an die Anwendung zur Optimierung der Supply Chains

Im Zentrum der Anwendung stehen die Kundenaufträge, die einem Walzwerk und einer Walzkampagne zugeordnet werden. Jedes Walzprodukt durchläuft den Finishing-Bereich mit definierten Arbeitsschritten; im minimalen Fall wird nur verladen. Die Zuordnung der Aufträge zu einem Walzprogramm determiniert

den Zeitpunkt der Fertigstellung. Der Versandtermin determiniert den Tag der Verladung auf dem vorab festgelegten Transportmedium. Im Falle einer fehlerfreien Produktion ist der Materialstrom weitgehend festgelegt.

Mit dem Walzprogramm und den zugeordneten Aufträgen kann der Bedarf an Brammen abgeleitet werden. In Abstimmung mit der Stahlwerkskampagne kann festgestellt werden, ob die anliegenden Walzprogrammanforderungen durch die vorhandenen Brammenbestände und die geplante Stahlwerksproduktion gedeckt werden kann. Ein weiterer Zusammenhang besteht zwischen der Stahlwerksproduktion und den Schrottqualitäten.

Die realitätsnahe Abbildung der Zusammenhänge im simulationsbasierten Planungssystem erlaubt es, die modellhafte Szenarienbildung im Planungssystem und die ermittelten Ergebnisse als valide Grundlage für reale Entscheidungen heranzuziehen. Dabei spielen folgende Aspekte eine Rolle:

— Hohe Datenqualität

Im Falle der Auftragsdaten müssen diese korrekt sein. Im Falle der Vorschau des Vertriebes ist zu prüfen, welchen Informationsgehalt die Plandaten haben und auf welche Planobjekte sie wirken. Beispiel: Die monatliche Vorschau der Auftragsvolumina eines Landes lässt wenig Rückschlüsse auf die Bedarfe an Semis zu, kann aber Aufschluss über die möglichen Materialströme in dieses Land geben.

— Vollständige und korrekte Abbildung der Wirkzusammenhänge

Die quantitativen Zusammenhänge zwischen den

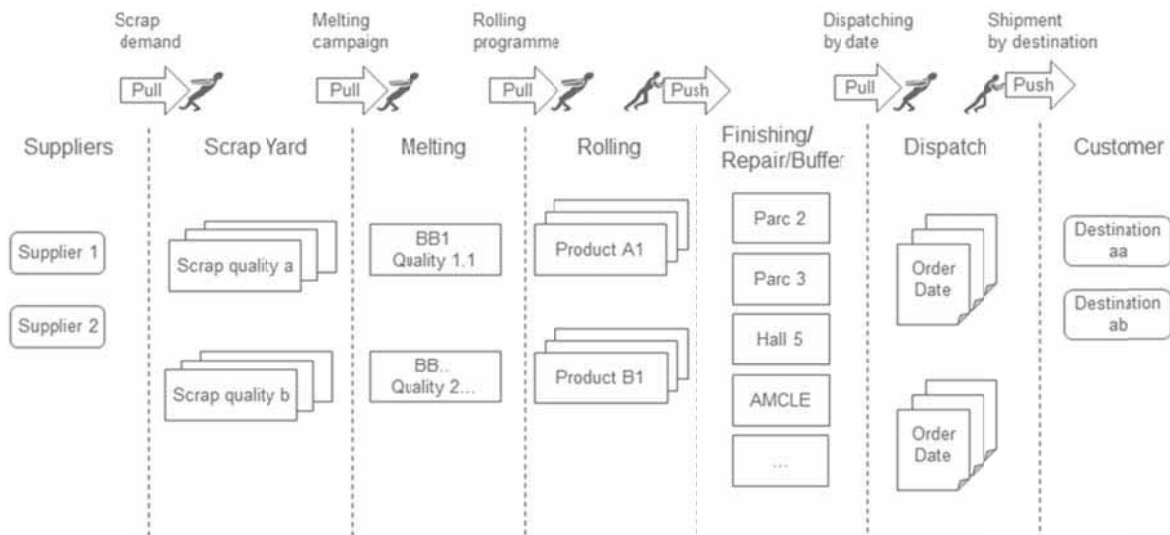


Abbildung 1: Wertschöpfungsstufen in der Stahlindustrie

Planungsbereichen sind präzise und in Bezug auf die Einflusskriterien vollständig zu erfassen. Die Ignoranz einer Einflussgröße kann zu fehlerhaften Aussagen führen. Daher ist eine hinreichende Validationsphase zur Überprüfung der Übertragbarkeit der Ergebnisse auf den Realzustand unabdingbar.

- Berücksichtigung des zeitlichen Fortschritts

Die dynamischen Zusammenhänge bedingen eine Berechnung der Parameteränderungen unter Berücksichtigung des Zeitfortschritts. Aufgrund der Komplexität der Zusammenhänge sind analytische Verfahren hierfür zu aufwendig. Aufgrund des stückgutorientierten Planungsbereiches bietet sich die Problemlösungsmethode der logistischen Simulation an.

2.4 Zielsetzung zum Einsatz des Planungssystems

Das Ziel des Stahlwerksbetreibers ist die Einführung eines Werkzeuges zur Optimierung der Supply Chain. Damit werden folgende Erwartungen verbunden:

- Verbesserte Planung der Supply Chain bis zur Auslieferung durch die Nutzung eines interaktiven Szenarienplanungssystems zur Vermeidung von Unterauslastungen und Engpässen
- Optimierung des Ressourceneinsatzes unter Berücksichtigung von Kapazitäten und Restriktionen
- Konflikterkennung unter Berücksichtigung der dynamischen Abhängigkeiten in Produktion und Logistik
- Absicherung der Vertriebsplanung und schließlich der Kundenzufriedenheit durch Überprüfung der Machbarkeit von Anfragen hinsichtlich Lieferterminen
- Erweiterung der Planungsfunktionalität durch den Vergleich von Szenarien mit veränderten Stellgrößen (Kampagnen, Kapazitäten, Bedarfe)
- Erhöhen der Reaktionsfähigkeit bei Änderungen durch schnell verfügbare Ergebnisdaten und -reports
- Einbindung in operative Planungsprozesse durch definierte Datenübernahmen z.B. anhand von SAP- und ASIA-Schnittstellen.

Entscheidende Voraussetzung für einen solchen Einsatz ist die Funktionalität, eine unmittelbare Darstellung der Auswirkungen einer Änderung auf die Kosten-, Auslastungs- und Terminsituation aufzeigen zu können. Bei einer Veränderung eines Prozess-, Be-

darfs- bzw. Kapazitätsparameters ist ein Simulationslauf zu starten, der in einer für den Anwender noch als unmittelbar wahrgenommenen Zeitspanne (< 3 Sekunden) Ergebnisse liefert.

Der Einsatz des Planungssystems ist sowohl für die kurz- und mittelfristige Planung vorgesehen. Im Kurzfristbereich hilft das System bei der Beantwortung der Frage, wie sich eine kurzfristige und unerwartete Änderung (Störung, Auftrags- bzw. Planänderung, etc.) auswirkt. Im mittelfristigen Planungsbereich kann das System Fragen beantworten helfen, inwieweit die anliegenden Vertriebsprognosen bzw. Leads in Kombination mit dem Produktionsprogramm zu Engpässen oder Unterauslastungen führen. Grundsätzlich lassen sich Planszenarien auf allen Stufen der Wertschöpfungskette durch Änderungen an den Planparametern vornehmen und hinsichtlich ihrer Auswirkungen analysieren.

3 Anwendungsbeschreibung

3.1 Systematik der Planung

Zur Einbindung der Simulation in die operativen Planungsprozesse sind die vorhandenen Stamm- und Bewegungsdaten aus den vorhandenen ERP-Systemen automatisiert zu übernehmen. Die Wirkzusammenhänge von der Schrottanlieferung bis zum Kunden sind in einem Simulationsmodell zu hinterlegen. Anhand der Ergebnisanalyse lassen sich notwendige Maßnahmen ableiten und durch Änderungen an den Stellgrößen erneut durch eine Simulationsberechnung bewerten. Mit der Funktionalität einer interaktiven Rückmeldung wird eine iterative Annäherung an einen Zielzustand durch Szenarien-Analysen durch den Planer ermöglicht. Die Bereitstellung einer solchen Planungsplattform verknüpft das Anwendungswissen des Planers mit einer virtuellen Welt, in der Was-wäre-wenn-Szenarien durchprobiert werden können und als Grundlage für eine Umsetzung in das reale System dienen. Durch ergonomisch einfach zu bedienenden Eingriffs- und Änderungsmöglichkeiten in der Planung lassen sich alle denkbaren Konfigurationen analysieren und bewerten und ermöglichen durch die Interaktivität eine quasi online-Anpassung der Kapazitäten an aktuelle Bedarfe.

3.2 Ablauf der Planung

Die Definition eines Planszenarios ist ein iterativer Vorgang. Zur Durchführung eines Planungslaufes (Simulation) sind die Phasen von Modellbildung, Definition von Szenarien, Berechnung und Auswer-

tung zu durchlaufen. Die Grundlagen der Simulation beinhalten die Modellierung der Stammdaten und der Planungsdaten. Die Definition legt fest, welcher Simulationslauf inhaltlich und zeitlich erfolgen soll. Die Durchführung entspricht der Berechnung durch die Simulation. Die Ergebnisse liegen am Ende der Simulation vor.

Die zugrundeliegende Logik ermöglicht schnelle Berechnungen und stellt die Auswertungsergebnisse quasi in Echtzeit zur Verfügung. Die Bedienoberflächen der Anwendung erlauben eine einfache Änderung von Planparametern per Tabelleneintrag oder Drag & Drop. Somit kann der Planer interaktiv und mit direkter Rückmeldung über die Auswirkungen in kürzester Zeit diverse alternative Szenarien durchspielen und bewerten.

Das Ist-Szenario ist dasjenige Szenario, welches nach Aufruf der Simulation die aktuelle Produktions- und Plansituation aufzeigt, die in den Stamm- und Bewegungsdaten der zugrundeliegenden ERP- und MES-Systeme und sonstigen Quellen hinterlegt sind. Darunter sind neben den Belegungsdaten der Produktion die Auftragsdaten sowie die Vorschau des Vertriebs zu nennen.

3.3 Modellbildung

Die Modellierung eines Simulationsmodells basiert auf einer Vielzahl an Daten, die strukturiert in Form von Stammdatensätzen der involvierten Planungsobjekte abgelegt werden können. Als Planungsobjekte können beispielsweise Aufträge, Prozesse, Ressourcen und Produkte unterschieden werden. Neben den Stammdaten müssen Plandaten erhoben werden, um das Zusammenwirken der Planungsobjekte zu beschreiben. Auch diese Plandaten werden anhand von Datensätzen zusammengefasst, womit nachfolgende Vorteile verbunden sind.

— Die Definition der Simulationsläufe erfolgt durch Angabe weniger, relevanter Datensätze.

— Bei nur geringfügigen Änderungen werden komplette Datensätze kopiert und selektiv Teildaten editiert. Dieser Vorteil wird bei der graphischen Modellierung genutzt: bei der Zuordnung von beispielsweise Walzaufträgen von einer Kampagne zu einer anderen, kann das neue Szenario im Anschluss unter eigenen Namen abgespeichert werden. Das neue Szenario kann mit den vorgenommenen Änderungen jederzeit wieder aufgerufen und nachvollzogen werden.

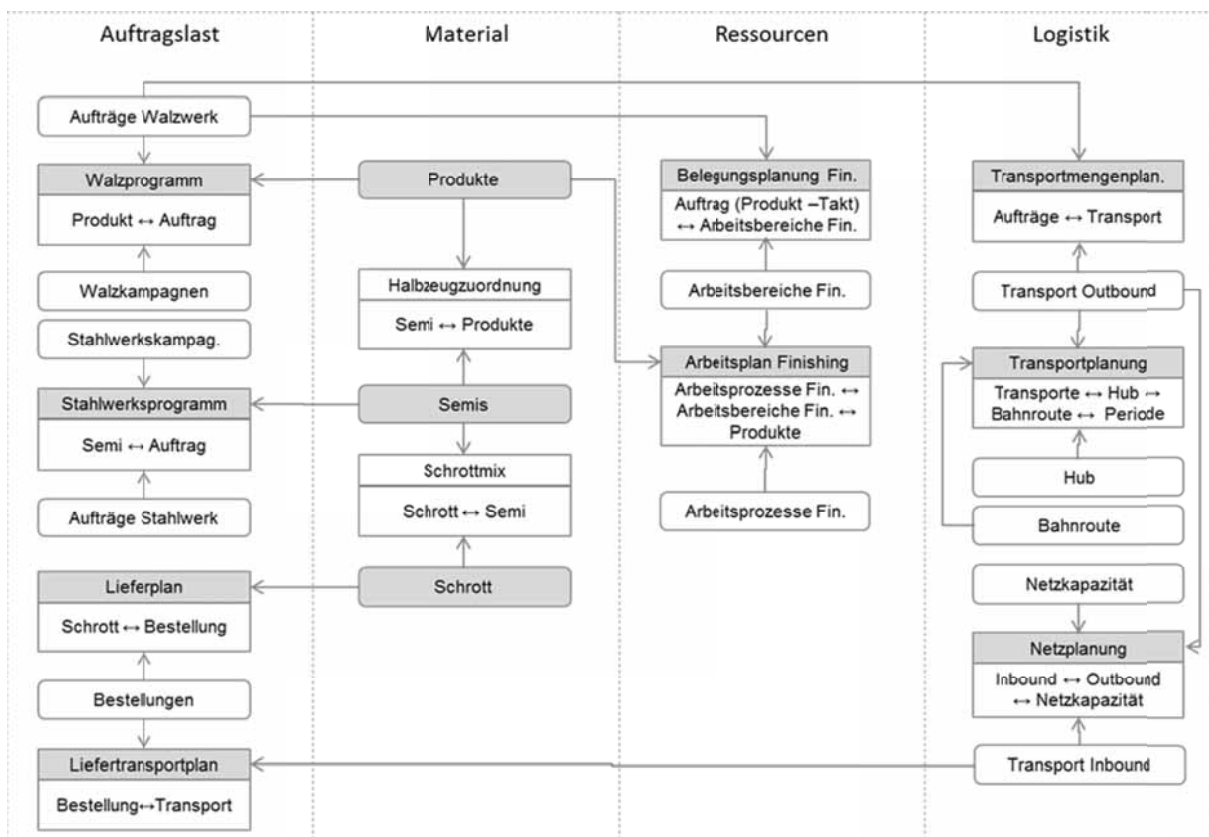


Abbildung 2: Beziehungsgeflecht zwischen Stammdaten und Plandaten

— Die Definition der Plandatensätze erfolgt durch Zuordnung von Stammdaten, die sich aus vorab gewählten Stammdatensätzen rekrutieren. Damit kann eine fehlerhafte Auswahl eines Planungsobjektes vermieden und die Datenkonsistenz durch einfache Prüfmechanismen auf Basis der vorhandenen Datensätze erfolgen.

— Die Struktur der Daten entspricht häufig denen von externen Quellen (z.B. tabellarischen Listen in MS Excel), die über Knopfdruck in einen Datensatz importiert werden können.

Die Zusammenhänge der Stamm- und Plandaten sind Abbildung 2 dargestellt. Im Kern der Gesamtarchitektur stehen die Kampagnen der Stahl- und Walzwerke, die kontinuierlich mit Aufträgen gefüllt werden. Zu jedem Walzauftrag, der einer Walzkampagne zugeordnet wird, bestehen Wechselbeziehungen zu den vorgelagerten Stufen der Stahlwerksplanung bis hin

zur Schrottanlieferung und nachgelagert zu Finishing und Distribution bis zum Kunden.

3.4 Definition eines Planungsszenarios

Mit der Änderung eines editierbaren Stellparameters können neue Szenarien bestimmt werden. In Abbildung 3 sind die editierbaren bzw. nicht editierbaren Stamm- und Plandaten im Zusammenhang aufgezeigt.

Der Anwender legt fest, welche Änderungen er am vorliegenden Szenario (z.B. Ist-Szenario) vornehmen möchte. Die Veränderung eines Planparameters erfolgt entweder durch die Eingabe eines modifizierten Wertes einer Stellgröße oder durch graphische Unterstützung durch Ziehen von Planobjekten von einer Planzuordnung zu einer anderen. Beispielsweise könnten auf diese Weise Kundenaufträge von einer Walzprofilkampagne zu einem anderen Zeitfenster verschoben werden.

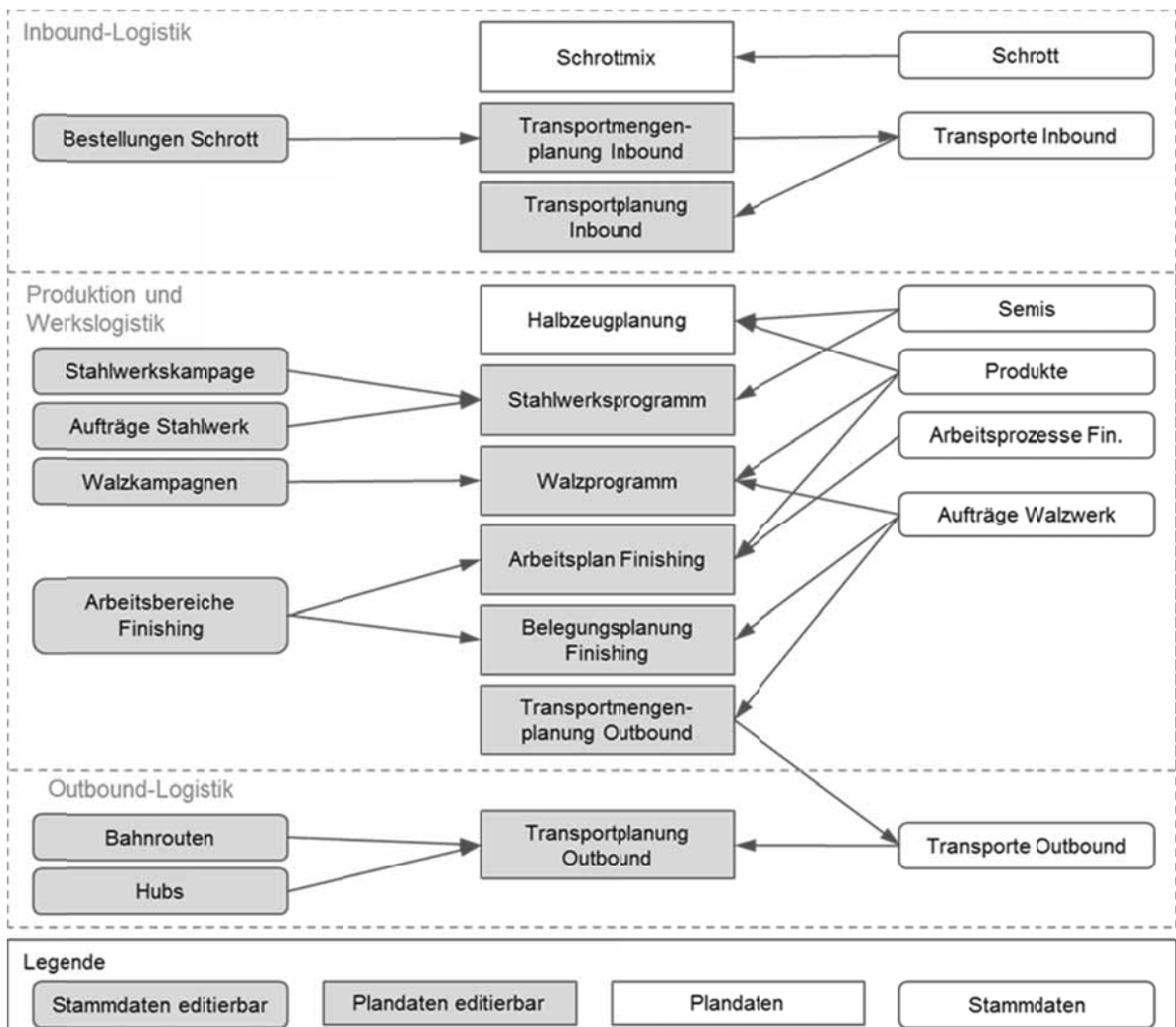


Abbildung 3: Editierbare Stamm- und Plandaten

3.5 Berechnung

Die Berechnung umfasst die Simulation. Die Simulation zeichnet sich durch sehr kurze Laufzeiten aus und erlaubt die Verteilung mehrerer Simulationen auf verfügbare Prozessoren des Servers.

Für die Wahrnehmung einer interaktiven Rückmeldung an den Benutzer sind Antwortzeiten des Systems von wenigen Sekunden bindend. Um die Simulationsläufe über mehrere tausend Prozessschritte mit mannigfaltigen Entscheidungspunkten zeitlich derart zu begrenzen, sind mehrere Maßnahmen zu ergreifen:

- Reduktion der in der Simulationsberechnung verwendeten Daten auf die notwendigen Indizes und Verknüpfung der Ergebnissimulationsdaten mit den für den Anwender relevanten Informationen
- Begrenzung des Simulationsumfangs auf die durch die Veränderung relevanten Bereiche durch ein vorgeschaltete Analyse der Wirkungstiefe der Änderung
- Vermeidung jedweder Schnittstellen und Transformationen zwischen den Funktionen von Simulation, Datenimport und -export, Client-Server-Anbindung, Datenverwaltung und -darstellung
- Anwendung der verteilten Simulation von z.B. unabhängigen Teilbereichen oder Replikationen durch Nutzung von Mehrprozessorrechnern

4 Fazit und Ausblick

Der Nutzen in der Einführung eines solchen Entscheidungsunterstützungssystems für die Vertriebs- bzw. Produktionsorganisation sowie für das Service Center ist die ganzheitliche Optimierung der Supply Chain von der strategischen bis zur operativen Planung:

- Alle für Entscheidungen notwendigen Informationen stehen sofort, jederzeit und in einem einheitlichen Datenlayout integriert über die gesamte Wertschöpfungskette zur Verfügung;
- Das interaktive Szenarien-Planungssystem erlaubt in Echtzeit den Vergleich von alternativen Last- und Systemkonfigurationen mit veränderten Stellgrößen wie Kampagnen, Kapazitäten und Bedarfen;
- Dadurch kann der Ressourceneinsatz optimiert und etwaige Konflikte unter Berücksichtigung von Kapazitäten, Restriktionen und den dynamischen

Abhängigkeiten in Produktion und Logistik vermieden werden;

- Präzise gerechnete Szenarien prüfen die Machbarkeit von Aufträgen (Capable-to-promise CTP) und erlauben die Abwägung zwischen möglichst kosteneffizienter Abwicklung und größtmöglichem Kundenservice;
- Bei Änderungen oder Störungen erhöht sich die Reaktionsfähigkeit und -schnelligkeit;
- Die vorhandenen ERP-/PPS-Systeme werden weiterhin genutzt und liefern die notwendigen Plan-, Stamm- und Bewegungsdaten. Änderungen an diesen Systemen sind nicht erforderlich.

Die Voraussetzungen zur Einführung eines solchen Systems sind eine systematisierte Organisation der Planungs- und Auftragsabwicklungsprozesse. Als Vorbereitung zur Einführung der simulationsbasierten Optimierung der Supply Chain werden daher zunächst ein Produktions- und Führungssystem einheitlich für alle Werke des Unternehmens definiert und eingeführt. Die damit verbundenen Anforderungen der Fachbereiche an die IT werden gesammelt und ebenfalls in ein standardisiertes ERP-Template für Stahl- und Walzwerkprozesse umgesetzt. Darauf aufbauend stehen einheitliche Prozesse sowie Daten zur Verfügung, die als Basis zur Planung und Steuerung in Echtzeit dienen.

Simulation von Abwassersystemen mit der MOSKAN Software

Jan Studzinski¹, Patryk Wójtowicz²

¹Systems Research Institute, Polish Academy of Sciences

²Wroclaw University of Technology, Faculty of Environmental Engineering

studzins@ibspan.waw.pl

In dem Beitrag ist die Modellierung und Simulation eines beispielhaften kommunalen Abwassersystems mit der Hilfe der im Institut für System Analyse (IBS PAN) entwickelten MOSKAN Software dargestellt. Die Software ist auf dem zur hydraulischen Berechnung bestimmten Algorithmus SWMM5 der Firma US EPA gestützt und die zu der Simulation benutzten und das Abwassersystem betreffenden Daten stammen aus dem Posener Wasserwerk. Das untersuchte Abwassersystem besteht sowohl aus dem sanitären wie auch dem Regen-Netzwerk und die Grundprobleme bei dessen Modellierung betreffen die Beschreibung des Regeneinzugsgebietes und die Gewinnung der die Regenfälle betreffenden Daten, was folgenderweise einen entscheidenden Einfluss auf die Kalibration des Netzwerkmodells hat. In dem Beitrag sind die Resultate der Modellierung und Simulation des Abwassersystems wie auch dessen Kalibration vorgeführt.

1 Einleitung

Das in IBS PAN entwickelte MOSKAN System dient zum komplexen Management von kommunalen Abwassersystemen, die von Sanitär-, Regen- und Gesamtnetzen bestehen [1]. Unter ‚Management‘ ist hier die hydraulische Berechnung von Abwassernetzen wie auch die Ausführung mehrerer der operationellen und strategischen Führung des Netzwerkes betreffenden Optimierungsaufgaben verstanden [2]. Zur hydraulischen Netzberechnung ist in MOSKAN der SWMM5 Algorithmus von US EPA adaptiert und die Optimierungs-Algorithmen und – Aufgaben wie auch der Interface des Systems sind in IBS PAN ausgedacht und erarbeitet. In dem Beitrag ist die Anwendung von MOSKAN zur Modellierung und Kalibration eines Teiles des Posener Abwassernetzes beschrieben.

2 Model des untersuchten Einzugsgebietes

Das untersuchte Einzugsgebiet in dem sich das untersuchte Abwassernetz befindet ist in Abb. 1 gezeigt. Das Gebiet hat die Fläche von 122 ha und es besteht meistens aus den industriellen, Wohnungs- und Dienstleistungs-Geländen, was bedeutet, dass die Gebietsfläche überwiegend wasserdicht ist. Das Einzugsgebiet und das in ihm lokalisierte Abwassernetz sind zur Modellierung zu digitalisieren, d.h. ist das Einzugsgebiet in die den das Regenwasser einsammelnden Kanälen entsprechenden Einheitsgebiete und ist das Abwassernetz in die bei der Modellierung zu berücksichtigten rechnerischen

Knoten und Leitungen zu teilen. Nach der Digitalisierung besteht das Einzugsgebiet aus 110 Einheitsgebieten und das Abwassernetz aus 275 rechnerischen Knoten und 285 rechnerischen Leitungen (Abb. 2). In dem Netzwerk ist auch 1 Regenwasserüberlauf einmontiert, wodurch das übermäßige Regenwasser in den in der Nähe vorbeipassenden Fluss abgeführt wird.



Abbildung 1. Das modellierte Einzugsgebiet

Bei der Modellierung des Einzugsgebietes nach dessen Teilung in Einheitsgebiete sind nachfolgend derer folgende Schlüsselparameter festzulegen:

- prozentualer Anteil undurchlässiger Fläche
- hydraulische Breite
- Rauheitsfaktoren durchlässiger und undurchlässiger Flächen
- Retentionsfaktoren durchlässiger und undurchlässiger Flächen
- Senkungsfaktor.



Abbildung 2. Teilung des Einzugsgebietes in Einheitsgebiete und deren prozentuelle Durchlässigkeitswerte.

In dem untersuchten Beispiel des Einzugsgebietes wurden für die definierten Einheitsgebiete:

- die Anteile derer undurchlässiger Flächen anhand der Daten aus dem im Wasserwerk installierten GIS System eingeschätzt;
- derer hydraulische Breiten laut der Formel: $W=A/L$ berechnet, wobei A – Fläche des Einheitsgebietes und L – Länge des jeweiligen Abflussweges des Regenwassers bedeuten;
- die Rauheitsfaktoren für derer durchlässige und undurchlässige Flächen anhand der Satellitenbilder festgelegt;

- die Retentionsfaktoren für derer durchlässige und undurchlässige Flächen anhand der Literaturdaten bestimmt;
- derer Senkungsfaktoren anhand der das numerische Gebietsmodell betreffenden Daten berechnet, wobei das bezügliche Gebietsmodell mittels GIS entwickelt worden ist.

Nach der Festlegung der Einzugsgebietsparameter sind nachfolgend die Modelle von Zuflüssen des kommunalen bzw. industriellen Abwassers und des Regenwassers in die Netzwerkanäle festzusetzen.

Das kommunale Abwasser fließt aus dem Wohnungs- und industriellen Teil des Einzugsgebietes in das Abwassernetz zu. Man hat angenommen, dass das ganze durch die Industrie und die Wohneinheiten gebrauchte Trinkwasser aus deren Wassernetze in das Abwassernetz übergeht. Die Daten zum gebrauchten Trinkwasser sind aus dem CIS System (*Customer Information System*) zu entnehmen. Man hat angenommen, dass die Wassernetznutzer, die in der Entfernung bis 25 m von den entsprechenden Kanälen liegen, ihr Abwasser hundertprozentig zu den Kanälen abführen.

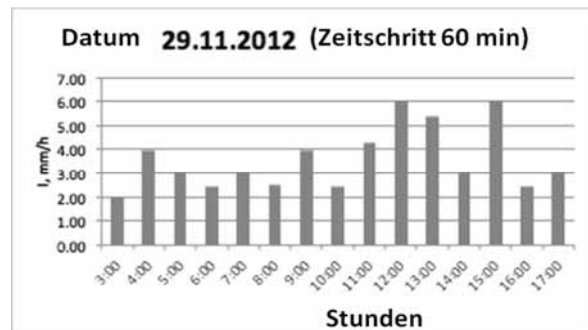


Abbildung 3. Regenfalldaten mit dem Zeitschritt von 60 min.

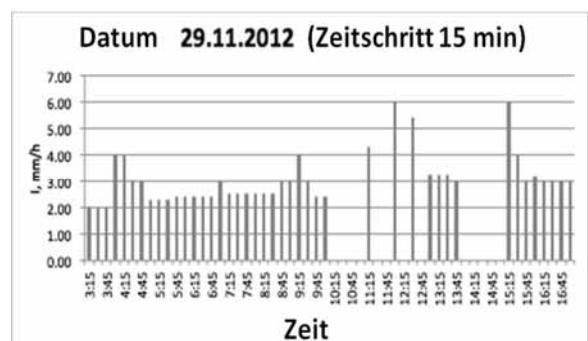


Abbildung 4. Regenfalldaten mit dem Zeitschritt von 15 min.

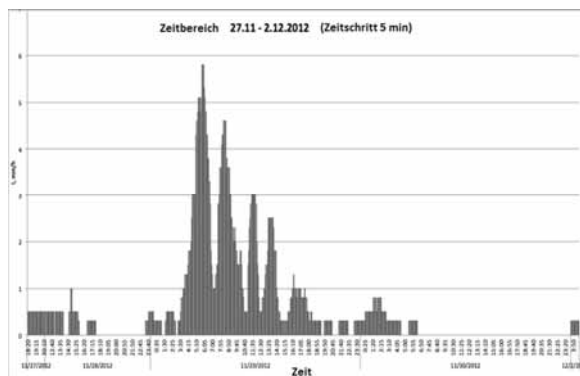


Abbildung 5. Regenfalldaten mit dem Zeitschritt von 5 min.

Die Modelle zur Beschreibung von Regenwasserzuflüssen in das Abwassernetz stützen sich auf den Regenfalldaten, die aus den Regenfallmesspunkten des im Einzugsgebiet installierten Regenfall-Monitoringssystems kommen. In dem untersuchten Einzugsgebiet stand ein solcher Messpunkt zur Verfügung. Die zur Modellierung gewonnenen beispielhaften Regenfalldaten sind in Abbildungen 3, 4 und 5 zu sehen.

3 Model des untersuchten Abwassernetzes

Das untersuchte Abwassernetz besteht aus den aus Beton oder Steinzeug in Jahren 1900 bis 1972 hergestellten Kanälen. Die Durchmesserwerte, Längen und Querprofilformen der rechnerischen Leitungen wurden anhand der Daten aus GIS festgesetzt. Die Leitungsrauheitswerte sind aufgrund der Daten vom Alter und Baumaterial von Kanälen festgelegt und der durchschnittliche Rauheitswert beträgt $n=0,015$.

Die am Abwassernetz installierten Messpunkte messen die Auffüllung der Kanäle mit dem durchschwimmenden Abwasser. Aus dem Wasserwerk wurden die Messdaten von 4 Messpunkten angeliefert (Abb. 6) und nach der Analyse der Daten hat sich herausgestellt, dass Messpunkt 4 außerhalb des untersuchten Einzugsgebietes liegt und seine Daten nicht auszunutzen sind. In ähnlicher Weise hat die Analyse der Daten von Messpunkt 3 gezeigt, dass sie nutzlos sind, da das Messgerät nur die konstanten Werte meldete, was unweigerlich von seiner Beschädigung zeugt hat.

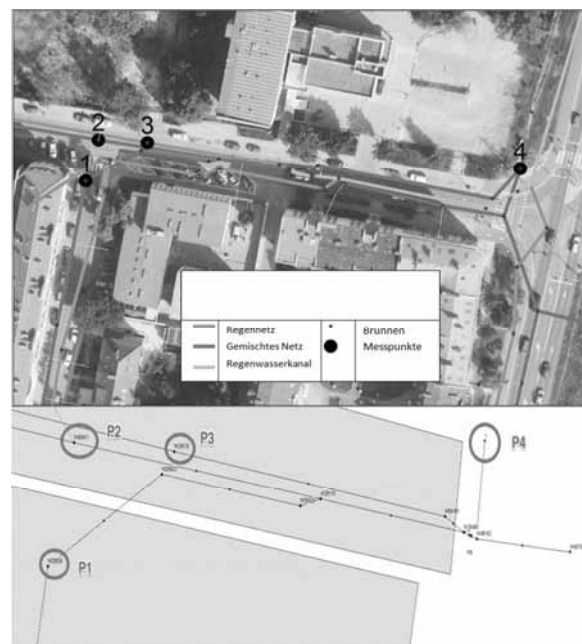


Abbildung 6. Lokalisierung der Abwassernetzpunkte im Netzwerk.

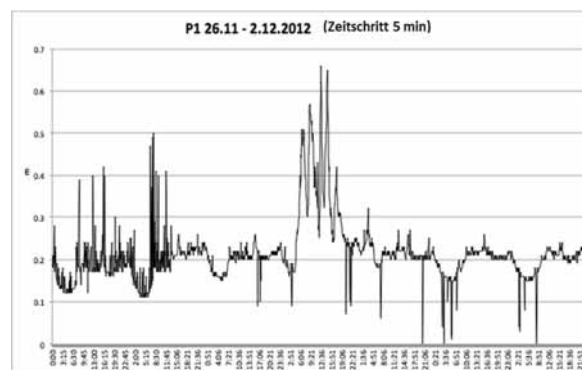


Abbildung 7. Messdaten zur Kanalauffüllung von Messpunkt 1.

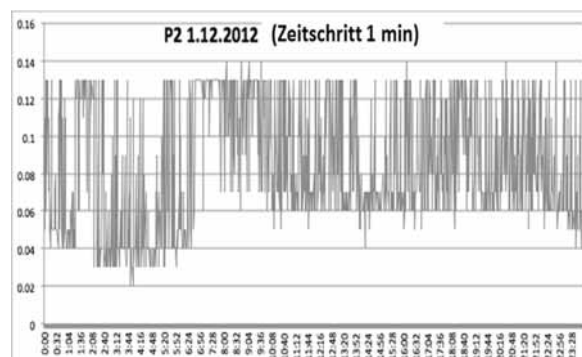


Abbildung 8. Messdaten zur Kanalauffüllung von Messpunkt 2.

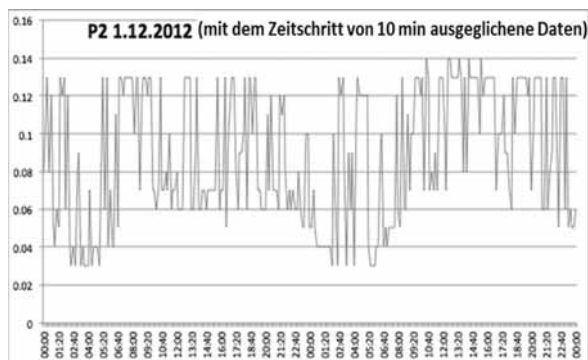


Abbildung 9. Messdaten zur Kanalauffüllung von Messpunkt 2.

Die beispielhaften Daten von den anderen Messpunkten sind in Abbildungen 7, 8 und 9 gezeigt. Der zur Modellierung ausgewählte Zeitbereich für die Messdaten beträgt 1 Woche von 26.11.2012 bis 02.12.2012.

4 Kalibrierung des Abwassernetzmodells

Der Prozess der Kalibrierung eines Abwassernetzes ist generell viel komplizierter als eines Wassernetzes. In dem letzteren Fall sind nur die Rauheitswerte der ausgewählten Leitungen und bzw. auch deren Durchmesser zu ändern und das einzige Problem besteht in geeigneter Leitungsauswahl. Gemeinhin werden zur Parameteränderung die Leitungen angewählt, in denen der Wasserfluss die größten Druckverluste verursacht. Im Fall der Abwassernetze sind bei der Kalibrierung deren Modelle ebenso die Rauheits- und Durchmesserwerte der Kanäle zu ändern, aber außerdem sind viele Möglichkeiten der Änderung der Parameter in den Modellen der Abwasserzuflüsse und des Einzugsgebietes, die einen entscheidenden Einfluss auf das Endresultat der Abwassernetzmodellierung haben.

Bei der Kalibrierung des untersuchten Abwassernetzmodells waren sowohl die genannten Parameter der Kanäle wie auch diese des Einzugsgebietes mit der Hand geändert und die resultierenden Simulationsergebnisse mit den Messdaten verglichen. Die Änderung von Parametern im Fall vom Einzugsgebiet betraf die Bodendurchlässigkeitswerte.

In Abbildungen 10 und 11 sind die Resultate der ausgeführten Kalibrierung gezeigt. Die Daten für Messpunkt 1 sind originell und mit Zeitschritt von 5 min aufgenommen. Die Daten für Messpunkt 2 sind mit Zeitschritt von 10 min ausgeglichen, während die

Messung mit Zeitschritt von 1 min gemacht worden ist.

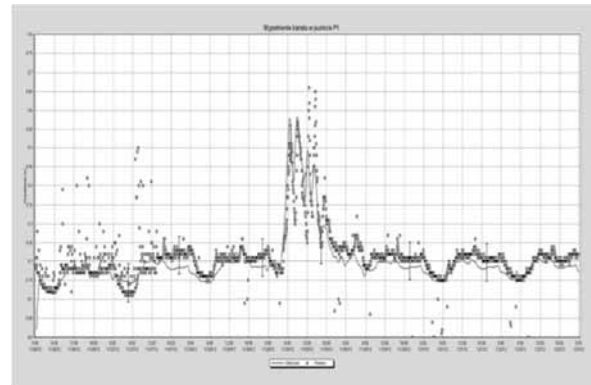


Abbildung 10. Resultate der Model-Kalibrierung für Messpunkt 1 (Messunsicherheit ± 5 cm).

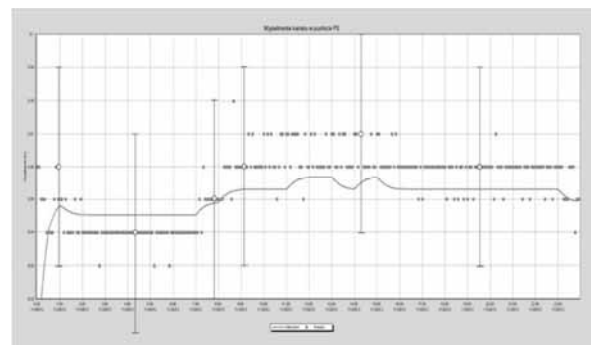


Abbildung 11. Resultate der Model-Kalibrierung für Messpunkt 2 (Messunsicherheit ± 3 cm).

Es geht aus Bildern 10 und 11 heraus, dass die Messdaten (punktweise gezeichnet) und die Modellresultate (mit stetiger Linie gezeichnet) für beide Messpunkte eine relativ gute Übereinstimmung ausweisen, wobei sind die Resultate für Messpunkt 1 viel besser als diese für Messpunkt 2. Dies ist dadurch zu erklären, dass die den Regen betreffenden Daten aus Messpunkt 2 schlechterer Qualität sind als diese aus Messpunkt 1. Aus diesem Grunde ist die Nachahmung der Messdaten aus Messpunkt 2 durch die Modellresultate eher qualitativ während bei den Messdaten aus Messpunkt 1 tritt schon eine gute qualitative Einzigkeit mit den Simulationsergebnissen auf.

5 Schlussfolgerungen

Modellierung von kommunalen Abwassernetzen erleichtert den täglichen Betrieb wie auch die Planung von Netzwerken. In dem letzteren Fall soll das hydraulische Abwassernetzmodell mit Optimierungsalgorithmen gekoppelt sein [3], wobei

die Mehrkriterien-Optimierung mit definierten technischen und finanziellen Kriterien sich für diese Aufgabe besonders günstig erweist.

Nur die mehrmalige Ausführung der Simulation des Abwassernetzmodells mit der optionalen Änderung dessen Parameter kann aber auch eine große Hilfe bei der Abwassernetzföhrung leisten durch:

- Schaffung der Möglichkeit der Ausfindung von Schwachstellen in der Arbeit des Abwassernetzes durch die Analyse der Simulationsresultaten.
- Visualisierung der Netzdynamik.
- Unterstützung der Netzplanung durch die mit der Hand durchgeföhrten Netzänderungen und die Bewertung der hervorgerufenen Wirkung anhand der Simulationsresultate; in dieser Weise können auch die Pläne für die Revitalisierung des Netzes entwickelt werden [4].

Zusätzliche Vorteile gehen aus der Kopplung des Abwassernetzmodells mit den Algorithmen der Kriging-Approximation [5]. Mit diesen Algorithmen können die farbigen Karten der Verteilung von Kanalauffüllungswerten bzw. Abwasserflusswerten im Netzwerk gezeichnet werden. Solche Karten ermöglichen eine schnelle und qualitative Einschätzung der Qualität des Abwassernetzbetriebes, was die nachfolgende quantitative Einschätzung der Netzarbeit erleichtert.

Auch die Kopplung des Abwassernetzmodells mit einem GIS-System, in dessen Datenbank alle geometrischen, technischen und technologischen und das Abwassernetz betreffenden Daten gespeichert sind, bringt große Vorteile für die Netzmodellierung. Diese Vorteile sind wie folgt:

- Richtige Anordnung aller Daten zum Abwassernetz in der GIS-Datenbank.
- Automatischer Import der die Struktur und die Parameter betreffenden Netzwerkdaten aus der GIS-Datenbank in das hydraulische Modell des Abwassernetzes.
- Hilfe bei der Festlegung der Einheitsgebiete des Einzugsgebietes im Fall von Regen- und Gesamtnetzen.
- Integrierung des untersuchten Abwassernetzes mit dem bezüglichen Wassernetz, was die automatische Zuföhrung des kommunalen Abwassers in das Abwassernetz möglich macht.

Die größten Probleme bei der Modellierung eines Abwassernetzes sind die Teilung des Einzugsgebietes in Einheitsgebiete und die Kalibrierung des Netzwerkmodells. Die beiden Handlungen werden zur Zeit manuell gemacht, was viel Zeit kostet und nicht immer ausreichend genau gemacht werden kann. Die Handlungen können aber automatisiert werden und an deren Automatisierung wird zur Zeit in IBS PAN gearbeitet.

6 Literatur

- [1] Sluzalec A., Studzinski J., Ziolkowski A. (2012): *MOSKAN – eine WWW Anwendung in der HTML-5-Technologie zur Bildung und Simulation von kommunalen Abwassernetzen*. In: *Simulation in Umwelt- und Geowissenschaften* (J. Wittmann, B. Page, Hrsg.) Workshop Hamburg 2012, Shaker Verlag, Aachen, 85-95.
- [2] Sluzalec A., Studzinski J., Ziolkowski A. (2013A): *Optimization of sewage structure in the integrated system for sewage design, management and revitalization – MOSKAN*. In: *Simulation in Umwelt- und Geowissenschaften* (J. Wittmann, M. Mueller, Hrsg.), ASIM-Mitteilung AM 146, 203-210.
- [3] Sluzalec A., Studzinski J., Ziolkowski A. (2013B): *Rechnerunterstützte Planung von kommunalen Abwassernetzen mittels des hydraulischen Modells und statischer Optimierung, Modellierung und Simulation von Ökosystemen* (Nguyen Xuan Thinh, Hrsg.), Workshop Koelplinsee 2012, Shaker Verlag, Aachen, 123-133.
- [4] Sluzalec A., Ziolkowski A. (2013C): *Revitalization of sewage networks by means of MOSKAN system*. In: *ISC 2013, Ghent, 22-24.05.2013* (Veronique Limere, El-Houssaine Aghezzaf, eds.), EOROSIS-ETI Publication, Ghent, 39-43.
- [5] Studzinski J., Bogdan L. (2007): *Application of kriging algorithms for environmental and engineering parameters approximation*. In: O. Hryniewicz, J. Studzinski, M. Romaniuk (eds.) *Environmental Informatics and Systems Research, EnviroInfo'2007 Conference*, Warsaw, 12-14 September 2007, Vol. 1, Shaker Verlag Aachen, 185-192.

Ein Prozessmodell zur Aufbereitung und Analyse von Sichtungsdaten von Delfinen und Walen vor La Gomera

Jochen Wittmann¹, Fabian Ritter², Aljoscha Marcel Everding¹, Jonas Amelong¹, Lisa Jermann¹

¹HTW Berlin, FB2, Umweltinformatik, Wilhelminenhofstr. 75A, 12459 Berlin

²M.E.E.R. e.V., Bundesallee 123, 12161 Berlin

wittmann@htw-berlin.de

Vor der Küste von La Gomera werden seit 1995 kontinuierlich Sichtungen von Walen und Delfinen im Rahmen von touristischen Walbeobachtungstouren erfasst. Dabei arbeiten die Tourveranstalter mit dem eingetragenen Verein M.E.E.R e.V. zusammen, der sich die wissenschaftliche Auswertung der Daten zum Ziel gesetzt hat. Ziel dieser Arbeit ist es nun, Anforderungen an ein Prozessmodell zur Erfassung und Analyse dieses Datenmaterials zu erstellen. Die Erfassung erfolgt unter erschwerten Umweltbedingungen auf hoher See, das Datenmaterial soll in einer Datenbank persistent gespeichert werden, die sowohl für die Mitarbeiter und die Touristen auf La Gomera zugreifbar ist, andererseits aber auch die Basis für die wissenschaftliche Auswertung darstellt. Durch entsprechende Rollen- und Rechte-Konzepte soll verhindert werden, dass die Daten zu breit gestreut werden und es zu Beeinträchtigungen der Tiere im Beobachtungsgebiet kommt, andererseits aber das Bedürfnis nach einfachem Zugriff für alle interessierten erfüllt wird. Das Paper schlägt einen Web-Access auf eine entsprechend geschützte Datenbank vor und ergänzt diese Zugriffsschiene um einen Datenexport zur statistischen Analyse sowie einen geographisch basierten Datenzugang über eine geeignete Layerstruktur im geographischen Informationssystem ArcMap.

1 Umfeld: Delfinbeobachtungen vor La Gomera und der M.E.E.R. e.V.

Die Kanarischen Inseln sind in vielerlei Hinsicht „Hot Spots“, nicht nur als Vulkane. Was Artenvielfalt von Walen und Delfinen (Cetaceen) und die Konzentration von Walbeobachtungstourismus angeht, nimmt der Kanarische Archipel eine Spitzenstellung ein. Mit 29 Cetaceenarten beherbergt er eine Vielfalt, die weltweit kaum einen Vergleich kennt. Es handelt sich um vulkanische Inseln, die von Gewässern bis über 3.000 m Tiefe umgeben sind, was die Präsenz von einigen rein pelagischen Cetaceenarten (z.B. Pott-, Grind- und Schnabelwale) in relativer Küstennähe begünstigt. Mehrere Arten sind fest ansässig, viele weitere regelmäßig anzutreffen. Gleichzeitig sind die Inseln ein Magnet für Millionen von Touristen – z.B. gehen rund eine halbe Million Touristen jährlich allein von Teneriffa aus auf Walbeobachtung – mehr als irgendwo anders in Europa. Dahingegen wird auf der Nachbarinsel La Gomera Whale Watching seit vielen Jahren auf nachhaltige Weise praktiziert, wozu M.E.E.R. e.V. mit seinem best-practice Projekt MEER La Gomera [1] maßgeblich beiträgt.

Datenreihen, welche die zu den Untersuchungen notwendigen langen Zeiträume abdecken, sind vergleichsweise rar. Die vom M.E.E.R. e.V. seit dem Jahr 1995 gesammelten Sichtungsdaten der Cetaceen vor La Gomera stellen eine der umfangreichsten Datensammlungen in Europa dar (in der Sichtungsdatenbank des M.E.E.R. e.V. befinden sich derzeit über 7.000 Sichtungen von 1995 bis 2012). Die Daten werden in Zusammenarbeit mit den Whalewatching-Ausfahrten, die zweimal täglich angeboten sind, erhoben. Dass es sich um durchgängig, d.h. für viele Jahre praktisch lückenlos erhobene Daten in einem artenreichen Cetaceenhabitat handelt, macht diese Datenbank in einer weiteren Hinsicht einzigartig: Es können nicht nur Unterschiede in Vorkommen und Verbreitung (für jede Art) zwischen den Jahren bzw. saisonal ermittelt werden, sondern auch langfristige Trends. Die Relation der langjährigen Sichtungsdaten mit den zugehörigen Umweltparametern konnte bislang nicht ausreichend untersucht werden. Um Untersuchungen mit diesem wertvollen Datenbestand zu ermöglichen, setzt sich dieser Artikel das Ziel, den Prozess von der Datenerhebung bis hin zur Darstellung und Analyse zu untersuchen und durch eine durchgängige Softwarelösung in allen seinen Phasen zu unterstützen.

2 Die Entwicklungsziele

2.1 Qualität der Basisdaten sicherstellen

Das gesamte Projekt lebt von dem Datensatz mit den täglichen Sichtsungsdaten, die auf den Ausfahrten erhoben werden. Dabei ist die Erfassung quasi ein "Nebenprodukt" der Leistung für die Touristen, dennoch müssen die Daten korrekt, vollständig und konsistent erhoben werden, damit sie für eine Analyse Verwendung finden können.

2.2 Statistische Auswertungen ermöglichen

Auf diesem Basisdatensatz aufbauend, soll es möglich sein, statistische Analysen durchzuführen. Der Zugang zu den Daten, eine komfortable Selektion der Attribute, die für die Analyse von Interesse sind, und die technische Schnittstelle zu etablierten Analysetools (z.B. MatLab, SPSS etc.) sind bereitzustellen.

2.3 Raumbezogene Auswertungen ermöglichen

Neben der klassischen statistischen Analyse sind raumbezogene Untersuchungen von hohem Interesse: Beispielsweise die Einflüsse von Wassertiefe, Wassertemperatur, Abstand zur Küste, o.ä. als raumbezogene Parameter, die das Vorkommen der einzelnen Arten beeinflussen. In allen Fällen spielt die geographische Lokalisation der Sichtungen die entscheidende Rolle. Auch derartige Untersuchungen sollen unterstützt werden. Dazu bietet sich die Verwendung eines Geographischen Informationssystems (GIS) an.

2.4 Verwendung des Datenmaterials für die Öffentlichkeitsarbeit

Zur Unterstützung dieser Forschungsarbeiten ist eine gute Öffentlichkeitsarbeit unerlässlich. Nur wenn Probleme und Forschungsziele anschaulich dargestellt werden können, ist eine ideelle wie finanzielle Unterstützung durch die Gesellschaft zu erwarten. Die Erstellung von Informationsmaterial, das diesen Zwecken dient und in dessen Mittelpunkt –wenn auch

in anderer Aufbereitung als für wissenschaftliche Zwecke- selbstverständlich auch die Beobachtungsdaten stehen, soll ebenfalls durch das Zielsystem ermöglicht werden.

Auch die Werbung von Teilnehmern für die Ausfahrten unter den Touristen zählt zu dieser Teilaufgabe. Schließlich ist letztlich nur durch diese touristisch geprägten Ausfahrten die kontinuierliche Beobachtung und Datenerfassung möglich.

2.5 Berücksichtigung von Datenschutzaspekten

Während die vorgenannten Punkte im wesentlichen auf einen einfachen, komfortablen Zugang zu den Sichtsungsdaten abzielen, darf nicht vergessen werden, daß diese auch einen schützenswerten Datenbestand darstellen. Zu viele Touristenboote führen zu einer Belästigung der Tiere, eine einfache Recherche, wo welche Tiere aktuell gesichtet sind, könnte zu ähnlichen Effekten wie in den afrikanischen Nationalparks führen, wo besonders interessante Sichtungen häufig einen Stau von Safarifahrzeugen nach sich ziehen und die Tiere in ihrem Verhalten erheblich beeinträchtigen. Aber auch Fischerei- und Jagdinteressen stehen einer vollständigen Öffnung des Datenmaterials entgegen und sollten beim Systemdesign berücksichtigt werden.

3 Der Ist-Zustand

Die aktuelle Ist-Struktur des Bearbeitungs- und Analyseprozesses zeigt Abbildung 1 im Überblick. Der bisherige Prozess beginnt mit der Datenerfassung der einzelnen Beobachtungen/Sichtungen auf dem Boot.

Während der Beobachtung wird –separat für jede Art- ein Formular in Papierform ausgefüllt und nach der Fahrt auf La Gomera manuell in eine Access Datenbank übertragen. Auf Anfrage wird der Inhalt der DB manuell in ein Excel-Sheet exportiert. Dieses dient als Ausgangspunkt zur statistischen Auswertung mithilfe von speziellen Software-Programmen (SPSS, R, MatLab,...).

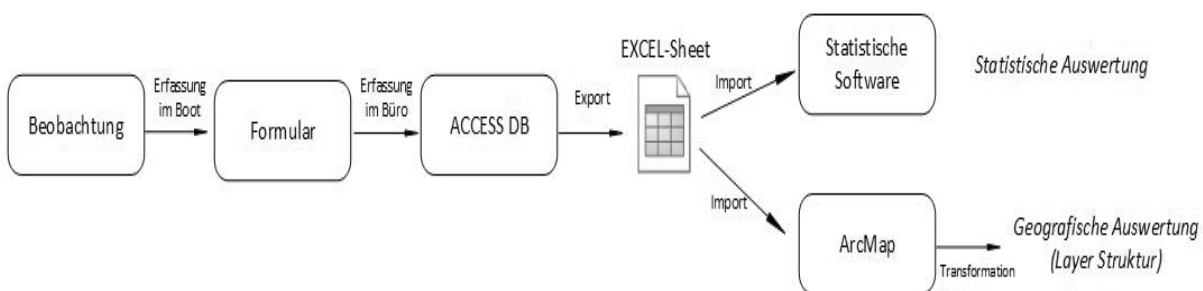


Abbildung 1: Die Ist-Struktur des Bearbeitungs- und Analyseprozesses

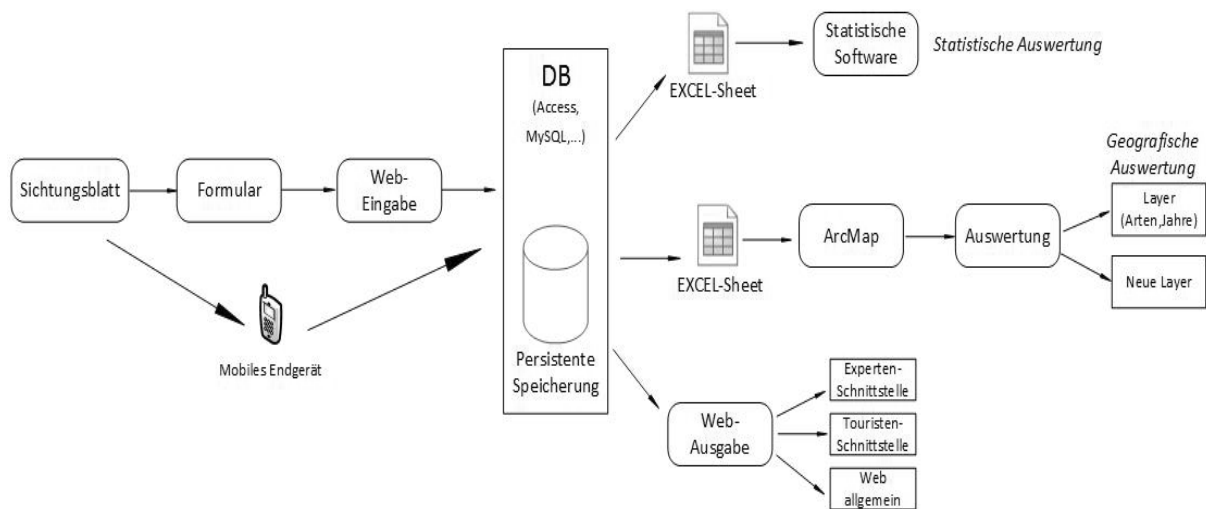


Abbildung 2: Der Bearbeitungs- und Analyseprozess im Zielsystem

Problem hierbei ist Konsistenz der Daten, da regelmäßig eine Vielzahl unterschiedlicher Datenbank-Versionen im Umlauf sind, die parallel bearbeitet und deren Synchronisation und Update keinem geregelten Workflow gehorcht.

Zum anderen dient der Excel-Export als Input für die raumbezogene Auswertung mit Hilfe des Programms ArcMap. Dort werden die Sichtungen als thematische Karten aufbereitet, die raum-zeitliche Analysen nach Arten, Jahreszeit, Meerestiefen, Abstand zur Küste, usw. erlauben.

4 Spezifikation des Zielsystems

4.1 Access/Excel durch zentrale DB-Lösung ersetzen

Um die in Kapitel 2 genannten Ziele zu erreichen, ist es notwendig, das in Kapitel 3 beschriebene System und den derzeit praktizierten Workflow zu verändern. Eine Übersicht über die Architektur des Zielsystems bietet Abbildung 2.

Größte Schwachstelle im aktuellen System ist die Verwendung einer auf Microsoft Access und Excel basierenden Software-Lösung. Dieser Ansatz hat seine Stärken im lokalen Büro-Einsatz. Ein lokaler Einsatz wird jedoch den Anforderungen nicht mehr gerecht. Die Datenbank soll von verschiedenen Orten aus beschreib- und abrufbar sein. Dies ist mit MS Access nicht ohne weiteres machbar. Eine Verwendung eines zentral auf einem Server laufenden relationalen Datenbankmanagementsystems ist somit notwendig. Dies ermöglicht eine konsistente Versionierung der Datenbank, was mit

dem bisherigen System und Vorgehen nicht gegeben war.

Die bisher in der Access-Datenbank abgebildete Datenstruktur ist ein gewachsenes System, das nicht den gängigen Standards der Datenmodellierung gerecht wird. Durch eine Neumodellierung und dabei einhergehende Normalisierung des bisherigen Datenbankdesigns werden Redundanzen und dadurch verursachte Anomalien verhindert. Daten werden sinnvoller zusammengefasst und angeordnet. Dies gewährleistet eine Konsistenz der abgelegten Daten, welche wichtig für die Datensicherheit ist und einen Fortbestand der Daten ohne Datenverlust bei Neu-Einträgen oder Änderung von Datensätzen garantiert.

Die Datenbank auf dem Server stellt das Herzstück des neuen Systems dar. Die Datenbank erhält zusätzlich ein serverseitiges Backend, das es ermöglicht, eine API zu implementieren, die von unterschiedlichen Clientsystemen (Web, Mobile etc.) angesprochen werden kann.

Diese API ermöglicht sowohl die Eingabe der Daten über unterschiedliche Clients, als auch deren Ausgabe auf einer Weboberfläche, sowie den Export ins Excel-Format.

Die Sicherheit der Daten hat eine sehr hohe Priorität. Durch aktuelle Standards der Authentifizierung wird gewährleistet, dass nur berechtigte Clients Zugriff auf das Backend und somit die Datenbank haben. Daten werden zudem ausschließlich verschlüsselt übertragen.

Durch das Ersetzen der bisherigen Lösung durch eine webbasierte Anwendung ist es möglich, dass mehrere Nutzer gleichzeitig an der Datenbank mit identischem Stand der Daten arbeiten können.

Zudem ist die Lösung plattformunabhängig nutzbar. Für die Benutzung ist ein Computer mit Internetverbindung und aktuellem Internet-Browser notwendig. Die Benutzung erfordert somit keinen Computer mit einem Betriebssystem aus der MS Windows-Familie und ist auch auf Geräten mit Linux oder Mac OS X lauffähig. Auch ist man nicht auf einen bestimmten Rechner angewiesen, da die gesamten Daten über das Internet immer aktuell verwendet werden können.

4.2 Excel-Schnittstelle optimieren

Die Systemarchitektur sieht vor, dass die Sichtsungsdaten zwar in einer zentralen Datenbank gehalten werden sollen, dass es darüber hinaus jedoch weiterhin möglich sein muss, die spezialisierten statistischen und raum-zeitlichen Auswertungen in den entsprechenden Spezialsoftwaresystemen durchzuführen. Die Architektur sieht an dieser Stelle auch weiterhin eine Excel-Schnittstelle als Austauschformat vor, weil sich der Import in die Zielsysteme mit diesem Format als sehr einfach herausgestellt hat und zudem eine transparente und lesbare Form der Datenübergabe darstellt.

Für die statistische Auswertung wird der gesamte Datensatz als eine einzige Tabelle exportiert. Das vereinfacht den Zugriff in den Importierenden Programmen.

Für die raumbezogene Auswertung im GIS wird die Tabelle schon beim Export so aufbereitet und in unterschiedlichen Sheets zusammengefasst, dass die Daten im Zielsystem GIS ohne zusätzliche Aufrufe von Selektionsmethoden direkt in die dort angelegten Layer-Strukturen übernommen werden können.

Auf diese Weise ist der Zugriff auf die Daten weiterhin recht einfach, allerdings entstehen wiederum Versionen des Datenbestandes, die nicht automatisch verwaltet und synchronisiert werden. Dieses Vorgehen erscheint pragmatisch und sinnvoll, weil in den Arbeitsschritten im GIS und bei der statistischen Auswertung die Basisdaten nicht verändert, sondern nur gelesen werden.

Zudem unterstützt das Konzept ein Update auf die aktuellste Version des Datenbestandes über einen halbautomatisierten Aufruf des Export-Import-Mechanismus über die Excel-Dateischnittstelle. Ein Bearbeiten bzw. Editieren der Basisdaten soll ausschließlich über die DB-Schnittstelle erlaubt sein.

4.3 Web-Ausgabe

Die Webausgabe der Daten soll auf drei Ebenen möglich sein.

1. Für Projektmitarbeiter ist ein komfortabler Zugriff auf die Datenbank anzubieten. Bisher verfügt das System lediglich über eine Access-Ansicht und die Weiterverarbeitung in Excel. Mit dem neuen System soll es für die Projektmitarbeiter möglich sein, eine komfortable Ansicht sowie die Möglichkeit der nachträglichen Datenbearbeitung im Browser zu erhalten. Auf Grund der geordneten Art und Weise, in der die Datensätze in der neuen Datenbank vorliegen, wird es möglich sein, jede benötigte Art von Darstellung, sei es als Report, Grafik oder Tabelle, schnell und einfach darzustellen.
2. Für Touristen: Mit dem Datenmaterial ist es leicht möglich, Touristen nach ihrer Fahrt Daten und Informationen ihrer Tour aufbereitet zugänglich zu machen. So können Touristen die Tour erneut erleben und haben einen Mehrwert und eine schöne Erinnerung an ihren Ausflug. Kritische Informationen lassen sich für diese Ansicht ausblenden. Eine weitere Verwendung der Sichtsungsdaten wäre eine Prognose für zukünftige Sichtungen. Touristen könnten bei der Buchung einer Tour bereits Informationen erhalten, welche Sichtungen unter den aktuellen Bedingungen möglich wären.
3. Für die allgemeine Info-Website: Über die API ist es leicht, Sichtsungsdaten in aufbereiteter Form auch auf der Webseite von M.E.E.R. e.V. zugänglich zu machen. Dabei greifen auch diese Darstellungen direkt und konsistent auf das originale Datenmaterial zu.

4.4 Dateneingabe optimieren

Eine Eingabemaske im Browser ersetzt die Eingabe in MS Access. Über mobile Endgeräte wie Smartphones oder Tablets wird es möglich, Sichtsungsdaten digital und unmittelbar in die DB einzugeben. Dies macht das Ausfüllen von Sichtungsbögen auf Papier und deren anschließende Übertragung in die Datenbank überflüssig. So kann ein Medienbruch vermieden werden, der nicht nur arbeitsintensiv, sondern auch sehr fehleranfällig ist. Es können so Übertragungsfehler durch un-

leserliche Handschrift oder Vertipper reduziert werden. Dies hat eine signifikant bessere Qualität der Daten zur Folge.

4.5 GIS-Strukturen zur Analyse bereitstellen

Zur Aufbereitung und Analyse im GIS lassen sich immer wiederkehrende Arbeitsschritte festmachen:

- a) Import der Datentabelle (DB-Anbindung)
- b) Anpassung der geographischen Koordinaten an das im GIS verwendete Datenformat
- c) Aufbereitung des Datensatzes in einzelne, standardmäßig anzulegende Layer mit den Sichtungen nach Art, Beobachtungszeitraum, Ort, ... inklusive einer ansprechenden Symbologie
- d) Ermittlung der Meerestiefe für die Positionen der Sichtungen
- e) Ermittlung des Abstands zur Küste für die Positionen der Sichtungen
- f) Bereitstellen von raum-zeitlichen Karten für weitere Umweltparameter wie z.B. Oberflächentemperatur des Meeres, Chlorophyllgehalt, etc..

Der letztgenannte Punkt wird über eine geeignete Recherche und die Einbindung der entsprechenden Themenkarten ins Arbeitsfenster des GIS gelöst.

Die Punkte a) bis e) stellen Arbeitsschritte dar, die

1. den Import bestehender Daten betreffen (Schritt a) bis c)) und durch ein einziges Update-Tool automatisch durchgeführt werden können, wenn die Grundstrukturen im GIS angelegt sind.
2. bestehende Informationen um raumbezogene Attribute bzw. Attributwerte ergänzen (Schritt d) und e)). Diese Funktion sollte benutzergesteuert ausgelöst und kontrolliert durchgeführt werden.

Für beide Typen von Aktionen bietet sich die Python-Skript-Schnittstelle in ArcMap [2] an, als benutzerfreundlichere Variante kann auch eine zusätzliche Visualisierung mit Hilfe des sogenannten Modelers [3] angeboten werden.

5 Die erreichten Teilziele

5.1 Aufbereitung des Datenmaterials

Zur Vorbereitung mussten zunächst die im Umlauf befindlichen Versionen des Excel-Sheets gesammelt und

zusammengeführt werden. Diese Arbeit wurde manuell durchgeführt, so dass jetzt eine Datenbank vorliegt, die fortlaufend vom Jahre 1995 bis März 2014 über 9000 Datensätze/Sichtungen umfasst.

Zudem wurden in diesem Arbeitsschritt anhand von Seekarten die fehlenden Attribute Meerestiefe und Abstand zur Küste erfasst und in der Access-Datenbank vervollständigt. Außerdem wurde die Korrektheit der Datensätze mit Hilfe der Sichtungsblätter überprüft und gegebenenfalls korrigiert. Damit steht für die Integration in die neue Systemarchitektur eine vollständige und konsistente Datenbasis zur Verfügung.

5.2 Implementierung der zentralen DB-Komponente

Zur Implementierung der Webanwendung ist die Programmiersprache Ruby und das Web-Framework Ruby on Rails zum Einsatz gekommen. Ruby on Rails gilt als sehr sicheres und stabiles Framework, welches aktiv weiterentwickelt und gepflegt wird. Es bringt von Haus aus bereits viele Features mit, die im Zielsystem zum Einsatz kommen sollen. Als Datenbankmanagementsystem ist die Verwendung von MySQL geplant. Durch Ruby on Rails ist es möglich, leicht auf andere Datenbankmanagementsysteme zu migrieren, falls MySQL später den Anforderungen nicht mehr gewachsen sein sollte. Die API basiert auf dem REST-Paradigma und verwendet als Austausch-Format JSON. Durch Verwendung von HTTPS werden die Daten zudem ausschließlich verschlüsselt übertragen. Für den View-Layer im Desktop-Bereich ist eine Mischung aus Twitter Bootstrap und einem JavaScript-HTML5-Framework wie Ember.js angedacht. Für mobile Endgeräte soll eine mit dem Mobile Web-JavaScript-Framework Sencha Touch generierte Webansicht verwendet werden. Ein nächster Schritt wäre die Webapplikation in einem PhoneGap-Container zu implementieren, um eine native Anwendung zu erhalten.

Alle verwendeten Technologien stehen unter Open Source-Lizenzen und sind frei von Lizenzkosten, was dem begrenzten Budget des gemeinnützigen Projektes entgegen kommt.

5.3 Aufbau von Datenstrukturen im GIS

Die in 4.5 skizzierten Datenstrukturen sind im System ArcMap angelegt und erlauben bereits die Darstellung erster Auswertungen in Form von Sichtungskarten, die gemäß der unterschiedlichsten, für eine Auswertung interessanten Attribute erzeugt werden können. Als

Beispiele sollen an dieser Stelle dienen (alle Abbildungen aus [4]):

- die Karte der Meerestiefen über alle Sichtungen (Abb.3)
- die Karte der Sichtungen unterschieden nach Art im Monat April von 1995 bis 2011 (Abb.4)
- die Karte der Sichtungen einer Art (Brydewal) im Monat September über sämtliche Jahre (Abb.5).

Ziel der Analyse kann es nun auf der Basis dieses Kartenmaterials sein, Zusammenhänge zwischen dem Auftreten der Arten untereinander, Nachbarschaften, Beobachtungszeit, Meerestiefen, Küstennähe und der zusätzlichen unter 4.5 genannten Parameter zu finden. Dabei sind die (geographische) Position der Sichtung und die zeitliche Dimension über die Sortierung nach Sichtungsjahr visualisierbar und erlauben eine Sicht auf die Basisdaten, die über die reine statistische Analyse hinausreicht.

6 Ausblick

In den hier beschriebenen Schritten konnte eine Softwareumgebung erstellt werden, die es den Fachexperten erlaubt, raum- und zeitbezogene Analysen auf dem wertvollen Datensatz der Cetaceen-Sichtungen vorzunehmen. Die Konsistenz der Datenhaltung ist durch die zentrale DB-Server-Lösung gesichert. Die unterschiedlichen Anwenderprofile sind durch ein entsprechendes Rollenkonzept abgebildet. Die Implementierung als Web-Lösung bietet Komfort unter Nutzung der aktuellen Softwarekomponenten, die GIS-Anbindung vereinfacht den Datentransfer zur raumbezogenen Analyse.

Als nächste Arbeitsschritte stehen nun die eigentlichen Untersuchungen von Korrelationen zwischen den Beobachtungsdaten und den Umweltparametern an. Hierbei stellt das Geoinformationssystem mit seinem Tool-Kit mächtige Werkzeuge zur Verfügung. Erste Überlegungen deuten dennoch darauf hin, dass es einen Analyse-Workflow geben wird, der die klassischen statistischen Methoden mit den GIS-Werkzeugen koppelt. Dieser Bereich wird auf informationstechnischer Seite weitere Überlegungen und Konzepte notwendig machen, die eine intuitive, problemorientierte Bedienung und Durchführung von raum-zeit-bezogenen statistischen Untersuchungen ermöglichen.

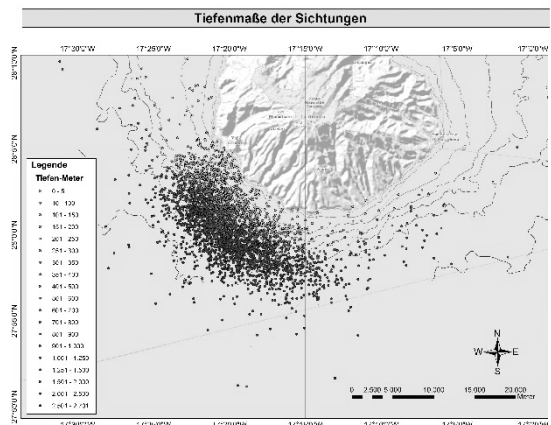


Abbildung 3: Meerestiefe an den Sichtsungspositionen (aus [4])

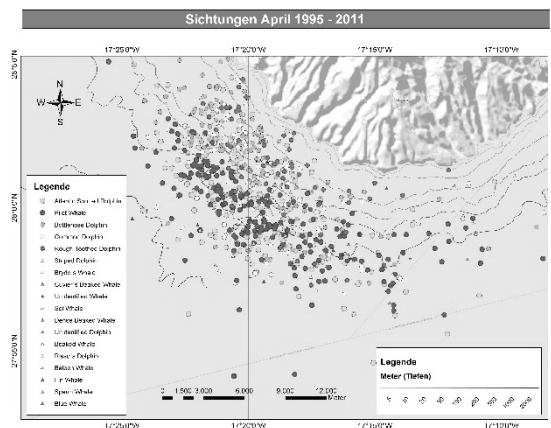


Abbildung 4: Sämtliche Sichtungen im April unterschieden nach Art (aus [4])

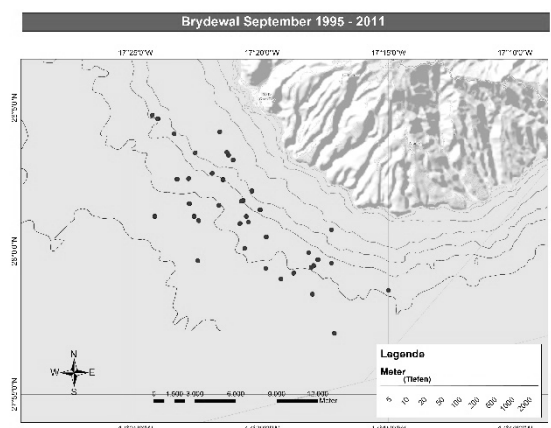


Abbildung 5: Sämtliche Brydewal-Sichtungen im Monat September (aus [4])

7 References

- [1] Fabian Ritter: Model for a Marine Protected Area designed for sustainable Whale Watching Tourism off the oceanic Island of La Gomera (Canary Islands), Report by M.E.E.R. e.V., Berlin 2012
- [2] ESRI: Introduction to Geoprocessing Scripts Using Python @, http://downloads2.esri.com/campus/PRD-pdfTOC/50127508_10.X.pdf (Zugriff am 14.07.2014)
- [3] ArcGIS-Tutor: An overview of ModelBuilder, http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=An_overview_of_ModelBuilder, (Zugriff am 14.07.2014)
- [4] Lisann Heuer: Darstellung von Sichtungsdaten von Walen und Delfinen vor La Gomera mithilfe eines Geographischen Informationssystems (GIS), Bachelorarbeit, HTW Berlin, FB2, Umweltinformatik, Berlin 2013

Entwicklung eines Logistiktools zur Grobmodellierung der Positionierung von Sammel- und Verwertungsstellen von organischem Abfall

Thomas Baur¹, Sascha Fieting¹, Christopher Witte¹, Dustin Müller¹, Jochen Wittmann¹,
Ruth Scheidegger²

¹HTW Berlin, FB2, Umweltinformatik, Wilhelminenhofstr. 75A, 12459 Berlin

²Swiss Federal Institute of Aquatic Science and Technology (Eawag), PF 611, 8600 Duebendorf, Switzerland

[S0520371 / S0523964]@htw-berlin.de

Für ein Projekt der Eawag, das sich mit der Sammlung und der Verwertung von organischen Abfällen in entwickelnden Ländern beschäftigt, soll ein Software-Werkzeug entwickelt werden, das eine Groboptimierung der Transportlogistik erlaubt, und im Sinne der Szenariotechnik die Lokalisation von Sammelstellen und Verwertungsanlagen abbilden kann. Das Werkzeug soll ohne tiefgehende Schulungsmaßnahmen zu verwenden, einfach parametrisierbar und auf Standard Windows-PCs lauffähig sein. Dieses Paper stellt den entwickelten Prototypen vor.

Die Architektur lehnt sich an das Layer-Prinzip der Geographischen Informationssysteme an und erlaubt dem Nutzer über eine Karte oder ein Luftbild des Untersuchungsgebietes (Layer 1) die eigentliche Logistikebene graphisch aufzubauen. Dazu werden zunächst die Quellen für den organischen Abfall definiert, indem man Flächen mit ähnlicher Besiedlungsstruktur identifiziert und entsprechend des Besiedlungstyps mit einem durchschnittlichen Aufkommen von organischem Abfall pro qm charakterisiert. Dieses Flächenmosaik wird als Layer 2 gespeichert. Auf einem dritten Layer können weitere logistisch notwendige Stationen definiert werden: Sammelstellen, Verwertungsstellen und punktuelle Quellen für organischen Abfall (Märkte, ...).

Darüber hinaus gibt es Verbraucher für die Produkte der Verwertungsstellen: z.B. Gärtnereien als Abnehmer von Kompost oder Fischzuchtanlagen als Abnehmer von Proteinen, die in Black-Soldier-Fly-Anlagen erzeugt werden. Punktobjekte sämtlicher Typen werden durch Transportverbindungen verbunden, die über das Transportmittel und ihre Kapazität spezifiziert sind. In einem durch diese Layer definierten Szenario kann nun interaktiv eine Kapazitätsprüfung und Wirtschaftlichkeitsrechnung für die einzelnen Stellen vorgenommen werden. Da die Position auch die geographischen Beziehungen abbildet, können auch sehr einfach (nämlich durch Verschieben der Objekte) alternative Standortentscheidungen untersucht werden. Wesentlich für die Entwicklung des vorgestellten Tools war der bewusste Verzicht auf eine Detailbeschreibung der Prozesse, die stets auch die Bestimmung einer Vielzahl von Parameterwerten nach sich gezogen hätte, zugunsten einer einfach zu bedienenden und mit einfach zu ermittelnden Pauschalwerten funktionierenden Grobmodellierung.

1 Einleitung

Bei diesem Projekt ist eine Software entstanden, die eine Grobmodellierung und der Positionierung von Sammel- und Verwertungsstellen von organischen Abfall auf Grundlage einer Satellitenkarte ermöglicht.

Bei der Entwicklung der Software musste besonders auf die Einfachheit und die Kompatibilität geachtet werden. Grund hierfür ist, dass die Zielgruppe, die Indonesischen Abfallbeauftragten, keine Erfahrung mit komplexen Softwareprodukten aufweisen und keine geeignete Infrastruktur für die Einrichtung einer GIS-basierten Anwendung und entsprechende Schulungen bereitstehen. Darüber hinaus sollte in der aktuellen

Projektphase keine detaillierte Wegeplanung und Wegeoptimierung stattfinden, es sollte den Planern ein ohne allzuviel Aufwand zu erstellendes und mit einfach zu beschaffenden Basisdaten zu parametrisierendes Grobmodell zur Verfügung gestellt werden, das bei der übersichtsmäßigen Abschätzung von Transporten und einer optimalen Positionierung von Sammelstellen und Verarbeitungseinrichtungen helfen kann.

1.1 Ziele

OSCAR ist eine nicht kommerzielle Software und steht für *Organized and Structured Collection of Agricultural Resources*. In ihrer ersten Ausbaustufe ist sie

darauf ausgerichtet, ein Entwicklungsprojekt der Eawag (Eidgenössische Anstalt für Wasserversorgung, Abwasserreinigung und Gewässerschutz) zu unterstützen. Da die Schweiz bereits über eine gute Wasserversorgung und –qualität verfügt, hat sich die Abteilung Wasser und Siedlungshygiene in Entwicklungsländern der Eawag zum Ziel gesetzt anderen Ländern, speziell Entwicklungsländer, zu unterstützen. Diese Länder verfügen über keine bzw. nur unstrukturierte Systeme um die dort herrschende Problematik mit (organischem) Abfall in den Griff zu bekommen. In Zusammenhang mit dem Projekt FORWARD (From ORganic WAstE to Recycling for Development) [1] wurde A ein Pilotprojekt in Zusammenarbeit mit der Hochschule für Technik und Wirtschaft (HTW-Berlin) und zwei indonesischen Städten – Pekalongan und Sidoarjo – ins Leben gerufen. Ziel dieses Projektes ist ein Software-Tool zu konzipieren und zu entwickeln, dass die dortigen Behörden bei der Strukturierung eines Abfallmanagements unterstützt. Das heißt die Abholung, Wiederverwertung und Entsorgung muss entsprechend visualisiert und strukturiert werden können. Daraus leitet sich unmittelbar die Zielsetzung der Arbeit ab.

- Durch die Unterstützung des Abfallmanagements für organische Abfälle, sollen die Behörden im Aufbau einer Kreislaufwirtschaft unterstützt werden, was einerseits eine sachgemäße Abfallentsorgung und damit eine Verbesserung der hygienischen Situation bewirken wird und andererseits über die Vermarktung der Endprodukte einer Verwertung der Stärkung der lokalen Wirtschaft dienen kann.
- Organischer Abfall ist in urbanen Gebieten ein ernsthaftes Problem. Dieser bleibt auf öffentlichen Straßen liegen und kontaminiert das Grundwasser. OSCAR soll dabei helfen den organischen Abfall als Wirtschaftsgut zu betrachten. Als Wirtschaftsgut besitzt der Abfall einen Wert und wird nicht mehr auf den Straßen liegen gelassen. OSCAR soll dabei unterstützen, die Verschmutzung der Umwelt durch organischen Abfall in urbanen Gebieten zu verringern.
- Im Projektland Indonesien sollen die Planungsaufgaben durch eine Software unterstützt werden, die auf Ebene der städtischen Planungsbehörden eingreift und die Transportkoordination im Abfallmanagement unterstützt. Bislang werden derartige Aufgaben mit analogem Kartenmaterial bearbeitet (z.B.

mit einer Stadtkarte mit Pins und Fäden). OSCAR soll diese Geschäftsprozesse mit Hilfe der Informationstechnologie effizienter gestalten.

1.2 Endprodukt

Das Endprodukt dieses Projektes ist eine intuitiv bedienbare Software, die der Eawag zum direkten Einsatz im Projekt übergeben wurde, um als ein Baustein des Gesamtprojektes in Indonesien zum Einsatz zu kommen.

1.3 Begrifflichkeiten

Um die Workflows der Anwendung zu verstehen, müssen zunächst die modellierbaren Objekte klargestellt werden. Insgesamt gibt es fünf Objektklassen, welche im folgenden Abschnitt näher erläutert werden.

- Gebiete

Gebiete mit einheitlichem Besiedlungsmuster sollen als Quellen von organischem Abfall dienen. Sie werden als Polygone in eine Karte eingezeichnet. Als Gebietstypen können ausgewählt werden: Siedlungsgebiet, Industriegebiet, Agrargebiet, usw. Anhand der Siedlungstypen lässt sich auf die Abfallart und den prozentualen Anteil von organischen Abfällen schließen. Gebiete sind damit ein Typ von Quellen von organischem Abfall. Weitere Entstehungsmöglichkeiten von Abfällen bieten die

- Punktuellen Quellen

Punktuelle Quellen sind eine besondere Art von Quellen. Sie sind nicht an Gebiete gebunden und können trotz Gebietstypes andere Abfallarten produzieren. Typische Beispiele sind Wochenmärkte, Fischmärkte, Gärtnereien, usw.

- Sammelstellen

Sammelstellen sind Zwischenlager für organischen Abfall, an denen das Aufkommen verschiedener Quellen, (die durchaus auch unterschiedlichen Typs sein können), zusammengetragen und kurzfristig zwischengelagert wird.

- Verwertungsstellen

Von den Sammelstellen werden die Abfälle weiter zu den Verwertungsstellen transportiert. Eine Verwertungsstelle könnte z.B. eine Biogasanlage oder ein Kompostierbetrieb sein. In Verwertungsstellen wird der Input „organischer Abfall“ behandelt und zu einem

Produkt, dem „Output“ der Verwertungsstelle, transformiert.

- Senken

Senken repräsentieren im OSCAR-Kontext die Abnehmer der Produkte der Verwertungsstellen, also sozusagen die Kunden. Die Produkte werden zu den Senken transportiert und verlassen dort den Betrachtungshorizont der Software.

- Verbindungen / Transportwege

Objekte lassen sich in OSCAR miteinander verbinden, sodass der Nutzer in der Lage ist, seine spezifischen Transportwege zu visualisieren und zu managen. Diesen Transportwegen können weitere Informationen hinzugefügt werden wie z.B.: Wegetyp (z.B. befestigt oder unbefestigt) und Transportfahrzeugtyp. Transportfahrzeugtypen können zum Beispiel sein: Zu Fuß mit Karren, Fahrrad mit Karren und Truck < 3,5t oder >3,5t. Zudem können Transportfahrzeugtypen noch bestimmte Eigenschaften zugeordnet werden, wie Kapazität (kg), maximale Geschwindigkeit (km/h und/oder mp/h), benötigte Arbeitskräfteanzahl und Kosten (Geldeinheit/h).

2 Anforderungen

2.1 Nicht-funktionale-Anforderungen

- Die Software muss simpel sein. Sie muss von ungeschultem Personal nach kurzer Einführung bedienbar sein.
- Die Anwendung muss auf low-budget Standardhardware lauffähig sein.
- Für die Umsetzung steht nur ein stark begrenztes Budget zur Verfügung, daher sollten bei der Nutzung von Drittsoftware und fremden Ressourcen nur freie Lizenzen genutzt werden.

2.2 Funktionale-Anforderungen

An dieser Stelle soll hier ein Auszug aus dem Lastenheft angeführt sein, der eine Liste der wichtigsten Anforderungen aufzeigt:

- Der Nutzer muss ein Satellitenbild des Untersuchungsgebietes als Hintergrund auf die Arbeitsfläche laden können.
- Der Nutzer muss verschiedene Szenarien für die Positionierung von Quellen, Sammel- und Verwertungsstellen sowie Senken auf Basis einer Karte anlegen und verwalten können.

- Dem Anwender muss eine Stammdatenverwaltung für die Werte der Basis-Parameter für die genannten Objektklassen sowie für die Transportoptionen zur Verfügung gestellt werden.
- Die definierten OSCAR-Objekte müssen per Drag-and-Drop auf der Karte platziert werden können.
- Die OSCAR-Objekte müssen miteinander verbunden werden können, um Transportwege abzubilden.
- Sämtliche Knoten (die Oscar-Objekte) und Kanten (Verbinder) müssen frei parametrisierbar sein. Die Parameterwerte stammen entweder als Defaults aus der Stammdatenverwaltung, oder sie können individuell manuell eingegeben werden.

Besonders zu erwähnen ist, dass die Software OSCAR zu diesem Zeitpunkt keine Simulationsengine besitzt. Im weiteren Verlauf dieser Arbeit werden unter anderen „Transportzeiten“ erwähnt. Diese beziehen sich jedoch auf eine Momentbetrachtung. Mit OSCAR soll eine Modellierung von Szenarios ermöglicht werden, jedoch keine Simulation über einen Zeitraum.

3 Marktüberblick

Da es sich bei den Anforderungen durchaus um eine Standardaufgabe aus der Transportlogistik handelt, war zunächst zu überprüfen, welche Systeme bereits am Markt sind und genutzt werden könnten.

Um diese Frage zu klären, wurde eine Marktanalyse mit Fokus auf die Open Source Community vorgenommen. In dieser wurde nach fertigen Lösungen bzw. partiellen Modulen geschaut, die in etwa den in Kapitel 2 beschriebenen Anforderungen entsprachen.

Die Marktanalyse ergab folgende Produkte, die im Rahmen dieses Papers nur sehr knapp in Bezug auf ihre Eignung für das Projekt charakterisiert werden sollen:

- **Inkscape**

Schwerpunkt von Inkscape [2] ist das grafische Editieren von .svg Dateien. Nach Einführung eigener xml-Schemata, die für eine benutzerfreundliche und anpassbare Umsetzung unerlässlich sind, ist die Nutzung der Software nicht mehr möglich.

- **Natural Resource Planner**

Natural Resource Planner [3] ist eine Webanwendung, die nicht ohne kontinuierlichen, leistungsstarken Internetanschluss genutzt werden kann. Nutzung im Projektkontext aus Gründen der vorgegebenen Infrastruktur nicht möglich.

- **QGIS**

QGIS [4] ist ein mächtiges, offenes Tool, welches alle Anforderungen abdeckt, jedoch als zu umfangreich und zu komplex für den Anwendungsfall bewertet werden muss.

- **Google Maps API**

Die Google Maps API [5] bietet Zugriff auf umfangreiches Kartenmaterial. Jedoch wird wiederum ein leistungsfähiger Internetzugang benötigt, welcher vor Ort nicht zur Verfügung steht. Daher kann diese API für unsere Zwecke nicht genutzt werden.

- **SVG-Edit**

SVG-Edit [6] ist wie Inkscape ein Editor für .svg Dateien. Aus denselben Gründen nicht nutzbar.

Die einzelnen Lösungen sind für sich gute Produkte, entsprachen aber nicht oder nur zum Teil unseren Anforderungen wie z.B. Persistenz, Komplexität, Rechenintensität oder Endbenutzerbestimmungen. Die Nutzung von fertiger Software ist also nicht möglich gewesen.

4 Eigenentwicklung

Die Recherche am Softwaremarkt ergab, dass es noch keine fertige oder erweiterbare Lösung gab, welche den Anforderungen gerecht wurde. Eventuell Einschränkungen und Anpassungen wurden abgeschätzt und als Ergebnis fiel die Entscheidung schließlich zugunsten einer kompletten Eigenentwicklung der Software aus.

4.1 Technologie

Die Anwendung ist in der Programmiersprache C# und dem .net Framework programmiert. C#.net zeichnete sich in einem Performancevergleich zwischen Java und C# als ressourcensparender und performanter aus.

OSCAR wurde mit dem WPF Framework aus dem Hause Microsoft [7] entwickelt. Es wurde bewusst darauf verzichtet, die Software als Webapplikation umzusetzen. Gründe hierfür ist die noch nicht vollständig entwickelte Telekommunikationsinfrastruktur in den Zielländern des Projektes.

- **Architektur**

Die Software wurde mit dem MVVM Entwurfsmuster umgesetzt. Dies soll den Produktlebenszyklus der Applikation verlängern, indem durch strikte Einhaltung von standardisierten Entwicklungsmethoden die Wart- und Erweiterbarkeit durch Drittentwickler ermöglicht wird.

OSCAR besteht aus drei Komponenten. Die erste Komponente stellt die Projektierungsebene dar. Hier werden Satellitenkarten eingeladen und Szenarien erstellt.

Die zweite Komponente ist die eigentliche Arbeitsoberfläche. Dem Anwender wird hier die eingeladene Karte angezeigt. Auf dieser Karte kann er anschließend per Drag-and-Drop-Mechanismus Objekte (z.B. Sammelstellen, Quellen, Senken, etc.) positionieren, verbinden und mit Werten belegen.

Die dritte Komponente ist die Stammdatenverwaltung. Hier werden unter anderem Siedlungstypen, Transportfahrzeuge, Abfallarten etc. definiert und parametrisiert, die im Anschluss auf der Arbeitsoberfläche als default-Belegungen für erzeugte Objekte genutzt werden können.

- **Datenhaltung**

Für die Datenhaltung sollte die freie und vom W3C empfohlene Spezifikation zur Beschreibung von Vektorgrafiken (SVG) genutzt werden. Hierbei stellte sich jedoch schnell heraus, dass sich die Stammdaten des Logistiktools nicht über die Standardspezifikation hinterlegen lassen.

Daraufhin wurde die Software mit einem eigens entwickeltem Serialisierer und Deserialisierer für XML-Dateien ausgestattet. Diese XML Dateien enthalten, in Verbindung mit der eingeladenen Karte, sämtliche Daten eines Projektes und können nach dem Export auf einem beliebigen anderen Rechner wieder eingeladen werden.

Die XML-Datei stellt damit auch eine einfache Schnittstelle zu Geographischen Informationssystemen oder Expertensystemen dar. Durch Verwendung des XML-Standards in OSCAR können andere Systeme mit OSCAR erstellte Projekte einlesen und mit Hilfe eines Parsers weiterverwenden.

- **Softwarekomponenten**

Um OSCAR erweiterbar zu halten, wurden die Layer des MVVM Patterns noch einmal in einzelne Module zerlegt. Die Module haben definierte Schnittstellen und können daher auf einfache Weise ausgetauscht werden. So ist es zum Beispiel möglich, die Komponente, welche für die Darstellung der Karte zuständig ist, gegen eine professionelle Tile-Map-Komponente auszutauschen ohne den Rest der Software anpassen zu müssen. Abbildung 1 zeigt das Komponentendiagramm der Anwendung mit allen Komponenten.

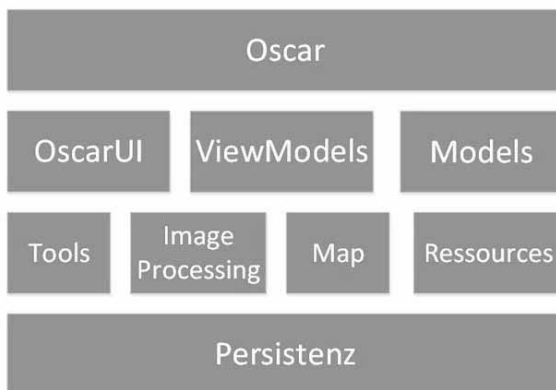


Abbildung 1. Komponentendiagramm

5 Beispielworkflow

In diesem Abschnitt wird ein Beispielworkflow dargestellt, welcher aufzeigt wie ein erstes Szenario angelegt und bearbeitet werden kann.

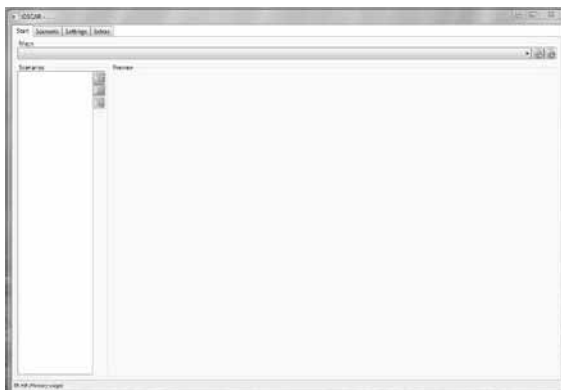


Abbildung 2. Startbildschirm

In Abbildung 2 ist der Startbildschirm dargestellt. Auf dessen Arbeitsfläche soll eine Karte bzw. ein Satellitenbild eingebunden und dargestellt werden. Dies ist mit der Erstellung eines Szenarios verbunden. Daher finden sich in der linken Menüleiste die entsprechen-

den Funktionen zum Laden, Neu-Erstellen und Löschen von Szenarien. Bereits erstellte Szenarien werden als Tre-View dargestellt.

Weiter Tabs sind diese Sicht sind Szenario, Settings und Extras.

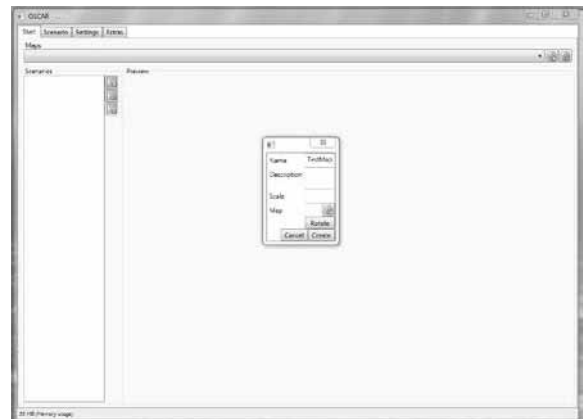


Abbildung 3. Karte einbinden

Der erste Schritt ist das Einbinden einer Karte (siehe Abbildung 3). Nach dem Eingeben eines Namens (z.B. den Namen einer Stadt), einer Beschreibung und des Maßstabs (optional), gelangt man durch Klicken des kleinen blauen Buttons zum Windows Explorer und kann nun die gewünschte Karte über den Pfad zum Speicherort einbinden.



Abbildung 4. Szenario erstellen

Nachdem die gewünschte Karte eingebunden ist, ein neues Szenario erstellt und geladen ist, läßt sich dieses nun entsprechend bearbeiten (siehe Abbildung 4). Mit Hilfe der Buttons (oben rechts) lassen sich nun interaktiv Gebiete, Quellen, Sammelpunkte und Behandlungsstellen eintragen.



Abbildung 5. fortgeschrittenes Szenario mit Verbindungen

Abbildung 5 ist ein Beispiel für ein bereits fortgeschrittenes Szenario. Zu sehen sind drei urbane Areale mit jeweils einer Quelle, welche mit einem Sammelpunkt und dieser wiederum mit einer Behandlungsstelle verbunden ist. Alle Gebietstypen wie z.B. *Area* lassen sich unter *Settings* mit weiteren Informationen versehen.

Nachdem ein Szenario modelliert wurde, müssen den gesetzten Objekten Eigenschaftswerte zugewiesen werden. Diese kommen entweder aus den Stammdaten (s. Abschnitt 6) oder werden manuell eingegeben.

6 Stammdatenverwaltung

Die Software OSCAR verfügt über eine Stammdatenverwaltung. In dieser können Objekte der verschiedenen Typen angelegt sowie default Properties zugewiesen werden. In der grafischen Modellierung des Szenarios kann auf diese Stammdaten zurückgegriffen werden. So kann beispielsweise einer gerade gesetzten *Area* sofort der Typ „Industrie-Gebiet“ zugewiesen werden. Abbildung 6 zeigt drei Beispieltypen für Transportfahrzeuge in der Stammdatenverwaltung.

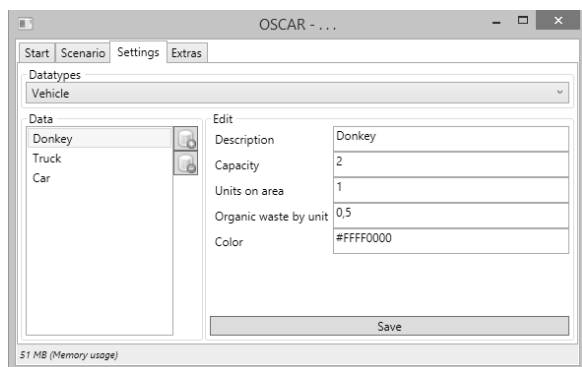


Abbildung 6. Stammdatenverwaltung

In der Stammdatenverwaltung können folgende Objektklassen gepflegt werden:

- Gebiete
- Quellen
- Behandlungsstellen
- Kunden
- Transportmittel
- Arbeiter

7 Transportwegberechnung

Die Verbindungen zwischen den Objekten können durch beliebig viele Punkte spezifiziert werden. Das bedeutet, dass mit Hilfe dieser Punkte die Verbindung dem Straßenverlauf folgen kann. Die Gesamtlänge der Strecke kann hierdurch durch einfaches aufaddieren der gezeichneten Pixel zusammen mit der Kartenskaliierung berechnet werden. Für Grobabschätzung kann aber auch einfach die Luftlinie gewählt werden.

Jedoch ist nicht jede Straße gleich beschaffen. Während eine Hauptstraße von vielen Fahrzeugen gleichzeitig in hoher Geschwindigkeit befahren werden kann, muss das bei Feldwegen nicht der Fall sein. Aus diesem Grunde wird den Strecken ein Verbindungskoeffizient zugewiesen. Dieser wirkt sich direkt auf die Durchschnittsgeschwindigkeit der genutzten Transportmittel aus. Abbildung 7 zeigt einen Auszug aus einem Stammdatenreport bezüglich der genutzten Fahrzeuge, deren Kapazitäten und den Durchschnittsgeschwindigkeiten.

Vehicles	Capacity [kg/vehicle]	Operating range [km]	Average speed [km/h]	Cost per vehicle [IDR/ (vehicle day)]	Max transport- time per vehicle per day [h/day]	Number workers of type work [Worker s/type]	Type worker
Hand cart	150	3	3	500	10	1	L
Small pickup truck	500	8	15	5000	10	1	D/C
Farm tractor and trailer	2500	12	20	15000	8	1	D/C
Open truck	10000	50	40	50000	8	1	C D

Abbildung 7. Auszug Stammdaten – Fahrzeuge

Aus den gewichteten Strecken und den Fahrzeugstammdaten, lassen sich nun in Abhängigkeit mit den Arbeiterstammdaten, Verbindungskosten berechnen.

8 Auswertung

Oscar bietet dem Nutzer Möglichkeiten das modellierte Szenario auszuwerten. Zuerst ist hier die Möglichkeit zu nennen, sich die Auslastungen der einzelnen Stellen anzeigen lassen zu können. Per Mausklick sieht der Bearbeiter beispielsweise, ob die Summe des

Auskommens aller verbundenen Quellen die Gesamtkapazität einer Sammelstelle überschreitet. In diesem Fall müsste eine weitere Sammelstelle angelegt oder die vorhandene Kapazität erhöht werden.

Eine weitere Möglichkeit des Auswertens ist die monetäre Auswertung des Szenarios. Zunächst wird zwischen 4 verschiedenen Kostenaufkommen unterschieden:

- Transportkosten (K_{VG})
Die unter Abschnitt 7 beschriebenen Kosten
- Sammelkosten (K_{SG})
Kosten die beim Sammelvorgang innerhalb einer Flächenquelle entstehen können.
- Lagerkosten (K_{LG})
Kosten die für das Betreiben der Sammelstellen anfallen
- Wiederverwertungskosten (K_{WG})
Hierunter fallen Kosten die zum Betrieb der Verwertungsstellen notwendig sind.

Durch diese Einzelkosten berechnet OSCAR die Gesamtkosten des Szenarios.

$$K_{GG} = K_{VG} + K_{SG} + K_{LG} + K_{WG}$$

Zur vollständigen Auswertung stellt OSCAR Gesamtkosten und Gesamteinnahmen gegenüber. Die Einnahmen des Systems entsprechen den Verkaufserlösen für Kompost, Protein oder Energie durch die Senken. Die Gesamt-Einnahmen (EG) in [Geldeinheit/Tag] sind die Summe der einzelnen Einnahmen (E) in den N Senken. Die Gesamteinnahmen berechnet OSCAR wie folgt:

$$E_G = \sum_{i=0}^N (E)_i$$

9 Ausblick

Die beschriebene Software wurde in ihrer ersten Version im August 2014 den Anwendern übergeben und kontinuierlich weiterbetreut und –entwickelt. Durch den Einsatz im beschriebenen Projekt in Indonesien erwarten wir uns Erfahrungen unter realen Bedingungen und ein Feedback, das direkt auf die Weiterentwicklung der Software wirken kann.

Darüber hinaus werden weitere Einsatzgebiete des Tools eruiert. Wegen seiner Modularität und Anpassbarkeit sind auch Einsatzgebiete geplant, die keinen Bezug zur Abfalllogistik aufweisen. Genaugenommen lässt sich der Ansatz nämlich auf jedes Netz übertragen, für dessen Knoten ein Raumbezug notwendig ist.

In Aussicht steht die Anwendung im Bereich der Infrastrukturplanung in Großstädten für die Analyse und Optimierung eines Abwassernetzes. Eine erste Abschätzung des Aufwandes macht klar, dass OSCAR in diesen Fachkontext sehr einfach übertragbar ist, wenn die Attributierung der Knoten an die geänderte Fachschale angepasst wird. Dies allerdings ist durch die modulare Systemarchitektur ohne größeren Aufwand umsetzbar.

Damit zeigt sich OSCAR als einfaches und universell einsetzbares Werkzeug zur Logistikplanung. Ziel ist nicht die detaillierte Analyse und Optimierung, sondern vielmehr eine Grobabschätzung von Planungsszenarien. Eine solche Grobplanung allerdings lässt sich mit dem vorgestellten Softwareprodukt mit überschaubarem Aufwand realisieren.

10 References

- [1] Project FORWARD, Eawag. Weitere Informationen unter http://www.eawag.ch/forschung/san-dec/gruppen/MSWM/projects_mswm/forward/index_EN
- [2] Inkscape (n.d.) [online]. Verfügbar unter: <http://www.inkscape.org> [15.07.2014]
- [3] Natural Resource Planner (n.d.) [online]. Verfügbar unter: <http://kars.ku.edu/maps/naturalresource-planner>
- [4] QGIS (n.d.) [online]. Verfügbar unter: <http://www.qgis.org>
- [5] Google Maps API (n.d.) [online]. Verfügbar unter: <https://developers.google.com/maps>
- [6] SVG-Edit (n.d.) [online]. Verfügbar unter: <https://code.google.com/p/svg-edit>
- [7] Microsoft WPF Framework (n.d.) [online]. Verfügbar unter: <http://msdn.microsoft.com/de-de/library/ms754130%28v=vs.110%29.aspx>

System-Dynamics-Diagramme als “Physical Modeling” - eine Frage der Kausalität

Peter Junglas¹

¹ FHWT Vechta/Diepholz/Oldenburg
peter@peter-junglas.de

System-Dynamics-Diagramme gelten als sehr einfaches Modellierungswerkzeug, das sich leicht mit traditionellen Methoden nachbilden lässt. Wie einige Beispiele zeigen, ist die Situation jedoch komplizierter: Die Kausalität - also die Frage, welche Anschlüsse eines Blocks Ein- oder Ausgänge sind - ist in manchen Fällen vom Zustand des Systems abhängig. Die Entwicklung der hier vorgestellten Modelica-Bibliothek erforderte daher einige Vorüberlegungen, die im weiteren ausführlich dargelegt werden.

1 Einleitung

Die System-Dynamics-Methode ist ein Modellierungsverfahren, das vor allem in nicht-technischen Gebieten wie der Ökonomie oder der Ökologie verwendet wird [1]. Kommerzielle Programme zur Erstellung und Simulation von System-Dynamics-Diagrammen sind weit verbreitet (z. B. Stella von isee systems [2]), es gibt aber auch eine frei verfügbare Implementation, die auf Modelica basiert [3].

Angesichts der grundsätzlich sehr einfachen Struktur der Diagramme könnte man annehmen, dass sie sich leicht durch geeignete Blöcke etwa in Simulink implementieren lassen. Es stellt sich aber heraus, dass in einigen Situationen die Signalflussmethode nicht gut zur Nachbildung geeignet ist. Der Grund dafür ist, dass die Kausalität der Blöcke - also die Frage, welche Anschlüsse Eingänge bzw. Ausgänge sind - nicht immer durch das Diagramm festgelegt ist, sondern sich abhängig vom Zustand des betrachteten Systems ändern kann. Für die Erstellung einer System-Dynamics-Bibliothek eignen sich daher auf “Physical Modeling” basierende Methoden besser, bei denen die Kausalität eines Blocks sich erst dynamisch im Kontext des gesamten Systems ergibt.

Im Folgenden wird das grundlegende Problem an einfachen Beispielen erläutert und eine in Modelica programmierte System-Dynamics-Bibliothek vorgestellt. Sie wurde für das Lehrbuch [4] erstellt und

kann von der Homepage des Autors frei heruntergeladen werden [5]. Die bereits vorhandene Modelica-Implementierung von Cellier et al. [3] beachtet die hier betrachteten Effekte nicht, da sie vor allem im Hinblick auf die bekannten Weltmodelle erstellt worden ist, bei denen keine Kausalitätsprobleme auftreten.

2 Grundlegende System-Dynamics-Diagramme

System-Dynamics-Diagramme setzen sich i. W. aus nur drei verschiedenen, aber sehr allgemeinen Grundbausteinen zusammen: Die Zustandsgrößen werden als Speicherelemente (**Reservoirs**) dargestellt, die ihren Wert durch Zufluss bzw. Abfluss ändern. Ventilbausteine (**Flows**) bestimmen die Stärke der Flüsse, sie verlaufen zwischen Reservoiren oder externen Quellen und Senken. Zur Berechnung können sie auf die Werte von Zwischengrößen zurückgreifen, die mit **Convertern** bestimmt werden. Abb. 1 zeigt diese Blöcke im Zusammenhang.

Als Beispiel werde ein einfaches Modell zur Bevölkerungsentwicklung betrachtet: Die Bevölkerungsgröße N ändert sich durch die Zahl G der Geburten pro Zeit und die Zahl T der Todesfälle pro Zeit. Für G wird eine konstante Rate g angenommen, während T zu-

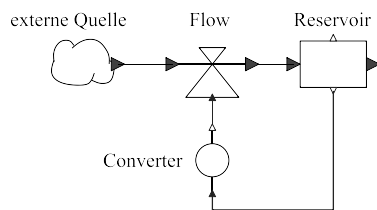


Abbildung 1: System-Dynamics-Basisblöcke

sätzlich eine feste Kapazitätsgrenze N_k enthält:

$$G = gN$$

$$T = tN \quad \text{mit} \quad t = \frac{t_0}{1 - N/N_k}$$

Im fertigen Modell (s. Abb. 2) werden die Parameter g , t_0 und N_k jeweils in Convertern bereitgestellt, ein weiterer Converter berechnet daraus die Rate t . Die Flows schließlich multiplizieren einfach ihre beiden Eingänge, um die Flüsse G bzw. T zu erhalten. Das Diagramm zeigt nur die grundsätzlichen Zusammenhänge zwischen den Größen, die konkreten Formeln sind implizit in den Blöcken enthalten.

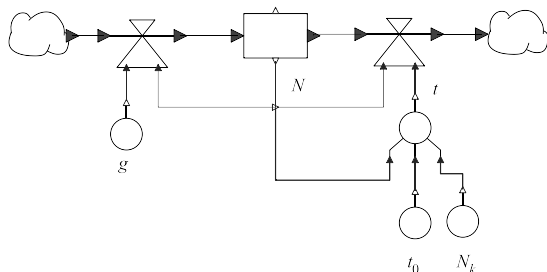


Abbildung 2: Modell bevölkerung

Die Kausalität der Bausteine ist hier ganz klar: Converter haben mehrere Eingänge und einen Ausgang, die untereinander oder mit entsprechenden Eingängen der Flows verbunden werden. Ein Flow berechnet daraus den Wert des Flusses und gibt ihn als Ausgang an die angeschlossenen Reservoirs weiter. Dabei darf man sich durch die Pfeile im Diagramm nicht verwirren lassen: Diese zeigen nur die (positive) Richtung der Flüsse an, die Signalflüsse dagegen verlaufen immer von den Flows zu den Reservoirs. Diese subtrahieren schließlich ihre beiden Eingangswerte und bestimmen durch einfache Integration den Wert der Zustandsgröße, der zusätzlich als Ausgangswert bereitgestellt wird. Dieses Prinzip wurde bei der Modelica-

Bibliothek von [3] verwendet und lässt sich leicht auch in Simulink implementieren.

3 Modelle mit variabler Kausalität

3.1 Reservoir mit Sättigung

Ein Reservoir-Baustein kann optional einen Minimal- und einen Maximalwert angeben. Im Beispielmodell abfluss (s. Abb. 3) hat das erste Reservoir S_1 einen Minimalwert von 0 und einen Startwert von 4, der abgehende Fluss ist konstant 0.5. Das Ergebnis der Si-

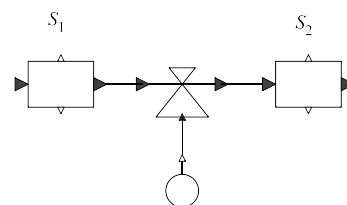


Abbildung 3: Modell abfluss

mulation zeigt Abb. 4: Aufgrund des konstanten Abflusses nimmt der Wert von S_1 linear ab, bis bei $t = 8$ der Minimalwert 0 erreicht ist, und bleibt dann konstant. Das nachgeschaltete Reservoir S_2 verhält sich gegenläufig, insbesondere bleibt auch sein Wert ab $t = 8$ konstant.

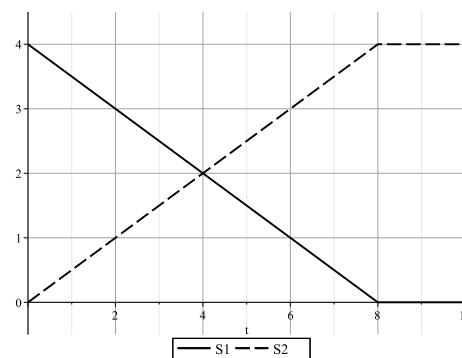


Abbildung 4: Ergebnis von abfluss

Auf den ersten Blick scheint es, als ließe sich dieses Modell leicht in Simulink nachbilden, indem man bei den Integratoren Saturation limits einführt. Aber

das Problem liegt tiefer: Zwar lässt sich der Wert von S_1 leicht auf 0 beschränken, aber das folgende Reservoir S_2 steigt trotzdem immer weiter an. Man muss vielmehr dafür sorgen, dass ab $t = 8$ der Fluss aus S_1 und damit auch in S_2 verschwindet. Es liegt hier ein Kausalitätsproblem vor: Bis $t = 8$ bestimmt der Flow-Baustein die Größe des Flusses, danach aber legt S_1 den Fluss auf 0 fest.

3.2 Modellierung einer Fertigungsmaschine mit Oven

Besonders deutlich wird das Problem beim Block Oven, einem diskreten Modell für eine generische Fertigungsmaschine. Er hat die drei Parameter `initialLoad`, `capacity` und `cookingTime` und verhält sich wie ein einfaches Backblech: Er wird zunächst gemäß seines Eingangsflusses beladen, bis die Kapazität erreicht wurde. Anschließend beginnt die Verarbeitungszeit, an deren Ende der gesamte Inhalt als Ausgangsfluss erscheint. Das grundsätzliche Verhalten des Blocks zeigt das Modell `oven1` (s. Abb. 5) mit den Parametern `initialLoad = 0`, `capacity = 3` und `cookingTime = 2`. Der Eingangsfluss ist auf den konstanten Wert 2 eingestellt, der Ausgangsfluss auf den Wert 1.

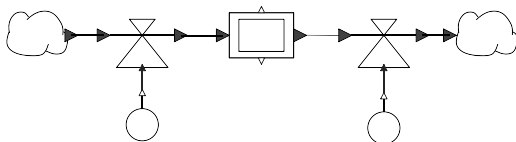


Abbildung 5: Modell oven1

Der Plot in Abb. 6 zeigt oben den Eingangs- und Ausgangsfluss, unten den Beladungszustand des Ovens. Zur Zeit $t = 2$ gelangen zunächst zwei Einheiten, dann nur noch eine in den Oven. Bei $t = 3$ ist er voll beladen, die Verarbeitungszeit beginnt. Danach, bei $t = 5$, werden drei Einheiten herausgegeben, gleichzeitig kommen schon die ersten zwei neuen Teile an. Das genaue Timing, insbesondere die Überlappung, sind natürlich diskutabel. Als Vorbild diente das Verhalten des entsprechenden Blocks im Programm Stella.

Die Größe des Eingangsflusses hängt hier in komplizierter Weise vom vorgeschalteten Flow-Baustein und vom Zustand des Ovens ab: Während der Beladungs-

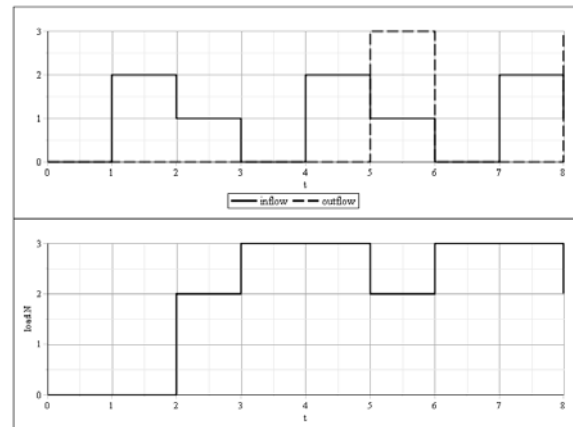


Abbildung 6: Ergebnis von oven1

phase gibt der Flow den Wert vor, bei Erreichen der Kapazität arbeiten Zustand des Ovens und Flow-Wert zusammen, während der Bearbeitungsphase setzt der Oven den Eingangsfluss auf Null. Der Ausgangsflow wird dagegen komplett vom Zustand des Oven bestimmt: Während der Beladungs- und Bearbeitungsphase ist er Null und steigt nur während des Ausladens auf den Wert `capacity`. Der im nachgeschalteten Flow-Baustein vorgegebene Wert wird völlig ignoriert.

Noch komplizierter wird es, wenn man das Modell `oven1` um Reservoir-Blöcke S_1 und S_2 an Ein- bzw. Ausgang erweitert: Wenn S_1 leer läuft – und nach unten auf Null beschränkt ist –, ergibt sich sein Ausgangsfluss durch seinen Inhalt, die (Maximal-)Größe des Flusses und den Beladungszustand des Ovens. Ergebnisse eines solchen Simulationslaufs zeigt Abb. 7.

Und sollte das Ausgangsreservoir nach oben beschränkt sein, bricht die Simulation zusammen: Der Oven will seinen Inhalt abgeben, aber das Ausgangslager hat nicht mehr genug Platz. Natürlich ist das eine Situation, die man nicht nur im Modell vermeiden will.

3.3 Modellierung von Oven in Simulink

Dass die Kausalität der Verbindungen sich mit dem Zustand ändern kann, heißt natürlich nicht, dass sich ein solches Modell mit der Signalflussmethode, also etwa in Simulink, nicht erstellen ließe. Allerdings

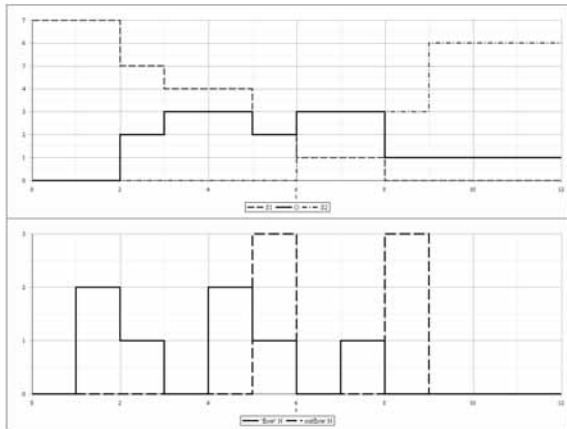


Abbildung 7: Ergebnis von oven2

muss man die Kausalitätsprobleme hier selbst lösen. Dazu erhält der Flow-Block neben seinem Eingang für den gewünschten Flow-Wert zwei weitere Eingänge, die die maximal möglichen Flüsse der beiden benachbarten Reservoirs bekommen. Durch einfache Bildung des Minimums wird daraus der tatsächliche Fluss bestimmt. Auch der Block Oven lässt sich mit etwas Mühe mit Bausteinen der Standardbibliothek aufbauen, z. B. indem man seinen Modelica-Code nachbildet. Er hat einen Eingang für den Eingangsfluss sowie zwei Ausgänge für den Ausgangsfluss und den aktuellen Beladungszustand. Dazu kommt ein weiterer Ausgang, der den momentan möglichen maximalen Eingangsfluss angibt. Daraus lässt sich leicht das Modell oven1 erstellen (s. Abb. 8), das die obigen Werte reproduziert.

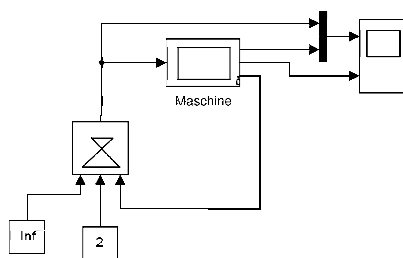


Abbildung 8: Modell oven1 in Simulink

Das Modell lässt sich leicht um zwei Standard-Reservoirs erweitern, wobei das Eingangslager in analoger Weise seinen Restbestand an den nachgeschalteten Flow meldet. Allerdings tritt hier ein Problem am Ausgang des Oven-Blocks auf: Er gibt seinen

Ausgangsfluss in voller Größe ab, auch wenn der Flow einen kleineren Wert vorgesehen hat. Um dies nachzubilden, enthält der Flow-Baustein als Parameter die Option, bei der Berechnung des Flusses seinen "Wunschwert" einfach zu ignorieren. Das fertige Modell (s. Abb. 9) reproduziert wieder die Ergebnisse von oben.

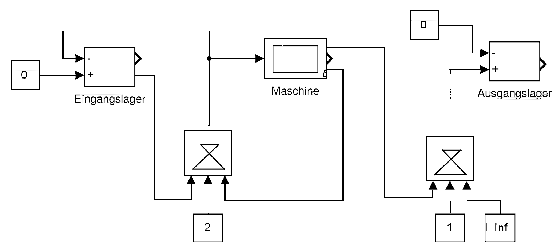


Abbildung 9: Modell oven2 in Simulink

Trotz dieser Erfolge erscheinen die Simulink-Modelle sehr adhoc, sie lassen die Einfachheit und Flexibilität der ursprünglichen System-Dynamics-Diagramme vermissen. Außerdem entsteht aufgrund der starren Regeln für die Anordnung von Ein- und Ausgängen am Block ein ziemlicher "Drahtverhau" an Signalleitungen. Eine etwas einfachere, aber weniger flexible Implementierung der hier betrachteten Beispiele wird in [4] beschrieben.

4 Modelica als Basis einer System-Dynamics-Bibliothek

4.1 Bibliotheken in Modelica

Die Frage der Kausalität stellt sich in auf Modelica basierenden Physical-Modeling-Systemen in ganz anderer Weise [6]: Statt festzulegen, welche Größen als Eingangswerte verwendet werden, um damit andere als Ausgangswerte zu berechnen, werden für jeden Block nur die verknüpfenden Gleichungen angegeben, ohne dass sie nach bestimmten Größen aufgelöst sind. Zusätzliche Gleichungen ergeben sich bei Verbindungen der Blöcke. Dazu unterscheidet man zwei Typen von Größen: Flow-Variable addieren sich an Verbindungspunkten zu Null – sie entsprechen oft Zeitableitungen von Erhaltungsgrößen –, Potential-Variable haben an einer Verbindung einen konstanten Wert.

Wie sich das aus diesen Gleichungen schließlich ergebende System – häufig ein DAE-System – simulieren lässt, ist ein schwieriges Problem, das zum Glück inzwischen gut verstanden ist [7]. Die dazu entwickelten Algorithmen wurden in Modelica-basierten Simulationsprogrammen wie Dymola oder MapleSim implementiert.

Die in Modelica erreichte hohe Flexibilität hat allerdings vor allem bei der Entwicklung von Block-Bibliotheken ihren Preis: Da nicht festgelegt wird, welche Größe wo berechnet wird, ist nicht automatisch gewährleistet, dass die Verknüpfung beliebiger Blöcke zu einem geschlossenen System führt, bei dem die Zahl der Gleichungen und Variablen gleich groß sind. Abhilfe schafft das in [8] vorgestellte Verfahren: Man definiert an den Verbindungspunkten (**Connectoren**) jeweils gleich viele Variable vom Flow- und Potential-Typ und gibt jedem Block so viele Gleichungen, wie er externe Flow-Variablen hat. Darüber hinaus kann man noch "gewöhnliche" Signalleitungen definieren, die als Ein- oder Ausgänge gekennzeichnet sind. Hier benötigt jede Ausgangsgröße natürlich eine entsprechende Gleichung im Block.

4.2 Konzeption des Connectors

Diese Überlegungen sollen nun beim Aufbau einer System-Dynamics-Bibliothek in Modelica angewendet werden. Startpunkt ist dabei die Definition eines geeigneten Connectors `MassPort`, der die Größe eines Flusses als Modelica-Flow-Variable `dm` enthält, entsprechend der integrierten Zustandsgröße `m`. Zwei Fragen müssen nun geklärt werden:

1. Welche Größe kommt als `dm` begleitende Potential-Variable in Frage?
2. Wie werden die Gleichungen auf die Blöcke, insbesondere auf Reservoirs und Flows, aufgeteilt?

Motiviert von der in Abschnitt 2 vorgestellten Grundsystematik von System-Dynamics-Modellen soll die Integration des Flusses im Reservoir geschehen, das dazu folgende Gleichung enthält:

$$\text{der}(m) = \text{inflow.dm} + \text{outflow.dm};$$

Dabei ist, wie in Modelica üblich, eine Flow-Variablen positiv, wenn der Fluss in den Block einströmt.

Ein Flow-Block dagegen berechnet den Wert von `dm` mit Hilfe seiner Eingangsgrößen. Dabei muss er aber, wie in Abschnitt 3 deutlich wurde, auch die Zustände der angeschlossenen Reservoirs berücksichtigen. Daher schicken diese die benötigten Informationen als reelle Variable `data`, die die Rolle der Potential-Variablen übernimmt. Angesichts der oben vorgestellten Beispiele gibt es drei Möglichkeiten, wie der Wert von `data` verwendet werden kann:

1. gar nicht, das Reservoir übernimmt den vom Flow vorgegebenen Wert,
2. der Flow wird auf `data` gesetzt,
3. der Flow wird durch `data` begrenzt.

Fall 1 entspricht der "Standard-Situation", Fall 2 z. B. dem Ausgang des Oven, Fall 3 tritt bei Reservoirs mit Sättigung oder bei der Beladung eines Oven auf.

Dieses Verhalten ließe sich implementieren, indem man im 1. Fall `data = 0` setzt und die beiden anderen Fälle durch das Vorzeichen von `data` unterscheidet. Dies verhindert aber die Möglichkeit, in zukünftigen Erweiterungen neue Verwendungsmöglichkeiten von `data` vorzusehen. Außerdem ist der Test auf Null bei reellen Variablen generell eine schlechte Idee. Daher wird der Connector um eine ganzzahlige Variable `info` erweitert, mit der ein Reservoir den entsprechenden Fall anzeigt. Diese Größe hat eine fest vorgegebene Kausalität: Sie wird im Reservoir berechnet und im Flow verwendet. Es müssen also zwei Varianten des Connectors verwendet werden, bei denen `info` als `output` bzw. `input` gekennzeichnet ist:

```
connector MassPortR
  "mass port of reservoirs"
  flow Real dm;
  Real data;
  output Integer info;
end MassPortR;
```

```
connector MassPortF
  "mass port of flows"
  flow Real dm;
  Real data;
  input Integer info;
end MassPortF;
```

4.3 Aufbau der System-Dynamics-Bibliothek

Nach diesen Vorüberlegungen lässt sich nun eine System-Dynamics-Bibliothek ohne Probleme aufbauen. Sie enthält die vier Unterpakete:

- Interfaces
- Reservoirs
- Converters
- Flows

In `Interfaces` befinden sich die Definitionen der Connectoren, Basisklassen und Hilfsfunktionen. Sie enthält insbesondere die Funktion `constrainedRate`, die den von einem Fluss vorgegebenen Wert mit den `data`- und `info`-Werten zweier `MassPorts` kombiniert, um den tatsächlichen Fluss zu bestimmen. Diese Berechnung wird über die Basisklasse `GenericFlow` in alle Flow-Bausteine integriert.

`Reservoirs` enthält als Standard-Reservoir die Blöcke `Stock` und `SaturatedStock` sowie `CloudSource` und `CloudSink` für externe Quellen bzw. Senken. Zum besprochenen Block `Oven` kommt noch `Conveyor` hinzu, der ein einfaches Modell für ein Transportband darstellt. Beide sind zeitdiskret, sie arbeiten mit vorgegebener `Samplerate`. Ergänzend gibt es auch diskrete Versionen `StockD` und `SaturatedStockD`.

Die Blöcke in `Flows` enthalten neben zwei `MassPorts` ggf. zusätzliche Signal-Eingänge zur Berechnung der Flüsse aus Hilfsgrößen. Die Blöcke in `Converters` sind ausschließlich mit Signal-Anschlüssen versehen, sie sind daher in Modelica als `block` definiert worden. In auf System-Dynamics-Diagramme spezialisierten Programmen gibt es in der Regel nur je einen `Converter`- und `Flow`-Block. Die benutzten Beziehungen werden als Parameter der Blöcke definiert und die Zahl der Eingänge entsprechend angepasst. Leider bieten die auf Modelica basierenden Programme keine solche Möglichkeit. Daher wurden für die häufigsten Verknüpfungen fertige Blöcke in den entsprechenden Unterbibliotheken bereitgestellt. Eine Besonderheit stellen `SwitchedConverter` und `TimeSwitchedConverter` dar, die in Abhängigkeit

von der Eingangsgröße oder der Zeit zwischen zwei Werten umschalten, sowie der `GraphConverter`, der einen funktionalen Zusammenhang durch Interpolation von Tabellenwerten herstellt, die aus einer Datei eingelesen werden. Weitere Blöcke lassen sich durch Ableiten von Basisblöcken mit wenigen Zeilen Modelica-Code erstellen.

5 Schlussfolgerungen

Für die Entwicklung einer flexiblen System-Dynamics-Bibliothek sind die im Physical Modeling gegebenen Möglichkeiten einer variablen Kausalität zumindest sehr hilfreich. Allerdings hätte man die im Connector verwendeten Flow- und Potential-Größen auch komplett als Signalgrößen definieren können. Die Frage der Kausalität hängt also auch davon ab, was man als "eine" Signalleitung auffasst. Auf jeden Fall eignen sich die hier vorgestellten Ideen gut dazu, in der Lehre den Studierenden einen ersten Einblick in die Problematik der Kausalität zu geben.

Die konkrete Umsetzung leidet momentan an Beschränkungen der Tools, es fehlt insbesondere die Möglichkeit, einen funktionalen Zusammenhang als Parameter eines Blocks einzugeben. Auch das im Detail unterschiedliche Verhalten des `pre`-Operators in Dymola und MapleSim und des `1/z`-Blocks in Simulink erschwerte die Arbeit nicht unerheblich. Hier besteht wohl noch ein gewisser Klärungs- bzw. Verbesserungsbedarf.

Zwar sind für die wichtigsten Zusammenhänge fertige Blöcke in der System-Dynamics-Bibliothek vorhanden, weitere lassen sich durch Ableiten vorhandener Blöcke mit wenigen Zeilen Modelica-Code leicht erstellen. Der typische Anwender von System-Dynamics-Software hat aber wenig bis gar keine Programmiererfahrung und ist von anderen Tools ein einfacheres Vorgehen gewöhnt. Für den Einsatz in der Lehre, speziell im Kontext verschiedener Modellierungsverfahren, ist diese Einschränkung aber nicht gravierend.

Literatur

- [1] B. Hannon und M. Ruth. *Dynamic Modeling*. Springer, New York, 2. Auflage 2001.
- [2] B. Richmond, S. Peterson und P. Vescuso. *An Academic User's Guide to STELLA*. High Performance Systems, Inc., Lyme, N.H., 1987.
- [3] F. E. Cellier. *World3 in Modelica: Creating System Dynamics Models in the Modelica Framework*. Proceedings of the 6th International Modelica Conference, Bielefeld, Germany, S. 393 – 400, 2008.
- [4] P. Junglas. *Praxis der Simulationstechnik*. Europa-Lehrmittel, Haan-Gruiten, 2014.
- [5] P. Junglas. *Homepage von "Praxis der Simulationstechnik"*. Online: <http://www.peter-junglas.de/fh/publications/simulation/index.html> (Aufruf 2014-06-13).
- [6] P. A. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley & Sons, New York, 2004.
- [7] F. E. Cellier und E. Kofman. *Continuous System Simulation*. Springer, New York, 2010.
- [8] H. Olsson, M. Otter, S. E. Mattsson und H. Elmqvist. *Balanced Models in Modelica 3.0 for Increased Model Quality*. Proceedings of the 6th International Modelica Conference, Bielefeld, Germany, S. 21 –33, 2008.

Virtual Stochastic Sensors for Hybrid Systems: Mutual Influence between Continuous and Discrete System Parts

Claudia Krull¹, Graham Horton¹
¹Otto-von-Guericke-Universität Magdeburg
claudia.krull@ovgu.de

Virtual Stochastic Sensors can reconstruct the unobservable behavior of discrete stochastic systems based on time-stamped output protocols. Previously we extended the underlying modeling paradigm by continuous reward values. Measurement protocols of these values were then used for the reconstruction of the discrete model behavior. The current paper goes one step further by enabling the continuous model part to influence the discrete behavior through arbitrary guard functions, which can enable or disable activities in the discrete model part. Thereby making the model truly hybrid. Two examples from medicine and the renewable energy sector are taken to exemplify the potential and current limitations of the paradigm. Solely based on infrequent measurements of one continuous value, are we able to reconstruct likely behavior of a complex hybrid model. This enlarges the application domain of Virtual Stochastic Sensors to many real life problems with hybrid properties.

1 Introduction

Most real world systems are of hybrid nature, exhibiting discrete as well as continuous behavior. Most often these two influence each other, requiring a powerful modeling paradigm to capture these dynamics accurately. To complicate matters further, the systems are often only partially observable, due to infrequent measurements, sparse sensor coverage or actually hidden parts.

Fluid stochastic Petri nets (FSPNs) [1] and stochastic reward nets (SRNs) [2] are two complex hybrid modeling paradigms. However, one has to be able to observe and analyze all processes involved in the real system, in order to accurately represent the system in one of these paradigms. Simulation and behavior prediction of partially observable systems becomes more and more vague, as the hidden fraction increases. We therefore propose using Virtual Stochastic Sensors to perform actual reconstruction of past behavior, based on measurements obtained from the real system.

Virtual Stochastic Sensors have been successfully applied to behavior reconstruction of discrete stochastic systems ([3]). The underlying modeling paradigm

Hidden non-Markovian Models has been extended to incorporate continuous reward values ([4]). As of now, we were not able to model influence of the continuous on the discrete model part.

In this paper we extend the paradigm to truly hybrid systems, by enabling the values of the continuous quantities to influence the enabling of discrete model transitions. This is done by incorporating the well-known concept of guard functions, which can be arbitrary logical expressions based on the current model state.

Two examples are taken to illustrate the modeling power and application potential of the extended modeling paradigm. One is taken from the renewable energy sector, the other from the medical domain. Using VSS we are able to reconstruct the dynamics of a small solar thermal energy system and its dependence on optional external heating, as well as the behavior of a type-one diabetic based on his blood glucose protocol and thus his adherence to his diet and injection schedule.

The proposed extension opens up a new class of systems for behavior reconstruction via VSS, making the technology applicable to real world problems of vari-

ous types.

2 Related and Previous Work

2.1 Hybrid Modeling Paradigms

There exist a number of hybrid modeling paradigms. The two presented here are both based on stochastic Petri nets (SPNs). They are chosen as representatives, because the HnMM paradigm borrows heavily from SPNs. A comprehensive introduction to stochastic Petri nets can be found in [5].

2.1.1 Fluid Stochastic Petri Nets

Fluid stochastic Petri nets were first introduced in [1] and have been considerably extended since then [6]. They extend SPNs by adding fluid places and fluid arcs. Fluid places can contain a positive real-valued amount of fluid. Fluid arcs connect transitions and fluid places. When one such transition is enabled, the fluid arc can fill or drain the fluid place with a constant flow rate. Later on, flush-out arcs were introduced that can change a fluid level instantaneously. Besides representing actual fluids, FSPNs are widely used in performance and reliability modeling, where the measures of interest are depicted as a fluid level. They are also used to abstract a potentially large number of discrete tokens via a continuous value. FSPNs enable the discrete and continuous model part to influence each other.

2.1.2 Stochastic Reward Nets

Stochastic reward nets (SRN) extend SPNs by adding external continuous measures [2]. Rate rewards may change these values continuously, based on the current system state. Impulse rewards can change the values instantaneously upon the completion of a system activity (firing of a transition). However, the value of the reward measures do not influence the behavior of the discrete system. Like FSPNs, SRNs are often used in reliability and performance analysis.

2.2 Virtual Stochastic Sensors

Virtual stochastic sensors have been introduced in [7] and can reconstruct unobserved behavior of discrete stochastic systems, based on observable system output. VSS are related to virtual sensors, which are used to determine a quantity of interest that is not or not easily measurable from values which are easier to obtain [8]. The quantity of interest can often be computed directly from the sensor readings, due to a well-formed relationship. In VSS, this relationship is stochastic, and can be expressed using so-called Hidden non-Markovian Models, which were first introduced in [9]. In HnMM the unobserved system is described by a discrete stochastic model, and the measurable quantities are represented by output symbols, which are emitted via a second stochastic process based on the system states or transitions.

HnMM in turn are based on the well known Hidden Markov Models [10]. These doubly stochastic processes use a discrete-time Markov chain (DTMC) [11] as hidden model, and generate an output symbol in every step depending on the current state of the DTMC. HnMM enable the state transitions to be described by arbitrary continuous distribution functions, and allow the concurrent enabling of multiple non-Markovian activities. This enables behavior reconstruction for a much larger class of systems compared to HMM and extensions [10].

In [4] hidden non-Markovian reward models (HnMRM) were introduced, which extend the discrete HnMM paradigm by continuous reward measures, as they are used in SRNs. This broadens the application range to include some hybrid systems. However, the discrete model part influences the continuous part, but not vice versa. The current paper closes this gap, and extends the paradigm to truly hybrid systems.

2.2.1 Proxel-based Behavior Reconstruction

There are three main behavior reconstruction tasks defined for HMM and other related paradigms. *Evaluation* determines the probability of a given model to have produced a given output sequence. It is most often used to determine which of several models most likely produced a given sequence, and therefor closely resembles the system having produced the observations. *Decoding* determines the most likely sequence

of system states to have produced a given output sequence, thereby helping to understand the unobserved dynamics of the system. *Training* tries to determine a likely model given an observed output sequence, but as opposed to evaluation by manipulating the model parameters, e.g. through iteratively refining an initial model.

The two behavior reconstruction tasks *Evaluation* and *Decoding* for HnMRM can be performed using the so-called Proxel method. For a detailed introduction to Proxels see [12, 13]. There exists currently no general solution to the third task of Training HnMM. However, it can be solved for subclasses and some special cases [14]. Proxels are a state space-based analysis method to iteratively discover all possible system developments in discrete time steps and quantify them with their respective probabilities. Proxels can be used to mimic the HMM reconstruction algorithms by iteratively generating likely paths that could have produced the given output sequence. Proxels are ideal for path reconstruction and can be applied to various types of HnMM, including HnMRM [4, 7, 14]. In this paper we will adapt the HnMRM behavior reconstruction algorithm to hybrid models.

3 Hybrid Hidden non-Markovian Models

In this section we will formally define hybrid hidden non-Markovian models (HHnMM) and briefly describe the corresponding behavior reconstruction algorithm. As basis we will take the Hidden non-Markovian reward models (HnMRM) as described in [4]. These use rate rewards associated to the discrete states and impulse rewards associated to the transitions to model the influence of the discrete on the continuous model part. The influence of the continuous on the discrete model part will be modeled through guard functions. Guard functions (GF) are used in some SPN paradigms to express complex marking-dependent enabling rules for transitions. A transition with a guard function is disabled, if the GF evaluates to false. Thus by including terms depending on the continuous values in the GF, we can make the discrete part of the model depend on the continuous quantities. One can for example disable a transition if a particular continuous quantity leaves a given range,

or enable an immediate transition when a threshold value is crossed, and thus invoke an instantaneous state change.

3.1 Formal Definition

The formal definition of HHnMM uses the state space of a discrete stochastic model and the values of the reward measures as hidden model part. An HHnMM is formally defined as a 6-tuple $HHnMM = (S, TR, A, \Pi, \vec{p}, rr)$ with trace O and path Q .

- S is the set of N discrete states of the system.
- TR is the set of state transitions, where each $TR_i = (dist, gf, ir, aging)$ can represent multiple state changes; $dist$ describes the continuous probability distribution function that governs a state transition; guard function gf is a boolean expression defining a complex enabling rule based on the current system state, default is *true* if not specified; $ir(\vec{p}, s, s')$ describes the changes to the vector \vec{p} due to impulse rewards when the transition fires, changing the system state from s to s' ; $aging$ is set to true, if the transition has a race age memory policy (i.e. it retains the time elapsed since activation, when it is disabled before firing, and resumes the countdown when re-enabled, without re-sampling the distribution).
- $A = \{a_{ij}\}_{N \times N}$ describes the complete state transition behavior and a_{ij} is the set of transitions that can cause the state transition from S_i to S_j .
- $\Pi = (\pi_1, \dots, \pi_N)$ is the vector containing the initial probabilities of all discrete states and π_i is the probability of the system to be in state S_i at time 0.
- $\vec{p} = (\rho_1, \dots, \rho_M)$ is the vector containing the initial values of all continuous quantities.
- The rate rewards are defined by the function $rr(\vec{p}, s)$. It describes the changes to the vector \vec{p} due to rate rewards in the form of a differential equation $\frac{d\vec{p}}{dt}$ depending on the state s .
- O is a sequence of measurements o_t of one (or more) of the continuous quantities in \vec{p} with time

stamps e_t and often referred to as a trace. without loss of generality we assume that ρ_1 is the sampled value.

- Q is a sequence of system states s_t with time stamps e_t and often referred to as a path.

The model $\lambda = \{A, \Pi, \vec{\rho}, rr\}$ completely determines the dynamic behavior of the system and is therefore often used as its representative.

We now have to adapt the behavior reconstruction algorithm introduced in [4] to include complex enabling rules in the form of guard functions.

3.2 Behavior Reconstruction of HHnMM

In this section we will describe the HHnMM behavior reconstruction algorithm adapted from ([4]). The general idea of the Proxel-based behavior reconstruction algorithm is to follow all possible system development paths in discrete time steps; on-the-fly determining possible discrete system states, resulting continuous quantities and the probability of each path. When a protocol entry o_t occurs within a time step, the respective continuous value ρ_1 of each path is compared to o_t . The path is kept, if the difference is smaller than a given threshold ε . Otherwise, the path will be deleted, and not considered further.

The newly included guard functions influence the enabling of transitions and therefore need to be checked, when determining whether a given transition is currently enabled. The following verbal description of the HHnMM behavior reconstruction algorithm highlights where that is necessary.

1. The initial Proxel representing the initial state of the model is created in t_0 .
2. For each time step t_k ($k = 1..k_{max}$)- Loop until there are no more Proxels in time step t_{k-1} .
3. Get the next Proxel P from time step t_{k-1} .
4. If there is a protocol entry with a time stamp between t_{k-1} and t_k , check whether the path is valid (Proxel reward value ρ_1 is close to protocol reward value o_t)
5. If there are immediate transitions enabled (**their guard function evaluate to true**), the Proxel represents a vanishing state,

add Proxels representing these immediate state changes to t_{k-1} , modifying the reward values according to the transitions impulse rewards.

If the path is still invalid, even after the impulse rewards, delete the Proxel.
6. If the state is not vanishing, modify the Proxels reward values according to the current rate reward, check path validity again
7. For all timed transitions that are enabled in the Proxels current state (**whose guard functions evaluate to true**)

add Proxels representing these state changes to t_k , modifying the reward values according to the transitions impulse rewards.

If the path is still invalid, even after the impulse rewards, delete the Proxel.
8. If there is a non-zero probability of no state change happening, and the current path is valid, add a Proxel, representing the case, that the system remains in the discrete state to t_k .

Determining the validity of a path needs to be done, when the time stamp of a protocol entry lies within the currently considered time step. The distance of the protocol value and the path reward value is tested at the beginning and at the end of the time step, i.e. before and after the rate rewards have been added; before and after a state change happens, i.e. before and after considering the impulse rewards. If the Proxels reward value is within ε of the protocol reward value at any of these instances, the path is deemed valid.

At the end of the reconstruction (after processing the complete protocol) The result of the algorithm is a list of system development paths, that could have produced the protocol with non-zero probability. These paths then need to be analyzed further to answer the questions posed for the given system. For the solar thermal system, the amount of external heating energy needed could be determined, giving an indication on whether the equipment performs to the given standards. One could also search for the most likely cause

of an unusually high need for external energy, be it an unusually large or an untimely usage profile. For the diabetic, one could determine how well the patient adheres to his dietary and medication plan, answering the following questions: Did he forget an insulin injection? How often did he eat a snack? The post-processing analysis can be done on the most likely path(s) or on the path(s) with the smallest divergence from the protocol. The following section shows some results for the two example systems.

4 Experiments

In this section we will demonstrate that Virtual Stochastic Sensors can reconstruct unobserved behavior from given protocols of hybrid systems. In order to test the validity of the reconstructed system behavior, we used simulation models to generate output protocols, also recording the actual model behavior. This will enable a direct comparison between the actual system behavior and the reconstructed path.

4.1 Demonstrating Path Reconstruction - Diabetes Example

The first experiment is conducted with the same application example chosen in [4]. The model describes the dynamics of the glucose insulin metabolism of a type-one diabetic, who can not produce his own insulin and has to inject insulin regularly. The interaction dynamics of the insulin level (IL) and glucose level (GL) have been taken from [15]. A graphical representation of the HHnMM can be seen in Figure 1 in the form of an SPN with impulse rewards. The rate rewards are depicted in the form of differential equations, since these are the same for all discrete states. We assume the patient to have two distinct states *hungry* and *full*, where after eating, he remains in state *full* for a time period that is determined by a normal distribution with mean $\mu = 7.5h$ and a standard deviation of $\sigma = 1h$. In state *hungry* he will again remain for a normally distributed time period with parameters $\mu = 0.5h$ and $\sigma = 0.1h$. This combination leads to eating intervals with an average length of eight hours. When the patient eats, his blood glucose level is increased by a fixed amount ($1000units$), which is an idealization. While eating, he should inject an also

fixed amount of insulin ($3000units$).

The patient sometimes forgets to inject his insulin, thus not increasing his blood insulin level. Furthermore, when the blood glucose level drops below a certain threshold ($5000units$), the patient eats a snack, increasing his glucose level by a smaller amount ($200units$). This dependence of the discrete model on the continuous measures is expressed by guard functions, making the model truly hybrid.

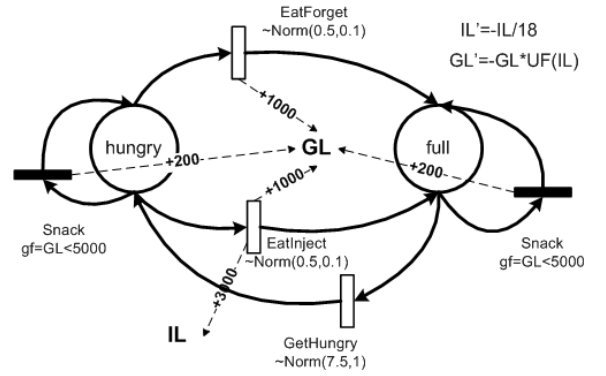


Figure 1: A graphical representation of the insulin glucose metabolism model of a diabetic in the form of a state space with associated rewards

We now assume, that the patient measures his blood glucose level at regular intervals and records the results in a protocol. From such a protocol we want to deduce, how often he ate snacks, and how often he forgot to inject his insulin dose. This could help a doctor to determine how well the patient adheres to his dietary and medication plan. Figure 2 shows a measurement protocol (trace), which was produced by a simulation of the patient dynamics.

The reconstruction algorithm was applied to the trace with 32 entries, which represented a time span of $100h$. Figure 3 (left) shows the simulated system path that produced the trace in Figure 2, which was recorded during the simulation.

The reconstruction algorithm resulted in a list of 45050 possible paths that could have produced the given trace ($\epsilon = 60$). The computation time of the reconstruction was 11 seconds. Figure 3 (right) shows the most likely of these reconstructed paths. A comparison with the original system path shows that the HHnMM reconstruction algorithm was able to almost exactly reproduce the simulated behavior that led to

time stamp	GL
2.36	5581
5.82	5109
9.27	6131
12.11	5827
14.82	5577
18.00	6191
20.83	5858
23.86	5559
26.30	6340
29.41	5923
32.43	5599
35.44	6271
38.30	5879
41.60	5518

Figure 2: Protocol of regular glucose level measurements with time stamps

time stamp	transition
6.79	Snack
8.69	Snack
8.73	EatForget
15.94	EatInject
26.14	EatInject
34.70	EatInject
42.65	EatInject
48.58	EatInject
58.82	EatInject
65.29	EatInject
72.95	EatInject
80.00	EatInject
86.42	EatInject
92.56	EatInject

time stamp	transition
7	Snack
9	Snack
10	EatForget
16	EatInject
25	EatInject
34	EatInject
42	EatInject
50	EatInject
58	EatInject
66	EatInject
73	EatInject
79	EatInject
87	EatInject
94	EatInject

Figure 3: Eating and injection times in original system path (left) and reconstructed system path (right)

the observed trace. In particular, it correctly reproduced the two snacks and the one forgotten injection. The differences in the time stamps are mostly within the discretization accuracy of $1.0h$.

We can conclude, that the HHnMM behavior reconstruction algorithm was successfully applied to a modified version of the glucose insulin metabolism model, correctly reproducing the patients deviations from his dietary and medication plan. On a more abstract level, we were able to reconstruct behavior of a hybrid system with complex dynamics within the continuous part, described by differential equations. The state space of the model is small, and the trace analyzed short, thus the runtime is not an issue here. However, memory and computation time are strongly influenced by model parameters, as shown in [4].

4.2 Demonstrating Path Reconstruction - Solar Thermal Heating

The second experiment was conducted with an application taken from the field of renewable energy. We have built a simple idealized model of a small home solar thermal system, using simple dynamics and constant rewards, to demonstrate the algorithm's ability to handle a larger state space and a differentiated rate reward structure. A graphical representation of the model is shown in Figure 4, again using SPN notation for the discrete model part. Here we can easily distinguish three independent discrete system parts, that

only interact through the continuous measure.

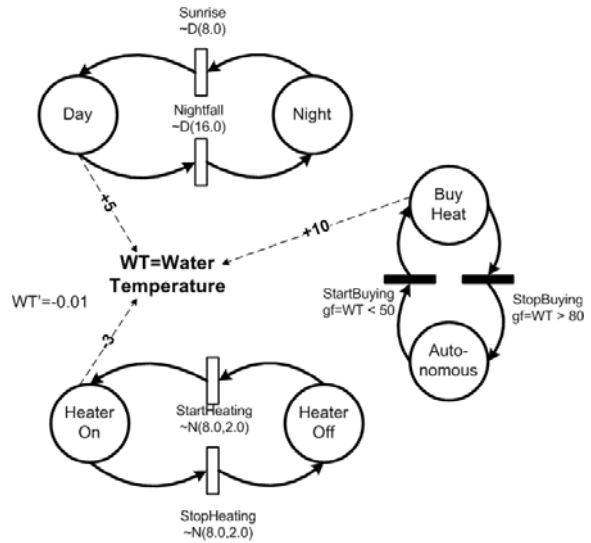


Figure 4: A graphical representation of the solar heater dynamics

The central element of the small solar heating system is the water tank with its water temperature WT as central attribute. The water temperature can be increased by the solar thermal collectors on the roof, or by an external heat source that has to be paid for. The temperature decreases, when the home heating system is turned on. The water temperature is bounded between $20^{\circ}C$ and $90^{\circ}C$. The first discrete system part is the rising and setting of the sun, which is abstracted

to a fixed $16h$ period of daytime and an fixed $8h$ period of night. When the sun is up, the water temperature increases linearly by $5^\circ C/h$. The second discrete part is the random usage of the home heating system. The heater switches between *On* and *Off* in normally distributed intervals with parameters $\mu = 8h$ and $\sigma = 2h$. When the heater is turned on, the water temperature decreases linearly by $3^\circ C/h$. The third element is the addition of external heating, which is switched on, when the water temperature falls below $50^\circ C$ and switched off, when it reaches $80^\circ C$. When the external heating is turned on, the water temperature increases linearly by $10^\circ C/h$.

We assume that the temperature of the water tank is recorded in a protocol of regular measurements. An example of such a trace is shown in Figure 5.

time stamp	WT
0	50
5.03	60.02
10.01	62.76
15.05	59.94
19.84	60.4
24.86	84.26
29.95	90
35.04	83.96
39.96	69.15
44.77	57.47
49.68	65.8

Figure 5: Protocol of regular water temperature measurements with time stamps

The reconstruction algorithm was applied to a trace with 132 entries, which represented a time span of $1000h$. In Figure 6 (left) shows the simulated system path that produced the trace in Figure 5, reduced to the periods of external heating needed.

time stamp	transition	time stamp	transition
45.27	StartBuying	46	StartBuying
48.69	StopBuying	50	StopBuying
358.91	StartBuying	359	StartBuying
361.62	StopBuying	362	StopBuying

Figure 6: External heating periods in original system path (left) and reconstructed system path (right)

The reconstruction algorithm resulted in a list of 115 possible paths that could have produced the given trace ($\epsilon = 2.0$). The computation time needed for

the reconstruction was 3 minutes. Figure 6 (right) shows the most likely of these reconstructed paths. The HHnMM reconstruction algorithm correctly reproduced the two periods of external heating needed. The differences in the time stamps are mostly within the discretization accuracy of $1.0h$. All but one of the random heating periods were also reconstructed correctly, but are not depicted here.

In conclusion, we were again successful in reconstructing the system behavior that lead to a given protocol. The solar heater system differs from the diabetes example in that the rate rewards are all linear, but the discrete model state space is larger, containing not two but eight discrete states, with two concurrent non-Markovian transitions being enabled simultaneously.

5 Conclusion and Outlook

The paper introduced a method to reconstruct unobserved behavior of hybrid systems from measurement protocols of the continuous variables. These virtual stochastic sensors (VSS) use hybrid hidden non-Markovian models (HHnMM) to represent the hybrid systems. HHnMM contain a complex reward structure to model influences of the discrete on the continuous part and guard functions that can disable transitions based on reward values to model the influence of the continuous on the discrete part. A medical example illustrates the potential complexity of the reward measures, and an example from the renewable energy sector shows a collection of discrete systems that interact through a continuous system. This extension of the previous hidden non-Markovian reward models (HnMRM) to truly hybrid systems extends the possible application fields of VSS for behavior reconstruction.

The current paper only demonstrates the ability of VSS to reconstruct a systems behavior solely based on infrequent measurements of continuous measures. The runtime and memory complexity of the analysis are not yet satisfactory, since they are highly dependent not only on the model but also on method parameters. Future research will focus on guidelines and automatic processes to help optimize the algorithm parameters for a specific application. Starting from the academic example used here we want to research actual applications in the renewable energy field, since

this is a highly dynamic area and most of the systems encountered there are inherently hybrid.

References

- [1] K. S. Trivedi and V. G. Kulkarni. *FSPNs: Fluid stochastic Petri nets*. In: Application and Theory of Petri Nets 1993, Publisher: M. Ajmone Marsan, Lecture Notes in Computer Science, Springer Berlin Heidelberg, p. 24-31, 1993.
- [2] J. Muppala, G. Ciardo and K. S. Trivedi. Stochastic Reward Nets for Reliability Prediction. *Communications in Reliability, Maintainability and Serviceability*, vol. 1, no. 2, p. 9-20, 1994.
- [3] C. Krull, G. Horton, B. Denkena and B. Dengler. *Virtual Stochastic Sensors for Reconstructing Job Shop Production Workflows*. Proceedings of the 8th EUROSIM Congress on Modelling and Simulation, in Cardiff, UK, September 2013.
- [4] C. Krull and G. Horton. *Hidden non-Markovian Reward Models: Virtual Stochastic Sensors for Hybrid Systems*. Proceedings of the Winter Simulation Conference, in Berlin, Germany, p. 224:1–224:12, 2012.
- [5] F. Bause and P. S. Kritzinger *Stochastic Petri Nets*. Vieweg, Germany, 2002.
- [6] B. Tuffin, D. S. Chen and K. S. Trivedi. Comparison of Hybrid Systems and Fluid Stochastic Petri Nets. *Discrete Event Dynamic Systems*, vol. 11, no. 1-2, p. 77-95, 2001.
- [7] C. Krull, R. Buchholz and G. Horton. *Virtual Stochastic Sensors: How to gain Insight into Partially Observable Discrete Stochastic Systems*. Proceedings of the 30th IASTED International Conference on Modelling, Identification and Control, Innsbruck. Austria, February 2011.
- [8] P. H. Ibargüengoytia and A. Reyes *Constructing Virtual Sensors Using Probabilistic Reasoning*. In: MICAI 2006: Advances in Artificial Intelligence, Publisher: A. Gelbukh and C. A. Reyes-Garcia, Lecture Notes in Computer Science, Springer Berlin Heidelberg, p. 218-226, 2006.
- [9] C. Krull and G. Horton. *HIDDEN NON-MARKOVIAN MODELS: FORMALIZATION AND SOLUTION APPROACHES*. Proceedings of 6th Vienna Conference on Mathematical Modelling, Vienna, Austria, p. 682-693, February 2009.
- [10] G. A. Fink *Markov Models for Pattern Recognition*. Springer, Berlin, Heidelberg, 2008.
- [11] G. Bolch, S. Greiner, H. de Meer and K. S. Trivedi *Queuing Networks and Markov Chains*. John Wiley & Sons, New York, 1998.
- [12] S. Lazarova-Molnar. *The Proxel-Based Method: Formalisation, Analysis and Applications*. Dissertation, Otto-von-Guericke-University Magdeburg, Germany, November 2005.
- [13] G. Horton. *A New Paradigm for the Numerical Simulation of Stochastic Petri Nets with General Firing Times*. Proceedings of the European Simulation Symposium 2002, in Dresden, Germany, p. 129-136, October 2002.
- [14] R. Buchholz. *Conversive Hidden Non-Markovian Models*. Dissertation, Otto-von-Guericke-University Magdeburg, Germany, Mai 2012.
- [15] J. Hartlove, D. Shaffer and S. Ragan. *Glucose-Insulin Model. A core learning goals activity for science and mathematics*, The Maryland Virtual High School. 2001.

Approaches for State Event Handling by Simulation Algorithm and via Model Description

Andreas Körner¹, Felix Breiteneker¹

¹Vienna University of Technology,
Institute for Analysis and Scientific Computing,
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria
andreas.koerner@tuwien.ac.at

This Paper deals with state events of hybrid systems and their handling in two different ways. One point of view is the algorithmic point which means to implement the handling of state events of a certain model via the simulation algorithm. The other way would be to adapt the model in this way that the handling of the event can be done more easily. These considerations are related to the relationship between the abstract mathematical model and the implementation in a simulation environment, the simulation model. The influence of the aspects of the implementation in the simulation environment to derive the mathematical description has to be considered also regarding the handling of state events. In the following paper some theoretical considerations are presented and an electrical circuit is used for illustration of the concepts.

1 Introduction, Notation and Basic Facts

State event handling describes the process of finding, administrating and executing a state transition in a hybrid model structure. In the following subsection a short definition of hybrid model structures and an introduction in these three processes related to state event handling will be given.

1.1 Mathematical Definition of Hybrid Dynamical Model Structures

In this subsection a short definition of a hybrid model will be given to define notation and connection to the following sections, more detailed in [1].

The mathematical definition of a hybrid model is given by a hybrid automaton. A *hybrid automaton* is defined by a tuple $(L, X, A, W, E, \text{Inv}, \text{Act})$ meanwhile each of the elements of the tuple is defined as follows:

- L is a finite set, called the set of *discrete states*.

- X is the continuous state space of the hybrid automaton in which the *continuous state variables* x take their values. In applications X fulfills typically $X \subset \mathbb{R}^n$, but also n -dimensional manifolds are possible.
- A is a finite set of symbols which labels the *nodes* of the automaton.
- $W = \mathbb{R}^q$ is the continuous communication space in which the continuous *external variables* w take their values.

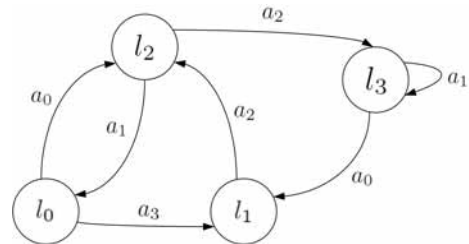


Figure 1: Basic model structure defined by a finite automaton, here illustrated with 4 nodes

- E is a finite set of edges, called *transitions* or *events*. Every edge is defined by a tuple $(l, a, \text{Guard}_{l,l'}, \text{Jump}_{l,l'})$, where $l, l' \in L, a \in A$. $\text{Guard}_{l,l'}$ is a subset of X and $\text{Jump}_{l,l'}$ is a relation defined by a subset of $X \times X$. The transition from the discrete state l to l' is called *enabled*, when the continuous state fulfills $x \in \text{Guard}_{l,l'}$. The continuous state x jumps to a value \tilde{x} if $(x, \tilde{x}) \in \text{Jump}_{l,l'}$.
- $\text{Inv}: L \rightarrow X$ is a mapping from the locations L to the set of subsets of X so that $\text{Inv}(l) \subset X, \forall l \in L$. Whenever the system is at location l , the continuous state x must satisfy $x \in \text{Inv}(l)$. $\text{Inv}(l)$ for $l \in L$ is called the *location invariant* of location l .
- Act is a mapping that assigns to each location $l \in L$ a set of differential–algebraic equations F_l , relating the continuous state variables x with their time–derivatives \dot{x} and the continuous external variables w by

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

The solution of these differential–algebraic equations are called the *activities* of the location l .

Thus the definition of a hybrid model is based on the mathematical concept of a graph this abstract definition of the model can be illustrated by using the graphical representation of a graph. In the case of hybrid models the discrete model structure is also defined by a finite automaton which can be illustrated like a graph, see Figure 1.

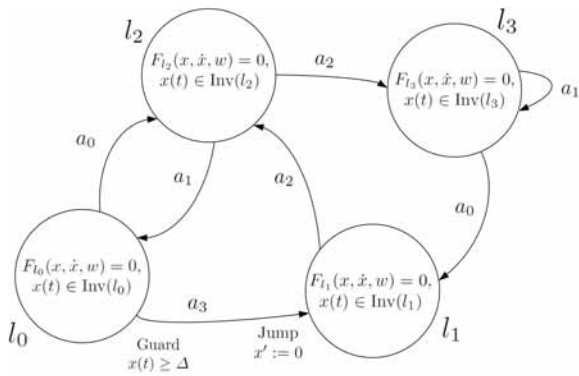


Figure 2: Hybrid model structure represented by a hybrid automaton, here illustrated with 4 nodes

Due to the definition of hybrid models the discrete states of this finite automaton are related to a continuous dynamical system and the transitions of the discrete system has to be connected to the continuous systems. This leads to the illustration of the hybrid automaton depicted in Figure 2.

1.2 System Simulation Formulation

The mathematical definition of events is done by using the theory of graphs. For system simulation a reformulation has to be done to achieve two mathematical parts, the differential equation and the algebraic equation.

Instead of a function F given in equation (1) two functions $f: \mathbb{R}^n \times \mathbb{R}^q \rightarrow \mathbb{R}^n$ and $g: \mathbb{R}^n \times \mathbb{R}^q \rightarrow \mathbb{R}^k$ are defined, as well as an initial value $x_0 \in \mathbb{R}^n$ is defined, to formulate the differential algebraic equation as

$$\begin{aligned} \dot{x} &= f(x, w), x(t_0) = x_0, \\ g(x, w) &= 0. \end{aligned} \quad (2)$$

The reformulation of the DAE system leads to some restrictions in the generality of the description. However the formulation (2) is used as the mathematical description in the simulation environment, see Fig. 3.

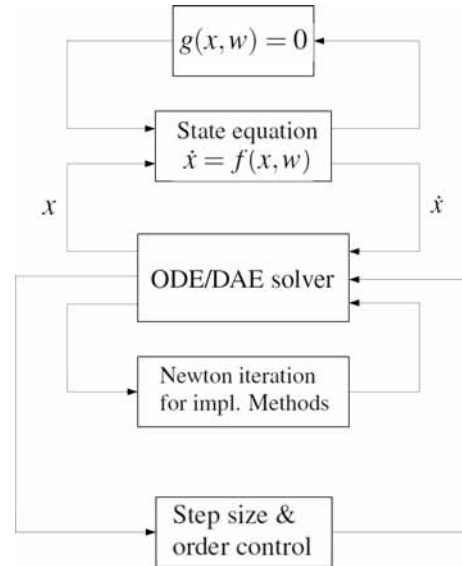


Figure 3: Structure for numerical simulation of DAE systems

One proper method to solve the problem of the event iteration is the Newton iteration. For the Newton iteration the function h has to be assumed as differentiable. If this is not fulfilled there are several other methods which can be used like regula falsi, bisection method and the secant method. With the given assumption the Newton iteration is without any doubt the most efficient one to iterate $h(x(t), w(t)) = 0$ resulting t^* . The Newton iteration for the event function h is given by

$$t_{k+1} = t_k - \frac{h(x(t_k), w(t_k))}{\dot{h}(x(t_k), w(t_k))}, \quad (6)$$

where \dot{h} denotes the derivative of the event function with respect to time t .

Simplifications are possible in case of using the newton method if the event function has a special structure. For the first consideration one of the simplest cases of this simplification will be introduced after this introduction the idea will be discussed generalised. Let's assume the event function $h: \mathbb{R}^n \times \mathbb{R}^q \rightarrow \mathbb{R}$ is defined by

$$h(x(t), w(t)) = x_l(t) - \hat{X}, \quad (7)$$

for $\hat{X} \in \mathbb{R}^+$. In this case the event function depends only on the l^{th} state variable. \hat{X} can be interpreted as an threshold. Here thresholds are used in a technical context for defining events. An explicit example will be given in section 3.

To apply the Newton method for finding zeros of the event function h given in (7) the derivative is needed. In this case the derivative is very simple because it is the univariate derivative of x_l with respect to t ,

$$\dot{h}(x_l) = \dot{x}_l. \quad (8)$$

In this simple equation a simplification is hidden. The derivative of the state variable is available, because it is given by $\dot{x} = f(x, w)$. More detailed the state equation is given by

$$\frac{d}{dt} \begin{pmatrix} x_1(t) \\ \vdots \\ x_l(t) \\ \vdots \\ x_n(t) \end{pmatrix} = \begin{pmatrix} f_1(x(t), w(t)) \\ \vdots \\ f_l(x(t), w(t)) \\ \vdots \\ f_n(x(t), w(t)) \end{pmatrix}.$$

This means that the value don't have to be derived separately and the derivative of h is available given by

$$\dot{h}(x_l) = \dot{x}_l = f_l(x(t), w(t)). \quad (9)$$

Of course the procedure is the same if the event function is defined as $h(x(t), w(t)) = ax_l(t) - b$ for given $a, b \in \mathbb{R} \setminus \{0\}$.

For more generalized functions the procedure is not that easy. Imagine an event function h defined by more than only one state variable,

$$h(x, w) = h(x_1(t), \dots, x_l(t), \dots, x_n(t)). \quad (10)$$

In this case a similar approach like introduced before is not possible. For a scalar function in multiple variables the Newton iteration is not applicable. Also finding a analytical solution for zeros of the function is not possible. Here are some numerical approaches required.

One possibility to find analytical approaches can be to find a way to distribute the event function in a vector valued function, where the number of multiple variables and the number of vector components are equal. In this case the event function $h: \mathbb{R}^n \times \mathbb{R}^q \rightarrow \mathbb{R}^n$,

$$h(x_1(t), \dots, x_n(t)) = \begin{pmatrix} h_1(x_1(t), \dots, x_n(t)) \\ \vdots \\ h_l(x_1(t), x_n(t)) \\ \vdots \\ h_n(x_1(t), x_n(t)) \end{pmatrix} \quad (11)$$

opens the possibility to apply the multivariate Newton iteration. But this formulation means to differ from the definition of a hybrid dynamical system - especially from the definition of the state transitions or events - and is possible in very specialised, maybe pathological examples.

3 Case Study: Oscillating Circuit with Diode

One suitable example to investigate an adaption of model structure is a oscillation circuit with a diode,

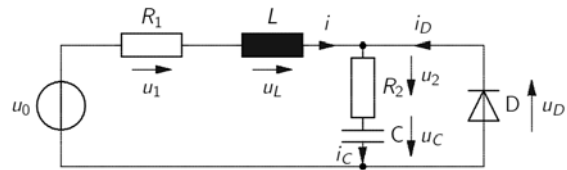


Figure 5: Oscillating circuit with diode

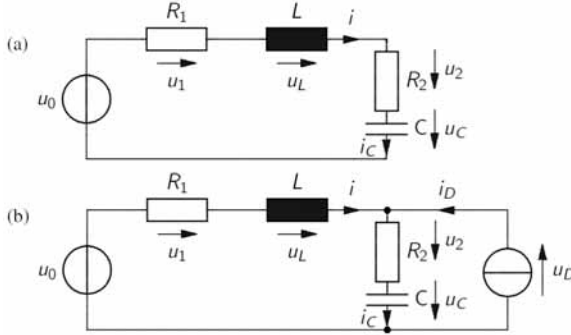


Figure 6: Two different areas of operation of the oscillating circuit with diode

illustrated in Figure 5. The diode provides the algebraic equation whereby the oscillating circuit supplies a differential equation.

The first definition is the model of the diode which has to be used to formulate the state equation and this model is also leading in the computation of the algebraic equation. In this case study the Shockley–Equation is used as a diode model and it is given by

$$i_D(u_D) = \begin{cases} I_S \left(e^{\frac{u_D}{U_T}} - 1 \right) & \text{for } u_D > 0, \\ 0 & \text{for } u_D \leq 0, \end{cases} \quad (12)$$

with $I_S, U_T > 0$. The first notice is, that this model defines the transition between the two states of the electric circuit implicit. The given definition of the model consequents u_D to be the variable which defines the transition from one state to the other. But this consideration is wrong, because the two states don't offer always the variable u_D because the diode switches on and off a part of the electric circuit, which is illustrated in Figure 6. Case (b) is represented by a voltage controlled current source, because this is the closest element which represents the diode equation.

The state vector in this example is given by $x = (u_C, i)$, the vector w by $w = (u_D, i_D, u_0)$. Kirchhoff's law applied on the right loop in Figure 5 leads to the equation

$$u_2 + u_C + u_D = 0. \quad (13)$$

This equation allows to express the algebraic variable u_D via the state variable u_C and u_2 . In the different states u_2 depends only once on the second state variable i otherwise on the sum of i and i_D . Because of

equation (13) the definition of the event function is given by

$$h(x, w) = -u_C - u_2. \quad (14)$$

The model equations for the two different circuits, shown in Figure 6 are given by the following two cases.

For $u_D \leq 0$; (a) in Figure 6:

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} u_C(t) \\ i(t) \end{pmatrix} &= \begin{pmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{R_1+R_2}{L} \end{pmatrix} \begin{pmatrix} u_C(t) \\ i(t) \end{pmatrix} + \\ &+ \begin{pmatrix} 0 \\ \frac{1}{L} \end{pmatrix} u_0(t) \end{aligned} \quad (15)$$

For $u_D > 0$; (b) in Figure 6:

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} u_C(t) \\ i(t) \end{pmatrix} &= \begin{pmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{R_1+R_2}{L} \end{pmatrix} \begin{pmatrix} u_C(t) \\ i(t) \end{pmatrix} + \\ &+ \begin{pmatrix} 0 \\ \frac{1}{L} \end{pmatrix} u_0(t) + \begin{pmatrix} \frac{1}{C} \\ -\frac{R_2}{L} \end{pmatrix} I_S \left(e^{\frac{u_D(t)}{U_T}} - 1 \right) \\ R_2 \left(i(t) + I_S \left(e^{\frac{u_D(t)}{U_T}} - 1 \right) \right) + u_D(t) &= 0 \end{aligned} \quad (16)$$

For simplicity the distinction of cases are separated by the sign of u_D which is not fully correct, because in case (a) u_D is not available and has to be represented by the event function (14). Due to the fact that the differential–algebraic equation in (b) is a DAE of index 1 the case can be reformulated in an ODE. For this the first derivative of the algebraic equation has to be calculated to

$$R_2 \frac{d}{dt} i(t) + \left(\frac{R_2 I_S}{U_T} e^{\frac{u_D(t)}{U_T}} + 1 \right) \frac{d}{dt} u_D(t). \quad (17)$$

Now the derivative of $i(t)$ can be inserted from the state equation and a ODE formulation of the discussed state is obtained. The new state space is given by the vector

$$x = (i(t), u_C(t), u_D(t)) \quad (18)$$

and the state transition results in a state space change. For this model formulation of the discussed state the event function

$$h(x) = u_D(t) \quad (19)$$

is appropriate and leads to the simplifications discussed in section 2.

4 Summary and Outlook

This paper shows that the modelling part of hybrid dynamical systems offers a wide range to create proper mathematical model descriptions. The case study shows that the mathematical formulation of a hybrid dynamical system is not unique and in some special situation it can lead to simplifications as presented.

The simplifications which are presented are related to the type of transitions and also to the type of changes after the event. Further work will be an detailed characterisation on the state events regarding type and activator of the change in the dynamical system, as addressed in [1].

References

- [1] A. Körner, F. Breitenecker. *Benchmarks for Hybrid Modelling*. Proceedings of the 2nd Annual International Conference on Business, Technology and Innovation, Chapter: Computer Science Information Systems and Mechatronics Robotics and System Engineering, in Durres, Albania, p. 224–233, 2013.
- [2] A. Körner, F. Breitenecker, N. Popper. *About an Alternative Method of Numerical Iteration for State Event Finding and Handling in System Simulation of Hybrid Dynamical Systems*, UK-SIM – AMSS 15th International Conference on Modelling and Simulation, in Cambridge, UK, p. 390 – 395, 2013.

Ereignislokalisierung durch die Henon-Methode

Franz Preyser¹, Bernhard Heinzl², Matthias Rößler³, Felix Breitenecker¹

¹Institut für Analysis und Scientific Computing, TU Wien

²Institut für Rechnergestützte Automation, TU Wien

³dwh GmbH, Simulation Services, Wien

Franz.Preyser@tuwien.ac.at

Ereignislokalisierung spielt im Bereich der Simulation mathematischer Modelle in Form von gewöhnlichen Differentialgleichungssystemen eine wichtige Rolle. Bei den dabei üblicherweise verwendeten Verfahren (z.B. Regula-Falsi), handelt es sich ausschließlich um iterative Verfahren. Im Folgenden soll ein nicht iteratives Verfahren namens “Henon” vorgestellt werden, welches in Matlab implementiert wurde. Nachdem kurz auf die Grenzen der Henon-Methode eingegangen wird, werden abschließend Resultate eines Laufzeitvergleichs mit dem Regula-Falsi-Verfahren präsentiert.

1 Einleitung

Beim üblichen numerischen Lösen von gewöhnlichen Differentialgleichungssystemen liefert das numerische Lösungsverfahren (ODE-Solver) Lösungswerte zu konkreten Werten der unabhängigen Variable (meistens die Zeit). Diese Werte haben entweder einen festen Abstand zueinander (fix-step-solver), oder die Abstände sind variabel (variable-step-solver). In manchen Fällen ist es jedoch wünschenswert, Lösungswerte bei ganz bestimmten Systemzuständen zu erhalten, da z.B. ab Erreichen dieses Zustandes, das verwendete Modell nicht mehr gültig ist und daher etwaige Modellparameter oder ganze Gleichungen geändert werden müssen.

1.1 Ein Beispiel

Ein Beispiel liefert eine sehr einfache Version des wohl bekannten „Bouncing Ball“. Hierbei wird die Position eines hüpfenden Balles berechnet, indem während der Flugphasen einfach die Bewegungsgleichung (freier Fall)

$$x(t) = x_0 + v_0 \cdot t - \frac{g}{2} \cdot t^2 \quad (1)$$

gelöst wird. Diese Gleichung gilt aber nur, solange der

Ball nicht am Boden aufschlägt. D.h. bei Erreichen des Systemzustandes $x(t) = 0$ muss das numerische Lösungsverfahren abgebrochen werden um entweder das Modell zu ändern, oder, wie hier möglich, den Systemzustand (Reflexion des Balles: $v(t) \mapsto -v(t)$). Es muss also das numerische Lösungsverfahren dazu gebracht werden, einen Lösungswert genau beim Aufprallzeitpunkt zu berechnen und danach abubrechen. In weiterer Folge soll die unabhängige Variable stets als Zeit t bezeichnet werden. Desweiteren verwenden wir fett gedruckte Variablen für vektorwertige Größen (z.B. Systemzustand \mathbf{x}).

1.2 Übliche Verfahren

Die übliche Herangehensweise ist nun, eine sogenannte Ereignisfunktion

$$S : \Omega \rightarrow \mathbb{R} \quad (2)$$

zu definieren. Diese Funktion bildet den Zustandsraum Ω stetig in die reellen Zahlen ab und ihre Nulldurchgänge kennzeichnen genau die Ereigniszeitpunkte. Im oben genannten Beispiel wäre das gesuchte Ereignis, das Erreichen der Ballhöhe 0 und die Ereignisfunktion deshalb:

$$S(x) = x \quad (3)$$

Das numerische Lösungsverfahren überprüft nach jeder Lösungswertberechnung, ob sich das Vorzeichen der Ereignisfunktion S geändert hat. Wird ein Vorzeichenwechsel detektiert, so wird ein Algorithmus zur Nullstellenlokalisierung angeworfen. Übliche Verfahren sind z.B. das Newton-, das Sekanten-, das Bisektions- oder das Regula-Falsi- Verfahren. Dabei handelt es sich ausnahmslos um iterative Verfahren. Das im Folgenden vorgestellte Henon-Verfahren, benannt nach dem Autor des Papers [1], aus dem die Idee zum Algorithmus stammt, soll hier einen anderen Weg gehen.

1.3 Henon

Das Henon-Verfahren kann als eine Erweiterung des Newton-Verfahrens gesehen werden und stellt deshalb auch dieselben Ansprüche an die Funktion, deren Nullstelle lokalisiert werden soll. In unserem Fall die Ereignisfunktion. Vernachlässigt man den numerischen Fehler und geht also davon aus, dass die für den Systemzustand \mathbf{x} berechneten Lösungswerte \mathbf{x}_n gleich den Werten der analytischen Lösung $\mathbf{x}(t_n)$ sind, so kann die Ereignisfunktion auch als Funktion der Zeit gesehen werden:

$$\tau(t) = S(\mathbf{x}(t)) \quad (4)$$

Ist nun diese Funktion τ in dem Intervall $(t_n, t_e + (t_e - t_n))$ stetig und monoton, wobei $\text{sign}(\tau(t_n)) \neq \text{sign}(\tau(t_{n+1}))$ und $\tau(t_e) = 0$ gelte ($t_e \dots$ Ereigniszeitpunkt) und ist zusätzlich die Ableitung $\frac{d\tau}{dt}$ in dem Intervall bekannt, so kann das Newton-Verfahren angewandt werden. Dieses macht nichts anderes, als die Nullstelle \tilde{t}_e der Tangente im Punkt $\tau(t_n)$ zu berechnen und als erste Näherungslösung für die Nullstelle t_e von τ heranzuziehen:

$$\tilde{t}_e = t_n - \left(\frac{d\tau}{dt}(t_n) \right)^{-1} \cdot \tau(t_n) \quad (5)$$

Anschließend wird der Systemzustand zum Zeitpunkt \tilde{t}_e berechnet, um danach erneut die Nullstelle der Tangente im Punkt $\tau(\tilde{t}_e)$ zu ermitteln. Dieser Vorgang wird solange iteriert, bis $\tau(\tilde{t}_e)$ nahe genug bei Null und somit \tilde{t}_e nahe genug bei t_e liegt. Gleichung (5) kann jedoch auch anders interpretiert werden: Da mit den obigen Voraussetzungen $\tau(t)$ im Intervall (t_n, t_e) invertierbar ist und die Inverse im Punkt $\tau(t_n)$ die Steigung $\left(\frac{d\tau}{dt}(t_n) \right)^{-1}$ hat, entspricht (5) genau einem

Schritt des expliziten Eulerverfahrens mit Schrittweite $-\tau(t_n)$, angewandt auf das AWP:

$$\begin{cases} \frac{dt}{d\tau}(\tau) &= \left(\frac{d\tau}{dt}(t) \right)^{-1} \\ t(\tau_n) &= t_n \end{cases} \quad (6)$$

Es liegt daher nahe, anstelle der iterativen Anwendung des Newton-/Expliziten Euler-Verfahrens, einmalig ein Verfahren höherer Ordnung anzuwenden, wie z.B. Runge-Kutta. Beim Henon-Verfahren passiert genau das. Eine wesentliche Voraussetzung für Newton wie auch für Henon ist also, im Unterschied zu allen anderen aufgezählten Methoden, dass die Ableitung der Ereignisfunktion analytisch bekannt und die Ereignisfunktion in einem kleinen Bereich um die zu lokalisierende Nullstelle invertierbar sein muss.

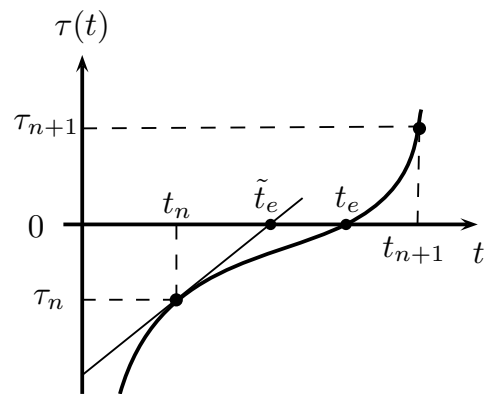


Abbildung 1: Eine Iteration im Newton-Verfahren.

2 Matlab - Implementierung

2.1 Algorithmusaufbau und Funktionsweise

Um die Brauchbarkeit der Henon-Methode zu überprüfen, haben wir sie in Matlab implementiert. Konkret wurde eine Wrapper-Funktion für Matlab ODE-Solver geschrieben, wobei auch der zu verwendende Solver (z.B. ode45) als Parameter übergeben werden kann. Diese Wrapper-Funktion kann somit völlig universell auf jedes Beispiel angewandt werden, bei dem üblicherweise ein Matlab-Solver zum Einsatz kommt und die Voraussetzungen für das Henon-Verfahren erfüllt sind. Es handelt sich dabei also um

Anfangswertprobleme(AWPs) der Bauart:

$$\begin{cases} \frac{d\mathbf{x}}{dt}(t) = \mathbf{f}(t, \mathbf{x}) \\ \mathbf{x}(t_0) = \mathbf{x}_0 \end{cases} \quad (7)$$

Zusätzlich zum AWP (7) muss noch eine Ereignisfunktion $S(\mathbf{x})$ angegeben werden. Sowohl zum Lösen von (7), als auch für das Henon-Verfahren wird der mittels Parameter festgelegte Solver verwendet. Es ist sogar möglich AWP (7) und AWP (6) so zusammenzufassen, dass lediglich der Wert einer Variablen K entscheidet, ob beim Lösen des resultierenden AWP gerade das eigentliche Problem (7) gelöst, oder eine Nullstelle der Ereignisfunktion lokalisiert wird.

$$\begin{cases} \frac{d\mathbf{y}}{d\tau}(\tau) = \mathbf{f}(t(\tau), \mathbf{y}) \cdot K(\tau, \mathbf{y}) \\ \frac{dt}{d\tau}(\tau) = K(\tau, \mathbf{y}) \\ \mathbf{y}(\tau_0) = \mathbf{y}_0 \\ t(\tau_0) = t_0 \end{cases} \quad (8)$$

(8) zeigt die Gestalt des oben erwähnten zusammengefassten AWP. Die unabhängige Variable τ nimmt nun je nach Wert von K die Rolle der Zeit t ($K \equiv 1$) oder die Rolle der Ereignisfunktion ($(S \circ \mathbf{x})(t)$ ($K = (\frac{d\tau}{dt})^{-1}$) an. \mathbf{y} entspricht in jedem Fall dem Systemzustand \mathbf{x} , nur einmal als Funktion der Zeit und einmal als Funktion des Ereignisfunktionswertes. Möchte man nun eine zwischen den Zeitpunkten τ_n und τ_{n+1} detektierte Nullstelle lokalisieren, ändert man den Wert von K von 1 auf $(\frac{d\tau}{dt})^{-1}$, integriert (8) von $\tau = S(\mathbf{y}_n)$ bis $\tau = 0$ und erhält somit den Systemzustand $\mathbf{x}(t_e) = \mathbf{y}(0)$ zum Ereigniszeitpunkt t_e , sowie t_e selbst. Anschließend wird K wieder zurück auf 1 gesetzt und mit der Integration von (8) bei $\tau = t_e$ fortgesetzt.

2.2 Vergleiche

Es wurde die Henon-Implementierung mit dem Regula-Falsi-Verfahren verglichen. Um den Vergleich möglichst fair zu gestalten, wurde nicht der Matlab-eigene Lokalisierungsmechanismus verwendet, sondern Regula-Falsi analog zu Henon, als Wrapper-Funktion implementiert. Betrachtet wurde die Performance beim Lösen des gedämpften und ungedämpften Bouncing-Ball Beispiels, unter Berücksichtigung der benötigten Sample und des gemachten Fehlers. Die zu

lokalisierenden Ereignisse waren dabei die Nullstellen der Ball-Flugbahn sowie deren Maximumsstellen. Als ODE-Solver wurden die Matlab-Solver *ode23* und *ode45*, als auch ein eigens programmierter fix-step-Solver verwendet, wobei letzterer nach einem expliziten Runge-Kutta-Verfahren 4. Ordnung und 4. Stufe arbeitet. Der fix-step-Solver dient zum gezielten Aufzeigen der Grenzen des Henon-Algorithmus(siehe Abschnitt 2.3).

Beispiel	Solver	$\frac{t_{RF}}{t_H}$	$\frac{t_{ML}}{t_H}$
BB	<i>ode23</i>	8.75	0.19
d BB	<i>ode23</i>	3.13	0.14
BB M	<i>ode23</i>	0.60	0.11
d BB M	<i>ode23</i>	1.06	0.10
BB	<i>ode45</i>	4.76	0.35
d BB	<i>ode45</i>	3.13	0.14
BB M	<i>ode45</i>	4.43	0.35
d BB M	<i>ode45</i>	2.98	0.25

Tabelle 1: Ergebnis Laufzeitvergleich:

t_H ... Laufzeit der Henon-Implementierung
 t_{RF} ... Laufzeit der Regula-Falsi-Implementierung
 t_{ML} ... Laufzeit des Matlab-Mechanismus
 BB ... Bouncing Ball
 BB M ... Bouncing Ball mit Maximumslokalisierung
 d BB ... gedämpfter Bouncing Ball
 d BB M ... ged. Bouncing Ball mit Maximumslokalisierung

Tabelle 1 zeigt die Laufzeit-Resultate der Bouncing-Ball Simulationen. Die Zahlen in den letzten beiden Spalten stellen dabei nicht die konkreten Laufzeiten dar, sondern das Verhältnis von Laufzeit des Regula-Falsi- bzw. des MATLAB-Verfahrens zu Laufzeit des Henon-Verfahrens. Diverse genauigkeitsbestimmende Parameter der Henon- und der Regula-Falsi-Implementierung wurden dabei so gewählt, dass der resultierende Fehler in etwa dem des Matlab-Mechanismus entspricht. Es ist zu erkennen, dass Henon in den meisten Fällen schneller arbeitet als Regula Falsi. Unter Verwendung von *ode23* benötigt Henon beim gewöhnlichen BB sogar weniger als 1/8 der Laufzeit von Regula-Falsi. Dafür ist beim BB mit Maximum-Lokalisierung Henon langsamer als Regula-Falsi. Allgemein hat sich herausgestellt, dass eine große Schrittweite des zugrundeliegenden Solvers das Laufzeitverhältnis zu Gunsten von Henon beeinflusst. Ein Indiz dafür liefert die Tatsache, dass

Henon unter Verwendung von *ode45* im Schnitt 3-mal so schnell arbeitet wie Regula-Falsi. Mit dem Matlab-internen Ereignislokalisierungsmechanismus kann jedoch keine der beiden selbst programmierten Versionen mithalten. Dies ist wohl auf die “umständliche” Implementierung als *Wrapper-Funktion* zurückzuführen.

2.3 Grenzen

An seine Grenzen stößt das Henon-Verfahren, wenn die Ableitung der Ereignisfunktion im Bereich der zu lokalisierenden Nullstelle sehr klein wird. Da die Ableitung beim Henon-Algorithmus in invertierter Form auftritt, kommt es in diesem Fall zu numerischen Schwierigkeiten, die sich in einem, mitunter massiv erhöhten, lokalen Fehler niederschlagen. Dies ist beispielsweise dann der Fall, wenn die analytische Lösung des ODE-Systems $x(t) = 1 - \varepsilon + \sin(t)$ lautet, wobei ε als sehr klein vorausgesetzt wird und sowohl die Nullstellen als auch die Extrema lokalisiert werden sollen. In diesem Fall liegen nämlich die Nullstellen und das Minimum von $x(t)$ sehr nahe beieinander. Abbildung 2 zeigt das Ergebnis einer solchen Simulation mit einem ε -Wert von 10^{-10} , wobei der Bereich der Nulldurchgänge stark vergrößert wurde. Man sieht, dass die Henon-Lösung ab dem lokalisierten Minimum stark von der analytischen Lösung abzuweichen beginnt.

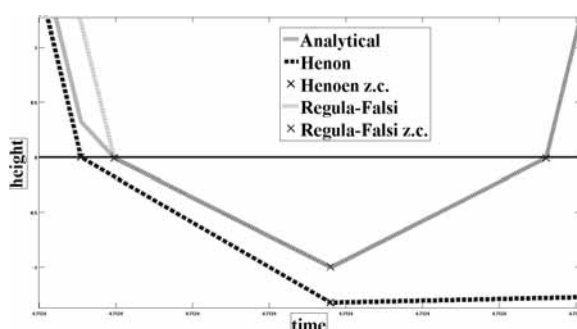


Abbildung 2: Die Lokalisierung der Nullstellen und des Minimums einer verschobenen Sinusfunktion. Die unterste strichlierte Linie zeigt die Lösung des Henon-Algorithmus. Die durchgehend gezeichnete Linie zeigt die abgetastete analytische Lösung und die dritte Linie (nur zu Beginn nicht deckungsgleich mit der analytischen Lösung) zeigt die Regula-Falsi-Lösung.

3 Zusammenfassung und Ausblick

Zusammenfassend ist zu sagen, dass das Henon-Verfahren durchaus Potential hat mit dem Regula-Falsi Verfahren mitzuhalten. Es ergeben sich jedoch eine Vielzahl von genauigkeitsbeeinflussenden Parametern, deren Kalibrierung eine Aufgabe für sich darstellt. Es wäre zudem notwendig mehrere verschiedenartige Beispiel zu betrachten und sich nicht, wie hier, im Wesentlichen auf den Bouncing Ball einzuschränken. Der Matlab-Ereignislokalisierungsmechanismus arbeitet um einiges schneller, was jedoch auf die “umständliche” Implementierung des Henon-Algorithmus als *Wrapper-Funktion*, aber auch auf einige Laufzeit-Optimierungstricks, die wohl beim Matlab-Mechanismus zum Einsatz kommen, zurückzuführen ist. In Sachen “Laufzeit-Optimierungstricks” schlummert jedoch ohne Zweifel auch im Henon-Algorithmus noch ungenutztes Potential. Weitere Untersuchungen wären somit angebracht.

References

- [1] M. Hénon: On the numerical Computation of Poincaré Maps, *Physica D: Nonlinear Phenomena*, 5(2), S. 412–414, 1982.

A robust schedule for Montpellier's Tramway network

Oliver Ullrich¹, Daniel Lückcrath², Ewald Speckenmeyer²

¹National Science Foundation's Industry-University Cooperative Research Center
Florida International University, ECS 243C, 11200 SW 8th St, Miami FL-33199

²Institut für Informatik, Universität zu Köln, Albertus-Magnus-Platz, 50923 Köln, Germany
oullrich@fiu.edu

The city of Montpellier in the Languedoc-Roussillon region of France features a fast growing tram network as a central part of its public service infrastructure. Here, as in many other tram networks, resources like tracks and stations are shared between different lines. Because of the resulting dependencies, small inevitable delays can spread through the network and affect its global performance.

This article examines whether a robust tram schedule may help to raise punctuality in Montpellier's tram network. To accomplish this, we apply a tool set designed to generate schedules optimized for robustness, which also satisfy given sets of planning requirements. These tools allow to compare time tables with respect to their punctuality and other key indicators.

After an introduction to the goals of this paper, we continue with a description of the tool set focusing on optimization and simulation modules. These software utilities are then employed to generate and simulate robust and non-robust schedules for Montpellier's tram network, which are subsequently compared for the resulting delays.

1 Introduction

The city of Montpellier in southern France is growing fast, its population has tripled in the last fifty years (see [6]). As major part of the city's public service infrastructure, the Tramway tram network is provided by Transports de l'agglomération de Montpellier (TAM). The first Tramway line was launched in 2000, it connects the eastern and western suburbs to the city center. Since then three more lines commenced operation. By now, about 282,000 passengers are served on each weekday (see [17]), which amounts to about half of the population of Montpellier's metropolitan area. Three more tram lines are commissioned, the first of which is planned to commence operation in 2017.

In Montpellier's tram network, several lines share resources like platforms, switches and tracks. Because of the resulting dependencies, small local delays can propagate to succeeding trams, build up to larger delays, and thus affect the network's global performance.

In this paper, we explore whether a robust schedule can help to reduce delays in Montpellier's Tramway network. We define robustness as the degree to which inevitable small delays are kept local to the immediately affected tram and do not spread through the network. To examine this, we apply a software tool chain which enables us to generate robust schedules, compare their feasibility and evaluate their punctuality and other key indicators.

This paper continues with a description of our approaches on optimization and simulation of tram schedules (section 2). It then focuses on the modeling and simulation of Montpellier's Tramway system. Robust and non-robust schedules are generated, simulated, and compared concerning the resulting delays (section 3). The paper closes with a short summary of lessons learned and some thoughts on further research (section 4).

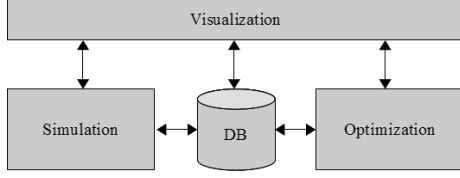


Figure 1: Architecture of project CATS

2 Simulation and optimization of tram schedules

The project “Computer Aided Traffic Scheduling” (CATS) is built around a database complying with the ÖPNV5 data model proposed by the association of German transport companies (Verband Deutscher Verkehrsunternehmen, see [23]). Visualization, optimization, and simulation modules are connected via operations on the database and through XML configuration files (see figure 1). Due to its compliance with the ÖPNV5 data model our framework is capable of working on many European tram networks.

For an in-depth description of the optimization method, see [21]. A more detailed discussion of the simulation software can be found in [7].¹

2.1 Optimization of tram schedules

Various approaches to optimize tram and railway schedules are known (see e.g. [1, 3, 4, 5, 9, 10, 12]). Most of them aim at one general objective like minimizing vehicle delay (see [10, 12]) or maximizing robustness to restrict the global impact of small, local disturbances (see [4, 5]). Others use a combination of objectives, like operational profit and robustness in [3], or combining social opportunity cost and operational cost in [11].

Because of the complex nature of the problem, many authors use heuristic approaches like Lagrangian heuristics (see [3]) or simulated annealing (see [11]). Others (see [1]) introduce exact algorithms for restricted subclasses, like chain and spider networks.

In our project, we combine heuristics and exact methods to generate optimal synchronized time tables for tram networks, targeting maximal robustness and ad-

herence to transport planning requirements at the same time. Those planning requirements originate from political, economic and feasibility reasons. Thus it is not sufficient to exclusively consider a general goal like robustness when generating time tables.

We use the scheduled time offset between two consecutively departing vehicles at a platform as an indicator for robustness. In an assumed tact interval of ten minutes, two lines could be scheduled with equidistant offsets of five minutes, which means that one or both involved vehicles could be late for more than four minutes without consequences for the following tram. Under an extremely unequal split of the available time span into a nine minute offset followed by a one minute offset, the first tram could have a delay of more than eight minutes without consequences to the following vehicle. On the other hand, would the second vehicle be even slightly late, the delay would spread to the follow-up tram. Since we are assuming typically small delays, we see an equidistant distribution as very robust, the occurrence of very small offsets as not robust.

So, to calculate the robustness of a time table λ we examine at each platform h of the network the scheduled time offset $\delta_{f, \text{pred}(f)}(h, \lambda)$ between any trip f and its predecessor $\text{pred}(f)$, i.e. the time elapsed between the departures of $\text{pred}(f)$ and f at platform h .

To reduce complexity we aggregate subsequent similar platforms operated by the same lines to a maximal platform type h' , weighted by the number of included platforms $\phi_{h'}$ (see figure 2). The reduced set of platforms is denoted by H' .

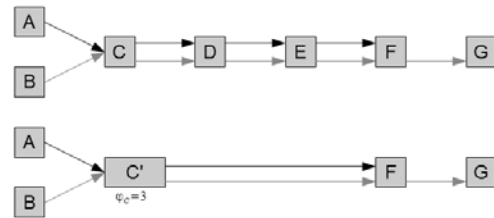


Figure 2: Example for platform reduction

To calculate the robustness Φ_a of schedule λ , we add the inverse of $\delta_{f, \text{pred}(f)}(h', \lambda)$ for each platform type $h' \in H'$ and all its trips, thus applying a penalty for

¹This section is an abbreviated version of [21], section 1.

small safety distances. With $f \in F_{h'}$ representing all trips that serve platform type h' under schedule λ , and $\phi_{h'}$ as the number of platforms represented by h' , the resulting function is as follows:

$$\Phi_a(\lambda) = \sum_{h' \in H'} \sum_{f \in F_{h'}} \frac{1}{\delta_{f, \text{pred}(f)}(h', \lambda)} * \phi_{h'} \quad (1)$$

Given is a set R of planning requirements, with r denoting a single requirement $r \in R$. In order to calculate the compliance with transport planning requirements we introduce $\rho_r(\lambda) \in \{1, 2, 3, \infty\}$ the compliance factor of requirement r under a schedule λ . A compliance factor of 1 means that the requirement is completely satisfied, 2 and 3 denote tolerable compliance, and ∞ means that the constraint is not met and the time table candidate λ must be rejected. We add the compliance values for all $r \in R$ and get the following:

$$\Phi_b(\lambda) = \sum_{r \in R} \rho_r(\lambda) \quad (2)$$

Depending on the network under consideration and the number of planning requirements, the two parts of the objective function may not be comparable directly. Thus we define a normalizing factor σ , which reflects the relationship between the lower bounds of both parts. The theoretically optimal distance $\delta_{f, \text{pred}(f)}^{opt}(h')$ of two trips $\text{pred}(f)$ and f on platform type h' is obtained by dividing the tact interval by the number of serving lines at that platform type. The best possible compliance factor ρ_r^{min} of a planning requirement $r \in R$ is the minimal value assigned by the planner, independent of the characteristics of the examined solution candidate. Typically $\rho_r^{min}=1$. We define σ as:

$$\sigma = \left(\sum_{h' \in H'} \sum_{f \in F_{h'}} \frac{1}{\delta_{f, \text{pred}(f)}^{opt}(h')} \right) / \sum_{r \in R} \rho_r^{min} \quad (3)$$

Combining $\Phi_a(\lambda)$ and $\Phi_b(\lambda)$ yields the overall objective function $\Phi(\lambda)$ (see equation 4), normalized by σ and weighted by a factor $0 \leq \alpha \leq 1$, the relative weight of the fulfillment of planning requirements.

$$\Phi(\lambda) = (1 - \alpha) * \sum_{h' \in H'} \sum_{f \in F_{h'}} \frac{1}{\delta_{f, \text{pred}(f)}(h', \lambda)} * \phi_{h'} + \alpha * \sigma * \sum_{r \in R} \rho_r(\lambda) \quad (4)$$

In our experiments, this weight is set to $\alpha = 0.5$, so that robustness and the fulfillment of requirements are equally important.

A valid solution also has to adhere to some other constraints. The first restriction requires each start time μ_i of each line variant i to be inside the tact interval, with $t_{interval}$ being the duration of the interval (see equation 5).

$$\forall i \leq |\lambda| : 0 \leq \mu_i < t_{interval} \quad (5)$$

Another restriction requires an offset of at least one minute between two departures of the trips f and $\text{pred}(f)$ at each platform type $h' \in H'$ (see equation 6). This means that no platform can be blocked by more than one train at any point of time, the schedule has to be free of collisions.

$$\forall h' \in H' : \forall f \in F : \delta_{f, \text{pred}(f)}(h', \lambda) > 0 \quad (6)$$

We identify seven types of transport planning constraints: Interval constraints, start time constraints, core line constraints, bidirectional track constraints, turning point constraints, warranted connection constraints and follow-up connection constraints. Upon closer inspection (see [20], section 6.2.3) it becomes clear that interval and start time constraints are fundamental and all other constraint types can be expressed using these two. E.g. a bidirectional track constraint can be expressed by two interval constraints covering opposing platforms. Subsequently only interval and start time constraints are considered in the remainder of this paper.

The presented model is implemented as a branch-and-bound solver, which starts with an initial solution computed by a genetic algorithm for performance reasons. For implementation details see [21] or [20], sections 6.2.4 and 6.2.5.

2.2 Simulation of tram schedules

Most rail-bound traffic simulation models are designed for long distance train or railway networks, see e.g. [8, 9]. While those systems feature similarities to tram networks, e.g. passenger exchange or maneuvering capabilities, they differ significantly in important aspects. Tram networks are often mixed, i.e. trams

travel on underground tracks as well as on street level, and are thus subject to individual traffic and corresponding traffic regulation strategies. Subsequently, tram behavior is a mixture between train and car behavior, e.g. line-of-sight operating/driving. Therefore a simple adaption of railway simulation methodologies is not feasible.

Our application is based upon a model-based parallelization framework (described in [20] and [22]), which exploits the embedded model's intrinsic parallelism. The mixed tram network is modeled as a directed graph with platforms, tracks and track switches represented by nodes. Connections between nodes are represented as edges. The distributions for the duration of passenger exchange are specific to platform and tram type with the combined duration of opening and closing the vehicle doors as minimum value. Vehicles encapsulate most of the simulation dynamics, which are based upon the event based simulation approach (as described in [2]). Thus trams change their state at events of certain types, like stopping or accelerating, which happen at discrete points in time. These state changes may trigger a change in the over-all system state and generate follow-up events, which are administrated in a priority queue.

Tram attributes are specified by the type of tram, which holds functions for the maneuvering capabilities, e.g. acceleration and braking. Main parameters of the simulation are the maximum driving velocity v_{max} , the dawdle probability $0 \leq p_d \leq 1$ (which maps the chance that a tram’s driver does not accelerate at a given moment due to external causes), and the dawdle factor $d > 1$ (which maps the amount of the delay caused by dawdling). For the experiments, these values are set as $p_d = 0.3$ and $d = 1.3$. A more detailed description of model and implementation can be found in [7].

3 Examining Montpellier's Tramway network

We apply the described software suite to Montpellier’s Tramway network (for an overview see figure 3) based on the time table data of 2013 (gathered from [13], [14], [15], and [16]). The system consists of 84 stations with 176 platforms and 46 track switches, con-

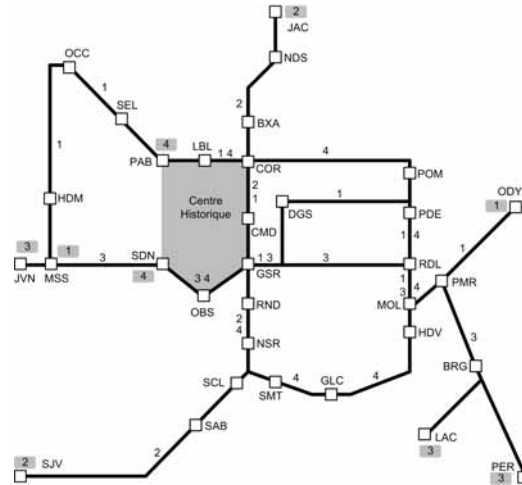


Figure 3: Montpellier's Tramway network

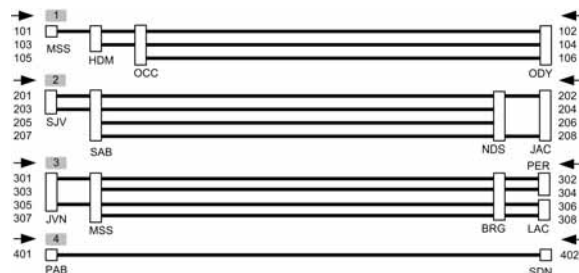


Figure 4: Montpellier's line routes

nected via 232 tracks (see [19]). These tracks cover a total length of ca. 56 kilometers, resulting in an average track length of about 241 meters. 1,215 trips per operational day are executed on four lines with 24 line routes (see figure 4), about 282,000 passengers are served on each weekday (see [17]).

3.1 Schedule generation

The schedule implemented by TAM has no global tact interval, trains serve the routes in varying patterns through the day. At peak times, lines 1 and 2 are traversing the city center every four to five minutes, with changing headway. Line 3 is served every six to eight minutes, the intervals between consecutive trains of line 4 are alternating between eight and nine minutes. To find an appropriate approximation of this, we assume a tact interval of eight minutes, and insert additional core lines 1A and 2A to double the frequency

of lines 1 and 2 to four minutes. A set of planning requirements is defined, which can be decomposed to 16 interval constraints. These include the additional core lines 1A and 2A, and minimum turn-around times at line ends.

The genetic algorithm is initialized with a population of 450 randomly generated individuals. The best fitness value of this first generation is 75.55 (average: 83.58, worst value: 95.00). In the course of 500 generations and a runtime of 313 seconds, the algorithm finds a best solution candidate with a fitness value of 75.25 (average: 75.51, worst value: 80.11). The branch-and-bound solver further enhances the minimal fitness value in the course of a 200 seconds run down to 75.22, and finds 128 optimum solutions.

3.2 Comparing generated schedules

We pick ten schedules each out of both the pool of initial solutions and the optimum solution pool and execute ten simulation runs for each of those 20 schedules. The maximum velocity is set to $v_{max} = 40$ km/h, a compromise between the observed inner city maximum speed of 30 km/h and the higher speed in some regions outside the city with exclusive track usage.

The runs under the initial schedules yield an average delay of all departures of 9.8 seconds. Under the best schedules the average delay is 8.2 seconds, which means a reduction of 16.3 percent or 1.6 seconds. The average delay of all delayed departures is reduced from 25.8 by 2.3 seconds or 8.9 percent to 23.5 seconds.

The frequency distribution of occurring delays was also collected (see figure 5). Under the optimal schedules, the numbers of delays in each bucket are reduced. This effect is especially significant for the larger delays of more than 60 seconds (see figure 6). The total number of departures with a larger delay is reduced from 521.3 under the random schedules by 210.7 departures or 40.4 percent down to 310.6.

As seen, robust schedules reduce the average delay in the Tramway network, though only by a small amount, and significantly reduce the number of larger delays. Under optimum schedules with their better distributed time offsets, many small delays can be made up for fast and do not spread to consecutive departures. A

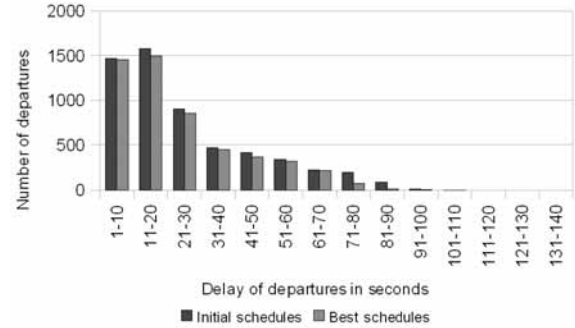


Figure 5: Frequency distribution of delays

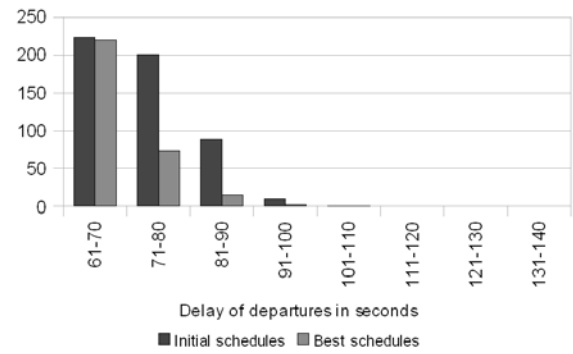


Figure 6: Frequency distributions of larger delays

higher robustness can thus help to reduce the number of larger delays by preventing inevitable small delays from accumulating over the simulation run.

To take a more detailed look at the model's behavior, we pick a typical schedule A (see table 1) with an objective function value of 92.69 from the genetic algorithm's initial pool of valid solution candidates, and a schedule B (see table 2) from the pool of best solutions. We examine both schedules by executing 100 simulation runs each and comparing the results.

Line/Route	1	1A	2	2A	3	4
01	0	6	3	1	7	3
02	5	3	3	1	6	2

Table 1: Scheduled departures at the routes' starting points under schedule A

Line/Route	1	1A	2	2A	3	4
01	6	1	4	0	1	5
02	3	7	7	3	0	5

Table 2: Scheduled departures at the routes' starting points under schedule B

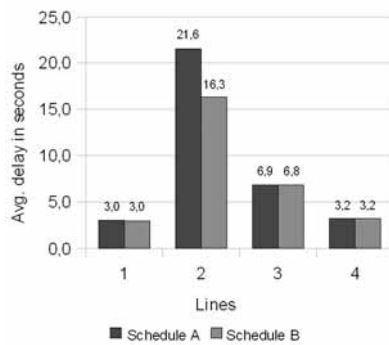


Figure 7: Average delay of lines

On average, schedule A yields a line delay of 8.7 seconds, which is reduced under schedule B by 17.2 percent or 1.5 seconds to 7.3 seconds. The only line that yields a significantly lower delay under the optimum schedule is line 2, with a reduction of 24.3 percent from 21.6 to 16.3 seconds (see figure 7).

To examine this, we take a closer look at trips 3 and 4 of tram 2005 (see figure 8), which serves the shorter routes 205 and 206 of line 2A. While the measured delays at several platforms vary, the most obvious differences are found in the regions of the town center around Corum (COR, see figure 3) and Gare Saint-Roch (GSR).

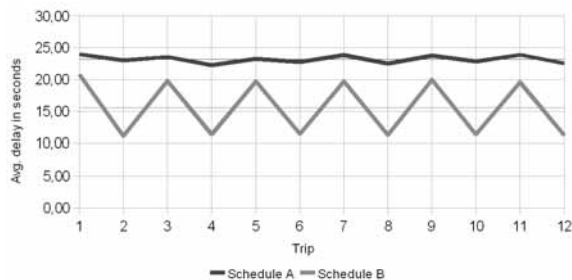


Figure 8: Average trip delays of tram 2005 serving line 2A

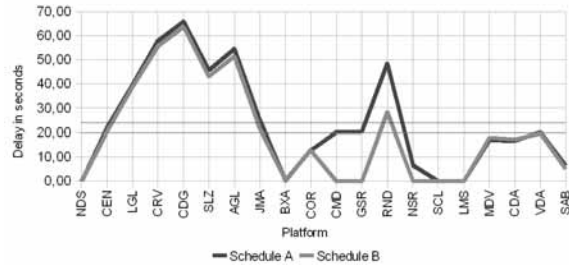


Figure 9: Average delays at platforms of trip 3 of tram 2005

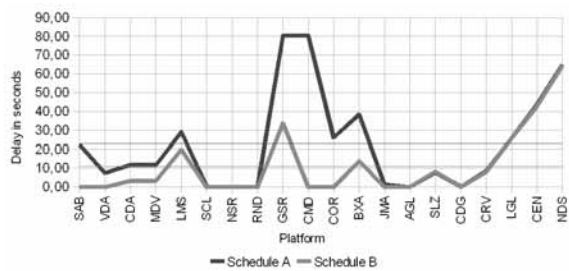


Figure 10: Average delays at platforms of trip 4 of tram 2005

Serving trip 3 in the direction of Sabines (see figure 9), trams of line 2A enter an array of switches they share with lines 1, 1A, 2 and 4 after the departure at Corum. Under schedule A, the vehicle has to wait to access these common resources, and cannot regain the resulting delay until after the stop at Nouveau Saint-Roch (NSR). Under schedule B with its better distributed time offset, these resources are instantly accessible to the tram.

On the return trip in the direction of Notre-Dame de Sablassou (see figure 10), the tram has to navigate four consecutive switches between the stations Rondelet (RND) and Gare Saint-Roch. It shares some of these switches with all other lines. Under the random schedule A, the vehicle gets behind a tram serving line 1, although it is scheduled to precede it by one minute. It therefore has to wait for that tram to clear the Gare Saint-Roch platform and thus gets a delay of about 80 seconds. It can start to regain the delay after lines 1 and 2 split course before Corum station.

As described, only line 2 (and its companion line 2A) shows a significantly lower delay under the robust schedule B, the other lines yield the same values under both schedules. Lines 1 and 4 run in parallel for a

while, then part way and rejoin after sections of different track lengths and planned driving times (see figure 3). Because of this, and considering that the optimizer can only generate valid schedules with a time offset of at least one minute at each platform, these lines are locked in relation to each other under all valid schedules. There is no way the optimizer can generate a better (or worse) schedule concerning these two lines. The same applies to the combination of lines 3 and 4: These are also locked under any valid schedule.

This phenomenon does not occur with line 2, which runs parallel to lines 1 and 4, but only in one contiguous section each. It does then split from these lines but does not rejoin them later. These lines are therefore not locked, the optimizer can schedule line 2 more freely.

The experiments show that the application of a robust schedule can help to reduce delays in Montpellier's tram network. They also show that robustness has its main impact in those regions of the networks where resources are shared by most line routes. In the Tramway network these regions are switch arrays near the stations Gare Saint-Roch and Corum.

3.3 TAM's applied schedule

To complete the picture of Montpellier's Tramway network, we also examine the schedule applied by TAM at the time of this writing. Because it adheres to no common tact interval and comprehends planning requirements unknown to the authors, the results cannot be compared directly to the generated schedules. Therefore, no insights about special traffic phenomena should be assumed.

As described in section 3.1, TAM's schedule has no common tact interval. Therefore, the numbers of the started trips per hour deviate in the sample period of 08.00 to 16.59 (see figure 11) from their counterparts of the generated schedules.

The data gathered by executing 100 simulation runs with the described parameters shows TAM's schedule to be in general range with the generated schedules. The average delay of departures of 8.1 seconds is slightly smaller than the value yielded by schedule B, and 1.7 seconds smaller than that of schedule A. The average delay of delayed departures has a value

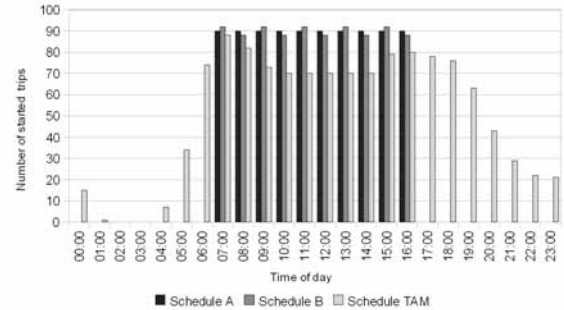


Figure 11: Number of started trips per hour

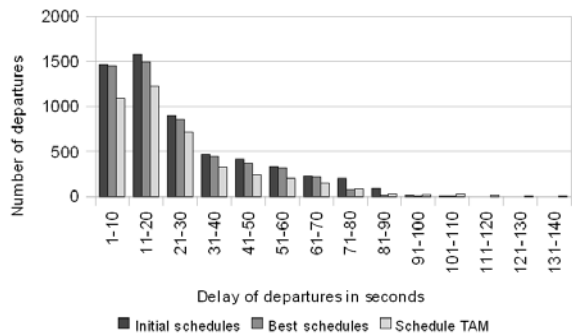


Figure 12: TAM's applied schedule - Frequency distribution

of 24.4 seconds and is therefore splitting the distance between schedule A (25.8 seconds) and schedule B (23.5 seconds). The number of larger delays is 314.4, on about the level of schedule A's value.

The frequency distribution shows that TAM's schedule yields a lower number of small delays, which are compensated by a higher number of delays of more than 70 seconds (see figure 12).

TAM's schedule yields line delays which are comparable to the values resulting from the generated schedules (see figure 13): line 1 has the same value under all three schedules, line 2 lies between the values of schedules A and B, lines' 3 and 4 delay values are a bit higher than their counterparts.

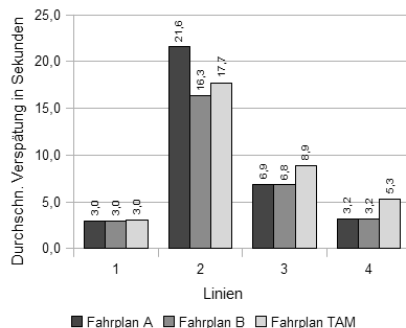


Figure 13: TAM's applied schedule - Average delay of lines

4 Conclusions and further research

This article showed an approach to examine the influence of robustness on a tram network. To accomplish this, we applied optimization and simulation tools designed to evaluate schedules optimized for robustness. These software utilities were employed to generate and compare robust and non-robust schedules for Montpellier's tram network, demonstrating that a robust schedule can indeed help to reduce delays in the Tramway network. The experiments showed that the main improvements center in those regions of the networks where resources are shared by most line routes. In the presented case these regions are the switch arrays near the stations Gare Saint-Roch and Corum.

Montpellier's Tramway network is expanding: a line 5 is currently being built and will connect the fast growing suburbs in the north and west to the inner city (see [18]). Supporting a rerouted line 4, this line will complete the ring track around the historical city center. Line 5 is planned to commence operation in 2017. The city of Montpellier already commissioned lines 6 and 7, their exact routes are still under consideration. Our model will be expanded with representations of these lines, the resulting model's behavior will be analyzed and compared to the existing model's.

We also plan to analyze under which general circumstances a robust schedule will increase punctuality in a tram network. The presented results of Montpellier's Tramway and of our hometown Cologne's KVB network (see [21]) will be utilized as a base for this.

5 Acknowledgements

This material is partially based upon work supported by the National Science Foundation under grants I/UCRC IIP-1338922, AIR IIP-1237818, SBIR IIP-1330943, III-Large IIS-1213026, MRI CNS-0821345, MRI CNS-1126619, CREST HRD-0833093, I/UCRC IIP-0829576, MRI CNS-0959985, and FRP IIP-1230661 and U.S. Department of Transportation under a TIGER grant.

References

- [1] Bampas, E., Kaouri, G., Lampis, M., Pagourtzis, A.: Periodic Metro Scheduling. In: Proceedings of ATMOS, 2006.
- [2] Banks, J., Carson, J.S., Nelson B.L., Nicol D.M.: Discrete-Event System Simulation. Pearson, 2010.
- [3] Cacchiana, V., Caprara, A., Fischetti, M.: A Lagrangian Heuristic for Robustness, with an Application to Train Timetabling. Transportation Science, Volume 46, Issue 1, pp. 124-133, 2012.
- [4] Caimi, G., Fuchsberger, M., Laumanns, M., Schüpbach, K.: Periodic Railway Timetabling with Event Flexibility. In: Networks, Volume 57, Number 1, pp. 3-18, 2010.
- [5] Genç, Z.: Ein neuer Ansatz zur Fahrplanoptimierung im ÖPNV: Maximierung von zeitlichen Sicherheitsabständen. Dissertation, Mathematisch-Naturwissenschaftliche Fakultät, Universität zu Köln, 2003.
- [6] Institut national de la statistique et des études économiques: La population de Montpellier Agglomération a triplé au cours des cinquante dernières années. http://www.insee.fr/fr/themes/document.asp?ref_id=16088, accessed on June, 26th, 2013.
- [7] Lückcrath, D., Ullrich, O., Speckenmeyer, E: Modeling time table based tram traffic. In: Simulation Notes Europe (SNE), ARGESIM/ASIM Pub., TU Vienna, Volume 22, Number 2, pp. 61-68, 2012.

- [8] Middelkoop, D., Bouwman, M.: SIMONE: Large Scale Train Network Simulations. In: B.A. Peters, J.S. Smith, D.J. Medeiros, M.W. Rohrer (Ed.): Proceedings of the 2001 Winter Simulation Conference, Arlington, 2001.
- [9] Nash, A., Huerlimann, D.: Railroad Simulation Using OpenTrack. In: Allan, J., R.J. Hill, C.A. Brebbia, G. Sciutto, S. Sone (Ed.): Computers in Railways IX, WIT Press, Southampton, pp. 45-54, 2004.
- [10] Schöbel, A.: A Model for the Delay Management Problem based on Mixed-Integer-Programming. In: Proceedings of ATMOS, 2001.
- [11] Speckenmeyer, E., Li, N., Lückcrath, D., Ullrich, O.: Socio-Economic Objectives in Tram Scheduling. Technical Report, Universität zu Köln, to appear.
- [12] Suhl, L., Mellouli, T.: Managing and preventing delays in railway traffic by simulation and optimization. In: Mathematical Methods on Optimization in Transportation Systems, pp. 3-16, 2001.
- [13] Transports de l'agglomération de Montpellier (Ed.: Subra, R.): Horaires tram 1. 2012.
- [14] Transports de l'agglomération de Montpellier (Ed.: Subra, R.): Horaires tram 2. 2012.
- [15] Transports de l'agglomération de Montpellier (Ed.: Subra, R.): Horaires tram 3. 2012.
- [16] Transports de l'agglomération de Montpellier (Ed.: Subra, R.): Horaires tram 4. 2012.
- [17] Transports de l'agglomération de Montpellier: Un réseau en étoile. http://www.montpellier-agglo.com/tam/page.php?id_rubrique=31, accessed on Feb, 21st, 2013.
- [18] Transports de l'agglomération de Montpellier: Le tracé du ligne 5. http://www.ligne5-montpellier-agglo.com/?page_id=16, accessed on Feb, 21st, 2013.
- [19] Transports de l'agglomération de Montpellier (Ed.: Subra, R.): Plan du réseau du centre de l'agglomération. 2012.
- [20] Ullrich, O.: Modellbasierte Parallelisierung von Anwendungen zur Verkehrssimulation - Ein dynamischer und adaptiver Ansatz. Dissertation, Univ. Köln, 2014.
- [21] Ullrich, O., Lückcrath, D., Franz, S., Speckenmeyer, E.: Simulation and optimization of Cologne's tram schedule. In: Simulation Notes Europe (SNE), ARGESIM/ASIM Pub., TU Vienna, Volume 22, Number 2, pp. 69-76, 2012.
- [22] Ullrich, O., Lückcrath, D., Speckenmeyer, E.: Model-based parallelization of discrete traffic simulation models. Submitted to ASIM 2014 - 22nd Symposium Simulation Technique, Berlin, 2014.
- [23] Verband Deutscher Verkehrsunternehmen e.V.: VDV-Standardschnittstelle Liniennetz/Fahrplan, VDV-Schriften 452, 2008.

Multiskalenansatz für reaktive Prozess- und Ablaufplanung

Matthias Bode, Volker Berkhahn
Institut für Bauinformatik, Leibniz Universität Hannover
bode@bauinf.uni-hannover.de

Es wird ein Algorithmus vorgestellt, der es ermöglicht das gekoppelte Problem von Prozess- und Ablaufplanung zu betrachten. Durch die Verwendung eines reaktiven Multiskalenansatzes wird eine Optimierung auf zwei verschiedenen Detailstufen durchgeführt. Dadurch kann auf geänderte Nebenbedingungen reagiert werden, ohne den gesamten zuvor generierten Plan zu verwerfen. Zur Optimierung auf makroskopischer Ebene wird ein koevolutionärer Algorithmus verwendet, auf mikroskopischer Ebene wird ein memetischer Algorithmus verwendet. Letzterer bettet eine Heuristik zur lokalen Verbesserung in einen genetischen Algorithmus ein. Für beide Algorithmen wird eine implizite Codierung der Lösungskandidaten entwickelt, die den realen Entscheidungsprozess nachbildet und dazu führt, dass keine Reparatur der Lösungskandidaten des genetischen Algorithmus nötig ist.

1 Einleitung

Im Rahmen der Produktion und Logistik sind die Prozess- und Ablaufplanung von essentieller Bedeutung. Bei der Prozessplanung wird für einen Prozess die Sequenz von Operationen bestimmt, die für seine Fertigstellung nötig sind. Die Ablaufplanung beschäftigt sich mit der Fragestellung, zu welchem Zeitpunkt die zuvor in der Prozessplanung bestimmten Operationen ausgeführt werden sollen. Zumeist werden in der Literatur beide Fragestellungen separat behandelt. Für die gesamtheitliche Optimierung ist es allerdings wichtig, die Wechselwirkungen zwischen beiden Teilbereichen zu berücksichtigen.

Um eine gewisse Flexibilität bei der Ablaufplanung zu ermöglichen, wird in vielen Veröffentlichungen ein *Multi-Mode-Resource-Constrained-Project-Scheduling-Problem* (MMRCPS) betrachtet, bei dem eine Operation in verschiedenen Moden ausgeführt werden kann. Erst in den letzten Jahren wurden Verfahren vorgestellt, die auf einer kombinierten Betrachtung von Prozess- und Ablaufplanung basieren. Qassim zeigt das Problem am Beispiel der Schiffsfertigung und bezeichnet es als *Integrated-Process-Planning-And-Scheduling-Problem* (IPSP) (vgl.[1]).

Die meisten realen Projekte unterliegen sehr vielen äußeren Einflüssen. Heutzutage gehört es zu den An-

forderungen an Algorithmen zur Erstellung von Projektplänen, flexibel auf solche Einflüsse reagieren zu können. In der Vergangenheit wurden zumeist holistische Optimierungsverfahren verwendet, bei denen der gesamte Zeitraum eines Projektes betrachtet und die Wahl der Operationen und deren Reihenfolge optimiert wurde (z.B. [2], [3]). Bedingt durch äußere Einflüsse können jedoch Änderungen der Nebenbedingungen auftreten und die ursprünglichen Lösungen ungültig werden. Eine der Schlüsselanforderungen an ein Optimierungsverfahren ist somit die Reaktionsfähigkeit auf plötzliche Änderungen der Nebenbedingungen.

Um diesen Anforderungen gerecht zu werden, wurde ein Multiskalenansatz entwickelt, der verschiedene Detailstufen des Optimierungsproblems betrachtet. Anhand eines Anwendungsbeispiels der Optimierung des Betriebes auf einem Containerterminal wird der Algorithmus getestet.

2 Problemdefinition

Inadäquate Prozess- und Ablaufplanung führt zu einer Verminderung der Produktivität. Daher ist deren Optimierung eine der Kernherausforderungen in der Produktion und Logistik. Im folgenden Abschnitt wird

dieses Optimierungsproblem mathematisch formuliert und anhand einer realen Problemstellung motiviert.

2.1 Anwendungsbeispiel Containerterminal

Intermodale Containerterminals sind Binnenlandeinfahrungen, auf denen Ladeeinheiten mit unterschiedlichen Transportsystemen zwischen verschiedenen Verkehrsträgern wie Zügen und Lastkraftwagen umgeschlagen werden. Der Umschlag findet größtenteils durch Portalkrane statt, die mehrere Gleise und LKW-Spuren überspannen. Auf größeren Containerterminals werden die Portalkrane durch Längsförderanlagen unterstützt, um die kostenintensiven Längsfahrten der Portalkrane zu minimieren. Diese Längsförderanlagen umfassen mehrere eigenständige Einheiten, die einen Transport der Ladeeinheiten ermöglichen. Hierbei gibt es schienenungebundene Varianten und schienenungebundene wie beispielsweise *Automatic-Guided-Vehicle* (AGV). Auf einigen Umschlagterminals sind außerdem mobile Umschlaggeräte im Betrieb, die Ladeeinheiten in Zwischenlager transportieren, die nicht direkt an einem der Umschlagmodule positioniert sind. Im Bereich der Prozessplanung muss unter Berücksichtigung von ggf. nötigen Zwischenabstellungen bestimmt werden, wie die Ladeeinheiten von ihrem Startort zu ihrem Bestimmungsort transportiert werden können. Die Ablaufplanung bestimmt daraufhin den Startzeitpunkt der Operationen auf den einzelnen Transportsystemen. Abbildung 1 stellt schematisch die Infrastruktur auf einem Containerterminal dar.

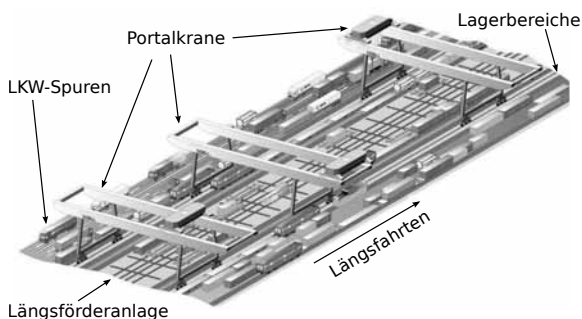


Abbildung 1: Schematischer Aufbau eines Umschlagterminals mit Zug- und Lastkraftwagenverkehr sowie einer Längsförderanlage im zentralen Bereich des Terminals. [4]

Zur Modellierung des Beispiels wurde ein objektorientierter Ansatz gewählt, der es ermöglicht die Bestandteile eines Containerterminals im Rahmen einer algorithmischen Optimierung zu nutzen. Es werden zwei verschiedene Detaillierungsstufen der Einplanungsvorschriften für die einzelnen Transportsysteme definiert. So kann bei einer makroskopischen Betrachtung die Längsförderanlage als *Black-Box* angefragt werden, die lediglich angibt, ob eine Ladeeinheit theoretisch transportiert werden kann und wie lange dieser Transport dauert. Bei dieser *Black-Box*-Implementierung wird auf eine Kollisionsabfrage zwischen den Transporteinheiten der Längsförderanlage verzichtet. Ebenso wird auf makroskopischer Ebene die Einplanung bei den Portalkranen vorgenommen. Hat der Kran innerhalb des angefragten Zeitfensters theoretisch die Kapazität um die Ladeeinheit zu transportieren, so wird diese Operation eingeplant. Für eine mikroskopische Betrachtung werden die Kranbewegungen im Detail abgebildet und es wird auf Kollisionen getestet, ebenso wird die Längsförderanlage abgebildet.

2.2 Prozessplanung

Die Prozessplanung auf einem Containerterminal beschäftigt sich mit der Fragestellung, über welche Transportsysteme eine Ladeeinheit von ihrem Start- zu ihrem Zielort befördert werden kann. Die Prozessplanung legt fest, welche Ladeeinheiten über eine Kranstafette mit Zwischenabstellung im Lager, über die Längsförderanlage des Terminal, oder mittels Direkttransport ausgeführt werden müssen, um das Gesamtsystem bezüglich einer zuvor definierten Zielfunktion zu optimieren.

Der Transport einer Ladeeinheit impliziert einen Job J_i , der in der Menge aller Jobs $J = \{J_1, \dots, J_i, \dots, J_n\}$ enthalten ist. Für jeden Job J_i gibt es eine Menge von Folgen von Operationen $M_{J_i} = \{m_{J_i}^0, \dots, m_{J_i}^k, \dots, m_{J_i}^m\}$.

Eine Folge $m_{J_i}^k = (O_{ik1}, \dots, O_{ikp})$ beschreibt dabei eine mögliche Reihenfolge von Teilaufgaben, die zur Fertigstellung des Jobs ausgeführt werden müssen. Eine solche Operationenfolge m_j^k wird im Folgenden Route genannt. Bei der Prozessplanung wird für jeden Job J_i aus der zugehörigen Menge M_{J_i} eine Route ausgewählt und somit festgelegt, durch welche Folge von Operationen der Job abgearbeitet wird. Durch die

Auswahl einer Route pro Job erhält man eine Menge V aller einzuplanenden Operationen.

Transportsysteme haben Arbeitsbereiche, die festlegen, in welchem Raum sie grundsätzlich operieren können. Diese Arbeitsbereiche überlappen sich teilweise und bilden Austauschbereiche, in denen Ladeeinheiten von einem Transportsystem zum anderen übergeben werden können. So existieren beispielsweise zwischen Portalkranen Austauschbereiche, die Kranstafetten für einen sukzessiven Transport durch verschiedene Portalkrane ermöglichen. Ebenso gibt es feste Austauschbereiche für die Übergabe einer Ladeeinheit zwischen den Portalkranen und der Längsförderanlage. Diese Austauschbereiche bilden die Knoten eines gerichteten Graphens, die über Kanten, die die Transportsysteme repräsentieren, verbunden sind (siehe Abbildung 2).

Um eine Auswahl an mögliche Routen zwischen Startressource und Zielressource eines Jobs zu generieren, wird ein *k-kürzeste-Wege-Algorithmus* verwendet (vgl. [5]). Diese Alternativrouten dienen als Lösungskandidaten für die Optimierung der Prozessplanung.

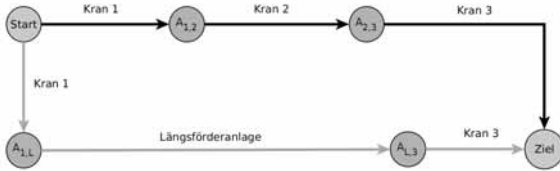


Abbildung 2: Einfacher Transportgraph, der zwei verschiedene Routen ermöglicht. Der obere Weg über eine Kranstafetten und die entsprechenden Austauschbereiche zwischen den Kranen ($A_{1,2}$ und $A_{2,3}$), sowie eine Alternativroute über die Längsförderanlage.

2.3 Ablaufplanung

Den Operationen $O_i \in V$, wobei V die Menge alle Operationen der Operationensequenzen enthält, die in der Prozessplanung ausgewählt wurden, werden in der Ablaufplanung feste Startzeiten S_i zugewiesen.

Ablaufplanprobleme sind durch Vorgängerbeziehungen, sogenannte technologische Vorschriften, zwischen Operationen gekennzeichnet. Hierbei sind zunächst Vorgängerbeziehungen innerhalb eines Job zu nennen, die durch die Anordnung in der Operationensequenz gegeben sind. Weiterhin führen Vorgänger-

beziehungen zwischen unterschiedlichen Jobs auch zu Vorgängerbeziehungen zwischen Operationen aus verschiedenen Jobs. Dies ist beispielsweise der Fall, wenn eine Ladeeinheit sich auf einem Zugplatz befindet, der gleichzeitig Ziel einer anderen Ladeeinheit ist.

Vorgängerbeziehungen lassen sich durch ein *Activity-on-Node-Diagramm* darstellen, bei dem die Knotenmenge V die Menge aller einzuplanenden Operationen repräsentiert. Falls eine Vorgängerbeziehung zwischen zwei Operationen O_i und O_j besteht, werden diese beiden Operationen mit Hilfe einer Kante $\langle i, j \rangle \in E$ der Kantenmenge E in Relation gesetzt. Der zeitliche Abstand zwischen den Startzeitpunkten zweier Aktivitäten O_i und O_j wird über das Kantengewicht δ_{ij} beschränkt:

$$S_j - S_i \geq \delta_{ij} \quad i, j \in V, \text{ falls } \langle i, j \rangle \in E. \quad (1)$$

Bei der Einplanung einer Operation werden zusätzlich sequenzabhängige Rüstfahrten berücksichtigt. So muss beispielsweise der Weg des Krans vom Zielort der vorangegangenen Operation zur Startposition der nachfolgenden ebenfalls einplanbar sein. Das bedeutet, dass der entsprechende Kran auch für die Rüstfahrt verfügbar und die Fahrt möglich sein muss. Das Kantengewicht ergibt sich zu $\delta_{ij}(x) = S_i + p_i + s_{ij}$, wobei S_i dem Startzeitpunkt der vorangegangenen Operation, p_i dessen Dauer und s_{ij} der Rüstzeit entsprechen.

Weiterhin sind der früheste Start ES_i , der späteste Start LS_i sowie das früheste Ende EE_i und das späteste Ende LE_i einer Operation i durch die Ankunftszeit der Ausgangsressource und die Abfahrtszeit der Zielressource geben. Die Definition der Zielfunktion ist vom spätesten Start abhängig (siehe 2.4), wohingegen der früheste Start eine weitere Restriktion des Problems bildet:

$$S_i \geq ES_i \quad (i \in V) \quad (2)$$

Zur Berücksichtigung der Ressourcenbeschränkung dient das *Active-Set-Prinzip*, das alle Operationen umfasst, die zu einem Zeitpunkt t ausgeführt werden:

$$\mathcal{A}(t) = \{i \in V | S_i \leq t < S_i + p_i\} \quad (t \geq 0) \quad (3)$$

Die Ressourcenauslastung der Ressource κ zum Zeitpunkt t ergibt sich zu

$$r_\kappa(t) = \sum_{i \in \mathcal{A}(t)} r_{i\kappa} \quad (t \geq 0), \quad (4)$$

mit dem Ressourcenbedarf $r_{i\kappa}$ der Operation i an der Ressource κ . Die Ressourcenauslastung $r_\kappa(t)$ darf die Ressourcenkapazität R_κ zu keinem Zeitpunkt t überschreiten, folglich muss gelten:

$$r_\kappa(t) \leq R_\kappa \quad (0 \leq t \leq \bar{d}), \quad (5)$$

wobei \bar{d} den Endzeitpunkt des Projektes beschreibt.

2.4 Optimierungsgrößen

Zumeist wird bei Ablaufproblemen die Minimierung der Gesamtprojektdauer als Optimierungsziel festgesetzt. Da das betrachtete Problem jedem Job ein eigenes Zeitfenster zuweist, in dem er abgefertigt werden muss, wäre diese Zielsetzung hier wenig aussagekräftig.

Daher ist das oberste Optimierungsziel die Minimierung der Anzahl an Jobs, die nicht rechtzeitig fertig gestellt werden können, der sogenannten DNF-Jobs. Ein Job J_i wurde nicht rechtzeitig fertig gestellt, wenn der mögliche Startzeitpunkt S_{ikp} einer Operation O_{ikp} der gewählten Route $m_{j_i}^k$ des Jobs größer als deren späteste Startzeitpunkt LS_{ikp} ist. Es lässt sich eine Indikatorfunktion definieren

$$\chi_{DNF} : J_i \mapsto \begin{cases} 1, & \text{falls } \exists O_{ikp} : S_{ikp} > LS_{ikp} \\ 0, & \text{sonst.} \end{cases} \quad (6)$$

Ein weiteres Optimierungsziel des betrachteten Problems ist die Minimierung der Rüstfahrten. Vor allem die Längsbewegungen der Krane sind sehr energintensiv, weswegen eine Operationenreihenfolge gesucht wird, die zu einer Minimierung der Rüstfahrten der Krane in Längsrichtung führt.

Als zu minimierende Zielfunktion wird ein mit dem Parameter α gewichtetes Mittel zwischen der Anzahl der nicht rechtzeitig abgefertigten Jobs und der Rüstfahrten der Krane untersucht:

$$\min \alpha \cdot \left(\sum_{i \in J} \chi_{DNF}(i) \right) + (1 - \alpha) \left(\sum_{\langle i,j \rangle \in E} s_{ij} \right). \quad (7)$$

3 Reaktiver Multiskalenansatz

Durch die Unsicherheit innerhalb der zeitlichen Restriktionen ist eine detaillierte Optimierung des gesamten Projekttraum nicht zu empfehlen. Die zumeist

rechenintensiv berechnete Lösung wird durch kleinere Änderungen an den Restriktionen unzulässig und kann nicht mehr genutzt werden. Gerade bei dem vorgestellten Containerterminal-Problem kann es zu Verspätungen der Züge und dadurch bedingten Änderungen an den zeitlichen Restriktionen der Jobs kommen. Daher wird ein Multiskalenansatz gewählt, der das Problem zunächst auf makroskopischer und später auf mikroskopischer Ebene betrachtet.

Das Ziel der Optimierung auf makroskopischer Ebene ist es, unter Vernachlässigung von exakten Nebenbedingungen und Informationen einen optimalen Ablaufplan zu generieren, der einen Prozessplan und eine grobe Operationenreihenfolge umfasst. Der generierte Prozessplan und die grobe Operationenreihenfolge dienen zur Abschätzung über den gesamten Projektzeitraum. So ist es möglich, Aussagen über die Machbarkeit zu treffen und auftretende Engstellen und Zeitfenster mit besonderer Ressourcenauslastung zu identifizieren um bereits im Vorhinein darauf reagieren zu können. Zur Optimierung auf makroskopischer Ebene wird der in Abschnitt 4 vorgestellte Algorithmus verwendet.

Um die Prozess- und Ablaufplanung als begleitendes System zur Entscheidungsfindung im praktischen Betrieb nutzen zu können, ist eine Betrachtung der Begebenheiten auf möglichst hohem Detailniveau nötig. Um die Anforderungen an die eingesetzte Computerhardware zu verringern, wird ein *Rolling-Horizon*-Algorithmus eingesetzt (vgl. [6]). Im Rahmen eines solchen Algorithmus wird eine detaillierte Betrachtungsweise des Ressourcenbedarfs lediglich innerhalb eines relativ kleinen Zeitfensters, ausgehend vom momentanen Zeitpunkt, angewandt. Iterativ werden einzelne Zeitfenster abgearbeitet, wobei bereits ausgeführte Operationen unverändert bleiben, Entscheidungen für zukünftige Operationen hingegen unter Umständen angepasst werden.

Die auf makroskopischer Ebene bestimmte Lösung dient als Initiallösung für eine Optimierung des aktuell betrachteten Zeitfensters. Es wird eine detaillierte Ablaufplanung mit detaillierteren Ressourcenanfragen und Kapazitätsüberprüfungen der Operationen durchgeführt, die im aktuellen Zeitfenster des makroskopischen Ablaufplans liegen. Hierzu dient der in Abschnitt 5 vorgestellte memetische Algorithmus.

Anpassungen auf mikroskopischer Ebene führen zu

Änderungen des Prozess- und Ablaufplans auf makroskopischer Ebene und haben somit Einfluss auf die nachfolgenden Zeitfenster. Parallel zur lokalen Optimierung innerhalb des aktuellen Zeitfensters wird der Gesamtplan über die gesamte Projektdauer in regelmäßigen Abständen neu evaluiert und optimiert. Folglich ist eine Änderung des Prozessplans und dementsprechend der Operationensequenz nur für den noch nicht eingeplanten Teil einer Route möglich.

Aufgrund der Modellierung des Problems auf mehreren Ebenen und der Verwendung der zuletzt bestimmten optimierten Lösungen ist es möglich, schnell auf Änderungen der Restriktionen zu reagieren ohne einen komplett neuen Prozess- und Ablaufplan zu generieren.

4 Makroskopische Optimierung

Zur Lösung des Optimierungsproblems auf makroskopischer Ebene wurde ein koevolutionärer genetischer Algorithmus entwickelt. Ein genetischer Algorithmus nach Holland ([7]) zählt zur Klasse der Evolutionären Algorithmen, deren Funktionsweise von der Evolution natürlicher Lebewesen inspiriert ist.

4.1 Koevolutionärer Algorithmus

Um die Wechselwirkungen der Prozess- und Ablaufplanung zu berücksichtigen wird ein koevolutionärer Algorithmus entwickelt bei dem sich zwei Populationen parallel entwickeln. Population I dient zur Optimierung des Prozessplans, Population II zur Optimierung der Operationenreihenfolge. Abbildung 3 beschreibt schematisch den Ablauf des entwickelten Algorithmus.

Die Fitness eines einzelnen Individuums aus einer der beiden Populationen beschreibt, wie gut es mit Individuen aus der jeweils anderen Population kooperiert, um einen guten Prozess- und Ablaufplan zu generieren. Da die Operationenreihenfolge direkt von der Prozessplanung abhängig ist, ist eine Aussage über die Fitness eines Prozessplans nur möglich, wenn eine Ablaufplanung vorgenommen wurde. Daher erfolgt die Evaluation immer am Gesamtproblem. Das heißt, dass bei der Decodierung und Evaluation eines Individuums aus der Prozessplanungspopulation ein Indi-

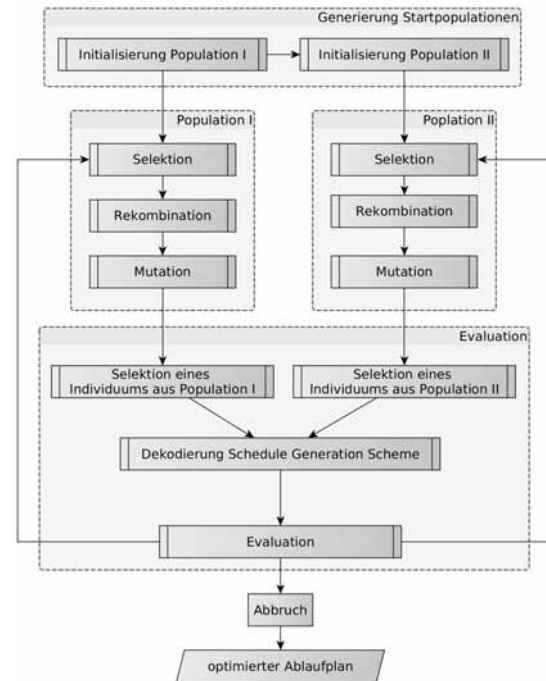


Abbildung 3: Ablauf des koevolutionären Algorithmus

duum aus der Ablaufplanungspopulation mittels eines geeigneten Selektionsverfahrens ausgewählt wird und die Fitness aus den Eigenschaften beider Individuen bestimmt wird. Analog verhält es sich bei der Fitnessbestimmung eines Individuums aus der Operationenreihenfolge.

4.2 Implizite Codierung

Im Rahmen eines genetischen Algorithmus für Ablaufplanprobleme werden die Startzeiten der einzuplanenden Operationen zumeist nicht direkt in die Codierung des Problems übernommen, da der Lösungsraum damit sehr groß werden würde und es schwierig wird, die Gültigkeit der Lösungen zu gewährleisten. Aus diesem Grund werden Ablaufpläne oft als Sequenz von Operationen über sogenannte permutationscodierte *Activity-Lists* repräsentiert (vgl. beispielsweise [2] und [3]). Zur Dekodierung wird ein *Schedule-Generation-Scheme* (SGS) verwendet, der einen gültigen Ablaufplan erzeugt.

Die erwähnten Codierungen haben den Nachteil, dass durch die Zunahme der Prozessplanwahl die generier-

te Lösungen häufig ungültig sind und durch Reparaturmechanismen angepasst werden müssen. Dabei tritt bei der Reparatur wiederum die Frage auf, auf welche Art und Weise ungültige Operationenreihenfolgen oder zeitlich nicht realisierbare Prozessplankombinationen aufgelöst werden sollen. Um diese Problematik zu vermeiden, wird hier sowohl für den Prozessplan als auch die Operationenreihenfolge eine implizite Codierung entwickelt, die zusammen mit einem angepassten SGS garantiert gültige Lösungen erzeugt.

Bei der Entwicklung der Codierung lässt sich für die Prozessplanung zunächst festhalten, dass nicht für jeden Job die optimale Operationensequenz gewählt werden kann, um ein Optimum bezüglich globaler Ziele zu erreichen. Für einige Jobs muss eine andere als die beste Route gewählt werden, um z.B. kapazitive Restriktionen zu erfüllen. Weiterhin bestehen zwischen Jobs Vorgängerbeziehungen (siehe Abschnitt 2.3). Das heißt, dass ein Job vor einem anderen begonnen oder beendet sein muss. Analog zum *Activity-on-Arrow-Diagramm* lässt sich einen Graph modellieren, dessen Zusammenhangskomponenten angeben, welche Jobs miteinander in Beziehung stehen. Für jeden Job einer solchen Komponente wird nun versucht, die zuvor bestimmte beste Route einzuplanen. Sollte dies nicht der Fall sein, da nicht alle zeitlichen Restriktionen eingehalten werden können, so wird versucht, die nächstmögliche Route einzuplanen. Innerhalb dieses iterativen Verfahrens wird nun für jeden Job der Zusammenhangskomponente eine gültige Route bestimmt. Die Codierung für eine Optimierung mittels eines genetischen Algorithmus definiert für ein Individuum aus der Population der Prozessplanoptimierung pro Zusammenhangskomponente eine Sortierstrategie, in der für die einzelnen Jobs Routen eingeplant werden. Wird ein Job durch eine Sortierstrategie erst später ausgewählt und eingeplant, so kann es sein, dass durch bereits festgelegte Routen anderer Jobs eine schlechtere Route für diesen Job gewählt werden muss. Die Sortierung der Jobs innerhalb einer Zusammenhangskomponente für die Einplanung hat somit einen impliziten Einfluss auf die Routenwahl.

Die aus dem Prozessplan stammenden Operationen werden mittels eines angepassten zeitorientierten parallelen SGS eingeplant. Bei einem parallelen SGS wird die Teilmenge $A_t \in V$ von Operationen bestimmt, die zum aktuellen Entscheidungszeitpunkt einplanbar sind. Bei einem auf *Activity-Lists* basierendem SGS

gibt die Liste eine Priorität vor, mit dem die jeweilige Operation ausgewählt wird, wenn sie Element von A_t ist. Bei der kombinierten Betrachtung von Prozess- und Ablaufplanung birgt eine Codierung mittels *Activity-Lists* diverse Schwierigkeiten. Daher werden für die Reihenfolge des Einplanens der Operationen aus A_t wiederum Sortierstrategien codiert. Für jeden Entscheidungspunkt definiert das Individuum eine Sortierstrategie, die auf die Teilmenge der ausführbaren Operationen A_t innerhalb eines Zeitschrittes des SGS angewendet wird. Tabelle 1 bietet einen Überblick über die zur Verfügung stehenden Sortierstrategien, z.B. über den frühesten Start ES_j einer Operation oder dessen Dauer p_j . Dabei handelt es sich überwiegend um Sortierstrategien, die in der Literatur empfohlen und dort exklusiv für die gesamte Projektdauer angenommen werden (vgl. [9]). Es lässt sich festhalten, dass die Codierung der jeweiligen Sortierstrategie einen impliziten Einfluss auf die Startzeiten der Operationen hat.

Bezeichnung	Formel
Earliest Start Time	$\min(ES_j)$
Earliest Finish Time	$\min(EF_j)$
Late Start Time	$\min(LS_j)$
Late Finish Time	$\min(LF_j)$
Min Slack	$\min(LS_j - ES_j)$
Max Slack	$\max(LS_j - ES_j)$
Shortest Processing Time	$\min(p_j)$
Longest Processing Time	$\max(p_j)$
Least Immediate Successors	$\min(S_j)$
Most Immediate Successors	$\max(S_j)$
Most Total Successors	$\max(\bar{S}_j)$
Greatest Rank Positional Weight	$\max(p_j + \sum_{i \in S_j} p_i)$
Least Rank Positional Weight	$\min(p_j + \sum_{i \in S_j} p_i)$

Tabelle 1: Mögliche Sortierstrategien für Operationen, angelehnt an [9]

Im Rahmen des SGS wird weiterhin für die aktuell betrachtete Operation je nach Detaillevel überprüft, ob die Kapazität aller beteiligten Ressourcen für die Dauer der Operation nicht überschritten wird. Eventuell nötige Rüstzeiten der jeweiligen Operation werden zusätzlich berücksichtigt. Werden alle Nebenbe-

dienungen eingehalten, so wird die Operation eingeplant und die Auslastung der Ressourcen aktualisiert. Sollte dies nicht der Fall sein, so wird versucht, die nächste Operation der sortierten Menge einzuplanen.

Für den genetischen Algorithmus werden *Uniform-Crossover*- und *Swap-Mutation*-Operatoren gewählt. Sowohl bei der Auswahl der Eltern als auch für die Evaluierung am Gesamtproblem wird ein fitnessproportionaler Selektionsalgorithmus gewählt. Weiterhin werden einige der besten Lösungen im Sinne eines Elitismus unverändert in die nächste Generation übernommen.

5 Mikroskopische Optimierung

Auf mikroskopischer Ebene wurde ein memetischer Algorithmus entwickelt, der auf der impliziten Codierung der Population für die Operationenreihenfolge basiert. Bei einem memetischen Algorithmus handelt es sich um eine Erweiterung eines klassischen genetischen Algorithmus um problemspezifische Informationen. Die Kernidee nach Mosacto [8] besteht darin, die globale genetische Suche mit einer geeigneten heuristischen lokalen Suche zu kombinieren. In welchen Phasen des genetischen Algorithmus eine lokale Suche durchgeführt wird, muss von Problemstellung zu Problemstellung abgewogen werden.

Zunächst werden auf dem Teilproblem für das aktuell betrachtete Zeitfenster Lösungskandidaten durch Rekombination und Mutation entsprechend der Population für die Operationenreihenfolge des koevolutionären Algorithmus erzeugt. Diese Lösungskandidaten werden durch eine lokale Suche auf dem Phänotypen verbessert. In diesem Fall wurde auf jedem erzeugten Individuum ein heuristisches Verfahren angewandt. Hierzu wird eine Nachbarschaft definiert und innerhalb dieser iterativ nach einer besseren Lösung gesucht. Die lokale Suche als heuristische Komponente ist im Gegensatz zum genetischen Algorithmus stark an die Problemstellung und die Optimierungsziele angepasst. Da hier die Minimierung der nicht rechtzeitig abgefertigten Ladeeinheiten und die Minimierung der Rüstfahrten in der Zielfunktion berücksichtigt werden, wird hier eine Heuristik vorgestellt, die nach diesen zwei Kriterien versucht, eine bessere Lösung in der Nachbarschaft zu finden.

Betrachtet man einen Ablaufplan, der mittels des SGS erzeugt wurde, lässt sich das Activity-on-Node Diagramm um zusätzliche Kanten, die die Vorgängerbeziehungen auf einer Ressource beschreiben, erweitern. Diese sogenannten kapazitiven Kanten geben die Operationenreihenfolge auf einer bestimmten Ressource an. Auf dem erweiterten Graphen lassen sich nun einige Methoden anwenden, die aus der Netzplantechnik bekannt sind. So gibt es für jeden nicht fertiggestellten Job mindestens einen Pfad, auf dem die Operationen liegen, die für die nicht termingerechte Abfertigung verantwortlich sind. Ein solcher Pfad lässt sich als kritischer Pfad des Teilproblems der Abfertigung eines bestimmten Jobs interpretieren. Um eine lokale Verbesserung zu erreichen, wird zunächst für jeden nicht fertiggestellten Job der kritische Pfad bestimmt. Diese Pfade lassen sich mittels ihrer kapazitiven Vorschriften in kritische Blöcke unterteilen. Es werden folglich Teilpfade erzeugt, auf denen sich nur Operationen befinden, die auf ein und der selben Ressource ausgeführt wurden.

Nun lässt sich eine Nachbarschaft definieren, die durch das Tauschen von Operationen innerhalb eines kritischen Blocks entsteht. Da die Rüstzeiten sequenzabhängig sind, kann theoretisch jeder Tausch innerhalb eines kritischen Blocks zu einer Verbesserung führen. Da die Evaluierung eines neuen Phänotypen innerhalb der lokalen Suche eines kompletten Neudurchlaufes des SGS bedarf und dadurch aufwendig ist, wird zunächst abgeschätzt, ob sich die Rüstzeit durch einen Tausch verringern würde. Ist dies der Fall, so wird der Phänotyp weiter untersucht, sonst wird er verworfen. Das entstandene lokale Optimum bezüglich der gewählten Heuristik wird daraufhin in die nächste Generation des genetischen Algorithmus aufgenommen.

6 Fazit und Ausblick

Es wurde ein Algorithmus vorgestellt, der es ermöglicht, die Prozess- und Ablaufplanung als gekoppeltes Problem zu betrachten. Um auf Änderungen an den Restriktionen reagieren zu können, wurde ein Multiskalenansatz mit verschiedenen Detailstufen entwickelt, der sowohl den gesamten Projektzeitraum als auch ein aktuelles Zeitfenster betrachtet. Zur Lösung des kombinatorischen Optimierungsproblems

auf makro- Ebene wurde ein koevolutionärer memetischer Algorithmus entwickelt, der eine implizite Codierung über Entscheidungsstrategien nutzt. Auf mikroskopischer Ebene wird eine detaillierte Ablaufplanung mittels eines memetischen Algorithmus vorgenommen, wobei das Verfahren um ein heuristisches lokale Suche erweitert wurde. Die implizite Codierung bildet den realen Entscheidungsprozess nach und nutzt dieses problemspezifisches Wissen zur Generierung von Lösungen. Dies führt dazu, dass nur gültige Lösungen erzeugt werden und die Problematik der Handhabung ungültiger Lösungen und somit der sonst schwierigen Einhaltung von Nebenbedingungen innerhalb eines genetischen Algorithmus umgangen wird. Die verwendete Heuristik ist an die Optimierungsziele angepasst und führt zu einer lokalen Verbesserung der Lösungen, die im genetischen Teil generiert wurden.

Für praxisnahe Tests des entwickelten Algorithmus wurden diverse Auswertungsroutinen implementiert. Die mittels des Algorithmus gewonnenen Ergebnisse wurden mit Messungen auf realen Containerterminals verglichen. Hierzu wurde eine event-diskrete Simulation entwickelt, die ein virtuelles Verladen der Ladeeinheiten gemäß des erzeugten Ablaufplans ermöglicht und den Ablauf visualisieren kann.

Als Vergleichsmaterial dienten Aufzeichnungen des Betriebs über zwei Monate auf einem exemplarischen Containerterminal. Die Ergebnisse zeigen, dass die bestimmten Prozess- und Ablaufpläne plausibel sind und darüber hinaus gegenüber den Messungen eine signifikante Rüstfahrtminimierung bei gleichbleibender Abfertigungsmenge der Ladeeinheiten ermöglichen.

Da die Optimierung auf makroskopischer Ebene einen sehr langen Zeitraum umfasst, sind Informationen über Nebenbedingungen mit einer Unschärfe behaftet. So können Verfügbarkeitszeiten von Ressourcen Störungen unterliegen. Weiterhin wird durch die niedrigere Detaillierungsstufen eine Modellunschärfe eingeführt, die ebenfalls berücksichtigt werden muss. Diese beiden Arten von Unschärfe lassen sich mit Hilfe von Fuzzy-Intervallen aus der Fuzzy-Set-Theorie modellieren. Mittels dieser ist es möglich, einen Fuzzy-Ablaufplan zu erstellen. Dieser kann dahin gehend optimiert werden, dass er robust gegenüber Störungen ist.

Literatur

- [1] R. Y. Qassim. *Integrated Process Planning and Scheduling and Multimode Resource Constrained Project Scheduling: Ship Block Assembly Application*. In: Manufacturing System, Dr. Faiza Abdul Azi, S. 381-404, 2012.
- [2] J. Alcaraz, C. Maroto und R. Ruiz. *Solving the Multi-Mode Resource-Constrained Project Scheduling Problem with genetic algorithms*. Journal of the Operational Research Society, S. 614-626, 2003.
- [3] M. Cervantes, A. Lova, P. Tormos et al. *A Dynamic Population Steady-State Genetic Algorithm for the Resource-Constrained Project Scheduling*. New Frontiers in Applied Artificial Intelligence Lecture Notes in Computer Science Volume 5027, S. 611-620, 2008.
- [4] K.-P. Franke *Entwicklungsperspektiven im Containerumschlag* in: Jahrbuch der Hafentechnischen Gesellschaft, Band 56, Schifffahrts-Verlag "Hansa", Deutschland, S. 90 - 94, 2010.
- [5] Jin Y. Yen *Finding the K Shortest Loopless Paths in a Network*. Management Science Volume 17, S- 712-716, 1971.
- [6] Z.Li und M. Ierapetritou. *Rolling horizon based planning and scheduling integration with production capacity consideration*. Chemical Engineering Science, Volume 65, Issue 22, S. 5887-5900, 2010.
- [7] J. H. Holland *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, Ann Arbor, 1975
- [8] P. Moscato. *A gentle introduction to memetic algorithms*. Handbook of Metaheuristics International Series in Operations Research & Management Science Volume 57, S. 105-144, 2003.
- [9] M. Masmoudia und A. Hait. *Project scheduling under uncertainty using fuzzy modelling and solving techniques*. Engineering Applications of Artificial Intelligence 1, S. 135-149, 2013.

A SimHeuristic Framework as a Decision Support System for large Logistics Networks with complex KPIs

Felix Dross¹, Markus Rabe²

¹TU Dortmund, Graduate School of Logistics

²TU Dortmund, IT in Produktion und Logistik

felix.dross@tu-dortmund.de

This paper describes the principles of a SimHeuristic Framework that can be used as a Decision Support System for large logistics networks with custom developed Key Performance Indicators (KPIs). The presented solution approach could also be referred to as a multi-objective simulation optimization. In contrast to other contributions to the field, the described multi-objective problem is more complex since KPIs are defined as figures on top of a hierarchical tree of Performance Measures within a Performance Measurement System. They are obtained through complex queries on a Data Warehouse into which all relevant enterprise data are loaded. Currently there exists no method to automatically combine possible actions in logistics networks and project their future effects on different Performance Measurement Systems. The optimization problem in this case represents a combination of continuous and discrete optimization, because the right parameterization and the right combinations of actions need to be found. In this paper, a detailed description of the problem scenario is given and a high-level solution approach is explained. Special attention is given to the accurate specification and categorization of possible actions. The authors explain the challenges and present their ongoing research in the field.

1 Introduction

This paper describes the design of a software framework that is intended to be later used as a *Decision Support System* for large logistics networks with complex, custom-developed Performance Measurement Systems. The problem which motivated the research has originally been proposed by a large, international trading company with over 100 warehouses in different countries and with an inventory of around 150,000 items on permanent stock. The company operates a large, complex and heterogeneous logistics network, which is a multi-echelon network with central, regional and local warehouses. A supply chain is called multi-echelon supply chain, if an item moves through more than one location before reaching the final customer [1]. Each warehouse has a sales division that receives customer orders on a consistent basis. A special characteristic of the logistics network are certain warehouses that can perform value-added services, for example cutting, drilling or milling. Shuttle transfers between the warehouses and the shipment of the goods to the customers are performed using separated fleets with variable amounts of vehicles.

In recent years, the logistics domain experts of the company have developed specialized and complex

Performance Measurement Systems in order to cope with the management of their very complex logistics network. Performance Measurement Systems unite different *Performance Measures* that relate to each other in a hierarchical form, which usually culminates in one *Key Performance Indicator* (KPI) [2]. Utilizing a *Data Warehouse*, the company retrieves their specialized KPIs periodically from continuously gathered data and provides them to the responsible managers in form of monthly reports, individually summarizing the KPIs for their field of interest. In order to ensure constant improvement of the logistics network regarding the KPIs, the company has developed dedicated *KPI Monitoring Systems* (KPIMS). Each KPIMS constantly monitors one KPI and sends an individually composed *KPI Alert* to a responsible manager, if the KPI leaves certain predefined corridors. A KPI Alert generally consists of two parts: A list of facts that caused the KPI to deteriorate and a set of possible actions that could be performed by the addressed manager in order to improve the KPI. While the automated surveillance of the KPIs intuitively seems to be a smart idea, the described scenario can induce unexpected problems which lead to a deterioration of the overall network situation instead of an improvement. One problem in this scenario accrues from the fact that the Performance Measure-

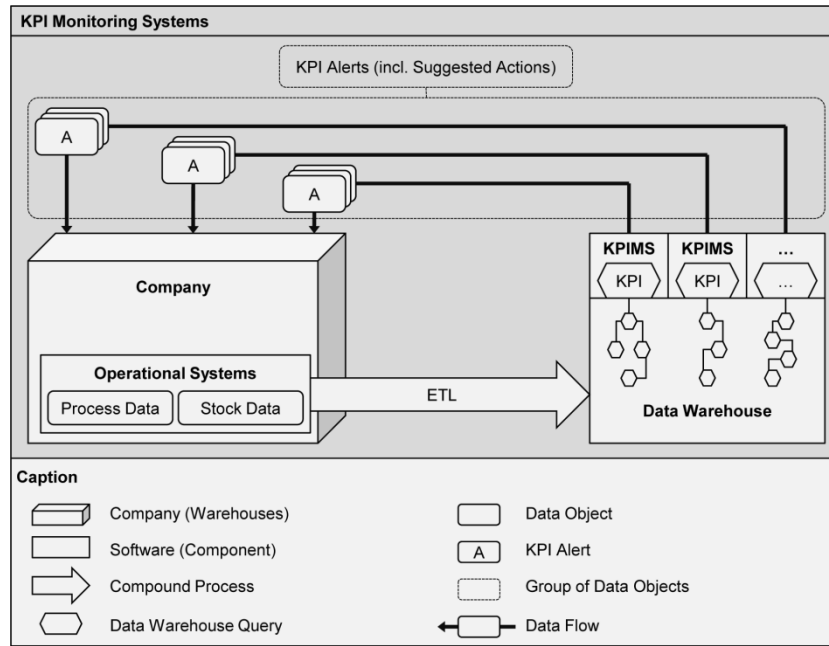


Figure 1. The Situation without the SimHeuristic Framework

ment Systems are designed and maintained by separated, highly specialized teams. They use very specific domain knowledge to define the structure of their own Performance Measurement System with only brief knowledge of the structure of the other complex systems. Each team specifies separately which conditions should trigger a KPI Alert and which content an alert should contain, especially which possible actions should be presented to the receiving manager. Therefore, the different KPIMS are not connected to each other and each alert is sent to the managers of the company independently. As a result, the actions suggested by one KPIMS could improve its own KPI while possibly deteriorating one or more other KPIs. The KPIMS were designed as an approach to improve the overall performance of the logistics network, but in the worst case, if the actions indicated by the KPIMS conflict with each other in too many cases, the whole concept could lead to discontent among the managers and deteriorations instead of improvements within the network. The company is therefore looking for a method to analyze the possible actions in a multi-objective way, i.e. to identify those actions that can improve one or more KPIs while not deteriorating the others. The method should group complementary actions into *Integrated Action Sets* and present them appropriately to the responsible managers.

The authors claim that it is possible to examine the influence of possible actions on the different Performance Measurement Systems in a computer simulation, before the alerts are sent to the responsible managers. The presented research is focused on designing

a method which should be able to find *Action Sets* that can improve one or more KPIs while not significantly deteriorating the other KPIs.

This paper is structured as follows: A description of the problem is given as a foundation for the ongoing research in Section 2. In Section 3, previous, related work in the field is briefly listed. An overview of the suggested solution is given and an outline on why there is additional need for research. The momentarily existing details of a solution approach are presented in Section 4. Finally, first conclusions are drawn and an outlook is given in Section 5.

2 Detailed Problem Description

The underlying mechanisms of the problem, as briefly described in Section 1, are illustrated in Figure 1. The process data and the stock data are continuously gathered by the operational systems. The operational systems include the transactional systems, e.g. the Enterprise Resource Planning System (ERP System) of the company, but potentially also other systems which gather information, e.g. a computer program for the management of transportation contracts. The gathered data are regularly transferred into a Data Warehouse and prepared for further data analysis. This compound, complex process contains the steps *Extract, Transform* and *Load* and is commonly abbreviated as ETL. Within this process, the data are first extracted from the operational systems, transformed to fulfill certain harmonization criteria, and are finally loaded into the Data Warehouse. A very

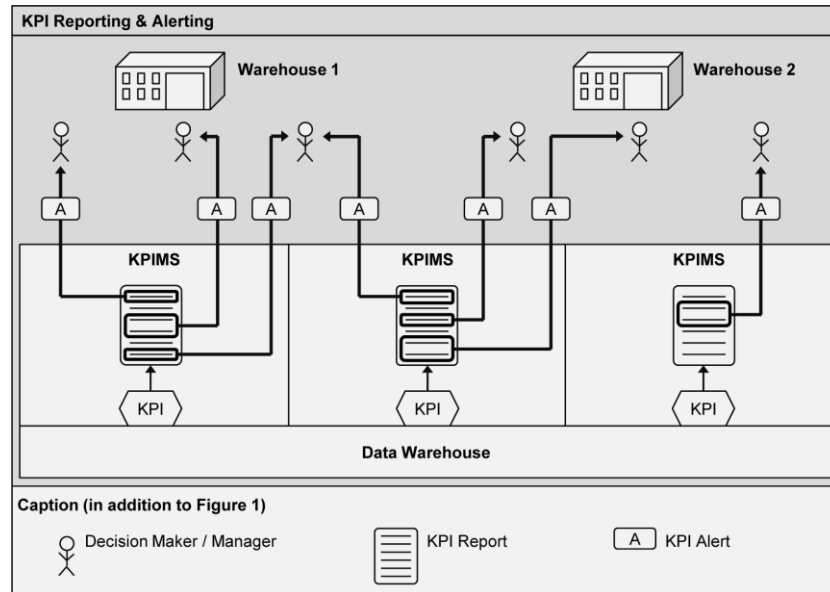


Figure 2. KPI Alerts are generated out of KPI Reports and sent to different managers

important aspect of a Data Warehouse is that both current and historical data are stored. This allows analyses over certain time horizons, which are crucial for identifying patterns and trends within the data. The automated analysis of data with the help of a Data Warehouse is called *Online Analytical Processing* (OLAP). OLAP represents an exploratory, interactive way to perform data analyses on the basis of a conceptual, multidimensional data model. The conceptual, multidimensional data model allows the domain experts for thinking in dimensions and classification hierarchies, which enables them to design complex Performance Measurement Systems and corresponding reports [2]. As mentioned in the introduction, Performance Measurement Systems unite different Performance Measures in a hierarchical form, culminating in one KPI. In Figure 1, the hierarchical structure of each Performance Measurement System is illustrated by a tree of Data Warehouse Queries, each of them culminating in one KPI. In each period, e.g. each month, each KPIMS automatically generates a KPI Report and examines it on the basis of specific, predefined criteria. Currently, the KPIMS use one of three possible ways to trigger a KPI Alert:

1. **Manually defined, fixed nominal value.** If a predefined, fixed nominal value is exceeded, an alert is triggered. The nominal value has to be arranged beforehand by the responsible managers.
2. **Automatically derived, fixed nominal value.** An analysis method such as e.g. regression analysis is used to automatically derive a rea-

sonable nominal value from previously gathered data.

3. Trend analysis.

An alert is triggered, if a KPI is showing a negative course over a longer period in a trend analysis.

A detailed illustration of the relationship between a KPI Report and a KPI Alert is given in Figure 2. After the KPI Alerts have been generated, they are sent to the responsible managers in the company's warehouses. Each warehouse can have up to three responsible managers for different areas of responsibility:

1. The *Purchasing Manager*, responsible for the replenishment of the stock.
2. The *Logistics Manager*, responsible for the distribution of the inventory.
3. The *Sales Manager*, responsible for the turnover of the products.

Depending on the KPIMS, a KPI Alert can have different aggregation levels, i.e. regarding only one article in only one warehouse or up to multiple articles in multiple warehouses. Some KPIMS are rather simple; others are very profound and sophisticated. They all have in common that they offer a list of possible actions that could be performed by the addressed manager in order to improve the respective KPI. Although the authors cannot reveal the detailed specification of each KPIMS, an overview of the monitored aspects is given by a structured list of the 8 KPIMS in Table 1. For a better understanding of the catalogue of possible actions, Table 2 shows how the

actions for the KPIMS "Stock Productivity" can be described systematically using attributes. Because each KPIMS targets different aspects of the logistics network, it also offers different types of actions. This leads to different action attributes for each KPIMS. It is beyond the scope of this paper to provide a table of action attributes for each KPIMS, but it forms an important aspect of the research, further explained in the remainder.

Monitored Area	KPIMS
Inventory	<ul style="list-style-type: none"> Stock Productivity Out-Of-Stock Monitoring Customer Reservations Article-Warehouse Allocations and Inventory Levels (As described in [3])
Staff	<ul style="list-style-type: none"> Productivity-Oriented Incentive System
Transportation	<ul style="list-style-type: none"> Truck Fleet Performance (As described in [4])
Machinery	<ul style="list-style-type: none"> Machine Utilization and Service Level
Delivery	<ul style="list-style-type: none"> Deliveries <i>On Time In Full</i> (OTIF)

Table 1. KPIMS, grouped by the monitored area

Attribute	Description
ID	Unique identifier of a possible action
Name	Descriptive name
Duration	Time until the full effects of the action should be visible (Currently ranging from 1 week to 52 weeks)
Target	Specifying which Performance Measure is mainly influenced
Operator	Concrete action, such as "Increase by X" or "Decrease by X"
Value	Parameterization for the attribute "Operator" (X)

Table 2. Categorization attributes for possible actions for the KPIMS "Stock Productivity"

The decoupled setup of KPIMS can produce highly irritating situations, which can be very demotivating for the receiving manager. For example, a Logistics Manager could receive a "Stock Productivity" alert for one item, and one possible action would be to decrease the stock of the alerted item. He or she performs the action and only a few days later receives another alert from the "Deliveries On Time In Full" KPIMS, which tells him or her to raise the stock of just the same item, because new orders for the item are expected. As a consequence, the manager would probably start to ignore the KPI Alerts, act according to his or her personal opinion or just not react at all. In an organization with decentralized decision makers, this decoupled KPIMS setup could also induce unpredicted conflicts between the receiving managers, if the KPIMS indicate incompatible actions to different receivers.

3 Solution approach and Related Work

The multi-objective optimization of logistics networks through the use of metaheuristics has been a well-researched field in recent years and could be a valid approach to address the company's need for an optimization solution. The class of *Stochastic Combinatorial Optimization Problems* (SCOPs) has been well described in [5] and based on the previously given description of the logistics network of the company it should be possible to reduce the problem to one of the described mathematical problem descriptions. One important aspect of the class of SCOPs is, e.g., that decisions need to be made "on line", i.e. to re-optimize the solution when its initial conditions have changed [6]. This would also hold true for the described situation. However, the complexity of the problem inhibits to just reduce the problem to one of the traditional combinatorial problems with stochastic parameters, such as for example the *Inventory Routing Problem with Stochastic Demands* (IRPSD) [6]. As mentioned above, the KPIs of the logistics network are obtained through complex Online Analytical Processing. Thus, even if it were possible to reduce the problem to one of the above-mentioned traditional problem descriptions, it would not be adequate to apply one of the well-researched metaheuristics for SCOPs, e.g. the ones summarized in the survey of Bianchi et al. [5]. The tailoring of a specialized metaheuristic is prevented, because the structure of each Performance Measurement System is too complex to inject it effectively into a metaheuristic. Furthermore, every KPIMS, including its underlying Performance Measurement System, is constantly improved and thus subject to constant change. It is therefore required to develop a method that is robust

towards changing definitions of KPIs and also robust against changing sets of possible actions. The author's solution approach to the problem is to design an *Automated Interdependence Analysis*, which can analyze the different actions regarding three major goals:

1. Improve the understanding of the interconnectedness of the different Performance Measurement Systems.
2. Identify how possible actions provided by the KPIMS can affect each other.

And, most importantly:

3. Produce multi-objective Integrated Action Sets for the decision-makers.

As a result, the system would optimize the network implicitly towards the KPIs by deciding about the size and content of the Action Sets. In the further development, the system would also determine the optimal parameterization for the different, possible actions. The authors plan to use *Discrete-Event Simulation* (DES) to model the logistics network of the company and then use optimization algorithms to search for Integrated Action Sets by applying possible actions to the simulation model. DES was chosen, because it represents a powerful tool that enables to model the complexity and uncertainty of the real-world problem without the limiting assumptions of a pure mathematical problem description ([7], [8] and [9]).

The proposed approach can be described as a *SimHeuristic* approach. The concepts of SimHeuristics have been described by Juan and Rabe in [6]. It can also be described as a *Simulation-Optimization* (SimOpt) approach with the simulation as an evaluation function of the optimization algorithm, as described by März et al. in [10]. The German VDI defined this as "Category D" approach [11]. Because of the huge complexity, both in the modeling and computation effort, an increasing amount of attention in the research community has been observed [12]. März et al. give a good introduction to the field and present 12 interesting case studies in [10]. A good starting point to delve further into the SimHeuristics research literature is the chapter by Andradóttir [13] in the Handbook of Simulation [14]. A very interesting article by Fu [15] highlights the importance of further research in the field of "Optimization for Simulation", as he entitles it. Good reviews of simulation optimization approaches have been published by Fu [16] and Andradóttir [17]. The survey of Swisher et al. [18] provides further updated references. The review of Jacobson and Schruben [19] is a good starting point to track back the

advances in the field. Recently, Güller and Noche presented an interesting approach to determine the appropriate inventory control parameters of a multi-echelon supply chain using an object-oriented simulation model paired with a Multiobjective Particle Swarm Optimization algorithm [1]. Although recent approaches have shown the feasibility of a SimHeuristic Framework, none has addressed the influence of possible actions on complex Performance Measurement Systems. As Fu explicitly outlined in [15], there is additional need for research to further close the gap between mere research approaches and recent software developments in the field of simulation optimization. Also very little has been done on multi-response simulation optimization [15]. The authors expect to gain interesting, further insights in these directions. The goal is to model not only the logistics network of the company, but also the different Performance Measurement Systems. Different Action Sets will be tested on the simulation model to find the ones that optimize the network situation in a multi-objective way, i.e. enhancing multiple Key Performance Indicators at once.

4 SimHeuristic Framework Details

Figure 3 shows an approach for a framework architecture, which can perform the proposed interdependence analysis and generate Integrated Actions Sets. First, the data of the company, such as process data, stock data, and structure data are used to create a DES model of the logistics network. Because the process data and stock data are gathered by the operational systems and are regularly transferred into the Data Warehouse, they can be comfortably evaluated for the parameterization of the simulation model. Necessary structure data, such as for example the capacities of certain warehouses or transportation trucks, have been systematically gathered by the company in separate databases and files.

The authors plan to follow a procedure model with continuous verification and validation as described in [20] and [21] to ensure a high credibility of the simulation model and the results produced by the whole framework. The quality criteria for simulations in production and logistics have been described by Wenzel et al. in [22]. The book is used as a complementary guide. For the implementation of the executable model, different approaches are currently evaluated. Besides the options to use a commercial, specialized supply chain simulation package as e.g. SimChain [23], Enterprise Dynamics [24] or OTD-NET [25], also more code-oriented programming frameworks such as SimPy [26] or SSJ [27] are considered. For the described framework, execution

speed is more important than e.g. a graphical editor or visual animations. The faster an Action Set can be evaluated, the more Action Sets can be tested in a given amount of time. Besides this aspect, a tight integration of the metaheuristic with the simulation model can lead to significant reductions in the runtime and eventually lead to better quality results of the framework [15]. Currently, the simulation model is abstracted as a *Simulation Unit* in Figure 3 and illustrated as a standalone component. Eventually, the authors strive to have a tight integration of the driving *Heuristic Unit* and the *Simulation Unit*.

The automated update of the *Simulation Unit* with new data from the operational systems of the company could be a very interesting point for further research and is considered. Technically it should be possible to implement an automated *Update Unit* but currently this is out of scope and not further investigated.

In the second preparatory step, the KPI logic of the *Data Warehouse* is copied into a *Shadowed Data Warehouse* (SDWH). This includes the data model of the *Data Warehouse* and the exact queries for each KPI. The reports are still generated by the real *Data Warehouse*, but the KPI Alerts are sent to the *Heuristic Unit* instead of the responsible managers. The *Heuristic Unit* is planned to combine the possible actions for each KPI Alert into Action Sets and test them on the simulation model with the help of an *Execution Engine*. The *Execution Engine* should be able to read action descriptions in a predefined format

and apply them to the simulation model. Previous to the first execution, the initial state of the simulation model and the data in the SDWH will be saved as a backup to enable the *Heuristic Unit* to reset the simulation model and the SDWH to its initial state. Furthermore, action-specific snapshots (backups of the simulation model and the data in the SDWH) will be taken before each execution to enable the *Heuristic Unit* to revert the last action. This is especially important, because each execution will in fact translate into a series of simulation runs. A series of simulation runs will be necessary to receive statistically correct results [21]. As briefly mentioned in Section 2, a good categorization and exact specification of possible actions is of high importance as a requirement for the further development of the framework. The categorization of actions will be needed to build a helpful decision-tree for the metaheuristic, and the exact specification of the actions is of crucial importance for the *Execution Engine* to work. Furthermore, the specification of the actions will provide vital insights into which details of the logistics network need to be modeled and which parts can be abstracted. Once an action has been executed on the simulation model, it is necessary for the *Heuristic Unit* to evaluate the effects of the tested action. The simulation is executed and the traces of the simulation experiment (multiple simulation runs) are extracted, transformed and loaded into the SDWH using an abstracted form of the ETL process. For performance reasons it will be interesting to investigate into which abstraction layers in the SDWH the resulting data should be written.

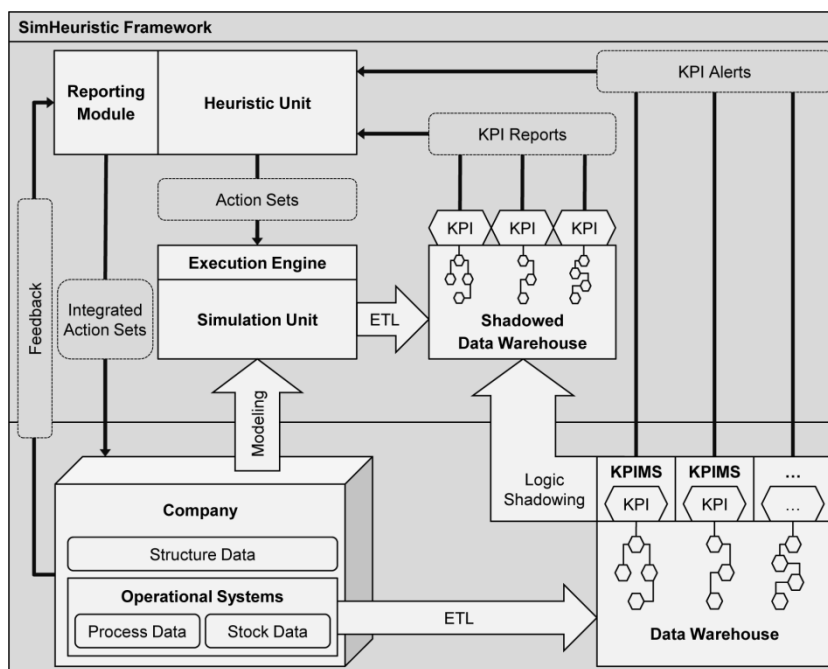


Figure 3. The architecture of the proposed SimHeuristic Framework

The Heuristic Unit receives the assessment of the executed action from the new resulting SDWH-KPIs. Once the action has been evaluated, the Heuristic Unit needs to decide whether another action should be executed on top of the resulting situation or whether the simulation model should be reset. The simulation model could be reset either to its initial state or to the previous state and a new action could be tested. Following this procedure, the Heuristic Unit would continuously explore some areas of the search space of possible action combinations. As one can suggest from the "Duration" attribute in Table 2, each action is defined with an estimated duration until the action is expected to show an effect on the network situation. This is especially important, because it will define the amount of periods which need to be simulated until the action can be evaluated. Another important aspect is to incorporate the consideration horizon of each Performance Measurement System. Some systems evaluate Performance Measures on the basis of a rolling 12 month horizon. Consequently, these aspects indicate the needed time for action evaluation. Because it would be computationally infeasible to evaluate all possible combinations of actions in an acceptable amount of time, a suitable heuristic has to be developed to evaluate only those action combinations that are promising to lead to an improvement of the overall logistics network situation regarding all KPIs under consideration.

Once the Heuristic Unit has found Integrated Action Sets that improve the network, the *Reporting Module* is planned to generate reports, which are sent to the responsible managers in the company. Because the actions for different receiver groups would be combined into one Integrated Action Set, it will be important to present the resulting, single actions with all interdependencies to the receiving managers. Otherwise, a thoughtless presentation could lead to a denial of the system. To enable the Heuristic Unit to learn from previously suggested Action Sets, a feedback mechanism is needed as illustrated in Figure 3. The feedback could be received manually from the executing managers or automatically from the surveillance of actually occurring changes in the real data in the next period. In this way, the framework could possibly offer some form of self-validation.

5 Conclusion and Outlook

Using the described setup, the authors aim to develop a SimHeuristic Framework that can be used as a Decision Support System for large logistics networks with complex KPIs. The paper presented a multidimensional problem in which parameterizable actions need to be combined into Integrated Action Sets. The

complexity of the problem is high, because it resembles a combination of a continuous and a discrete optimization problem [15]. Because the system is planned to be used as a Decision Support System, the authors assume that it is critical to assert a high credibility of the simulation model and the overall framework [21]. The next steps regarding the presented research are the ongoing review of further related literature on the subject and the development of a first prototype of the framework as briefly described in this paper. Subsequent, important decisions regarding different aspects of the framework need to be made. The adequate form of categorization and specification of the possible actions is particularly important, because it leads to the analysis of affected parts in the simulation model and the execution time of the evaluation runs. The authors laid the foundation for further research by describing the principles of the SimHeuristic Framework and producing a well discussed problem description as a foundation for proceeding research. Once the framework is setup, the concrete development of the Heuristic Unit will be part of the ongoing research. Beyond this, further research in automated model parameterization, parallel execution of evaluation runs (simulation experiments) and self-validation with subsequent data from the real world systems seem to be very interesting fields of research to improve the performance of the system.

6 References

- [1] M. Güller, B. Noche. *Development of a Simulation-based Optimisation Environment for a Capacitated Multi-Echelon Production-Inventory System*. Tagungsband der 15. ASIM Fachtagung: Simulation in Produktion und Logistik, in Paderborn, pp. 427–435, 2013.
- [2] A. Bauer, H. Günzel. *Data-Warehouse-Systeme: Architektur, Entwicklung, Anwendung*, 4th ed., dpunkt-Verl, Heidelberg, 2013.
- [3] C. Bruns, T. Hegmanns. *Performance Measurement System for a Changeable Stocking Strategy in Distribution Systems of Commercial Enterprises*. Proceedings of the 6th International Colloquium on Business & Management (ICBM), in Bangkok, Thailand, 2013.
- [4] M. Biesen, A. Tavakoli, T. Hegmanns. *Managing logistics performance of numerous facilities under consideration of facility-individual preconditions*. Proceedings of the 6th International Colloquium on Business & Management (ICBM), in Bangkok, Thailand, 2013.

- [5] L. Bianchi, M. Dorigo, L.M. Gambardella, Gutjahr, Walter J. W.J. *A survey on metaheuristics for stochastic combinatorial optimization*. Natural Computing: an International Journal 8, pp. 239–287, 2009.
- [6] A.A. Juan, M. Rabe. *Combining Simulation with Heuristics to solve Stochastic Routing and Scheduling Problems*. Tagungsband der 15. ASIM Fachtagung: Simulation in Produktion und Logistik, in Paderborn, pp. 641–649, 2013.
- [7] J. Banks. *Introduction to simulation*. Proceedings of the 2000 Winter Simulation Conference, Orlando, FL, USA, pp. 9–16, 2000.
- [8] J. Banks. *Discrete-event system simulation*. 5th ed., Prentice Hall, Upper Saddle River, N.J, Singapore, 2010.
- [9] S. Robinson. *Simulation: The practice of model development and use*. Wiley, Chichester, Eng, Hoboken, N.J, 2004.
- [10] L. März, W. Krug, O. Rose, G. Weiger. *Simulation und Optimierung in Produktion und Logistik: Praxisorientierter Leitfaden mit Fallbeispielen*. Springer, Heidelberg, 2011.
- [11] Verein Deutscher Ingenieure. *Richtlinie 3633 Blatt 12: Simulation und Modellierung - Simulation und Optimierung*. In preparation for green print, Beuth, Berlin, 2013.
- [12] F.D. Mele, G. Guillén, A. Espuña, L. Puigjaner. *A Simulation-Based Optimization Framework for Parameter Optimization of Supply-Chain Networks*. Ind. Eng. Chem. Res. 45, pp. 3133–3148, 2006.
- [13] S. Andradottir. *Simulation optimization*. In: Handbook of simulation. Principles, methodology, advances, applications, and practice, Editor: J. Banks, Wiley, New York, 1998.
- [14] J. Banks. *Handbook of simulation: Principles, methodology, advances, applications, and practice*. Wiley, New York, 1998.
- [15] M.C. Fu. *Feature Article - Optimization for simulation: Theory vs. Practice*. INFORMS Journal on Computing 14, pp. 192–215, 2002.
- [16] M.C. Fu. *Optimization via simulation: A review*. Ann. Oper. Res. 53, pp. 199–247, 1994.
- [17] S. Andradottir. *A review of simulation optimization techniques*. Proceedings of the 1998 Winter Simulation Conference, Washington, DC, USA, pp. 151–158, 1998.
- [18] J.R. Swisher, P.D. Hyden, S.H. Jacobson, L.W. Schruben. *Simulation Optimization: A Survey of Simulation Optimization Techniques and Procedures*. Proceedings of the 2000 Winter Simulation Conference, Orlando, FL, USA, pp. 119–128, 2000.
- [19] S.H. Jacobson, L.W. Schruben. *A review of techniques for simulation optimization*. Operations Research Letters, pp. 1–9, 1989.
- [20] M. Rabe, S. Spieckermann, S. Wenzel. *A new procedure model for verification and validation in production and logistics simulation*. Proceedings of the 2008 Winter Simulation Conference, Miami, FL, USA, pp. 1717–1726, 2008.
- [21] M. Rabe, S. Spieckermann, S. Wenzel. *Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken*. Springer-Verlag, Berlin, Heidelberg, 2008.
- [22] S. Wenzel, S. Collisi-Böhmer, H. Pitsch, O. Rose, M. Weiß. *Qualitätskriterien für die Simulation in Produktion und Logistik: Planung und Durchführung von Simulationsstudien*. Springer-Verlag, Berlin, Heidelberg, 2008.
- [23] K. Gutenschwager, K. Aliche. *Supply Chain Simulation mit ICON-SimChain*. In: Logistik Management, Editors: T. Spengler, S. Voß and H. Kopfer, Physica-Verlag, Heidelberg, pp. 161–178, 2004.
- [24] INCONTROL Enterprise Dynamics. <http://www.incontrolsim.com/>. Accessed on June 27, 2014.
- [25] OTD-NET. http://www.iml.fraunhofer.de/de/themengebiete/supply_chain_engineering/produkte/otd-net.html. Accessed on June 27, 2014.
- [26] SimPy. <http://simpy.sourceforge.net/old/>. Accessed on June 27, 2014.
- [27] SSJ: Stochastic Simulation in Java. <http://simul.iro.umontreal.ca/ssj/indexe.html>. Accessed on June 27, 2014.

Simulation-based Dispatching in Job Shops

Tao Zhang, Oliver Rose

Universität der Bundeswehr München

tao.zhang@unibw.de

The dispatching problem is a sequential decision problem. The early decisions determine the later decisions while the later decisions influence the performance of the early decisions. A branching tree with a time axis is introduced to present the problem. A node denotes a decision point. A branch represents the selection of one job from all waiting jobs. The corresponding value of nodes on the time axis is the time when the decisions need to be made. In the real case it is impossible to enumerate all paths in the tree and then evaluate them. A simulation-based approach is proposed. A four-tuple is adopted to illustrate our approach, $M = (E, A, S, V)$. E is the environment (i.e. situation) at the decision point. A is alternatives, i.e. jobs. S is the simulation. V is the evaluation. Once we need to make a dispatching decision, the simulation will be used to evaluate each alternative, i.e. each job waiting in front of the machine. The simulation model is initialized with the environment at the decision point before being started. For each job the simulation runs once (determined model) or several times (stochastic model). The selection of jobs is dependent on the evaluation on the basis of the simulation results.

1 Introduction

In job shops, when a machine becomes free and several jobs are waiting in front of it, we should decide which job will be processed first. This is a dispatching problem. After we have made the decision and dispatched one job on the machine, if this machine or other machines become free, the same type of decision needs to be made again. Obviously, during the manufacturing we have to make a huge number of such dispatching decisions. These decisions make up a sequential decision. Generally, the dispatching rules, such as FIFO, SPT and EDD, are a simple way to make such decisions. A dispatching rule is assigned to each machine or all machines. In this case, the simulation is used to evaluate different dispatching rules and the dispatching rule with the best performance will be chosen and used in practice [1, 2]. However, these dispatching rules are very rough methods and cannot adapt to the changing situation in the manufacturing line[3]. In addition, the sequential decision problem is mostly described as a Markov decision process (MDP)[4]. Yih, Y. and A. Thesen [5] formulated the scheduling problem as semi-Markov decision problems and used a non-intrusive ‘knowledge acquisition’ method to reduce the size of the state space. Gabel, T. and M. Riedmiller [6] modeled the job shop scheduling problem as an multi-agent reinforcement learning and attached to each resource an adaptive agent that makes its job dispatching decisions independently of the other agents and improves its dispatching behavior by trial and error employing a reinforcement learning algorithm. However, it is very difficult to find a transition func-

tion for the MDP model due to the complexity of the manufacturing line. The simplified transition function lost many key issues in the manufacturing line, such as setup, breakdowns, reworking, and so on. It is also difficult to find a reward function which is compatible with the scheduling objectives. In this paper, we propose a simulation-based approach. Once we need to make a dispatching decision, the simulation will be used to evaluate each alternative, i.e. each job waiting in front of the machine. For each job the simulation runs once (determined model) or several times (stochastic model). The selection of jobs is dependent on the evaluation on the basis of the simulation results.

The paper is structured as the following. In section 2 the dispatching problem is presented in a branching tree with a time axis. A four-tuple is adopted to illustrate our approach. And we will answer three key questions: 1) how many times do the simulations run; 2) how long do the simulations run; 3) how to select the base-rule; 4) How to evaluate the selection according to the simulation results. In Section 3 an approach used to evaluate our approach is addressed. Finally, an experiment is reported in section 4.

2 Simulation-based Dispatching

2.1 Branching tree with a time axis

The dispatching problem is a sequential decision problem. For sequential decisions, the early decision determines the later decision while the later decision influences the performance of the early decision[7]. It

can be presented in a branching tree with a time axis, shown in Figure 1.

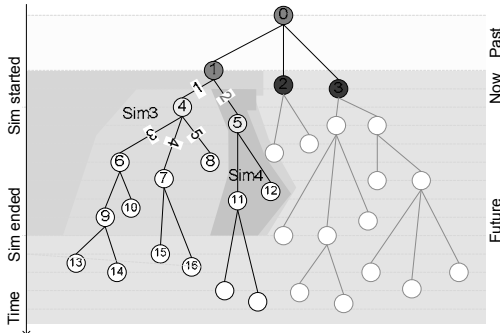


Figure 1. A branching tree with a time axis

A node denotes a decision point. A branch represents the selection of one job from all waiting jobs. The corresponding value of nodes on the time axis is the time when the decisions need to be made. A path from the root node to the last node is related to an active schedule. As time goes on, the tree will become bigger and bigger due to the exponential explosion. In the real case it is impossible to enumerate all paths and then evaluate them.

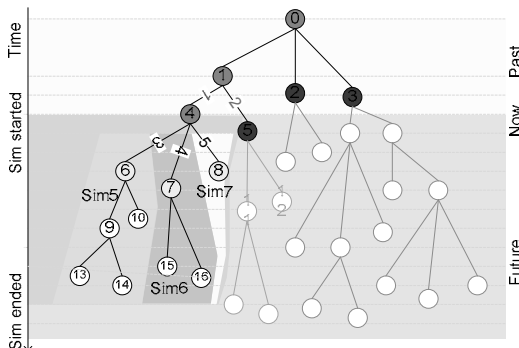


Figure 2. The branching tree after time advances

We use an example to give a general idea of our approach first. In figure 1 we assume that current time is at decision point 1 and we need to select one job from two. Thus, the simulation sim3 and sim4 will start respectively from the decision point and end at a given time. The sim3 will simulate the future situation while we select job 1. The sim4 deals with the selection of job 2. The results of the sim3 and sim4 will be used to evaluate the selection of job 1 and job 2. We can see that in both the sim3 and sim4, there are also lots of decision points. For these decision points we specify a base-rule to dispatch jobs. We assume that the selection of job 1 performs better, so we select job 1. When time advances to the decision

point 4, shown in Figure 2, we have to select one job from three. Thus the simulation sim5, sim6 and sim7 will start. The remaining work will be done in the same manner.

2.2 Four-tuple of our approach

Now we give a more detailed description on our approach. A four-tuple is adopted to illustrate our approach, $M = (E, A, S, V)$. E is the environment (i.e. situation) at the decision point. A is alternatives, i.e. jobs. S is the simulation. We call the simulation as alternative simulation. V is the evaluation. Before we start the simulations, the simulation model is initialized with the environment at the decision point. For each job, the simulation runs once or several times. The job will be selected according to the evaluation of the simulation results.

The environment information includes all online jobs' conditions and all machines' conditions. A machine's condition can be described by its state and the time that the state lasts. In addition to its state and the time that the state lasts, a job's condition includes its position and progress. The position describes which machine, buffer or transport line the job can be found on. The progress indicates which step the job reaches. In addition, scheduled events, e.g. starting maintenance on one machine, are also concerned. These events will be directly put into the event list of the simulation and will occur during the simulation run.

After the simulation initialization, in order to evaluate one alternative, a related "starting processing event" is put into the event list and the event occurring time is set to current time. Then the simulation starts. The simulations will forecast what will happen next. For the simulation and the evaluation, there are four key questions: 1) how many times do the simulations run; 2) how long do the simulations run; 3) how to select the base-rule; 4) How to evaluate the selection according to the simulation results.

Usually, if the simulation model is determined, the simulations just run once for each alternative. If the model is stochastic, the simulations need to run many times. The more the simulation runs, the better results we obtain. However, because the machines are free when we make the decision, we should make our decision as soon as possible in order to maximize the utilization. The number of times can be calculated according to the time allowance in the real system.

For the question 2), theoretically, the simulation ends when the effect of current decision disappears. But, it is hard to determine this time. Thus, the period of simulation is decided by the number of completed jobs n after the decision point.

$$n = m + WIP / 2 \quad (1)$$

where m is the number of completed jobs from the decision point to the time that jobs in the queue at the decision point are finished. WIP is the work in process level at the decision point.

The selection of the base-rule is dependent on the objectives of the dispatching problem. An optimal rule for a single machine problem with the same objective usually is set to a base-rule. For example, if the objective is to minimize the cycle time, the base-rule is Shortest Processing Time (SPT); if the objective is to minimize the makespan, the base-rule uses Longest Processing Time (LPT)[8].

The evaluation is also related to the dispatching objective. A priority value is calculated for each job according to its simulation results. The job with the highest priority will be selected. Here, we give two formulas for two different objectives. If the objective is to minimize the cycle time,

$$prio = 1 / \sum_{p \in P} (\omega_p \sum_{j \in J_p} c_j / n_{J_p}) \quad (2)$$

where p denotes a product; P is the set of products.

ω_p is the weight of product p . The weight of a production is the ratio of the product's throughput to the

total throughput, which meets $\sum_{p \in P} \omega_p = 1$. j is a job.

J_p is the set of completed jobs whose product type is p .

c_j is job j 's cycle time. n_{J_p} is the number of jobs in the set J_p .

If the objective is to minimize total weighted tardiness

$$prio = 1 / \sum_{p \in P} (\omega_p \sum_{j \in J_p} \max(C_j - d_j, 0) / n_{J_p}) \quad (3)$$

where C_j is job j 's completion time and d_j is job j 's due date. If the simulation runs many times, the average value of the priorities is adopted.

3 Requirements on the Simulation

In this paper, we mainly focus on the application of the simulation rather than the simulation modeling. Because the performance of our approach is deeply dependent on the simulation model, we list some requirements on the simulator and the simulation model here.

In the simulation model, the processing time can be deterministic or stochastic. Machines need setup and break down randomly. The repairing time is either deterministic or stochastic. The preventive maintenance also needs to be considered. Some common dispatching rules and release policies should be included. The model must provide the interface to initialize itself and set the base rule. For the simulator, it must be flexible enough to load different job shop models and enable us to operate the event list. It has to be able to monitor the WIP level of the model. The simulator can run many times under the same initialization and we can specify the end condition for the simulator. The simulator also needs to output the results in a certain format for evaluating. The more details of the simulation can be found in the paper [9] that we presented in the conference WSC2012.

4 Approach to Evaluate our Approach

Before each alternative simulation run starts, the simulation model should be initialized according to the environment information of the real system at the decision point. However, it is impossible and impractical to connect to the real system and read the environment information now. Therefore, in order to evaluate the simulation-based dispatching approach, we replace the real system with a simulation model. The simulation is always running like a real system. We call the simulation as environment simulation. Once a decision needs to be made, the environment simulation will pause. The decision-making procedure will start. The alternative simulation is initialized according to the current state of the environment simulation. When the decision-making is done, the environment simulation will take the decision and continue. The performance of the simulation-based dispatching approach can be calculated according to the results of the environment simulation.

5 Experiment

The proposed approach is used in a manufacturing system. The system contains 6 machine groups, 24

machines and produces 4 products (Pa, Pb, Pc and Pd) with 4 process flows and 4 different outputs. There are no batch processing machines. Interval times of releasing jobs follow the normal distribution. Both sequence dependent and independent setups are needed. The interval between two breakdowns is subject to the exponential distribution and the repairing time follows an exponential distribution, too. The objective of dispatching is to minimize the cycle time.

We compare our approach with the dispatching rules including first in first out (FIFO), shortest processing time (SPT) and SPT+ by the simulation. SPT+ is an improved version of SPT. In the SPT+, we set a maximal waiting time. If a job's waiting time is longer than the maximal waiting time, the job will be dispatched first so as to avoid that a job with a long processing time has to wait too long. The results are shown in Table 1.

Rule	Pa	Pb	Pc	Pd	Sum.
FIFO	8.78	8.28	9.51	8.80	8.85
SPT	24.34	6.09	12.94	5.02	13.80
SPT+	10.25	9.83	9.07	7.76	9.49
Sim.	6.55	5.68	7.87	5.25	6.50

Table 1. Average cycle time of jobs grouped by product type

SPT is an optimal rule for minimizing the cycle time in the single machine problem. In this case, we can see that SPT performs worse than any other approaches for products Pa and Pc. The reason is that Pa and Pc have a longer raw processing time than Pb and Pd. When we use the SPT+ rule, the cycle times of products Pa and Pc go down, but go up for products Pb and Pd. The simulation-based approach is the best for products Pa, Pb and Pc and the second best for the product Pd. Overall, the simulation-based approach performs better than others.

6 Conclusions

In the general simulation-based dispatching, the simulation is used to evaluate the dispatching rules offline. The simulation runs many times for each dispatching rule, then the best rule is selected. In our approach, the simulation is used to make decisions online. While a decision needs to be made, an alternative simulation starts to run for each alternative, i.e. each job. The best job will be selected according to

the results from the alternative simulations. During a simulation run, a base rule is used to make decisions. All simulation runs use the same base-rule, so it is fair to judge alternatives by the simulation. From the experiment results we can see that our approach performs better than any other dispatching rule. The better performance is attributed to the forecast function of the simulation. The other dispatching rules only pay attention on current situations without consideration of the future.

7 References

- [1] A. Hübl, H. Jodlbauer, and K. Altendorfer, *Influence of dispatching rules on average production lead time for multi-stage production systems*. International Journal of Production Economics, p. 479-484, 2013.
- [2] C. Rajendran and O. Holthaus, *A comparative study of dispatching rules in dynamic flowshops and jobshops*. European Journal of Operational Research, p. 156-170, 1999.
- [3] T. Zhang and O. Rose, *Intelligent dispatching in dynamic stochastic job shops*. Winter Simulation Conference (2013) in Washington D.C., p. 2622-2632, 2013.
- [4] M. L. Littman, *Algorithms for sequential decision making*, Dissertation on, Brown University, 1996.
- [5] Y. Yih and A. Thesen, *Semi-Markov decision models for real-time scheduling*. International Journal of Production Research, p. 2331-2346, 1991.
- [6] T. Gabel and M. Riedmiller, *Adaptive Reactive Job-Shop Scheduling with Learning Agents*. International Journal of Information Technology and Intelligent Computing in, p., 2007.
- [7] M. L. Littman, *Algorithms for Sequential Decision Making*, Dissertation on, Brown University, Rhode Island, 1996.
- [8] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, Springer Publishing Company, New York, 2008
- [9] T. Zhang and O. Rose, *Developing an agent-oriented parallel simulator for production processes*, in *Proceedings of the Winter Simulation Conference 2012*, Winter Simulation Conference: Berlin, Germany. p. 1-2.

Anforderungen und Einsatzmöglichkeiten der Fluidsimulation in der frühen Phase des Produktentwicklungsprozesses

Stefan Krottil¹, Peter Stich¹, Gunther Reinhart¹

¹Projektgruppe RMV des Fraunhofer IWU

Stefan.Krottil@iwu.fraunhofer.de

Die Simulation hat sich im Produktentwicklungsprozess als hilfreiches und valides Hilfsmittel etabliert. Zur prinzipiellen Untersuchung von Alternativen und zur Überprüfung von Funktionalitäten eignen sich gerade in der frühen Phase des Produktentwicklungsprozesses Simulationen zur Vermeidung von Fehlentwicklungen sowie kostenintensiver Prototypen. Jedoch ist eine einfache Konzeptüberprüfung fluiddynamischer Prozesse auf Basis der aktuell eingesetzten Simulationsmethoden nur mit hohem Zeitaufwand möglich. Alternative Simulationsmethoden können Abhilfe schaffen. Zum Vergleich dieser werden daher zuerst die Anforderungen in der frühen Phase des Entwicklungsprozess an die Simulation aufgeführt. Auf dieser Basis wird ein multikriterieller Vergleich der Euler- und Lagrangeschen Berechnungsmethoden durchgeführt. Anschließend wird eine neue Möglichkeit zum Einsatz der Fluidsimulation in den frühen Phasen betrachtet. Durch eine zeitgleich durchgeführten Simulation und der daraus resultierenden direkten Wissensgenerierung kann der Entwicklungsprozess unterstützt werden. Darüber hinaus wird eine Methode zur Wiederverwendung relevanter Simulationsmerkmale aus der interaktiven Simulation zur Güteverbesserung bisheriger Simulationsmethoden dargestellt.

1 Produktentwicklungsprozess

Kürzere Produktlebenszyklen, steigende Variantenvielfalt sowie steigende Mechatronisierung von produktionstechnischen Anlagen lässt den Produktentwicklungsprozess stärker in den Fokus rücken. Erhöhte Variabilität und Dynamik sowie verbesserte Unterstützung durch mechatronikorientierte Entwicklungswerkzeuge sind eine Möglichkeit, diese Herausforderungen entgegen zu treten. Darüber hinaus müssen Risiken in der frühen Phase des Entwicklungsprozess möglichst früh erkannt und minimiert werden, um den späteren wirtschaftlichen Einfluss von Fehlentwicklungen gering zu halten. [1]

1.1 Simulation im Produktentwicklungsprozess

Aus diesem Grund sind gute Entscheidungen in den frühen Phasen des Produktentwicklungsprozesses von essentieller Bedeutung. Zur Validierung unterschiedlicher Konzepte und Entwürfe hat sich die Simulation als effektives Hilfsmittel etabliert. So gaben 60 % aller Unternehmen an, dass Simulations- und Visualisierungstechnologien die Qualität von Produkten verbessern und 52 % nennen eine Beschleunigung des Produktentwicklungsprozess als Vorteil. [2]

Der prozessuale Ablauf ist in heutigen Entwicklungsprozessen noch durch eine iterative Abfolge von Konstruktions- und Simulationsphasen geprägt (vgl. Abbildung 1) [3]. Ein Iterationsprozess aus Sicht der

Simulation gliedert sich in grob in drei Phasen: Die erste Phase ist die Simulationsvorbereitung, in welcher beispielsweise digitale Modelle vernetzt und beispielsweise die Randbedingungen für die Simulation aufbereitet werden. Daraufhin erfolgt der eigentliche Simulationsablauf. Die Resultate werden abschließend zur Auswertung an die Ingenieure zurückgegeben. [4]

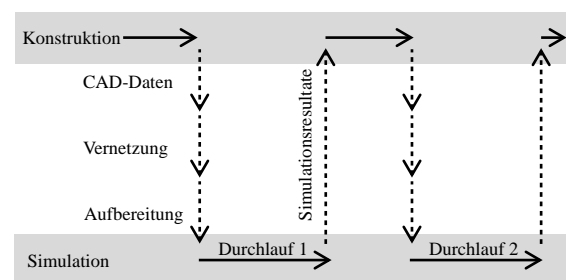


Abbildung 1: Ablauf des heutigen Simulationseinsatzes im Entwicklungsprozess

1.2 Fluidische Effekte in der Simulation

Zur Maximierung des Nutzens der Simulation sind eine möglichst einfache Erstellung der Simulationsmodelle und eine schnelle Durchführung der Simulationsstudien notwendig. Thermisches Verhalten oder fluidische Effekte, wie beispielsweise Abfüllprozesse, werden heute in vielen Produktentwicklungen meist nicht simulativ oder erst sehr spät nach der grundlegenden

genden Konzeptionsphase validiert. Dies resultiert aus mehreren Gründen [5]:

- Die Durchführung einer Simulation ist auf Grund des Berechnungsaufwandes sehr zeitintensiv.
- Die Erstellung der Simulationsmodelle und die Bestimmung der komplexen Randbedingungen sind in Fluidsimulationen aufwändig.
- Die Wiederverwendbarkeit der einzelnen Modelle ist stark eingeschränkt. Nach jedem Entwicklungszyklus müssen meist komplett neue Modelle erstellt werden.
- Es existiert kein Simulationsverfahren für fluidische Problemstellungen, welches eine einfache Nutzung erlaubt. Je nach Problemfall müssen für präzise Ergebnisse die korrekten Simulationsmethoden und -parameter ausgewählt werden. Aus diesem Grund muss auch die Ergebnisinterpretation von Experten durchgeführt werden.
- Die Kosten für kommerzielle Software sind sehr hoch.

Zusammenfassend bedeutet dies, dass Fluide meist nur in speziellen Fällen simuliert werden. Beispielanwendungen sind aerodynamische Untersuchungen von Autos oder Flugzeugen sowie Triebwerken. Industrielle Prozesse wie beispielsweise Abfüllprozesse, werden jedoch oft noch durch heuristische Versuche validiert und umgesetzt.

2 Computational Fluid Dynamics

Die Grundlage zur Simulation für fluidische Problemstellungen ist die Navier-Stokes Gleichung. Auf Grund der Komplexität sind analytische Lösungen nur in einfachen Ausnahmefällen möglich. Daher wurden unterschiedliche numerische Lösungsmethoden entwickelt, welche je nach Anwendungsgebiet unterschiedliche Stärken und Schwächen aufweisen. Grundsätzlich können die Verfahren in zwei Klassen eingeteilt werden: Eulersche- und Lagrangesche-Methoden. Auch Mischformen der beiden Methoden wurden zur Kompensation der individuellen Nachteile entwickelt. Diese werden in dieser Publikation jedoch nicht tiefer behandelt.

2.1 Eulersche-Methoden

In der Industrie werden heute meist gut validierte Algorithmen eingesetzt, welche auf einer vorab berechneten Vernetzung der Fluiddomäne basieren. Auf Grund des geringeren Rechenaufwandes findet die Vereinfachung mit der Reynolds-Averaged-Navier-Stokes (RANS) Methode die größte Verbreitung [5]. Darüber hinaus werden in Fragestellungen mit der Notwendigkeit einer sehr hohen Ergebnisgenauigkeit vornehmlich im wissenschaftlichen Umfeld die Large-Eddy-Simulation (LES) [6] und die Direkte Numerische Simulation (DNS) eingesetzt (vgl. Abbildung 2) [7].

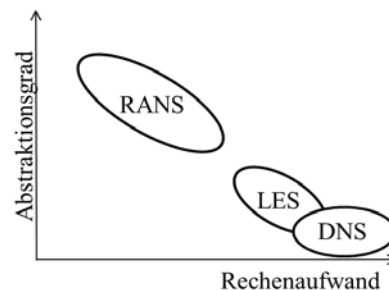


Abbildung 2: Vergleich heutiger CFD-Verfahren

Den Lösungsmethoden ist wegen der inhärenten Grundstruktur ein Zusammenhang gemein: die Genauigkeit der Resultate sowie der Berechnungsaufwand und somit die Resultatbereitstellung hängt maßgeblich von der Güte und Granularität der Vernetzung der Fluiddomäne ab. Diese wird weitestgehend automatisiert generiert. Lokale Optimierungsmaßnahmen der Diskretisierung erfolgen oftmals auf Basis von Erfahrungswissen durch den Nutzer. Auf dieser Grundlage werden der räumliche Abstraktionsgrad und somit auch der Fehler bestimmt. [8]

Um die auftretenden Fehler der Berechnungsverfahren zu minimieren werden oft adaptive Verfahren eingesetzt. Diese zeichnen sich durch eine situationspezifische, dynamische Anpassung der Vernetzung während des eigentlichen Simulationsablaufes aus. [9]

Jedoch können trotz dieser und weiterer Verbesserungen der Simulationen nach den Eulerschen-Methoden nicht alle Problemfälle simuliert werden. Auf Grund ungelöster Probleme muss somit der Nutzer weiterhin Expertise zur Abschätzung der Resultate nutzen [10] [5]. Beispielsweise stellen Turbulenzen auf Grund ihres scheinbar chaotischen Verhaltens trotz einer

sehr langen Forschungshistorie noch immer große Probleme dar [11].

2.2 Lagrangesche -Methoden

Zur besseren Modellierung, beispielsweise von turbulenten Effekten, können die kaum industriell genutzten Lagrangesche-Methoden verwendet werden. Im Gegensatz zu Eulersche-Methoden wird die Simulationsdomäne nicht auf Basis eines Netzes definiert. Stattdessen dienen Partikel als Diskretisierungsbasis. Dadurch können einige methodeninhärente Schwierigkeiten der eulerschen Methoden umgangen werden, welche auf der Notwendigkeit einer Vernetzung basieren. Als Beispiele sind hier die zeit- und berechnungsaufwändige Netzgenerierung, das adaptive Regenerieren der Netzqualität sowie Modellierungsprobleme bei Mehrphasenströmungen und sich bewegende Randbedingungen sowie große Deformationen zu nennen. [12][13] Darüber hinaus können Lagrangesche-Methoden auf Grund guter Skalierbarkeit, großer Zeitschritttoleranz sowie der Möglichkeit einer starken Parallelisierung echtzeitfähig simuliert werden [14][15]. Zur Lösung von fluidischen Problemstellungen gibt es zahlreiche Methoden. Als Beispiel ist die Smoothed Particle Hydrodynamics Methode (SPH) zu nennen, welche auf Grund der bereits stark fortgeschrittenen Forschung für industrielle Einsätze geeignet ist. [16]

3 Konzeptionsphasenspezifische Anforderungen an die Simulation

Heute eingesetzte Simulationsmethoden für Fluide sind zur Bereitstellung präziser Resultate ausgelegt. Diese Ausrichtung wird jedoch nicht den Anforderungen der frühen Phase gerecht. Daher werden im Folgenden vom Stand der Wissenschaft und Technik abgeleitete, allgemeine und fluidsimulationsspezifische Anforderungen definiert.

3.1 Anforderungen in der frühen Phase

Auf wirtschaftswissenschaftlicher Sicht ist vor allem die frühe Phase des Entwicklungsprozesses, das heißt die Konzeptionsphase, von großer Relevanz: Grundsätzliche Entscheidungen können kostengünstig revidiert und unterschiedliche Konzept- und Wirkprinzipialalternativen validiert werden. Je später sich ein Produkt im Entwicklungsprozess befindet, desto größer sind die Kosten von Änderungen. Aus diesem Grund ist es notwendig, möglichst früh im Entwicklungsprozess unterschiedliche Alternativen zu entwickeln und zu Testen. Zur Vermeidung von kostenin-

tensiven realen Prototypen eignen sich Simulationen, welche in der Konzeptionsphase eine schnelle Bewertung hinsichtlich der Zielgrößen ermöglichen. Die Genauigkeit der Simulationsergebnisse spielt zu diesem Zeitpunkt des Entwicklungsprozess eine untergeordnete Rolle. Eine größere Bedeutung hat die prinzipielle Validierung der Funktionen und Alternativen. Die Gewichtung zwischen der Simulationengenauigkeit und -geschwindigkeit verändert sich jedoch bis hin zu Detailstudien immer mehr. (vgl. Abbildung 3) [16]

Diese Spezifika der Konzeptionsphase resultieren in speziellen Anforderungen an die Simulationen.

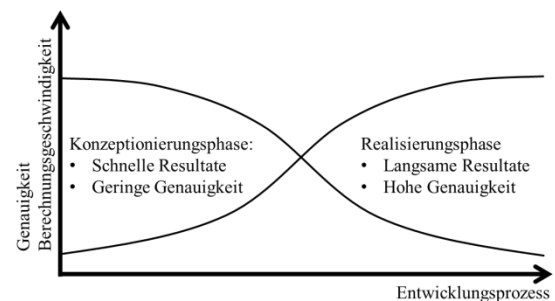


Abbildung 3: Phasenspezifische Anforderungen an die Simulation

3.2 Anforderung an die Simulation in der frühen Phase

Die frühe Phase des Entwicklungsprozess zeichnet sich somit durch folgendes Merkmal aus: Simulationsgeschwindigkeit geht vor Simulationspräzision. Auf Basis dieses Leitsatzes ergeben sich für die Fluidsimulation phasenspezifische Anforderungen.

Wie in Kapitel 1.1 erläutert, besteht ein Simulationsdurchlauf aus den drei Phasen der Vorbereitung, Durchführung und Ergebnisbereitstellung.

Aus diesen Zusammenhängen ergeben sich somit folgende individuelle phasenspezifische Hauptanforderungen:

- **Geringer Modellierungsaufwand:** Eine einfache und schnelle Modellierung der zu untersuchenden Simulationsszenarien, welche die Präzisionsanforderungen der Entwicklungsphase entsprechen.
- **Hohe Berechnungsgeschwindigkeit:** Eine schnelle Berechnung der Simulation.
- **Schnelle Ergebnisbereitstellung:** Eine schnelle Bereitstellung der Ergebnisse in einer für den Anwender aufbereiteten Form.

Neben einer kurzen Zyklusdauer einer Simulationsdurchführung ist es für die Anwender jedoch auch notwendig, möglichst große Freiheiten bei der Modellerstellung zu haben. Im Wesentlichen ergibt sich daher folgende weitere Hauptanforderung:

- **Hohe Modellfreiheit:** Ein hoher Freiheitsgrad in der Erstellung von Simulationsmodellen, beispielsweise in der geometrischen Dimensionierung der Abmessungen sowie eine einfache Modellierung komplexer Randbedingungen.

Darüber hinaus ergeben sich noch Nebenanforderungen, welche nur indirekt aus den Anforderungen des Produktentwicklungsprozess resultieren. Stattdessen unterstützen diese die bei der Auswahl der Berechnungsmethode für die grundlegenden Zielstellungen der schnellen Resultatbereitstellung und Ergebnispräsentation:

- **Stabilität:** Die Simulationsmethode soll auch bei grober temporärer und geometrischer Diskretisierung unbedingt stabil sein.
- **Parallelisierbarkeit:** Zur Sicherstellung einer schnellen Ergebnisbereitstellung spielt in den komplexen und berechnungsintensiven Algorithmen die Parallelisierbarkeit eine zunehmend große Rolle. Nimmt man die Entwicklung der letzten Jahre in den Betrachtungsfokus, so fällt auf, dass das Mooresche Gesetz weiterhin eingehalten wird. Jedoch basiert diese Entwicklung in den letzten Jahren auf der zunehmenden Parallelisierung mehrerer Recheneinheiten. Die Taktfrequenzen der Recheneinheiten bleiben jedoch weitestgehend konstant.
- **Berechnungseffizienz:** Eine weitere Anforderung ist eine hohe Berechnungseffizienz des genutzten Algorithmus. Berechnungseffizienz ist hier definiert als das inverse der Anzahl der notwendigen mathematischen Operationen je Volumeneinheit und Zeitschritt. Je kleiner die Berechnungseffizienz ist, desto weniger Berechnungsschritte sind notwendig und desto schneller können Resultate eines Problemfalles bei gleicher Hardwarekonfiguration des Simulationsrechners bereitgestellt werden.
- **Konvergenzgeschwindigkeit:** Eine hohe Konvergenzgeschwindigkeit ist in den nu-

merischen Operationen zur schnellen Lösung der komplexen Gleichungen notwendig.

- **Auflösungsadaptivität:** Durch eine adaptive Verfeinerung der geometrischen Diskretisierung kann die Fluiddomäne einfach diskretisiert werden. Durch die Adaptierbarkeit können jedoch valide Simulationsergebnisse bei geringerer gesamter Berechnungskapazität sichergestellt werden, da je nach Bedarf die geometrische Auflösung an die Anforderungen angepasst werden können.
- **Methodenvalidität:** Auf Grund der hohen Komplexität der Navier-Stokes Gleichungen ist eine Gültigkeit von Berechnungsergebnissen nicht sichergestellt. Aus diesem Grund sind validierte Algorithmen als verlässlicher anzusehen.

4 Auswahl der Simulationsmethode für frühe Entwicklungsphasen

Zur Auswahl eines geeigneten Simulationsprinzips für fluidische Effekte muss der Stand der Wissenschaft und Technik (vgl. Kapitel 2) mit den Anforderungen abgeglichen werden. Dieser qualitative Vergleich ist in Tabelle 1 dargestellt.

Auf Grund der fehlenden Notwendigkeit von vorab definierten Vernetzungen liegen Vorteile gerade in der Modellierungsgeschwindigkeit sowie der möglichen Modellierungskomplexität bei den lagrangeschen Methoden. Darüber hinaus können durch geeignete Algorithmik lagrangesche Partikel unabhängig voneinander berechnet werden. Eine schwächere Berechnungseffizienz sowie eine niedrigere Konvergenzgeschwindigkeit können somit durch eine gesteigerte Parallelisierbarkeit sowie einer größeren Toleranz in den Zeitschrittweiten zu einer schnelleren Ergebnisbereitstellung bis hin zur Echtzeitfähigkeit dienen. Eine Abhängigkeit zwischen der Ergebnispräsentation und Berechnungsgeschwindigkeit ist in der Bewertung jedoch stets zu beachten.

Anforderung	Euler	Lagrange
Modellierungsgeschwindigkeit	-	+
Berechnungsgeschwindigkeit	-	+
Ergebnisbereitstellung	O	O
Modellierungskomplexität	-	+
Stabilität	O	+
Parallelisierbarkeit	-	+
Berechnungseffizienz	+	O
Konvergenzgeschwindigkeit	+	O
Auflösungsadaptivität	O	+
Methodenvalidität	+	-

Tabelle 1: Vergleich der Modellierungsprinzipien für die Fluidsimulation in den frühen Phasen des Produktentwicklungsprozess

Zusammenfassend lässt sich jedoch sagen, dass in der Konzeptionsphase lagrangesche Methoden auf Grund ihrer einfachen Modellierung und relativ einfachen Algorithmik im Vorteil sind, wohingegen eulersche Methoden auf Grund der langen und intensiven Forschungsausrichtung valide Algorithmen sind, deren Fokus auf einer hohen Ergebnispräzision liegt. Den Anforderungen der frühen Phasen des Entwicklungsprozess können somit lagrangesche Methoden besser gerecht werden.

5 Einsatzpotenziale der Fluidsimulation in frühen Phasen

Die dynamischere Modellerstellung und Simulation im Zuge einer interaktiven Konzeptionierung kann mit Hilfe der lagranschen Methoden zur Beschleunigung der Konzeptvalidierungen beitragen. Eine Möglichkeit zur Unterstützung der frühen Entwicklungsphasen ist die interaktive Konzeption, das heißt, dass Änderungen am Modell in einer zeitgleich durchgeführten Simulationen übertragen und deren Resultate sofort am digitalen Modell einsehbar sind.

In heutigen Entwicklungsprozessen werden in iterativer Abfolge Konzepte erstellt und simulativ abgesichert (vgl. Kapitel 1.1). Nutzt man jedoch die Möglichkeit der Echtzeitfähigkeit und die einfachen Mo-

dellierungsmöglichkeiten von lagrangeschen Methoden, können Konstruktions- und Modellmodifikationen direkt am virtuellen Modell noch während der Konstruktionsphase dargestellt und simuliert werden. Wie in Abbildung 4 zu sehen ist, wird aus dem heutigen iterativ aufeinanderfolgenden Konstruktions- und Simulationszyklus ein gleichzeitiges interaktives Konstruieren. Die drei Phasen Simulationsvorbereitung, -durchführung und Ergebnisbereitstellung laufen parallel ab.

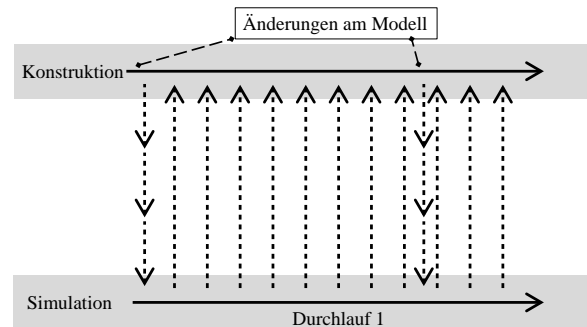


Abbildung 4: Simulationseinsatz in der interaktiven Konstruktion

5.1 Simulationsvorbereitung

Der Prozess der Simulationsvorbereitung ist immer dann notwendig, falls das Simulationsmodell sich ändert. Beispielhafte Auslöser sind Änderungen der Randbedingungen oder Modifikation am simulierten Objekt. Auf Basis der Nutzung von lagrangeschen Simulationsmethoden kann die Phase jedoch stark vereinfacht werden, wodurch eine Unterbrechung der nicht notwendig ist. Diese Aufwandsreduktion ist maßgeblich auf den nicht notwendigen Prozessschritt der Vernetzung möglich. Stattdessen werden Starrkörper, welche in der Fluidomäne liegen, als Dirichlet Randbedingung definiert. Darüber hinaus führen Modifikationen der Starrkörper, also Änderungen des Konstrukteurs am Produktkonzept, nicht zu einem vollständigen Neuaufbau der Simulationsdomäne und deren Randbedingungen. Stattdessen werden durch einen berechnungssparsamen Vergleich der Eingangsmodelle die Versionsunterschiede erkannt und das Simulationsmodell nur auf Basis der Unterschiede modifiziert.

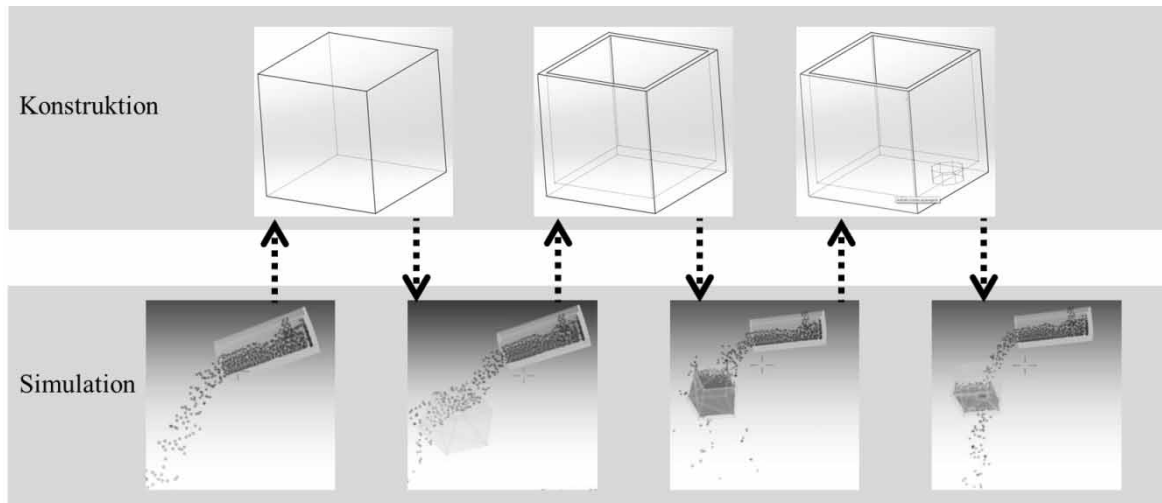


Abbildung 5: Umsetzungsbeispiel der interaktiven Konzeption – Aufbau eines Sammelbeckens. Dabei wird jeder Entwicklungsschritt sofort in eine Simulation übertragen. Jede Änderung ist ohne Verzögerung in der Simulation erkennbar.

5.2 Simulationsdurchführung

Die Simulation wird permanent und ohne Unterbrechung im Hintergrund durchgeführt. Änderungen am Modell werden online durchgeführt. In der Simulationsdurchführung kann es dadurch jedoch zu Schwankungen in der Berechnungszeit eines Zeitschrittes kommen, wodurch eine harte Echtzeitfähigkeit nicht sichergestellt werden kann. Somit wäre auch eine visuelle Abschätzung der Entwicklungsschritte auf Basis der Simulationsergebnisse nicht mehr möglich.

Aus diesem Grund ist zur Sicherstellung eines bestimmten Verhältnisses zwischen Realzeit und Simulationszeit eine Dynamisierung der Zeitschrittweite der Berechnungen notwendig. Basis des Problems ist, dass die Berechnungsdauer eines Zeitschrittes nicht konstant ist. Dies resultiert zum einen von anderen Prozessen, welche auf dem Computer aktiv sind, zum anderen aber auch auf der Variabilität der Modellgröße oder der Berechnungskomplexität des individuellen Zeitschrittes. Wie in Abbildung 5 dargestellt, würde eine konstante Simulationszeitschrittweite zu großen Asynchronitäten zwischen Real- und Simulationszeit führen. Aus diesem Grund werden die Zeitschrittweiten an die Dauer der vorangegangenen Berechnungsschritte dynamisch angepasst. Dadurch kann über mehrere Zeitschritte die weiche Echtzeitfähigkeit sichergestellt werden.

Eine hohe Priorisierung der Synchronisierungsstrategie der unterschiedlichen Zeitschrittweiten resultiert jedoch bei großen Diskrepanzen in einer großen Fluktuation der Ergebnispräzision. Aus diesem Grund

kann eine langsamere Anpassung der Zeitsynchronität von Vorteil sein. Dies ist beispielsweise der Fall, falls nur Teilprozesse validiert werden und somit eine globale Verzögerung keine Relevanz auf die Simulationsergebnisse hat.

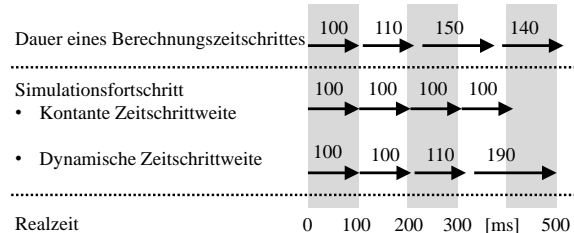


Abbildung 6: Synchronitätsabsicherung zwischen Real- und Simulationszeit

5.3 Ergebnisbereitstellung

Die Visualisierung der Auswirkungen von Konzeptänderungen auf die Prozesse kann auf Grund der Echtzeitfähigkeit direkt und ohne Verzögerungen an den Nutzer ausgegeben werden. Dadurch ist eine Interaktivität der Konzeption gegeben – Änderungen haben sofortige Auswirkungen auf aktuelle Simulationenläufe.

6 Umsetzungsbeispiel

Eine erste Umsetzung der interaktiven Konzeption wurde anhand eines Abfüllprozesses validiert. Während des Entwicklungsprozesses konnte in Echtzeit die Simulation durchgeführt werden. Als Simulationemethode wurde auf Grund seiner relativ großen Verbreitung sowie der Verfügbarkeit von validierten

Algorithmen die SPH-Methode eingesetzt. Abbildung 6 zeigt einen zeitlichen Verlauf der Simulation sowie die Entwicklung während des Entwicklungsprozesses auf. Dabei werden in Echtzeit die Modelländerungen aus der Entwicklungssoftware an die Simulationsumgebung übergeben. Die Simulation des fluidischen Prozesses muss nicht neu gestartet werden. Somit wird die Simulation permanent und ohne Unterbrechungen im Konzeptionsprozess durchgeführt. Darüber hinaus passt sich das Produktdesign nach Änderungen ohne Verzögerungen an. Somit ist eine sofortige visuelle Auswertung und Plausibilitätsüberprüfung möglich.

7 Wissensbasierte Vernetzung

Neben den vorgestellten Vorteilen der lagrangeschen Methoden und der Möglichkeit einer interaktiven Konzeption gewinnt in späteren Phasen des Entwicklungsprozess die Präzision der Resultate gegenüber der Ergebnisbereitstellungszeit an Bedeutung. Daher ist in diesen Entwicklungsstadien der Einsatz der klassischen, gut validierten eulerschen Methoden von Vorteil. Jedoch können die vorhandenen Simulationsresultate zur Reduktion des Aufwandes der netzbasierten Modellbildung genutzt werden.

Heute erfolgt die Vernetzung, welche in herkömmlichen CFD-Verfahren notwendig ist, meist nach Erfahrung des jeweiligen Simulationsexperten. Kritische Gebiete, beispielsweise Wirbelzentren, müssen durch das Erfahrungswissen manuell angepasst werden. Alternativ eignen sich hierfür adaptive Netzverfeinerungen. Als eine Folge dessen kann eine gleichbleibende Qualität nicht sichergestellt werden. Darüber hinaus kann auch auf Grund der nur wenigen neutralen Qualitätskennzahlen wie beispielsweise die Diversität der Zellgrößen oder Minimale/Maximale Winkel von Vernetzungen keine allgemeingültige Aussage über die Qualität von Erstvernetzungen gegeben werden. Die Netzqualität steigt meist erst durch wiederholtes Simulieren und darauf folgendes (adaptiven) Anpassung der Vernetzung. [8][9]

Die Resultate der lagrangeschen Simulation können durch schnelle Berechnungen Unterstützungsdienste leisten. Auf Basis der Simulationsergebnisse werden je nach Anwendungsfall feiner zu vernetzende Regionen ausgewählt. Die Auswahl kann dabei aus unterschiedlichen Zusammenhängen zusammengesetzt sein (z. B. Verwirbelung, Druckabfall, Geschwindigkeitsdivergenzen). Auf dieser Basis wird ein merkmalspezifisches Netz generiert.

8 Ausblick

Durch die Integration der Fluidsimulation in den Konstruktionsprozess können Alternativen schnell und einfach miteinander verglichen werden, wodurch das wirtschaftliche Risiko von Entwicklungen reduziert werden kann. In Zukunft wird die automatisierte Modellübergabe an eulersche Vernetzungen untersucht. Dadurch kann das Einsatzgebiet der interaktiven Konzeption bis hin zur Systemintegration erweitert und ein harter Schnitt zwischen den Simulationsmethoden vermieden werden.

9 References

- [1] E. Abele und G. Reinhart. *Zukunft der Produktion*. Hanser, Deutschland, 2011.
- [2] T. Döbler. *Simulation und Visualisierung in der Produktentwicklung*. MFG-Stiftung Baden-Württemberg, Deutschland, 2008.
- [3] U. Hedtstück. *Simulation diskreter Prozesse*. Springer, Deutschland, 2013.
- [4] S. Lecheler. *Numerische Strömungsberechnung*. Springer, Deutschland, 2014.
- [5] D. L. Davidson. *The Role of Computational Fluid Dynamics in Process Industries*. The Bridge Vol. 32 No. 4, S. 9-14, 2002.
- [6] P. Sagaut und S. Deck. *Large eddy simulation for aerodynamics: status and perspective*. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 367, S. 2849-2860, 2009.
- [7] N. J. Georgiadis, D. P. Rizzetta, and C. Fureby. *Large-Eddy Simulation: Current Capabilities, Recommended Practices, and Future Research*. AIAA Journal Vol. 48 No. 8, S. 1772-1784, 2010.
- [8] J. Kortelainen. *Meshing tools for open source CFD-a practical point of view*. VTT, Espoo, Finland, 2009.
- [9] T. J. Chung. *Computational Fluid Dynamics*. Cambridge University Press, England, 2002.
- [10] M. Hölling und H. Herwig. *CFD-TODAY: Anmerkungen zum kritischen Umgang mit kommerziellen Software-Programmpaketen*. Forschung im Ingenieurwesen Vol. 68 No. 3, S. 150-154, 2004.

- [11] Z. J. Wang et al. *High-order CFD methods: current status and perspective*. International Journal for Numerical Methods in Fluids Vol. 72: S. 811–845, 2013.
- [12] G. R. Liu. *Meshless Methods*. CRC press, USA, 2010.
- [13] V. P. Nguyen, T. Rabczuk, S. Bordas, and M. Duflot. *Meshless methods: a review and computer implementation aspects*. Mathematics and computers in simulation, Vol. 79 No. 3, S. 763-813, 2008.
- [14] F. Zhang., X. Shen, X. Long, B. Zhao und L. Hu. *Real-time particle fluid simulation with wcsph*. Pacific Graphics Short Papers, S. 29-34, 2012.
- [15] B. Solenthaler and P. Renato. *Predictive-corrective incompressible SPH*. ACM transactions on graphics (TOG). Vol. 28 No. 3. ACM, S. 40, 2009.
- [16] G. R. Liu und M. B. Liu. *Smoothed Particle Hydrodynamics – a meshfree particle method*. World Scientific Publishing, Singapur, 2003.
- [17] K. Ehrlenspiel. *Integrierte Produktentwicklung*. Hanser, Deutschland, 2002.

Simulation in der Fluidverfahrenstechnik – Virtuelle Inbetriebnahme in einer heterogenen Softwarelandschaft

Franz Schönberg, Marcel Nawrath, Jens Jäkel
HTWK Leipzig, Institut für Mess-, Steuerungs- und Regelungstechnik
franz.schoenberg@htwk-leipzig.de

Virtuelle Inbetriebnahmen sind eine Möglichkeit, immer komplexere Automatisierungssysteme in immer kürzerer Zeit zur Einsatzreife zu bringen. Wesentliche Voraussetzung für eine virtuelle Inbetriebnahme ist ein simulationsfähiges Anlagenmodell. Bei der Erstellung der Anlagenmodelle, der Testdurchführung und der abschließenden Auswertungen greifen oftmals spezialisierte Softwaremodule aus verschiedenen Anwendungsbereichen ineinander. Eine durchgängige Werkzeugkette unterschiedlicher Softwaremodule im Sinne eines integrierten Engineerings kann den Aufwand für eine virtuelle Inbetriebnahme verringern. In diesem Beitrag wird ein Konzeptentwurf für eine Werkzeugkette zur vereinfachten virtuellen Inbetriebnahme dynamischer Prozesse präsentiert, wie sie beispielsweise in der Fluidverfahrenstechnik auftreten. Diese basiert auf der Integration der Application Programming Interface (API)-Funktionalitäten und dem Erfassen aller Engineeringdaten in einem Datenmodell.

Bei der Erstellung des Anlagenmodells kommen erprobte Lösungen zum Einsatz, die in einem Baukasten zusammengefasst werden. Eine erprobte Lösung enthält neben physikalischen Modellen auch weitere Informationen, beispielsweise zu typischen Sensorkonfigurationen. Dadurch wird die Werkzeugkette, auch für Anwender ohne vertiefte Kenntnisse in der Simulationstechnik, einfacher in der Nutzung. Die vorgestellte prototypische Umsetzung verdeutlicht den angestrebten Arbeitsprozess und soll als Referenzdesign zu einer weiteren Standardisierung im Bereich des integrierten Engineering anregen.

1 Einleitung

Technologische Prozesse werden heute von modernen Automatisierungssystemen gesteuert. Teilweise kommen dabei komplexe Reglerstrukturen und Schutzsysteme zum Einsatz. Mit dem Umfang der Automatisierung steigt auch der Entwicklungsaufwand. Fehler im Automationsdesign werden wahrscheinlicher. Üblicherweise fallen diese Fehler erst bei der Inbetriebnahme der Anlage auf und verlängern dadurch die Inbetriebnahmedauer [1].

Die Technologien der virtuellen Inbetriebnahme (VIBN) ermöglichen es wesentliche Arbeitsaufgaben, wie zum Beispiel den Test und die Optimierung des Automatisierungssystems, zeitlich vorzuverlagern. Dies wird durch eine gekoppelte Simulation des Anlagenmodells mit der Automatisierungsanlage er-

reicht. Dabei wird unterschieden zwischen Hardware-in-the-Loop-Simulation (HiLS) und Software-in-the-Loop-Simulation (SiLS). Bei der SiLS (zum Teil auch als Systemsimulation bezeichnet) wird eine virtuelle Automatisierungsanlage, bei der HiLS eine reale Automatisierungsanlage mit dem virtuellen Anlagenmodell verschaltet. Die Möglichkeiten der VIBN lassen sich nach [2] wie folgt zusammenfassen:

- Verkürzung der Inbetriebnahmedauer
- Höhere Softwarequalität
- Inkrementelle und entwicklungsbegleitende Softwareerstellung
- Optimierungsmöglichkeiten der Steuerkonzepte
- Zerstörungsfreie Prüfung und Reproduktion von Fehlerszenarien
- Möglichkeit der Schulung und Präsentation

Gefördert durch das Bundesministerium für Bildung und Forschung (BMBF) unter dem Förderkennzeichen 03FH087PX2.

Trotz der vielfältigen Vorteile der VIBN werden, gerade bei einem Anlagenmodell für komplexe dynamische Prozesse, die Kosten der Modellerstellung oft nicht durch den Nutzen aufgewogen. Eine Möglichkeit der Kostensenkung ist die größtmögliche Automatisierung des VIBN-Prozesses. Im Idealfall kann ein Anlageningenieur ohne vertiefte Simulationskenntnisse auf Knopfdruck ein Simulationsmodell aus seinen Planungsdaten erzeugen. Um diesen Idealzustand zu erreichen, müssen die eingesetzten Softwarewerkzeuge untereinander perfekt im Sinne einer Werkzeugkette interagieren. In [3] wird dieser „Zusammenschluss unabhängiger Engineering-Werkzeuge zu einem Gesamt-Werkzeug“ auch als integriertes Engineering bezeichnet. Dabei geht die „angestrebte Integration [...] über Komponenten-, Lebenszyklus- und Gewerkegrenzen hinweg.“

In diesem Beitrag wird ein Softwarekonzept für die VIBN von Steuerungen mittels eines integrierten Engineering in einer heterogenen Softwarelandschaft vorgestellt. Das Konzept orientiert sich maßgeblich an der VIBN von Steuerungen dynamischer Prozesse im Bereich der Fluidverfahrenstechnik. In diesem Bereich sind Anlagen meist Unikate und durch nichtlineare Dynamik gekennzeichnet. Die Modellbildung verursacht aufgrund des Aufwands häufig hohe Kosten. Komplettlösungen (Softwaresuiten wie SIMIT von Siemens) sind in diesem Bereich selten verfügbar. Eine Übertragung des vorgestellten Konzeptes auf weitere Anwendungsbereiche ist aber ohne größere Modifikationen möglich.

Ausgehend von aktuellen Forschungen wird in den folgenden Abschnitten zunächst ein Konzept einer Werkzeugkette erarbeitet. Auf Basis dieser Werkzeugkette wird eine prototypische Umsetzung abgeleitet. Zum Abschluss werden die wesentlichen Erkenntnisse zusammengefasst und ein Ausblick gegeben.

2 Stand der Technik

Das Bestreben, den Engineering-Prozess durch verbesserte Arbeitsabläufe zu optimieren, führt heute in der industriellen Praxis zum Wunsch einer zunehmenden Integration von Softwarewerkzeugen unterschiedlicher Anbieter [4]. Die Integration kann dabei sowohl die erhobenen Daten, als auch die verwendeten Benutzeroberflächen umfassen. Eine auf Simulink [5] basierende VIBN-Lösung für den Bereich der Fluidverfah-

renstechnik wird in [6] vorgestellt. Ein beschriebenes Merkmal sind virtuelle Konnektoren, welche auch in Simulink den Eindruck einer physikalischen und auf Objekten basierender Modellierung erwecken. Durch die Realisierung aller gewünschten Funktionalitäten innerhalb einer einzelnen Anwendung kann ein durchgängiger Arbeitsprozess für den Anwender sichergestellt werden. Dieser Komfort wird jedoch mit einigen Nachteilen erkauft. Bei der Modellierung ist eine definierte Modellstruktur einzuhalten. Darüber hinaus ist eine Erweiterung der Simulationsbibliothek durch die Beschreibung der virtuellen Konnektoren schwierig. Eine Integration spezialisierter Softwarewerkzeuge könnte den mit einer Erweiterung verbundenen Aufwand reduzieren.

Ein Beispiel für eine vollständige Integration mit dem Ziel der VIBN wird in [7] vorgestellt. Das beschriebene Werkzeug CADSIMA ist eine eigenständige Software, welche sich die Funktionen der kommerziellen Softwarewerkzeuge „Autodesk Inventor“, „EPlan“ und „SimulationX“ zu Eigen macht. Das Projekt stellt jedoch in erster Linie eine Individuallösung dar. Ein allgemeines Konzept für die Integration von Software im Bereich der VIBN wird weder auf Datenebene, noch auf Bedienebene gezeigt.

Eine flexible Werkzeugintegration benötigt Schnittstellen insbesondere auf Ebene der erhobenen Daten (sogenannte Informationsmodelle). Als Standard für die strukturierte und plattformunabhängige Erfassung von Daten hat sich das XML-Format etabliert. Im Bereich der Automatisierungstechnik wurden, basierend auf XML, in den vergangenen Jahren Formate wie zum Beispiel AutomationML [8] für den strukturierten Datenaustausch spezifiziert.

Auch auf dem Gebiet der dynamischen Prozesssimulation gibt es erste Beispiele für ein erfolgreiches integriertes Engineering auf Basis von Informationsmodellen. In [9] wird ein solches Konzept für die aufwandsreduzierte Erstellung von Simulationsmodellen vorgestellt. Ausgehend von der Anlagenstruktur in Form eines Rohrleitungs- und Instrumentenfließschemas (R&I-Schema) werden im Datenformat CAEX schrittweise alle Informationen zur Anlage angereichert. Dies kann durch unterschiedliche Softwarewerkzeuge und im Rahmen des normalen Anlagenengineering erfolgen. Anschließend wird aus den erfassten Informationen automatisch ein Simulationsmodell für eine VIBN erstellt. Eine Umsetzung unter Verwendung von AutomationML ist in ähnlicher Wei-

se denkbar, da CAEX einen Teil von AutomationML darstellt.

Reine Modellbeschreibungssprachen weisen in der Regel keine strenge Strukturierung der enthaltenen Informationen auf. Eine solche Strukturierung würde die einfache Nutzung von Modellinformationen außerhalb von Simulationsanwendungen ermöglichen. In [10] wird die strukturierte Datenhaltung in der Beschreibung von Simulationsmodellen am Beispiel von Modelica diskutiert. Die strukturierte Erfassung der Daten des Engineeringprozesses und ein Mapping auf Modellbeschreibungssprachen wäre wünschenswert. In [11] wird ein solches Mapping von einer im CAEX-Format gespeicherte Objektstruktur auf die Modelica-Standardbibliothek [12] vorgestellt.

Auch vorkompilierte Modelle ließen sich in ein Informationsmodell integrieren. Eine standardisierte Schnittstelle zu vorkompilierten Modellen stellt das Functional Mockup Interface (FMI) zur Verfügung [13]. Eine Functional Mockup Unit (FMU) nutzt das FMI für die standardisierte Kommunikation zwischen dem in der FMU gekapselten physikalischen Simulationsmodell und der Simulationsumgebung. Die Beschreibung von Modelleigenschaften erfolgt hier bereits strukturiert im XML-Format. Eine Integration dieser FMU-Schnittstellenbeschreibung in das AutomationML-Informationsmodell wird in [14] beschrieben. Die FMU werden dabei aus einem Modelica Modell erzeugt. Der Einsatz von FMU kann den Schutz geistigen Eigentums zum Ziel haben. Anders als in Modelica lassen sich FMU ausschließlich durch gerichtete Signale untereinander verbinden. Ein Konzept physikalischer Konnektoren wie in Modelica ist nicht vorgesehen. Eine Optimierung über alle Systemgleichungen hinweg ist bei der Verschaltung von FMU daher nicht mehr möglich.

Bei der Modellbildung aus den Komponenten einer Modellbibliothek kommt es innerhalb einer Anwendungsdomäne oder Unternehmensumgebung immer wieder zu ähnlichen Aufgabenstellungen. Probleme, die in der Vergangenheit bereits gelöst wurden, treten so immer wieder neu auf. Diesen Umstand gilt es bei dem Entwurf von Informationsmodellen zu beachten. Eine mögliche Strukturierung stellen Lösungsbaukästen dar. Diese Baukästen beinhalten funktionsfähige Lösungen als Bauteile und werden beispielsweise in der VIBN von Werkzeugmaschinen eingesetzt [15, 2, 1]. Die Bauteile vereinen mechanische Funktionen und dienen bei der Erstellung

von Simulationsmodellen als Platzhalter. Das Bauteil-konzept bietet neben der erneuten Nutzung bereits erarbeiteter Lösungen eine Trennung der Prozesse der Simulationserstellung durch den Anwender und der Bauteilerstellung durch den Entwickler. Dadurch wird eine parallele Entwicklung von Simulationsmodellen und Bauteilen ermöglicht.

3 Konzept für ein integriertes Engineering in der Fluidverfahrenstechnik

Ein durchgängiger Arbeitsprozess für eine VIBN benötigt Softwareunterstützung in folgenden Bereichen:

- Grafische Modellerstellung
- Physikalische Modellierung und Zustandsberechnung von Fluiden
- Signal- und Hardwaremanagement
- Echtzeitsimulation (insbesondere für HiLS)
- Ergebnisdarstellung und -verwaltung

Gleichzeitig muss eine Werkzeugkette in der dynamischen Prozesssimulation folgende Randbedingungen erfüllen:

- Es gibt harte Echtzeitanforderungen, wie sie nur von speziellen Betriebssystemen sichergestellt werden können.
- Auch Ingenieure mit geringerer Erfahrung in der Simulationstechnik sollen in der Lage sein VIBN-Simulationen zu erstellen.
- Die Wartbarkeit von Modellen wird durch Standardisierung sichergestellt.
- Softwarekomponenten sollen sich einfach austauschen lassen.

Aus den benötigten Funktionen und den gegebenen Randbedingungen leiten sich wesentliche Forderungen an ein Konzept für die Softwareintegration ab (vgl. u.a. [16]):

1. Alle Funktionalitäten der integrierten Software werden als Module ausgeführt. Dadurch wird eine einfache Erweiterbarkeit gewährleistet.

2. Es gibt keine Bindung an bestimmte Softwarehersteller.
3. Alle im Modellierungsprozess erhobenen Daten werden zentral verwaltet. Dafür kommen allgemein akzeptierte Standards zum Einsatz.
4. Es gibt eine Bibliothek vorgefertigter Modelle (Lösungen). Jede Lösung wird darauf überprüft, in der definierten Softwareumgebung simulationsfähig zu sein.
5. Es findet eine strikte Trennung zwischen der Erstellung der Lösungen und der Erstellung der Simulation für die VIBN statt.
6. Templates verschiedener Arbeitsprozesse sind den Lösungen zugeordnet und transportieren auf diese Weise benötigtes simulationstechnisches Know-How.
7. Es kommt eine standardisierte Modellierungssprache zum Einsatz, welche Modelle als physikalische Objekte beschreiben kann.
8. Das Gesamtmodell liegt in einem weitgehend plattformunabhängigen Format vor und ermöglicht so die Kompatibilität mit gängigen Echtzeitbetriebssystemen.

Derzeit kann keine bestehende Werkzeugkette alle gestellten Forderungen für die VIBN der Automatisierung dynamischer Systeme der Fluidverfahrenstechnik erfüllen. Eine Realisierung erscheint nach dem heutigen Stand der Technik jedoch möglich.

Die Forderungen 1-3 bestimmen wesentlich die Struktur der Softwareintegration in die Werkzeugkette. Abbildung 1 zeigt den vorgeschlagenen Aufbau.

Alle wesentlichen Funktionalitäten der VIBN werden in Module ausgelagert. Die für die Simulation benötigten Daten werden in einem strukturierten Datenmodell erfasst. Dieses gliedert sich idealerweise mit dem Ziel der Standardisierung in ein Informationsmodell, wie beispielsweise AutomationML, ein. Weiterhin ist ein Import von Modellbibliotheken aus externen Quellen, wie beispielsweise Webservern, in das Informationsmodell vorzusehen. Dies kann der zentralen Verteilung von Updates oder dem Know-How-Schutz dienen.

Das Datenmodell ermöglicht in dem entwickelten Konzept die Übernahme von Planungsdaten aus allen Phasen des Anlagenengineerings. Unter

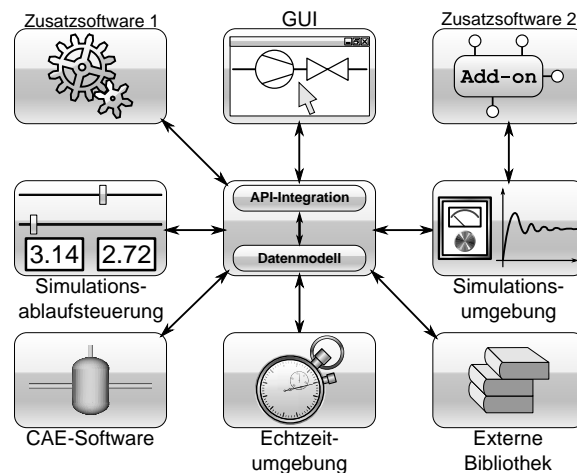


Abbildung 1: Struktur eines integrierten Engineerings für die VIBN in der Fluidverfahrenstechnik

Umständen können bei der Planung einer VIBN-Softwareumgebung auch Abweichungen von dieser Struktur sinnvoll sein. Dies gilt insbesondere für Werkzeugketten bei VIBN-Dienstleistern. Für diese Dienstleister sind Planungsdaten oft nicht digital oder nur in sehr unterschiedlichen Formaten verfügbar. In diesem Fall beschränkt sich die Werkzeugkette auf die reine Integration der für die Simulation und die Auswertung benötigten Komponenten.

Für die Unabhängigkeit von bestimmten Softwareherstellern ist die Trennung von grafischer Darstellung und gewünschter Funktionalität entscheidend. Dieses Vorgehen entspricht dem „Model View Controller“ (MVC) Ansatz, wie er bei der Entwicklung eigenständiger Softwareprodukte zum Einsatz kommt. Durch die Integration einer eigenständigen grafischen Benutzeroberfläche (GUI) in den Modellbildungsprozess werden mehrere Effekte erreicht. Einerseits führen Änderungen der integrierten Softwarewerkzeuge nur zu kleinen Änderungen der GUI und erhalten so den Anlerneffekt. Andererseits werden Softwarelizenzen nicht mehr für den gesamten Modellbildungszeitraum benötigt, was sich auf die Lizenzkosten positiv auswirken kann.

Die Forderungen 4-6 nehmen Bezug auf die Strukturierung der Daten in einer Modellbibliothek unter Erweiterung des Bauteilbegriffs. Bauteile nach dieser Definition enthalten neben mehreren physikalischen Modellen auch grafische Elemente, Templates zu Messstellen und Testfällen. Die physikalischen

Modelle eines Bauteils (Objekte) unterscheiden sich unter anderem durch ihre Modellierungstiefe. Eine hierarchische Datenstruktur erfasst die im Bauteil gespeicherten Informationen (siehe Abbildung 2).

Ein Bauteil ist immer eine bereits erprobte Lösung. Es wird in einem von der VIBN-Modellbildung getrennten Prozess (weiter-) entwickelt. Auch konzeptionell wird zwischen den Softwarekomponenten der VIBN-Modellbildung durch den Anwender und der Erstellung des Baukastens durch den Entwickler unterschieden. Die zugehörigen Softwarebetriebsmodi unterteilen sich in einen Anwendermodus und einen Entwicklermodus.

Durch die Unveränderlichkeit der Bauteile im Anwendermodus werden einheitliche Modellierungsstandards beispielsweise innerhalb einer Firma garantiert. Ein Bauteil verfügt im Anwendermodus ausschließlich über Parameter, die die physikalische Modellstruktur nicht beeinflussen.

Die Datenstruktur des VIBN-Modells ist für den Anwendermodus in Abbildung 3 dargestellt. Ausgehend von der grafischen Beschreibung werden die Elemente der Anlage (Bauteile und Konnektoren) sowie deren Abhängigkeiten in dieser Datenstruktur erfasst. Die Datenstruktur der Simulationsbeschreibung kann wiederum Teil eines standardisierten Informationsmodells sein. Ein Konnektor stellt entweder eine direkte (verlustlose) Verbindung zwischen zwei Bauteilen her oder weist selbst die Eigenschaften eines Bauteils auf. Als Konnektor geeignete Bauteile haben genau zwei gleichartige physikalische Schnittstellen. Mit Hilfe eines Bauteilcontainers werden alle Attribute eines Bauteils innerhalb der Datenstruktur als Kopie abgelegt. Auch wenn ein Bauteil im Entwicklermodus weiterentwickelt wird, bleibt eine bereits erstellte Simu-

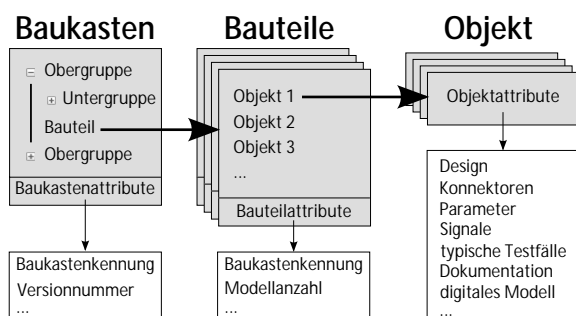


Abbildung 2: Logische Strukturierung der Objekte eines Baukastens und beispielhafte Eigenschaften (Attribute)

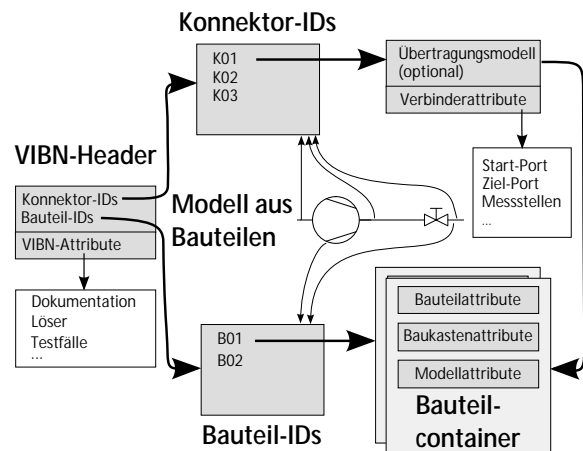


Abbildung 3: Datenstruktur eines VIBN-Simulationsmodells im Anwendermodus

lation unverändert. Über einen Migrationsassistenten können veränderte Bauteildaten dann in bestehende Bauteilcontainer integriert werden.

Um die Forderungen 7 und 8 zu erfüllen, spezifizieren wir als Modellbeschreibungssprache Modelica. Modelica ist ein offener und objektorientierter Standard in der Modellierung. Der für die Kompilierung der Bauteile relevante Modelica-Code wird im Bauteilcontainer erfasst. Erst nach dem Erstellen des Gesamtmodells durch den Anwender werden alle Modelica-Code-Fragmente zu einem Code zusammengeführt und für das Zielsystem kompiliert. Dadurch werden eine bestmögliche symbolische Vereinfachung auf Ebene der Systemgleichungen und eine bestmögliche Unabhängigkeit von der Systemumgebung erreicht. Darüber hinaus ermöglicht Modelica die Nutzung von externem C-Code oder von Funktionen aus vorkompilierten Bibliotheken. Damit wird die Voraussetzung für die Einbindung von externen Berechnungsmodulen (beispielsweise für Gaszustandsberechnungen) geschaffen. Im Gegensatz zu Modelica erlaubt C-Code auch ein effektives Debugging dieser Algorithmen.

4 Prototypische Werkzeugkette

Eine prototypische Werkzeugkette für die VIBN der Automatisierung prozesstechnischer und fluidverfahrenstechnischer Anlagen lässt sich aus dem Konzept des integrierten Engineering ableiten. Diese orien-

tiert sich im Anwendermodus an den praktischen Arbeitsabläufen von Dienstleistern für VIBN. Ausgangspunkt für die Erstellung eines Anlagenmodells für die Fluidverfahrenstechnik ist dabei in der Regel das R&I-Schema der zu modellierenden Anlage. Bei einem Dienstleister liegt das R&I-Schema meist nur als schriftliches Dokument vom Kunden vor. Aus diesem Grund verzichtet die prototypische Werkzeugkette zunächst auf die Möglichkeit des Imports aus CAE-Tools.

Ziel der Werkzeugkette ist die bestmögliche Unterstützung des Arbeitsprozesses. Die Schritte eines VIBN-Arbeitsprozesses folgen einer chronologischen Abfolge und sind in Tabelle 1 dargestellt. In der Tabelle werden außerdem die beteiligten Softwaremodule sowie die ausgetauschten Daten charakterisiert. Vermittelt wird der Daten Im- und Export sowie der Aufruf der einzelnen Module durch die API-Integration.

Die in der prototypischen Werkzeugkette enthaltenen Softwaremodule nutzen im Sinne eines integrierten Engineerings kommerziell oder frei verfügbare Softwareprodukte Dritter. Die API-Integration gewährleistet dabei, dass innerhalb der Werkzeugkette ein Softwareprodukt jederzeit durch ein anderes mit ähnlichem Funktionsumfang ersetzt werden kann. Die Wahl der zu verwendenden Softwareprodukte kann von wirtschaftlichen Faktoren abhängig sein oder sich aus dem gebotenen Funktionsumfang ergeben.

Die in der prototypischen Werkzeugkette verwendeten Softwarewerkzeuge sind:

1. Microsoft Visio [17] als GUI mit der Möglichkeit grafischer Modellierung. Dass Visio sinnvoll für die Erstellung von R&I-Schemata eingesetzt werden kann, wird auch durch die Software Visual PlantEngineer [18] gezeigt.
2. ITI SimulationX [19] als Modelica-Interpreter eingesetzt ermöglicht den Export von Simulationsmodellen in zahlreiche Formate.
3. Mathworks Simulink [5] stellt zusammen mit dem Simulink Coder den Compiler zur Verfügung. Darüber hinaus eignet sich Simulink für die Modellvalidierung.
4. Mathworks xPC Target [20] wird auf Grund einer großen Zahl unterstützter Hardwarekomponenten als Echtzeitbetriebssystem eingesetzt.

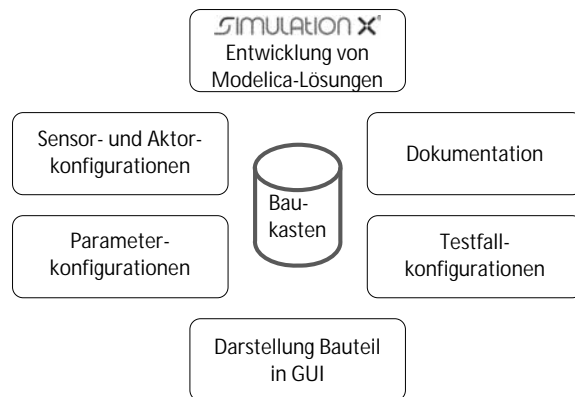


Abbildung 4: Anreicherung von Informationen im Entwicklermodus

5. Mathworks MATLAB [21] dient einer scriptgesteuerten Auswertung der Simulationsergebnisse.

Da die API-Gestaltung von Software vom Hersteller abhängig ist, kann ein Austausch von Softwarekomponenten mit erheblichem Aufwand verbunden sein. Um diesen Aufwand zu reduzieren ist ein Plugin-System vorgesehen. Einmal entwickelte Plugins können so beliebig weiter verteilt werden.

Das Modul für das Mapping der Modelica-Objekte in das Anlagenmodell lässt sich nicht durch eine verfügbare Software darstellen und ist Teil der für die Werkzeugkette vorzunehmenden Implementierungen. Aus der Spezifikation des Anwendermodus der Werkzeugkette folgen unmittelbar die Anforderungen an einen Entwicklermodus für die Bauteile des Baukastens. Im Entwicklermodus werden die Bauteile mit verschiedenen Informationen in Form von Attributen angereichert (siehe Abbildung 4). Der Hauptaufwand der Bauteilentwicklung liegt dabei in der physikalischen Modellierung. Diese wird nicht im Entwicklermodus sondern im Falle der prototypischen Umsetzung mit SimulationX vorgenommen. Da eine bestimmte Lösung immer nur einmalig in den Baukasten aufgenommen wird, beschränkt sich die GUI des Entwicklermodus auf einen dateibasierten Import von Design und Modelica-Objekten sowie textfeldbasierte Eingabemasken für alle weiteren Informationen. Durch eine einfach gestaltete GUI im Entwicklermodus wird der Implementierungsaufwand der Werkzeugkette reduziert.

Tabelle 1: Chronologischer VIBN-Arbeitsprozess mit assoziierten Modulen und Daten

Arbeitsschritt	Module	Erzeugte Daten	Benötigte Daten
Objektschema (R&I) aus Bauteilen erstellen	GUI	<ul style="list-style-type: none"> Struktur der Bauteilverschaltung 	<ul style="list-style-type: none"> Verfügbare Konnektoren Darstellung der Bauteile
Parametrierung	GUI	<ul style="list-style-type: none"> Parameter 	<ul style="list-style-type: none"> Parameterlisten
Schnittstellen für Sensoren und Aktoren konfigurieren	GUI	<ul style="list-style-type: none"> Sensorik- und Aktorik-konfiguration 	<ul style="list-style-type: none"> Typische Sensorik- und Aktorikkonfiguration
Testfalldefinition	GUI	<ul style="list-style-type: none"> Testfälle 	<ul style="list-style-type: none"> Typische Testfall-konfigurationen
Erstellung des Anlagenmodelles in Modelica aus verfügbaren Informationen	Mapping-Modul	<ul style="list-style-type: none"> Anlagenmodell in Modelica 	<ul style="list-style-type: none"> Parameter Sensorik- und Aktorik-konfiguration Testfälle Struktur der Bauteilverschaltung
Übersetzen des Anlagenmodells in algorithmisches Modell (bsp. C-Code)	Modelica-Interpreter	<ul style="list-style-type: none"> Algorithmisches Modell 	<ul style="list-style-type: none"> Anlagenmodell in Modelica
Übersetzen in echtzeitfähigen Maschinencode	Compiler	<ul style="list-style-type: none"> Echtzeitcode 	<ul style="list-style-type: none"> Sensorik- und Aktorik-konfiguration Algorithmisches Modell
Simulation	Echtzeit-OS	<ul style="list-style-type: none"> Rohergebnisse 	<ul style="list-style-type: none"> Echtzeitcode
Testauswertung	Auswertemodul	<ul style="list-style-type: none"> Aufbereitete Ergebnisse 	<ul style="list-style-type: none"> Rohergebnisse Auswertekriterien (Skripte)

5 Zusammenfassung und Ausblick

Der vorliegende Beitrag stellt eine Werkzeugkette für eine vereinfachte VIBN-Modellerstellung mittels Softwareintegration vor. Diese Vereinfachung wird durch einen durchgängigen Arbeitsprozess erreicht. Kernbestandteil ist eine modulare Software-Architektur für ein integriertes Engineering. Diese wird neben einer einheitlichen Datenbasis auch durch eine übergeordnete API-Integration und eine eigenständige Benutzeroberfläche charakterisiert. Die Eingliederung getesteter Lösungen in einen Baukasten ermöglicht eine strikte Trennung der Komponenten der Werkzeugkette in einen Anwender- und einen Entwicklermodus. Der Anwender wird bei der Integration neuentwickelter Bauteile in bestehende Modelle durch einen Migrationsassistenten unterstützt.

Das vorgestellte prototypische Referenzdesign in-

tegriert Funktionen der Softwarewerkzeuge SimulationX, xPC Target und MATLAB/Simulink. Als Editor für die grafische Modellierung wurde Microsoft Visio vorgeschlagen. Nach den Ergebnissen praktischer Tests in der Industrie sollen weitere Software-schnittstellen in das prototypische Referenzdesign implementiert werden.

Ein generelles Problem bei der Softwareintegration für VIBN ist das Fehlen offener Referenzimplementierungen. So lassen sich Aufwand und Nutzen eines integrierten Engineering in der Praxis kaum abschätzen. Aus diesem Grund soll eine Veröffentlichung des Referenzdesigns stattfinden. Dafür soll ein weiterentwickeltes Datenmodell auf Basis von AutomationML zum Einsatz kommen. Insbesondere die Integration von API-Plug-Ins ist im Rahmen einer Nutzercommunity sinnvoll. Damit soll ein Beitrag zur stärkeren Verbreitung von VIBN in der Prozessindustrie geleistet werden.

Literatur

- [1] G. Wünsch. *Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme*. Dissertation, TU München, 2007.
- [2] A. Lindworsky. *Teilautomatische Generierung von Simulationsmodellen für den entwicklungsbegleitenden Steuerungstest*. Dissertation, TU München, 2011.
- [3] T. Tauchnitz. Integriertes Engineering – wann, wenn nicht jetzt! *atp edition*, 1:S. 46–53, 2013.
- [4] S. Mühlenkam. PROCESS-Umfrage zeigt: Anwender sind vom Nutzen integrierter Engineering-Systeme überzeugt. online, 2013. URL http://www.process.vogel.de/anlagenbau_engineering/articles/420703/
- [5] Mathworks Simulink. online, 2014. URL www.mathworks.de/products/simulink/
- [6] R. Schulze, R. Huber, F. Schönberg und J. Jäkel. Virtuelle Inbetriebnahme von Verdichter- und Turbinensteuerungen. *VDI Automatisierungskongress Baden-Baden*. 2014.
- [7] U. Schob, R. Böttcher, T. Blochwitz, O. Oelsner und M. Winter. Model based Virtual Startup of Automation Systems. *Proceedings 7th Modelica Conference, Como, Italy, Sep. 20-22*. 2009.
- [8] AutomationML e.V. online, 2014. URL <https://www.automationml.org>
- [9] M. Barth, M. Strube, A. Fay, P. Weber und J. Greifeneder. Object-oriented engineering data exchange as a base for automatic generation of simulation models. *Tagungsband der IEEE IECON 2009 the 35th Annual Conference of the IEEE Industrial Electronics Society*. 2009.
- [10] U. Reisenbichler, H. Kapeller, A. Haumer, C. Kral, F. Pirker und G. Pascoli. If we only had used XML... *5th Modelica conference*. 2006.
- [11] M. Barth. *Automatisch generierte Simulationsmodelle verfahrenstechnischer Anlagen für den Steuerungstest*. Dissertation, Fakultät für Maschinenbau der Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg, 2011.
- [12] Modelica. online, 2014. URL <https://www.modelica.org/>
- [13] Functional Mockup Interface. online, 2014. URL <https://www.fmi-standard.org/>
- [14] O. Graeser, B. Kumar, O. Niggemann, N. Moriz und A. Maier. AutomationML as a Shared Model for Offline- and Realtime-Simulation of Production Plants and for Anomaly Detection. *Informatics in Control, Automation and Robotics*, S. 195–209. Springer Berlin Heidelberg, 2013.
- [15] A. Kufner. *Automatisierte Erstellung von Maschinenmodellen für die Hardware-in-the-Loop-Simulation von Montagemaschinen*. Dissertation, Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen der Universität Stuttgart, 2012.
- [16] T. Tauchnitz. Schnittstellen für das integrierte Engineering. *atp edition*, 1:S. 30–36, 2014.
- [17] Microsoft Visio. online, 2014. URL <http://office.microsoft.com/de-de/visio/>
- [18] x-visual Visual PlantEngineer. online, 2014. URL <http://www.x-visual.net>
- [19] ITI SimulationX. online, 2014. URL <http://www.simulationx.com>
- [20] Mathworks xPC Target. online, 2014. URL <http://www.mathworks.de/products/xpctarget>
- [21] MathWorks Matlab. online, 2014. URL <http://www.mathworks.de/products/matlab/>

Effizienter Vergleich von CFD- Simulationen und Experimentaldaten durch Standardisierung und Automatisierung

Andreas Probst, Ansgar Ringleb, Wolfgang Schlüter

Kompetenzzentrum Strömungssimulation - KSTS

Hochschule Ansbach

a.probst@hs-ansbach.de

Moderne Forschungsmethoden ermöglichen durch die Kombination von analytischen, experimentellen und numerischen Methoden detaillierte Strömungsuntersuchungen. Mit steigender Rechenleistung gewinnt die numerische Analyse von Strömungen immer mehr an Bedeutung. Dennoch ist diese Entwicklung noch nicht so weit voran geschritten, um experimentelle Untersuchungen zu ersetzen. Auf absehbare Zeit wird es nicht möglich sein, komplexe Strömungen allein durch Simulationen zu untersuchen. Vielmehr müssen die Vorteile von Experimenten und Simulationen kombiniert werden, um Analysen effizient durchführen zu können.

Die Validierung von CFD-Simulationen (CFD= Computational Fluid Dynamics) erfolgt häufig durch Geschwindigkeitsmessungen mit Hilfe der Laser-Doppler-Velocimetry (LDV / LDA). Bei dieser Vorgehensweise entstehen speziell bei transienten Strömungen große Datenmengen, die nur schwer händisch ausgewertet werden können. Um die Validierung effizient zu gestalten, ist es notwendig die Erfassung der Messwerte und die Datenextraktion aus der CFD-Simulation zu standardisieren. Einhergehend mit der einheitlichen Strukturierung der Datensätze ist ein automatisierter Vergleich möglich und die Genauigkeit einer Simulation kann beurteilt werden.

1 Einleitung

Strömungsberechnungsprogramme sind mittlerweile weit verbreitete Hilfsmittel, die in Forschung und Industrie gleichermaßen zum Einsatz kommen. Sie tragen maßgeblich zur Effizienzsteigerung bei der Entwicklung neuer Apparate bei. Numerische Untersuchungen bieten Vorteile bei der Analyse von experimentell schwer handhabbaren Problemstellungen. Flexible Möglichkeiten der Datenextraktion und die frei wählbaren Randbedingungen erleichtern Untersuchungen unter Betriebsbedingungen.

Bei der Durchführung einer CFD-Simulation wird ein mathematisch-physikalisches Modell in Form eines Anfangs-Randwertproblems näherungsweise abgebildet. Es ist durchaus möglich, dass eine physikalisch falsche aber numerisch konvergente Lösung plausibel erscheint. Beispielsweise können die Residuen trotz der Eingabe physikalisch falscher Randbedingungen konvergieren. Eine solche Lösung als richtig anzusehen, kann schwerwiegende Konsequenzen haben.

Deshalb ist es notwendig, CFD-Simulationen mittels experimentell ermittelter Daten zu validieren. Die

genaueste Technik für derartige Analysen ist die LDV. Dabei handelt es sich um ein nichtinvasives laseroptisches Messgerät mit dem punktuell transiente Geschwindigkeitsverläufe aufgezeichnet werden können. Bei dieser Messtechnik treten nur minimale Messfehler von unter 0,5 % auf.

Soll eine Simulation mittels experimentell ermittelter Daten validiert werden, muss eine Vielzahl von Werten miteinander verglichen werden. An jedem Messpunkt entsteht eine hohe Anzahl an Geschwindigkeitswerten (> 1000), die sich mit der Anzahl der Messpunkte multiplizieren. Um dieses hohe Datenaufkommen effizient mit den Simulationsdaten vergleichen zu können, muss dieser Prozess standardisiert und automatisiert werden.

Für die Standardisierung ist es notwendig, den Mess- und Simulationsprozess zu charakterisieren und geeignete Schnittstellen zu identifizieren. Dabei ist auf spezifische Einschränkungen in Versuch und Simulation zu achten. Dafür muss zunächst das Vorgehen zur Messdatenerfassung und Extraktion der Simulationsdaten festgelegt werden. Danach kann die Schnittstelle zur Validierung definiert werden. Abschließend

wird die Verarbeitung der Datensätze mit Hilfe eines speziell entwickelten Auswertungstools auf MATLAB-Basis und einer LabVIEW Anwendung umgesetzt.

2 Stand der Wissenschaft

Die Validierung von CFD-Simulationen ist Teil vieler Veröffentlichungen. Jede strömungstechnische Fragestellung, die mit Hilfe von CFD-Simulationen untersucht wird, muss auch validiert werden. Händisch nachrechenbar sind nur sehr einfache Geometrien. Komplexe Strömungsvorgänge müssen deshalb oft durch Experimente validiert werden. (Oberkampf & Trucano, 2002) Aufgrund der vielseitigen Anwendungsgebiete der CFD existieren nur auf spezifische Anwendungsfälle bezogene Automatisierungslösungen. Oft liegen nur eingeschränkt Messdaten vor, die nicht während des aktuellen Entwicklungsprozesses erhoben wurden. Eine Automatisierung erfolgt aus diesem Grund oft nur für die Durchführung und Auswertung der Simulationsdaten. Die vorliegende Untersuchung behandelt die Automatisierbarkeit der Simulationsvalidierung von Rohrströmungen mit beliebigen Einbauten unter dem Gesichtspunkt einer ganzheitlichen Betrachtung von Simulation und Experiment.

3 Referenzfall

Als Referenzfall wurde der Einfluss von Störkörpern auf Volumenstromzähler untersucht. Durch immer kleiner werdende Übergabestationen und die damit einhergehenden kompakteren Bauräume steigt der Einfluss von Störkörpern auf Volumenstrommessungen. Für die Zulassung von Messgeräten muss die Funktion der Geräte unter dem Einfluss von unterschiedlichen Störeinflüssen nachgewiesen werden. Auch die Einflüsse des eigentlichen Messeinsatzes selbst sind von großer Bedeutung, da die Geräte zur Kalibrierung und Prüfung oft hintereinander geschaltet werden (Reihenprüfung). In Übergabestationen sind Raumkrümmer als äußerst kritisch einzustufen, da dadurch die Strömung in Drall versetzt wird. In der Prüfvorschrift ist ebenfalls ein so genannter Drallgenerator vorgesehen, mit dem die Strömung ebenfalls in Drall versetzt wird. Volumenstromzähler müssen trotz dieser Störeinflüsse über den gesamten Messbereich zuverlässige Ergebnisse liefern. Dadurch entstehen zahlreiche Variationen, die zu untersuchen sind. Im vorliegenden Beispiel sind es trotz der Beschränkung auf ein Minimum 27 Variationen.

4 Versuchsaufbau

Um experimentelle Untersuchungen in höchster Qualität und mit vertretbarem zeitlichem Aufwand durchführen zu können, wurde ein Prüfstand zur Vermessung von Luftströmungen konzipiert (Abb. 1). Dabei wurde speziell auf die Schnittstellenkompatibilität der eingesetzten Messgeräte geachtet, um die Messdatenerfassung und Versuchsdurchführung weitestgehend zu automatisieren. Dadurch wird die ganzheitliche digitale Erfassung aller Messwerte und die Steuerung der Messgeräte mittels LabVIEW oder spezieller Software der Gerätehersteller ermöglicht.

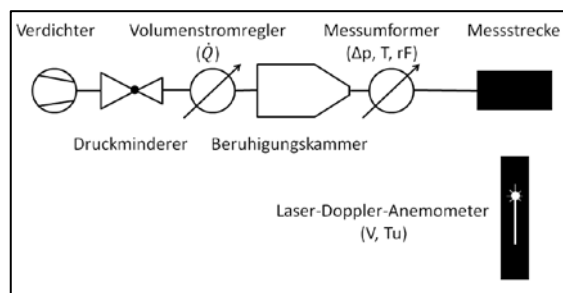


Abbildung 1: Anlagenschema des LDV-Luftprüfstands

Um die Messungen exakt und reproduzierbar durchführen zu können, ist die Erzeugung einer definierten Strömung eine wesentliche Voraussetzung. Dazu wird ein Verdichter in Verbindung mit einer Druckminderstufe eingesetzt. Mit dieser Kombination ist es möglich einen Volumenstrom mit konstantem Vordruck zu erzeugen. Die Regelung und Einstellung der exakten Menge erfolgt mit Volumenstromreglern. Anschließend wird die Strömung mit einer speziell ausgelegten Beruhigungskammer von Störeinflüssen entkoppelt und ein ungestörtes Strömungsprofil erzeugt. Zur Definition der Randbedingungen besteht die Möglichkeit Druck-, Temperatur- und Feuchtigkeitsmessungen innerhalb der Versuchsgeometrie durchzuführen.

Für die Erfassung der Strömung in der Messstrecke kommt ein 1D-Laser-Doppler-Anemometer (LDA) der Firma ILA GmbH zum Einsatz (Abb. 2). Dabei handelt es sich um ein Messgerät zur berührungslosen, punktuellen und nicht invasiven Erfassung von Geschwindigkeits- und Turbulenzgrößen. Dieses laseroptische Messgerät ist in Rückstreuunganordnung ausgeführt, wodurch nur sehr kleine Eingriffe in die strömungsführende Geometrie notwendig sind, um den optischen Zugang zu gewährleisten. (Oertel, 1989)

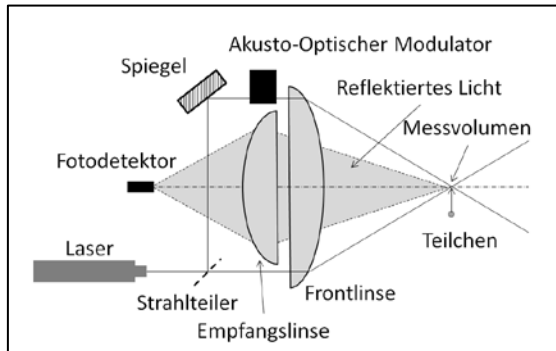


Abbildung 2: Messprinzip Laser-Doppler-Anemometer

Zur experimentellen Untersuchung des Referenzfalls wurde eine Messstrecke konstruiert, mit der Strömungsprofile vor und nach einem Messeinsatz unter dem Einfluss von speziellen Störkörpern wie beispielsweise Raumkrümmer, Drallgenerator oder einem weiteren Messeinsatz erfasst werden können (Abb. 3).

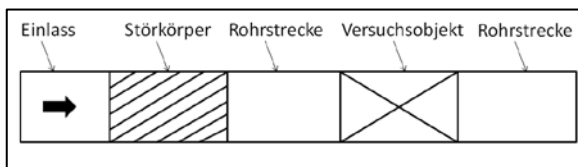


Abbildung 3: Messstrecke

Für die Validierung der Simulation wurden bestimmte Betriebspunkte ausgewählt und experimentell vermessen. Der Einlass wurde mit einem ungestörten Kolbenprofil beaufschlagt. Direkt im Anschluss befinden sich die Störkörper. Über ein Verbindungsstück ist das eigentliche Versuchsobjekt angeflanscht und mit einer Rohrstrecke als Auslauf verbunden. Der komplette Versuchsaufbau hat eine Gesamtlänge von 4 m. Davon werden 1,5 m für die Strömungskonditionierung (Abb. 1) benötigt und die eigentliche Messstrecke (Abb. 3) hat eine Länge von 2,5 m.

5 CFD-Software

Die vorliegenden Untersuchungen wurden mit der Software Star CCM+ der Firma CD Adapco durchgeführt. Dabei handelt es sich um eine kommerzielle Software zur Strömungssimulation. Für die Berechnung turbulenter Strömungen in Rohren wird vornehmlich das k-ε-Modell oder SST-Modell verwendet. Aufgrund der Modellierung der turbulenten Effekte entstehen Abweichungen zwischen Simulationsergebnis und realer Strömung. Um diese Abweichungen zu beziffern und Ergebnisse kritisch bewerten zu können, ist die Validierung einer Simulation

notwendig. (Ferziger & Peric, 2002) Bei der Bearbeitung von ingenieurstechnischen Fragestellungen werden oft hohe Anzahlen von geometrischen Variationen untersucht (Tab. 1). Hier liegen die Vorteile einer CFD-Simulation. Die Simulation dieser Varianten lässt sich mit Hilfe von Makros und Skripten automatisieren. Dieses Vorgehen ermöglicht die zeit-sparende Untersuchung derartiger Fragestellungen.

6 Validierung

Im ersten Schritt müssen geeignete und aussagekräftige Vergleichswerte gefunden werden. Für die Validierung von Strömungssimulationen können mehrere Ansatzpunkte identifiziert werden. Über den Vergleich von Geschwindigkeitsprofilen hinaus besteht die Möglichkeit, weitere statistische Größen, wie den Turbulenzgrad, zu analysieren. Geschwindigkeitssignale ($u(t)$) stationärer turbulenter Strömungen die mit hoher zeitlicher Auflösung aufgenommen wurden (>1000 Hz) schwanken mit einer für die Turbulenz charakteristischen Standardabweichung um den Mittelwert. Die Abweichung vom Mittelwert (\bar{u}) für ein Messereignis (Burst) wird als Schwankungsgröße (u'_i) bezeichnet. Der Turbulenzgrad entspricht der Standardabweichung von der mittleren Strömungsgeschwindigkeit für einen Punkt der Strömung und berechnet sich aus dem Verhältnis der Schwankungsgrößen zum Mittelwert (Gl. 1).

$$T_u = \frac{1}{n-1} \sum_{i=1}^N \sqrt{(\bar{u} - u'_i)^2} \quad (\text{Gl. 1})$$

Durch die Vermessung von Geschwindigkeits- und Turbulenzprofilen lassen sich aussagekräftige Vergleichswerte generieren. Derartige Profile können mit vertretbarem Zeitaufwand experimentell ermittelt (Abb. 4) und einfach aus der Simulation extrahieren werden.

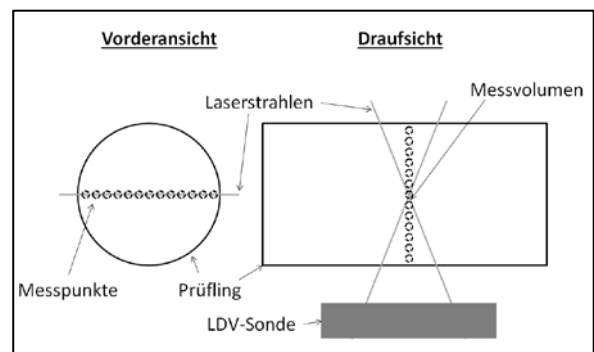


Abbildung 4: Messung von Strömungsprofilen

Ein Abgleich von Profilen und Punktmessungen die an vorher definierten Bilanzflächen und bestimmten

Bereichen der Strömung aufgenommen werden dient zur Bewertung der Abweichungen.

7 Frequenzanalyse

Turbulente Geschwindigkeitsfelder zeichnen sich durch starke Irregularitäten aus. Dies führt zu einer hohen Informationsdichte. Daher muss die Menge an Informationen auf ein angemessenes Maß reduziert werden. Eine geeignete Vorgehensweise zur Identifikation von Regelmäßigkeiten in turbulenten Strömungen ist die Frequenzanalyse. (Durst, 2006)

Die gebräuchlichste Methode zur Frequenzanalyse ist die diskrete Fourier- Transformation. Die Modifikation des Algorithmus führt zur Fast Fourier Transformation (FFT). Dadurch lässt sich der Rechenaufwand um den Faktor 100 reduzieren.

Die Herausforderung bei der Frequenzanalyse von LDV-Messdaten ist das nicht äquidistante Auftreten von Messereignissen, sogenannten Bursts (Abb.5). Aufgrund der hohen Standardabweichung der Zeitabstände zwischen den einzelnen Messereignissen kann in diesem Fall die DFT nur stark eingeschränkt angewendet werden. Für die Analyse von LDV-Messdaten ist das Lomb-Scargle Periodogram (L-S Periodogram) effizienter. Ein Periodogram beinhaltet das Spektrum eines Signals über der Frequenz und eignet sich zur Identifikation von dominanten Geschwindigkeiten im Verlauf der nicht äquidistanten Zeitsignale aufgrund der Gewichtung der einzelnen Messereignisse. (Zechmeister & Kürster, 2008) Dazu wird jedem Burst ein entsprechendes Zeitfenster zugeordnet und je nach Dauer zwischen vorhergehenden und erneuten Burst gewichtet.

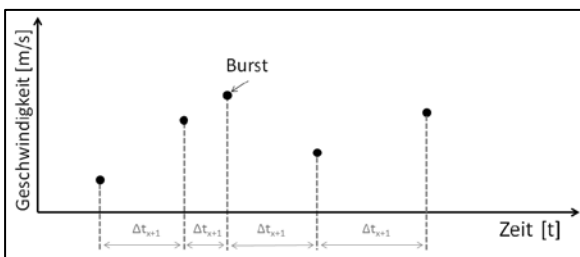


Abbildung 5: Nicht äquidistantes LDV-Zeitsignal

Voraussetzung für die Frequenzanalyse ist eine entsprechend hohe Messrate. Das Abtasttheorem nach Shannon besagt, dass der Abstand zwischen den Funktionswerten mindestens um den Faktor zwei größer sein muss als die maximal auftretende Frequenz (f_n) (Gl. 2). Dazu müssen sich die Stützpunkte allerdings auf den Extremwerten befinden.

$$f_s > 2f_n \quad (\text{Gl.2})$$

Um eine Frequenz mit zufällig verteilten Stützpunkten exakt bestimmen zu können wird eine Abtastrate benötigt, die um den Faktor zehn höher ist als die Amplitudendauer der höchsten auftretenden Frequenz (National Instruments, 2011).

$$f_s > 10f_n \quad (\text{Gl.3})$$

Somit muss bei der Frequenzanalyse besonders auf die Messbedingungen geachtet werden, um die notwendigen Abtastraten zu erreichen. Kann das Abtasttheorem nicht eingehalten werden, ist zu beachten, dass eventuell aufgrund einer zu geringen Abtastrate bestimmte Frequenzen nicht analysierbar sind und im schlimmsten Fall stochastische Schwankungen als der Grundströmung überlagerte Frequenzen detektiert werden (Abb. 6).

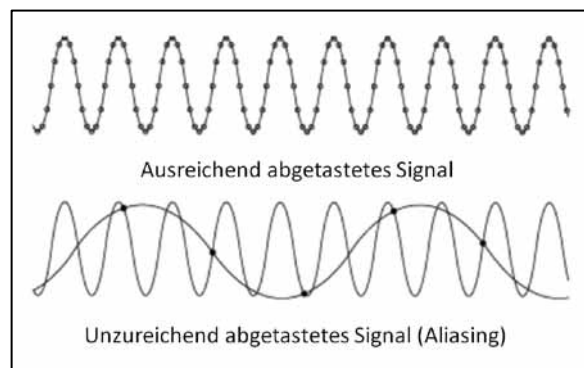


Abbildung 6: Nyquist Aliasing

8 Standardisierung

Für die Durchführung einer Strömungsuntersuchung muss zwingend eine detaillierte Projekt- und Versuchsplanung durchgeführt werden um klar zu definieren, an welchen Stellen gemessen werden muss. Anschließend werden die experimentellen Untersuchungen durchgeführt und parallel kann die Simulation aufgesetzt werden. Da allerdings das Experiment bei der Wahl der Messpunkte weniger flexibel ist wie die Simulation, muss die Generierung der Simulationsdaten nach den Vorgaben des Experiments ablaufen. Abschließend kann die Interpretation und Auswertung erfolgen (Abb. 7).

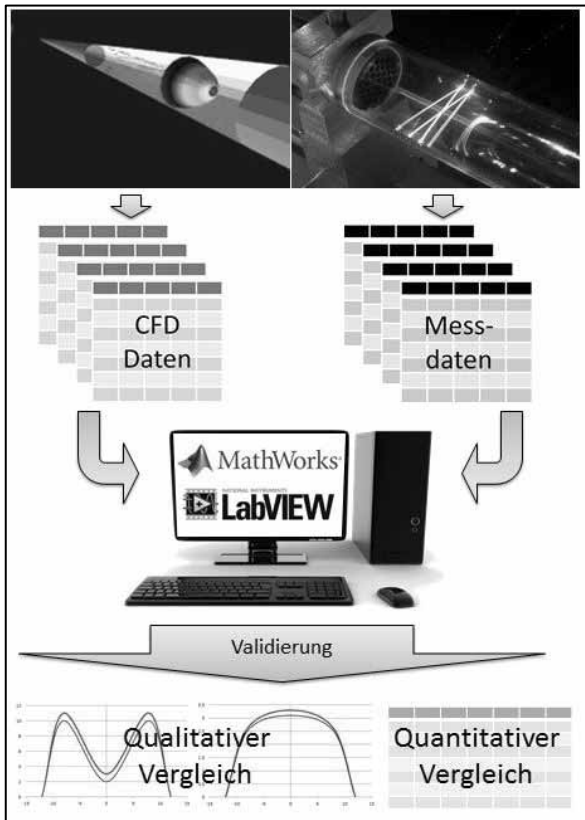


Abbildung 7: Validierungsprozesses

9 Analyse der Datensätze

Die Vorarbeit, um eine CFD-Simulation zeitsparend mit Hilfe experimenteller Daten zu validieren, beginnt mit der Analyse der Datenstrukturen. Neben dem LDV-System kommen eine Reihe weiterer Messgeräte zur Aufnahme der Randbedingungen zum Einsatz. Die Beschaffenheit der einzelnen Datensätze der Messgeräte und der Möglichkeiten der Simulationsauswertung spielen für die Vergleichbarkeit eine entscheidende Rolle. Besonderer Wert muss dabei auf die Messtechnik gelegt werden, da die Berücksichtigung zahlreicher Randbedingungen notwendig ist. (Tab. 2).

Tabelle 1: Übersicht Messgeräte

Gerät	Einheit	Messprinzip	Anzahl
Volumenstromregler	nl/h	Thermisch	5
Differenzdruckmessgerät	Pa	Piezo-elektrisch	3
Absolutdruckmessgerät	Pa	Piezo-elektrisch	1
Luftfeuchtigkeit	%	Kapazitiv	3
Temperatur	°C	Widerstand	3

Das Datenaufkommen und die Anforderungen an die Signalverarbeitung für die Referenzmessgeräte sind geringer als die des LDV-Systems. Für diese Geräte ist eine Aufzeichnung der einzelnen Signale mit 1 Hz ausreichend. Deshalb ist es problemlos möglich diese Daten mittels LabVIEW zu erfassen und zu archivieren.

Das LDV-System hingegen erfordert eine komplexe Signalverarbeitung. Die Messdateien beinhalten die einzelnen Geschwindigkeitssignale und die nicht äquidistante Zeitinformation. Dadurch ist es möglich, Frequenzanalysen durchzuführen und höhere Momente zu berechnen, wie beispielsweise den Turbulenzgrad.

Seitens der Simulation können im transienten Fall die Daten mit der gleichen Struktur als Zeitsignal extrahiert werden. Für stationär gerechnete Strömungen muss die mittlere Strömungsgeschwindigkeit direkt exportiert werden und die Turbulenzintensität mit Hilfe von Feld Functions in der CFD-Software berechnet werden.

10 Automatisierung

Die Aufbereitung der Experimental- und Simulationsdaten wurde mit einem selbstentwickelten MATLAB-Tool realisiert. Da nach der Ausrüstung des Prüfstands mit der entsprechenden Messtechnik und Signalverarbeitung fast alle Logfiles in einem standardisierten Format und Aufbau erfasst werden können, ist eine automatisierte Verarbeitung möglich (Abb. 8)

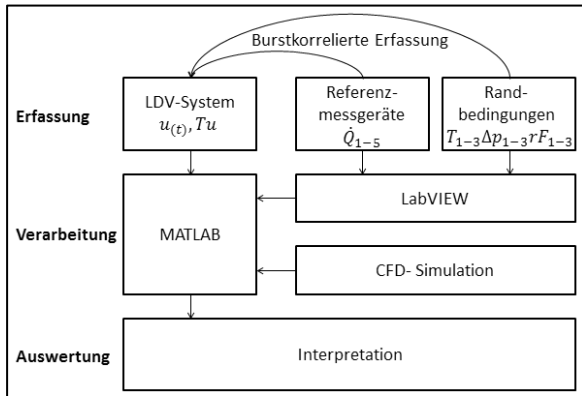


Abbildung 8: Datenverarbeitung

Die einzige Ausnahme bilden die Daten des LDV-Systems. Um auch hier eine einheitliche Struktur nach einer ersten Modifikation der Logfiles zu gewährleisten muss das Zeitsignal der Strömungsgeschwindigkeit aus dem LDV-System ausgelesen werden. Dafür sind bestimmte Parameter notwendig. Dazu zählen beispielsweise die optischen Eigenschaften der Messsonde und die Einstellungen der Filter zur Signalverarbeitung aus der LDV-Software.

In der CFD-Simulation werden Monitorplots definiert und die Geschwindigkeitskomponente in Hauptströmungsrichtung in eine .csv Datei geschrieben, um die Simulationsdaten in gleicher Weise zu generieren.

Die Kernfunktion des Tools ist der Vergleich der Daten und die Berechnung der statistischen Vergleichsgrößen. Dazu werden die Datensätze gesammelt eingelesen, die statistischen Werte aus Experimental- und Simulationsdaten ermittelt und die entsprechenden Strömungsprofile aus den Punktwerten erstellt. Für weiterführende Untersuchungen können anschließend höhere statistische Momente als Vergleichsgrößen berechnet und Frequenzanalysen durchgeführt werden. Die grundlegende Voraussetzung ist dabei die Zusammenführung in einer Software (Abb. 8), um die Daten in einer übersichtlichen Struktur interpretieren zu können (Abb. 9).

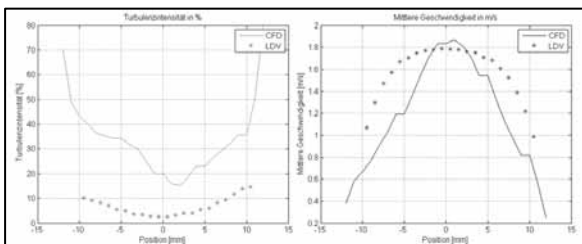


Abbildung 9: Qualitativer Profilvergleich

Um den Vergleich unterschiedlicher Punktabstände in einem Profil zu ermöglichen wurde eine lineare 2-Punkt-Interpolation zur Approximation der Daten angewendet.

11 Zusammenfassung

Die Methodenentwicklung und Automatisierung mittels Makros und anderen Tools ist ein viel diskutiertes Thema im Bereich der CFD. Oft wird hierbei die ganzheitliche Betrachtung von Experiment und Simulation unterschätzt. Der Arbeitsaufwand für die Validierung einer Simulation ist nicht zu vernachlässigen. Inkompatible Datensätze von Experiment und Simulation führen zu einem stark erhöhten Auswertungsaufwand. Wird dieser Arbeitsschritt händisch durchgeführt, ist zusätzlich mit einer höheren Fehlerquote zu rechnen.

Besonders umfangreich ist die Untersuchung von instationären oder turbulenten Strömungen. Dabei müssen räumlich und zeitlich stark schwankende Messgrößen erfasst werden, was zu großen Datenmengen führt. Gleichzeitig zeigt eine Analyse der Datenstruktur, dass diese einfach und wiederholend ist, so dass eine Automatisierung ein gutes Aufwand-Nutzen-Verhältnis aufweist.

Zur Vorbereitung der Auswertungs-Automatisierung wurde der Workflow einer strömungsmechanischen Untersuchung, unter Berücksichtigung der experimentellen und simulationsbedingten Einschränkungen, definiert. Im nächsten Schritt ist die Datenerfassung und -verarbeitung mittels Richtlinien standardisiert worden. Dadurch entstehen unabhängig vom Versuchs- und Berechnungsingenieur einheitliche Datensätze. Eine klar definierte Vorgehensweise ist die Grundvoraussetzung für die erfolgreiche Automatisierung.

Die Herausforderung bei der Aufnahme der Messdaten besteht in der Erzeugung geeigneter Strömungsbedingungen und der Gewährleistung des optischen Zugangs. Experimentelle Untersuchungen stellen hohe Anforderungen an die klar definierte Erzeugung der Strömung und den Versuchsaufbau. Positionierungsungenauigkeiten der Sonde und unbeabsichtigte Störungen der Strömung beeinflussen die Messergebnisse negativ. Um derartige Fehler zu minimieren ist besonders auf diese Faktoren zu achten. Die Messdaten wurden mit einem 1D-LDV-System mit Rückstreuunganordnung aufgenommen, welches es ermöglicht, Strömungen mit minimalen Eingriffen für den optischen Zugang zu untersuchen. Dazu wurde ein

Prüfling in einem Luftprüfstand von einem genau definiertem Volumenstrom mit vorgegebenen strömungsmechanischen Eigenschaften durchströmt. Die Validierung erfolgt über Bilanzflächen, für die Profile einer stationären Strömung (turbulent oder laminar) experimentell und simulativ erfasst wurden.

Die Datenausgabe für die weitere Verarbeitung und Interpretation der Ergebnisse erfolgt zusammengefasst in einer .xls Datei. Außerdem gibt das Tool die entsprechenden Standardgraphen wie mittlere Geschwindigkeit, Turbulenzintensität sowie absolute und prozentuale Abweichungen zwischen Experiment und Simulation aus und speichert diese ab.

Dadurch wird die Genauigkeit von CFD-Simulationen beziffert und bei inakzeptablen Abweichungen können Anpassungen vorgenommen werden. Außerdem können Untersuchungen zeitlich effizienter durchgeführt werden und es wird zur Reduzierung der Fehlerhäufigkeit bei der Auswertung von experimentellen und numerischen Strömungsuntersuchungen beigetragen.

12 Literaturverzeichnis

- Durst, F. (2006). *Grundlagen der Strömungsmechanik*. Berlin: Springer.
- Ferziger, J., & Peric, M. (2002). *Numerische Strömungsmechanik*. Berlin: Springer.
- National Instruments. (2011). *www.ni.com*. Abgerufen am 16. 07 2014 von <http://zone.ni.com/reference/en-XX/help/371361H-01/lvanlsconcepts/aliasing/>
- Oberkampf, W., & Trucano, T. (2002). *Verification and Validation in Computational Fluid Dynamics*. Albuquerque.
- Oertel, H. (1989). *Optische Strömungsmesstechnik*. Karlsruhe: G. Braunsche Hofdruckerei und Verlag.
- Zechmeister, M., & Kürster, M. (2008). *The generalised Lomb-Scargle periodogram*. Heidelberg.

Thermische Simulation von Batteriesystemen mittels CFD sowie deren Vergleich mit einem vereinfachten Simulationsmodell

Bernd Scheiderer¹, Wolfgang Schlüter¹

¹Kompetenzzentrum Strömungssimulation, Hochschule Ansbach
b.scheiderer@hs-ansbach.de

Im vorliegenden Beitrag wird die Modellerstellung einer dreidimensionalen Strömungssimulation (engl.: Computational Fluid Dynamics, CFD) für das thermische Verhalten eines Batteriesystems erläutert. Dafür müssen die physikalischen Zusammenhänge erarbeitet und das zu simulierende Batteriesystem untersucht werden. Für das Design der Batteriezelle wird das Battery Design Studio (BDS) verwendet. Die Batteriezelle wird in das CFD-Modell des verwendeten CFD-Programms STAR-CCM+ übernommen. Innerhalb des CFD-Modells gilt es, die richtigen physikalischen Modelle und eine zweckmäßige Diskretisierung des Systems zu wählen. Ziel der Diskretisierung ist hierbei, eine schnelle Simulation mit dennoch vertretbarer Qualität der Ergebnisse zu ermöglichen.

Weiterhin werden die CFD-Simulationsergebnisse mit Ergebnissen eines vereinfachten Simulationsmodells und mit Messergebnissen verglichen. Als vereinfachtes Simulationsmodell dient ein MATLAB-Modell, welches die Finite-Volumen-Methode (FVM) für die Simulation des Wärmetransports anwendet. Anhand dieses Vergleichs und der Unterschiede der beiden Modelle werden die Vor- und Nachteile der Modelle dargelegt.

1 Einleitung

Die thermische Analyse des Systems mit der Prognose, wie hoch die Temperaturen eines Batteriesystems im Betrieb werden, ist ein wichtiger Schritt in der Entwicklung von Batteriesystemen. Für die Lebensdauer und die Leistungsfähigkeit von Batterien ist deren Temperatur ein sehr wichtiger Faktor. Zudem müssen Lithium-Ionen-Zellen aufgrund von Sicherheitsaspekten unterhalb einer festgelegten Temperatur betrieben werden [1]. Für die Entwicklung von Batteriesystemen mit solchen Zellen und deren Kühlung ist es somit von großer Bedeutung, welche Temperaturen bei verschiedenen Belastungen in solchen Systemen entstehen und wie verschiedene Kühlsysteme die Batterietemperatur beeinflussen. Dies zeigt, wie wichtig die Temperaturproblematik bei der Entwicklung effizienter und damit intelligent gestalteter Batteriesysteme ist.

Mit Hilfe der numerischen Simulation der Wärmetransport- und Strömungsvorgänge in den Batteriezellen und der Batteriekühlung kann eine präzise Analyse der Temperaturverteilung in solchen Systemen erfolgen. Dies ermöglicht die Aufdeckung von Hotspots sowie Engstellen in der Strömungsführung der Kühlgeometrie. Zudem können Kausalitäten visualisiert und analysiert werden, die messtechnisch nur mit sehr hohem Aufwand ermittelbar sind.

In diesem Beitrag wird die Erstellung eines CFD-Modells für die thermische Simulation von Batteriesystemen erläutert. Die Simulationsergebnisse dieses Modells werden zudem mit den Ergebnissen eines vereinfachten Simulationsmodells und realen Messwerten verglichen. Damit kann überprüft werden, in wie weit die Simulationsergebnisse der beiden Modelle mit der Wirklichkeit übereinstimmen. Auch die Vorteile des jeweiligen Modells lassen sich anhand des Vergleichs herausarbeiten.

2 Systemanalyse

Um die Ergebnisse der beiden Modelle mit den Messwerten vergleichen zu können gilt es, das Zelldesign, das Design des Batteriesystems und auch die Randbedingungen in beiden Modellen mit denen der Messung möglichst identisch zu gestalten.

2.1 Reales Batteriesystem

Das vermessene Batteriesystem besteht aus acht elektrisch in Reihe geschalteten Lithium-Ionen-Zellen. Die Zellen des Batteriesystems besitzen eine prismatische Form und sind als Traktionsbatterien für Fahrzeuge konzipiert. Mit 25 Ah Kapazität und einer Leistung über 800 W sind die Zellen wesentlich leistungsfähiger als herkömmliche Batteriezellen für den Betrieb von Geräten der Unterhaltungselektronik. Aber auch ihre Abmessungen unterscheiden sich deutlich (Tab. 1).

Kapazität [Ah]	Leistung [W]	Höhe [mm]	Breite [mm]	Dicke [mm]
25	≥ 800	91	148	26,5

Tabelle 1: Nenndaten und Maße der Batteriezellen

Die acht Lithium-Ionen-Zellen sind geometrisch in Reihe angeordnet. Unterhalb der Zellen ist ein Kühlkanal angebracht (Abb. 1). Die darin zirkulierende Wasser-Glykol-Mischung kühlt die Zellen während des Betriebs.

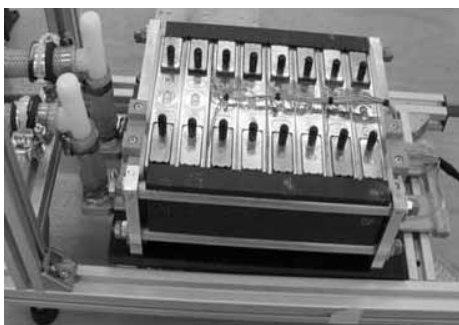


Abbildung 1: Batteriesystem mit acht Zellen in Reihe

2.2 Messung am realen Batteriesystem

Der pulsformige Lade- und Entladestrom betrug bei der Messung 90 A. Die zyklische Ladung bzw. Entladung erfolgte im Zellspannungsbereich von 3,0 V bis 4,1 V. Dies entspricht nahezu einem SOC-Hub von 100 %. Die Messung wurde innerhalb einer Klimakammer durchgeführt. Dies ermöglichte es, eine konstante Umgebungstemperatur von 30 °C sicherzustellen. Für das Kühlmedium wurden eine Temperatur von 25 °C und ein Volumenstrom von 2,5 L/min vorgegeben. Beendet war die Messung, als sich die lokalen Temperaturmaxima bei etwa 57 °C stabilisierten (Abb. 2).

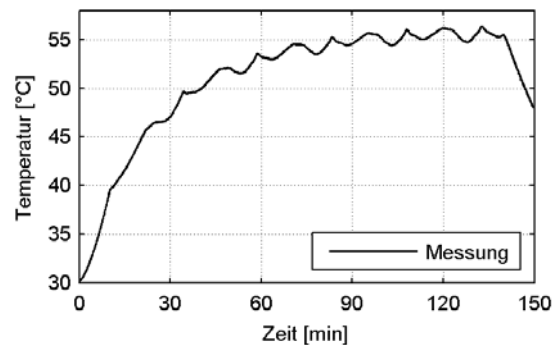


Abbildung 2: Temperaturverlauf der Messung

3 FVM-Modell

Eine Batteriezelle besteht im FVM-Modell aus einem elektrischen Zellmodell, einem thermischen Zellmodell sowie aus dem Kühlmodell. Die thermische Verlustleistung für das thermische Zellmodell wird mit dem elektrischen Zellmodell berechnet, dieses wird vom Zellhersteller in Form eines MATLAB/Simulink-Bocks zur Verfügung gestellt. Aufgrund dieser Gegebenheit, wurde das MATLAB-Modell in die MATLAB/Simulink-Umgebung eingebettet.

Zur Simulation der Temperaturen innerhalb der Zelle wird im thermischen Zellmodell ein Finite-Volumen-Ansatz verwendet. Nach der Einteilung einer Zelle in eine endliche Zahl von Volumenelementen wird mithilfe einer Wärmebilanz für jedes einzelne Volumenelement der Wärmefluss von diesem Volumenelement zu seinen benachbarten Elementen

berechnet. Dieses Verfahren hat den Vorteil, dass in der Simulation direkt der Wärmefluss \dot{Q} zwischen den Zellen berechnet wird (Gl. 1) und nicht die Fourier'sche Differenzialgleichung (Gl. 2) gelöst werden muss.

$$\dot{Q} = \frac{dQ}{dt} = \lambda \cdot \frac{T_2 - T_1}{\Delta x} \cdot A \quad (1)$$

Die Lösung der Fourier'sche Differenzialgleichung ist wesentlich aufwändiger, da sie eine partielle Differentialgleichung für das Temperaturfeld $T(x, y, z, t)$ ist.

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(\lambda \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(\lambda \frac{\partial T}{\partial z} \right) + \dot{w} \quad (2)$$

Das Modell der Kühlung ist so konzipiert, dass die Kombination aus Zellmodell und entsprechendem Modell der Kühlung beliebig oft hintereinander geschaltet werden kann, um eine Batterie aus vielen Zellen zu bilden. Um den Teil des Gesamtkühlkanals zu modellieren, der Einfluss auf eine einzelne Zelle nimmt, wird für die Kühlung des Zellmodells nur der Abschnitt des Kühlkanals modelliert, der sich direkt unter einer Zelle befindet. Für dieses Teilstück werden im Modell die Erhöhung der Temperatur des Kühlmediums und die damit verbundene, vom Kühlmedium aufgenommene, Kühlleistung berechnet. Der Wärmeübergang vom Kühlkanal in das Kühlmedium wird durch das physikalische Phänomen der erzwungenen Konvektion beschrieben. Zur Berechnung der konvektiven Wärmeübertragung (Gl. 3) ist der Wärmeübergangskoeffizient α zu bestimmen.

$$\frac{\dot{Q}}{A} = \dot{q} = \alpha \cdot (T_W - T_\infty) \quad (3)$$

Der Wärmeübergangskoeffizient α kann mithilfe der Ähnlichkeitstheorie, speziell mit der Nußelt-Zahl Nu bestimmt werden (Gl. 4).

$$Nu = \frac{\alpha \cdot L}{\lambda} = \frac{(\dot{q}/\Delta T)_{konv}}{(\dot{q}/\Delta T)_{WL}} \quad (4)$$

Nach der Vorgabe aller Parameter der Kühlung gibt das Modell der Kühlung die Kühlleistung und die

Temperatur des Kühlmediums nach der jeweiligen Batteriezelle aus.

Das Modell für den Zellenstack mit acht Zellen wurde durch acht hintereinander geschaltete Einzelzellmodelle realisiert. Im Gesamtbatteriemodell sind die einzelnen Zellenmodelle ausschließlich über das Kühlmedium verbunden. Die berechnete Austrittstemperatur des Kühlmediums vom vorhergehenden Kühlkanalmodell gibt dabei die Eintrittstemperatur des Kühlmediums in das darauffolgende Modell vor.

4 CFD-Modell

Die Modellierung des Systems in der CFD-Software STAR-CCM+ gliedert sich in die Bereiche: Geometrie- und Physikmodellierung sowie die Vernetzung des Simulationsgebiets. Besonders die Wahl der physikalischen Modelle sowie eine zweckmäßige Vernetzung sind für gute Simulationsergebnisse wichtig [2].

4.1 Geometrie- und Physikmodellierung

In den Batteriezellen entstehen thermische Verluste, die von der Kühlung des Batteriesystems weg transportiert werden müssen. Diese thermischen Verluste werden im Zellmodell berechnet. Die Batteriezelle des Batteriesystems wird nicht in STAR-CCM+ modelliert, sondern im Battery Design Studio (BDS). Das BDS ist ein auf die Modellierung von Batteriezellen spezialisiertes Programm. Die im BDS modellierte Einzelzelle und der Lastzyklus des zu simulierenden Batteriesystems werden anschließend in STAR-CCM+ importiert. In STAR-CCM+ wird das Batteriesystem auf Basis dieser Einzelzelle modelliert.

Die Informationen zu den genauen Maßen der Anode, der Kathode, des Separators und der jeweiligen Verbindungselektroden liegen für die vermessene Zelle nicht vor. Bei der Zellmodellierung ist eine intelligente Anpassung der Parameter einer im BDS vorhandenen Lithium-Ionen-Zelle nötig um, ohne alle Parameter der zu simulierenden Batteriezelle zu kennen, diese näherungsweise modellieren zu können.

Das in STAR-CCM+ integrierte Battery Simulation

Module (BSM) dient der Erstellung von gesamten Batteriesystemen. Nach dem Import des im BDS erstellten Zellenmodells und des Lastprofils, wird die Anzahl der seriellen und parallelen Zellen im Zellverbund vorgegeben. Zusammen mit der geometrischen Modellierung der Kühlung sind damit die Abmessungen des Batteriesystems definiert.

Physikalisch müssen die Materialien und zu Grunde liegenden Simulationsmodelle definiert werden. STAR-CCM+ enthält für die Batteriesimulation das speziell hierfür bereitgestellte Battery-Modell. Das Modell der Kühlung besteht einerseits aus dem Aluminiumkanal und andererseits aus der flüssigen Wasser-Glykol-Mischung darin. Damit die Kühlung in der Simulation der Messung entspricht, werden der Volumenstrom und die Temperatur am Vorlauf vorgegeben. Zur Abbildung der Turbulenz im Fluid wurde das $k-\omega$ -SST Turbulenzmodell eingesetzt.

4.2 Vernetzung des Simulationsgebiets

Die zweckmäßige Vernetzung ist maßgeblich für eine schnelle Simulation mit hinreichender Qualität der Ergebnisse. Sind die Strömungsgebiete von komplexer geometrischer Form wie das zu simulierende Batteriesystem, sind Polyedernetze zu bevorzugen. Diese bieten Vorteile hinsichtlich des Konvergenzverhaltens und damit der Simulationszeit [3].

Da die erzeugte Geometrie der Kühlung ein äußerst grobes Oberflächennetz aufweist, ist es sinnvoll dieses vor der volumetrischen Vernetzung zu optimieren. STAR-CCM+ stellt hierfür den sogenannten "Surface Remesher" zur Verfügung, der die ausgewählten Oberflächen mit einem Netz aus Dreiecken der gewünschten Größe versieht. Für die volumetrische Vernetzung der Rechengebiete der Kühlung wird wegen der genannten Vorteile ein polyederförmiges Netz gewählt. Die wandnahen Gebiete werden zur besseren Auflösung mit Prismenschichten (sog. Prismalayer in STAR-CCM+) versehen.

Batteriezellen bestehen mit der Anode, der Kathode, dem Separator, dem Elektrolyt und dem Aluminiummantel aus einer großen Anzahl verschiedener Komponenten. Aufgrund der sich stark unterscheidenden Geometrien und Materialien der verschiedenen Komponenten stellen diese jeweils andere Anforderungen an deren Diskretisierung. Um ein Netz

mit guter Qualität für die gesamte Batteriezelle zu erzeugen, stellt STAR-CCM+ ein spezielles Vernetzungsmodell, den "Battery Mesher", zur Verfügung. Mit diesem Vernetzungsmodell ist es möglich dafür geeignete Bereiche mit einem Prismennetz zu diskretisieren. Die restlichen Bereiche der Batteriezellen werden hingegen mit Polyedernetzen diskretisiert. So ist es möglich die Vorteile der beiden Netzarten für die Diskretisierung der Batteriezellen einzusetzen. Ein Prismennetz eignet sich besonders für die Diskretisierung von dünnen Strukturen. Der Aufbau einer Batteriezelle hat mit den enthaltenen Kontakten und dem Zellgehäuse solche dünnen Strukturen. Weiterhin werden durch den „Battery Mesher“ automatisch regional verfeinerte Netze ("Volumetric Controls") für die besonders feinen Strukturen der Batteriezellen erzeugt. Die Verwendung des „Battery Meshers“ erfordert die Optimierung des Oberflächennetzes [4], deshalb wird auch bei der Vernetzung der Batterie der "Surface Remesher" angewendet.

Die Qualität der Vernetzung wurde hinsichtlich der Zellqualität geprüft und mit einer Netzstudie wurde der Einfluss der Diskretisierung auf die Simulationsergebnisse untersucht. Die Zellqualität ist ein Maß für die geometrische Abweichung einer Zelle von einer ideal gleichmäßigen Zelle. Eine niedrige Zellqualität kann in strömungstechnisch anspruchsvollen Regionen zu numerischen Fehlern und mitunter zu schlechtem Konvergenzverhalten führen. In Regionen mit niedrigen strömungstechnischen Gradienten sind schlechte Zellen meistens hinnehmbar.

In der Netzstudie wurde deutlich, dass die grösste Diskretisierung (Zellgröße 100 %) nicht für die Simulation geeignet ist, da hier die simulierten Temperaturen gegenüber der Temperaturen der feinsten untersuchten Diskretisierung (Zellgröße 12,5 %) extrem abweichen. Bei einer Zellgröße von 50 % reduzieren sich die Abweichungen der Temperaturen bereits auf ein akzeptables Niveau und die Simulationszeit bleibt um den Faktor 6 niedriger als bei der Verwendung der Zellgröße von 12,5 %. Somit bietet die Zellgröße von 50 % des Ausgangswertes einen guten Kompromiss zwischen Genauigkeit und niedriger Rechenzeit und wurde daher zur Durchführung der Simulation verwendet.

5 Vergleich der Modelle

Um die Ergebnisqualität des CFD- und FVM-Modells zu vergleichen, werden deren Simulationsergebnisse mit der beschriebenen Messung gegenübergestellt. Weiterhin werden die Vor- und Nachteile der beiden Modelle erarbeitet und dargestellt.

5.1 Übereinstimmung mit den Messwerten

Die Ergebnisse werden hinsichtlich der Temperaturmittelwerte und der Temperaturspreizung innerhalb der Batteriezellen verglichen. Der Vergleich der Temperaturmittelwerte zeigt, dass das CFD-Modell den Temperaturverlauf zu Beginn der Messung weniger gut abbildet als das FVM Modell (Abb. 3).

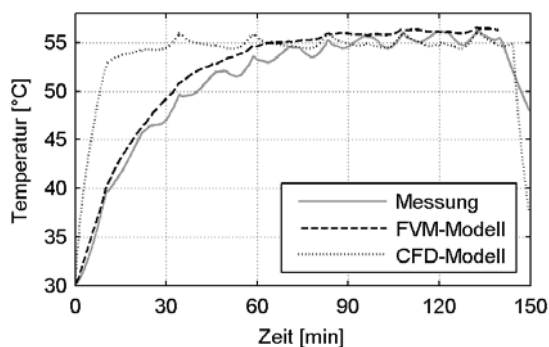


Abbildung 3: Vergleich der mittleren Zelltemperaturen der Modelle und Messung

Vorteile hat das CFD-Modell hingegen bei der Abbildung der lokalen Temperaturmaxima, die nach jedem Entlade- und Ladevorgang auftreten (Abb. 4).

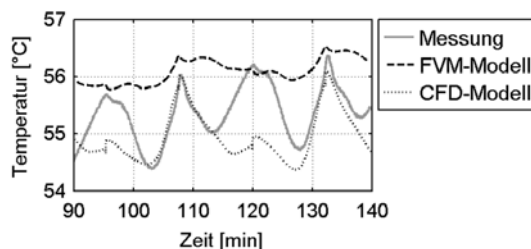


Abbildung 4: Vergleich der simulierten und gemessenen Temperaturspitzen

Auch bei der simulierten Temperaturspreizung in der Höhe der Batteriezellen liefert das CFD-Modell bessere Ergebnisse als das FVM-Modell. Die simulierten Temperaturen des CFD-Modells weichen maximal ein Grad von den gemessenen Temperaturen ab. Die Abweichung der simulierten Temperaturen des FVM-Modells ist im Gegensatz dazu mit bis zu drei Grad deutlich größer (Abb. 5).

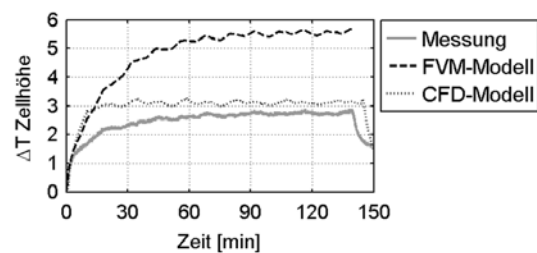


Abbildung 5: Vergleich der simulierten und gemessenen Temperaturspreizung in der Höhe der Batteriezellen

Hinsichtlich der Temperaturspreizung in der Breite der Zelle zeigt sich hingegen eine andere Tendenz. Das FVM-Modell weist hier eine signifikant größere Übereinstimmung mit den Messwerten auf als das CFD-Modell (Abb. 6).

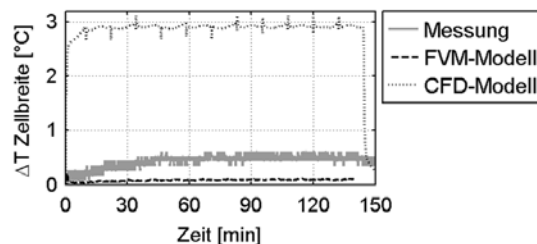


Abbildung 6: Vergleich der simulierten und gemessenen Temperaturspreizung in der Breite der Batteriezellen

Zusätzlich zu diesen Vergleichen, welche die Temperaturen am Gehäuse der Batteriezellen betrafen, wurde die simulierte und gemessene Rücklauftemperatur des Kühlmediums gegenübergestellt. Die Simulationsergebnisse der Rücklauftemperatur sind bei beiden Modellen nahezu identisch mit den Messwerten (Abb. 7). Aus diesem Vergleich kann somit keine Präferenz für ein Modell abgeleitet werden.

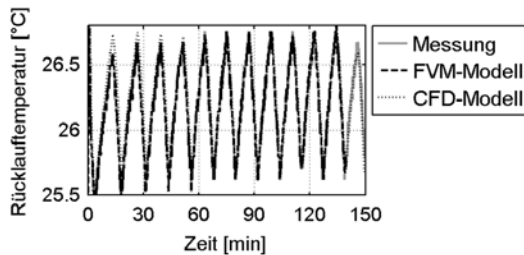


Abbildung 7: Vergleich der simulierten und gemessenen Rücklaufftemperaturen

5.2 Bewertung von CFD-Modell und FVM-Modell

Ein erheblicher Nachteil des FVM-Modells stellt die Tatsache dar, dass zur Berechnung der thermischen Verlustleistung der Batteriezellen ein Zellmodell des Zellherstellers erforderlich ist. Ohne ein solches Modell ist das FVM-Modell in der beschriebenen Form nutzlos. Das CFD-Modell ist auch, ohne die Batteriezelle in allen Details zu kennen, in der Lage, die maximale Temperaturhöhe mit annehmbarer Genauigkeit zu simulieren.

Weiterhin bietet das CFD-Modell wesentlich mehr Möglichkeiten in der Ergebnisdarstellung als das FVM-Modell. Durch die wesentlich höhere Auflösung der Geometrie im CFD-Modell stellt dieses das Batteriesystem exakt dar. Der Verlauf der Oberflächentemperaturen der Zellen (Abb. 8) oder des Gesamtsystems können zudem direkt in der Simulation betrachtet und als Video abgespeichert werden.

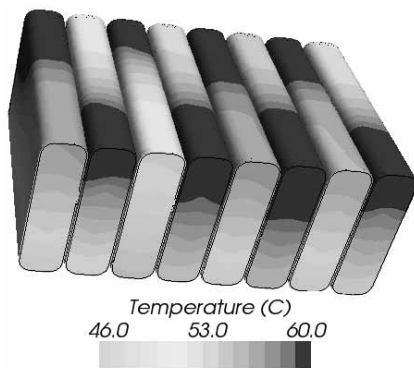


Abbildung 8: Oberflächentemperaturen der Batteriezellen während der Simulation

Im Gegensatz zum FVM-Modell, das auf ein Kühlkonzept festgelegt ist, kann im CFD-Modell eine gänzlich andere Kühlungsvariante mit relativ wenig Aufwand umgesetzt werden. Die Simulation der Strömung des Kühlmediums erlaubt zudem eine qualitative Analyse der Strömung und damit die Optimierung der Kühlung hinsichtlich der Strömungsführung.

Die Gegenüberstellung der Simulationsgeschwindigkeit der beiden Modelle zeigt erwartungsgemäß erhebliche Vorteile für das FVM-Modell. Dieses benötigt an einem normalen Rechner für die Simulation des Gesamtbatteriesystems nur die 0,8-fache Zeit der realen Zeit. Das CFD-Modell benötigt auf einem HPC-Cluster mit 20 parallelen Threads die 11-fache Zeit der realen Zeit. Dieser Vergleich ist jedoch nur bedingt aussagekräftig, da im CFD-Modell 130377 Volumenelemente berechnet wurden, wohingegen im FVM-Modell lediglich 672 Volumenelemente und somit Temperaturwerte simuliert worden sind. Wird die Simulationszeit auf ein Volumenelement normiert, ist die CFD-Simulation um den Faktor 50 schneller als die FVM-Simulation. Es gilt auch zu bedenken, dass für die Strömungssimulation spezielle CFD-Software und zur Batteriesimulation noch weitere kostenpflichtige Lizenzen für das BDS und BSM erforderlich sind.

Der Vergleich der Simulationsergebnisse mit den Messwerten zeigte zudem, dass das CFD-Modell den Temperaturverlauf des Batteriesystems zu Beginn weniger gut nachbilden konnte als das FVM-Modell. Dafür konnte das CFD-Modell bei der Abbildung der lokalen Temperaturmaxima dessen höhere Genauigkeit zeigen. Somit spricht der Vergleich der simulierten Temperaturen mit den Messwerten nicht ausnahmslos für eines der beiden Modelle.

In Summe bietet das CFD-Modell mehr Vorteile als Nachteile gegenüber dem FVM-Modell. Die teils erheblichen Nachteile, wie beispielsweise die wesentlich höhere Simulationszeit, können aber je nach Anwendungsfall für die Verwendung des FVM-Modells sprechen. Es ist also abhängig vom jeweiligen Anwendungsfall das eine oder das andere Modell zu bevorzugen. Das FVM-Modell ist aufgrund der kürzeren Rechenzeit besser für einfache Analysen geeignet. Soll hingegen die Kühlung, insbesondere die Strömungsführung, optimiert werden, ist die Verwendung eines CFD-Modells unausweichlich.

6 Zusammenfassung

Im vorliegenden Beitrag ist die Erstellung eines CFD-Modells für das thermische Verhalten eines Batteriesystems beschrieben. Neben der für die Umsetzung der CFD-Simulation wichtigen Analyse des zu simulierenden Batteriesystems wurden die Modellierung des Systems sowie der Vergleich der CFD-Simulationsergebnisse mit den FVM-Simulationsergebnissen und den Messergebnissen erläutert.

Bei der Zellmodellierung ist eine intelligente Anpassung der Parameter einer im BDS vorhandenen Lithium-Ionen-Zelle nötig, um auch ohne alle Parameter der zu simulierenden Batteriezelle zu kennen, diese näherungsweise modellieren zu können. Innerhalb des CFD-Modells ist besonders auf die physikalischen Modelle und eine zweckmäßige Vernetzung zu achten. Diese sind maßgeblich für eine schnelle Simulation mit dennoch vertretbarer Qualität der Ergebnisse. Anhand der Untersuchungen der Diskretisierung konnte die Simulationszeit des CFD-Modells bei nahezu gleichbleibender Ergebnisqualität um den Faktor 6 reduziert werden.

Im Vergleich der CFD-Simulationsergebnisse mit den FVM-Simulationsergebnissen und den Messergebnissen konnte festgestellt werden, dass das CFD-Modell bei der Abbildung des Temperaturverlaufs zu Beginn der Messung größere Abweichungen zeigt als das FVM-Modell. Vorteile hat das CFD-Modell hingegen bei der Simulation der lokalen Temperaturmaxima, die nach jedem Entlade- und Ladevorgang auftreten. Wie nicht anders zu erwarten, benötigt das FVM-Modell wesentlich weniger Rechenzeit. Die Gegenüberstellung der Bewertung beider Modelle zeigte, dass abhängig vom Anwendungsfall das eine oder das andere Modell zu bevorzugen ist.

References

- [1] J. M. Tarascon und M. Armand. *Issues and challenges facing rechargeable lithium batteries*. Nature, 414(6861):359367, 2001.
- [2] M. Casey und T. Wintersgerste. *ERCOFTAC Special Interest Group on Quality and Trust in Industrial CFD-Best Practice Guidelines*. European Research Community on Flow, Turbulence and Combustion, 2000.
- [3] M. Perić und S. Ferguson. *The advantage of polyhedral meshes*. CD-adapco, 2007.
- [4] CD-adapco Team. *User Guide STAR-CCM+ 8.02.011*. 2013.

Untersuchung der Skalierbarkeit von Batteriemanagementsystemen mithilfe von CAE-Simulation

Matthias Roch^{1,2}, Robert Feind¹, Florian Quantmeyer¹, Xiaobo Liu-Henke¹

¹Ostfalia Hochschule für angewandte Wissenschaften
38302 Wolfenbüttel

²*mat.roch@ostfalia.de*

Zusammenfassung. Moderne elektromobile Anwendungen profilieren sich maßgeblich durch ihre Energiespeicher und der daraus resultierenden Reichweite. Weit verbreitet sind Lithium-Ionen-Zellen. Zur Steigerung der Kapazität, zum Monitoring und zur Überwachung von Batteriepacks, die aus mehreren Zellen zusammengesetzt sind, werden Batteriemanagementsysteme (BMS) eingesetzt. An der Ostfalia Hochschule wird ein BMS mit dezentraler Topologie modellbasiert entwickelt und erprobt. Die dezentralen Zellmodule kommunizieren per I²C-Bus mit dem Mastercontroller, welcher übergeordnete Regelungsfunktionen übernimmt und Stellsignale wiederum über den Bus an die Zellen sendet. Die Verwendung eines Bussystems ermöglicht eine modulare, einfach erweiterbare Bauweise. Aufgrund der Buskommunikation tritt jedoch Latenz von Nachrichten auf, die mit zunehmender Zellenanzahl steigt. Das vorliegende Paper thematisiert die Untersuchung der Skalierbarkeit des typischen eingebetteten Systems BMS. Es werden die Latenzzeiten bei erhöhter Zellenanzahl sowie deren Einfluss auf die Regelungsfunktionen und Schätzalgorithmen des BMS-Controllers betrachtet. Weiterhin werden die Effekte infolge veränderlicher Auflösungen von Sensorsignalen untersucht.

1 Einleitung

Das Thema Elektromobilität ist längst zu einem omnipräsenten Thema avanciert. Eine entscheidende Rolle, ob und wie schnell sie sich gegen konventionell angetriebene Fahrzeuge durchsetzen kann, spielt die weitere Entwicklung des Energiespeichers, der Batterie. Ein wichtiger Schritt lag in der Verwendung von Lithiumbatteriezellen, mit denen aufgrund höherer Energiedichte die Gesamtmasse des Batteriepacks gegenüber beispielsweise Bleibatterien drastisch reduziert werden kann.

Dadurch steigen allerdings auch die Anforderungen an das Batteriemanagementsystem (BMS), das Steuergerät der Batterie, erheblich: Wie bei der Verwendung anderer Zelltypen werden mehrere Lithiumzellen in Reihe zu einem Batteriepack verschaltet, um eine höhere Spannung zu erhalten. Beim Verlassen des Arbeitsbereiches können die Lithiumzellen allerdings beschädigt werden. Kleinste

fertigungsbedingte Unterschiede in den Batterieparametern provozieren im Betrieb zwangsläufig unterschiedliche Ladestände. Moderne Batteriemanagementsysteme überwachen jede einzelne Zelle des Batteriesystems und erlauben somit einen sicheren Betrieb. Des Weiteren tragen sie zu einer besseren Ausnutzung der Batterie bei, indem sie bei unterschiedlichen Ladeständen der einzelnen Zellen den Ladeausgleich steuern. Weiterhin sagen sie zuverlässig die Restladungsmenge voraus. Dazu umfasst das BMS neben dem eigentlichen technischen System verschiedene Sensoren, ein Computersystem und Aktoren. Es handelt sich demnach um ein typisches eingebettetes System. Zur Beherrschung der Systemkomplexität wird der durchgehend modellbasierte, verifikationsorientierte Entwicklungsprozess angewendet. Dieser Prozess kann derzeit durch entsprechende sehr teure CAE-Werkzeuge in weiten Teilen unterstützt werden.

Im Rahmen eines EFRE¹-geförderten Projektes werden an der Ostfalia Hochschule für angewandte Wissenschaften Low-Cost Open-Source Entwicklungsplattformen für BMS untersucht, um eine Alternative zu teuren Entwicklungstools (z.B.dSPACE) und auf dem Markt erhältlichen Black-Box-Steuergeräten zu bieten. Details dazu können [1] entnommen werden. Das hier vorgestellte Arbeitspaket beinhaltet die modellbasierte Untersuchung der Skalierbarkeit eines prototypischen BMS. Skalierbarkeit beschreibt dabei die Möglichkeit, die Anzahl der Zellen zu erhöhen und das System so zu erweitern. Da eine höhere Anzahl von Zellmodulen für erhöhte Latenzen bei der Messwerterfassung sorgt, soll dieser Einfluss in einer Simulation untersucht werden. Weiterhin lassen sich aus dieser Simulation die Anforderungen an die Hardwarekonfiguration in Bezug auf das Bussystem im Allgemeinen und die Auflösung der Messgrößen ermitteln.

2 Batteriemanagementsystem

Dieser Abschnitt befasst sich mit der Beschreibung des Realsystems BMS. Es werden die Funktionen erläutert, bevor auf die hardwareseitige Umsetzung eingegangen wird.

2.1 Funktionen des BMS

Die Funktionen des BMS lassen sich in fünf Kategorien unterteilen, siehe Abbildung 1.

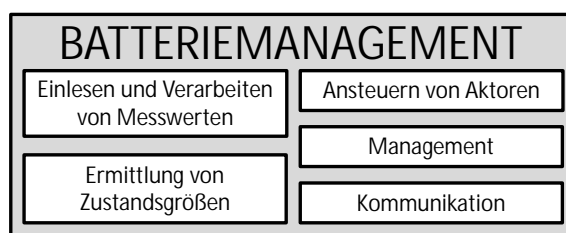


Abbildung 1: Funktionskategorien des BMS

Das BMS überwacht permanent die Zustände des Systems. Dazu gehört neben der Messung von Gesamtspannung und Gesamtstrom auch die Messung jeder Zellspannung und etwaiger Strangströme.

¹EFRE: Europäischer Fonds für regionale Entwicklung

Außerdem werden die Temperaturen an den Zellen überwacht. Das Einlesen und Verarbeiten dieser Messwerte bildet die Grundlage für alle Algorithmen des BMS.

Da nicht alle relevanten Größen, wie zum Beispiel der aktuelle Ladezustand (*SoC: State of Charge*), direkt erfasst werden können, müssen diese aus den vorhandenen Messwerten berechnet werden. Neben dem SoC ist hier die Schätzung des Alterungszustandes (*SoH: State of Health*) und die Leistungsschätzung zu nennen.

Je nach Betriebszustand (Entladevorgang, Ladevorgang, Ladeausgleich, etc.) wird der jeweilige Vorgang durch das Management überwacht. Aus den vorhandenen gemessenen und ermittelten Daten prüft es, ob die Grenzen des Arbeitsbereiches eingehalten werden. Im Fall eines Fehlers greift das Management ein, um eine Gefährdung des Benutzers oder eine Beschädigung der Batterie auszuschließen.

Als weitere Funktion soll der Ladeausgleich (*Balancing*) hervorgehoben werden. Durch kleinste fertigungsbedingte Unterschiede in den Batterieparametern resultieren im Betrieb zwangsläufig unterschiedliche Ladestände. Da die Batterie stets nur so stark belastet bzw. geladen werden darf, bis eine Zelle die obere oder untere Grenzspannung erreicht, kann die Kapazität des gesamten Packs nie komplett genutzt werden. Von einem Belastungszyklus zum nächsten werden die Unterschiede zwischen den Ladeständen der einzelnen Zellen größer und die abrufbare Energiemenge drastisch reduziert. Beim passiven Ladeausgleich wird im Ladebetrieb gezielt Energie höher geladener Zellen an einem Widerstand dissipiert, sodass alle Zellen voll geladen werden können. Eine komplexere Möglichkeit ist der aktive Ladeausgleich, bei dem Ladungen einer Zelle in eine andere übertragen werden, was einen besseren Wirkungsgrad zur Folge hat.

Dazu muss ein BMS entsprechende Aktoren ansteuern können: Zur Steuerung der Ladeausgleich-Ströme sind Leistungstransistoren erforderlich. Ferner muss das Batteriepack über ein Schütz von der Belastungseinheit bzw. dem Ladegerät getrennt werden können.

Weiterhin verfügt das BMS über eine Kommunikationsschnittstelle, um Informationen mit anderen Systemen austauschen zu können. Dieser Informationsaustausch ist erforderlich, da das BMS die Bat-

terieleistung nicht aktiv regeln kann. Dies geschieht durch das überlagerte Energiemanagement. Es koordiniert die elektrischen Leistungen der Verbraucher, so dass die Betriebsgrenzen der Batterie nicht verletzt werden. Dazu benötigt es die Informationen vom BMS zur maximal entnehmbaren Leistung. Die Kommunikation erfolgt über eine CAN-Schnittstelle.

2.2 Hardware-Plattform des BMS

Es sind verschiedene Hardwarestrukturen möglich, die eine Erfüllung dieser Funktionen erlauben. Hier wurde eine teildezentralisierte Lösung mit Zellmodulen und zentralem BMS-Controller gewählt, siehe Abbildung 2.

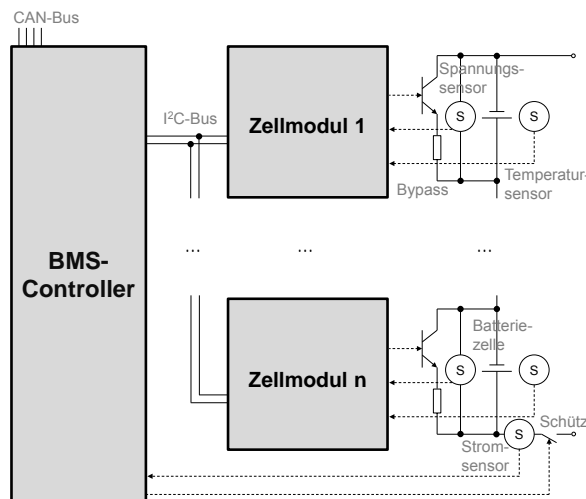


Abbildung 2: Struktur der BMS-Hardware

Hierbei werden die Funktionen verteilt ausgeführt: Die Zellmodule erfassen die lokalen Messgrößen und steuern die Transistoren für den Ladeausgleich an. Die Diagnose- und Managementfunktionen werden auf dem zentralen, leistungsstarken BMS-Controller ausgeführt, der die Messwerte in digitalisierter Form per I²C-Bus bezieht. Nach dem Ausführen der Regelalgorithmen gibt der Controller PWM-Tastgrade per Bus an die jeweiligen Zellmodule zurück. Mit den Tastgraden steuern die Mikrocontroller einen MOSFET an, der als Aktor des passiven Ladeausgleichs die Dissipation überschüssige Energie über den Bypass ermöglicht.

Der große Vorteil dieser denzentralisierten Architek-

tur liegt im geringen Aufwand der Erweiterung. Weitere Zellen lassen sich über weitere Zellmodule einfach an den I²C-Bus anbinden, über den die Kommunikation zwischen den Zellmodulen und dem Controller erfolgt.

Der Kommunikationsumfang pro Zelle ist dabei konstant und der Abbildung 3 zu entnehmen.

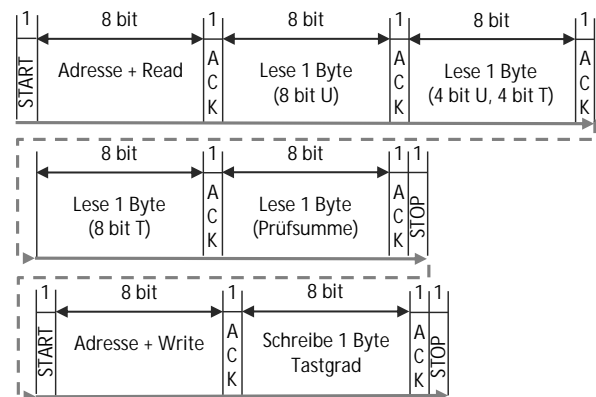


Abbildung 3: Zusammensetzung der I²C-Kommunikation

Die Kommunikation erfolgt byteweise und geht vom Master (= BMS-Controller) aus. Er beginnt die Übertragung mit dem Ausgeben eines Startbits, wodurch die Slave-Module aktiviert werden, und mit dem Senden der Adresse (7 bit) des gewünschten Slaves. Das letzte Bit dieses Bytes enthält die Information, ob der Master Daten aus dem Lesebuffer beziehen (1) oder in den Schreibpuffer übergeben (0) möchte. Ist einem Slave die entsprechende Adresse zugewiesen, bestätigt dieser per Acknowledge (ACK)-Bit. Die übrigen Slaves gehen in den Ruhezustand über. Empfängt der Slave einen Schreibbefehl, erwartet der Slave das Datenbyte und bestätigt den Empfang mittels ACK. Im Falle eines Lese-Befehls geht die Übertragung vom Slave aus und der Master bestätigt per ACK. Durch das Senden der Stop-Bedingung beendet der Master die Übertragung, wodurch der Slave in den Ruhezustand übergeht.

Bei einer Übertragungsrate von 100 kbit/s ergibt sich eine Übertragungsdauer pro Zelle von 0,67 ms. Die Kommunikation erfolgt nacheinander nach gleichem Schema für alle angeschlossenen Zellen, was eine Latenz zwischen Lesewerten einer Zelle zur Folge hat. Der Einfluss dieser Latenz auf die Regelalgorithmen des Controllers wird mithilfe einer Simulation unter-

sucht, woraus sich Aussagen zur Skalierbarkeit ziehen lassen.

3 Aufbau des Simulationsmodells

Das Simulationsmodell des Batteriesystems setzt sich analog zum Realsystem aus Controller, Batteriezellen mit Zellmodulen und einer Belastungseinheit oder einem Ladegerät zusammen. Abbildung 4 verdeutlicht den Aufbau. Das Modell verfügt über mehrere Batteriezellen, um Parameterstreuung und unterschiedliche SoCs nachbilden und damit die Balancing-Funktion untersuchen zu können. Die für die Untersuchung wesentlichen Bestandteile werden in der Folge vorgestellt.

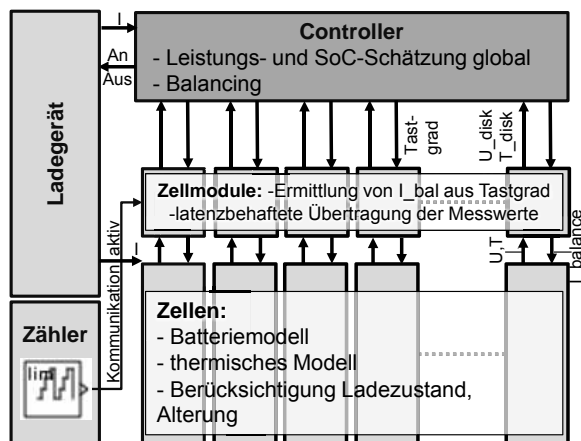


Abbildung 4: Vereinfachter Aufbau des BMS-Modells

3.1 Modellierung der Batteriezellen

Aufgrund des stark nichtlinearen Verhaltens der Batteriezellen sind komplexe Batteriemodelle notwendig. Eine ausführliche Beschreibung ist [2] zu entnehmen. Die Klemmenspannung u hängt sowohl vom SoC, von der Temperatur als auch von der Alterung der Zelle (SoH: *State of Health*) ab. Demnach lässt sich das Batteriemodell in vier Teilsysteme unterteilen, siehe Abbildung 5.

Das dynamische Verhalten kann mit einer Reihenschaltung von RC-Gliedern approximiert werden, wie das Ersatzschaltbild in Abbildung 6 veranschaulicht.

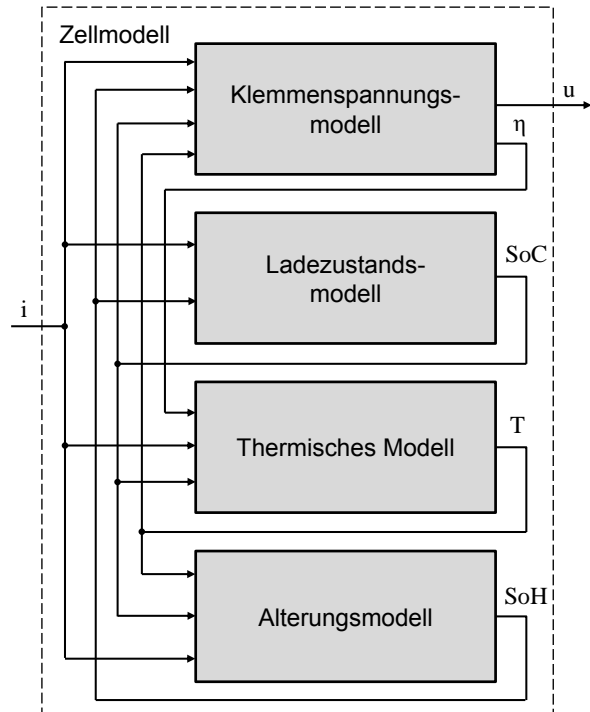


Abbildung 5: Modell einer Batteriezelle [3]

Die Anzahl der verwendeten RC-Glieder sollte einen sinnvollen Kompromiss aus Genauigkeit der Nachbildung und Rechenaufwand bilden. Eine Erhöhung der Anzahl bedeutet längere Berechnungsdauer sowie Verbesserung der Nachbildungsgenauigkeit.

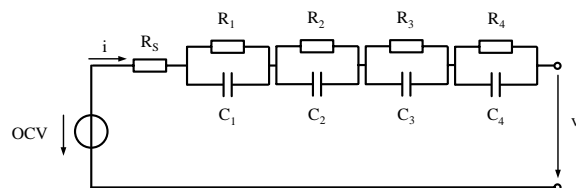


Abbildung 6: Ersatzschaltbild

Die Parameter der Widerstände und Kapazitäten wurden an realen Zellen vermessen und sind nichtlinear abhängig von SoC, Temperatur und SoH. Die Vorgehensweise zur Parameteridentifikation ist ebenfalls in [2] dokumentiert.

Der SoC wird ermittelt, indem der Batteriestrom auf die Gesamtkapazität bezogen und nach Gleichung 1 über der Zeit aufintegriert wird.

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

Im hier betrachteten Ladevorgang muss zusätzlich der Coulomb'sche Wirkungsgrad η_c berücksichtigt werden.

3.2 Funktion zur Ermittlung der Batteriezustände

Da die Ermittlung der Leistungsfähigkeit und weitere Softwarefunktionen auf dem aktuellen Ladezustand basieren, werden hohe Anforderungen an die Genauigkeit der Ladezustandsermittlung gestellt. Die Integration des gemessenen Stroms oder der Vergleich der gemessenen Spannung mit einem Batteriemodell erzielen für eine störungsfreie Simulationsumgebung genaue Lösungen, haben sich jedoch in der Praxis als nicht langzeitstabil bzw. unzureichend genau erwiesen. Das *Extended Kalman Filter (EKF)* mit komplexem Batteriemodell hat sich für eine genaue Ladezustandsermittlung bewährt.[4]

Die Struktur des diskreten Extended Kalman Filter ist in Abbildung 7 gegeben. Es wird parallel zum realen Prozess (hier durch das kontinuierliche nichtlineare Systemmodell repräsentiert) ausgeführt.

Auf Basis des gemessenen Stroms (= Systemeingang) und der aktuellen Zuständen des Systems wird der neue Zustandsvektor sowie der Systemausgang (= Klemmenspannung) präzisiert. Durch den Vergleich der präzisierten mit der gemessenen Spannung erfolgt mithilfe der Kalman-Verstärkung die Korrektur des präzisierten Zustandsvektors. Diese Prädiktions- und Korrekturschritte werden fortlaufend wiederholt. Für eine detaillierte Beschreibung des Entwurfs wird auf [3] verwiesen.

Die maximale Lade- bzw. Entladeleistung einer jeden Batteriezelle ist durch ihre Betriebsgrenzen bestimmt. Diese Betriebsgrenzen sind ein vom Hersteller vorgegebener Maximalstrom sowie eine untere und obere Spannungsgrenze. Ein Verlassen dieses Betriebsbereichs führt zu irreversiblen Schäden bis hin zur Zerstörung der Batteriezelle. Die aktuellen Leistungsgrenzen werden mithilfe eines inversen Batteriemodells für einen bestimmten Zeitraum berechnet (Bsp.: die Batteriezelle kann für einen Zeitraum von 10 s maximal 100 W abgeben).

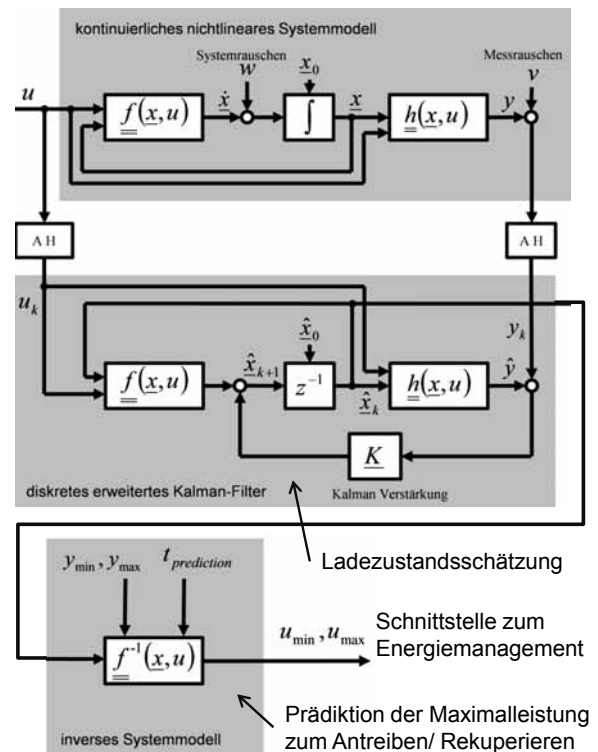


Abbildung 7: Diskretes Extended Kalman Filter mit kontinuierlichem Systemmodell [3]

In einer vielzelligen Batterie erreicht die Zelle mit dem höchsten Ladezustand beim Laden zuerst die obere Spannungsgrenze und die Zelle mit dem niedrigsten Ladezustand beim Entladen zuerst die untere Spannungsgrenze. Daher ist es nicht sinnvoll, einen durchschnittlichen Ladezustand der Gesamtbatterie zu berechnen, sondern die Ermittlung der Batteriezustände an der leersten und vollsten Batteriezelle zu orientieren. Aus diesem Grund ist das EKF auf dem realen Controller zweifach zu implementieren.

Das Vorgehen basiert auf der Kenntnis der einzelnen Spannungen. Da diese auf den Zellmodulen erfasst werden und erst an den Master kommuniziert werden müssen, wo die Batteriezustände ermittelt werden, sind die Latenzen, die durch die Buskommunikation auftreten, beim simulationsbasierten Test der Softwarefunktion in jedem Fall zu berücksichtigen.

3.3 Modellierung des Busverhaltens

Die serielle Kommunikation des Controllers mit den Zellmodulen wird mithilfe eines Zählers modelliert, dessen Zählerwert der Anzahl der Zellen entspricht. Jedes Zellmodul liest die aktuelle Ausgabe des Zählers ein und vergleicht sie gegen die eigene Adresse. Stimmen die Werte überein, ist das entsprechende Modul an der Reihe und gibt die zu diesem Zeitpunkt aktuellen Zustände der Zelle an den Controller weiter. Nach einem kompletten Zählerdurchlauf wird der nächste Werte geliefert. Das Batteriemodell wird unterdessen kontinuierlich simuliert.

Ein Takt des Zählers repräsentiert die Dauer einer Kommunikation zwischen Controller und Zellmodul und wird, wie die Simulationsschrittweite, nach Abschnitt 2.2 zu $0,67 \text{ ms}$ gewählt. Der Einfluss der Anzahl der Zellmodule auf die Latenz wird in Abbildung 8 verdeutlicht.

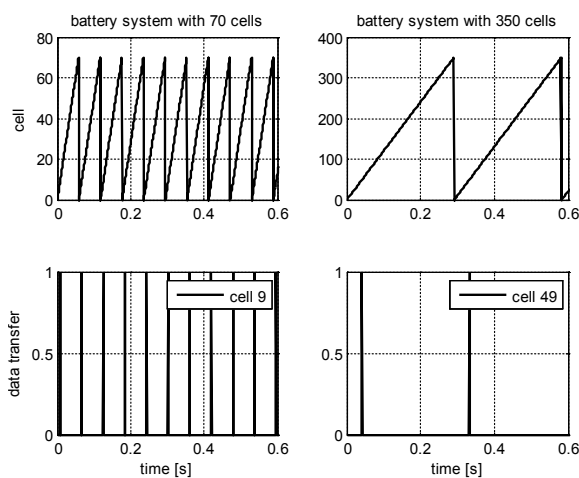


Abbildung 8: Einfluss der Zellenanzahl auf die Latenz

4 Modellbasierte Untersuchung

Um den Einfluss der Buslatenz zu ermitteln, wird ein Ersatzcontroller im Modell ergänzt, der die Klemmenspannung latenzfrei und ohne Diskretisierung aus dem Batteriemodell bezieht, und ebenfalls den SoC aus einem EKF ermittelt, siehe Abbildung 9.

Im Zuge der Untersuchung wird die Schätzung des Ladezustandes des EKF für verschiedene Anzahlen an

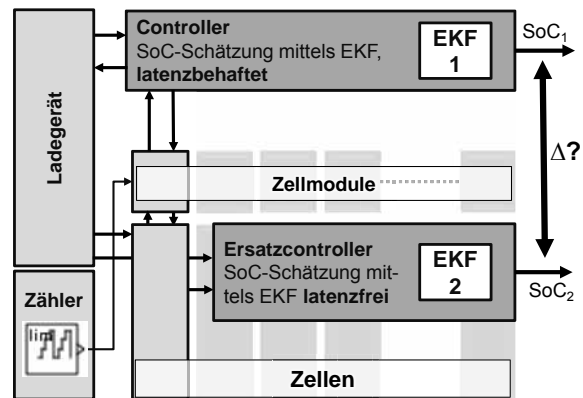


Abbildung 9: Erweitertes Modell

Zellen ausgewertet, um den Einfluss der Buslatenz zu analysieren. In Abbildung 10 ist die Schätzung des Ladezustandes aus dem EKF für eine variierte Zellanzahl der latenzfreien Schätzung des Ersatzcontrollers gegenübergestellt.

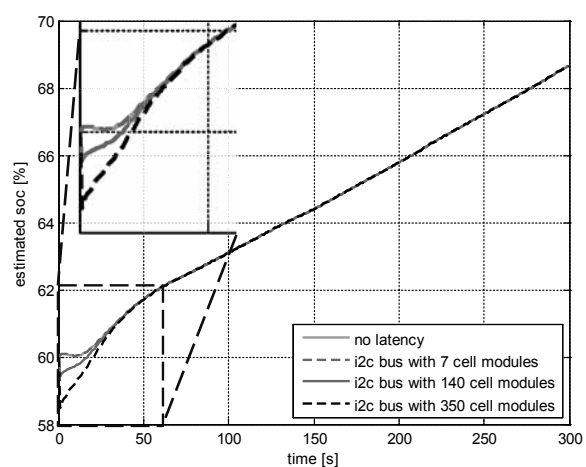


Abbildung 10: Erweitertes Modell

Die resultierenden Abweichungen sind als sehr gering einzuordnen. Selbst bei einer für die Praxis unrealistischen Anzahl von 350 Zellen fällt der SoC anfangs um lediglich 2 % ab. Bereits nach einer Dauer von etwa 30 s ist der Fehler kompensiert.

Ebenso sind die Auswirkungen auf die Balancing-Funktion sehr gering. Abbildung 11 zeigt die Klemmenspannung und den Balancing-Strom über den Bypass für die Zellen mit geringstem (links) und höchstem Ladezustand (rechts) sowie einer weiteren Zelle

mit einem dazwischen liegenden Ladestand. Die Verläufe für 7 bzw. 350 Zellen sind dabei nahezu deckungsgleich. Das dezentralisierte Batteriemanagementsystem mit I²C-Bus verfügt somit über sehr gute Skalierbarkeit.

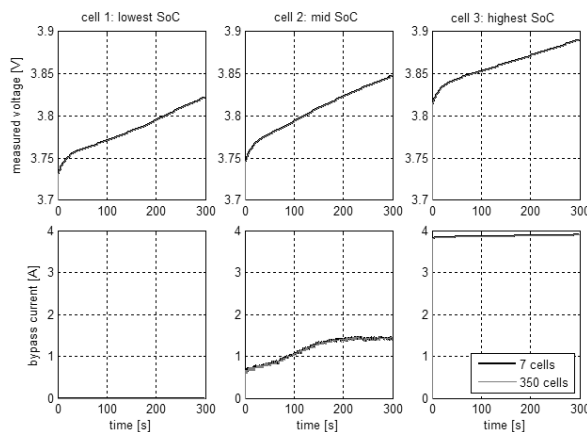


Abbildung 11: Auswirkung der Latenz auf die Balancing-Funktion

Anhand des erstellten Simulationsmodells lassen sich diverse weitere Eigenschaften des BMS hinsichtlich ihres Einflusses auf die Schätzung des SoC untersuchen. Ein interessanter Aspekt ist unter anderem die Auswirkung der Auflösung beim Erfassen der Messwerte auf dem Zellmodul. Abbildung 12 zeigt die Auswirkung der Diskretisierung.

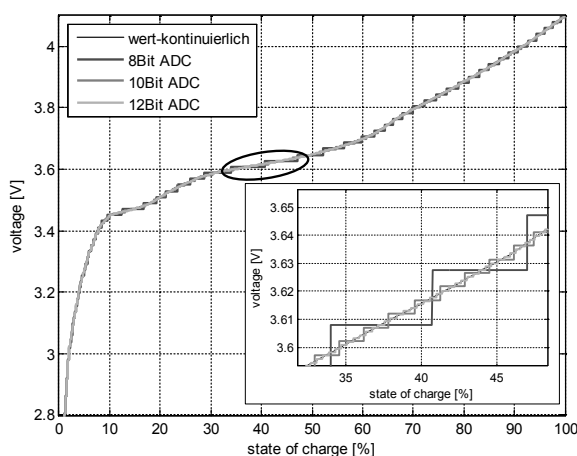


Abbildung 12: Spannungsauflösung im Arbeitsbereich einer Batteriezelle

Insbesondere bei Verwendung von 8-Bit-ADC wird

die Spannung nur unzureichend aufgelöst. Aufgrund der flachen Charakteristik des Spannungsverlaufes wird in einem SoC-Bereich von bis zu 7 % derselbe Spannungswert erfasst. Es ist daher von einer Beeinträchtigung der Ladezustandsschätzung auszugehen, was in der Folge untersucht werden soll. Dazu wird die Ladezustandsschätzung mit EKF mit Spannungssignalen verschiedener Auflösungen ausgewertet. Die Resultate sind in Abbildung 13 zu sehen.

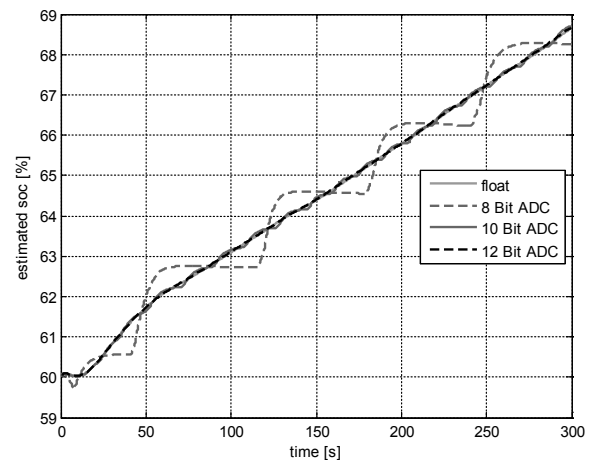


Abbildung 13: Einfluss der ADC-Auflösung auf die Ladezustandsschätzung

Wie zu erwarten war, liegt bei geringerer Auflösung eine deutliche Beeinträchtigung vor. Diese nimmt jedoch mit zunehmender Auflösung ab und lässt sich durch Einsatz eines 12-Bit-ADC nahezu vollständig eliminieren.

5 Resümee und Ausblick

Das Ziel dieser Arbeit lag darin, anhand der in einem Simulationsmodell abgebildeten Software-Funktionen eines Batteriemanagementsystems auch die Anforderungen an die Hardware in Bezug auf Bussystem, Messbereich, Auflösung, etc. festzulegen. Dazu wurde ein umfassendes Modell für BMS erläutert. Betrachtet wurde dabei ein BMS dezentraler Topologie, in dem Messdaten an den Batteriezellen durch Mikrocontroller erfasst werden und über I²C-Bus an einen Controller übermitteln.

Anhand der Funktionen des Ladezustandsschätzers mit Extended Kalman Filter sowie des Ladeausgleichs

wurde der Einfluss der Buslatenz auf die Funktion des Batteriesystems untersucht. Bei der gewählten Hardwarestruktur des BMS ist die Latenz proportional zur Anzahl an Zellen. Es konnte nachgewiesen werden, dass sich das BMS auf eine hohe Anzahl an Zellen skalieren lässt, ohne dass die Funktionen Einschränkungen unterliegen.

Weiterhin wurde der Einfluss der Auflösung der Messgrößen auf die Funktion Ladezustandsschätzung analysiert. Mithilfe der Untersuchung konnten geeignete Bauteile ausgewählt werden.

Zukünftig ist die Untersuchung anderer Bussysteme, wie zum Beispiel der CAN-Bus, für die Anwendung im BMS geplant.

Außerdem soll eine weitere Dezentralisierung der Regelstruktur untersucht werden. So könnte auf jedem Zellmodul ein EKF implementiert werden, um jeden SoC lokal zu ermitteln und den Kommunikationsumfang zu reduzieren. Das würde höhere Anforderungen an die Zellmodul-Hardware stellen, aber auch den Ladeausgleich basierend auf den Ladezustandsinformationen aller Zellen ermöglichen.

References

- [1] X. Liu-Henke et al. *Investigation of low-cost open-source platforms for developing of mechatronic functions with Rapid Control Prototyping*. 10th International Conference Mechatronic Systems and Materials, Opole, Polen, S. 114, 2014
- [2] F. Quantmeyer; X. Liu-Henke *Modelling the Electrical Behavior of Lithium-Ion Batteries for Electric Vehicles*. Solid State Phenomena 214, pp 40-47. 2014
- [3] F. Quantmeyer; X. Liu-Henke *State-of-Charge Estimation for Lithium-Ion Batteries in Electric Vehicles using Extended Kalman Filtering*. Journal of Energy Challenges & Mechanics. Aberdeen. 2014
- [4] G. L. Plett *Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs. Part 2: Modelling and identification*. Journal of Power Sources 134, pp 262-276. 2004

Performance Prediction and Optimization for Industrial Sieves by Simulation: a two-tier Approach

Christian Jaeger¹, Reza Housseini¹, Manfred Hertwig¹,
Thomas Hofer², Rudolf Marcel Fuchsli¹

¹Institute of Applied Mathematics and Physics
Zurich University of Applied Sciences, Winterthur, Switzerland

²Ammann Group, Langenthal, Switzerland
jaeg@zhaw.ch

We present a numerical study of sieving behavior on industrial sieves, composed of several vibrating screens, a process widely used in diverse industries. The modeling approach is twofold: on the one hand, particle flow is modeled in some detail by means of a discrete element model (DEM). This allows studying the influence of various parameters on the behavior of individual particles, particularly transport velocities and collision rates. Computational complexity however forbids the simulation of an entire sieve as a DEM. Instead, the overall sieving behavior is modeled separately by means of a more phenomenological model, the so called thick layer model (TLM), which is based on mass-balance equations, that translate into an ordinary differential equation. The TLM obtains its most crucial input parameters as results of the DEM. Comparison of simulation results with measurements shows, that this combined approach is capable of accurately describing the sieving process at a reasonable computational cost.

1 Introduction

Sieving or screening is the standard operation for the classification of particles according to their size, used in many industries. Experimental investigation of this process dates back to at least the middle of the 20th century, e.g. [8, 10]. The intricate dependencies of the sieving quality on various parameters (e.g. oscillation amplitude, frequency and sieve inclination) that were found, incited an interest in model-based studies. Existing models can be divided into those that are essentially based on the theory of stochastic processes and those relying on discrete element (DEM) simulations. Interest in the latter seems to receive new momentum with the increased availability of computing power in recent years, e.g. [4, 11] and others, see below. A widely used model is from Standish [13]. It is based on the simple assumption that screening is a first-order process. This means that the material on the screen falls through the mesh with a constant rate, giving rise to an exponentially falling function

of time, respectively of distance travelled along the mesh. The work of Andrzejczak and Wodzinski [3] is similar in assuming that passage through the mesh happens as a first order process with constant rates to be determined experimentally. In addition, they try to model the effect of a thick bed by assuming that only those particles can fall through the mesh that are in a so-called discharge layer of some given thickness, directly above the screen. The model of Sultanbawa et al. [14] is another macroscopic approach which is based on constant rates and mass balances. Instead of fitting the rates, the focus is on a related parameter q , the ratio of concentrations of undersize particles in the inlet and overtails streams. The theory is applied to a cascade of sieves and a graphical technique is developed to predict the concentration of fines with time. With the rapid progress in computer efficiency, microscopic modeling and simulation techniques have become popular in the last 10 years. The idea is to model individual particles and their interaction among each other and with the screen and to numerically follow their trajectories. The different sub-classes of such

an approach, such as event driven rigid particle mechanics, molecular dynamics, direct simulation Monte Carlo, or lattice models are described in detail in a review paper by Herrmann [9]. Most often, the discrete element method is applied to screening simulations. In two recent publications, this method is applied to a tumbling sorting machine by Alkhaldi [2] and to a multi-deck banana screen by Dong et al [5]. A three dimensional DEM model of a vibrating screening process is developed in [4] to study effects of amplitude, frequency and angle on screening quality and to develop empirical formulae to link particle-deck collisions with these parameters¹. The advantage of such a microscopic approach lies in the fact that the screening process parameters such as vibration amplitude and frequency, screen length and width, inclination angle, mesh size, and so on, are part of the model and their influence can be studied. Nonetheless, the usability is still limited by the computational complexity of the task. [2] states, that the simulation of a screening process of 9900 particles, which takes 45 seconds in real time, takes about one week of simulation time.

2 Simulation Method

The aim of the present study is to combine the aforementioned advantages of DEM simulations with the flexibility and computational lightness of a macroscopic approach. For this purpose two separate models are used to describe the screening process: a simplified DEM and the macroscopic thick layer model (TLM). The screening parameters considered for the simulation are the inclination angle of the sieve, the oscillation angle, the oscillation amplitude, frequency and the load rate.

2.1 The Discrete Element Model

Several Open-Source DEM packages are available, e.g. (in no particular order) LMGC90 [6], SICONOS [1], YADE [12], as well as commercial packages like EDEM. We used YADE to simulate a stripe, about of 0.2m width and 5m length of a sieve deck with a continuous inlet of particles. The simulated length corresponds to an actual sieve. The reduction in width

¹in the entire paper, *collision* refers to a collision between a particle and a sieve deck.

is justified by the fact, that particle velocity components across sieve width are very small. Rigid walls have been simulated along the edges of the stripe. The particles themselves were modelled as triangular pyramids, each composed of four balls. This allows simulation of non-spherical particle geometry (which would be the computationally simplest case) still at low computational cost compared to simulation of freeform-shapes. A snapshot of a simulation is shown in figure 1. Spherical particle geometry is

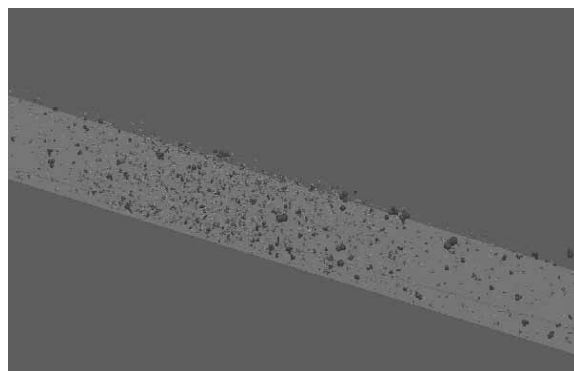


figure 1: Snapshot showing the motion of particles on a screen. The (tetrahedral) Particles are colored according to diameter. No actual sieve has been modeled, i.e. particles cannot fall through the screen.

used in e.g. [5], but in our simulations produced systematically too high transport velocities as compared to measurements. Particle size distribution can be set as a simulation parameter to correspond to the actual input on the real sieve, with particles mm – which in practice pass very quickly through the mesh - ignored. The simulated, vibrating screen is impenetrable for the particles: the purpose of the DEM is only to calculate transport velocities and collision rates in function of the parameters. The actual sieving simulation is the task of the TLM. The simplifications allow scanning the parameter space given in Table 1.

2.2 The Thick Layer Model

A sieve consists of several decks with meshes of different sizes. A schematic representation is given in fig. 2. Material is transported from top left to bottom right and should fall into the bag according to size, with bag sizes increasing from left to right. For the simulation each deck is divided into a number of cells and the

Incline Angle [deg]	0	5	10	15	20
Amplitude [mm]	2	4	6	8	10
Oscillation Angle [deg]	0	5	10	15	20
Frequency [rpm]	800	900	1000	1100	1200
Load rate [t/h]		140	230	320	

table 1: Parameterspace for DEM Simulation. All the possible parameter combinations have been simulated by the DEM.

amount [kg] of matter per fraction and cell is the state variable of the model. Under some simplifications, the relative error in due to the discretization into cells can be shown to be of order (by passing to the corresponding PDE without discretization) and can be chosen to keep this error within a few percent. The sieve must classify the matter according to size into a number of bags, the diameters that should fall in one particular bag form a so called fraction. The flow on the sieve is then modeled as an ODE

$$\begin{aligned}
 \frac{d\varphi_{i,j,n}}{dt} = & I_{i,j,n} - \left(\frac{v_{i,j,n}}{\lambda_{i,j,n}} + \sigma_{i,j,n} p_{i,j,n} \right) \varphi_{i,j,n} \\
 & + \frac{v_{i,j-1,n}}{\lambda_{i,j-1,n}} \varphi_{i,j-1,n} \\
 & - \sigma_{i-1,j,n} p_{i,j,n} \varphi_{i-1,j,n}
 \end{aligned} \quad (1)$$

using the mass-balance. The interesting output is mainly the distribution of particle sizes in the bags in the stationary state.

Here, the index (i, j, n) refers to fraction n in cell j on deck i . φ [kg] is the mass, I [kg/s] is the input stream, v [m/s] is the transport velocity, λ [m] is the length of the cell, σ [1/s] is the mean collision rate with the sieve deck and p [-] is the (geometric) probability per collision of a particle of passing through the mesh. The parameters v and σ are calculated from the DEM and fed to the TLM as input. The ODE is non-linear, since v and σ depend on the mass inside the

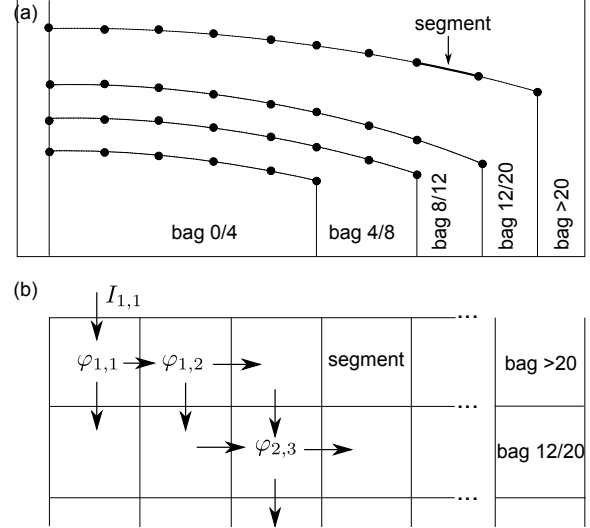


figure 2: Schematic representation of a sieve with five decks, each subdivided into several segments, (a) as simplified sieve geometry and (b) schematic. The state vector of the TLM is composed of the masses of each fraction in each segment. The equations governing the TLM are mass-balances of particles flowing in and out of each segment.

cell. The geometric probability p of a particle passing through the mesh in one collision can be approximated as [4]

$$p = \frac{(a-d)(\arccos(\theta) - d)}{(a+b)^2 \cos(\theta)} \quad (2)$$

where a, b and d are the sizes of the (rectangular) aperture and particle respectively and θ is the inclination angle. In our model, the particle size d is the mean diameter of particles in each fraction. The probability of one particle to remain on the screen after N collisions is then given by $p_N = (1-p)^N$. This is valid for individual particles, however as the particles progress along a sieve deck, the mean diameter d within a given fraction tends to increase (the smaller particles having already fallen through the mesh). Therefore a Weibull-Type distribution

$$p_N = \exp\left(-(\mu/k)^N\right) \quad (3)$$

which takes this “ageing effect” into account has been found more adequate. This is illustrated in figure 3.

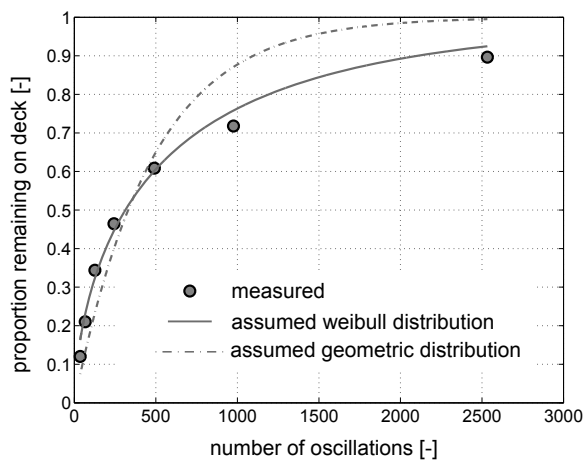


figure 3: proportion of particles of one fraction remaining on the deck after a number of oscillations. This proportion would follow a geometric distribution, if all particles were the same size (dash-dotted line). Since the proportion of larger particles rises with time, a Weibull-type distribution describes the measurement much better.

The additional parameters μ and k can be calculated from the (Gaussian) size distribution within the fraction or simply fitted to measurements.

2.3 Computing time

All simulations were performed on a standard personal computer (Intel Xeon cpu, 2.27GHz) with a Linux-type (CentOS 6) operating system. The computationally expensive part is the DEM. Each of the 1875 possible parameter combinations from table 1 was simulated for 5s simulated time, which took in total one week of computing time. The results from this simulation are velocities and collision rates, which were sampled at 0.01s intervals during the simulations.

Once the DEM part is completed, the transport velocities and collision rates can be used as input for the TLM (interpolated between the simulated parameters if desired). The TLM is an ODE which can be solved via a standard solver in the numerical open-source package GNU Octave [7]. The simulation can be stopped, once a stationary state is reached, which is normally the case after less than two minutes simulated time, corresponding to only a few seconds computing time (on the same processor). This allows for a

very fast estimation of sieve performance. The TLM simulations could also be performed e.g. during sales discussions at a potential customers site.

3 Measurements

Simulations were carried out under the real conditions of actual sieves, so they could be compared to field measurements of transport velocity and collision rate. The measurements were conducted on two sites by filming the stone on the top deck with a high-speed camera. Figure 4 shows the situation. The transport

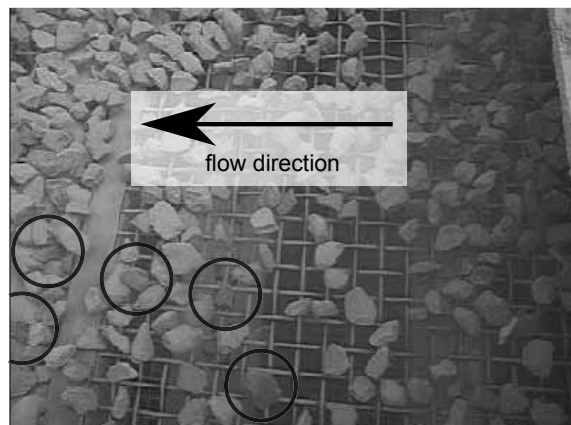


figure 4: Stone on the sieve as filmed during the measurements. The encircled stones entered the sieve at the same time and give an indication to the variability of the transport velocity.

velocity can be assessed with good confidence. At 7° inclination and an amplitude of 5.25 mm, the mean value was measured as 0.3 m/s and at 19° inclination and 3.5 mm amplitude it was measured to be about 0.4 m/s. The collision rate is much more difficult to assess. The measurements indicate, that at typical parameter values there will on average be less than one collision per oscillation.

Quality indicator for the sieving process are the amounts of under- and oversize, i.e. the amount of particles that are too small respectively too big for the bag they effectively landed in. Oversize can mostly be avoided by choice of an appropriate mesh size, thus undersize is the preferred quality measure. This quantity has a sharp bend when the load rate is increased over a critical point as illustrated in figure 5.

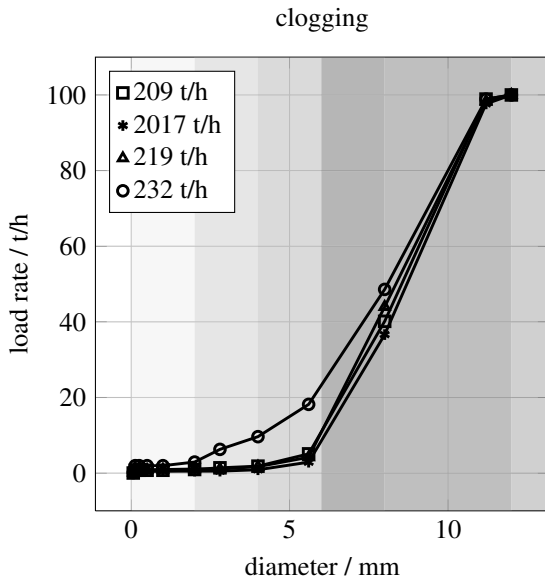


figure 5: Cumulative size distribution of the 8/12[mm] bag for different load rates. The distribution for 231.8[t/h] sharply differs from the lower load rates: at this load the bag contains much more undersize

4 Results

4.1 Model validation

Transport velocities and collision rates have been calculated for the parameters in table 1. The mean transport velocity is in good agreement with the measurement, as can be seen in figure 8, collision rates have not been measured.

This sharp bend of undersize at a critical load rate is qualitatively reproduced by the simulation as illustrated in figure 6.

To judge the overall agreement of the DEM/TLM model, we compared the cumulative distribution of particle sizes in the different bags, which showed very good agreement with the measurements, as is illustrated in figure 7.

4.2 Predictions

Intuitively, one expects good sieving quality at high collision rate/velocity ratios, thus maximizing the

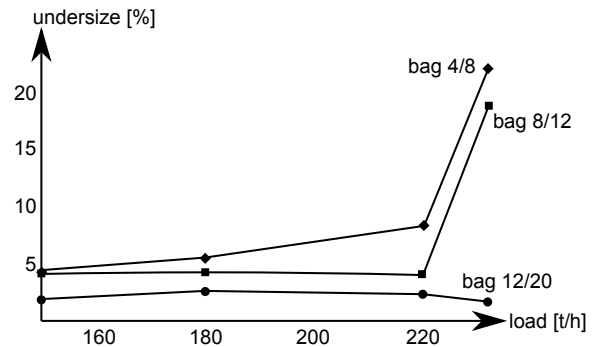


figure 6: Simulated undersize in three bags at different load rates. The qualitative behaviour, i.e. the sharp increase of undersize at a critical load rate is well reproduced by the model, the absolute quantities of undersize differ between measurement and model

number of collisions per length. The influence of parameters on these values can be investigated with the DEM. Figure 8 shows projections to the frequency/amplitude plane at base values of the remaining parameters. Collision rate shrinks with growing amplitude and is less affected by frequency. Mean transport velocity grows with amplitude and frequency. However, more information than just velocities and collision rates are needed, to determine optimal sieving parameters. E.g. lowering the velocity can lead to an accumulation of particles which implies a reduction of the average number of collisions between the particles and the screen, because more particles will not touch the screen anymore. This effect is not fully captured by our DEM, since no sieve-mesh has been modelled.

Figure 9 shows the influence of amplitude, frequency and an inclination of the entire sieve on the percentage of undersize in one particular bag. This is a result of the combined DEM-TLM. Good choices for amplitude/frequency combinations are clearly distinguishable in the plot. Moreover the sieving quality for a given choice can be influenced through the inclination angle. It is thus possible to optimize, e.g. for realizing good quality with lower amplitude/frequency (i.e. smaller drive capacities) or for realizing consistently optimum quality over all bags.

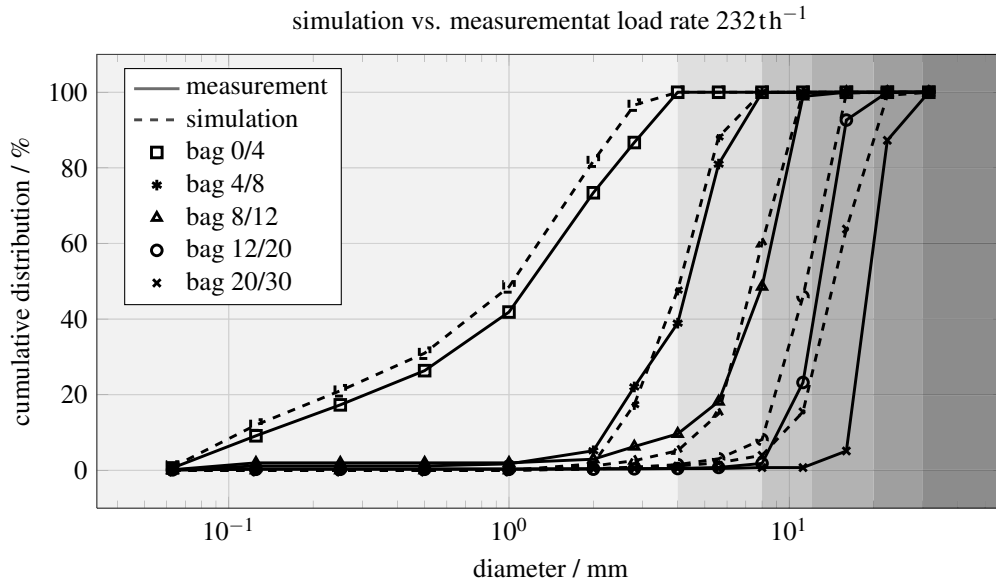


figure 7: Shown are measured and simulated data for bag 0/4, bag 4/8, bag 8/12, bag 12/20 and bag 20/30. The accumulated amount of particles with diameter x divided by the total amount, has been measured before and after the screening. It is compared to the accumulated distribution from the simulation. Each data pair corresponds to the output of a deck.

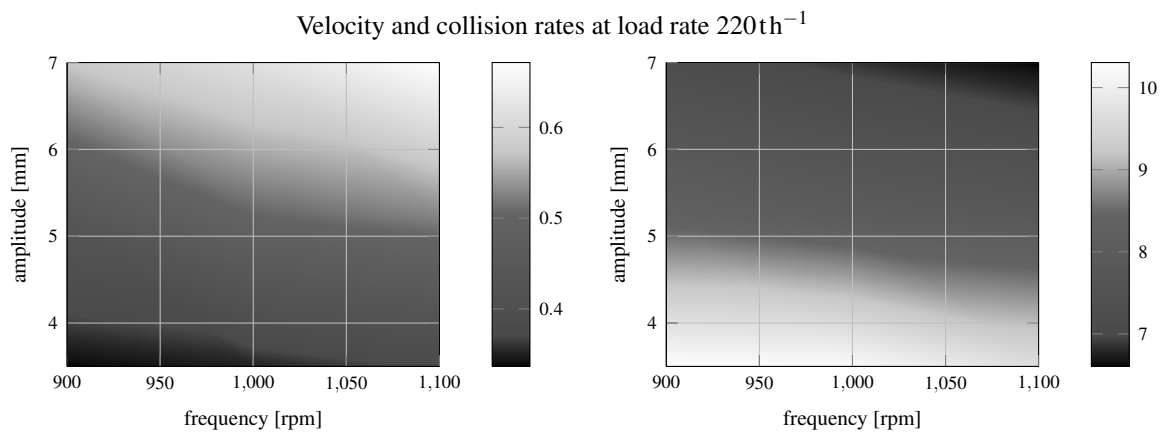


figure 8: transport velocities in m/s and collision rates in $1/\text{s}$ vs. frequency and amplitude as calculated by the DEM.

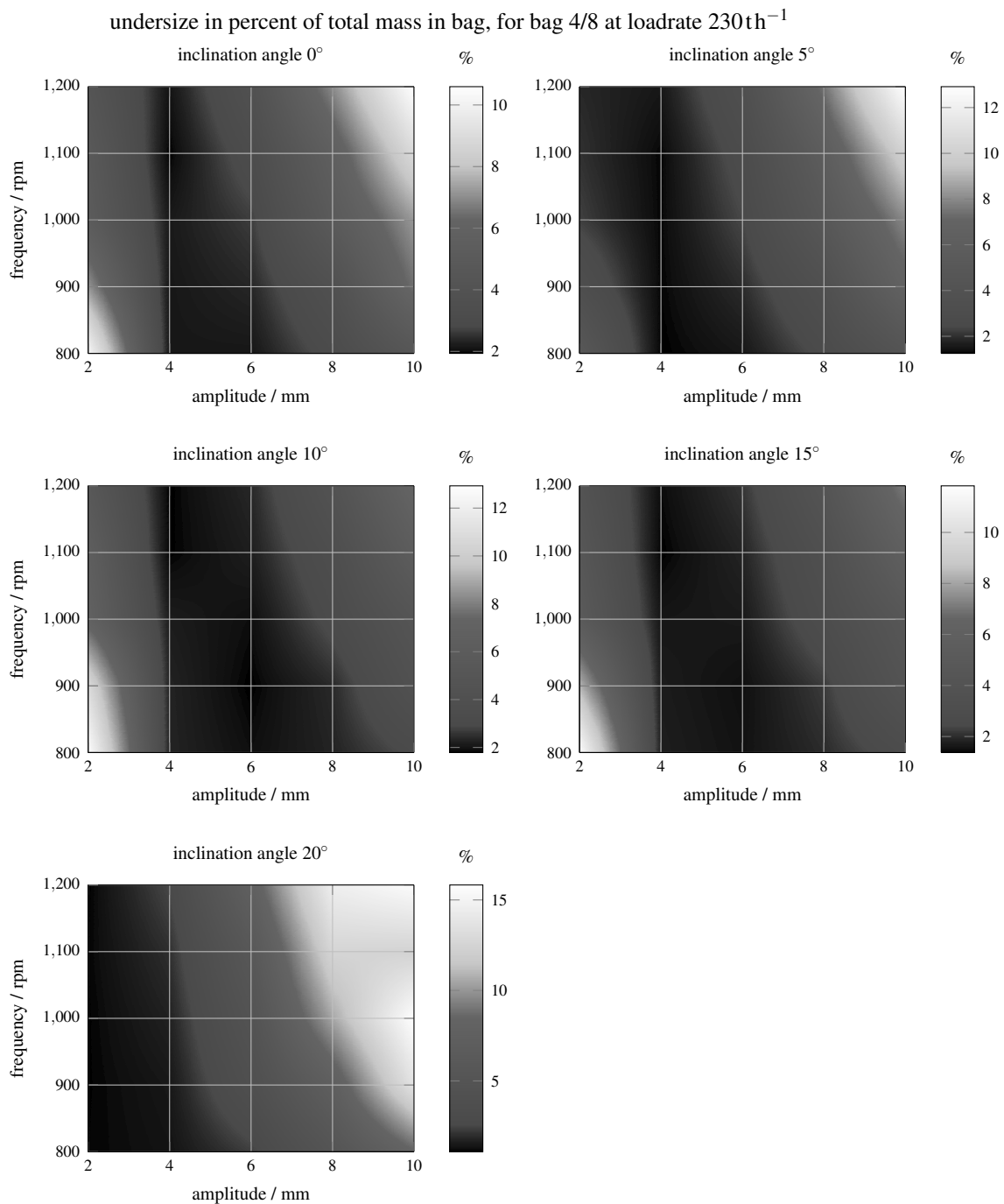


figure 9: Percentage of undersize in one bag vs. frequency and amplitude for different inclination angles. Here inclination angle refers to an inclination of the entire sieve in addition to the inclination of decks in the sieve.

5 Conclusions

A combined model approach composed of a microscopic (Discrete Element) model and a macroscopic (Thick Layer) model has been proposed to simulate vibrating screening processes. This two-tier approach to simulation permits to simulate an entire sieve at a modest computational cost, the entirety of the sieve being simulated by the computationally cheap thick layer model, based on inputs from the more expensive discrete element model, which in turn simulates only a representative but small excerpt of the sieve. Despite the simplicity of the approach, the result of the simulation is in good agreement with measurements done on real hot mineral screens in the field. A more complete validation will help to enhance some details of the model, but the tendencies that can be observed already with the actual model make it possible to detect interdependencies of the different parameters, and their influence on the screening result.

6 Acknowledgements

The present work has been funded by the Swiss Commission for Technology and Innovation (KTI). The project is a collaboration between the Ammann Group and the Zurich University of Applied Sciences (ZHAW). It is dedicated to the memory of the late Prof. Dr. Peter Biller on whose ideas the entire project is based.

References

- [1] V. Acary and F. Perignon. An introduction to siconos. Technical report, INRIA, 2007.
- [2] H. Alkhalidi. *Contact Investigations of Granular Mechanical Media in a Tumbling Sorting Machine*. PhD thesis, Universitaet Stuttgart, 2007.
- [3] P. Andrzejczak and P. Wodzinski. Model of screening in the layer. *Powder Handling and Processing*, 6:263–272, 1994.
- [4] K.J. Dong, B. Wang, and A.B. Yu. Modeling of particle flow and sieving behavior on a vibrating screen: From discrete particle simulation to process performance prediction. *Industrial & Engineering Chemistry Research*, 52:11333–11343, 2013.
- [5] K.J. Dong, A.B. Yu, and H. Brake. Dem simulation of particle flow on a multi-deck banana screen. *Minerals Engineering*, 22:910–920, 2009.
- [6] F. Dubois and M. Jean. The non smooth contact dynamic method: recent Imgc90 software developments and application. *Lecture Notes in Applied and Computational Mechanics*, 27:375–378, 2006.
- [7] J.W. Eaton, D. Bateman, and S. Hauberg. *GNU Octave version 3.0.1 manual: a high-level interactive language for numerical computations*. CreateSpace Independent Publishing Platform, 2009.
- [8] R.T. Fowler and S.C. Lim. The influence of various factors upon the effectiveness of separation of a finely divided solid by a vibrating screen. *Chemical Engineering Science*, 10:163 – 170, 1959.
- [9] H.J. Herrmann and S. Luding. Modeling granular media on the computer. *Trans IChemE*, 79:211–218, 2001.
- [10] M.L. Jansen and Glastonbury J.R. The size separation of particles by screening. *Powder Technology*, 1 (6):334–343, 1968.
- [11] J. Shi, Y. Huang, S. Han, B. Hu, and G. Xie. Fitting method research of sieving parameters of vibration screen and sieving efficiency based on dem. *Advanced Materials Research*, 655:20–23, 2013.
- [12] V. Smilauer, E. Catalano, B. Chareyre, S. Dorofeenko, J. Duriez, A. Gladky, J. Kozicki, C. Modenese, L. Scholtes, L. Sibille, J. Stransky, and K. Thoeni. Yade documentation, 2010.
- [13] N. Standish. The kinetics of batch sieving. *Powder Technology*, 41:57 – 67, 1985.
- [14] F.M. Sultanbawa, W.G. Owens, and S.S. Pandiella. A new approach to the prediction of particle separation by sieving in flour milling. *Trans IChemE*, 79:211–218, 2001.

Beer Distribution Game - Erfolgreich dank Regelungstechnik

Steinegger, Rolf Dipl. Ing. ETH; Barbey, Hans-Peter Prof. Dr.-Ing.

Zürcher Hochschule für Angewandte Wissenschaften, Fachhochschule Bielefeld

rolf.steinegger@zhaw.ch

Das Beer Distribution Game BDG kann als eine einfache Simulation zur Veranschaulichung zentraler Herausforderungen im Supply Chain Management SCM gelesen werden: Peitscheneffekt und Aufschaukelung von Bestelländerungen. Wie werden maximaler Gewinn und adäquate Lieferfähigkeit erreicht? Mit einem PID-Regler, basierend auf den üblichen Beschaffungsstrategien aus der Logistik (Bestellpunktverfahren und Bestellrhythmusverfahren) und dem SCM. Der Proportional- und der Integralanteil führen die Sicherheitsbestände und Bestellmengen dem Inputsignal nach. Der Differentialanteil sorgt dafür, dass die Lager weder leer laufen noch überlaufen, d.h. rasch den sich ändernden Kundenbestellungen nachgeführt werden.

Die Regleroptimierung erfolgt experimentell mittels Simulationsläufen. Die üblichen Anforderungen an das Führungsverhalten werden erfüllt: Stabilität, Genauigkeit, Schnelligkeit und Dämpfung, Robustheit. Die Ziele werden für alle Charakteristiken von Kundenbestellreihen zuverlässig und nahe dem theoretischen Optimum erreicht: Konstanz, Schwankung um Mittelwert und Trends. Der Bullwhip-Effekt lässt sich in Beschaffungsketten mit zeitlichen Verzögerungen und variablen Kundenbestellmengenreihen nicht grundsätzlich vermeiden, sondern nur kontrollieren.

Die Wirksamkeit systemischer Eingriffe bestätigt das Systems Engineering: 1. Veränderung der Anzahl Elemente im System (Verkürzung der Lieferkette), 2. Reduktion der Verzögerungseffekte (Verkürzung von Lieferzeiten) und 3. durchgängiger Informationsfluss (SCM). Die Betrachtung von Lager- und Transportkosten ergibt eine weitere Möglichkeit zur Optimierung des Reglers.

Keywords: Beer Distribution Game, Bullwhip-Effekt, Supply Chain Management, Bestellpunktverfahren, Bestellrhythmusverfahren, PID-Regler, Regelungstechnik.

1 Das Beer Distribution Game

1.1 Der Bullwhip-Effekt

Der Peitscheneffekt (Bullwhip-Effekt) ist zentrales Phänomen im Supply-Chain-Management. Es ergibt sich aus dynamischen Prozessen der Lieferketten. Bereits kleine Veränderungen der Endkundennachfrage können zu grösseren Schwankungen der Bestellmengen führen, die sich entlang der logistischen Kette wie ein Peitschenhieb aufschaukeln. Dieses Phänomen gehört zum "mainstream der Logistik" (Werner 2008 u.a.) [2].

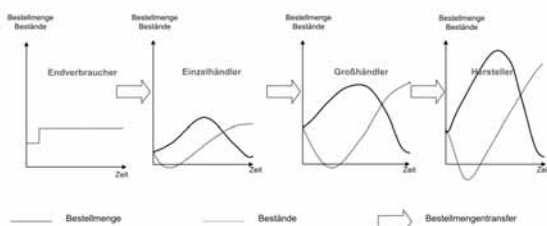


Abbildung 1. Ursachen für den Peitscheneffekt [1].

Es werden unter anderem vier Ursachen für den Peitscheneffekt identifiziert: Verarbeitung von Nachfragesignalen, Auftragsbündelung, Engpasspoker und Preisschwankungen [1].

Darüber hinaus ist der Peitscheneffekt systemimmanent. Er kann nicht einfach vermieden werden, indem die genannten Ursachen beseitigt werden werden.

1.2 Die spielerische Umsetzung

Kennen Sie das "Beer Distribution Game" (BDG)?

Das um 1960 am Massachusetts Institute of Technology (MIT) entwickelte BDG basiert auf den Arbeiten von Forrester (1958 / 1975) [3].



Abbildung 2. BDG: System und Logistik.

1.3 BDG - more than a game?

Und kennen Sie auch die Lösung des Problems?

Können Sie einem unbelasteten Anfänger einen Algorithmus erklären, mit dem er oder sie erfolgreich spielt? Ziele sind ja bekanntlich, dass die gesamte Lieferkette "automatisch" in die Gewinnzone kommt, dass der Gewinn der gesamten Lieferkette möglichst hoch wird und dass der Peitscheneffekt vermieden wird. Damit ist auch impliziert, dass die Lieferfähigkeit hoch ist, denn Lieferrückstände werden mit einem Malus bestraft. Diese Fragestellung fokussiert auf die Lösung(en) statt auf die Probleme – diese können als bekannt vorausgesetzt werden.

1.4 Erkenntnisse

- 1) Es ist möglich, einen oder mehrere geeignete Algorithmen zu beschreiben, mit denen eine Bestellmenge (und weitere Bestellparameter) berechnet werden, so dass der Erfolg im Spiel kontrolliert werden kann.
- 2) Lösungsalgorithmen sind möglich für Kundenbestellungen, welche um einen Mittelwert schwanken - wie auch für Kundenbestellungen, welche Trends unterliegen (Boom und Baisse).
- 3) Es lässt sich abschätzen, wie hoch der Gewinn im Spiel maximal sein kann. Die gefundenen Lösungs-Algorithmen lassen sich damit vergleichen. Die Algorithmen lassen sich untereinander vergleichen und bestimmten Anwendungsfällen (Vergleich von Modell und Realität) zuordnen.
- 4) Der Peitscheneffekt lässt sich in Beschaffungsketten mit zeitlichen Verzögerungen und variablen Kundenbestellungsmengenreihen nicht grundsätzlich vermeiden, sondern nur kontrollieren.
- 5) Es lässt sich eine Rangfolge bestimmen für die Wirksamkeit systemischer Eingriffe - Beispiele sind die Veränderung der Anzahl Elemente im System, die Verkürzung von Lieferzeiten etc.
- 6) Es lassen sich konkrete Gemeinsamkeiten und Unterschiede von Modell und Realität beschreiben.

2 Erfolgreiche Spielstrategien

2.1 "Order in - order out"

Wenn die Kundenbestellungen um einen Mittelwert (7) schwanken, ist es im BDG am einfachsten und erfolgreichsten, jeweils in jeder Runde dieselbe Menge nachzubestellen, wie gerade abverkauft wird. Damit kann sowohl finanzieller Erfolg erzielt werden wie auch eine ansprechende Lieferbereitschaft garantiert werden (z.B. Service Level 95 % und Gedeckter Anteil des Bedarfs 99 %), wenn das Lager in etwa doppelt so hoch gross ist wie die durchschnittliche Kundenbestellmenge (für Standardabweichung $\sigma = 2.4$).

Eine Abschätzung mit der Black Box-Methode nach Haberfellner (2012) [4] zeigt, dass der experimentell ermittelte Gewinn pro Runde von 5.4 durchaus im Bereich des maximal möglichen liegt.

Kosten- / Erfolgsfaktoren	Einheiten	Gewinn / Einheit	Erfolg
verkaufte Einheiten	7	1	+ 7
Lager	7	- 0.19	- 1.3
Transport	7	0	0
Lieferverzug	0.1	- 1	- 0.1
Gewinn			5.6

Tabelle 1. Maximal möglicher Gewinn, ermittelt mit der Black-Box-Methode. Alle Angaben sind Durchschnittswerte pro Spielrunde bei Beschaffungsstrategie "order in - order out".

Werden auch Transportkosten berücksichtigt, so hängt die Wahl der erfolgreichsten Spielstrategie vom Verhältnis von Lager- und Transportkosten ab. Je nachdem werden die bekannten Beschaffungsstrategien aus der Logistik (Bestellpunktverfahren oder Bestellrhythmusverfahren) oder aber die order in - order out-Strategie bessere Resultat bringen.

2.2 Bestellpunkt- u. Bestellrhythmusverfahren

Die folgenden Grafiken zeigen beispielhaft das experimentell ermittelte Systemverhalten für das Bestellpunktverfahren bei verhältnismässig kleinen Bestellmengen. Dies ist eine verhältnismässig erfolgreiche Strategie. Allerdings ist der Gewinn deutlich kleiner als bei "order in - order out". Der Grund sind die hohen Lagerkosten (pro Einheit).

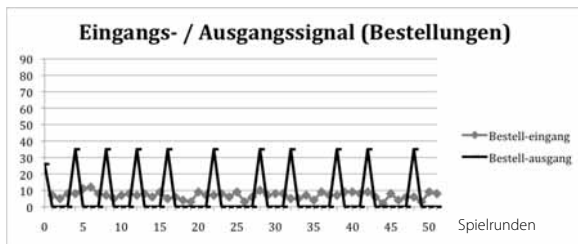


Abbildung 3. Das Ausgangssignal ist ein Abbild des variablen Bestellzyklus.

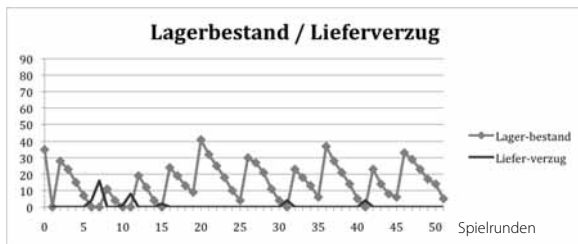


Abbildung 4. Der Lagerbestand zeigt die typische Kennlinie.

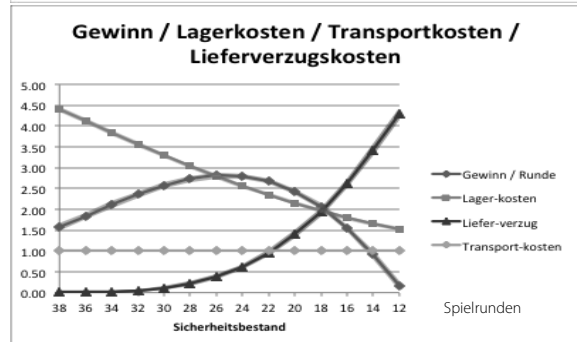
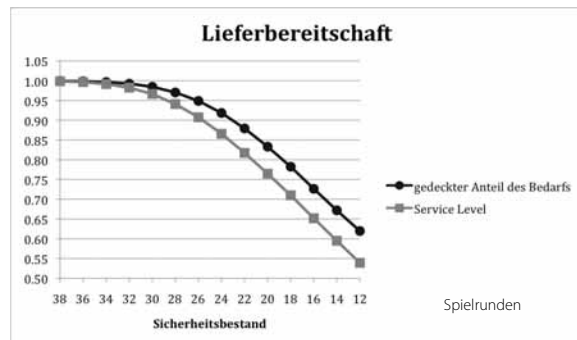


Abbildung 5. Lieferbereitschaft und Gewinn zeigen das Dilemma: Es ist nicht möglich, beide gleichzeitig zu optimieren.

2.3 PID-Regler zur Bewältigung von Trends

Wenn sich die Bestellungen zufällig entwickeln sind die Mitglieder der Lieferantenkette vor Probleme gestellt, für die es im Lehrbuch nicht ohne Weiteres eine einfache Anleitung gibt. Gomes/Probst (1999) [5] plädieren in solchen Fällen dafür, verwandte Systeme aus Wissenschaft und Technik für die Systemanalyse und zur Problemlösung beizuziehen.

Es wird eine Lösung aus der Regelungstechnik gezeigt, mit dem auch diese Herausforderung erfolgreich gemeistert wird. Ein solcher PID-Regler wurde erstmals von Dejonckheere (2003) [6] vorgeschlagen. Mann (2009) [7] bezeichnet "Regelung" als Vorgang, einen tatsächlichen Wert (sog. Istwert) mit einem gewünschten Wert (sog. Sollwert) fortlaufend zu vergleichen und den tatsächlichen Wert entsprechend zu korrigieren.

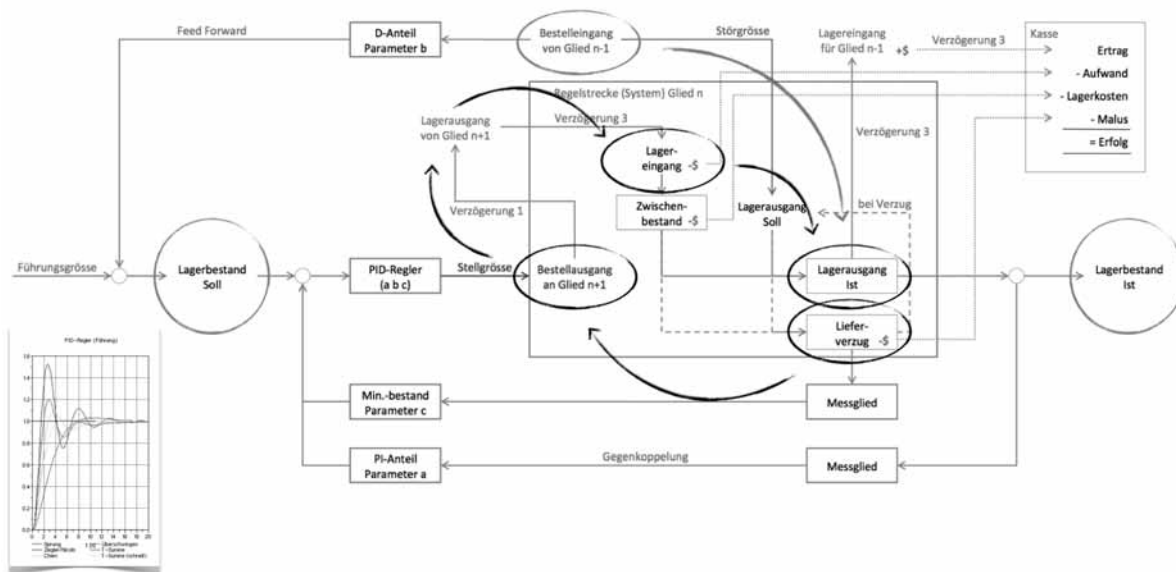


Abbildung 6. PID-Regler-Schema mit zentralem Wirkungskreis für ein Glied in der Beschaffungskette.

Der Proportional- und der Integralanteil führen die Sicherheitsbestände und Bestellmengen dem Inputsignal nach. Der Proportionalregler führt zu einer bleibenden Regelabweichung. Dies führt bei grösseren Störungen des Eingangssignals (Kundenbestellungen Boom) rasch zu leeren oder übervollen Lagern. Das Beispiel zeigt eine Änderung der Bestellmenge durch den Kunden um +2. Es resultiert eine Lagerveränderung von $-4 \cdot 2 = -8$, wobei 4 die Verzögerung der Lieferung ist.

Der Integralregler korrigiert den Fehler. Er bringt das Ausgangssignal zum Schwingen oder – bei korrekter, d.h. gedämpfter Reglereinstellung – asymptotisch zum Angleichen an das Eingangssignal (Parameter a).

Der Differentialanteil sorgt für Schnelligkeit und raschestmögliche Nachführung der Ist- an die Sollgrösse. Die Lager sollen weder leer noch überlaufen. Allerdings erhält das nachfolgende Glied der Kette ein stark überproportional vergrössertes

Eingangssignal (Bullwhip). Er wird deshalb ebenfalls in der Wirkung eingeschränkt (Parameter b).

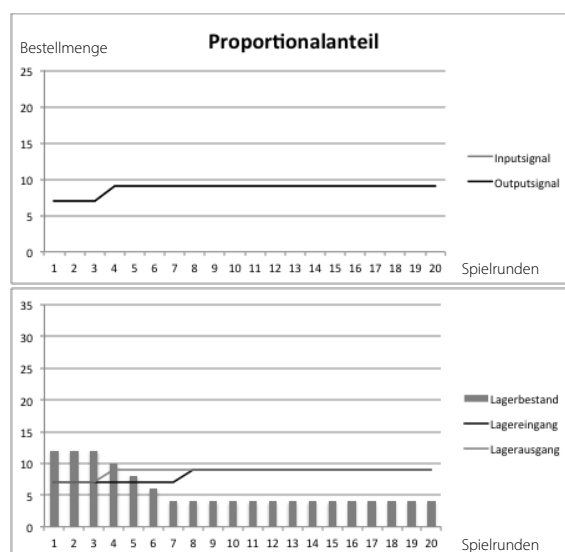


Abbildung 7. Proportionalregler mit bleibender Regelabweichung.

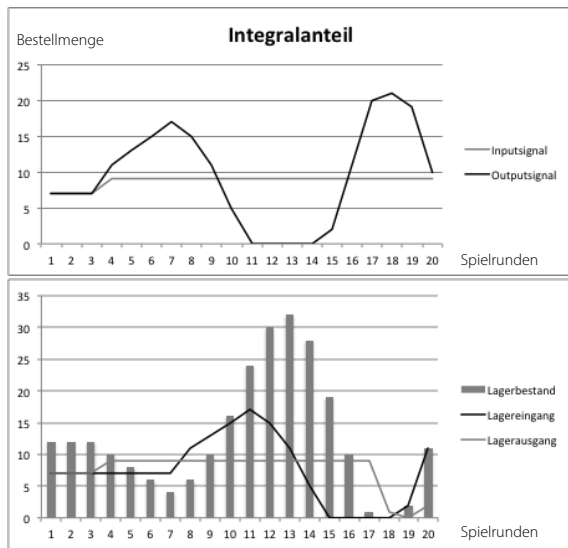


Abbildung 8. Integralregler: Fehlerkorrektur, resp. angeregter Schwingung (bei fehlender Dämpfung).

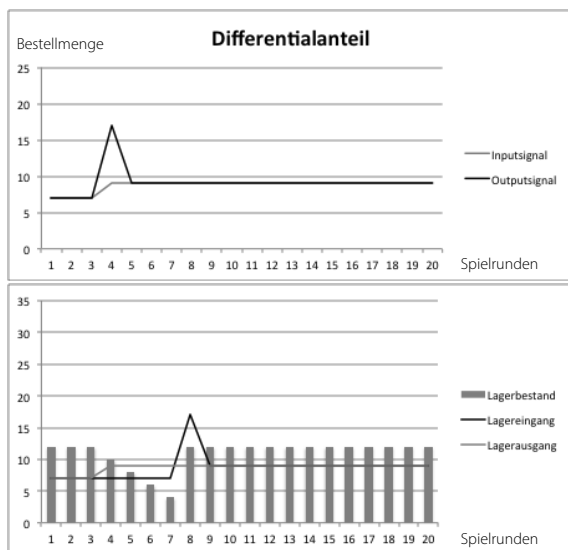


Abbildung 9. Differentialregler: Nachführung der Ist- an die Sollgrösse.

Die mathematische Formulierung für die lautet:

$$A = P + I + D \quad (1)$$

$$A = B \cdot (1 - a) + (S - L + B) \cdot a + T \cdot \Delta B \cdot b \quad (2)$$

P = Proportional-Anteil

I = Integral-Anteil

D = Differential-Anteil

B = Kundenbestellung [Anzahl Waren-Einheiten]

S = Sicherheitsbestand [Anzahl Waren-Einheiten]

L = Lagerbestand [Anzahl Waren-Einheiten]

T = Lieferverzugszeit [Anzahl Spielrunden]

ΔB = Bestellmengenänderung gegenüber Vorrunde [Anzahl Waren-Einheiten]

A = Bestellausgang (Nachbestellung beim Lieferanten) [Anzahl Waren-Einheiten]

2.4 Reglereinstellung

Mit Hilfe der Tabellenkalkulation werden eine statistisch relevante Zahl an Simulationsläufen mit jeweils 5000 "Spielrunden" durchgeführt werden. Die Reglereinstellung umfasst die folgenden Schritte.

1. Optimierung des vordersten Gliedes in der Lieferantenkette, d.h. Einstellung der Parameter anhand von Führungsverhalten und Sprungantwort.

Die "Störung" (= Änderung in der Kundenbestellung) soll einerseits schnellstmöglich nach hinten durchgereicht werden. Andererseits soll die Störung aber auch rasch und ohne übermässiges Überschwingen abklingen, d.h. die Fläche zwischen Sollwert und Ist-Wert soll minimal sein (absolute Werte). Dies entspricht der allgemeinen Vorgehensweise zur Optimierung der Reglerparameter nach Mann (2009) [8].

Mit der Bedingung "kein Überschwingen" kann der Parameter a beschränkt werden (Integral-Glied). Mit der Anforderung "raschestmögliche Informationsweitergabe" kann der Parameter b (Differential-Glied) bestimmt werden.

2. Optimierung der weiteren Glieder in der Lieferantenkette (Reglereinstellung) und jeweils Überprüfung, ob die Einstellung der davor liegenden (kundennahen) Glieder damit noch sinnvoll korrespondiert.

3. Bestimmung der Sicherheitsbestände (Parameter c) auf der Basis von Simulationsläufen zur Minimierung der Lagerbestände und damit der Lagerkosten.

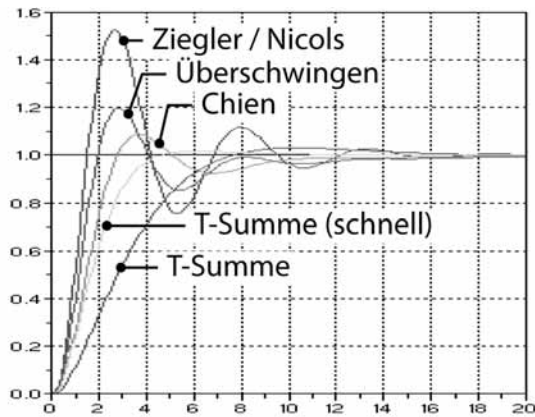


Abbildung 10. Verfahren zur Reglereinstellung und entsprechende Führungsverhalten (Wikipedia.org/Einstellregeln.png, bearbeitet).

Die rasche, aber gemässigte (Über-)Reaktion auf Veränderungen der Kundenbestellungen macht den Unterschied zum Misserfolg – zusammen mit relativ hohen Sicherheitsbeständen. Diese beiden Einstellungen erlauben gleichzeitig Gewinne und hohe Lieferbereitschaft. Dies entspricht dem Verfahren der T-Summe oder der T-Summe (schnell).

Parameter	1	2	3	4
1-a Proportional-Anteil	0.84	0.89	0.92	0.93
a Integral-Anteil	0.16	0.11	0.08	0.07
b Differential-Anteil	0.50	0.10	0.00	0.00
c Sicherheitsbestand	20	40	60	80

Tabelle 2. Die Regelparame-ter a, b und c aus experimenteller Optimierung mittels Simulationen für definierte Eingangssignale mit Trends für die vier Glieder der Beschaffungskette (1 = Detailhändler, 4 = Hersteller).

Das Eingangssignal basiert auf einer Veränderung pro Zyklus.: Min. = -5, Ø = 0, Max. = +5, $\sigma = 2.41$. Dies ist die mathematische Formulierung für zwei Würfel: Ein „grüner Würfel“ gibt die Plus-Zahlen vor, ein „roter Würfel“ zeigt die Minus-Zahlen. Aus der Differenz ergibt sich die neue Bestellmenge des Kunden, abgeleitet aus der vorhergehenden Bestellmenge (Steinegger 2009) [9].

3 Systems Engineering

Eine Optimierung nach dem Systems Engineering (Haberfellner 2012) [10] zeigt in der Simulation eine klare Rangfolge für die Wirksamkeit der Eingriffe:

1) Eine Verkürzung der Lieferkette bedeutet eine tiefgreifende, strukturelle Vereinfachung. Sie ist deshalb die wirkungsvollste Massnahme, um im BDG den finanziellen Erfolg zu erhöhen und gleichzeitig die Lieferbereitschaft zu gewährleisten.

2) An zweiter Stelle steht die Verminderung der Verzögerungseffekte (kürzere Lieferzeiten) und damit die Verminderung der Rückkoppelungseffekte.

3) Schliesslich leistet auch der durchgängige Informationsfluss ("open books") im Sinne des SCM einen Beitrag zur Optimierung der Gewinne. Diese Simulation bestätigt die Ergebnisse von Dejonckheere (2004) [11].

Die folgenden Abbildungen zeigen die Sprungantworten für optimierte Systeme und Regler im Vergleich.

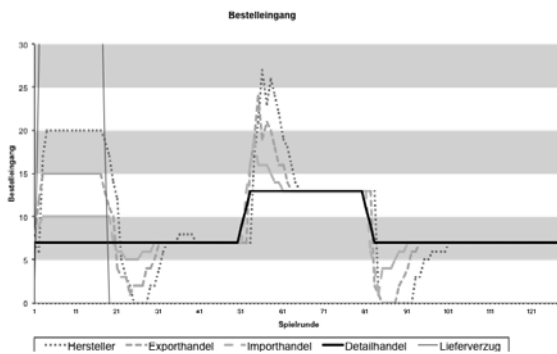


Abbildung 11. Klassische Spielanlage.

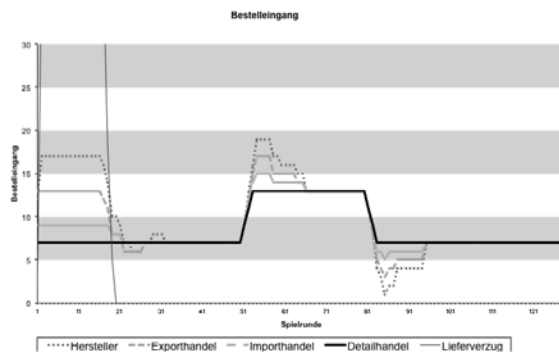


Abbildung 12. "Open Books" (SCM).

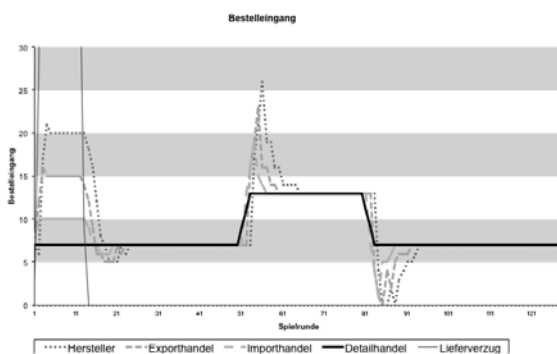


Abbildung 13. Verkürzte Lieferzeiten (2 statt 4).

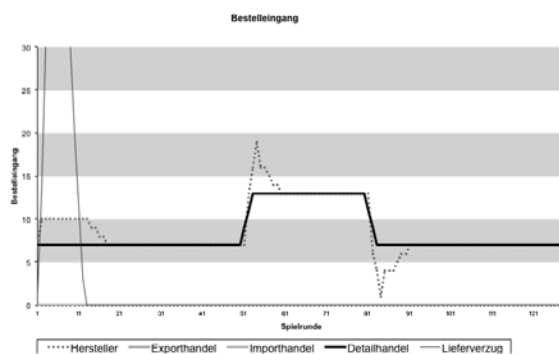


Abbildung 14. Verkürzte Lieferkette (2 statt 4).

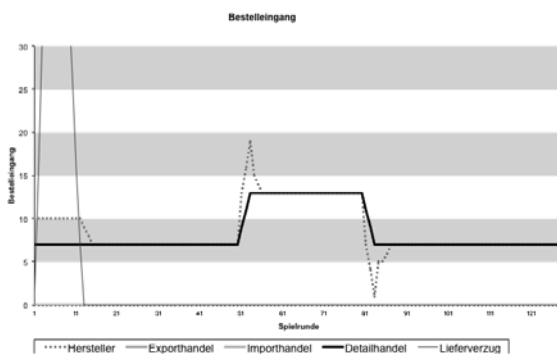


Abbildung 15. Verkürzte Kette und "Open Books".

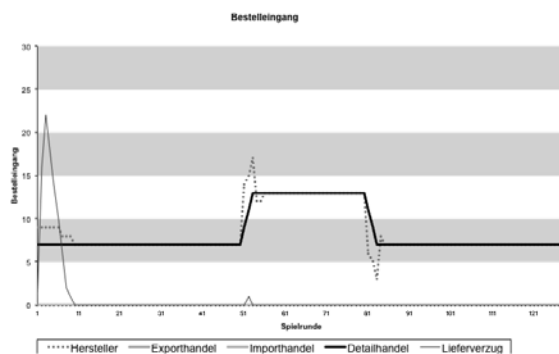


Abbildung 16. Verkürzte Kette, Lieferzeiten und "Open Books".

4 Vergleich von Modell und Realität

4.1 Lineare Lieferantenkette vs. Lieferantennetzwerk

Der definierten, linearen Kette aus gleichartigen Elementen mit einem Produkt (Bier) stehen in der Realität vielfältige, dynamische Beschaffungsketten mit verschiedenen Elementen und Beziehungen für mehrere Produkte mit verschiedenen Charakteristiken gegenüber. Das System ist zwar in beiden Fällen komplex, im Modell aber ist es abgeschlossen sowie

durch konstante Systemeigenschaften und Randbedingungen definiert (statt offen und veränderlich).

4.2 Lagerkosten vs. Transportkosten

Die hohen Lagerkosten und die fehlenden Transportkosten geben einen klaren Anreiz, um häufig zu bestellen (Bestellzyklus 1, also PID-Regler). In der Realität hängen Lagergröße und Bestellzyklen und damit der Entscheid für die "richtige" Beschaffungsstrategie vom Verhältnis von Lager- zu Transportkosten ab (u.a.). Damit im Zusammenhang stehen

selbstverständlich auch die Sicherheitsbestände und die Lieferbereitschaft, resp. die Lieferverzugskosten und die "lost sales".

4.3 Bestandspunktverfahren und Bestellrhythmusverfahren

Bei den bekannten Beschaffungsstrategien aus der Logistik werden Trends bewältigt, indem bei jeder Bestellung die Bestellparameter neu berechnet werden (z.B. Sicherheitsbestand, Bestellmenge, oder Bestellrhythmus) (Hosang 2009) [12].

Diese Strategien schneiden im BDG im Vergleich zum PID-Regler verhältnismässig schlecht ab, weil ab dem 3. Glied in der Beschaffungskette der "Ausnahmenzustand" herrscht. Durch die streng lineare Struktur der Kette hat jedes Glied nur genau einen Abnehmer, was zu sehr hohen Lagerkosten führt.

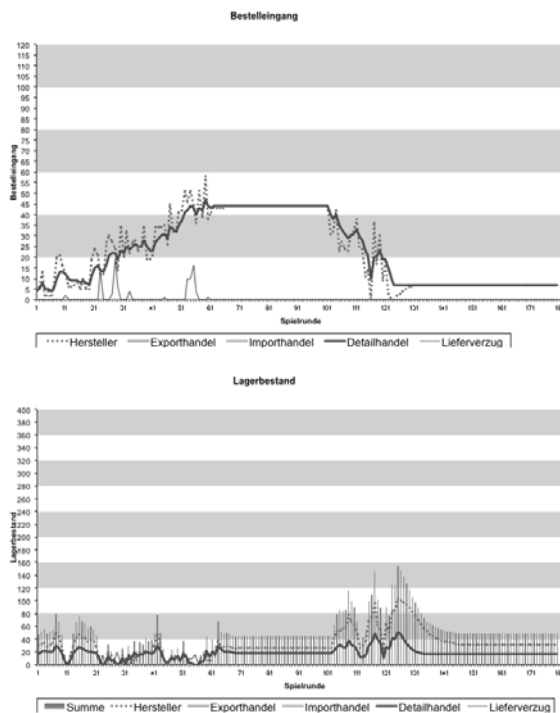


Abbildung 17. Trendverhalten für PID-Regler: Bestelleingang und Lagerbestand.

In diesem Fall hängt die Wahl der besten Strategie vom Verhältnis der Stückkosten zu den Logistikkosten und vom Verhältnis der Lager- zu den Transportkosten ab.

4.4 Eingangssignal

Das Eingangssignal (Kundenbestellmenge) wird in der Regel für die ganze Kette mehr oder weniger gleichmässig angenommen, d.h. jedes Glied in der

In der Realität hätte jedes Glied eine mehr oder weniger ausgeglichene Bestellreihe zu bewältigen, weil sich die Einzelbestellungen mehrere Abnehmer zu einem "Rauschen" überlagern. Ein sinnvoller Vergleich zwischen den Strategien betrachtet deshalb lediglich eine Beschaffungskette mit zwei Gliedern, was als "Direktimport" vom Detailhändler beim Hersteller interpretiert werden kann.

Die folgende Gegenüberstellung zeigt ein optimiertes System mit verkürzter Lieferkette, verkürzter Lieferverzögerung und "open books" (SCM). Auffallend sind die hohen Lagerkosten bei der klassischen Beschaffungsstrategie. Dies ist der Modellannahme im BDG geschuldet.

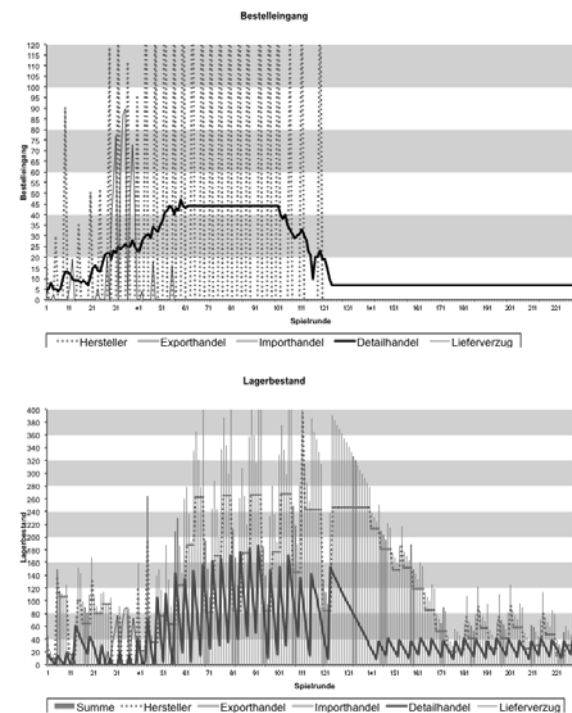


Abbildung 18. Trendverhalten für Bestellpunktverfahren (Mindestzyklusdauer = 4 Runden).

Beschaffungskette hat eine breite Kundschaft, sodass durch Überlagerung der Kundenbestellreihen eine statistische Verteilung entsteht (keine grossen Einzelbestellungen, die sich mit Phasen ohne jeden Bestelleingang abwechseln wie in den hinteren Gliedern der Beschaffungskette des BDG, wenn die klassischen Beschaffungsstrategien angewendet werden).

In der Realität dürfte ein um einen Mittelwert schwankendes Eingangssignal den Normalfall darstellen, d.h. die Kundenbestellreihe entspricht einem "Rauschen". Damit wird das Signal mathematisch eindeutig beschreibbar (mit statistischen Methoden). Trends dürften eher die Ausnahme bilden.

Ebenso ist das deterministische (= konstante) Inputsignal mit singulärer Irregularität (einmalige Änderung) aus der Spielanlage eine Modellannahme, die verhältnismässig weit von der Realität weg ist. Sie eignet sich aber sehr gut für die Veranschaulichung des Bullwhipeffektes und der Notwendigkeit für ein Reglerdesign sowie dessen Einstellung - und sie entspricht genau der Idee der Sprunganwort zur Optimierung des Führungsverhaltens.

4.5 "Nur Varietät kann Varietät absorbieren"

Das BDG dient also insgesamt als nützliches Modell für das Verständnis der "Realität". Es zeigt sich, dass es mehrere Modelle braucht, um die mannigfaltigen Aspekte der Realität zu erklären und zu verstehen. Wiegand (2005) [13] spitzt hierzu das Gesetz von Ashby wie folgt zu: "Nur Varietät kann Varietät absorbieren." - will heissen: Man kann ein Problem mit gegebener Komplexität nur mit Hilfe eines mindestens ebenso komplexen „Werkzeugkastens“ gut beschreiben und lösen.

5 References

- [1] [http://de.wikipedia.org/wiki/Peitscheneffekt_\(Supply-Chain-Management\)](http://de.wikipedia.org/wiki/Peitscheneffekt_(Supply-Chain-Management)). 10.06.2014.
- [2] Werner, Hartmut (2008, 3. vollständig überarbeitete und erweiterte Auflage): Supply Chain Management – Grundlagen, Strategien, Instrumente und Controlling. Gabler Deutschland. (S. 38f)
- [3] Forrester, Jay (1975): Industrial Dynamics: A Major Breakthrough for Decisions Makers. In: Collected Papers of Jay W. Forrester. Wright-Allen Press, Inc. (Seite 23) Erstmals publiziert in Harvard Business Review 36, no. 4 (July-August 1958), pp 37-66.
- [4] Haberfellner, Reinhard (Hrsg., 2012): Systems Engineering, 12. durchgesehene Auflage. Orell Füssli, Verlage industrielle Organisation Zürich. (S. 38)
- [5] Gomes / Probst (1999, 3. Auflage): Die Praxis des ganzheitlichen Problemlösens - vernetzt denken, unternehmerisch handeln, persönlich überzeugen. Haupt Berne. (S. 72ff)
- [6] Dejonckheere, J. et al (2003): Measuring and avoiding the bullwhip effect: A control theoretic approach. European Journal of Operational Research 147 (2003) 567–590.
- [7] Mann, Heinz et al (2009, 11. neu bearbeitete Auflage): Einführung in die Regelungstechnik - analoge und digitale, Fuzzy-Regler, Regler-Realisierung, Software. Carl Hanser Verlag München. (S. 13)
- [8] Mann, Heinz et al (2009, 11. neu bearbeitete Auflage): Einführung in die Regelungstechnik - analoge und digitale, Fuzzy-Regler, Regler-Realisierung, Software. Carl Hanser Verlag München. (S. 210ff)
- [9] Steinegger, Rolf (2009): Spielstrategien für die Simulation einer Lieferantenkette aus der Logistik ("Bierspiel"). Zürcher Hochschule für Angewandte Wissenschaften.
- [10] Haberfellner, Reinhard (Hrsg., 2012): Systems Engineering, 12. durchgesehene Auflage. Orell Füssli, Verlage industrielle Organisation Zürich. (S. 41ff)
- [11] Dejonckheere, J. et al (2004): The impact of information enrichment on the Bullwhip effect in supply chains: A control engineering perspective. European Journal of Operational Research 153 (2004) 727–750.
- [12] Hosang, Jürg (2009): Supply Chain Management. Skript zur Vorlesung Ökonomische und Soziale Systeme 2. Zürcher Hochschule für angewandte Wissenschaften. (S. 16ff).
- [13] Wiegand, Jürgen (2005): Handbuch Planungserfolg - Methoden, Zusammenarbeit und Management als integraler Prozess. vdf Hochschulverlag ETH Zürich. (S. 205)

Serienproduktionsbegleitende Materialflusssimulation für Maßhaltigkeitsoptimierung in der komplexen Karosseriefertigung

Anke Beckmann^{a*}, Martin Bohn^a, Peter Gust^b

¹Daimler AG, Benzstraße, 71063 Sindelfingen

²Universität Wuppertal, Institut für Konstruktionstechnik

**anke.beckmann@daimler.com*

Globalisierung und steigender Wettbewerb stellen große Herausforderungen an die Effizienz und Effektivität produzierender Unternehmen. Eine Folge davon sind unter anderem immer kürzere Entwicklungszyklen. Hinzu kommen hohe eigene Qualitätsstandards und das steigende Qualitätsbewusstsein der Kunden. Im Sektor der Fahrzeugindustrie sind – unter Anderem auch durch Medienberichte – Fugenmaße und –übergänge zwischen Bauteilen in das öffentliche Interesse gerückt. Tatsächlich unterliegen im Rohbau alle Einzelteile und Zusammenbauten Fertigungsabweichungen. Dabei ist die Streuung innerhalb verschiedener Bauteilchargen nachweislich sehr ähnlich, während die Mittelwertverschiebung oft signifikant ist. Diese Mittelwertverschiebungen gilt es auszugleichen, bzw. bestmöglich zu nutzen. Dafür sind neue innovative Lösungen, wie das Bauteil-Matching weiterzuentwickeln und für komplexe Großserienproduktionen anwendbar zu machen. Der vorliegende Beitrag zeigt eine Methode auf, wie komplexe Materialflussprozesse mit wenigen Ressourcen prognostiziert werden können, um so auf Fertigungsstreuungen flexibler reagieren zu können. Langfristig können auf diese Weise Methoden, wie das Bauteil-Matching im Sinne höchster Qualitätsansprüche förderlich umgesetzt werden.

1 Höchste Anforderungen für produzierende Unternehmen von Konsumgütern

Die Automobilindustrie ist geprägt von höchsten Qualitätsstandards bei stetig steigenden Kosten der Ressourcen für die Herstellung und die flächendeckende Anwendung automatisierter Prozesse. Gleichzeitig steigt der Kundenwunsch nach kostengünstigen, aber dennoch qualitativ hochwertigsten und individuellen Fahrzeugen. Damit die Automobilhersteller im globalen Wettbewerb bestehen können, müssen sie mit immer kürzeren Entwicklungszyklen auf dem Markt präsent bleiben. Die steigenden Qualitätsanforderungen erfordern von den Herstellern eine immer stärkere Überwachung ihrer Produkte bevor diese zum Kunden ausgeliefert werden. Je später Fehler entdeckt und behoben werden müssen, desto teurer wird es für den Hersteller [1]. Jeder Herstellungsprozess unterliegt Fertigungsabweichungen, die durch ungünstig aufsummierende Zusammenbauprozesse in späteren Schritten zu Nacharbeit oder zum Ausschuss führen können (Abbildung 1). In der Automobilindustrie finden oftmals für die Auslegung solcher Toleranzen die 6σ -Methode oder analog auch 8σ -Methode Anwendung [2], [3]. Fertigungsabwei-

chungen unterliegen im Rohbau überwiegend der Normalverteilung.

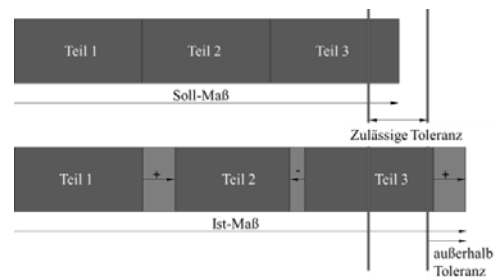


Abbildung 1. Schematische Darstellung, ungünstig aufsummierendes Schließmaß

Durch die Verwendung von statistischen Grundsätzen können die Toleranzen der Einzelteile größer gehalten werden, um höchste Qualität bei gleichbleibendem Fertigungsaufwand zu produzieren. Da die Worst-Case-Fälle sehr selten auftreten (Abbildung 2), ist dadurch der Ausschuss dennoch klein zu halten. Diese Methode empfiehlt sich vor allem in der Serienproduktion mit hohen Stückzahlen. Trotz des Vorliegens des statistischen Phänomens, können durch Mittelwertverschiebungen des Prozesses, bedingt durch Umwelteinflüsse, Werkzeugverschleiß oder

Ähnlichem, mehr Nacharbeit oder Ausschuss erzeugt werden. Um den verschobenen Prozess wieder in die Toleranzgrenzen zu „verschieben“, sind an dieser Stelle eine Senkung der Streuung des Merkmals, die Aufweitung der Toleranzgrenzen oder die Rückverschiebung des Prozesses für eine Verringerung des Ausschusses möglich. Oftmals stehen für Prozessänderungen am Einzelteil bereits kostspielige Werkzeugänderungen oder komplexe Prozessumstellungen im Raum. Die alleinige Aufweitung der Toleranzgrenzen reicht jedoch auch meist nicht, bzw. bringt Schwierigkeiten in später folgenden Zusammenbau-stufen mit sich. Erfolgsversprechend hingegen ist die Verringerung der Streuung. Da die Rohbauproduktion allerdings bereits auf einem sehr hohen Maßhaltigkeitsniveau stattfindet, ist die Entwicklung innovativer Methoden notwendig.

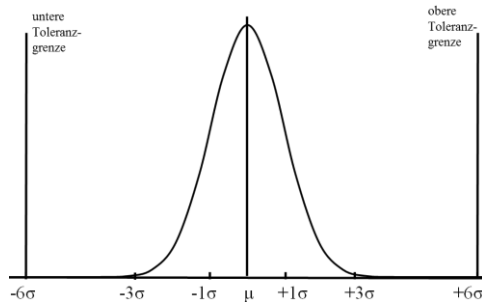


Abbildung 2. Normalverteilung mit Ausschussbereich nach 6σ

In produzierenden Unternehmen, die Zusammenbauten erzeugen, wird die Idee des Bauteil-Matchings immer verbreiteter. Von der Idee bis zur Umsetzung muss jedoch immer ein steiniger Weg beschritten werden. Oftmals bleibt es eine Vision, da viele Baustellen auf lokaleren Ebenen zuerst bearbeitet werden müssen, um so einen Fortschritt realisieren zu können. Ziel des vorliegenden Beitrags ist es, die Möglichkeiten eines Bauteil-Matchings in der Automobilindustrie – im Karosseriebau – vorzustellen, dessen Auswirkungen und Schwierigkeiten zu diskutieren und die daraus entstandene Methode der Bauteilverfolgung bzw. Vorhersagesimulation der hochkomplexen Materialflussprozesse einer großen Serienfertigung zu präsentieren. Eine prognostizierende Materialflusssimulation ist unabdingbar für die Realisierung eines Bauteil-Matchings im Karosseriebau und ist damit ein wichtiger Schritt zur Industrie 4.0.

Das vorliegende Projekt gliedert sich nach Abbildung 3. Am Anfang steht die Vision: höhere Präzision zu erzeugen, d.h. die Streuung der Zusammenbauten zu

verringern. Die Idee zur Lösung des Problems soll an dieser Stelle das Bauteil-Matching darstellen. Um ein sinnvolles Bauteil-Matching zu realisieren, müssen die hoch komplexen Materialflussprozesse innerhalb einer Großserienproduktion identifiziert und modelliert werden. Erst dann können die Prozesse vorhergesagt werden, wobei hier die Unsicherheiten im Materialflussmodell eindeutig definiert und beachtet werden müssen. Gerade die Flexibilität der Fertigungsstraßen, durch den Wunsch der Kunden nach Individualität, macht Logistikprozesse schwer vorhersagbar, da die Fahrzeuge auftragsorientiert gefertigt werden und nicht variantensortiert.

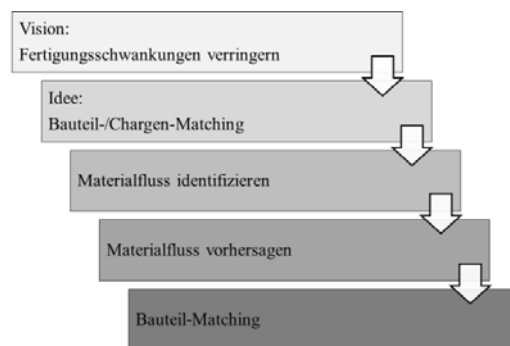


Abbildung 3. Ablaufdiagramm des Vorhabens

2 Industrie 4.0

Das Bauteil-Matching ist ein Teilgebiet des sich rasant entwickelnden Forschungsbereichs Industrie 4.0. Industrie 4.0 beschäftigt sich mit der Digitalisierung der realen Welt, sowie der immer stärkeren Vernetzung verschiedener Systeme und ist sogar von der Bundesregierung in der „Hightech-Strategie für Deutschland“ benannt [4]. Durch die Vision „Smart Factory“ sind alle Komponenten wie in einem sozialen Netzwerk miteinander verbunden. Ziel einer intelligenten Produktion ist es jedes Produkt zu jedem Zeitpunkt eindeutig identifizieren und lokalisieren zu können, sowie zusätzlich alternative Wege zum Zielzustand aufzuzeigen [5]. Neben der Idee und der Umsetzung an sich muss der entscheidende Aspekt der Datenstandardisierung der Teilnehmer beachtet und einbezogen werden. Die Datenstandardisierung ist in vielen Anwendungsgebieten jedoch nicht von heute auf morgen umsetzbar. Es müssen Schnittstellen zwischen den vielen interdisziplinär wirkenden Bereichen hergestellt und diese mittels gut ausgebauter Breitbandverbindungen für eine Echtzeitweitergabe der Daten ausgestattet werden. Insgesamt ist diese Vision für alle denkbaren Bereiche von Bedeutung, wie für die Luftfahrt, die Logistik, die Produktion

oder sogar auch für das soziale Leben. Trotz vorhandener konkreter Ideen wird noch ein langer Weg bis zur vollständigen Ausweitung zu beschreiten sein. Der folgende Beitrag weist einen ersten Wegschritt basierend auf geringsten Investitionen für die Automobilindustrie aus.

3 Anwendungsgebiet: Serienproduktion von Fahrzeugkarosserien

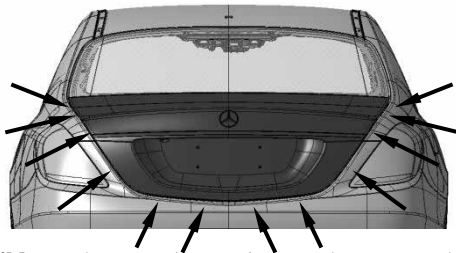


Abbildung 4. Darstellung eines verbauten Heckdeckels im Fahrzeug. Herausstellen der wichtigen Fugenbereiche

Fahrzeugkarosserien bestehen aus mehreren hundert Teilen. Um die Komplexität überschaubar zu halten, wurde für den vorliegenden Beitrag das abgeschlossene Produkt „Heckdeckel“ ausgewählt. Grundsätzlich lassen sich die Ausführungen aber auch auf jedes andere Produkt übertragen. Der Heckdeckel wurde gewählt, da er mit verschiedenen Bauteilen wichtige Fugen bildet (Abbildung 4). Durch die verschiedenen Gegenstücke, wie Seitenwand oder Stoßfänger, entsteht ein komplexes Zusammenspiel der Geometrie der Heckdeckelöffnung zum Heckdeckel.

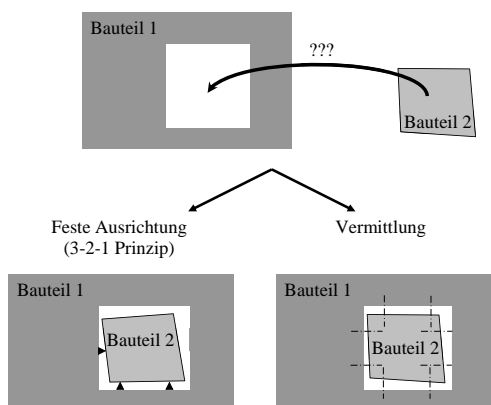


Abbildung 5. Ausrichtungsmethoden im Vergleich [2]

Für den Einpassungsprozess verschiedener Bauteile hat sich das Verfahren „Best-Fit“ etabliert. In Abbildung 5 wird schematisch der Verbau von einem Bau-

teil in ein anderes beschrieben. Beim Verfahren der festen Ausrichtung ist es nur möglich eine einzelne Fuge zu priorisieren und diese dann optimal einzustellen. Da aber beide zusammentreffenden Bauteile Formabweichungen aufweisen können, ist es erfolgsversprechender die Toleranzen zu vermitteln, so dass das bestmögliche mittlere Ergebnis erzeugt wird. Zusätzlich ist es hier aber auch denkbar, das bestmögliche Ergebnis in Bezug auf eine definierte Anzahl von Fugen zu erzeugen. Dabei müssen dann bestimmte Fugen höher priorisiert werden als andere, was durchaus sinnvoll sein kann, wenn bestimmte Fugen sehr stark im Blickfeld liegen, andere eher nicht. Im Fügeprozess wird dieses Anpassen durch einen automatisierten Algorithmus erledigt.

4 Materialfluss identifizieren

Großunternehmen verfügen über große Daten- und Wissensmengen. Dies bedeutet einen Vorteil, soweit es auch gelingt, diese bestmöglich zu nutzen. Eine effektive und umfassende Nutzung stellt jedoch eine große Herausforderung dar. So bedeutet auch die Vision „streuungsärmere Fertigung“ die Umsetzung eines interdisziplinären Projekts mit einer sehr hohen Anzahl an Mitwirkenden (Abbildung 6).

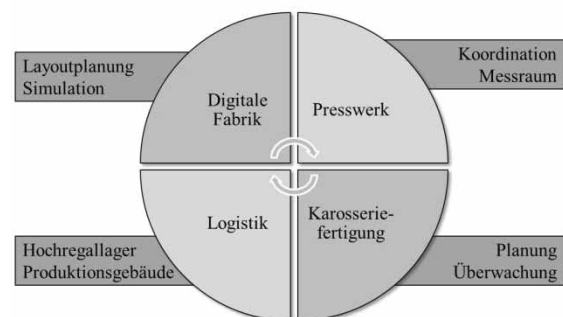


Abbildung 6. Darstellung des Zusammenspiels, der während der Produktion am Produkt beteiligten Bereiche

Das, in Kapitel 2, beschriebene Problem der durchgängigen Schnittstellen und der möglichen Breitbandverbindungen kommt hier zum Tragen. Jeder Bereich arbeitet mit eigenen Systemen und Fragestellungen, so dass eine Vereinheitlichung noch viel Zeit in Anspruch nehmen wird [6].

Abbildung 7 zeigt beispielhaft den Ablauf der Heckdeckelproduktion. Vor der Heckdeckelherstellung befindet sich zusätzlich ein Puffer, bzw. ein Wartepplatz für Ladungsträger, der die durchgängige Produktion auch bei Ausfall der vorgelagerten Stationen

eine ungenügende Weitergabe von Produktdaten. Erst ab einer Kundenzuordbarkeit werden Produktdaten auf einem RFID-Chip gespeichert und an die nachfolgenden Stationen und Bereiche weitergegeben. Davor sind nur die Ladungsträger markiert, wobei diese auch nur an bestimmten Orten gescannt werden. Zusätzlich gibt es zur eindeutigen Identifizierung Ritzprägenummern in den Einzelteilen. Diese sind jedoch bisher nicht digital erfasst. Grundsätzlich überwacht das Logistiksystem nur Transportwege und nicht Wege innerhalb der Produktion. Das bedeutet Stempelzeiten beziehen sich immer nur aufs Hochregallager und nicht auf die Ankunft oder gar die Verwendung in der Produktionsanlage. Für die Modellbildung müssen diese „Lücken“ überwunden werden.

4.2 Schätzungen von Parametern

Nachdem die Unsicherheiten identifiziert sind, müssen für das Modell die Parameter bestimmt werden. Da die handelsüblichen Produktions- und Logistikbegleitenden Systeme nicht alle benötigten Daten bereitstellen, werden im Folgenden drei Methoden vorgestellt, durch dessen Kombination die Parameter abgeschätzt werden können. Diese drei Methoden wurden vor allem durch ihre Einfachheit, Kosteneffizienz und schnelle Umsetzbarkeit bei dennoch ausreichender Genauigkeit gewählt. Sie sind daher vor allem für etablierte Prozesse anwendbar, in denen innovative Prozesssteuerungen eingeführt werden sollen, bevor der lange Weg über die Schnittstellenvereinheitlichung und die IT-Umstrukturierung stattgefunden hat.

Messen von Hand

Für die erfolgreiche Nutzung der Handmessmethode gibt es in verschiedenen Arbeiten Anhaltspunkte (vgl. z.B. REFA [7]). Um statistisch sichere Aussagen zu treffen, werden mit Hilfe von Methoden der REFA benötigte Stichprobengrößen definiert. Grundsätzlich können so Ladungsträger händisch verfolgt und die durchschnittliche Transportzeit ermittelt werden. Das Handmessverfahren eignet sich für vollständig sichtbare Prozesse oder für Prozesse mit sichtbarem Start- und Endpunkt. Oft reichen schon 15 Wiederholungen der Beobachtung an unterschiedlichen Tagen und Zeiten, um die Spannbreite des Prozesses abzuschätzen.

Verwendung von Erfahrungswerten

Für sehr etablierte Prozesse mit langjährigen Mitarbeitern kann auch der Erfahrungsschatz hilfreich sein.

Hier ist auf den Wahrheitsgehalt durch die Stichprobengröße zu achten, da über die Jahre eine subjektive Meinung des Einzelnen in die Aussagen einfließt.

Durchschnittswert durch digitale Fabrik

Simulationsergebnisse sind immer höchstens so genau wie das zu Grunde liegende Modell und dessen Eingangsdaten. Dies gilt auch für die digitale Fabrik. Diese bildet die Produktion im Entwicklungsstand weit vor Serienstart ab. Bis hin zum Serienanlauf verändern sich jedoch oft noch wichtige Parameter. Da diese Modelle nicht nachträglich aktualisiert werden, sondern nur zu einer ersten Abschätzung im frühen Stadium dienen, weichen sie oft vom realen Endstand ab. So sind die Ergebnisse von diesen Materialflusssimulationen auch später oft nur Anhaltspunkte.

Alle erhaltenen Ergebnisse müssen abschließend auf Verwendbarkeit geprüft werden. Um die Methoden weiter anzuwenden, muss eine Prüfung der Verteilungsform bzw. vorab eine Abschätzung durchgeführt werden. Die meisten Fertigungs- und Logistikprozesse lassen sich mit einer Normalverteilung beschreiben. Wird die Verteilungsform bestätigt, sind so ohne tiefergehende Beachtung der Unsicherheiten Vorhersagewahrscheinlichkeiten der Materialflussprozesse auszugeben. Die Unsicherheiten spielen automatisch in die Verteilungsform ein. Daher ist es wichtig, dass der Prozess stabil läuft, und sich nicht mehr in der Anlaufphase befindet, da die Verteilungsform aus einer Stichprobe des Prozesses zu bestimmen ist. Nur ein bestehendes Muster ist auf eine bestimmte Verteilungsform hin zu extrapolieren. In der Literatur finden sich hierzu einige Tests zur Überprüfung der Verteilungsform einer Menge von Werten [8]. Im vorliegenden Materialfluss, auf dem die Studie basiert, liegen Normalverteilungen vor. Abbildung 9 zeigt schematisch den Weg eines Bauteils zur Anlage. Am Anfang steht der Auftrag, das Teil von A nach B zu transportieren. Der darauf startende Transport ist mit Unsicherheiten behaftet. Die zeitliche Ausdehnung des Transports wird über den berechneten Mittelwert und die Verteilung beschrieben. Nach dem Transport verharren die Teile im Puffer bis zum Verbau. Auch die Zeit im Puffer wird durch eine Normalverteilung beschrieben. Über die Aufsummierung der zeitlichen Geschehnisse nach dem zentralen Grenzwertsatz ist auch der Zeitpunkt des Verbau mit einer Normalverteilung weiter zu geben.

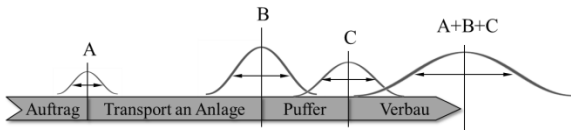


Abbildung 9. Schematische Darstellung der möglichen zeitlichen Verschiebung auf Grund von Unsicherheiten im System

5 Materialfluss vorhersagen

Nachdem die Grundzüge des Materialflusses geklärt sind, werden die Prozesse auf Bauteilebene erarbeitet. Hierbei spielen die Abrufzeiten und die Chargenumfänge eine Rolle. Eine Charge bezeichnet einen zusammenhängenden Presswerksauftrag eines Einzelteils. Produktionsaufträge werden über die benötigten Rohteile im Presswerk eingepflegt. Sobald die gelieferte Chargengröße im Hochregallager bekannt ist, wird die Reichweite, also der Zeitraum, in dem die Charge verbaut wird berechnet, abhängig vom vorhergesagten Startzeitpunkt. Die Chargengrößen schwanken je Abpressung. Außerdem schwankt das mit Hilfe der Taktzeit berechnete Ende der Reichweite der jeweiligen Chargen zusätzlich um die Störzeiten der Anlage. Abbildung 10 beschreibt diesen Sachverhalt. Die Chargenreichweite von Charge A schwankt um den vorhergesagten Zeitpunkt mit unterschiedlichen Wahrscheinlichkeiten. Die Chargenreichweite von Charge B kann auf Grundlage dessen im Voraus mittels der Unsicherheiten von Charge A vorausberechnet werden. Sobald Charge A seinen „wahren“ Chargenwechselzeitpunkt zu Charge B (bzw. seinen Erstabruf aus dem Hochregallager) zurückmeldet, wird der Chargenwechsel von Charge B zur nachfolgenden Charge genauer berechnet und verschiebt sich dementsprechend. Insgesamt ist das System als selbstoptimierendes System aufgebaut. Es werden so immer mehr Unsicherheiten ausgeschlossen, so dass nur noch die nicht vermeidbaren Unsicherheiten, wie Störfälle im Modell vorkommen. Beim Bauteil-Matching auf Zusammenbauebene hingegen genügt die Information über den Umfang der Menge („Charge“), die sich in einem Ladungsträger befindet. Da es hier keine Presswerk-Chargen mehr gibt, werden so „virtuelle Chargen“ erstellt, die von der Menge her immer gleich bleiben. Es wird somit berechnet, wann der fertigbeladene Ladungsträger an der nachfolgenden Anlage wieder entladen wird. Dementsprechend müssen dann die Ladungsträger passend zur Anlage losgeschickt werden, um verbaut zu werden. Dadurch können Pufferbestände verringert werden oder Bauteile sogar vorab sortiert

werden, um dann just-in-time zur Anlage geschickt zu werden, sodass ein Bauteil-Matching möglich wird.

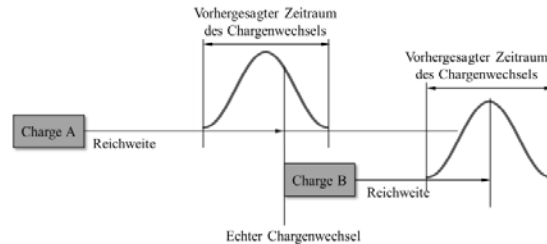


Abbildung 10. Schematische Darstellung der Chargenreichweite

Für die Vorausberechnung der Chargenstarts und -enden müssen die Chargenbezeichnung zur Identifizierung, der tatsächlich vom Presswerk gelieferte Umfang oder die Größe der Ladungsträger, sowie die berechnete bzw. nachträglich erfasste Startzeit der Charge abgegriffen werden. Darüber hinaus müssen im Modell standortspezifisch Parameter festgelegt werden, die erfassen, wie die Anlage arbeitet (Mehrschichtbetrieb, Pausen...etc.) und welche Feiertage oder Werksferien beachtet werden müssen.

Abbildung 11. Eingabe Maske des in VBA programmierten Materialflusssimulationstool

Die Realisierung des Modells ist mit Hilfe von VBA Programmierung möglich. Dies soll am Beispiel der Heckdeckelfertigung beschrieben werden. Zur Programmierung wurde ein auf drei Großteile (Beplankung oben, Beplankung unten und das Innenteil) vereinfachtes Modell zu Grunde gelegt. Das Tool wurde so vorbereitet, dass sich der Werker über eine Eingabemaske (Abbildung 11) eine Charge auswählen kann, für die er sich den Verwendungszeitraum und die parallel zu verbauenden Chargen anzeigen lassen möchte (Abbildung 12). Oder er wählt ein Datum und lässt sich anzeigen, welche Chargen zu diesem Zeitpunkt verbaut werden (Abbildung 13). Hinter diesen Ein- und Ausgabemasken liegen die Kalkulationstabellen mit den materialflussspezifischen Parametern, wie Transportzeitverteilung oder Feiertage und weitere Daten, die oben beschrieben wurden.

Bauteil:	A222 757 0009	
Charge:	2014_1D	Enddatum
	2014_1D	03.02.2014 00:00
0010	2014_1E	06.02.2014 00:00
	2014_1D	04.02.2014 00:00
0109		

Abbildung 12. Ausschnitt der Ergebnisausgabe nach Chargenfilterung

A222 757 0009 Charge	gel. Umfang	Startzeit	Endzeit
2014_3E	2886	01.04.2014 09:19	09.04.2014 00:00

A222 757 0010 Charge	gel. Umfang	Startzeit	Endzeit
2014_3G	2896	01.04.2014 03:07	09.04.2014 00:00

A222 757 0109 Charge	gel. Umfang	Startzeit	Endzeit
2014_3E	1766	03.04.2014 08:37	08.04.2014 00:00

Abbildung 13. Ausschnitt der Ergebnisausgabe nach Einsatzdatumfilterung

6 Bauteile matchen

Nachdem das Simulationsmodell des Materialflusses aufgestellt ist, sollen die Simulationsergebnisse entsprechend verwertet und für die Produktionsprozesse genutzt werden. Für eine sinnvolle Nutzung müssen die verschiedenen Einzelteile oder Zusammenbauten mindestens stichprobenartig vermessen und die Daten hinterlegt werden. Der Heckdeckel besteht aus sieben Einzelteilen, wobei das formstabile Innenteil sehr maßgebend ist [9]. Der Zusammenbau Heckdeckel ist nach dem Falzprozess eindeutig in seiner Maßhaltigkeit definiert und nicht mehr veränderbar. Mittels der Heckdeckelöffnung bilden sich dann die Fugenwerte. Bereits vorher muss auf die Verringerung der Fugenmaße geachtet werden. Es gibt an dieser Stelle zwei Möglichkeiten der Einflussnahme. Einerseits kann schon mit dem Chargenmatching der Zusammenbau beeinflusst werden. Wenn aktuell in der Karosseriefertigung größere Heckdeckelöffnungen produziert werden, wird eine Charge mit eher schmalen Einzelteilen zurückgehalten und erst wieder freigegeben, wenn die Produktion der Karossen wieder im gewünschten Rahmen läuft. Da im Voraus berechnet werden kann, ab wann die nächste Charge verwendet wird, kann abgeschätzt werden, welche Maßhaltigkeiten die produzierten Karosserien aufweisen werden.

Andererseits ist aber auch eine Einflussnahme erst auf Zusammenbauebene denkbar. Der Produktionsprozess ist in Abbildung 8 beschrieben. Danach werden die fertigen Heckdeckel auf Ladungsträger platziert und zur Anlage gebracht, wo die Heckdeckel an die Karosserien verschraubt werden. Dieses Aufladen auf Ladungsträger kann hier bereits sortiert gesche-

hen. So werden Ladungsträger mit verhältnismäßig breiten Heckdeckeln und Ladungsträger mit Heckdeckel mit verhältnismäßig schmalen Abmaßen zusammengestellt. Durch das Materialflusssimulationstool kann nun der Zeitpunkt vorausgesagt werden, wann dieser Ladungsträger an der Anlage ankommen wird bzw. muss. Zu diesem Zeitpunkt müssen dann passende Karosserien mit jeweils schmälere oder breitere Heckdeckelöffnungen vorbei kommen, so dass eine gute Kombination erzeugt werden kann. Die Potenziale so eines Bauteil-Matchings sind im Folgenden beispielhaft aufgezeigt.

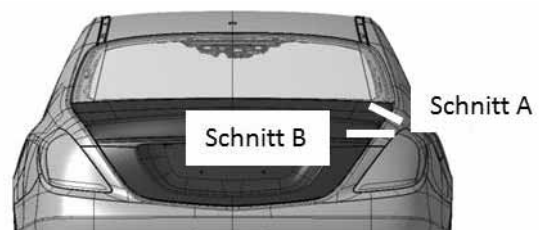


Abbildung 14. Schematische Darstellung der Positionen von Schnitt A und B für Abbildung 12 und 13

Abbildung 14 zeigt beispielhaft ausgewählte Messschnitte einer Fugenmaßmessaanlage. Abbildung 15 und 16 zeigen die jeweiligen Fugenmaße von 10 verschiedenen Fahrzeugen (gestrichelt). Die durchgezogenen Linien zeigen die möglichen Potenziale der Streuungsverringerung der Fugenbilder mittels Bauteil-Matching dieser 10 Fahrzeuge bzw. der Heckdeckel untereinander. Beide Graphiken sind aufs Fugensollmaß normiert.

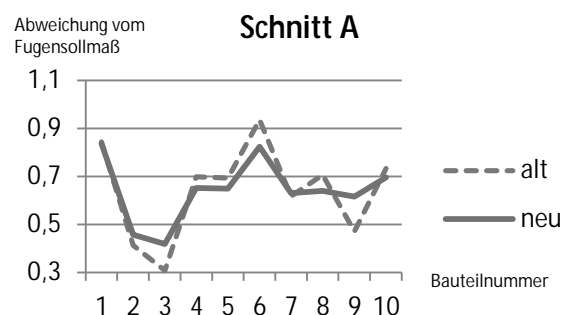


Abbildung 15. Fugenbild an Schnitt A

Es zeigt sich, dass die Streuung im Bereich von Schnitt A um 30 % verringert werden kann und im Bereich von Schnitt B um 20%.

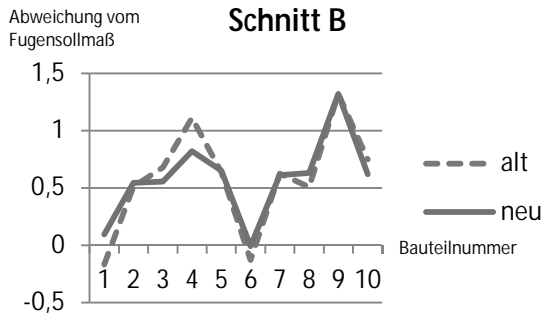


Abbildung 16. Fugenbild an Schnitt B

Abschließend sei nochmal darauf hingewiesen, dass die dargestellte Beispielanalyse nur einen kleinen Ausschnitt der Möglichkeiten des Bauteil-Matchings und dessen Potenziale mit vorgelagerter Materialflusssimulation zur Veranschaulichung der Anwendbarkeit darstellt.

7 Zusammenfassung und Ausblick

Die hier vorgestellte Methode der prognostizierenden Materialflusssimulation ist eine innovative Methode zur Vorhersage und Optimierung von Produktionsprozessen mit einem hohen Potenzial und deutlichen Vorteilen gegenüber herkömmlichen Produktionsprozessoptimierungen. In der vorliegenden Arbeit werden für den Karosseriebau an Hand des Beispiels der Heckdeckelfertigung die Vorteile der produktionsbegleitenden Materialflusssimulation mit terminierter Vorhersage für das Bauteil-Matching aufgezeigt. Die vorgestellte Methode lässt sich ohne Schwierigkeiten auf andere Anlagen oder sogar andere Industriezweige übertragen und erfordert geringe Investitionen.

Die Validierung des nach dieser Methode erstellten Tools zur Vorhersage der Materialflussprozesse im vorliegenden Projekt der Großserienfertigung in der Automobilindustrie bestätigte die Vorhersagen und damit das Modell. Eine weitere Minimierung der im Modell enthaltenen und beschriebenen Unsicherheiten kann darüber hinaus durch eine bessere Kommunikation zwischen den Stationen erreicht werden. Dies erfordert jedoch größere Investitionen, die sich allerdings auch nur lohnen, wenn darauf aufbauend auch die Produktion auf automatisiertes Bauteil-Matching umgestellt wird. Diese „Smart-Factories“ würden noch größere Investitionen und Zeit benötigen und sind daher nur schwer umsetzbar. Im vorliegenden Beitrag wurde gezeigt, dass stattdessen auch schon mit kleinen Investitionen ein großer Schritt in

die Richtung innovativer Fertigungsoptimierung und -steuerungen bestritten werden kann.

8 Quellen

- [1] J.Schwarze. *Kundenorientiertes Qualitätsmanagement in der Automobilindustrie*. Dissertation, Gabler Edition Wissenschaft, ISBN 3-8244-7778-5, 2003.
- [2] K. Hetsch, M. Bohn. *Toleranzmanagement im Automobilbau*. Hanser, München, ISBN 978-3446434967, 2013.
- [3] B. Klein. *Toleranzmanagement im Maschinen- und Fahrzeugbau*. Oldenbourg Wissenschaftsverlag, ISBN 978-3486578502, 2005.
- [4] Bundesministerium für Bildung und Forschung. *Hightech-Strategie für Deutschland*. online verfügbar unter <http://www.hightech-strategie.de/de/59.php>, Abgriff am 29.12.2013.
- [5] H. Prof. Dr. Kagermann. *Industrie 4.0 in der praktischen Anwendung – Erfahrungen und Erwartungen*. Industriekonferenz Brandenburg, Zentrum für Luft- und Raumfahrt, Deutschland, 2013.
- [6] H. Gerpott. *F&E und Produktion: theoretische und empirische Analysen zu Schnittstellenproblemen im Innovationsprozeß unter besonderer Berücksichtigung personalwirtschaftlicher Aspekte*. München, Zeitschrift für Personalforschung 5. Jahrgang, 1991.
- [7] REFA – *Sonderdruck Methodenteil 2, ausgewählte Methoden zur prozessorientierten Arbeitsorganisation*. REFA, 1. Auflage, 2002.
- [8] P. Zöfel. *Statistik verstehen, Ein Begleitbuch zur computergestützten Anwendung*. Addison-Wesley Verlag, ISBN 978-3827316905, 2000.
- [9] A.Beckmann, M. Bohn und P. Gust. *Tolerance Simulation in the Assembling Process based on Experimental Data from Series Production*. 13th CIRP Conference on Computer Aided Tolerancing, 2014.

Ich danke meinen studentischen Mitarbeitern Max Riekert und Lucas Benjamin Wienke für die unter meiner Betreuung geleisteten Arbeiten, die im Zusammenhang mit den Inhalten diesen Beitrags stehen.

Modelling Obesity in Austria with System Dynamics

Barbara Glock¹, Patrick Einzinger¹, Felix Breiteneker²

¹dwh simulation services, dwh GmbH, Vienna, Austria

²Institute for Analysis and Scientific Computing, Vienna University of Technology, Vienna, Austria

barbara.glock@dwh.at

More than one third of the Austrian population is obese or overweight. Not only the adult population suffers from this chronic disease, but also children. Therefore a modular system dynamics model is built that simulates the Austrian population and the development of the prevalence of obesity over time from 1999 until 2050. Furthermore, three interventions are tested: 1.) a reduction of caloric intake of 80 kilocalories per day, 2.) a reduction of caloric intake of 160 kilocalories per day and 3.) a reduction of caloric intake of 80 kilocalories per day together with an increase of physical activity that is equivalent to an expenditure of 80 kilocalories per day. All interventions show a decrease of the prevalence of obesity. Additionally intervention 1 and 2 are compared, showing that a reduction of the doubled amount of caloric intake (intervention 2) results in a reduction of the prevalence of obesity that is less than twice as much as that in intervention 1. Intervention 2 and 3 are compared, showing that an increase in physical activity increases the prevalence of obesity more than a decrease of caloric intake of the same caloric value. A probabilistic sensitivity analysis for validation is done as well.

1 Introduction

Obesity, defined as an enormous amount of body tissue that also may impair health, is in Austria especially among children, a health concern of paramount importance [1]. This chronic disease, measured by the body mass index (BMI), which is calculated by dividing the body mass in kilogram through the square of body height in centimetre, as seen in equation 1, has a long list of co-morbidities, which affect each person and also the health care system.

$$BMI = \frac{Bodyweight [kg]}{(Bodyheight [cm])^2} \quad (1)$$

The classification of the BMI according to World Health Organisation WHO [2] can be seen in table 1 and is valid for the adult population. There is a different classification for children.

Classification	BMI
underweight	< 18,50
normal weight	18,50 – 24,99
overweight	25,00 – 29,99
obese	≥ 30,00

Table 1: BMI-categories according to WHO.

There are three further specifications of the BMI-category *obese*.

These are important if surgeries are modelled too, because a surgery is usually performed with a BMI greater than 40.

There is a correlation of eating and physical activity habits of children to that of their parents. According to a dissertation of the University of Trier [3] a child learns from the parents by the message “Clear your plate!” that this is the amount of food that has to be consumed, instead of listening to the body’s feeling of satiety. Therefore, a connection of eating habits and physical activity habits from parents to their children is implemented as well.

There are three interventions that are tested with the simulation model:

1. A reduction of the caloric intake of 80 kilocalories per day in the adult obese and overweight population
2. A reduction of the caloric intake of 160 kilocalories per day in the adult obese and overweight population
3. A reduction of the caloric intake of 80 kilocalories per day and an additional increase in physical activity by an increase of caloric expenditure of 80 kilocalories per day in the adult obese and overweight population

Furthermore, it is researched if a reduction of the doubled amount of caloric intake reduces the preva-

lence of obesity twice as much (comparison of interventions 1 and 2). Also the effect of physical activity in comparison to the effect of a reduction of caloric intake is evaluated (comparison of intervention 2 and intervention 3).

2 Method: System Dynamics

System dynamics is a top-down modelling technique that was developed in 1951 by J. Forrester [4] and it is a graphic notation for ordinary differential equations. The population is divided into *stocks*. These stocks represent the state of the system. Each state can be changed by *flows* going in or out of the stocks. Furthermore, there are *parameters* (constants) and *auxiliaries*, which are used for algebraic equations. *Sources* and *Sinks* are the boundaries of the system.

A simple stock and flow diagram is shown in figure 1 depicting a stock together with an inflow and a parameter that regulates the amount of the inflow. The flow is in some way dependent on the stock, shown by the arrow pointing from the stock to the valve of the flow.

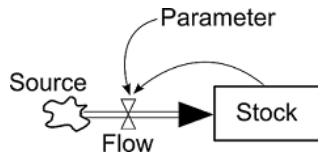


Figure 1. A simple stock and flow diagram.

Each stock represents a differential equation in accordance to time t . The equations for the stock and the flow can be stated as shown in equation 2 and 3.

$$\text{Stock}'(t) = \text{Flow}(t) \quad (2)$$

$$\text{Flow}(t) = \text{Parameter} \cdot \text{Stock}(t) \quad (3)$$

These differential equations can also be written as integrals for each stock together with an initial value of the stock at time t_0 .

3 Model

The model structure is set up modular and consists of a population model that simulates the demographic development of the population and of a disease model that simulates the development of obesity over time, from 1999 to 2050. The advantage of this modular architecture is on the one side that single model parts can be scanned easier for errors than one big complex model. On the other side these model parts can be reused for other purposes or models by simply cou-

pling them through defined interfaces. The two independent model parts that are implemented here are connected by a special defined interface, which will be described later on.

3.1 Population Model

The population is divided into 192 stocks for 96 age classes and 2 sexes. The stocks, representing parts of the population, are changed by births, deaths, migration and aging as show in figure 2 for the first two age-classes for men and women.

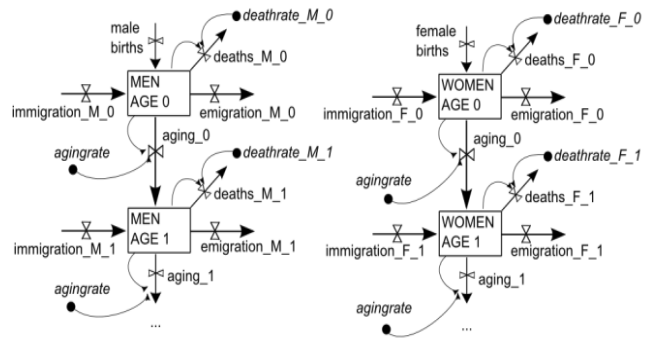


Figure 2. Part of the structure of the population model.

The death rates are dependent on the population (e.g. see link from stock to $deaths_M_0$), whereas migration is given as an absolute number in each flow and is not dependent on the size of the population. Births are calculated by multiplying the one-year age-specific fertility rates with the stocks of fertile women (aged 15 to 45). This calculation is not shown in this figure, except for the already calculated total births into the stocks at age 0.

The data for death rates r_s is calculated out of the by Statistics Austria [5] provided death probability p_s for each age-class, sex and simulated year, as seen in equation 4.

$$r_s = -\ln(1 - p_s) \quad (4)$$

For the years 2010 until 2050 prognosticated death probabilities, also provided by Statistics Austria, are used.

The birth rates are given as one-year age-specific fertility rates, being the number of life-births per 1.000 women for the years 1999 until 2010. For the forecast years 2011 until 2050 there are only the estimated *total fertility rate*, being the number of children a women would birth in her life, if there are the same fertility conditions in her future life as they are now, and the estimated *average fertility age* available. These are used to calculate an estimation of the one-

year age specific fertility rates, which are used as birth rates for the forecast years.

Data for migration is also provided and estimated by Statistics Austria. For the years 1999 until 2010 there is gender-specific data for 5-year age-classes available. These are uniformly distributed, split up in 1 year age classes and used for immigration as net-in- and for emigration as net-out-flows in the simulation. For the forecast years 2011-2050 there is only one value, aggregated for sex and age, for each year available. This value is split into age- and sex-classes in such a manner as the average distribution is in 1999 to 2050.

3.2 Disease Model

For the disease model the population is divided into 10 age-classes, 2 sexes and the 4 BMI-categories as shown in table 1.

The demographic development is covered by the population model. In the disease model only the changes within one age-class and sex between adjacent stocks (according to the BMI-category) are allowed. This means that obese people can only change into the normal weight stock by passing the overweight stock, as seen in figure 3.

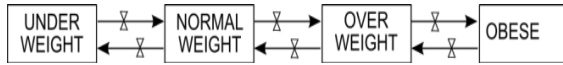


Figure 3. Flows in the disease model between the stocks within one age-class and sex

The rates, regulating the amount of the flows, stay constant in the base run and change in 2010 if an intervention is tested. An intervention only calculates *caloric* changes. Therefore, the core part of the disease model is the translation of *changes in energy balance* within a stock to *changes of rates* between the BMI-categories between the stocks.

The energy balance is calculated by energy intake minus energy expenditure. The Energy intake is calculated by the kilocalories taken in by eating food or drinking beverages. The energy expenditure EE is calculated as shown in equation (5).

$$EE = BMR + TEF + PA \quad (5)$$

The basal metabolic rate (BMR) depicts the energy the body needs to maintain the basal functions each day at indifference temperature of 28 degrees Celsius and with empty stomach. The estimation of the BMR requires age, weight, height and gender of the person. The equations used in the model are the *Revised Har-*

ris-Benedict equations [6] and are shown in equations 6 and 7.

These equations were originally developed in 1919 by J.A. Harris and F.G. Benedict and served as an adequate estimation of the basal metabolic rate for a long time. In 1984 these equations were revised when new measurements occurred.

$$\begin{aligned} BMR_{Men} = & 88,362 \text{ kcal} \quad (6) \\ & + \left(13,397 \frac{\text{kcal}}{\text{kg}} \cdot \text{weight} [\text{kg}] \right) \\ & + \left(4,799 \frac{\text{kcal}}{\text{cm}} \cdot \text{height} [\text{cm}] \right) \\ & - \left(5,677 \frac{\text{kcal}}{\text{year}} \cdot \text{age} [\text{years}] \right) \end{aligned}$$

$$\begin{aligned} BMR_{woman} = & 447,593 \text{ kcal} \quad (7) \\ & + \left(9,247 \frac{\text{kcal}}{\text{kg}} \cdot \text{weight} [\text{kg}] \right) \\ & + \left(3,098 \frac{\text{kcal}}{\text{cm}} \cdot \text{height} [\text{cm}] \right) \\ & - \left(4,330 \frac{\text{kcal}}{\text{year}} \cdot \text{age} [\text{years}] \right) \end{aligned}$$

The factors for weight and height are for men larger than for women resulting in a larger BMR for men than for women. The thermic effect of foods (TEF) is the energy that is used to process the food a person eats (usually 10% of the value of caloric intake). Physical activity (PA) also contributes to the energy expenditure.

Each stock in the disease model has an *average* energy balance, calculated as mentioned above. The energy balance in 1999, at simulation start, is given and then used to calculate the *change* of energy balance throughout the simulation. This *change of energy balance* (ΔK) is then transformed into a *change of rates* (ΔF) between the BMI-categories as seen in equation 8.

$$\Delta F = \text{MAX} \left\{ 0; 0,5 \cdot \frac{\Delta K}{(b_c - b_m) \cdot (h_m)^2 \cdot k} \cdot \frac{365}{12} \right\} \quad (8)$$

This equation was adapted from a paper by Homer et al. [7] and includes the BMI-cut-off points b_c to the adjacent BMI-category, the average BMI b_m in each category, the average height h_m in each sex-, age-class- and BMI-category-specific stock and furthermore, an empiric estimation of the average kilocalories per kilogram of weight-change k (8.050 kcal/kg) according to Forbes [8].

As mentioned before, there is a correlation between energy intake and energy expenditure from parents to that of their children. Therefore, a weighted average of energy expenditure, respectively energy intake, over all parents-ages (19-64 years) and severity degrees is calculated. The percentage of the caloric change during simulation to the initial value in 1999 is then translated to a change in the energy expenditure, respectively energy intake, within the age-classes of the children (0-18 years). For example, if the energy intake of parents increases for 2% then the energy intake of children will increase for 2% as well. This *average* over all age-classes and BMI-categories is used, because there is no data on how many adults, categorized by their BMI, actually do have children at a specific age is available. The weighting of this average value is done in a manner that “younger” parents with probably “younger” children have a greater influence on their children than “older” parents do have on their “older” children. Furthermore, the percentage of, for example, overweight parents having overweight children is not known as well.

There are two kinds of rates between the BMI-categories, the *upflow rate*, calculating the change of stocks from a lower BMI-category to a higher BMI-category, and the *downflow rates*, calculating the change of stocks from a higher BMI-category to a lower BMI-category (see figure 3). During the simulation these rates are changed in a manner that was mentioned before by matching the change of energy balance to a change of rates. This means that an *initial* value of the rates has to be provided. There was no data on changes of weight within a year for the Austrian population available that would describe weight gaining or weight losing. Therefore the downflow rates are estimated out of data from the *National Health and Nutrition Survey* (NHANES) [9] from the USA. This dataset includes person-specific data for age, weight, height, sex and, most important, for the estimated weight one year ago. Out of this data the BMI and the BMI one year ago can be calculated and used for an estimation of the rates for a downwards BMI-category change. The upflow rates in the model were then calibrated. A dataset depicting the state of the system, meaning the distribution of BMI-category within the population in 2006/2007 [10], was used for minimizing the objective function, the difference between the simulated prevalence of obesity in Austria and the actual prevalence in 2006/2007.

3.3 Interface

The population model and the disease model are connected by an interface that calculates the *changes* of the stocks due to births, aging, migration and deaths in the population model and transfers them to the stocks of the disease model. The comparison of the structure of these two model parts is shown in table 2.

	Population model	Disease model
age-classes	96	10
sex	2	2
BMI-category	--	4

Table 2: Structure comparison of the population model and the disease model.

The changes of the stocks in the population model are for each sex aggregated according to the age-classes in the disease model. Then they are split up into the stocks of the BMI-categories in the disease model within this age-class and sex, as shown in figure 4 for the first and the last age-class of men in the disease model.

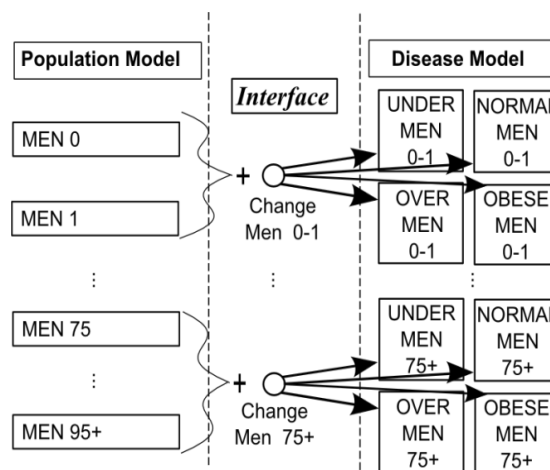


Figure 4: Interface between population model and disease model.

This is done in a manner that the distribution of the BMI-categories of the changes from the interface show the same distribution of BMI-categories as in the disease model: If there are 50% people normal weight and 20% overweight in the disease model at this time, then the changes of the stocks of the population model are split up and 70% are transferred to the stock of normal weight people and 20% are transferred to the stock of overweight people.

In this model there is no connection from the disease model back to the population model, since no obesity-related deaths are simulated. The reason for that is that there are no direct obesity related deaths. Usually no one dies because of overweight, but sometimes because of obesity associated diseases, like problems with the heart. The modular setup allows coupling other model parts simulating co-morbidities. If such a model part, simulating an obesity associated disease, is connected, it is possible to redefine the interface or to define an interface from that model part to the population model and then allow information-flows to the population model and also simulate disease related deaths.

4 Results

The base run shows an increase in the prevalence of obesity and overweight in 2050. For example there are 338.110 obese male aged 19-64 in 2050, whereas in 1999 there were 217.983 obese male people. Within the female adult population the number increased from 198.933 in 1999 to 326.361 in 2050. Of course this result includes the demographic development as well, but there is still an increase of the prevalence of obesity within the relative numbers.

4.1 Intervention 1: Reduction of caloric intake

If there is a reduction of the caloric intake for 80 kilocalories per day starting in 2010 within the adult obese and overweight population, there is a reduction of the prevalence of obesity and overweight in 2050. The comparison of base run and intervention 1 is shown in figure 5 for the obese population (for males and females together).

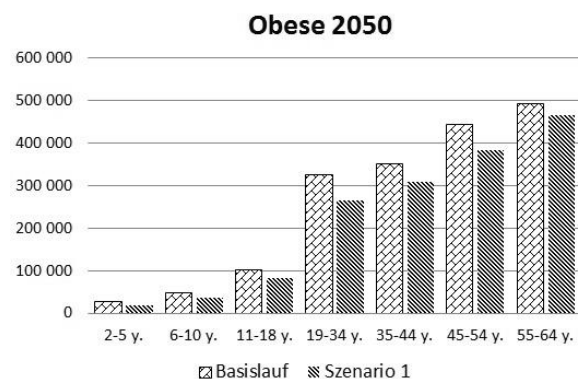


Figure 5. Comparison of the results from base run to intervention 1 (reduction of caloric intake for 80 kilocalories per day).

The results are depicted for some age-classes and the vertical axis shows the absolute number of obese people in 2050.

4.2 Intervention 2: Doubled reduction of caloric intake

In intervention 2 the caloric intake is reduced for 160 kilocalories per day in the obese and overweight adult population. Due to the fact that a correlation between adult eating behaviour and that of children is implemented, a decrease of the caloric intake in adults also reduces the caloric intake within children.

The prevalence of obesity and overweight reduces as well as in intervention 1, but the effect of the reduction compared to the effort of reducing the caloric intake for twice as much, is for intervention 2 smaller than for intervention 1, as seen in table 3.

In intervention 1 the prevalence of obesity was reduced for 32% within children, whereas in intervention 2 only a reduction of 44% can be shown, both compared to the base run. This effect is not twice as much as the effort of reducing the doubled amount of caloric intake was.

	Intervention 1 vs. Base run	Intervention 2 vs. Base run
children 2-18 years	-32%	-44%
adults 19-64 years	-18%	-31%

Table 3: Comparison of the reduction of the prevalence of obesity due to intervention 1 to that due to intervention 2.

This means that the effect of a decrease of the prevalence of obesity, and it is the same for overweight, reduces relatively to the effort.

4.3 Intervention 3: Reduction of caloric intake together with an increase in caloric expenditure

In intervention 3 the caloric intake is reduced for 80 kilocalories per day for the obese and overweight population together with an increase of physical activity of the amount of 80 kilocalories of energy expenditure per day. This means that each day people have 160 kilocalories less than they have in the base run. This looks like it is the same reduction of calories as in intervention 2. A comparison of intervention 2 and 3 show that the prevalence of obesity in intervention 3 is reduced as well, but, as seen in figure 6, additional physical activity instead of eating less has

a larger effect on the prevalence of obesity. The prevalence of obesity in intervention 3 is slightly more reduced than the prevalence of obesity in intervention 2.

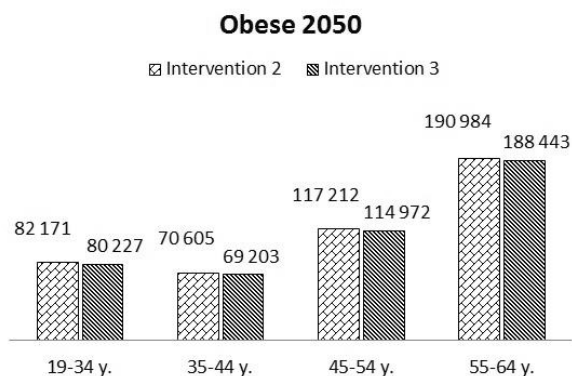


Figure 6: Comparison of intervention 2 and intervention 3 for the obese population and some age-classes.

The reason why intervention 3 has a better effect is the thermic effect of foods: by eating more in intervention 3 the thermic effect of foods is higher than that in intervention 2 and therefore the energy balance is reduced slightly more by the thermic effect of foods, although the overall reduction of kilocalories, by physical activity or eating less, is in both interventions the same.

4.4 Validation: Probabilistic sensitivity analysis

A probabilistic sensitivity analysis is done by generating 400 sets of parameters for the up- and downflow rates, which are the parameters with the biggest uncertainty due to the fact that they are partially calculated out of the NHANES study and partially calibrated. A normal distribution is used with the original value of each parameter as expected value. The comparisons of the base runs to the interventions for all sets of parameters show that in each case the intervention results in a decrease of the prevalence of obesity and overweight.

5 Conclusion

Each intervention leads to a decrease of the prevalence of obesity and overweight. It can be shown that a reduction of caloric intake is effective, but a further reduction is not that effective as the reduction at the beginning. Furthermore, due to the thermic effect of foods a combined treatment with physical activity and a reduction of caloric intake is more effective than simply reducing the food intake.

Furthermore, the modular setup allows a high degree of reusability of model parts. For example the popula-

tion model can be used in another model where the demographic development of a population is required. The parameterization can be changed to that of another population, if another population is researched. Obesity is a disease with a long list of comorbidities, like diabetes or hypertension, and the modular setup allows an integration of other model parts that can be connected through another interface. This also means that other influences can be modelled as well. A cost model, calculating the costs of the disease due to medication or surgery, can be connected as well.

This model makes qualitative statements for treatment strategies of obesity and overweight and provides a high degree of reusability for model parts due to the modular setup.

6 References

- [1] A. Rieder. *Erster österreichischer Adipositasbericht 2006 - Grundlage für zukünftige Handlungsfelder: Kinder, Jugendliche, Erwachsene*. Wien, 2006.
- [2] World Health Organization (WHO): <http://www.who.int/topics/obesity/en>. Access: April 2010.
- [3] B. Lindel. *Stressbezogene Einflussfaktoren auf das Essverhalten übergewichtiger Kinder*. Dissertation, Universität Trier, Juli 2002.
- [4] J. Forrester. *Industrial dynamics*. Cambridge, MA, 1961.
- [5] Statistics Austria: <http://www.statistik.at/webde/statistiken/bevoelkerung/index.html>. Access: 03. April 2010.
- [6] A.M. Roza, H.M. Shizgal. *The Harris Benedict equation reevaluated: Resting energy requirements and the body cell mass*. American Journal of Clinical Nutrition, Vol. 40, Nr. 1, p. 168-182, 1984.
- [7] J. Homer, B. Millstein, W. Dietz, D. Buchner, E. Majestic. *Obesity population dynamics: Exploring historical growth and plausible futures in the U.S.*. 24th International System Dynamics Conference, Nijmegen, July 2006.
- [8] GB. Forbes. *Deliberate overfeeding in women and men: Energy costs and composition of the weight gain*. British Journal of Nutrition, Vol. 56, Nr.1, p. 1-9. 1986.

- [9] National Health and Nutrition Examination Survey:
[http://www.cdc.gov/nchs/nhanes/nhanes questionnaires.htm](http://www.cdc.gov/nchs/nhanes/nhanes_questionnaires.htm). Access: 07th Nov. 2012.
- [10] Statistik Austria. *Österreichische Gesundheitsbefragung 2006/2007. Hauptergebnisse und methodische Dokumentation*. Wien 2007.

Die Darstellung der Compliance in der Modellierung des Herz-Kreislauf-Systems

Tamara Vobruba¹, Andreas Körner¹, Bernhard Hametner², Felix Breiteneker¹

¹Institut für Analysis und Scientific Computing, Technische Universität Wien

²AIT Austrian Institute of Technology GmbH, Wien

tamara.vobruba@tuwien.ac.at

Die mathematische Modellbildung und Simulation hat für die Forschung auf dem Gebiet des Herz-Kreislauf-Systems große Bedeutung und es haben sich im Laufe der Zeit verschiedene Modellierungsansätze entwickelt. In dieser Arbeit wird das bereits validierte statische Kompartimentmodell von Möller und Popovic implementiert, welches den Windkesselleffekt berücksichtigt. Man findet in der Literatur verschiedene Darstellungen für die damit im Zusammenhang stehende Compliance, die Elastizität der Gefäßwände. Es soll ein sinnvoller Ansatz herausgearbeitet, sowie dessen Auswirkung untersucht werden. Das Modell wird mit unterschiedlichen Darstellungen getestet, wobei sich herausstellt, dass trotz der komplexen Zusammenhänge eine konstante Größe für die Compliance herangezogen werden kann.

1 Einleitung

Da in vielen Ländern Erkrankungen des Herz-Kreislauf-Systems eine der häufigsten Todesursachen sind, ist die Bedeutsamkeit der Forschung auf diesem Gebiet sehr groß. Die Modellbildung und Simulation ist hier ein wichtiger Bestandteil. In dieser Arbeit wird ein Sechs-Kompartimentmodell untersucht, welches den Windkesselleffekt berücksichtigt. Ein Winkessel ist ein mit einer Pumpe verbundener Druckbehälter, der im Mittelalter zur Löschung von Bränden eingesetzt wurde. Der durch die Pumpe erhöhte Druck im Kessel sorgt für einen regelmäßigen Wasserstrahl und speichert gleichzeitig das Wasser. [1] Zu vergleichen ist dies mit dem Verhalten der großen Arterien, welche sehr dehnbar sind. Die Idee arterielle Gefäße durch den Windkesselleffekt zu beschreiben kam bereits Ende des 19. Jahrhunderts auf. [2] Die Compliance $C = \frac{\Delta V}{\Delta P}$ wird durch die Volumensänderung nach der Druckänderung beschreiben. Die Dehnbarkeit der Gefäße hängt von komplexen Mechanismen ab, weil die Gefäße mit wachsendem Innenradius ihre Volumenskapazität steigern können. [3] Es besteht also zwischen der Compliance der Gefäße und dem dort herrschenden Druck ein nichtlinearer Zusammenhang.

2 Das Kompartimentmodell

In dem Kompartimentmodell, welches in *Modeling, simulation and parameterestimation of the human cardiovascular system* von Dietmar Möller und Dobrivoje Popovic dargestellt wird, werden Druck und Fluss über eine Herzschlagperiode gemittelt. Das Herz wird durch ein Doppelpumpensystem beschrieben, wobei nur die Herzkammern modelliert werden und das System der Vorhöfe vernachlässigt wird. Es bilden also linke und rechte Herzkammer als Pumpe je ein Kompartiment. Weiters wird der Körperkreislauf und der Lungenkreislauf je in arterielles und venöses System unterteilt. Dadurch dass das arterielle System durch ein Kompartiment dargestellt wird, wird angenommen, dass hier überall der gleiche Druck herrscht und sich dieser damit mit unendlicher Geschwindigkeit ausbreitet. [4]

2.1 Das Herz

Eine grundlegende Überlegung ist die Herzkammer als dehnbaren Schlauch darzustellen. Es wird weiters der Frank-Starling-Mechanismus berücksichtigt, der den Zusammenhang zwischen Schlagvolumen und

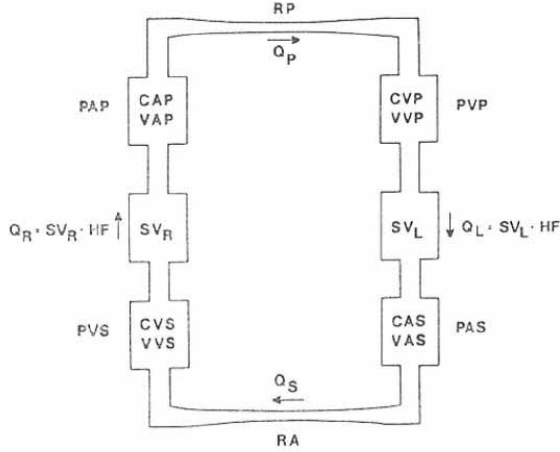


Abbildung 1: Darstellung der Kompartments aus [4]

enddiastolischem Volumen beschreibt.

In der Gleichung für den Fluss aus dem rechten Ventrikel Q_R fließen neben der Diastolendauer t_D , dem Widerstand R_R , der Ventrikelstärke K_R und der Compliance im Ventrikel C_R als Konstanten, der Druck im venösen systemischen Kompartiment PVS und im arteriellen pulmonalen Kompartiment PAP ein, welche sich durch das Lösen des Differentialgleichungssystems (3) ergeben.

$$Q_R = HR \frac{C_R \cdot PVS \left(1 - e^{\frac{-t_D}{C_R \cdot R_R}}\right)}{\frac{PAP}{K_R} \left(1 - e^{\frac{-t_D}{C_R \cdot R_R}}\right) + e^{\frac{-t_D}{C_R \cdot R_R}}} \quad (1)$$

$$Q_L = HR \frac{C_L \cdot PVP \left(1 - e^{\frac{-t_D}{C_L \cdot R_L}}\right)}{\frac{PAS}{K_L} \left(1 - e^{\frac{-t_D}{C_L \cdot R_L}}\right) + e^{\frac{-t_D}{C_L \cdot R_L}}}$$

2.2 Die Blutgefäße

Zur mathematischen Beschreibung des Drucks wird das Blut als inkompressibel und von konstanter Dichte angenommen. Damit kann die Volumensänderung im arteriellen systemischen Kompartiment durch die Bilanz von Zufluss und Abfluss dargestellt werden

$$\frac{dVAS}{dt} = Q_L - Q_S. \quad (2)$$

Es ergibt sich für die Drücke in den verschiedenen Kompartments ein lineares inhomogenes Differentialgleichungssystem, wobei die Compliance in den jeweiligen Kompartments CAS , CVS , CAP und CVP , sowie der periphere Widerstand RS beziehungsweise RP , berücksichtigt wird.

$$\begin{aligned} \frac{dPAS}{dt} &= \frac{1}{CAS} \left(Q_L - \frac{PAS - PVS}{RS} \right) \\ \frac{dPVS}{dt} &= -\frac{1}{CVS} \left(Q_R - \frac{PAS - PVS}{RS} \right) \\ \frac{dPAP}{dt} &= \frac{1}{CAP} \left(Q_R - \frac{PAP - PVP}{RP} \right) \\ \frac{dPVP}{dt} &= -\frac{1}{CVP} \left(Q_L - \frac{PAP - PVP}{RP} \right) \end{aligned} \quad (3)$$

3 Simulationsergebnisse

3.1 Simulationsergebnisse bei konstanter Compliance

Zunächst wird die Compliance als konstant angenommen und die in [4] angeführten Werte für $CAS_1 = 0.52 \frac{ml}{mmHg}$, $CAP_1 = 2 \frac{ml}{mmHg}$, $CVS_1 = 51.92 \frac{ml}{mmHg}$ und $CVP_1 = 4 \frac{ml}{mmHg}$ verwendet.

Variable	Wert	Einheit
PAS	117.78	mmHg
PVS	7.15	mmHg
PAP	17.17	mmHg
PVP	10.85	mmHg
Q_R	105.37	$\frac{ml}{s}$
Q_L	105.37	$\frac{ml}{s}$

Tabelle 1: Ergebnisse bei konstanter Compliance

Das Modell schwingt sich nach kurzer Zeit ein, was aus Abbildung 2 hervorgeht. In Tabelle 1 sieht man, dass die simulierten Werte bei einem Normaldruck in arteriellen systematischen Kompartiment von $80 - 120 \text{ mmHg}$ und einem Fluss von etwa $88 \frac{ml}{s}$ tendenziell, zu hoch sind. Dies ist ein Phänomen, das auch Yvonne Stickler bei der Simulation nach Möller und Popovic anführt. Sie führte weiters eine Validierung ihrer Ergebnisse mittels eigener Messdaten

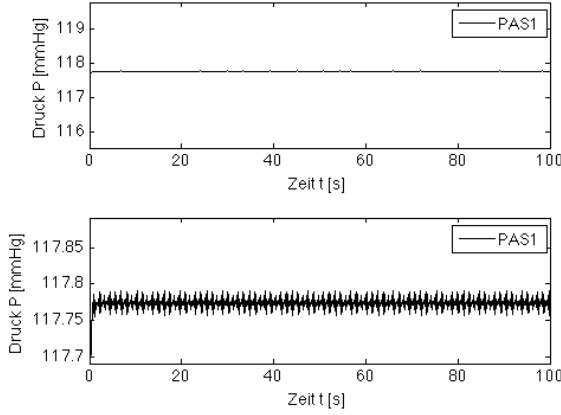


Abbildung 2: PAS bei konstanter Compliance

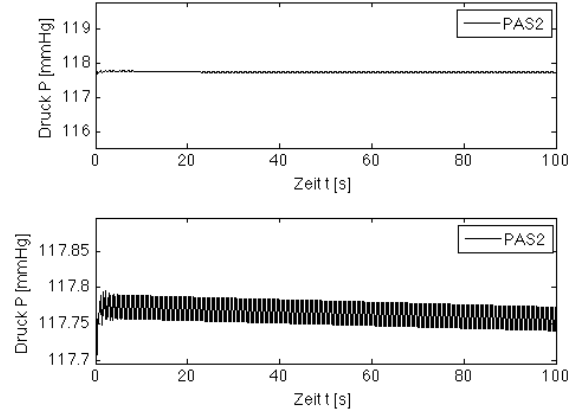


Abbildung 3: PAS bei druckabhängiger Compliance

durch und kam zu der Erkenntnis, dass jedoch qualitativ ein ähnliches Verhalten zu erkennen ist. [4]

3.2 Simulationsergebnisse bei druckabhängiger Compliance

Um dem nichtlinearen Zusammenhang zwischen Compliance und Druck Rechnung zu tragen, wird eine druckabhängige Darstellung der Compliance angesetzt. In der Diplomarbeit von Yvonne Stickler [4], in der Diplomarbeit von Armin Fohler [5], sowie auch in der Dissertation von Johannes Kropf [6] wurde ein Polynom vierten Grades herangezogen um eine Näherung für die nichtlineare Abhängigkeit der Compliance im arteriellen systemischen Kompartiment darzustellen.

$$CAS_2 = CASN \left(1 - \frac{PASN^4}{1 + PASN^4} \right) \quad (4)$$

Es ist $PASN = \frac{PAS}{100} \text{ mmHg}$ der normierte Druck und $CASN = 1.5 \frac{\text{ml}}{\text{mmHg}}$ die normierte Compliance, die hier als Konstante festgelegt wird. In den anderen Kompartiments lassen sich CVS , CAP und CVP als Relation von absolutem Füllvolumen K_{CVS} , K_{CAP} und K_{CVP} und Druck angeben.

$$CVS_2 = \frac{K_{CVS}}{PVS}, CAP_2 = \frac{K_{CAP}}{PAP}, CVP_2 = \frac{K_{CVP}}{PVP} \quad (5)$$

Aus Abbildung 3 geht hervor, dass sich die Werte für den Druck mit konstanter Compliance kaum von den Werten mit druckabhängiger Compliance unterscheiden. Auch das Verhalten der Schwingungen ist sehr ähnlich.

3.3 Simulationsergebnisse bei zeitabhängiger Compliance

Um den Einfluss der Compliance auf das Modell noch genauer zu betrachten, wird diese als zeitabhängige Größe angesetzt. Mittels der Addition einer im Verhältnis zur Compliance gestauchten oder gestreckten Sinusfunktion wird eine Störung eingebunden.

$$\begin{aligned} CAS_3 &= CAS + 0.1 \cdot \sin t \\ CVS_3 &= CVS + 10 \cdot \sin t \\ CAP_3 &= CAP + \sin t \\ CVP_3 &= CVP + \sin t \end{aligned} \quad (6)$$

Die Ergebnisse unterscheiden sich nur minimal von den anderen Simulationen, was in Abbildung 4 zu erkennen ist.

4 Schlussfolgerungen

Da die Gefäßwände der Aorta die dickste muskuläre Schicht besitzen, ist hier der Windkesseneffekt am aus-

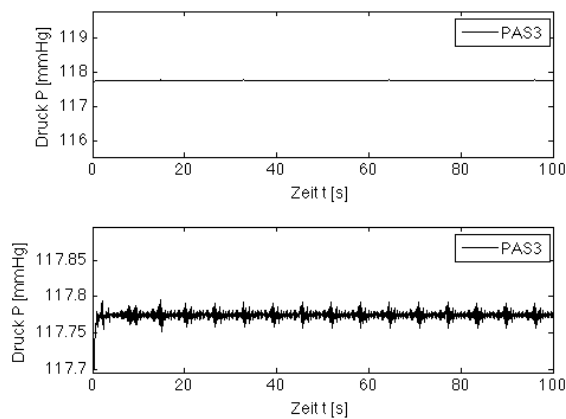


Abbildung 4: PAS bei zeitabhängiger Compliance

geprägtesten. Weil hier die Auswirkung der Compliance am besten beobachtet werden kann, wird das Hauptaugenmerk auf den Druck im arteriellen systemischen Kompartiment, PAS, gelegt. Man sieht, dass sich das System bei allen drei Simulationen ähnlich in kurzer Zeit einschwingt. Die eingeschwungenen Werte variieren nur minimal. Das Verhalten der Schwingung selbst zeigt auch nur leichte Unterschiede.

Bei normaler Herzfrequenz ist der Einfluss der Compliance auf das System gering, bei hoher Herzfrequenz ist er am größten. Dies ist darin zu begründen, dass die Werte für die Compliance in der Inhomogenität des Differentialgleichungssystem (3) in direktem Zusammenhang mit der Herzfrequenz stehen. In Abbildung 5 sieht man, dass sich die Werte für den Druck um weniger als 1 mmHg unterscheiden. Auch die Werte für den Fluss unterscheiden sich um weniger als $0,5 \frac{ml}{s}$.

Bei dem statischen Kompartimentmodell werden Werte simuliert, die über eine Herzschlagperiode und Körperregionen gemittelt sind. Es lassen sich keine detaillierten Aussagen bezüglich dem genauen Zeitpunkt und Ort treffen. Jedoch ist das Modell intuitiv und kann das Herz-Kreislauf-System in seiner Gesamtheit darstellen, wodurch auch Phänomene im venösen System oder im Lungenkreislauf betrachtet werden können.

Es hat sich experimentell gezeigt, dass bei diesem Kompartimentmodell die Compliance durchaus als

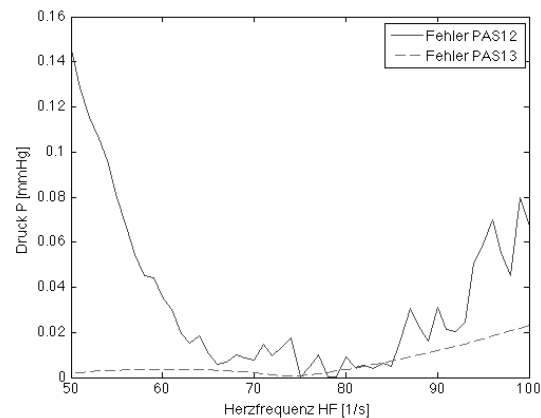


Abbildung 5: Vergleich von PAS bei konstanter und druckabhängiger bzw. zeitabhängiger Compliance

konstant angenommen werden kann. Die Werte für die Elastizität der Gefäßwände im arteriellen systemischen Kompartiment hängen von komplexen Mechanismen ab. Bei einem detaillierteren Modell wäre es notwendig diese komplexen Zusammenhänge herauszuarbeiten, bei dem allgemeinen Kompartimentmodell zeigen sich aber keine relevanten Auswirkungen.

References

- [1] Westerhof N., Lankhaar J., Westerhof B.: *The Arterial Windkessel*. Med Biol Eng Comput 47:131-141,2009.
- [2] Quick C.: *Integrated Arterial Hemodynamics*.
- [3] Golenhofen K.: *Basislehrbuch Physiologie*. Urban und Fischer Verlag, 2006.
- [4] Stickler Y.: *Dynamisches Modell des Herzkreislaufs mit Regelung*. Technische Universität Wien, 2002.
- [5] Kropf J.: *Multiscale Blood Flow Modelling*. Technische Universität Wien, 2007.
- [6] Fohler A.: *Modellierung und Parameterschätzung eines erweiterten Kompartimentmodells des Herzkreislaufsystems unter Berücksichtigung blutdruckregulatorischer Mechanismen*. Technische Universität Wien, 2011.

MAPLIS - Matrixorientierte Simulation, wieder aufgenommen

Wilfried Tettweiler, Krailling
wtettweiler@compuserve.com

Zusammenfassung: Bei makroskopischen Simulationen (z.B. unter Verwendung der Simulationssprache DYNAMO und deren Nachfahren POWERSIM und VENSIM) muss man Koeffizienten von Differenzengleichungen durch die üblichen statistischen Verfahren der Informationsreduzierung (Regressionsanalyse, Faktorenanalyse usw.) bestimmen und nimmt damit im Allgemeinen einen Informationsverlust in Kauf. Im MAPLIS-Konzept treten an die Stelle der (skalaren) Koeffizienten wahlweise ein- oder mehrdimensionale Matrizen (Tabellen). In diesen werden Inhalte von ebenso mehrdimensionalen Kontingenztafeln unmittelbar verwendet, so dass auf Informationsreduzierung weitgehend verzichtet werden kann. Dabei liegt es ganz im Ermessen des Modelldesigners, wie viele Dimensionen er jeweils für angebracht hält. Eine Besonderheit dieser Modelldefinitionsmethode ist darüber hinaus, dass sich Wegenetze für Personen, Waren, Gelder u.ä. endogen oder exogen dynamisch ändern können, ohne dass das Modell selbst geändert werden muss.

1 Vorwort

Dieser Vortrag stellt eine Simulationssprache vor, deren Design zwar in der Praxis langjährig erprobt ist, der zugehörige Compiler oder gar eine grafische Modellentwicklungsumgebung aber nie über die Testphase hinaus weiter entwickelt wurde. Vielleicht trifft sie aber gerade jetzt - dank der erfreulichen Entwicklung von Preisen und Fähigkeiten für IT in den vergangenen zwanzig Jahren - auf das Interesse der Simulationisten-Gemeinschaft, so dass die Arbeit an der Stelle, an der sie Mitte der 1990er Jahre aus politischen und finanziellen Gründen beendet wurde, nun wieder aufgenommen wird.

In einer laufend komplexer werdenden Wirklichkeit erhält Planungssoftware einen immer höheren Stellenwert. Hier füllt MAPLIS (**Matrixorientierte Plan-spiele und interaktive Simulationen**) als Werkzeug für jede Art quantitativer Planung eine Lücke. Seine Anwendung kann ein sehr breites Spektrum abdecken, das von der Investitionsplanung des Managements in Wirtschaftsbetrieben bis hin zur Schulraumbedarfsplanung in der öffentlichen Verwaltung reicht.

MAPLIS versteht sich formal als mesoskopische Alternative zwischen eher mathematisch, technisch-physikalisch oder auch makroskopisch orientierten Mehrzweck-Simulationswerkzeugen wie z.B. MATLAB und mikroskopisch orientierten wie z.B. SIMSCRIPT oder GPSS.

Noch eine Anmerkung zur Namensgebung: Simulation mit Einzelnen, durch Skalare dargestellt \triangleq "mikroskopisch". Simulation mit Gruppen in Zellen der jeweils gemeinsamen Merkmalskombination in multidimensionalen Tabellen \triangleq "mesoskopisch". Simulation mit skalaren Koeffizienten, die einzelne Bestandteile hoch aggregiert lediglich statistisch betrachtet \triangleq "makroskopisch".

2 Historischer Überblick

2.1 FORTRAN-Bibliothek (IBM 1975)

Die ersten Modelle wurden durch CALLs der Routinen einer FORTRAN-Laufzeitbibliothek realisiert.

2.2 Vektorisierung (Cray 1980)

Ein Glücksfall war, dass eine Cray-1, der weltweit erste Vektorrechner in der Grundlagenforschung, am Institut für Plasmaphysik in Garching installiert wird, auf der auch die Landeshauptstadt München ihr Simulationsmodell ausführen konnte. So verringerte sich die Ausführungszeit des Modells für eine 10-Jahresprognose von Tagen auf Stunden.

2.3 Compiler (kommandoorientierte Programmiersprache 1990)

Mit einem Compiler-Compiler [1] wurde MAPLIS mit einem Oberflächendesign für Microsoft Windows erzeugt und getestet, sowie die Laufzeitbibliothek in diesem Rahmen nach PASCAL (Delphi) migriert.

3 Designidee

MAPLIS soll System-Dynamics-Simulationen vereinfachen und unterstützen durch

- Verringern des Statistikaufwandes, möglichst jedoch ohne die Geschehensvielfalt der Wirklichkeit bei der Abbildung in ein Modell zu reduzieren
- einfache Bedienbarkeit insbesondere für die Statistiker-Gemeinschaft, z.B. durch Fortfall der Notwendigkeit, Modelle mit Differentialgleichungen oder DYNAMO-artigen Formeln zu definieren

- Ausführung von Kommandos in Skripten oder durch direkte Interaktivität
- die Möglichkeit, unterschiedliche Szenarien geschlossen in einem Durchlauf durchzuspielen

Das Design dieser formalen Sprache hat mehrere geistige Väter:

- die Modellierung basiert auf Flussdiagrammen nach Art der gebräuchlichen Simulationssprachen für makroskopische Simulationen mit Quellen, Reglern und Senken, wie von Meadows in den Weltmodellen [2] eingesetzt,
- die Datenspeicherung orientiert sich am Tensor-konzept der Programmiersprache APL und der Form der Tabellen wie in MS-Excel verwendet,
- die Syntax lehnt sich an die der weit verbreiteten Statistiksprache SPSS an (siehe 4.2 unten)

Der Forderung nach einfacher Bedienung entsprechend sind lediglich Sprachelemente für die Formulierung und Simulation von System-Dynamics-Modellen vorhanden, für Analyse und Ergebnisvergleich werden andere Werkzeuge benötigt.

3.1 Periodenbasierte Simulation

Der zeitliche Verlauf wird durch die implizit vorhandene Dimension Periodennummer dargestellt, die einen äquidistanten Zeittakt (z.B. Monate, halbe und ganze Jahre) abbildet. Die gesamte Datenbasis (alle Bestands- und Flussgrößen) wird je Periode auf der Basis der vorangegangenen Periode aufs Neue erzeugt. In Zuweisungen wird bei Variablen daher auf Syntaxelemente für den Zeitbezug (z.B. Postfixe wie in DYNAMO: .K, .J, .JK, .KL) verzichtet. Soll in der aktuellen Periode auf Werte weiter vorangegangener bzw. früherer Perioden zurück gegriffen werden, muss dies durch die weitere Dimension Periode/Zeit (TIME) explizit deklariert werden. Dann können mit der Funktion LAG auch Werte früherer Perioden adressiert werden.

3.2 Vorauslaufenden Statistikaufwand reduzieren

Für die Sozial- und Wirtschaftswissenschaften sind die gebräuchlichen Simulationssprachen gewöhnlich nicht griffig, da sich diese Materie häufig einem Erfassen in analytisch-algebraische Ausdrucksmittel wie Polynome und Differenzen- oder Differenzialgleichungen widersetzt, mit denen skalare Bestands- und Flussgrößen (Levels und Rates) verknüpft werden. Letztere sind vorab mit mehr oder minder anspruchsvollen statistischen Auswertungsmethoden wie Regressions- oder Faktorenanalysen zu bestimmen. MAPLIS-Modellvariable hingegen enthalten als Bestandsgrößen absolute Häufigkeiten und als Flussgrößen i.a. relative Häufigkeiten, die recht einfach in je nach gewünschtem Detaillierungsgrad vieldimensionalen

Kontingenztafeln ermittelt werden. Es braucht daher vom Datentyp her nicht zwischen Bestands- und Flussgrößen, Hilfsvariablen und Konstanten unterschieden werden. Alle Rechenfunktionen sind nicht-skalar ausgelegt, sie können auf Vektoren (Zahlenreihen), Matrizen (Tabellen) und mehrdimensional angeordnete Daten (auch Tensor genannt) angewendet werden. Relative Häufigkeiten aus den Kontingenztafeln bestimmen, nach welchem Verteilungsschlüssel Mengen geteilt werden, in welche Werte vorgegebene Werte transformiert werden oder welche Vertauschungen oder Zusammenfassungen unter problembezogenen Gesichtspunkten stattfinden sollen.

3.3 Multidimensionale Daten verschiedener Aggregationsniveaus

Die multidimensionale Matrix ist Form der Darstellung aller am Prozess beteiligten Einheiten (Personen, Materialien, Informationen usw.) sowie der extern oder endogen mittels Verteilen oder Berechnen bestimmten Quoten.

Auf der Basis der Definition von Dimensionen mit einer endlichen Wertemenge werden Modellvariable deklariert. Entsprechend ihrem jeweiligen Detaillierungsgrad unterscheiden sich die Aggregationsniveaus und damit der Speicherbedarf.

Typisches Beispiel für den Einsatz der multidimensionalen Sprache MAPLIS ist die Prognose der Bevölkerung der Landeshauptstadt München und deren kleinräumige Entwicklung [3]. Bei diesem komplexen Untersuchungsgegenstand wurde eine große Menge von vieldimensionalen Beziehungen berücksichtigt. Die einzelnen Variablen sind je nach Bedarf unterschiedlich stark differenziert. Beispielsweise hat die für die Prognose zentrale Bestandsgröße Einwohnerbestand einen sehr hohen Detaillierungsgrad mit den Dimensionen Alter, Geschlecht, Nationalität, Wohntyp und Stadtviertel, für diese wurden pro Periode $109 \cdot 2 \cdot 2 \cdot 3 \cdot 429 = 561.132$ Zellen verarbeitet.

Erweitert man die Variablendefinition um eine weitere Dimension Handlungs- oder Planungsalternative für unterschiedliche Ansätze mit drei bis fünf Merkmalen (z.B. worst case bis best case) auf, spart man sich Wiederholungsläufe und kann die Ergebnisse verschiedener Experimente in ein und derselben Tabelle vergleichen.

3.4 Aggregationsniveauwechsel

Für den Wechsel der Aggregationsniveaus - den Kern der Verarbeitung mehrfacher Dimensionen wie ausführlich in [4] beschrieben - werden die Funktionen AGGREGATE (Aggregieren) und DISAGGREGATE (Disaggregieren) vorgehalten. Aggregieren bedeutet Wegfall mindestens einer Dimension durch Zeilen-

oder Spalten-Summenbildung, Disaggregieren hingegen Hinzunehmen einer Dimension, deren Bestandsgrößen entsprechend dem Wertevorrat der hinzugekommenen Dimension verteilt werden. Bei beiden Funktionen ist neben der einfachen Rechenvorschrift der Summierung bzw. Teilung auch vorgesehen, auf der Basis entsprechender Zuordnungstabellen Zusammenfassungen von Merkmalsgruppen (Option PERMUTATION, in Anlehnung an das SPSS-Kommando RECODE) zu berechnen, womit eine Dimension durch eine andere ersetzt wird. Typisches Beispiel ist das Aggregieren vom Alter zu Altersgruppen, wo ca. hundert Merkmale zu einigen wenigen zusammengefasst und entsprechende Teilsummen gebildet werden.

Besonders bemerkenswert ist (weil man Vergleichbares in traditionellen Simulationssprachen nicht findet) dass beispielsweise Wegenetze mit Übergangsmöglichkeiten erst zur Laufzeit durch Besetzen von Permutationsvektoren definiert werden können. Berechnungen zielen auf die endogene Modifikation der Modellstruktur ab (siehe 4.2 unten, wo sich Schulkarrieremöglichkeiten dynamisch ändern können). Ergebnisse von Sortierungen und Kategorisierungen können dynamisch Vorschriften ändern zum Teilen, Zusammenfassen, Vertauschen und Transformieren von Daten. Eine solche Modellierungstechnik ist auf die Finanzindustrie übertragbar, wo sich Geldflüsse ebenfalls dynamisch immer wieder neue Wege suchen.

Beim Disaggregieren (auf der Basis entsprechender Wahrscheinlichkeiten bzw. Häufigkeiten) ist auch eine Option verfügbar, ganzzahlige Werte zu berechnen, beispielsweise beim Bestimmen von Stückzahlen oder Personenanzahlen.

Bei Zuweisungen und Berechnungen ist die Funktion EXPAND im Einsatz, falls für unterschiedliche Aggregationsniveaus identische Häufigkeiten/Wahrscheinlichkeiten gelten sollen.

3.5 Modellsegmentierung

Modelle können in hierarchisch strukturierte Untermodelle zerlegt und einzeln getestet werden.

3.6 Tabellen lernen laufen

In MAPLIS formulierte Modelle arbeiten mit Tabellen in ihrem periodischen Verlauf. Während der Modellausführung kann ein aktueller (Daten-)Zustand wo nötig gespeichert und/oder wieder geladen werden. So können Tabellenfolgen aus ex-post-Simulationen mit Tabellenfolgen aus einem Datenspeicher (z.B. einem Data Warehouse) verglichen werden, um Modelle zu validieren. Auch eine filmische Präsentation von prognostizierten Tabellenfolgen ist vorstellbar.

4 Die formale Sprache MAPLIS

Die Sprache und ihre Anwendung wurden auf einigen Tagungen vorgestellt und veröffentlicht:

- 1982 MAPLIS - Matrixorientierte Sprache für Planspiel und interaktive Simulation [5]
- 1983 MAPLIS - A Matrix Oriented Language for Modelling and Interactive Simulation [6]
- 1996 MAPLIS - Simulation with Work Sheets in Hyper Space [7]

4.1 Spracheigenschaften

MAPLIS ist eine prozedurale Sprache ohne echtes Timing (siehe 3.1 oben), daher wird die Dimension Periode i.a. nicht deklariert. Explizit deklariert werden muss jedoch die Dimensionalität von Modellvariablen wegen des zumindest theoretisch unbegrenzt multidimensionalen Speicherkonzepts.

MAPLIS kennt lediglich den Datentyp Float. Int wird als Float und Logical als Float mit den Werten 0=False, 1=True behandelt. Die Darstellung als Mantisse plus Exponent mit der hardwareabhängigen Rundungs- und Sättigungsproblematik wird nicht berücksichtigt, desgleichen der Wert NULL (nil, k.A.). Somit wird auf jede Datentypprüfung verzichtet, nicht jedoch auf die Prüfung der Verträglichkeit der Reihenfolge von Dimensionen.

Für Eingabe und Ausgabe (insbesondere für Auswertungen) sind Datei- und Datenbankschnittstellen vorgesehen.

Für Zuweisungen und Berechnungen im eigentlichen Simulationsteil (Modelldefinitionsteil) stehen alle Standardmethoden aus Mathematik und Statistik zur Auswahl, Ergänzungen hierzu können künftig beispielsweise entsprechend einer der mathematischen Software-Programmbibliotheken wie den NAG Numerical Libraries der Firma "The Numerical Algorithms Group Ltd", Oxford vorgenommen werden.

Die gesamte Syntax hat die Backus-Naur-Form (EBNF-Notation), so dass für den Einsatz des Compiler-Compilers lediglich die entsprechenden semantischen Aktionen für die Aufrufe der MAPLIS-Laufzeitbibliothek eingefügt werden mussten.

4.2 Syntaxbeispiel

Die Programmiersprache MAPLIS ähnelt in seiner Nomenklatur der Kommandosprache von SPSS, um Anwendern aus dem Statistikbereich den Einstieg zu erleichtern. Sie ist praxisnah und erhebt den Anspruch, ingenieurmäßiges Modellieren zu ermöglichen.

Als Beispiel hierfür dient das einfache Modell einer Schülerprognose bis hin zur mittleren Reife für einen einzelnen Schulstandort mit den Schularten Grund-, Haupt- und Realschule sowie Gymnasium. Die Be-

standsgröße S (Schüler) hat die Dimension KL (Klassenstufe). In ihr sind drei Schularten, gruppiert nach Grundschule 1-4, Gymnasium 5-10, Hauptschule 11-15, Realschule 16-19, enthalten. So kann die Klassenfolge in den Schularten dynamisch gestaltet werden.

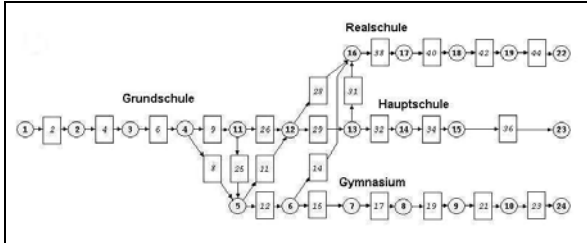


Abb. 1 aktuelles Wegenetz der Übergangsmöglichkeiten (der Knoten 25 mit Abgängen-Wegzügen ist der Übersichtlichkeit halber weggelassen)

Auch das Wegenetz der Übergänge mit Knoten (Klassenstufen, mit Wertevorrat KL) und Kanten (Übergangsmöglichkeiten) kann dynamisch geändert werden. Die Dimension UE (Übergänge) ist hier auf maximal 45 Möglichkeiten ausgelegt.

Die bis zu 45 Klassenübergänge sind durch die mit UE dimensionierte Verteilung Q und das Permutationsvektorpaar DPXZA (Quellen) und APXZA (Ziele) festgelegt.

Dieses Modell, in MAPLIS-Syntax formuliert, besteht aus einem Deklarationsteil, in dem Dimensionen (Indexe) und darauf beruhend Variablen mit ein bis zwei Dimensionen sowie deren Etiketten festgelegt werden:

Run name	Schülerdurchfluß simulieren
Comment	nachfolgend Deklarationen der Dimensionen (Indizes) und darauf basierend der Variablen
N of values	WE (2), KL (25), UE (45)
Value labels	WE (1) Klassenziel nicht erreicht (2) erfolgreich / KL (4) 4. Jgst. Grundschule (5) 5. Jgst. Gymnasium (10) 10. Jgst. Gymnasium (11) 5. Jgst. Hauptschule (22) Abschlüsse Realschule (23) Abschlüsse Hauptschule (24) Mittl.Reife Gymnasium (25) Abgänge-Wegzüge
Variable list	SY, SZÄ (KL), S (KL), E, SWE (KL by WE), SW, SE (KL), Q, DPXZA, SXZA, APXZA (UE)
Var labels	SY Zuziehende Schüler / SZÄ Vorrückende bzw. übertretende Schüler / S Schüler insgesamt / E Erfolgswahrscheinlichkeit / SWE Nach Erfolg getrennte Schüler / SW Wiederholende Schüler / SE Erfolgreiche Schüler / Q Übertrittswahrscheinlichkeit / DPXZA Herkunftsmerkmal nach Übertrittsart / SXZA Schüler nach Übertrittsart getrennt / APXZA Zielmerkmal nach Übertrittsart /

Abb. 2 Deklarationsteil

Diesem folgt die Modelldefinition, deren Besonderheit ist, dass die Kommandofolge DISAGGREGATE und AGGREGATE die Dynamisierung des Wegenetzes ermöglicht, da sich die Werte in der Verteilung Q und dem Permutationsvektorpaar DPXZA (Quellen)

und APXZA (Ziele) von Periode zu Periode endogen oder exogen ändern können:

Add model	SDFSIM Schülerdurchfluß-Simulationsmodell
Document	Vorgehensweise: Schüler verteilen auf Vorrückende und Wiederholende, Vorrückende verteilen entsprechend der Übertrittsart, Zusammenführen zu neuen Klassenverbänden und Hinzufügen der Wiederholer.
Disaggregate	Result = SWE Source = S Distribution = E
Compute	SE = sub (SWE,2)
Compute	SW = sub (SWE,1)
Comment	Wiederholer werden erst später wieder gebraucht, nun entsprechend Übertrittswahrscheinlichkeiten auf die verschiedenen Alternativen verteilen
Disaggregate	Result = SXZA Source = SE Distribution = Q
Aggregate	Permutation = DPXZA
Aggregate	Result = SZÄ Source = SXZA Permutation = APXZA
Compute	S = SW + SZÄ
End model	

Abb. 3 Modelldefinitionsteil

Für das Schuljahr 1982/83 wird ein Testlauf durchgeführt, in dem die Anfangswerte (Quelle S und Parameter E und Q, DPXZA sowie APXZA) eingelesen, das Modell ausgeführt und ein Ergebnis (Vorstufe der Senke SZÄ, in der Wiederholer noch nicht hinzuge-rechnet sind) angezeigt wird:

Read data	S, E, Q, DPXZA, APXZA
1082,1095,1109,1154,353,368,371,366,369,365,971,997,771,766,737,351,410,422,393/	
*** WARNING: VARIABLE S INCOMPLETE DATA ***	
.04, .03, 2*.02, .04, .07, .14, .14, .15, .12, S*.01, .04, 2*.16, .08, 6*0, .96, .97, 2*.98, .96, .93, .86, .86, .85, .88, 5*.99, .96, 2*.84, .92, 6*0/	
*** MESSAGE: VARIABLE E DATA ACCEPTED ***	
.030, .970, .005, .995, .005, .995, .005, .235, .760, .028, .069, .903, .02, .05, .93, .03, .97, .03, .97, .01, .99, .01, .99, .003, .051, .946, .002, .237, .761, .017, .072, .911, .093, .907, .01, .99, .001, .999, .025, .975, .007, .993, .055, .945/	
*** MESSAGE: VARIABLE Q DATA ACCEPTED ***	
2*1, 2*2, 2*3, 3*4, 3*5, 3*6, 2*7, 2*8, 2*9, 2*10, 3*11, 3*12, 3*13, 2*14, 2*15, 2*16, 2*17, 2*18, 2*19/	
*** MESSAGE: VARIABLE DPXZA DATA ACCEPTED ***	
25, 2, 25, 3, 25, 4, 25, 5, 11, 25, 12, 6, 25, 16, 7, 25, 8, 25, 9, 25, 10, 25, 24, 25, 5, 12, 25, 16, 13, 25, 16, 14, 25, 15, 25, 23, 25, 17, 25, 18, 25, 19, 25, 22/	
*** MESSAGE: VARIABLE APXZA DATA ACCEPTED ***	
Run model	SDFSIM Sequenz = 1982 Entry
*** MESSAGE: MODEL SDFSIM, PERIOD 1982, INTERRUPT AT ENTRY ***	
*** MESSAGE: MODEL SDFSIM, PERIOD 1982, INTERRUPT AT RETURN ***	
Print	SZÄ
-> KL	

I SZÄ I Vorrückende bzw. übertretende Schüler	

1 2 3 4 5 6 7 8	
	4. Jgst. 5. Jgst.
	Grundschu Gymnasium
I 0. I 1008. I 1057. I 1081. I 315. I 306. I 318. I 309. I	

9 10 11 12 13 14 15 16	
	10. Jgst. 5. Jgst.
	Gymnasium Hauptschu
I 305. I 313. I 859. I 933. I 751. I 695. I 688. I 306. I	

17 18 19 20 21 22 23 24	
	Abschluss Abschluss Mittl.Rei
	e Realsch e Hauptsc fe Gymnas
I 337. I 336. I 352. I 0. I 0. I 342. I 727. I 321. I	

25	
Abgänge-W	
egzüge	
I 206. I	

Stop	
*** MESSAGE: MODEL SDFSIM, PERIOD 1982, STOP AT RETURN ***	
Finish	

Abb. 4 Testlauf des Modells

Bei der angezeigten Tabelle ist zu beobachten, dass mit dem Modell und den Eingaben das erwartete Ergebnis erzielt wurde: Die Schüler sind um eine Klasse vorgerückt und es wurden Anzahlen für Abschlüsse berechnet.

Um einen räumlichen Bezug (z.B. für Schulstandorte) zu realisieren, ergänzt man im Deklarationsteil lediglich eine weitere Dimension, das Modell selbst

braucht aber nicht geändert zu werden (siehe unten 5.1 Schülerprognose).

5 Modellentwürfe und -anwendungen

Auf der Basis des matrixorientierten Modelldesigns wurden zwei Projekte durchgeführt:

- bayerische Schüler (1979) [8]: ein Schülerzahlenprognosemodell
- Stadtplanung München (1980) [3]: Prognose von Bevölkerungsentwicklung, Bautätigkeit, Arbeitsstättenplanung und Schulraumplanung sowie eine kartografische Ergebnisdarstellung.

5.1 Schülerprognose

Im konkreten bayerischen Schülermodell wurden analog zum obigen Beispiel die Schularten Grund-, Haupt, Real- und Wirtschaftsschule sowie Gymnasium mit den Dimensionen Schularten, Übertrittsquoten, Schulstandorte und innerbayerische Migration simuliert. Die Flüsse zwischen diesen stellen sich grafisch wie folgt dar:

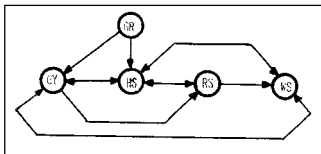


Abb. 5 Flüsse zwischen Schularten

Als weitere Dimension kam Schulstandort hinzu, entsprechend den am jeweiligen Standort vertretenen Schularten wurden die Permutationstabellen DPXZA (Quellen) und APXZA (Ziele) angepasst. Die Migration zwischen Schulstandorten ließ sich in einer relativ schwach besetzten Übergangsmatrix beschreiben, in ihr sind nur jene Zellen mit hohen Wahrscheinlichkeiten besetzt, die Flüsse der Art "verbleibt am Schulstandort" oder "wechselt an einen anderen Schulstandort, weil nur dort die entsprechende weiterführende Schulart existiert" beschreiben.

5.2 Bevölkerungsprognose

Das Münchner Simulationsmodell [9], die wichtigsten Dimensionen wurden bereits in 3.3 oben genannt, umfasste mehr als 30 Untermodelle, einige hiervon als Beispiele:

- Kleinräumiges Bevölkerungsprognosemodell
- Kleinräumiges Schülerprognosemodell
- Allokationsmodell zur Ermittlung des Versorgungsgrades der Stadtviertel
- Kleinräumiges Erwerbspersonenprognosemodell

Die Belange der Fachplanung wurden insbesondere dadurch befriedigt, dass dank der Dialogfähigkeit des Simulators Rechenläufe jederzeit unterbrochen wer-

den, Zwischenergebnisse abgefragt, abgespeichert und wieder unter neuen Gesichtspunkten weiter verarbeitet werden konnten. So wurde während der Simulation sowohl den Erfordernissen klassisch-mathematischer Modellgesetzmäßigkeiten genügt wie auch die Möglichkeit geschaffen, im Modell Eingriffe in die Datenstruktur entsprechend planerischer Zielsetzungen vorzunehmen.

6 Ausblick

Für die Methode der mesoskopischen Simulation haben sich die Realisierungschancen auf PCs in den letzten 30 Jahren fortlaufend verbessert:

6.1 Moderne Hard- und systemnahe Software

Die Prozessorkernanzahl und -geschwindigkeit sind gestiegen. Die Speicher wurden größer und erschwinglicher, nicht nur intern (RAM) sondern auch extern, diese außerdem schneller (Solid State Disks). Für den Datenaustausch gibt es platzsparende und preiswerte Transportmedien (USB-Platten und -Sticks), der Zugriff auf die Cloud ist mittlerweile Standard. Die Parallelisierung hat auch in den PCs durch die Nutzung von Grafikprozessoren (GPUs) Einzug genommen, z.B. in der Parallel Computing Toolbox von MathWorks. Dabei abzuwägen ist, ob die schnellere Rechenleistung nicht durch Up- und Downloadzeiten wieder kompensiert wird.

PC-Betriebssysteme (64bit) sorgen für den bei großen Matrizen erforderlichen größeren Adressraum. Es gibt sehr leistungsfähige Datenbanksysteme (s.u.) und standardisierte Schnittstellen für Import und Export von Daten (Quasistandards MS Excel, xBASE) erleichtern das Synchronisieren von simulierten mit realen Daten.

Und die Preisentwicklung bleibt weiter erfreulich.

6.2 Visuelle Programmierungsumgebung

Wünschenswert ist ein Icon-basiertes grafisches Interface zur Modelldefinition, beispielsweise mit Pfeilen für Ursache-Wirkungs-Beziehungen, oder auch die farbliche Gestaltung des Desktop mit Flächen, die Variablen gleichen Aggregationsniveaus enthalten, und nicht zuletzt die automatische Umwandlung der grafischen in eine skriptbasierte Modelldarstellung und umgekehrt.

Für das künftige Design der grafischen Modellentwicklungsumgebung gibt es bereits einige bewährte Vorlagen, an denen man sich orientieren kann: Attune Exposé (Attune Group, Inc., www.attunegroup.com), isee STELLA, Powersim Studio (Powersim Software AS, Bergen, www.powersim.com) oder Vensim Development Environment (Ventana Systems, Inc., Harvard, USA, übrigens wurden viele DYNAMO-

Modelle mittlerweile in VENSIM-Modelle konvertiert, www.vensim.com oder www.simcon.de/vensim)

6.3 Verbesserungen für Eingabe und Ausgabe

Die vorhandenen einfachen Dateischnittstellen wären um entsprechende (proprietäre) Schnittstellen für Microsoft Excel und xBASE-Tabellen zu vervollständigt.

Eine integrierte Schnittstelle zu einem GIS-System wäre nötig, um Simulationsergebnisse mit geographischem Bezug darzustellen.

6.4 Datenbankschnittstelle

Besonders für der Speicherung schwach besetzter Matrizen bietet sich die Verwendung einer Datenbankschnittstelle an: z.B. ist bei einer Abbildung innerstädtischer Migration als Matrix, die Umzüge beschreibt, überwiegend nur die Hauptdiagonale mit großen Anzahlen besetzt (dort finden sich alle Haushalte, die nicht umziehen), der größte Teil der übrigen Zellen ist jedoch unbesetzt. Hier - und nicht nur hier - sind die Konzepte neuer Datenbanken zu berücksichtigen. Schlagworte hierzu sind: strukturierte Datenbanken (noSQL), Multidimensional Clustering und Multidimensional Expressions (MDX).

Für die Auswertung und grafische Darstellung der multidimensionalen Simulationsergebnisse kann MAPLIS in neue Werkzeuge wie Canias-BI (Industrial Application Software GmbH, Karlsruhe, www.caniaserp.de) integriert werden.

7 References

Da meines Wissens die Methode matrixorientierte Simulation seit 20 Jahren keine wesentliche Weiterentwicklung genommen hat, ist die hierfür gelistete Literatur entsprechend alt:

- [1] H. Mössenböck, *A Generator for Fast Compiler Front-Ends*, Zürich: Report 127, Dept. Informatik, ETH Zürich, 1990.
- [2] D. Meadows, D. Meadows, E. Zahn und P. Milling, *Die Grenzen des Wachstums. Bericht des Club of Rome zur Lage der Menschheit*, Stuttgart, Deutsche Gesellschaft des Club of Rome, 1972.
- [3] W. Tettweiler, *Simulationsprogramm zur Prognose der kleinräumigen Bevölkerungsentwicklung MIPROS 470*, München: Referat für Stadtplanung der Landeshauptstadt München, 1980.
- [4] W. Tettweiler, *MAPLIS - A Matrix Oriented Language for Modelling and Interactive Simulation in First European Simulation Congress ESC 83*, Bde. %1 von %2Informatik-Fachberichte Band 71, P. W. Ameling, F. Breiteneker und W. Kleinert, Hrsg., Aachen, September, 12–16, 1983 Proceedings, Springer Berlin Heidelberg, 1983, pp. 314-319.
- [5] W. Tettweiler, *MAPLIS - Matrixorientierte Sprache für Planspiel und Interaktive Simulation in Simulationstechnik, 1. Symposium Simulationstechnik, Erlangen, 26.-28. April 1982, Proceedings.*, M. Goller, Hrsg., Erlangen, Springer Informatik-Fachberichte ISBN 3-540-11605-2, 1982, pp. 173-180.
- [6] W. Tettweiler, *MAPLIS - A Matrix Oriented Language for Modelling and Interactive Simulation in Fourth International Conference on Mathematical Modelling in Science and Technology, Zürich, 15.-17. August*, X. J. Avula, Hrsg., Zürich, Pergamon Press New York, Oxford, Toronto, Sydney, Paris, Frankfurt ISBN 0-08-030156-8, 1983, pp. 339-344.
- [7] W. Tettweiler, *MAPLIS - Simulation with Work Sheets in Hyper Space in EUROSIM'96, Simulation Congress, Proceedings of the 1995 EUROSIM Conference*, I. H. Felix Breiteneker, Hrsg., Wien, ARGESIM Report No. 2 ISBN 3-901608-02-8 (Elsevier 1995 ISBN 0-444-82241-0), 1995, pp. 111-114.
- [8] W. Tettweiler, *Einführung in das Rechnungsprogramm MACSIM/MICSIM zur Simulation des Schulaufbaus in Bayern*, München: Staatsinstitut für Bildungsforschung und Bildungsplanung, 1979.
- [9] U. Kubatzky und W. Tettweiler, *Dialogfähige Simulationsmodelle als Entscheidungshilfe praktischen Planungshandelns in Öffentliche Verwaltung und Datenverarbeitung, Heft 11*, Bd. Heft 11, pp. 9-13, 1981.

Modellierung von Evolutions Systemen mit Stochastischen Zellulären Automaten

Günter Schneckenreither^{1,2}, Niki Popper², Felix Breiteneker¹

¹Institut für Analysis und Scientific Computing, Technische Universität Wien

²dwh simulation services, Neustiftgasse 57-59, 1070 Wien

gschneck@seth.asc.tuwien.ac.at

Dieser Beitrag beschäftigt sich mit Zellulären Automaten als Methode der Simulation von dynamischen Systemen. Insbesondere wird an einem Beispiel methodisch gezeigt wie ein deterministisches System durch einen stochastischen Formalismus (*Stochastische Zelluläre Automaten*) beschrieben werden kann. Eine derartige Übersetzung zwischen deterministischen und stochastischen Formulierungen eines Systems ermöglicht Analysemethoden die in der ursprünglichen Formulierung nicht verfügbar sind. Insbesondere kann eine lineare Formulierung des Systems dazu verwendet werden, um ein äquivalentes Raum- und Zeit-kontinuierliches System zu beschreiben.

1 Einleitung

Zelluläre Automaten sind iterative Systeme, die sich durch eine – für gewöhnlich regelmäßige, räumlich interpretierte – Anordnung von vielen Zustandsgrößen auszeichnen. Die Zustandsvariablen werden als Zellen betrachtet, die zu jedem Zeitpunkt einen Zustand aus einer gegebenen Menge von Zuständen annehmen. Die Iteration der Zellzustände erfolgt durch lokale Aktualisierungsregeln, die den zukünftigen Zustand einer Zelle von den Zuständen der Zellen in einer lokalen Umgebung – basierend auf der gegebenen Anordnung – abhängig machen. Diese Eigenschaft ist ein weiteres wesentliches Merkmal von Zellulären Automaten.

Um Zelluläre Automaten als eine Methode der Modellbildung und Simulation verwenden zu können, werden keine Einschränkungen an den Zustandsraum oder die Aktualisierungsvorschriften gestellt. Weiters ist es wünschenswert, dass eine mathematische Beschreibung zur Verfügung steht und Methoden bekannt sind, die Modellvergleiche und die Identifikation von Parametern erlauben.

Ein mathematischer Formalismus für Zelluläre Automaten sollte auch stochastische Zustände

berücksichtigen und auch für diesen Fall Analysemethoden erlauben bzw. bereitstellen.

Ausserdem kann das Konzept von Zellulären Automaten als Diskretisierung von raum- und zeitkontinuierlichen Systemen verstanden werden. Oft stellt gerade die mathematische Beschreibung und Begründung solcher Übergänge eine große Herausforderung dar.

1.1 Mathematische Beschreibung

Als *Evolutions System* (vgl. [1, 2, 3]) wird eine Gleichung bzw. Iterationsvorschrift

$$S(t + h, x) = (\mathcal{E}_h S)(t, x) \quad (1)$$

verstanden, wobei die Abbildungen $S : T \times X \rightarrow \mathbb{S}$ für alle $t \in T$ in einem Banachraum $\mathfrak{S} \subseteq \mathbb{S}^X$ liegen.

Sind die Operatoren \mathcal{E}_h linear ($L(\mathfrak{S})$) und bilden eine stark-stetige Halbgruppe, so existiert ein infinitesimaler Erzeuger \mathcal{A} und das Evolutions System kann als Differentialgleichung

$$\partial_t S(t, x) = (\mathcal{A} S)(t, x) \quad (2)$$

beschrieben werden. Der Satz von Hille-Yosida

beschreibt die Umkehrung dieser Existenzaussage.

Andererseits kann der topologische Vektorraum X zu einem *Zellraum* M diskretisiert werden (sowie auch die Zeit $T \dots$). Das resultierende diskrete iterative System ist ein *Zellulärer Automat* mit dem Evolutionsoperator $\mathcal{E} = \mathcal{E}_1 : \mathbb{S}^M \rightarrow \mathbb{S}^M$ falls die für Zelluläre Automaten charakteristische Eigenschaft der *Lokalität* indirekt durch

$$S(t+1, m) = \mathcal{F}\left((S(t, n))_{n \in \mathcal{N}(m)}\right) \quad (3)$$

mit (lokaler) Nachbarschaftsabbildung $\mathcal{N} : M \rightarrow M^N$ und Aktualisierungsfunktion $\mathcal{F} : \mathbb{S}^N \rightarrow \mathbb{S}$ gegeben ist [4].

Im Zusammenhang mit den vorherigen Überlegungen ist es also erstrebenswert, dass die Abbildung \mathcal{F} linear ist.

Falls für alle $(t, m) \in T \times M$ die Zellzustände $S(t, m)$ Zufallsvariable $(\Omega, \mathfrak{A}, \mathcal{P}) \rightarrow (\mathbb{S}, \mathfrak{B})$ sind, sprechen wir von einem *stochastischen Zellulären Automaten*, der als Markov Prozess [5] interpretiert werden kann. Die Iterationsvorschrift des Stochastische Zelluläre Automaten ist dann durch einen einheitlichen *lokalen Markov Kern* [4]

$$\mathcal{K}(\mathbf{s}, B) = \mathcal{P}(S(t+1, x) \in B | (S(t, y))_{y \in \mathcal{N}(x)} = \mathbf{s}) \quad (4)$$

definiert wobei $\mathbf{s} \in \mathbb{S}^N$ und $B \in \mathfrak{B}$.

Unter der Annahme, dass die Zustände immer und überall mit einer bestimmten Verteilungsfamilie (Familie von Verteilungen mit einem Parameter) $(W_\theta)_{\theta \in \Theta}$ verteilt sind (*parametrisierter stochastischer Zellulärer Automat* [4]), kann versucht werden, den Übergang als Abbildung $\hat{\mathcal{F}} : \Theta^N \rightarrow \Theta$ zu interpretiert.

Eine andere Möglichkeit (speziell im Fall $\mathbb{S} = [a, b] \subset \mathbb{R}$), die abstrakte Formulierung über Markov Kerne zu konkretisieren, liegt in der Diskretisierung der Verteilungsfunktion oder Dichte als Vektoren in $[0, 1]^k$ bzw. \mathbb{R}_+^k .

1.2 Inhalt

Einerseits kann die stochastische Formulierung oder Interpretation von deterministischen Sys-

temen ein vereinfachtes äquivalentes deterministisches System liefern (prominentes Beispiel ist die Lattice Boltzmann Methode). Oder die Anwendung von statistischen Methoden wie Varianzanalyse, stationäre Verteilungen, etc. erlaubt das Verhalten des stochastischen Systems zu analysieren um Rückschlüsse auf das ursprüngliche System zu machen. In diesem Zusammenhang kann zum Beispiel für einen konkreten Anwendungsfall die Verwendung von Monte Carlo Ansätzen mathematisch exakt begründet werden.

Andererseits können offensichtlich stochastische Systeme durch Diskretisierung bzw. Parametrisierung in deterministische Systeme übergeführt werden.

Am Beispiel der “Game of Life” Aktualisierungsvorschriften wird gezeigt, wie der stochastische und der deterministische Formalismus zusammenhängen und entwickelt werden können. Die konkrete Umsetzung dieser Identifikation ist natürlich für jeden Anwendungsfall unterschiedlich. Das “Game of Life” bietet sich auf Grund der relativ einfachen Regeln und Zustände an. Ein sinnvollerer aber auch anspruchsvollerer Anwendungsbeispiel wäre z.B. die Identifikation von Reaktions-Diffusions Gleichungen mit entsprechenden stochastischen Evolutions Systemen oder Zellulären Automaten.

Im Anschluss werden mögliche Analyse- und Simulationmethoden basierend auf der vorgenommenen Identifikation diskutiert.

2 Anwendungsbeispiel

Die Definition des “Game of Life” sieht einen diskreten Zustandsraum $\mathbb{S} = \{0, 1\}$ vor. Oft werden die Zustände 0 und 1 auch als “tot” bzw. “lebend” bezeichnet. Die Iterationsgleichung kann explizit als

$$S(t+1, m) = [S(t, m)f_a(\Sigma) + (1 - S(t, m))f_d(\Sigma)] \quad (5)$$

angegeben werden, wobei $f_a = \chi_{\{\frac{2}{8}, \frac{3}{8}\}}$ und $f_d = \chi_{\{\frac{3}{8}\}}$. Die Akkumulation der Zustände der Nachbarzellen wird in der Variablen Σ als $\Sigma(t, m) =$

$\frac{1}{N} \sum_{n \in \mathcal{N}(m)} S(t, n)$ zusammengefasst. Die Nachbarschaftsabbildung $\mathcal{N} : M \rightarrow M^8$ beschreibt in diesem Fall eine klassische Moore-Nachbarschaft ohne die mittlere Zelle, da diese separat berücksichtigt wird.

Wird der Zustandsraum als das Intervall $\mathbb{S} = [0, 1]$ angenommen, dann bleibt die Iterationsvorschrift (5) unverändert. Lediglich die Aktualisierungsfunktionen $f_{a,d}$ müssen als $f_a = \chi_{[\frac{2}{9}, \frac{4}{9})}$ bzw. $f_d = \chi_{[\frac{3}{9}, \frac{4}{9})}$ neu definiert werden (siehe Tabelle 1). Durch die Wahl der Aktualisierungsfunktionen liefern beide Ansätze dasselbe Ergebnis (die für das “Game of Life” typischen Muster ...).

Da wir das “Game of Life” als abstraktes Anwendungsbeispiel betrachten, erlauben wir uns, das diskrete System ($\mathbb{S} = \{0, 1\}$) als schlechte Approximation bzw. Vereinfachung des kontinuierlichen Systems ($\mathbb{S} = [0, 1]$) zu verstehen. Um die Simulation dieses Systems mit diskreten Methoden interessanter zu gestalten, glätten wir die “Aktualisierungsfunktionen”. Es bieten sich z.B. Funktionen der Form

$$x \mapsto \frac{1}{2} \left(\tanh \left(\frac{x-a}{\delta} \right) - \tanh \left(\frac{x-b}{\delta} \right) \right) \quad (6)$$

an. Mit Hilfe eines Parameters λ kann zusätzlich erreicht werden, dass $f_{\lambda,a} \rightarrow \chi_{[\frac{2}{9}, \frac{4}{9})}$ und $f_{\lambda,d} \rightarrow \chi_{[\frac{3}{9}, \frac{4}{9})}$ in sämtlichen Normen für $\lambda \rightarrow 0$. Siehe Abbildung 1.

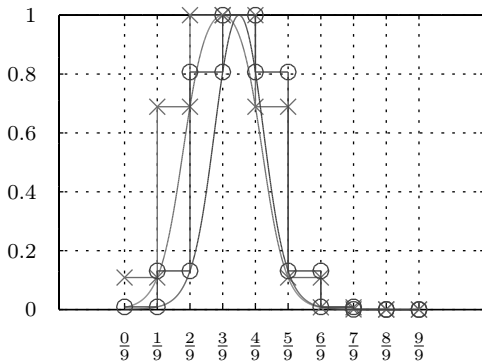


Abbildung 1: Die Aktualisierungsfunktionen $f_{\lambda,a}$ (Grün bzw. Kreuze) und $f_{\lambda,d}$ (Rot bzw. Kreise) für ein $\lambda > 0$ und deren Vektordarstellungen (Diskretisierung) für $k = 9$. Die Qualität der verwendeten Diskretisierung über das Maximum ist nicht optimal aber ausreichend.

Das resultierende nichtlineare System soll nun

über den stochastischen Formalismus approximiert werden. Der Grenzübergang $\lambda \rightarrow 0$ kann dabei zur Validierung bzw. zum Vergleich mit dem diskreten Modell $\mathbb{S} = \{0, 1\}$ verwendet werden.

2.1 Stochastische Formulierung

Wir nehmen nun an, dass die Zustände der Zellen Zufallsvariable mit Werten in $\mathbb{S} = [0, 1]$ sind und betrachten die messbare Funktion

$$\mathcal{F} : s_{n \in \mathcal{N}(m) \cup \{m\}} \mapsto \left(s_m, \sum_{n \in \mathcal{N}(m)} s_n \right) \mapsto \underbrace{[s_m f_a + (1 - s_m) f_b]}_{=: g(s_m)} \left(\sum_{n \in \mathcal{N}(m)} s_n \right) \quad (7)$$

und die Zufallsvariable $S(t + 1, m) := \mathcal{F}((S(t, n))_{n \in \mathcal{N}(m) \cup \{m\}})$.

Wir nennen den resultierenden stochastischen Prozess einen *pseudo-stochastischen Zellulären Automaten*, da

$$\mathcal{P}(S(t+1, m) \in B \mid (S(t, n))_{n \in \mathcal{N}(m) \cup \{m\}} = \mathbf{s}) = 1 \iff \mathcal{F}(\mathbf{s}) \in B. \quad (8)$$

Unter der Bedingung, dass alle geforderten Dichten existieren, kann diese Transformation von Zufallsvariablen auch als Transformation von Dichten formuliert werden.

Aus den Dichten $\rho_n(s) := \rho(t, n)(s)$ für $n \in \mathcal{N}(m) \cup \{m\}$ kann die Dichte von Σ als

$$\sigma(s) = \bigoplus_{n \in \mathcal{N}(m)} \rho_n := \rho_{n_1} * \dots * \rho_{n_N} \quad (9)$$

und daraus die Dichten $\gamma_{a,b}$ von $f_{a,d}(\Sigma)$ mit Hilfe des Transformationssatzes berechnet werden. Die Dichte $\rho(t + 1, m)(p)$ kann dann vermutlich als Integral der Form

$$\int \int \gamma_a(s) \gamma_b(t) \int \rho_m \left(\frac{c}{s} \right) \rho_m \left(\frac{c-p}{t} \right) dc dt ds \quad (10)$$

berechnet werden.

Ein anderer Ansatz um die Dichte $\rho(t + 1, m)$ zu berechnen betrachtet eine andere Zerlegung von

\mathcal{F} . Dabei ist g in Gleichung (7) eine lineare ebenfalls messbare funktionswertige Abbildung von \mathbb{S} in den Funktionenraum $\{af_a + (1-a)f_d, a \in [0, 1]\}$. Die Verteilung der Zufallsvariable $g(S_m)$ kann mit dem Transformationssatz bestimmt werden. Jede Funktion aus dem Funktionenraum ist eine Linearkombination $f = af_a + (1-a)f_d$. Die Inverse g^{-1} ist die Abbildung $f \mapsto \frac{f-f_d}{f_a-f_d} = a$. Die Dichte von $g(S_m)$ ist folglich

$$\gamma(f) = \rho(g^{-1}(f)) \left| \frac{dg^{-1}(f)}{df} \right| = \frac{\rho(a)}{\|f_a - f_d\|} \quad (11)$$

und man kann sagen, dass die funktionale Zufallsvariable $g(S_m)$ durch die Abbildung g die Dichte ρ von S_m übernimmt (*).

Dieser Ansatz und die Verknüpfung der Dichten von $g(S_m)$ und Σ soll im nächsten Abschnitt konkretisiert werden.

2.2 Diskretisierung

Wir zerlegen das Intervall $[0, 1]$ in k Kompartimente (Tabelle 1) $\mathbb{S} = \{\frac{0}{k-1}, \dots, \frac{k-1}{k-1}\}$ und betrachten die Dichten als (stochastische) Spaltenvektoren (!) ρ, σ, \dots

1	2	3	4	...	k
$\frac{0}{k-1}$	$\frac{1}{k-1}$	$\frac{2}{k-1}$	$\frac{3}{k-1}$	\dots	$\frac{k-1}{k-1}$
$[\frac{0}{k}, \frac{1}{k})$	$[\frac{1}{k}, \frac{2}{k})$	$[\frac{2}{k}, \frac{3}{k})$	$[\frac{3}{k}, \frac{4}{k})$	\dots	$[\frac{k-1}{k}, \frac{k}{k})$

Tabelle 1: Bei N Nachbarn und $\mathbb{S} = \{0, 1\}$ bietet sich als Zustandsraum für Σ die Menge $\{\frac{0}{k}, \dots, \frac{k}{k}\}$ mit $k = N + 1$ an. Anhand des Standardfalls $N = 8$ können die diskreten Werte $\frac{2}{8}$ bzw. $\frac{3}{8}$ des “Game of Life” mit halboffenen Intervallen in $[0, 1]$ identifiziert werden.

Da σ als Faltung im Raum $[0, 1]^{k+(k-1)(N-1)}$ liegt, diskretisieren wir auch $f_{a,d}$ wie in Abbildung 1 dargestellt als Vektor $\mathbf{f}_{a,d}$ in $[0, 1]^{k+(k-1)(N-1)}$. Die Funktionsauswertung $\mathbf{f}_{a,d}(\frac{j}{k-1})$ entspricht dann dem Skalarprodukt $\mathbf{f}_{a,d}^\top \mathbf{e}_j$.

Unter der Sichtweise (*) und da $\|\mathbf{f}_a - \mathbf{f}_d\|$ relativ klein ist, ersetzen wir nun die deterministische Funktionsauswertung $g(s_m)$ aus (7) durch

$$\begin{aligned} \mathbf{f} &:= \mathbb{E}[g(S_m)] = \sum_{j=0}^{k-1} g\left(\frac{j}{k-1}\right) \rho_j \\ &= \mathbf{f}_a\left(\frac{0}{k-1}, \dots, \frac{k-1}{k-1}\right) \rho + \mathbf{f}_d\left(\frac{k-1}{k-1}, \dots, \frac{0}{k-1}\right) \rho \\ &= g(\mathbb{E}[S_m]) =: G\rho \quad (12) \end{aligned}$$

und erhalten die Matrix $G \in [0, 1]^{(k-1)N+1 \times k}$.

Die Anwendung der Funktion \mathbf{f} auf die Verteilung σ lässt sich nun wie folgt motivieren: Dabei ist zu beachten, dass \mathbf{f} als Durchschnitt von \mathbf{f}_a und \mathbf{f}_d betrachtet werden kann. Ist die Masse von σ in der Nähe von $\frac{2}{8}$ und $\frac{3}{8}$ (bzw. in den entsprechenden Kompartimenten) konzentriert, so ist das Ergebnis groß bzw. nahe bei 1. Ist hingegen in diesem Bereich die Wahrscheinlichkeit gering, so soll das Ergebnis nahe bei 0 liegen. Definiert man die (Spalten-stochastische) Matrix (aus $[0, 1]^{k \times (k-1)N+1}$)

$$F := B + \mathbf{b}\mathbf{f}^\top = \begin{pmatrix} (1, \dots, 1) - \mathbf{f}^\top \\ \vdots \\ \mathbf{f}^\top \end{pmatrix}, \quad (13)$$

dann folgt für $\sigma \approx \|\mathbf{f}\|_1^{-1} \mathbf{f}$, dass $\rho := F\sigma$ die Masse bei den größeren Indizes konzentriert. Für einen Vektor, der orthogonal zu \mathbf{f} ist, konzentriert sich die Masse des Produkts in den vorderen Indizes. Das heißt, dass die Orthogonalität von σ zu \mathbf{f} entscheidet, ob die Zelle den Status “lebend” oder “tot” haben soll. Der vertikale Übergang der Zeilen (Index z) von F kann z.B. durch $z^{\lambda-1}(1 - \mathbf{f}^\top) + (1-z)^{\lambda-1} \mathbf{f}^\top$ modelliert werden (**). Es ist auf jeden Fall zu beachten, dass die Spaltensummen immer 1 sein müssen. Abbildung 2 visualisiert eine typische Massenverteilung von F .



Abbildung 2: Visualisierung der Massenverteilung (Rot entspricht großer Masse, Blau entspricht geringer Masse) in der Übergangsmatrix F bei $k = 9$, $N = 8$ und gleichverteiltem ρ . Die Funktionen $f_{a,d}$ bzw. der Parameter λ wurde von Abbildung 1 übernommen. Der vertikale Übergang in (13) wurde jedoch quadratisch modelliert.

Dieser heuristische Ansatz kann formal über

den Transformationssatz ($\mathbf{f}^\top = c\mathbf{f}^{-1}$, Frobenius–Perron Operator, ...)

$$\rho(y) = \sum_{i=1}^2 \sigma(f_i^{-1}(y)) \left| \frac{df_i^{-1}(y)}{dy} \right| \quad (14)$$

bewiesen werden, wobei hier f_i die Hälften der am (bzw. an einem) Maximum ($\max f = 1$) geteilten Funktion f bezeichnen (Abbildung 1). Dazu muss auch der Zielraum von \mathbf{f} diskretisiert werden und die resultierende Matrix ist mit jeweils einem Element gleich 1 pro Spalte das (pseudo-stochastische) Analogon zu F und beschreibt die Transformation der Zufallsvariablen Σ .

Die Matrix F aus (13) ist in gewisser Weise die Übergangsmatrix eines stochastischen Zellulären Automaten, hängt aber von der (lokalen) Verteilung ρ_m bzw. vom Erwartungswert $\mathbb{E}[S_m]$ ab. Die Matrix F kann im Gegensatz zu (**) auch aus der Matrix aus dem vorherigen Absatz durch Vor- und Nachstellen von Diffusionsmatrizen erzeugt werden.

Ausführlich angeschrieben definiert F eine deterministische Iterationsvorschrift

$$\begin{aligned} \hat{\mathcal{F}}: ([0, 1]^k)^{N+1} &\rightarrow [0, 1]^k: \\ (\rho(t, m), \rho(t, n_1), \dots, \rho(t, n_N)) &\mapsto \\ \mapsto F(\rho(t, m)) \bigoplus_{n \in \mathcal{N}(m)} \rho(t, n) &=: \rho(t+1, m), \end{aligned} \quad (15)$$

die das Verhalten der Verteilungen der Zustände eines stochastischen Zellulären Automaten mit Zustandsraum $\mathbb{S} = \{\frac{0}{k-1}, \dots, \frac{k-1}{k-1}\}$ beschreibt, der für geringe Diffusion bzw. passenden Übergang in (**) gegen den pseudo-stochastischen Zellulären Automaten aus (7) und (8) konvergiert.

Der folgende Abschnitt veranschaulicht eine Vereinfachung dieser Identifikation für parametrisierte stochastische Zelluläre Automaten am Beispiel des Zustandsraumes $\mathbb{S} = \{0, 1\}$

2.3 Diskretisierung in zwei Kompartimente

Sind die Zustände $S(t, m)$ Zufallsvariable mit Zustandsraum $\mathbb{S} = \{0, 1\}$, dann ist $S(t, m)$ Bernoulli-

verteilt mit einem Parameter $\theta(t, m) \in [0, 1]$ und die/der Wahrscheinlichkeitsfunktion/-vektor ist $\rho(t, m) = (1-\theta, \theta)^\top \in [0, 1]^2$. Der Wahrscheinlichkeitsvektor σ der Summe Σ liegt gemäß der $(N-1)$ -maligen Anwendung des Faltungsoperators in $[0, 1]^{N+1}$.

Es existiert also ein Spalten-stochastische Übergangsmatrix $F \in [0, 1]^{2 \times N+1}$ mit

$$\rho_{t+1} = \begin{pmatrix} 1-\theta_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} (1, \dots, 1) - \mathbf{f}_t^\top \\ \mathbf{f}_t^\top \end{pmatrix} \sigma_t. \quad (16)$$

Da σ einer verallgemeinerten Binomialverteilung (auch Poisson-Binomialverteilung genannt) genügt, kann die Massenverteilung explizit angegeben werden

$$\sigma_i(t, m) = \sum_{\substack{|A|=(i-1) \\ A \subseteq \mathcal{N}(m)}} \left[\prod_{n \in A} \theta(t, n) \prod_{l \in A^c} (1 - \theta(t, l)) \right] \quad (17)$$

wobei A die $(i-1)$ -elementigen Untermengen der N -elementigen Menge $\mathcal{N}(m)$ durchläuft. Weiter ist $\mathbf{f}_t = \theta_t \mathbf{f}_a + (1 - \theta_t) \mathbf{f}_d$ also insgesamt

$$\theta(t+1, m) = (\theta(t, m)(\mathbf{f}_a - \mathbf{f}_d)^\top + \mathbf{f}_d^\top) \sigma(t, m). \quad (18)$$

(18) beschreibt einen deterministischen Zellulären Automaten, der *konjugiert* zum pseudo-stochastischen Zellulären Automaten mit messbarer Aktualisierungsvorschrift (7) ist. Diese Identifikation trifft immer dann zu, wenn die stochastischen Zustände mit einer Verteilungsfamilie verteilt sind $S(t, m) \sim W_\Theta$.

3 Ergebnisse und Ausblick

Ein experimentelle Vergleich der Verschiedenen stochastisch-diskretisierten Modelle mit dem deterministisch-kontinuierlichen System wurde in *GNU Octave* implementiert. Abbildung 3 zeigt visualisierte Zellzustände der verschiedenen Ansätze und deutet den Grenzübergang $\lambda \rightarrow 0$ an.

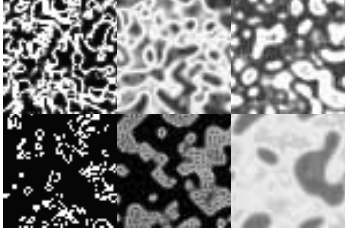


Abbildung 3: Erste Spalte: kontinuierliche Zustände; Zweite Spalte: Wahrscheinlichkeitsdichte bei zwei möglichen stochastischen Zuständen; Dritte Spalte: 10 Kompartimente, Summe der Wahrscheinlichkeiten der zweiten Hälfte der Kompartimente; In der oberen Zeile $\lambda = 1$ und in der unteren Zeile $\lambda = 0.05$. Die Kantenlänge des Zellgitters ist 50 und die Anzahl der Iterationsschritte ist 20.

3.1 Zeit-kontinuierliche Formulierung

Wie man aus (13) sieht, lässt sich (15) als multilineare Abbildung formulieren. Für $\mathbf{a} := (\frac{0}{k-1}, \dots, \frac{k-1}{k-1})$ und $\mathbf{d} := 1 - \mathbf{a}$ gilt

$$B + \mathbf{b}\mathbf{f}^\top = (B\mathbf{a} + B\mathbf{d} + \mathbf{b}\mathbf{f}_a^\top \mathbf{a} + \mathbf{b}\mathbf{f}_d^\top \mathbf{d})\rho. \quad (19)$$

Also ist (15) auch eine lineare Abbildung von Tensoren

$$\hat{\mathcal{F}} : ([0, 1]^k)^{\otimes(N+1)} \rightarrow [0, 1]^k \quad (20)$$

und der Zelluläre Automat kann formal im Kontext von (1) betrachtet werden [4]. Für das kontinuierliche (siehe (5)) bzw. das parametrisierte Modell (18) ist das nicht offensichtlich.

Für die Überführung in ein Differentialgleichungsmodell ist außerdem eine Neuformulierung der Nachbarschaftsabbildung mit Gewichtungen notwendig. Für die Gewichtung der Nachbarzellen bietet sich typischerweise eine zweidimensionale Normalverteilung an, die als Diffusionsprozess interpretiert werden kann. Zuerst wird wieder ein kontinuierliches Modell formuliert

$$S(t+h, x) = [S(t, x)f_{h,a} + (1 - S(t, x))f_{h,b}] \left(\int_{\Omega} \alpha_h(y-x)S(t, y) dy \right). \quad (21)$$

Über eine lokale Taylorreihendarstellung gelangt man von der Integraldarstellung zum Laplace Operator.

3.2 Statistische Methoden

Das Auftreten der unregelmäßigen Strukturen in den stochastischen Modellen hängt stark vom Eigenvektor zum Eigenwert 1 der Übergangsmatrix F ab. Um von Eigenvektoren sprechen zu können, muss zuerst eine quadratische Version von F betrachtet werden. Für gewöhnlich, falls F eine Abbildung im selben Vektorraum ist, spricht man von einer stationären Verteilung. In der verwendeten Notation entspricht F der Funktion $\hat{\mathcal{F}}(\rho, \cdot)$.

Die Abbildung $\rho \mapsto \hat{\mathcal{F}}(\rho, \cdot)$ verändert die stationäre Verteilung kaum, da sich f_a und f_d nur minimal unterscheiden (das hängt natürlich von der Modellierung (6) ab). Wesentlich ist, dass die (verallgemeinerte) Binomialverteilung bzw. Normalverteilung (Zentraler Grenzwert Satz) σ hauptsächlich ihren Erwartungswert verändert je nach Zustand der Nachbarzellen.

Liegt der normierte Erwartungswert $\mathbb{E}[\Sigma]$ im Bereich des Maximums der Abbildung f also im Bereich $\frac{2}{8}$ bis $\frac{3}{8}$, so ist ein großer Wert für S im nächsten Zeitschritt wahrscheinlich (Das Skalarprodukt $f \cdot \rho$ ist groß). Zusammenfassend kann man das erwartete mittlere Verhalten des stochastischen Systems durch die Iterationsvorschrift $\mathcal{F}(\mathbb{E}[S], \mathbb{E}[\Sigma])$ beschreiben.

Betrachtet man das Modell mit zwei Kompartimenten so gilt $\mathbb{E}[S] = \theta_m$ und $\mathbb{E}[\Sigma] = \sum_{n \in \mathcal{N}(m)} \theta_n$. Die Iterationsvorschrift stimmt also wie erwartet mit der Formulierung des kontinuierlichen Systems (5) überein. Der selbe Zusammenhang kann auch für das k -Kompartiment Modell gezeigt werden.

Aus den Experimenten kann man erkennen, dass eine große Varianz der stationären Verteilung verhindert, dass sich die Verteilungen der Zellzustände einer stationären Verteilung annähern.

3.3 Schlußfolgerung

Auch bei einer exakten mathematischen Beschreibung können keine allgemeingültigen Verfahren zur Identifikation von deterministischen und stochastischen Modellen eines Evolutions Sys-

tems abgeleitet werden. Vielmehr zeigt sich, dass basierend auf der Diskretisierung von Wahrscheinlichkeitsverteilungen bzw. des Zustandsraumes individuelle Ansätze verwendet werden müssen.

Es lässt sich daraus folgern, dass vermutlich auch für den Übergang von kontinuierlichen zu diskreten Modellen individuelle Methoden notwendig sind.

Diese Ansätze leiten sich jedenfalls zum größten Teil von den gegebenen lokalen Aktualisierungsvorschriften ab.

References

- [1] K.J. Engel und R. Nagel. *One-Parameter Semigroups for Linear Evolution Equations*. In: Graduate Texts in Mathematics, Springer-Verlag, New York, 2000.
- [2] J.A. Goldstein. *Semigroups of Linear Operators and Applications*. In: Oxford Mathematical Monographs. Oxford University Press, New York, 1985.
- [3] K. Taira. *Semigroups, Boundary Value Problems and Markov Processes*. In: Springer Monographs in Mathematics, Springer-Verlag, Berlin, 2004.
- [4] G. Schneckenreither. *Zelluläre Automaten als Methode in der Systemsimulation*. Diplomarbeit (n.v.), Technische Universität Wien, Österreich.
- [5] D. Leao Jr., M. Fragoso, und P. Ruffino. *Regular Conditional Probability, Disintegration of Probability and Radon Spaces*. In: Proyecciones Journal of Mathematics 23.1, S. 15–29, 2004.

Frühzeitige Serverausfallserkennung mittels Vorhersagemethoden und Outlieranalyse

Matthias Wastian¹, Dr. Felix Breitenecker², Michael Landsiedl¹

¹dwh GmbH, Neustiftgasse 57-59, 1070 Wien, Österreich

²Technische Universität Wien, Wiedner Hauptstraße 8-10, 1040 Wien, Österreich

matthias.wastian@dwh.at

In dem vorliegenden Paper werden unterschiedliche Ansätze diskutiert, um ungewöhnliche und außerordentliche Ereignisse, für die man eine separate und spezialisierte Betrachtung als wünschenswert erachtet, anhand von in Form einer multivariaten Zeitreihe regelmäßig mitprotokollierten Daten möglichst frühzeitig zu detektieren und bei entsprechender Möglichkeit sogar vorherzusagen. Nach einem kurzen Überblick über diverse Anwendungsbereiche, in denen diese Fragestellung der abnormalen Eventdetektion auftritt, wird der Fokus auf die frühzeitige Erkennung von Serverausfällen gelegt. Die zwei elementaren Bausteine des vorgestellten Modells sind - nach geeigneter Datenvorverarbeitung der großen Anzahl der durch Server Monitoring gemessenen Server Features - ein univariater Prädiktor und ein multivariater Ausreißer-detektor. Großen Einfluss auf diese Modellbausteine haben dabei Methoden aus den Bereichen des Data Mining, des Machine Learning und des Soft Computing.

Als Prädiktoren werden sowohl klassische SARIMA-Modelle als auch Neuroprädiktoren, die einen bestimmten Typ von künstlichen Neuronalen Netzen zur Vorhersage nutzen, vorgestellt. Auf eine kurze Präsentation von Werkzeugen zur Ausreißer- bzw. Novumserkennung, insbesondere einer One-Class Support Vector Machine und des Ansatzes der winkelbasierten Ausreißerdetektion, folgt eine abschließende Diskussion der bis dato erzielten Simulationsergebnisse. Der Text endet mit einem Ausblick auf mögliche Modellerweiterungen und zukünftige Arbeiten.

1 Grundbegriffe und Anwendungen von abnormaler Eventdetektion

Definition 1 (Event). Ein Ereignis wird als ein Vorkommnis definiert, das zu einer bestimmten Zeit an einem bestimmten Ort geschieht und eine bestimmte Dauer besitzt. Dieses Vorkommnis kann Teil einer Kette von Vorkommnissen sein und von einem vorherigen Event verursacht werden oder ein weiteres Event auslösen. Es ist möglich, dass mehrere Events zur gleichen Zeit und/oder am selben Ort auftreten.

Definition 2 (Abnormales Event). Ein abnormales Event wird als Ausreißer in einer Eventkette definiert und ist somit ein Event, das so weit von den anderen Events abweicht, so dass ein begründeter Verdacht

auftritt, dass es von etwas verursacht worden ist, dass nicht vom Normalverhalten des Systems herrührt, und dass es das gesamte Systemverhalten nachhaltig ändern könnte.

Ausgehend von obigen Definitionen ist es naheliegend, dass abnormale Eventdetektion vor allem bei der Betrachtung von derartigen Systemen von Interesse ist, bei denen die Gewährleistung eines Mindestmaßes an Sicherheit relevant ist. Neben dem diesem Paper zugrunde liegenden Beispiel der Serverausfallsvorhersage sind folgend einige weitere Anwendungsbereiche aufgelistet:

- Vorhersage/Früherkennung von Naturkatastrophen wie Überflutungen, Hurricanes, Erdbeben oder Tsunamis
- Vorhersage/Früherkennung von Aktienmarkttein-

brüchen

- Erkennung von unbefugten Netzwerkzugriffen
- Audio- und Videoüberwachung: Erkennung von Massenpaniken, Verkehrsanalyse
- Ambient Assisted Living.

1.1 State of the Art

Es ist eingangs zu bemerken, dass abnormale Eventdetektion und vor allem die dafür sinnvollerweise verwendbaren Methoden massiv von der Struktur der gewonnenen Daten sowie von der Art des Systems abhängen.

Für den verwendeten Prädiktor, der auf den statistischen ARIMA-Modellen beruht, liefert [1] eine gute Zusammenfassung der theoretischen Grundlagen, anhand derer direkte zeitliche Abhängigkeiten, Saisonalität, Differenzenbildung u.Ä. modelliert werden. Die Verwendung von künstlichen Neuronen Netzen (NN) als Vorhersagewerkzeug handeln u.a. [2] und [3] ab.

Im Bereich der durchaus auch in einem hochdimensionalen Raum notwendigen Ausreißeranalyse zur abnormalen Eventdetektion wird die Anwendung einer One-Class Support Vector Machine (OC-SVM) u.a. in [4], [5], [6], [7] und [8] vorgeschlagen, diskutiert und evaluiert. Alternativ stellen wir die winkelbasierte Ausreißerdetektion vor, deren Grundlagen in [9] erarbeitet wurden. Ein beschleunigter Algorithmus findet sich in [10].

Weiters stechen als Methoden zur abnormalen Eventdetektion insgesamt die Verwendung von statistischen Methoden ([11]), Clustering ([12]), Replicator Neural Networks ([13]), Sparse Reconstruction Cost ([14]) und Wavelet Decomposition ([15]) heraus.

2 Daten

Die für die Simulationsläufe verwendeten Server sind IBM Lotus Notes Server, von denen mittels Server Monitoring bis zu 1439 unterschiedliche Features mit einer konstanten Abtastfrequenz zwischen einmal pro Minute und einmal pro 15 Minuten gemessen wurden. Neben historischem Datenmaterial stand auch ein

Software Tool von IBM zur Verfügung, dass künstliche Datensets generieren kann, indem jeder Client eine simulierte Benutzerlast zum Server schickt, der die mitprotokollierten Statistiken retourniert.

Um diese enorme Datenmenge einzuschränken, wurde mittels Expertenwissen der Serveradministratoren eine Kategorisierung aller Server Features in vier unterschiedliche Prioritätsklassen vorgenommen. 14 Features erhielten die höchstgögliche Priorität 0, 73 entweder Priorität 0 oder Priorität 1. Die meisten Simulationsläufe wurden mit diesen beiden Gruppen durchgeführt.

Akkumulierte Werte (z.B. die Anzahl der verschickten Mails seit Start des Server Monitoring) wurden differenziert und nach einem kurzen Check auf offensichtlich falsche Messdaten wurden diese sowie Messungen, die keinen Wert (bzw. einen NaN-Wert) geliefert hatten, aus dem Datensatz gelöscht. Keiner der vorgeschlagenen Algorithmen kann nämlich mit NaN-Values umgehen.

Da sich die Wertebereiche der Messgrößen teilweise gravierend voneinander unterscheiden, wurde eine Normalisierung der Daten durchgeführt. Vor Verwendung des Neuro-Prädiktors wurde insbesondere ein sog. Minmax-Mapping angewandt:

Algorithmus 1 (Minmax-Mapping). Gegeben sei der Datensatz $\{x_1, \dots, x_n\}$ mit Minimum x_{min} und Maximum x_{max} . Das folgende Mapping transformiert diesen affin zu $\{y_1, \dots, y_n\}$ im Intervall $[y_{min}, y_{max}]$.

$$f_{minmax} : [x_{min}, x_{max}] \rightarrow [y_{min}, y_{max}],$$

$$x \mapsto y = \frac{(y_{max} - y_{min})(x - x_{min})}{x_{max} - x_{min}} + y_{min}$$

y_{min} und y_{max} sind im Prinzip frei wählbar; da damit Inputgrößen für ein NN generiert werden, wurden $y_{min} = -0.6$ und $y_{max} = 0.6$ gesetzt. Als Alternative zum Minmax-Mapping ist auch ein klassischer Z-Score denkbar.

3 Prädiktoren

Die Prädiktoren des vorgestellten Serverausfallserkennungsmodells arbeiten univariat. Für jedes der m Features liegt eine Zeitreihe mit einer gewissen Anzahl von in der Vergangenheit erfolgreich gemessenen

Werten vor. Der aktuelle Wert von Feature i wird mit $x_{i,n}$ bezeichnet. Es gilt also, die nächsten Messwerte $x_{i,n+1}, i = 1, \dots, m$ möglichst akkurat vorherzusagen. Von einer multivariaten Vorhersage wurde aufgrund der deutlich schlechteren Performance (längere Rechenzeiten, größere Vorhersagefehler) und der Tatsache, dass die kausale Abhängigkeit der Features untereinander sehr sparse ist, nach einigen Tests abgesehen.

Grundlegende Modellannahme ist, dass die Prädiktoren im gewünschten Normalbetrieb (i.e., wenn keine abnormalen Events auftreten) gute Vorhersagen liefern; die Vorhersagefehler zumindest zu Beginn eines abnormalen Events aber einer anderen Verteilung entspringen.

Alle Prädiktoren müssen relativ starke saisonale Effekte berücksichtigen. So hängt für einen durchschnittlichen Büroserver die Anzahl der eingeloggten User an einem Montagmorgen um 8:00 sicherlich stärker von dem entsprechenden Wert exakt eine Woche zuvor ab als von jenem am selben Montag um 7:00.

3.1 SARIMA-Modell

Durch Analyse der partiellen (PACF) und der Autokorrelationsfunktion (ACF) einer univariaten Zeitreihe werden zunächst die Differenzenordnungen d und D sowie anschließend die restlichen Modellparameter bestimmt, um das $(p, d, q) \times (P, D, Q)_s$ -SARIMA-Modell zu erstellen. Parameter, die mit Großbuchstaben bezeichnet werden, gehören dabei zum saisonalen Teil mit Periode s . p und P bezeichnen die Anzahlen der autoregressiven Terme, q und Q die der Moving-Average-Terme. Wie genau so eine Analyse vonstattegeht, beschreibt [1].

3.2 Neuro-Prädiktor-Modell

Künstliche Neuronale Netze eignen sich aufgrund ihrer nichtlinearen, datengesteuerten Struktur sehr gut als Vorhersagemodell. Auch wenn sie Saisonalität implizit mitabbilden können, wurde eine explizite Modellierung bevorzugt, um die Anzahl der Inputneuronen klein zu halten. Ein Neuro-Prädiktor sucht also eine möglichst gute Funktion f mit $x_{n+1} = f(x_n, x_{n-1}, \dots, x_{n-d+1}, x_{n-s}, \dots, x_{n-2s}, \dots)$.

Anhand der gegebenen Zeitreihe ist es einfach, ge-

labelte Daten für das NN zu erstellen, um dieses überwacht lernen zu lassen. Die vorhandenen Daten werden dem Konzept der Cross-Validierung folgend aufgeteilt: 70% bilden den Trainings-, 15% den Test- und 15% den Validierungsdatensatz. Laut [16] ist es sinnvoll, die Kardinalität des Trainingssets in etwa als Quotient der Anzahl der Gewichte des NN und des Klassifizierungsfehlers zu wählen. Das vorgeschlagene NN besitzt einen Input-Layer mit so vielen (n_i) Neuronen, wie Variable in der obigen Funktion f auftreten, einen Output-Layer mit einem Neuron ($n_o = 1$), das den Vorhersagewert x_{n+1} ausspuckt, sowie eine versteckte Schicht, deren Neuronenanzahl mittels der geometrischen Pyramidenregel als $n_h = \alpha \sqrt{n_i n_o}, \alpha \in [0.5, 2]$, festgelegt wird. Aktivierungsfunktion im versteckten Layer ist der tanh, jene in der Output-Schicht die lineare. Der Levenberg-Marquardt-Trainingsalgorithmus stoppt, wenn eine der Abbruchbedingungen erfüllt ist:

- Die Anzahl der Trainingsepochen überschreitet das vorgegebene Maximum.
- Die Anzahl aufeinanderfolgender Epochen mit steigendem Fehler im Validierungsset überschreitet den vorgegebenen Wert (etwa 6).
- Der Fehler im Testdatensatz liegt unter einem vorgegebenen Zielwert.

3.3 Vergleich

Stehen mehrere Prädiktor-Modelle vergleichbarer Qualität zur Auswahl, sollte jenes mit dem niedrigsten AIC- oder BIC-Wert verwendet werden.

Ob SARIMA-Modelle oder Neuro-Prädiktoren effizienter sind, beantwortet die Fachliteratur unterschiedlich. In [16] wurden folgende Aspekte herausgearbeitet: Haben Zeitreihen ein langes Gedächtnis, sind die Ergebnisse ähnlich. Bei einem kurzen Gedächtnis liefern NN manchmal deutlich bessere Resultate. Bei Zeitreihen unterschiedlicher Komplexität ist ein optimal getunt NN ebenfalls effizienter.

Im konkreten Anwendungsbeispiel der Serverausfallserkennung kommt noch hinzu, dass gemeinsame Parameter für alle Zeitreihen das Modell deutlich vereinfachen. Dies ist mit Neuro-Prädiktoren sicherlich effizienter umsetzbar.

4 Ausreißerdetektoren

Grundlage für die Ausreißerdetektoren ist eine multivariate Analyse der Vorhersagefehler durch einen der beiden oben vorgestellten Prädiktoren. Modelliert wird die Ausreißeranalyse als $(1 + x)$ -Klassifizierungs-Task, d.h. modelliert wird nur die Klasse der normalen Daten. Der Zugang entspricht halbüberwachtem Lernen (vgl. [17]). Neben diesem Modellierungsansatz war auch die Dimension der Vorhersagefehler entscheidend für die Methodenwahl; rein distanzbasierte Verfahren scheitern in hochdimensionalen Räumen aufgrund des Fluchs der Dimensionalität.

Noch nicht implementiert wurde der vielversprechend klingende Ansatz von [20], der auf der Verwendung von Genetischen Algorithmen für Projektionen in niedrigdimensionale Räume beruht.

4.1 Winkelbasierte Ausreißererkennung

Seien A, B und C beliebige Punkte in einem Datensatz \mathcal{D} mit n m -dimensionalen Vektoren. Der winkelbasierte Outlier-Faktor $ABOF$ von einem Punkt A sei definiert als:

$$ABOF(A) = \text{VAR}_{B, C \in \mathcal{D}} \left(\frac{\langle \vec{AB}, \vec{AC} \rangle}{\|\vec{AB}\|^2 \|\vec{AC}\|^2} \right)$$

Je kleiner $ABOF(A)$, desto größer die Zugehörigkeit von A zur Klasse der Outlier. Auch wenn $ABOF$ ebenfalls Distanzen verwendet, so ist deren Einfluss beim winkelbasierten Ansatz deutlich geringer als bei rein distanzbasierten Methoden.

Um den Rechenaufwand zu reduzieren, kann anstatt der Varianz aller möglichen Winkel mit Scheitel A auch nur die Varianz jener Winkel hergenommen werden, die durch die k nächsten Nachbarn von A entstehen. Der Algorithmus wird damit $O(m^2 + k^2 m)$ statt $O(m^3 n)$.

4.2 One-Class Support Vector Machine

Die OC-SVM kann als reguläre Zwei-Klassen-SVM interpretiert werden, bei der alle Trainingsdaten in der

ersten Klasse liegen und der Ursprung das einzige Element der zweiten Klasse ist. Es wird versucht jene Hyperebene W mit Normalvektor w und Bias b zu finden, die die meisten, aber nicht alle Elemente der ersten Klasse mit maximalem Abstand vom Ursprung abtrennt. Dass nicht alle Elemente abgetrennt werden, verhindert den Overfitting-Effekt.

$$wx - b = 0$$

Neben einem Gauss-Kernel κ (vgl. [18]) werden Schlupfvariable ξ_i verwendet, um die Fehlerterme unter einem vorgegebenen Budget zu halten.

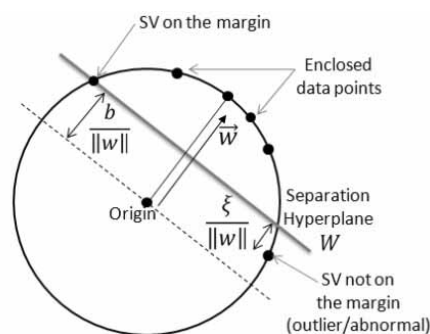


Abbildung 1: One-Class Support Vector Machine ([8])

Das OC-SVM Optimierungsproblem ist äquivalent zum dualen quadratischen Programmierungsproblem mit Lagrange-Multiplikatoren α , das mit Standardmethoden und unter Verwendung der Karush-Kuhn-Tucker-Bedingungen gelöst werden kann. Ein neuer Vorhersagefehlervektor wird dann anhand der Entscheidungsfunktion

$$f(x) = \sum_i \alpha_i \kappa(x_i, x) - b$$

bewertet, die positiv für Inlier und negativ für Outlier ist.

5 Simulationsergebnisse und Ausblick

Eingangs soll bemerkt werden, dass eine exakte Definition des Terminus *Serverausfall* auch in Expertenkreisen nicht existiert. Jedwede Einschränkung

des Betriebs ist aber definitiv unerwünscht. Während mehrere Testläufe erkannten beide Ausreißerdetektoren mit Leichtigkeit, wenn der Status eines Servers von idle zu busy wechselte oder umgekehrt. Eine umfangreiche Modellvalidierung kann aber letztlich erst bei Veröffentlichung der entsprechenden Software und ausreichenden Tests im Alltagsbetrieb erfolgen. Als Benchmark-Datensatz wurden zudem die Ölpreise über mehrere Jahrzehnte hergenommen: Die Ölkrise von 1979 wurde als abnormales Event eindeutig erkannt.

Mittels mehrerer Tuning Parameter kann die Sensitivität der Serverausfallserkennung vom jeweiligen Systemadministrator bei Bedarf individuell bestimmt werden. Da mehrere Modellteile erst trainiert werden müssen, braucht die Gesamtsoftware im Anwendungsfall eine gewisse Zeit, bis sie sinnvolle Warnungen ausgeben kann.

Noch nicht umgesetzt, aber naheliegend sind eine Kombination der Ergebnisse der beiden Ausreißerdetektoren sowie eine Unifizierung der Outlierscores (vgl. [19]). Alternativ scheint auch der Ansatz von [20] mittels Projektionen in niedrigdimensionale Räume und der Verwendung eines Genetischen Algorithmus vielversprechend.

Literatur

- [1] Robert Nau. *Forecasting - Decision 411*, Online-Kurs, verfügbar unter people.duke.edu/~rnau/411home.htm, 2005.
- [2] G. Peter Zhang, Douglas Kline. *Quarterly Time-Series Forecasting With Neural Networks*. *IEEE Transactions on Neural Networks*, Volume 8, Nummer 6, IEEE Computational Intelligence Society, S. 1800-1814, 2007.
- [3] Sven Crone, Rohit Dhawan. *Forecasting Seasonal Time Series with Neural Networks: A Sensitivity Analysis of Architecture Parameters*. Tagungsband *International Joint Conference on Neural Networks 2007*, IEEE, Orlando, Florida, S. 2099-2104, 2007.
- [4] Katherine Heller, Krysta Svore, Angelos Keromytis, Salvatore Stolfo. *One Class Support Vector Machines for Detecting Anomalous Windows Registry Accesses*. Tagungsband *Workshop on Data Mining for Computer Security der IEEE International Conference on Data Mining 2003*, Melbourne, Florida, S. 2-9, 2003.
- [5] Paul Evangelista, Piero Bonnisone, Mark Embrechts, Boleslaw Szymanski. *Fuzzy ROC Curves for the 1 Class SVM: Application to Intrusion Detection*. Tagungsband *13th European Symposium on Artificial Neural Networks 2005*, d-side, Bruges, Belgien, S. 345-350, 2005.
- [6] Riu Zhang, Shaoyan Zhang, Yang Lan, Jianmin Jiang. *Network Anomaly Detection Using One Class Support Vector Machine*. Tagungsband *MultiConference of Engineers and Computer Scientists 2008*, Volume 1, IAENG, Hong Kong, 2008.
- [7] Stephan Dreiseitl, Melanie Osl, Christian Scheibböck, Michael Binder. *Outlier Detection with One-Class SVMs: An Application to Melanoma Prognosis*. Tagungsband *AMIA Annual Symposium 2010*, S. 172-176, 2010.
- [8] Sébastien Lecomte, Régis Lengellé, Cédric Richard, Francois Capman, Bertrand Ravera. *Abnormal Events Detection using Unsupervised One-Class SVM - Application to Audio Surveillance and Evaluation*. Tagungsband *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance 2011*, IEEE, Klagenfurt, Österreich, S. 124-129, 2011.
- [9] Hans-Peter Kriegel, Matthias Schubert, Arthur Zimek. *Angle-Based Outlier Detection in High-Dimensional Data*. Tagungsband *14th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining 2008*, Las Vegas, Nevada, ACM, New York, S. 444-452, 2008.
- [10] Ninh Pham, Rasmus Pagh. *A Near-Linear Time Approximation Algorithm for Angle-Based Outlier Detection in High-Dimensional Data*. Tagungsband *18th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining 2012*, ACM, New York, S. 877-885, 2012.
- [11] Valery Guralnik, Jaideep Srivastava. *Event Detection from Time Series Data*. Tagungsband *5th ACM SIGKDD International Conference on*

- Knowledge Discovery and Data Mining*, ACM New York, USA, S. 33-42, 1999.
- [12] Fan Jiang, Ying Wu, Aggelos Katsaggelos. *Abnormal Event Detection Based on Trajectory Clustering by 2-Depth Greedy Search*. Tagungsband *IEEE International Conference on Acoustics, Speech and Signal Processing 2008*, IEEE, Las Vegas, Nevada, S. 2129-2132, 2008.
- [13] Simon Hawkins, Hongxing He, Graham Williams, Rohan Baxter. *Outlier Detection Using Replicator Neural Networks*. Tagungsband *4th International Conference on Data Warehousing and Knowledge Discovery 2002*, Aix-en-Provence, Frankreich, veröffentlicht in *Lecture Notes in Computer Science 2454*, Springer, S. 113-123, 2002.
- [14] Yang Cong, Junsong Yuan, Ji Liu. *Sparse Reconstruction Cost for Abnormal Event Detection*. Tagungsband *24th IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Colorado Springs, Colorado, S. 3449-3456, 2011.
- [15] Mitsutoshi Suzuki, Hitoshi Ihara. *Development of Safeguards System Simulator Composed of Multi-Functional Cores*. *Journal of Power and Energy Systems*, Volume 2, Nummer 2, J-Stage, Japan, S. 899-907, 2008.
- [16] Aloy Palit, Dobrivoje Popovic. *Computational Intelligence in Time Series Forecasting - Theory and Engineering Applications*. Springer Verlag, London, 2005.
- [17] Victoria Hodge, Jim Austin. *A Survey of Outlier Detection Methodologies*. *Artificial Intelligence Review*, Volume 22, Ausgabe 2, Kluwer Academic Publishers, Niederlande, S. 85-126, 2004.
- [18] Bernhard Schölkopf, Alexander Smola. *Learning with Kernels*. MIT Press, Cambridge, Massachusetts, 2002.
- [19] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, Arthur Zimek. *Interpreting and Unifying Outlier Scores*. Tagungsband *11th SIAM International Conference on Data Mining*, Mesa, Arizona, 2011.
- [20] Charu Aggarwal, Philip Yu. *An Effective and Efficient Algorithm for High-Dimensional Outlier Detection*. *The VLDB Journal 14*, Springer-Verlag, S. 211-221, 2005.

Zur Entwicklung des Gleichungslösersubsystems der verteilten parallelen Simulationsumgebung

V. Kushnarenko^{1,3}, M. Resch², V. Svjatnyj¹, S. Wesner³

¹ Nationale technische Universität Donezk (DonNTU), Fakultät für Computerwissenschaften und Technologien (FCWT), Artemstraße 58, 83001 Donezk, Ukraine

² Höchstleistungsrechenzentrum Stuttgart (HLRS), Nobelstraße 19, 70550 Stuttgart, Deutschland

³ Kommunikations- und Informationszentrum (KIZ) Universität Ulm, Albert-Einstein-Allee 11, 89081 Ulm, Deutschland

kushnar_vova@mail.ru; resch@hlrs.de; vsvjatnyj@gmail.com; stefan.wesner@uni-ulm.de

Das Problem der Entwicklung von vollfunktionellen Modellierungs- und Simulationssoftware für die verteilte parallele Simulationsumgebung (VPSU) wird durch die VPSU-Dekomposition auf bestimmte Menge von Subsystemen gelöst. Die Hauptanforderungen an Subsysteme der VPSU wurden definiert. Funktionalität, Strukturorganisation und Entwicklungskonzeption des Subsystems des Gleichungslösers wurden dargestellt. Erfahrungen der Implementierung von Gleichungslösern zeigen, dass die Verwendung von neuen blockartigen numerischen Verfahren (BAV) gute Perspektive hat, weil die BAV höhere Genauigkeit und Konvergenz, besseres Verhältnis „Genauigkeit-Geschwindigkeit“ im Vergleich mit traditionellen Verfahren haben. Die parallele BAV-Lösern werden im Rahmen der Zusammenarbeit von FCWT und HLRS entwickelt und auf dem Beispiel des dynamischen Netzobjektes mit verteilten Parametern untersucht.

1 Einleitung

Das Problem der Entwicklung von vollfunktionellen Modellierungs- und Simulationssoftware für die verteilte parallele Simulationsumgebung (VPSU) wird durch die VPSU-Dekomposition auf bestimmte Menge von Subsystemen gelöst [1]. Die zwischen den Subsystemen verteilte gesamte VPSU-Funktionalität wird aus der Darstellung von komplexen dynamischen Systemen (KDS) als Objekten der parallelen Modellierung und Simulation abgeleitet (Abb.1).

Das VPSU-Subsystem ist eine Teilkomponente von VPSU-Hardware, Systemsoftware, Modellierungs- und Simulationssoftware, die benutzerfreundlich eine gewisse Gruppe von inhaltlich nahen Funktionen auf den entsprechenden Etappen der parallelen Modellierung und Simulation von komplexen dynamischen Systemen mit konzentrierten und verteilten (DSKP, DSVP, Abb.1) Parametern realisiert [2]. Betrachten wir die Hauptfragen der Entwicklung des VPSU-Subsystems der Gleichungslösern (SuGL).

2 Die Hauptanforderungen

Subsystem soll den VPSU-Anforderungen [1] entsprechen. Die subsystemspezifischen Anforderungen werden von Subsystementwickler mit der Berücksichtigung der Anforderungen an Simulationsmitteln,

Subsystemfunktionen sowie VPSU- Entwicklungskonzept [1] formuliert und mit den Entwicklern der anderen Subsysteme sowie mit dem Projektleiter vereinbart. Seitens des VPSU-Projektentwicklers wurden folgende Hauptanforderungen zum Subsystem der parallelen Gleichungslösern erläutert.

1. Benutzerfreundlichkeit des Subsystems bedeutet transparente Präsentation von vorhandenen Gleichungslösern, Anwesenheit der notwendigen Erklärungen bezüglich der ausgewählten numerischen Verfahren, bequeme Eingabe der relevanten Parametern, die Möglichkeit der interaktiven Änderungen von Modellparametern, präsentationsdeutliche grafische Darstellung der Simulationsergebnissen bzw. Gleichungssystemlösungen.

2. **Objektorientierte Entwicklung der parallelen Gleichungslöser** ist die Voraussetzung für die Integration der für Benutzer zur Verfügung stehenden konventionellen Simulations- und Entwicklungswerkzeugen mit der Ressourcen der verteilten parallelen Simulationsumgebung.

3. Es sollen die neue parallele numerische Verfahren und Ansätze zur Löserentwicklung, wie beispielsweise die blockartigen Verfahren (BAV) für Cauchy-Probleme, angewendet werden.

4. Anwendung der bekannten Gleichungslösern bzw. numerischen Verfahren mit der Parallelisierungsansätzen entsprechend den DSKP-, DSVP-Topologien wird für die Nachhaltigkeit des Subsystems hinsichtlich der vorhandenen Erfahrung in den verschiedenen Gegenstandsbereichen beitragen.

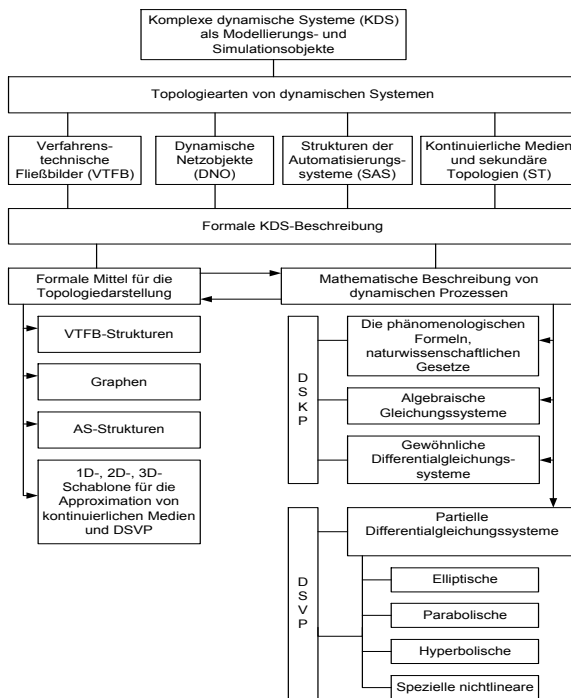


Abbildung 1. Komplexe dynamische Systeme als Objekte der Modellierung und Simulation

3 Funktionalität und Strukturorganisation des Subsystems

Das Subsystem der Gleichungslösern soll die folgenden Hauptfunktionen erfüllen:

- Bereitstellung der Bibliothek von parallelen numerischen Verfahren und effizienten parallelen Gleichungslösern;
- Kommunikation mit den Subsysteme der Topologieanalyse, Gleichungsgenerierung und der virtuellen parallelen Simulationsmodelle (VPSM), Zusammenfassung des Simulationsproblems bezüglich der untersuchten komplexen DSKP, DSVP und ihren Topologien;
- Auswahl der parallelen numerischen Verfahren und Formierung der diskreten VPSM auf möglichen Parallelitätsebenen von dynamischen Systemen;

- Planung der MIMD-Implementierung von diskreten VPSM auf bestimmten Parallelitätsebenen;
- das Lösen von Gleichungssystemen der ausgewählten diskreten VPSM auf vorgegebener
- Parallelitätsebene mit Hilfe der in den DSKP-, DSVP-Bibliotheken eingebauten parallelen
- Gleichungslösern;
- die Analyse der Konvergenz, Stabilität und Genauigkeit der Lösungen, Optimierung der veränderlichen Parametern;
- Abschätzung und Optimierung der Effizienzmerkmalen von parallelen Lösungen in einem Vergleich mit den sequentiellen Verfahren;
- Formierung der Lösungsergebnisse für die anschauliche Präsentation mit Hilfe der Mittel des Visualisierungssubsystems;
- Interaktive Bereitstellung der obigen Funktionen in Integration mit den Mittel des VPSU-Dialogsubsystems.

Die Funktionalität wird über die numerische Verfahren, Parallelisierungsansätze für DS-Topologiearten und HW/SW-Strukturorganisation des Subsystems (Abb.2) realisiert.

Ein Lösungsprozess von DSKP-, DSVP-Gleichungssystemen vom Start mittels Benutzeroberfläche (BOF) bis der Visualisierung zeigt Abb.3.

4 SuGL-Entwicklungskonzeption

4.1 Problemstellung:

Es ist das Subsystem der parallelen Gleichungslösern zu entwickeln, zu implementieren und auf den Testsimulationsaufgaben experimentell zu untersuchen. Als Zielrechnersysteme werden Ressourcen der VPSU-Hardwarestruktur vorgesehen. Die Projektlösungen sollen als unabhängige von Hardwareplattform konzipiert werden und erst bei der Implementierung die Charakteristiken von Ressourc-Komponenten berücksichtigen.

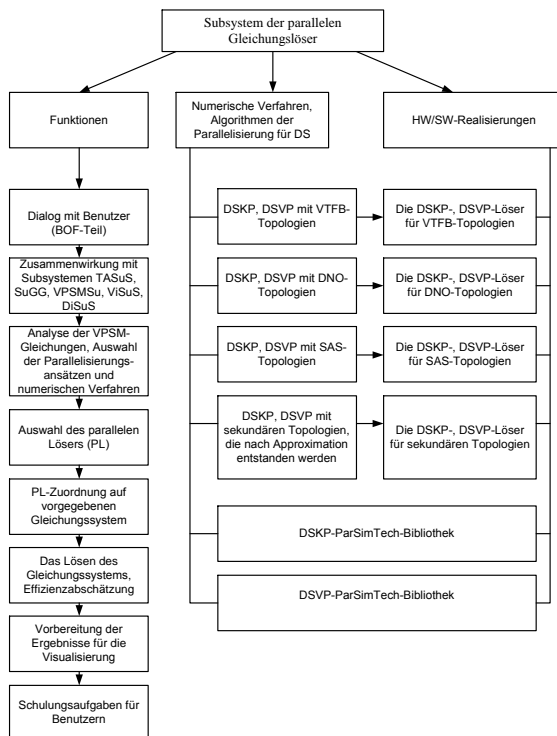


Abbildung 2. Zur Funktionalität und Realisierung des Subsystems

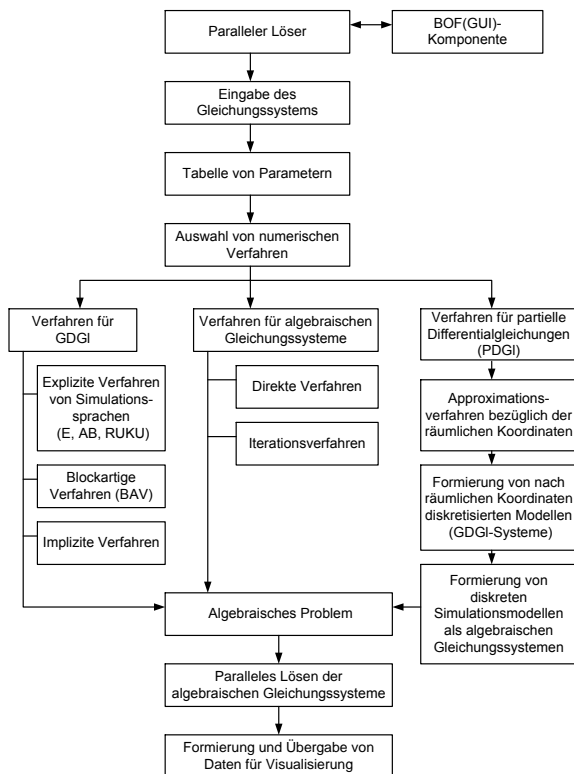


Abbildung 3. Lösensprozess in dem Subsystem des parallelen Gleichungslösers

4.2 Spezifikation der obigen SuGL-Funktionen

Spezifikation der obigen SuGL-Funktionen wird in folgenden Formen erstellt: verbale Spezifikation; grafische Darstellung von Funktionszusammenhängen und Verbindungen mit den anderen Subsystemen; formale Beschreibung der Funktionen.

4.3 Beschreibungen

Beschreibungen: des OO-Ansatzes zur Entwicklung; des Hardware-Teils und des Systemsoftware-Umgebung bezüglich der aktuellen VPSU-Struktur; des Dialogteils der SuGL-Funktionen, die in der Integration mit dem Dialogsystem erfüllt werden sollen.

4.4 Entwicklung und Prüfung des Simulationssoftware-Teils.

Zur Entwicklung und Prüfung des Simulationssoftware gehört das Nächste:

- Entwicklung und Spezifikation der Funktionsalgorithmen
- UML-Diagrammen der Subsystem-Simulationssoftware
- Vereinbarungen mit den anderen Subsystemen
- Vorbereitung der Funktionstestaufgaben für jede Funktion
- Vorbereitung der Subsystem-Testaufgaben für Prüfung der Subsystems-Simulationssoftware
- Implementierung und Prüfung der Funktionsalgorithmen in der gewählten Systemsoftware-Umgebung
- Implementierung und Prüfung der modellierungs- und simulationspezifischen BOF/(GUI)-Funktionsalgorithmen

4.5 Prüfung der Subsystem-Simulationssoftware

Prüfung der Subsystem-Simulationssoftware basiert sich auf der Ausführung von Testaufgaben. Diese Entwicklungskonzeption gilt für die parallelen Gleichungslöser, die in den Simulationsumgebungen verschiedener Gegenstandsgebieten mit den KDS allen möglichen Topologien und formalen Beschreibungen (Abb.1) verwendet werden. Als Modellierungs- und Simulationstest für subsystemübergreifende Prüfung des entwickelten Subsystems (VPSU-TEST) wird die Aufgabe „Parallele Modellierung und Simulation des dynamischen Netzobjektes mit verteilten Parametern (DNOVP)“ vorgesehen. In [3] wird die formale DNOVP-Beschreibung ausführlich betrachtet. Die virtuellen parallelen DNOVP-

Simulationsmodelle werden auf vier Parallelitätsebenen (PE) erstellt. Seitens des Subsystems SuGL soll eine Reihe von Testaufgaben für alle Parallelitätsebenen vorbereitet werden. Für PE-j ($j=1,2,3,4$) sind die Aufgaben ähnlich:

- Prüfung der Subsystemfunktionen „rückwärts“ bis zum ursprünglichen Kodierung des DNO-Graphen für PE-j;
- Prüfung der Topologieinformation, die für Simulation auf PE-j notwendig ist, gemeinsam mit dem Subsystem der Topologieanalyse;
- Prüfung der Anfrage und Eingabe des diskreten Simulationsmodells für PE-j gemeinsam mit dem Subsystem der Gleichungsgenerierung;
- Prüfung der für PE-j notwendigen DNO-Parametern gemeinsam mit dem Datenbank-Subsystem;
- Start der Prozesse des Lösen von PE-j-Gleichungssystemen aus Arbeitsplatzrechner des Modellentwicklers in Zusammenarbeit mit dem Dialogsubsystem;
- Speicherung der Simulationsergebnisse und Prüfung der entsprechenden Funktionen des Visualisierungssubsystems.

5 Implementierungserfahrungen

5.1 Die Gleichungslösern aufgrund der konventionellen numerischen Verfahren

Es wurden die SuGL-Hauptfunktionen anhand der DNOVP-Beispiele des Gegenstandsgebiets „Grubenbewetterungssysteme“ mit den Gleichungslösern aufgrund von Euler(E)-, Adams-Bashfort(AB)- sowie Runge-Kutta(RuKu)-Verfahren im Rahmen der parallelen DNOVP-Simulationsumgebung [1,2,4] realisiert. Nach der Dekomposition und Ortsdiskretisierung des Bewetterungsnetzgraphen mit den $m = 117$ Kanten und $n = 61$ Knoten geben diese Arten von Prozessen insgesamt mehr als 100000 Gleichungen des virtuellen parallelen minimalkörnigen Simulationsmodells auf erster Parallelitätsebene (1 Gleichung – 1 MIMD-Prozeß). Diese Gleichungen werden nach den Angaben des Topologieanalysators mit dem Gleichungsgenerator automatisch erstellt und mit den E-, AB- und RuKu-Lösern auf jeder DNOVP-Parallelitätsebene gelöst.

5.2 Die Gleichungslösern aufgrund der blockartigen numerischen Verfahren (BAV)

Die Implementierungen und Untersuchungen von BAV-Lösern haben wir vom Modell der j -Kante und des kleinen DNOVP ($m=8, n=4$) begonnen. Die dynamischen Prozesse in der j -Kante werden mit den Gleichungen

$$\begin{cases} -\frac{\partial P_j}{\partial \xi} = r_j Q_j^2 + \frac{\rho}{F_j} \frac{\partial Q_j}{\partial t} + r_j(\xi, t) Q_j^2 \\ -\frac{\partial P_j}{\partial t} = \frac{\rho a^2}{F_j} \frac{\partial Q_j}{\partial \xi}, \end{cases} \quad (1)$$

beschrieben. Hier sind: P_j, Q_j – Druck und Luftstrom der Koordinate ξ entlang, die von AKI- bis zum EKK-Knoten errechnet wird; r_j – spezifischer aerodynamischer Widerstand; F_j – die Querschnittsfläche der Kante (Luftwegstrecke); ρ – Luftdichte; a – die Schallgeschwindigkeit im Luft; $r_j(\xi, t)$ – regelbarer Widerstand; ξ_r – die Ortskoordinate des regelbaren Widerstands (z. B., ein Schieber).

Die Randbedingungen für (1) sind die Druckfunktionen P_{AKI}, P_{EKK} in den Knoten der j -Kante. Die Anfangsbedingungen sind $P_j(\xi, 0), Q_j(\xi, 0)$.

Durch Approximation der Gleichungen (1) nach Linienverfahren mit der Ortschrittweite $\Delta \xi$ erhalten wir für k -Element der j -Kante das Gleichungssystem:

$$\begin{aligned} \frac{P_{jk} - P_{j,k+1}}{\Delta \xi_j} &= \frac{\rho}{F_j} \frac{dQ_{jk}}{dt} + r_j Q_{jk} |Q_{jk}| + r_j(\xi_p, t) Q_{jk} |Q_{jk}| \\ \frac{Q_{jk} - Q_{j,k+1}}{\Delta \xi_j} &= \frac{F_j}{\rho a^2} \frac{dP_{j,k+1}}{dt} \end{aligned} \quad (2)$$

Jede Kante wird bei der Approximation nach M_j Elementen Q_{j1}, \dots, Q_{jM_j} zerteilt. Dabei wird M_j abhängig von der Kantenlängen l_j bei der gleichen Ortschrittweite $\Delta \xi$ für ganzes DNOVP als

$$M_j = l_j / \Delta \xi \quad (3)$$

berechnet. Die Nummerierung von Druckwerten als $P_{j1}, P_{j2}, \dots, P_{jM_j+1}$ erfolgt. Für dynamisches Netzobjekt mit verteilten Parametern wird jede Kante nach obige Approximation durch zwei Vektoren Q_j, P_j ($j = 1 \dots m$) präsentiert:

$$Q_j = (Q_{j1}, Q_{j2}, \dots, Q_{jM_j})^T \quad (4)$$

ist Luftstrom in j -Kante,

$$P_j = (P_{j1}, P_{j2}, \dots, P_{jM_j+1})^T \quad (5)$$

ist Druck in j -Kante. Ortsdiskretisiertes Modell der j -Kante beinhaltet M_j Gleichungssysteme (2) bei $k=1,2,\dots,M_j$. Für die numerische Berechnung der Vektorenkomponenten in (4) und (5) werden die Gleichungen (2) zur Form des ortsdiskretisierten Simulationsmodells mit $v=k$ umgewandelt:

$$\begin{cases} \dot{Q}_{jv} = \alpha_j(P_{jv} - P_{j,v+1}) - \beta_j Q_{jv} |Q_{jv}| - \beta_{rj} Q_{jv} |Q_{jv}| \\ \dot{P}_{j,v+1} = g_j(Q_{jv} - Q_{j,v+1}) \end{cases} \quad (6)$$

Die α_j , β_j , β_{rj} , g_j sind die von aerodynamischer j -Kanteparametern abhängigen Koeffizienten.

Wir werden zwischen den Luftstrom(Q)- und Druck(P)-Gleichungen unterscheiden.

Für die weiteren Umformungen führen wir die folgenden Bezeichnungen ein:

$$\begin{cases} \dot{Q}_{jv} = F_{jv}; \\ F_{jv} = \alpha_j(P_{jv} - P_{j,v+1}) - \beta_j Q_{jv} |Q_{jv}| - \beta_{rj} Q_{jv} |Q_{jv}|; \\ \dot{P}_{j,v+1} = D_{jv+1}; \\ D_{jv+1} = g_j(Q_{jv} - Q_{j,v+1}) \end{cases} \quad (7)$$

Bei der Anwendung des blockartigen Lösungsverfahrens werden wir aus (6) und (7) das diskrete Simulationsmodell (BAV-DSM) [3] bekommen. Fürs Einzugs-Vierpunkts-Verfahren sehen die BAV-DSM-Gleichungen bei $\beta_{rj}=0$ so aus [3]:

$$\begin{aligned} Q_{jvn,1} &= \frac{\tau}{720}(251F_{jvn,0} + 646F_{jvn,1} - 264F_{jvn,2} + 106F_{jvn,3} - 19F_{jvn,4}) + Q_{jvn,0}; \\ Q_{jvn,2} &= \frac{\tau}{90}(29F_{jvn,0} + 124F_{jvn,1} + 24F_{jvn,2} + 4F_{jvn,3} - F_{jvn,4}) + Q_{jvn,0}; \\ Q_{jvn,3} &= \frac{3\tau}{80}(9F_{jvn,0} + 34F_{jvn,1} + 24F_{jvn,2} + 14F_{jvn,3} - F_{jvn,4}) + Q_{jvn,0}; \\ Q_{jvn,4} &= \frac{2\tau}{45}(7F_{jvn,0} + 32F_{jvn,1} + 12F_{jvn,2} + 32F_{jvn,3} + 7F_{jvn,4}) + Q_{jvn,0}; \\ F_{jvn,0} &= \alpha_j(P_{jv,0} - P_{jv+1,n,0}) - \beta_j Q_{jv,n,0} |Q_{jv,n,0}|; \\ F_{jvn,1} &= \alpha_j(P_{jv,0} - P_{jv+1,n,1}) - \beta_j Q_{jv,n,1} |Q_{jv,n,1}|; \\ F_{jvn,2} &= \alpha_j(P_{jv,0} - P_{jv+1,n,2}) - \beta_j Q_{jv,n,2} |Q_{jv,n,2}|; \\ F_{jvn,3} &= \alpha_j(P_{jv,0} - P_{jv+1,n,3}) - \beta_j Q_{jv,n,3} |Q_{jv,n,3}|; \\ F_{jvn,4} &= \alpha_j(P_{jv,0} - P_{jv+1,n,4}) - \beta_j Q_{jv,n,4} |Q_{jv,n,4}|. \end{aligned} \quad (8)$$

$$\begin{aligned} P_{jv+1,n,1} &= \frac{\tau}{720}(251D_{jv+1,n,0} + 646D_{jv+1,n,1} - 264D_{jv+1,n,2} + 106D_{jv+1,n,3} - 19D_{jv+1,n,4}) + P_{jv+1,n,0}; \\ P_{jv+1,n,2} &= \frac{\tau}{90}(29D_{jv+1,n,0} + 124D_{jv+1,n,1} + 24D_{jv+1,n,2} + 4D_{jv+1,n,3} - D_{jv+1,n,4}) + P_{jv+1,n,0}; \\ P_{jv+1,n,3} &= \frac{3\tau}{80}(9D_{jv+1,n,0} + 34D_{jv+1,n,1} + 24D_{jv+1,n,2} + 14D_{jv+1,n,3} - D_{jv+1,n,4}) + P_{jv+1,n,0}; \\ P_{jv+1,n,4} &= \frac{2\tau}{45}(7D_{jv+1,n,0} + 32D_{jv+1,n,1} + 12D_{jv+1,n,2} + 32D_{jv+1,n,3} + 7D_{jv+1,n,4}) + P_{jv+1,n,0}. \\ D_{jv+1,n,0} &= g_j(Q_{jv,n,0} - Q_{jv+1,n,0}); \\ D_{jv+1,n,1} &= g_j(Q_{jv,n,1} - Q_{jv+1,n,1}); \\ D_{jv+1,n,2} &= g_j(Q_{jv,n,2} - Q_{jv+1,n,2}); \\ D_{jv+1,n,3} &= g_j(Q_{jv,n,3} - Q_{jv+1,n,3}); \\ D_{jv+1,n,4} &= g_j(Q_{jv,n,4} - Q_{jv+1,n,4}). \end{aligned} \quad (9)$$

Die Werte $Q_{vn,0}$ und $P_{v+1,n,0}$ im Nullpunkt des Blocks n werden als die Werte im letzten Punkt des vorigen Blocks für $n > 1$ ($t_{n-1,k} = t_{n,0}$) definiert. Für jede Kante des Testnetzobjektes (Abb.4) werden die Gleichungssysteme (8), (9) mit den entsprechenden Indizes $j = 1, 2, \dots, 8$ und $v = 1, 2, \dots, M_j$ generiert.

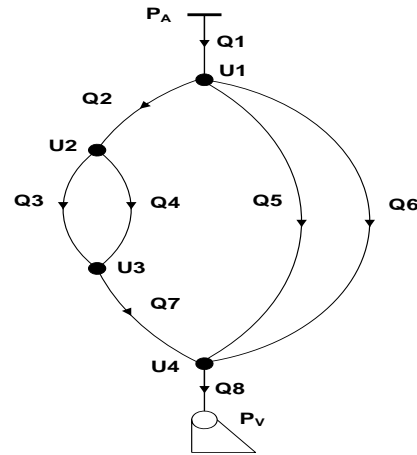


Abbildung 4. Graph des Testnetzobjektes mit 8 Kanten und 4 Knoten

Die Randbedingungen $P_{U1} \dots P_{U4}$ werden mit den Knotengleichungen

$$\begin{aligned} dP_{u1} / dt &= \gamma_{u1}(Q_{1M1} - Q_{21} - Q_{51} - Q_{61}), \\ dP_{u2} / dt &= \gamma_{u2}(Q_{2M2} - Q_{31} - Q_{41}), \\ dP_{u3} / dt &= \gamma_{u3}(Q_{3M3} + Q_{4M4} - Q_{71}), \\ dP_{u4} / dt &= \gamma_{u4}(Q_{7M7} - Q_{81} + Q_{5M5} + Q_{6M6}), \end{aligned} \quad (10)$$

beschrieben. Entsprechend dem System (9) wird jede Knotengleichung in BAV-Form umgewandelt und ihre gesuchte Variable P_{Uw} ($w=1,2,3,4$) wird im gesamten Rechenprozess berechnet.

Die Parameter des Testnetzobjektes (Abb.4) sind für Ortschrittweite $\Delta\xi=40\text{m}$ in Tabelle 1 zusammengefasst.

Ka\Par	l_j	E_i	t_i	$\Delta x, \text{m}$	M_i	α	β	γ	P_A	P_V	Q_{S7}
Q1	800	6.5	0.00048	40	20	0.13	0.00252	523	0.0		71.6
Q2	150	6.5	0.00048	40	4	0.13	0.00252	523			28.9
Q7	150	6.5	0.00048	40	4	0.13	0.00252	523			28.9
Q8	650	6.5	0.00048	40	16	0.13	0.00252	523		-4100	71.6
Q3	1925	4.67	0.00105	40	48	0.093	0.004	728			14.4
Q4	1410	5.98	0.0019	40	35	0.12	0.009	569			14.9
Q5	1720	4.5	0.00112	40	43	0.09	0.004	756			19.2
Q6	934	4.97	0.00137	40	23	0.1	0.0055	684			23.7

Tabelle 1. Parameter des Testobjektes

Es wurden die sequentiellen Gleichungslösern nach Blockdiagramm Abb.5 implementiert, um die Arbeitsfähigkeit der blockartigen Verfahren zu prüfen und die erreichten Simulationsergebnisse mit den Ergebnissen der konventionellen Verfahren zu vergleichen. Hier sind BAV2 und BAV4 entsprechend die Einschrittsverfahren mit den $k=2$ und $k=4$ Berechnungspunkten im jeweiligen Block.

In Abb. 6 – 9 sind die dynamische Prozesse in der Kante Q8 dargestellt, die nach den schnellen Änderungen der Randbedingung von $P_V=0$ bis zum $P_V = -4100$ (Tab.1) entstehen. In den allen anderen Kanten die Prozesse sehen ähnlich aus. Analyse der Simulationsergebnissen lässt uns folgendes feststellen:

- BAV4 zeigt richtig die Zeitverzögerungen in verschiedenen Kantenelementen, mit hoher Genauigkeit wird Q8-Testwert im stationären Zustand erreicht (Abb.6); ähnliche Prozesse laufen in allen anderen Kanten;
- Alle Verfahren bei $\tau=0.0001$ liefern die physikalisch korrekte Q- und P-Prozesse, es werden die richtigen Testwerte im stationären Zuständen erreicht (Abb. 7);
- Die Empfindlichkeit der Verfahren zur Vergrößerung der Zeitschrittweite zeigen Abb.8 und Abb.9: bei $\tau=0.01$ bleiben nur BAV2, BAV4 und AB-Verfahren stabil und zeigen fast dieselbe Prozesse; beide blockartige Verfahren erlauben noch größeren Schrittweiten (Abb.9).

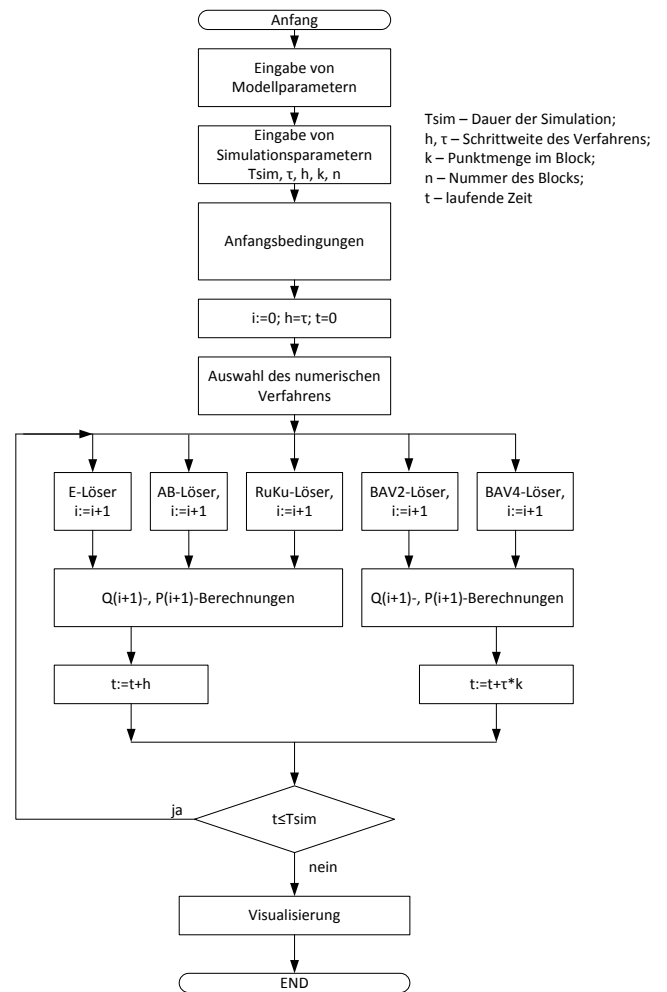


Abbildung 5. Blockdiagramm des sequentiellen Gleichungslösers

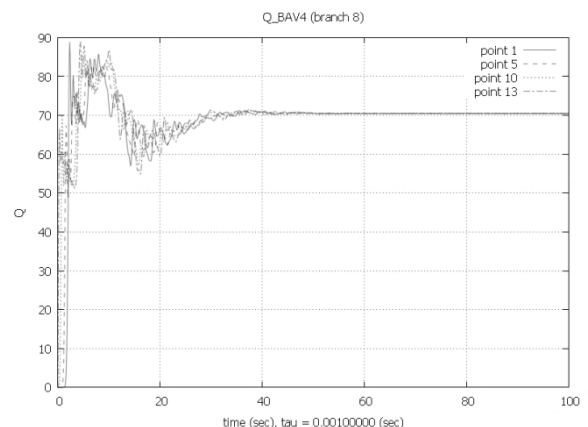


Abbildung 6. Dynamische Prozesse Q8 in den Kantelementen 1, 5, 10, 13 nach BAV4

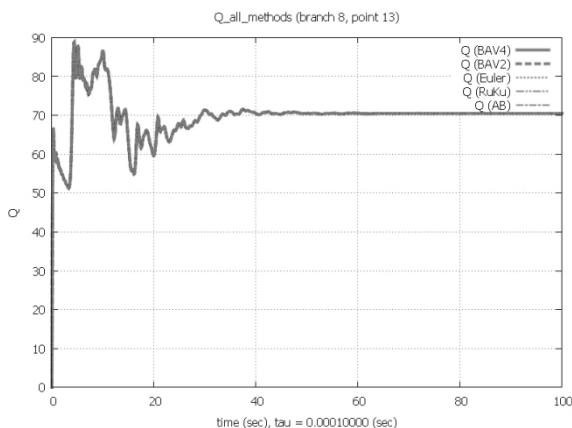


Abbildung 7. Dynamische Prozesse Q8, Kantenelement 13, alle Verfahren mit $\tau = 0.0001$

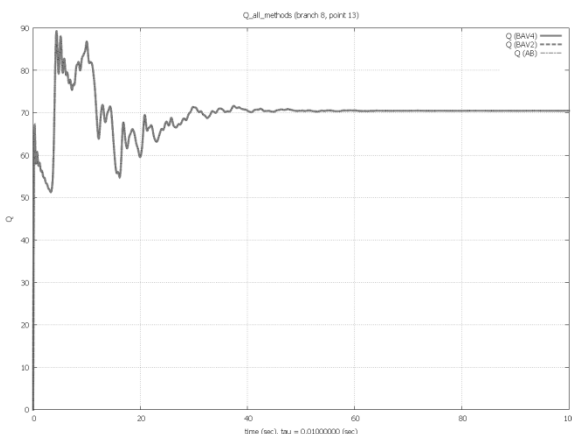


Abbildung 8. Vergleich von BAV und des AB-Verfahrens mit $\tau = 0.01$

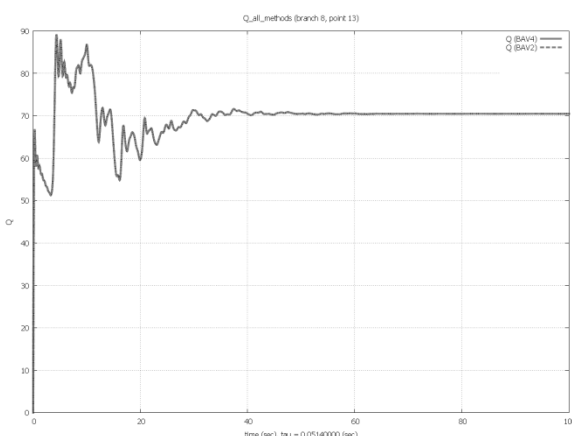


Abbildung 9. Vergleich von BAV2 und BAV4; andere Verfahren sind mit dem $\tau = 0.05$ nicht arbeitsfähig

6 Zusammenfassung und Ausblick

Die vorgestellten Ergebnisse liefern einen Beitrag zur Entwicklung des Subsystems von Gleichungslösern

(SuGL) der verteilten parallelen Simulationsumgebung (VPSU). Subsystem erfüllt wichtigste Funktionen der VPSU, nämlich effizientes paralleles Lösen der Gleichungssysteme, die in der Form von Simulationsmodellen dargestellt werden. Die Implementierungserfahrung der rechnergestützten Erstellung von Modellen, Simulationsmodellen und Gleichungslösern für dynamischen Netzobjekte mit verteilten Parametern (DNOVP) hat gezeigt, dass SuGL gute Perspektive der Anwendung von blockartiger paralleler numerischer Verfahren hat. Diese Verfahren haben höhere Genauigkeit und Konvergenz, besseres Verhältnis „Genauigkeit-Geschwindigkeit“ im Vergleich mit traditionellen Verfahren. Die weiteren Untersuchungen von BAV sind mit den theoretischen Analyse der Parallelisierung von Gleichungslösern, Implementierung und experimentellen Prüfung von BAV-Lösern, Anwendung von BAV in SuGL, Integration mit den anderen Subsystemen der verteilten parallelen Simulationsumgebung verbunden.

7 Literatur

- [1] Feldmann L.P., Resch M., Svjatnyj V.A., Zeitz M.: Forschungsgebiet: parallele Simulationstechnik. In: DonNTU, FRTI-Werke, Reihe "Probleme der Simulationstechnik und rechnergestützten Projektierung von dynamischen Systemen", Band 9(150). – Donezk, 2008, S. 9-36. Auch in: ASIM'09, Tagungsband Vortragskurzfassungen, Cottbus, 2009.
- [2] Svjatnyj V., Kushnarenko V., Shcherbakov O., Resch M.: Dekomposition einer verteilten parallelen Simulationsumgebung. In: DonNTU-Werke, Reihe "Probleme der Simulationstechnik und rechnergestützten Projektierung", №1(10)-2(11). – Donezk 2012, 227-234.
- [3] Svjatnyj V.A., Kushnarenko V.G.: Virtuelle Simulationsmodelle und ein Devirtualisierungsvorgang der Erstellung von parallelen Simulatoren für dynamische Netzobjekte mit verteilten Parametern. ASIM-Konferenz STS/GMMS 2012, Wolfenbüttel 2012. ASIM Mitteilung AM 140, 49-56, ISBN 978-3-901608-39-1.
- [4] Moldovanova O.V. Problemorientierte parallele Simulationsumgebung. / O.V. Moldovanova, V.A. Svjatnyj, L. Feldmann, M. Resch, U. Küster // In: DonNTU-Werke, Reihe "Informatik, Kybernetik und Rechentechnik", №93. – Donezk, 2005. – S. 145–150.

Model-based parallelization of discrete traffic simulation models

Oliver Ullrich¹, Daniel Lückerrath², Ewald Speckenmeyer^z

¹National Science Foundation's Industry-University Cooperative Research Center,
School of Computing and Information Sciences, Florida International University,
ECS 243C, 11200 SW 8th St, Miami FL-33199

²Institut für Informatik, Universität zu Köln, Albertus-Magnus-Platz, 50923 Köln, Germany
oullrich@fiu.edu

To re-establish regular operations in a tram traffic network after a large disturbance, e.g. resulting from vehicle breakdown or station closure, the viability of several rescheduling and rerouting strategies has to be evaluated prior to their implementation. Here, a multi-modal traffic simulation system can help to enhance the decision quality. Such a system obviously faces tight time constraints, so simulation data has to be acquired fast.

In this paper we propose a method for the parallel execution of discrete traffic simulation models, which would accelerate data generation in comparison to a sequential model. To assess this method's dynamic behavior in real-world applications, some experiments conducted on a software system modeling schedule based tram traffic are presented.

After giving an introduction to the scope and aim, we show some background on the parallelization of discrete simulation models. The main part of the paper begins with the proposal of a method to parallelize the execution of simulation models with problem specific properties. Some estimations of the method's efficiency are shared, followed by several experiments to highlight its dynamic behavior in real-world applications. The paper ends with a short summary and some thoughts on further research.

1 Introduction

When severe disturbances occur in tram networks, e.g. originating from broken down trams, closed stations, or other blocked resources, traffic operators have to apply rescheduling and rerouting strategies (see [10] and [12]) to reestablish regular operations. To be effective, these strategies are inevitably multi-modal: trams are rescheduled to compensate for cancellations, regional and local buses are rerouted to relieve the tram network, some transit operators even co-operate with taxi companies (see [26]). To evaluate the applicability of a given rescheduling or rerouting strategy prior to its implementation in the real-world system, a multi-modal simulation software is needed. Operators obviously face tight time constraints for their decisions, so simulation data has to be acquired fast.

In this paper we propose a method for the parallel execution of discrete traffic simulation models. We do not aim for a general approach, which would be equally well applicable for all discrete models, but for a method that utilizes some specific properties of a subclass of models, including traffic simulation models. The traffic planners' laptop or desktop computers constitute the target platform of the resulting simulation tools; the method thus should utilize their capacity for small scale parallel processing. To employ the

available resources effectively, the method applies a dynamic and adaptive load balancing scheme. From the model's point of view, the mechanics of parallelization and load balancing are transparent, so that these internals can be changed without compromising the subsequent use of the model. A sequential model of tram traffic is already in place (see [11]); the accompanying representation of bus traffic is only partly implemented yet (see [24]).

This paper continues with a presentation of some background on general methods of parallel simulation and their customization in practical applications (section 2). We then present an approach on parallel execution of simulation models that share some characteristics like being spatially explicit, having mostly local dependencies, and having computational load generated by transient entities moving through a network (section 3), followed by some efficiency estimations (section 4). Based on an implementation of the approach, some experiments are conducted, focusing on the parallel computation of artificial loads, and the parallel simulation of tram traffic (section 5). The paper closes with a short summary of the lessons learned and some thoughts on future work (section 6).

2 Background

A discrete simulation model comprises of entities which communicate with each other via simulation events or messages, and which change their states at discrete points in model time. The main task when parallelizing a discrete simulation model is to avoid the occurrence of parallelization artifacts, i.e. to assure that parallel execution yields the same results as sequential execution. This is not a trivial task because a model usually contains both spatial dependencies (which are often local, i.e. between neighboring entities) and temporal dependencies (which are also often local, between close points in model time).

Several approaches to the parallelization of discrete simulation models are known: The simplest approach is the concurrent execution of several sequential experiments (see [15]). Outsourcing of maintenance functions (e.g. random number generation or database access) to secondary processors is also relatively simple but usually does not scale very well. A special case of this is the parallel administration of central data structures, usually applied to the future event list (for an overview of suitable data structures, see [18]). Time based parallelization techniques, i.e. computing different simulation time intervals in parallel, are scaling very well in principal but are only suited for specific models (a few applications are known, as described in [6] or [7]).

A well examined, and also well scaling technique is model based (also called spatial) parallelization, which can be found in both literature (see [5], pp. 39) and real-world applications (e.g. see [1], [13] or [19]).

2.1 Model based parallelization

Model based parallelization utilizes the model's inherent parallelism, i.e. that many state changes can be executed independently from each other. To achieve this, the model is decomposed into partial models, which are assigned to the involved processors. With this method, entities of different partial models communicate via messages sent over the network or a common cache memory.

Obviously, careful synchronization of the model's execution is necessary. The local causality constraint (see [5], pp. 52) demands that each entity executes its concerning simulation events in a non-decreasing order regarding their scheduled time of occurrence. Non-adherence to this constraint may result in causality errors which invalidate the simulation results.

Two categories of synchronization methods are known: Conservative synchronization methods prevent the out-of-order execution of simulation events by technical measures, thus guaranteeing adherence to the local causality constraint. Important conservative synchronization mechanisms include synchronization via null messages (described in [2] and [3]),

deadlock detection and recovery (described in [4]), and synchronized execution (see [17] and [21]). Optimistic synchronization methods execute the partial models as fast as possible, and thereby allow violations of the local causality constraint. The methods detect and subsequently repair these violations by rejecting and re-computing the invalid regions of the simulation run. The best known optimistic approach is the aptly named "time warp", first described in [8].

2.2 Load balancing for parallel discrete simulation systems

Resulting from the typical dependencies in simulation models, the simulation's execution speed is generally dependent on the processor which advances slowest in simulation time. It is therefore necessary to incorporate a load balancing system into the simulation engine. This system does not aim at high utilization of the processors' capacity alone, but also has to consider an uniform advance in simulation time. Discrete simulation systems therefore often apply special load balancing schemes.

Those methods can be characterized as dynamic, static, adaptive, non-adaptive, local, centered, or hierarchical (see [14], pp. 12): A *dynamic* load balancing method continuously considers imbalances which develop from shifts in the model's computational load, and re-assigns partial models to appropriate processors while executing the simulation run. In contrast, *static* methods estimate the load and assign partial models to processors in a preprocessing step before the start of the simulation run, and thus don't consider dynamic changes in the model's activity. *Adaptive* methods consider fluctuations in the available processor power originating from the demand of dynamic processes belonging to third parties. In inhomogeneous computer networks adaptive methods also consider the dissimilar performance power of the respective processors. A *non-adaptive* approach ignores those fluctuations. In *local* methods, the processors only exchange data within limited neighborhoods and act on this local information alone, while *centered* methods utilize a marked controller process to which the other processors report. *Hierarchical* methods usually organize communication in a tree topology.

A load balancing method viable for traffic simulation models on a PC or laptop computer should be both dynamic and adaptive, and thus has to consider both the varying computational load of the model, and the changing availability of resources on a non-exclusively used machine. Such a dynamic and adaptive method requires a load measure which considers both the available processing power and the engaged processors' advance in simulation time. A centered method is usually easier to implement and quite ade-

quate for a PC system with only eight to sixteen processor cores; if a method is targeted at a massively parallel system it should avoid a potential bottleneck by utilizing a hierarchical or local scheme.

3 An approach to the parallelization of discrete traffic simulation models

To be viable for the proposed method, a model has to comply to some prerequisites: It has to be spatially explicit and representable as a sparsely populated graph; dependencies have to be typically local, so that neighborhood relationships can therefore be exploited; the model's activity, and thus the computational load, has to be produced by transient entities which steadily move through the model, so the load typically shifts slowly and is caused by several simulation steps. Many traffic simulation models comply with these prerequisites (see e.g. [11], [16], and [24]). The proposed parallelization method is flexible regarding simulation paradigms: the implemented models can be event based, process based, agent based, or be based on the activity scanning approach.

The method builds upon three layers of abstraction (see figure 1): On the *computer network layer* processors and processor cores form a star-shaped graph, connected by a local area network or a shared cache memory; the *partial models* are assigned to these processors, and connected by the communication occurring during the simulation run; the *model graph* builds up on that and consists of model nodes, which represent entities, and their connecting edges. Transient entities (here represented by tokens) map the dynamically changing model activity; they move through the model graph along its edges.

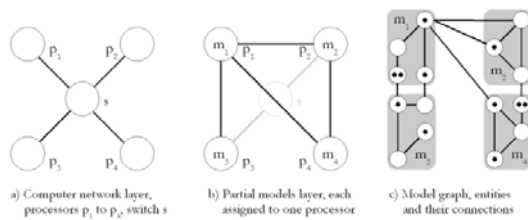


Figure 1. Three layers of abstraction

Before the start of a simulation run, the model is partitioned, and the resulting partial models are assigned to the participating processors (see figure 2). A heuristic method (see [9]) is applied in this static load balancing step to reduce the number of edges between partitions, and thus to reduce the communication load.

During the simulation run, tokens move from node to node and thus generate load imbalances, which have to be handled dynamically (see figure 3).

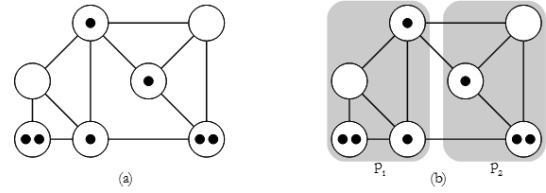


Figure 2. Static load balancing

To accomplish this, a processor p_i which is over-charged, but has a neighbor p_j which is not fully loaded, selects a number of model nodes and shifts them over to p_j (see figure 4). This adjustment is done iteratively in the course of several simulation steps, until a stable state is reached.

The method uses a dynamically calculated load measure, considering both the changes in the model's activity and the time elapsing while computing, and therefore is dynamic and adaptive. It also exploits knowledge of regional dependencies to keep down communication load; existing neighborhood relations are not affected by the mechanism. Load balancing is carried out when all processors have entered the synchronization barrier (see [5], pp. 65-96), and are thus done with processing step t , but did not yet start with processing step $t+1$. The method prefers to shift model nodes from a slow processor p_s to a fast processor p_f in a way that iteratively further reduces the communication load during the simulation run. To accomplish this, it classifies each model node v in the set V_{p_s} of all nodes hosted by processor p_s in one of four priority classes:

4. All $v_i \in V_{p_s}$;
3. each node v_i in 4 which has an edge to a node $v_j \in V_{p_f}$, with any $p \neq p_s$;
2. each node v_i in 3 which has an edge to a node $v_j \in V_{p_f}$, which is hosted by a processor p_f which is not operating at full capacity; and
1. each node v_i in 2 with a greater number of edges to nodes hosted by processor p_f than to nodes administrated by p_s .

The method prefers to shift nodes from class 1, followed by class 2 and 3. Nodes which are only members of class 4 are not shifted.

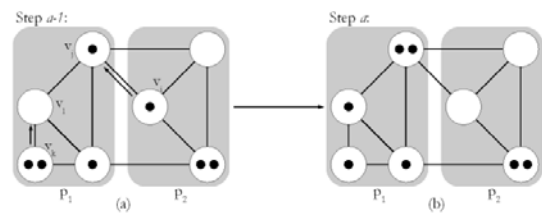


Figure 3. Dynamic load shifting

Each simulation step consists of three phases: The *computation of the model*, the duration of which is dependent on the partial model's activity, and which the load balancing method tries to distribute equally over all processors; the *synchronization* phase, in which processors already finished with computing wait for the others to finish their tasks; and the *communication and load balancing* phase, in which the method shifts model nodes from fully loaded processors to underloaded ones. The load balancing step itself consists of three phases (see figure 5): load measurement, load assessment, and load shifting. For a more detailed description of the method see [22], pp. 61-76.

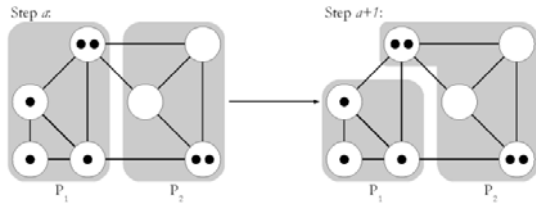


Figure 4. Dynamic load balancing

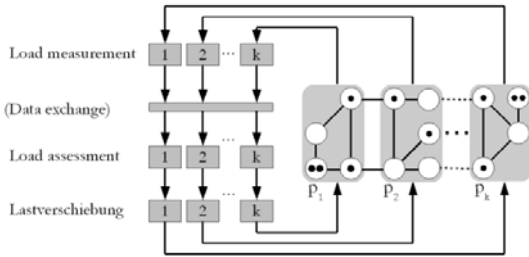


Figure 5. Phases of the load balancing process

4 Scalability and efficiency

As described, each simulation step i consists of three phases: computation of the model, synchronization, and communication, whose computational complexity for each processor p is denoted by $t_m(p, i)$, $t_s(p, i)$, and $t_c(p, i)$ (see figure 6). These values can be estimated, which in turn yields estimations for the scalability and efficiency of the proposed method.

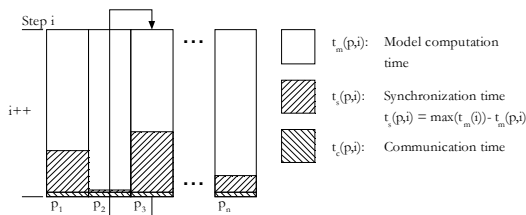


Figure 6. Phases of the simulation step

The combined complexity $t_g(i)$ of each simulation step i in a system with k processors for a bad case (load balancing mechanism is switched off and computational loads for each processor p in each step i are drawn randomly, for a detailed description see [22], pp. 76-84) is shown in equation 1:

$$t_g(i) = \underbrace{c_m * (1 - \frac{1}{k+1})}_{t_m(p,i) + t_s(p,i)} + \underbrace{6 * (k - 1) + c_n * \frac{k+1}{k^2}}_{t_c(p,i)} \quad (1)$$

Here, c_m denotes the computational load generated by the model, and c_n denotes the number of transient entities to be moved between model nodes.

The average time complexity $t_g(i)$ of a single step i (load balancing mechanism is switched on, computational loads are shifting smoothly through the model) is shown in equation 2:

$$t_g(i) = \frac{c_m}{k} + \underbrace{6 * (k - 1) + c_n * \frac{k+1}{k^2} + 2 * c_l * (k - 1)}_{t_c(p,i)} \quad (2)$$

Here, c_l denotes the number of resident entities which have to be transferred in the context of load balancing.

The scalability of the method is thus mainly dependent on the values c_m and c_n , and in the average case also on c_l . These values are all properties of the model, and are thus not influenced by the method itself. In the average case, with the partial model's loads shifting in a benign way, the method shows linear scaling. The manifesting scaling factor for an individual model is directly dependent on the ratio of its computation load $t_m(p, i)$ to its communication load $t_c(p, i)$. Or, to put it simple: Bigger models scale better.

The expected efficiency of the method for a model with unfavorable arbitrary load imbalances can be shown to be always greater than 0.5 (see equation 3).

$$E(t_{busy}(k)) > E(t_{idle}(k)) \quad \forall k > 1 \quad (3)$$

For a detailed analysis of computational complexity, scalability and efficiency, see [22], pp. 76-86.

5 Experiments

The proposed method was implemented as a C++ framework (described in [22], chapter 4), and is utilized in two different scenarios. To keep influences of a complex real-world model with often irregular properties at a minimum, the framework is first applied to the computation of artificial loads moving through a randomly generated graph. This is followed by a real-world application in the simulation of time-table based tram traffic.

Experiments are run using notebook computers with 6 gigabytes memory and an Intel Core i7-740QM processor with four cores running at 1.73 gigahertz. The turbo boost technology for accelerating single thread applications is turned off for the experiments. For experiments with more than four processor cores, several notebooks are connected by a 100 megabit ethernet switch.

5.1 Computation of artificial loads

The first set of experiments is conducted on the parallel computation of artificial load generated by token movements on a randomly generated graph. Tokens stay at a node for a certain number of simulation steps, and then move over an edge to a randomly picked neighboring node. During each simulation step, these tokens generate computational load. This is generated by executing the Sieve of Eratosthenes algorithm (see [20], pg. 85) to identify prime numbers up to an upper bound. This upper bound q_{max} is set to the sum of the base load of the node v_i and the token's weights: $q_{max} = l_{base} + l_{token} * |T_i|$.

To generate the graph, n nodes with a base load of $l_{base} = 10.000$ are generated. For each node v , two nodes $u_1 \neq v$ and $u_2 \neq u_1 \neq v$ not yet connected to v are chosen randomly. Then, two edges $v \leftrightarrow u_1$ and $v \leftrightarrow u_2$ are added to connect v to those nodes. A token is generated for every fifth node v_i with $i \bmod 5 = 0$. Each of those tokens has a weight of $l_{token} = 10.000$, a maximum retention period of $t_{max} = 100$ simulation steps, and a current retention time t_i drawn from a uniform distribution between 0 and t_{max} . This value t_i is decreased by one during each simulation step. When it reaches zero, an edge outgoing from its current host node is selected randomly; the token is then moved over this edge and is re-initialized by its new host with $t_i = t_{max}$. In the conducted experiments, each simulation run consists of 500 steps, the load balancing scheme is active.

We begin by running a mid size instance of $n = 400$ nodes on up to eight processor cores. Average runtime, speedup values, and marginal utility are shown in table 1 and figure 7. A second series of asymptotic experiments begins with a graph consisting of $n=100$ nodes and 200 edges on a single processor core, going up to $n=800$ nodes on eight processors. Average runtime, scaling factor and marginal utility are shown in table 2 and figure 8.

5.2 Simulation of time-table based tram traffic

The proposed method was then utilized to parallelize a sequential simulation engine of time-table based tram traffic (described in [11]). The resulting software tools were applied to the KVB network of Cologne, Germany (see [23]), and the TAM Tramway network of Montpellier, France (see [25]).

#Proc.	Runtime (sec)	Speedup	Marg. utility
1	2,010.0	1.00	1.00
2	1,010.3	1.99	0.99
3	682.9	2.95	0.96
4	516.5	3.89	0.94
5	421.0	4.77	0.88
6	372.1	5.40	0.63
7	340.9	5.90	0.49
8	325.8	6.17	0.27

Table 1. Runtime and speedup for the computation of artificial loads

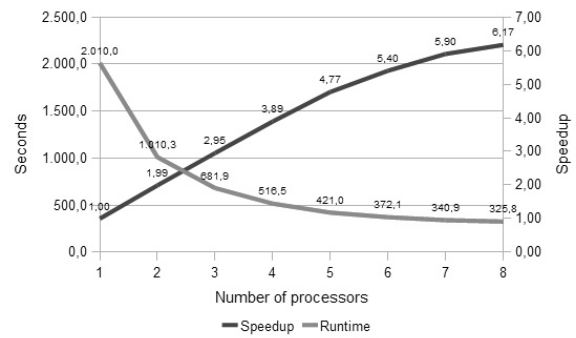


Figure 7. Runtime and speedup for the computation of artificial loads

#Proc.	Graph		Scalability		
	V	E	Runtime (sec)	Scaling factor	Marg. utility
1	100	200	495.8	1.00	1.00
2	200	400	499.1	1.99	0.99
3	300	600	523.4	2.84	0.86
4	400	800	521.4	3.80	0.96
5	500	1,000	521.7	4.75	0.95
6	600	1,200	550.5	5.40	0.65
7	700	1,400	551.8	6.29	0.89
8	800	1,600	568.4	6.98	0.69

Table 2. Runtime and scaling factor for the computation of artificial loads

The described experiments are conducted on a model of Cologne's network. It consists of 528 platforms and 58 track switches connected via 584 tracks. These tracks cover a total length of 407.4 kilometers, resulting in an average track length of 697.6 meters. 15 lines with 182 line routes are served by 178 vehicles which execute 2,814 trips per operational day. The simulation engine was run on up to eight processors of the described type, under the parameter set described in section 5.1. Average runtime, speedup and marginal utility are shown in table 3 and figure 9.

5.3 Results and discussion

The simulation models are executed 10 times per measuring point.

For the computation of artificial loads (see figure 7 and table 1), the method yields a high gain in speedup for up to five processors (speedup 4.72), which flattens when more processors are added. With partial models getting smaller, the ratio of synchronization and communication time to model computation time rises, so efficiency is declining. For the asymptotic experiments (see figure 8 and table 2) a linear regression yields a function $s(k)=0.86*k+0.27$ for scaling, and $T(k)=10.16*k+483.3$ for run time. Under the described conditions the method thus shows a linear scalability with a scaling factor of around $0.86*k$.

The simulation of time-table based tram traffic shows mixed results (see figure 9 and table 3): For up to four processor cores - based on a single parallel computer - the speedup rises to 2.83, and then caves in to 1.89 when the fifth processor - connected via LAN - is added. The reason for this behavior is the significantly higher communication cost between LAN connected computers in relation to communication between parallel processor cores. The ratio of high communication cost to a relatively low computation cost for the distributed partial models forbids an effective execution on LAN connected computers. A linear regression for the first four measuring points yield a function of $z(k)=0.62*k+0.5$ for speedup, and $T(k)=-55.03*k+289.5$ for run time. The last four points yield functions of $z(k)=0.12*k+1.3$ and $T(k)=-7.71*k+177.3$. The model instance is therefore large enough to be efficiently run on a parallel computer, but too small to be executed expediently on LAN connected computers.

For a more in-depth discussion of the experiments see [22], chapters 5 and 6.

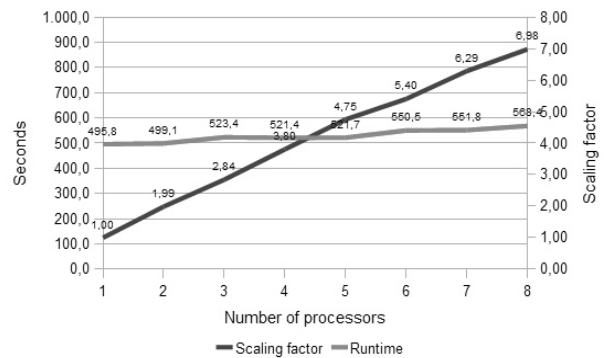


Figure 8. Runtime and scaling factor for the computation of artificial loads

#Proc.	Runtime (sec)	Speedup	Marg. utility
1	263.8	1.00	1.00
2	144.5	1.82	0.82
3	106.4	2.44	0.62
4	93.1	2.83	0.39
5	139.8	1.89	-0.94
6	130.6	2.02	0.13
7	121.0	2.12	0.10
8	117.3	2.25	0.13

Table 3. Runtime and speedup for the simulation of time-table based tram traffic

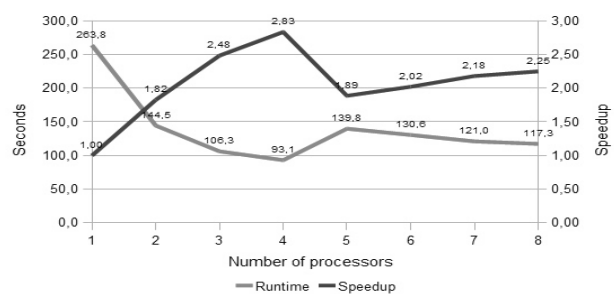


Figure 9. Runtime and speedup for the simulation of time-table based tram traffic

6 Summary and further research

We presented an approach to the parallel execution of traffic simulation models, which includes a dynamic and adaptive load balancing scheme. Some thoughts on scalability and efficiency were shared: even under adverse circumstances the efficiency does not get lower than 0.5. Experiments conducted on the computation of artificial loads yield a speedup of 3.89 on four processors and 6.17 on eight processors. Parallel execution of a tram traffic model shows a speedup of 2.83 on four processor cores. A higher speedup might be reached when executing a larger model.

In a next step, the multi-modal model will be extended by a representation of bus transit.

7 Acknowledgements

This material is partially based upon work supported by the National Science Foundation under grants I/UCRC IIP-1338922, AIR IIP-1237818, SBIR IIP-1330943, III-Large IIS-1213026, MRI CNS-0821345, MRI CNS-1126619, CREST HRD-0833093, I/UCRC IIP-0829576, MRI CNS-0959985, FRP IIP-1230661, and U.S Department of Transportation under a 2013 TIGER grant.

8 References

- [1] Avril, H., Tropper, C.: *The Dynamic Load Balancing of Clustered Time Warp for Logic Simulation*. In: Proceedings of the tenth workshop on Parallel and distributed simulation, pp. 20-27, 1996.
- [2] Bryant, R.E.: *Simulation of Packet Communication Architecture Computer Systems*. Computer Science Laboratory. Technical Report, Cambridge, Massachusetts, Massachusetts Institute of Technology, 1977.
- [3] Chandy, K.M., Misra, J.: *Distributed Simulation: A Case Study in Design and Verification of Distributed Programs*. In: IEEE Transactions on Software Engineering, SE-5(5), pp. 440-452. 8, pp. 250-255, 1965.
- [4] Chandy, K. M., Misra, J.: *Asynchronous distributed simulation via a sequence of parallel computations*. In: Communications of the ACM, Vol. 24, No. 4, pp. 198-205, 1981.
- [5] Fujimoto, R. M.: *Parallel and Distributed Simulation*. New York: John Wiley & Sons, 2000.
- [6] Greenberg, A. G., et al.: *Algorithms for Unboundedly Parallel Simulations*. In: ACM Transactions on Computer Systems, Vol. 9, No. 3, pp. 201-221, 1991.
- [7] Heidelberger P., Stone, H.: *Parallel Trace-Driven Cache Simulation by Time Partitioning*. In: Proceedings of the 1990 Winter Simulation Conference, pp. 734-737, 1990.
- [8] Jefferson, D.R.: *Virtual Time*. In: ACM Transactions on Programming Languages and Systems, Vol. 7, No. 3, pp. 404-425, 1985.
- [9] Kernighan, B. W., Lin, S.: *An Efficient Heuristic Procedure for Partitioning Graphs*. Bell Syst. Tech Journal, Volume 49, Number 2, pp. 291-307, 1970.
- [10] Lückerrath, D.: *Thoughts on restoration of regular tram operation*. In: Proceedings of Sommertreffen Verkehrssimulation 2012, AM 143, ARGESIM/ASIM Pub., TU Vienna, pp. 4-6, 2012.
- [11] Lückerrath, D., Ullrich, O., Speckenmeyer, E.: *Modeling time table based tram traffic*. In: Simulation Notes Europe (SNE), ARGESIM/ASIM Pub., TU Vienna, Volume 22, Number 2, pp. 61-68, 2012.
- [12] Lückerrath, D., Ullrich, O., Speckenmeyer, E.: *Applicability of rescheduling strategies in tram networks*. In: Proceedings of ASIM-Workshop STS/GMMS 2013 - ARGESIM Report 41, AM 145, ARGESIM/ASIM Pub., TU Vienna, 2013.
- [13] Meisgen, F.: *Dynamic Load Balancing for Simulations of Biological Aging*. In: International Journal of Modern Physics C, Vol. 8, Issue 3, pp. 575-582, 1997.
- [14] Meisgen, F.: *Dynamische Lastausgleichsverfahren in heterogenen Netzwerken*. Aachen: Shaker Verlag, 1998.
- [15] Merz, M., Bröcker, E.: *Einsatz von Open Source Frameworks zur Parallelisierung von Dymola Simulationen*. In: Proc. ASIM/GI Workshop STS/GMMS, Ed.: W. Commerell, ISBN 978-3-9810998-3-6, Ulm, März 04-05, S. 283-289, 2010.
- [16] Nagel, K., Schreckenberg, M.: *A cellular automaton model for freeway traffic*. In: Journal

- de Physique I, Volume 2, Issue 12, pp. 2221-2229, 1992.
- [17] Nicol, D. M.: *The cost of conservative synchronization in parallel discrete event simulations*. Journal of the Association of Computing Machinery, Vol. 40, No. 2, pp. 304-333, 1993.
- [18] Rönnegren, R., Ayani, R.: *A Comparative Study of Parallel and Sequential Priority Queue Algorithms*. In: ACM Transactions on Modeling and Computer Simulation, Vol. 7, No. 2, pp. 157-209, 1997.
- [19] Schlagenhaft, R.: *Dynamischer Lastausgleich optimistisch synchronisierter, verteilter Simulation*. In: Proc. ASIM-Workshop VSPP, 1999.
- [20] Sedgewick, R.: *Algorithms in C++*, Vol. 1. Boston: Addison-Wesley, 1998.
- [21] Steinman, J.: *SPEEDES: Synchronous parallel environment for emulation and discrete event simulation*. Advances in Parallel and Distributed Simulation, SCS Simulation Series, Vol. 23, pp. 95-103, 1991.
- [22] Ullrich, O.: *Modellbasierte Parallelisierung von Anwendungen zur Verkehrssimulation - Ein dynamischer und adaptiver Ansatz*. Dissertation, Univ. Köln, 2014.
- [23] Ullrich, O., Lückerrath, D., Franz, S., Speckenmeyer, E.: *Simulation and optimization of Cologne's tram schedule*. In: Simulation Notes Europe (SNE), ARGESIM/ASIM Pub., TU Vienna, Volume 22, Number 2, pp. 69-76, 2012.
- [24] Ullrich, O., Proff, I., Lückerrath, D., Kuckertz, P., Speckenmeyer, E.: *Agent-based modeling and simulation of individual traffic as an environment for bus schedule simulation*. In: Proceedings of mobil.TUM 2013, 10 pg, to appear.
- [25] Ullrich, O., Lückerrath, D., Speckenmeyer, E.: *A robust schedule for Montpellier's Tramway network*. In: Proceedings of ASIM 2014 - 22nd Symposium on Simulation Technique, Berlin, 2014, 9 pg.
- [26] Zeng, A.Z., Durach, C.F., Fang, Y.: *Collaboration decisions on disruption recovery service in urban public tram systems*. In: Transportation Research Part E, 48, pp. 578-590, 2012.

Cloud Computing Applied to Calibration of Flood Simulation Models

Marco Brettschneider¹, Bernd Pfützner², Frank Fuchs-Kittowski¹

¹Hochschule für Technik und Wirtschaft HTW Berlin

²Büro für Angewandte Hydrologie BAH Berlin

Marco.Brettschneider@htw-berlin.de

Flood water simulations have to be accurate in order to prevent losses and must therefore be based on well calibrated rainfall runoff models of sufficient complexity. IaaS cloud computing has been proven useful as a fast and cost-efficient extension of local computers, in the case, using the calibration of the hydrologic modeling system ArcEGMO as an example. On the other hand, cloud-based systems cannot serve as a replacement for a local computer, as long-term usage is still too expensive and there surely are performance differences regarding the offered architectures. Ultimately, the usage of cloud computing as a viable extension of local computing capacities is only reasonable if state of the art computer technologies with a fair pricing system can be offered by IaaS providers.

1 Introduction

Flood waters and flash floods are an ever-leading cause for significant destruction in the areas surrounding rivers [1]. In Germany, for example, the Elbe floods of 2002 and 2013 caused damages totaling millions of euros [2]. The simulation of flood waters, used to determine design parameters for the planning of flood protection measures, also serves to aid in predicting a flood event. This is of high importance in the preemptive assessment of possible risks from flood waves, allowing ample time to implement measures for prevention [3]. The simulation itself can be subdivided into one hydrological part to express the runoff processes in the area and another hydraulic part to illustrate the actual flood characteristics. In urban areas the contribution of urban drainage is often added as well [4]. This study focuses on the hydrological aspect of flood water simulation, which is used for general rainfall runoff models.

Depending on the chosen model complexity, the calibration of a rainfall runoff model can be a very time-consuming process. Robust values for the model parameters can often only be determined by a large number of model runs. Additionally, high-resolution models found in large-scale applications risk reaching the limits of computing capacity rather quickly. If one would like to test several parameterizations, he would first have to think about alternative methods of computation.

The computing technologies available are constantly changing, and there are now various ways to involve greater computing capacity for a more detailed de-

scription of hydrological processes. These new technologies include multi-core and many-core architectures of current processors and graphics chips [e.g. 5], and non-local computational resources, which can enhance the local ones. Among the latter, cloud computing is one of the most prominent of these technologies. Here, the modeler outsources his calculations onto virtual computers which can be generated on demand from server farms of cloud service providers. The particular potential of the cloud lies in the scaling of resources: for every application, a corresponding configuration of computing power, memory, disk space, and operating system can be found, enabling the potential coupling of multiple virtual machines.

Cloud computing is still in an early stage. Reliable findings on the use of cloud computing in the field of hydrology and, in particular, for the simulation of flood events are not yet available. There are few reports on the application of cloud computing in hydrology, whereas their focus is mainly on other aspects such as uncertainty analyses, by using precalibrated models [e.g. 6].

Because cloud projects can be implemented quickly and at comparatively low monetary costs, it is reasonable to gain first-hand experience in small pilot projects [7]. In this paper, the outsourcing of model calculations for calibrating a rainfall runoff model for flood simulation on virtual machines in the cloud will be discussed. Within this study, particular emphasis is placed on the attainable computing power and the scalability of resources. A company that creates and maintains the models is also always trying to balance

the monetary costs with the resulting benefit, which is why the cost aspect is considered here as well.

The paper is organized as follows: in chapter 2, the hydrological modeling system ArcEGMO is presented as a basis for flood simulations. Also, the potential of cloud computing is briefly discussed, which is then used to frame the requirements for using clouds in the context of flood simulation. Chapter 3 presents the configurations and the setup of an investigation on cloud services based on an example scenario. In chapter 4, the results of the application example are presented, thereby demonstrating how cloud computing can meet the requirements set in chapter 2. In chapter 5, the potential of cloud computing for complex hydrological model calibration is finally evaluated in terms of computing power and monetary cost efficiency, accompanied by an outlook on further developments.

2 Requirements for cloud computing to assist rainfall runoff model calibration

2.1 Rainfall runoff modeling with the hydrological modeling system ArcEGMO

The hydrological modeling system ArcEGMO can be applied for physically sound simulations of all relevant processes of the water balance and flow regime in catchments of different forms and scale ranges in space and time [8, 9]. Its modular design, the possibilities for a variable surface structure and a set of interfaces for coupling with other models allow for the processing of a large variety of practical and scientific problems (Fig. 1).

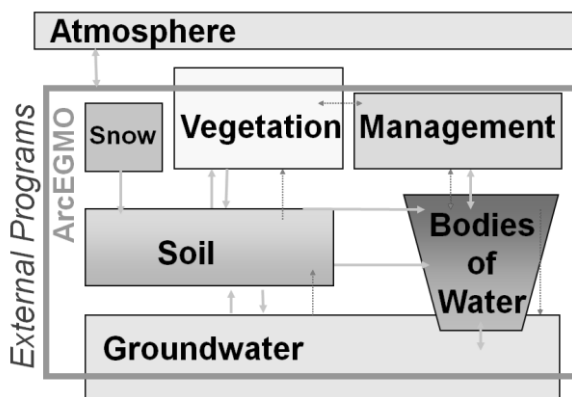


Figure 1. Domains of ArcEGMO

ArcEGMO has been used successfully in a variety of studies on flood generation for different landscape

types and field sizes [e.g. 10, 11]. It became clear that the parameterization of the model is crucial for the realistic representation of catchment processes and dynamics. Since the calibration of the model based on real measured values is an iterative process, a large number of model runs is needed to assess the effect of different parameter values on the ability to correctly predict river runoff. With complex models, the computation time can quickly become a limiting factor in the search for optimal parameters. Therefore, in the following it should be evaluated to what extent cloud computing can be specifically helpful as a way of fast acquisition of additional computing capacity to expand locally limited resources.

2.2 Potential of Cloud Computing

Cloud computing is a form of flexible and reliable deployment and use of IT resources and IT services. This may be IT infrastructure (network, servers, storage), platforms or applications of all kinds. These are provided as a service over the internet with minimal administrative effort in real time and charged according to consumption [7].

Cloud services are generally subdivided in the three levels of "Infrastructure as a Service (IaaS)", "Platform as a Service (PaaS)" and "Software as a Service (SaaS)". IaaS provides a minimum infrastructure for computing, used for storing data and network connectivity. The separation from PaaS is blurred, but the latter is usually associated with the development and deployment of Web services. With SaaS, complete software is directly provided for online use [12]. Due to the focus on pure processing power IaaS is used in this study and briefly described below.

With IaaS, IT resources are provided as virtual instances with complete virtual hardware and an operating system. From the user's perspective, IaaS looks like actual physical hardware: the user can control the entire operating system and the software running on it, and there are no restrictions in terms of applications that can be hosted by the system. To manage the infrastructure, specific web services are made available, which a user can use to start as many instances (virtual servers) as needed for a particular task in just a few minutes, as well as remove them quickly. The disadvantages of IaaS include the fact that an automatic scaling of the computing entities can be difficult to implement, the cost can increase rapidly if used for an extended period of time, and that there is

at least the theoretical possibility of an instance failure [13].

Companies invest in cloud computing especially because of its potential for cost reduction in situations where extended compute resources are needed only from time to time. Further benefits are in the shift from fixed costs (investments, etc.) to variable costs, as seen in the increase of the flexibility and scalability of IT resources as well as in the demand-based billing of IT services [7].

2.3 Requirements for the calibration of ArcEGMO on IaaS clouds

A well-calibrated rainfall runoff model is the key ingredient for reliable flood simulations and predictions for preventing damage. The effort required to run several model for an iterative search of robust parameter values can be lowered by cloud computing, but only when the calculation times are the same or shorter as for local computers and the resulting monetary costs remain low. For payment models without long-term booking, the costs depend partly on the time between generating and removing an instance, as well as on additional fees for data storage, data backup and communication with or between instances [14, 15]. The additional services listed above were not considered in this study, because the size, and with that the monetary costs of transferred input and output, are negligible, as is the loss of information and the expected additional costs due to sudden failure of an instance regarding model calibration. In this survey, the computing times are the determining criterion for assessing the benefits of cloud computing for the calibration of rainfall runoff models. The time consumed by delays in data transfer is important as well, but is usually small compared to computation time, especially when calibrating with lots of spatial elements.

The cloud services of Amazon and Microsoft offer configurations which are comparable to those of regular desktop computers, but also instance types, which are on the same level as modern server computers [16, 17, 18]. The services of Amazon EC2 and Microsoft Windows Azure are therefore used for the upcoming considerations. In addition to the computation times and the monetary costs, the extent to which computing power can be increased by each provider will also be investigated, as well as how the costs evolve in those cases.

3 Configuration of IaaS services for the calibration of flood simulations

3.1 Example scenario

To oppose monetary costs and benefits of cloud computing, many different approaches are already available which focus on the profitability at the scale of larger companies and governmental institutions. These are, for example, based on heuristic methods or break-even analyses [e.g. 15, 19, 20]. Here, the chosen method is also heuristic, but the focus is on the performance of the various types of instances. In the following, we will show based on a realistic scenario, how powerful virtual IaaS machines are in terms of rainfall runoff model calculations with ArcEGMO, with regard to simultaneous consideration of the respective costs. The description of the scenario in terms of the configuration of the model and the IaaS services are the topics of this chapter.

3.2 Setup of the modeling system ArcEGMO

ArcEGMO is applied to a real world example catchment, which, in general, is also used for training purposes [9]. The catchment comprises an area of 59km² with lowland characteristics and consists of 362 spatial elements (hydrotropes), whereas a period of 5.5 years in daily resolution is calculated. The model input data consists of different hydrological and meteorological time series for runoff, precipitation and climate variables. The runoff data is used as the target during model calibration.

After ArcEGMO has been prepared based on its control files, a simulation can be started. The initialization stage of a model run is dominated by file reading and writing processes on the hard drive, while the simulation stage is characterized by memory access. For this reason, a subdivision of the overall calculation time regarding these two stages is applied.

ArcEGMO works sequentially, that is, only one processor core per simulation is used. The amount of memory in use is governed by the number of spatial elements that have to be processed during simulation. In this example scenario, the number is comparatively small, and, therefore, the computation time is short. Because of the sequential processing of ArcEGMO, only scalability based on the different processor types and disk architectures can be considered here. Since currently only Amazon provides multiple computing and storage architectures, the investigation on scalability refers only to EC2.

3.3 The applied IaaS services

Windows Azure possesses properties both of PaaS and IaaS, and thus covers a large range of possibilities for the development and operation of its own software, with a focus on web applications. Amazon's Web Services (AWS) concentrate on flexibility and the combination of resources provided for any kind of use with focus on IaaS. Moreover, there is a wide range of operating systems which, if chosen, already have additional software and software development tools preinstalled.

Within AWS, EC2 is the service to deploy virtual computing resources. Billing is exact to the hour, additional hard drive and communication capacities can be booked with surcharge [16]. In Windows Azure, virtual machines are started and monitored via the Azure user interface. The selected configuration of the instance and the use of additional services or storage capacities determine again the final costs per hour [18].

3.4 The Selection of instance configurations

For both considered cloud providers, configurations have been selected which correspond to the hardware of a local desktop computer. An exact match and comparison between all three configurations is not possible since both providers use different hardware vendors (Intel for Amazon, AMD for Microsoft). The comparison can therefore be performed only in an approximate way.

The local computer has an E7400 dual-core processor from Intel with 64bit architecture and 2.8 GHz per core, 3 GB RAM, and HDD hard drive. The Amazon m1.large configuration was chosen as the appropriate instance, with 2 cores of an Intel Xeon processor (the clock frequency has not been specified by Amazon), and 7.5 GB RAM [17]. For Windows Azure the A2 configuration with AMD Opteron 2377 EE processor (2 x 1.6 GHz cores) and a total of 3.5 GB of memory has been selected [18, 21]. **Table 1** summarizes all configurations.

Table 2 shows a summary of both providers for the pricing models that do not require a reservation or subscriptions. The costs are exact to the hour. The costs per 30 days are related to a continuously active instance. Additional fees can be incurred for additional services such as storage, databases, network infrastructure, etc., but these services were not needed here, so their cost has been excluded. A local machine was not listed in this table, since variable and fixed

costs must be considered together, which prevents an exact comparison.

Amazon EC2 m1.large	Windows Azure A2	Local Computer
- Intel Xeon Processor, 2 Cores, 64bit	- AMD Opteron Processor, 2 Cores à 1,6 GHz, 64bit	- Intel E7400 Dual-Core à 2,8 GHz, 64bit
- 7 GB RAM	- 3,5 GB RAM	- 3 GB RAM
- Windows Server 2012	- Windows Server 2012	- Windows 7

Table 1. Hardware configuration of the local desktop computer and the selected cloud instance types

In this study, only the pricing models "on demand" and pay-as-you-go" have been investigated. These relate to a spontaneous provision of instances without having them reserved in advance.

	Amazon EC2 m1.large on demand	Windows Azure A2 pay-as-you-go
Price [Euros per hour]	0.26	0.135
Price [Euros per 30 days]	190.66	97.20

Table 2. Costs for cloud computing on Amazon's EC2 and Microsoft's Windows Azure for the pricing models "on demand" and "pay-as-you-go" (source: 16, 18)

To investigate the potential performance increase with improved cloud resources, instances furnished with the latest processor generation were included into the survey. The m3 instances use Intel Xeon E5-2670 Sandy Bridge processors and provide SSD hard drives, as well as c3 instances, which, in turn, are equipped with Xeon E5-2680v2 - Ivy Bridge processors [17]. For both the m3 and c3 instance types architectures with two cores have been acquired.

On Windows Azure, there currently is only one generation of instances available, which is why the used processors and hard disk capacity should be on the same level of performance. Additional test confirmed this assumption, but the according results are not presented here.

4 Results of the investigation of IaaS services for the calibration of flood simulations

4.1 Comparison of all configurations

For Microsoft and Amazon, hardware configurations can be found which are similar to those of the local computer, even though a direct comparison is not possible due to differences regarding the hardware component suppliers for both computing services. Furthermore, Amazon also offers the latest processor generation at the level of current server computers. Regarding the long term monetary costs, the acquisition of a local device is cheaper: currently, a computer as it is used and presented here is available for approximately 600€[22]. Also, if other variable costs like e.g. electricity are taken into account as well, IaaS services with costs of around 200€ and 100€ for a month of demand to pay at Amazon and Microsoft, respectively, are of benefit only in cases of short and short-term application (Table 2). Pricing models with reservation or subscription were not considered here, since in this case the hiring of server capacities would be more favorable [e.g. 23].

4.2 Results for ArcEGMO model runs

Figure 2 shows the calculation times utilized by each provider in general and in comparison to the local computer. It is noteworthy that the virtual machines consume more time than the local device. These differences arise mainly during the simulation phase. This suggests that the pure computing power of technologically similar cloud instances is less than that of a corresponding local computer. Reading and writing processes on the hard disk during the model initialization are thus characterized by a very similar consumption of time.

However, if instances of the latest hardware generation can be applied, as provided by Amazon, there will be a significant performance improvement (**Fig. 3**).

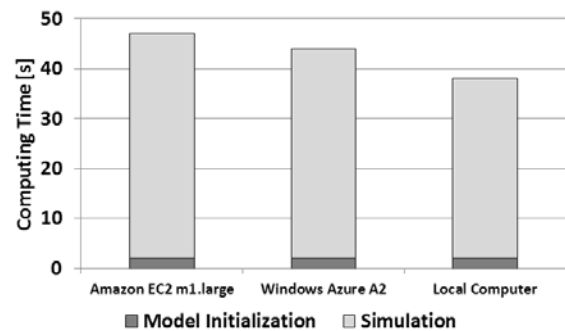


Figure 2. Computing time of ArcEGMO on different platforms

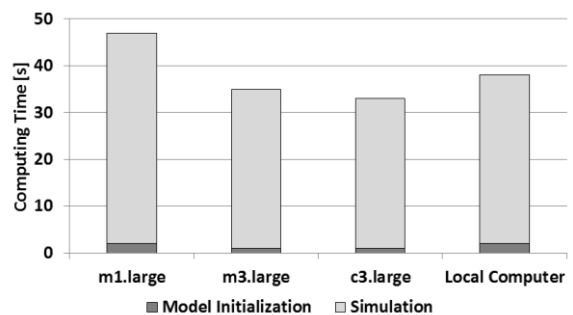


Figure 3. Differences between the Amazon EC2 instance types and the local machine in the required computing time

If the number of cores per instance is increased, the speed of computing, as expected, cannot be increased, since these additional resources are not applicable due to the sequential programming of the model (results are not presented here). Surprisingly, in Amazon's pricing model, the acquisition of the latest processor hardware is cheaper than using older generations (16; **Fig. 4**).

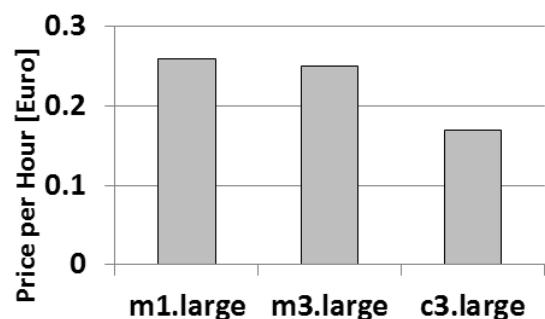


Figure 4. Price per hour for different Amazon EC2 instance types with no additional fees

Considering Amazons “on demand” prices without reservation on a monthly basis (180 € and 125 € for m3.large and c3.large), long term use is still too expensive. For less complex areas where the overall computation time is small, the monetary costs without additional storage and communication services are below 20 cents per hour for an instance with the latest and most powerful computing hardware. However, if large, highly spatially resolved models are calibrated so that the computing power and memory demands increase drastically for several days, costs are quickly increased. If, in that case Amazon instances equipped with the latest hardware generation are applied, one can still expect good operation performance of the model compared to a local device.

5 Conclusion and Outlook

Flood water simulations must be reliable in order to prevent losses and therefore need to be based on well calibrated rainfall runoff models of sufficient complexity. Using the hydrological modeling system ArcEGMO, it could be demonstrated that multiple model calculations, as necessary for model calibration, can also be implemented on virtual machines in IaaS clouds. However, the computing power is slightly below that of a comparable local computer with similar hardware. The outsourcing of model calculations is only reasonable if the latest computer hardware generation is offered by IaaS providers. The costs incurred thereby are low, especially for models with little complexity.

Due to the sequential structure of the modeling system ArcEGMO, additional processor cores cannot yet be exploited for a model run, which is why only a small increase in performance through improved processor technology in the cloud can be achieved. In this case, cloud computing is currently only an alternative in the particular circumstances of a short-term, increased computational demand, which can not be fulfilled by local computing resources.

The trend of recent years towards technologies of software parallelization will have an influence on the development of novel hydrologic models. The conversion of the model system ArcEGMO towards the integration of multiple cores is currently tested by the authors. The advantage is that calculations can be distributed to multiple cores, which reduces the computing time per simulation depending on the proportion of parallelizable source code. Disadvantages are the major programming effort and the greater need

for technical expertise. Existing code is not indefinitely parallelizable, so cloud computing will remain attractive in the long run, since parallelization is well supported due to the scaling of virtual hardware and since the pricing levels are expected to decrease.

Basically, there are further possibilities to decrease the amount of time required for model calibration. Optimization methods for automated model calibration, for example, can be instructed to distribute the individual model runs during calibration on multiple processor cores and to assume the management of the model runs, if supported. Corresponding studies by the authors, which also take cluster computers into account, are in preparation.

6 Acknowledgements

This study was created as part of the "EEnVEB" project. The project is funded by the European Union (European Social Fund).

7 References

- [1] B. Bates, Z. W. Kundzewicz, S. Wu and J. Palutikof: *Climate Change and Water. IPCC Technical Paper VI, Intergovernmental Panel on Climate Change (IPCC, Geneva)*. www.ipcc.ch, 2008.
- [2] Gesamtverband der Deutschen Versicherungswirtschaft: *Erste Schadensbilanz: Hochwasser 2013 verursacht 180000 versicherte Schäden in Höhe von fast 2 Milliarden Euro*. Pressemitteilung. Berlin, 2013.
- [3] M. Bornebusch: *Hydraulische und Hydrologische Modellsimulationen als Planungswerkzeug für Hochwasser-Schutz-Maßnahmen*. Aachen, 2006.
- [4] S. Maßmann, F. Jakobs, F. Sellerhoff, J. Feldmann, H. Sieker, C. Lange, Y. Om and R. Hinkelmann: *Hyd³Flow - Integrierte hydrologische und hydro-numerische Modellsysteme für eine verbesserte Hochwasservorhersage*. Forum für Hydrologie und Wasserbewirtschaftung, "Tag der Hydrologie 2010", Braunschweig, 2010.
- [5] E. Rouholahnejad, K.C. Abbaspour, M. Vejdani, R. Srinivasan, R. Schullin and A. Lehmann: *A parallelization framework for cali-*

- bration of hydrological models. Environmental Modelling & Software 31 (May), 2012.
- [6] V. Moya Quiroga, I. Popescu, D.P. Solomatine and L. Bociort: *Cloud and cluster computing in uncertainty analysis of integrated flood models*. Journal of Hydroinformatics 15(1), 2012.
- [7] BITKOM: *Cloud Computing - Evolution in der Technik, Revolution im Business*. BITKOM-Leitfaden, Berlin, 2012.
- [8] B. Pfützner: *Description of ArcEGMO; Official homepage of the modelling system ArcEGMO*. www.arcegmo.de, ISBN 3-00-022290-5, 2002.
- [9] B. Pfützner, P. Hesse, S. Mey and B. Klöcking: *N-A-Modellierung mit ArcEGMO*. In: J. Dietrich und M. Schöninger (Editors): *Hydro-Skript*. <http://www.hydroskript.de>, 2013.
- [10] B. Pfützner and P. Hesse: *Anwendung hydrologischer Modelle für die Hochwasserbemessung - Erfahrungen aus Sachsen-Anhalt*. In: A. Schulte, C. Reinhardt, A. Dittrich, R. Jüpner und V. Lüderitz (Editors): *Hochwasserdynamik und Risikomanagement - neue Ansätze für bekannte Probleme? Beiträge zum Gemeinsamen Kolloquium am 24.11.2011 in Berlin*. Berichte aus der Geowissenschaft Freie Universität Berlin, Aachen, 2011
- [11] U. Büttner, V. Thieming, E. Lechtaler and B. Pfützner: *Kap. 2 Untersuchungsgebiet*. In: Schumann, A. (Editor): *Entwicklungen integrierter Lösungen für das operationelle Hochwassermanagement am Beispiel der Mulde*. Schriftenreihe Hydrologie / Wasserwirtschaft Ruhr-Universität Bochum, 2009.
- [12] Tonninger, W.: *Die Cloud-Gretchen-Frage: IaaS oder PaaS?* <http://businessreadyblog.wordpress.com/2011/02/25/die-cloud-gretchen-frage-iaas-oder-paas/>, 2011 (validated 11.3.2014).
- [13] M. Walterbusch and F. Teuteberg: *Datenverluste und Störfälle im Cloud Computing: Eine quantitative Analyse von Service Level Agreements, Störereignissen und Reaktionen der Nutzer*. Proceedings: Multikonferenz Wirtschaftsinformatik, Paderborn, 2014.
- [14] E. Deelman, G. Singh, M. Livny, B. Berriman and J. Good: *The Cost of Doing Science on the Cloud: The Montage Example*. In: Proceedings of the 2008 ACM/IEEE conference on Supercomputing, article No. 50. IEEE Press, Piscataway, 2008.
- [15] C.I. Berendes, M. Ertel, T. Röder, T. Sachs, T. Süptitz and T. Eymann: *Cloud Computing lohnt sich (noch) nicht*. 11th International Conference on Wirtschaftsinformatik, Leipzig, 2013.
- [16] Amazon Web Services: *Amazon EC2 – Preise*. aws.amazon.com/de/ec2/pricing, 11.3.2014.
- [17] Amazon Web Services: *Amazon EC2-Instances*. aws.amazon.com/de/ec2/instance-types/, 11.3.2014.
- [18] Microsoft: *Windows Azure: Virtuelle Computer – Preisdetails*. www.windowsazure.com/de-de/pricing/details/virtual-machines/, 11.3.2014.
- [19] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia: *A View of Cloud Computing*. Communications of the ACM 53 (4), 2010.
- [20] R. Matros, P. Stute, N.H. von Zuydtwyck and T. Eymann: *Make-or-Buy im Cloud-Computing – Ein entscheidungsorientiertes Modell für den Bezug von Amazon Web Services*. Bayreuther Arbeitspapiere zur Wirtschaftsinformatik, Bayreuth, 2009.
- [21] C.H. Hochstätter: *Microsofts Sicht auf die Cloud: Windows Azure im Praxistest*. www.zdnet.de/41554705/microsofts-sicht-auf-die-cloud-windows-azure-im-praxistest/, 2011 (validated 11.3.2014).
- [22] Innova Handelshaus AG: www.innova24.biz/item/computer-und-navigation/pc-systeme/acer-desktop-veriton-s670g-intel-core2duo-e7600-2gb-intel-g-793994.htm, 2013 (validated 13.3.2014).
- [23] Hetzner Online AG: www.hetzner.de/hosting/produktmatrix/rootserver-produktmatrix/, 2014 (validated 13.3.2014).

Simulating ventricular elastance with a heart-arterial interaction model

Anita Gerstenmayer¹, Bernhard Hametner², Stephanie Parragh^{1,2}, Thomas Weber³,
Siegfried Wassertheurer²

¹Department for Analysis and Scientific Computing, Vienna University of Technology,
Vienna, Austria

²Health & Environment Department, AIT Austrian Institute of Technology GmbH,
Vienna, Austria

³Cardiology Department, Klinikum Wels-Grieskirchen, Wels, Austria
e0926858@student.tuwien.ac.at

Mathematical models are necessary to understand the cardiovascular system and can assist in clinical conditions. The purpose of this paper is to describe a simple model based on the interaction of the ventricle with the arterial system that is able to simulate ventricular properties, such as ventricular elastance or blood pressure, using arterial pressure as an input. To achieve this aim, an already validated heart-arterial interaction model is adapted. It consists in the coupling of a Windkessel model for the arterial system with the time-varying elastance concept for the ventricular function. During the ejection phase ventricular elastance can be calculated from arterial pressure using differential equations. For the rest of the cardiac cycle the elastance curve needs to be extended appropriately. The model is tested using an invasively measured pressure data set for simulation. Measured aortic pressure is used to compute ventricular elastance, which is then used as an input to the original model and the resulting ventricular pressure curve is compared to the measurement. The simulation results show overall a good agreement with the measured data and therefore suggest that the model could be used to assess ventricular properties.

1 Introduction

For the early diagnosis and treatment of cardiovascular diseases a comprehensive knowledge of the cardiovascular system is essential. Various mathematical models for simulating the heart and arterial system have been developed in order to contribute to its understanding and even assist clinical diagnosis [1]. For clinical applications complex models are only of limited use, as the assessment of the required parameters can be very difficult [2]. Thus, for this purpose simple models whose parameters have a straightforward physiological interpretation are necessary.

A very well-known and validated class of simple lumped-parameter models of the arterial system are the so-called Windkessel models. Their name derives

from the assumption that the large, compliant arteries act as a Windkessel where blood is stored during systole in analogy to an old-fashioned hand-pumped fire engine [3]. Numerous extensions of the original model consisting of a compliance and a resistance element have been developed by introducing new elements [4]. A way to describe ventricular activities was elaborated by Suga et al. in form of the elastance model [5]. Elastance is defined as the ratio of ventricular blood pressure to volume and thus provides a measure for the stiffness of the cardiac muscle.

Models for the heart and the arterial system like the ones introduced above are often considered separately. However, blood pressure and flow result from the interaction between the ejecting ventricle and the arterial system. Therefore models describing the coupling of heart and arteries have been established. An ex-

ample for such a heart-arterial interaction model was proposed by Segers et al. [6]. Their simulations are based on the assumption that important ventricular properties such as ventricular elastance are known. In clinical practice though, the assessment of ventricular properties is very difficult and expensive. More frequently, aortic blood pressure and flow are measured non-invasively. For this reason, the aim of this work is to adapt the heart-arterial interaction model in a way to derive properties of the left ventricle, namely elastance, blood pressure and volume, from aortic pressure measurements. In this paper, model approaches that accomplish this task are presented. In order to test the possible applicability of these models, simulations and measured data are compared.

2 Methods

The modeling approach is based on a coupling of the time-varying elastance concept describing the function of the left ventricle with the classical three- or four-element Windkessel model representing the arterial system. The time-varying elastance $E(t)$ is assumed to relate ventricular blood volume V_v to ventricular blood pressure P_v by

$$E(t) = \frac{P_v(t)}{V_v(t) - V_0}, \quad (1)$$

where V_0 is an experimentally determined correction volume [5]. It has been shown by Senzaki et al. that the elastance curves when normalized both by amplitude and time to peak amplitude are of nearly constant shape in human hearts [7]. Thus presuming this normalized shape as given, E is determined by four parameters: minimal and maximal value E_{min} and E_{max} , heart rate and time to reach maximal elastance. In [6] this approach for modeling the heart was taken to find model equations that use elastance as an input.

Windkessel models are lumped-parameter models describing the hemodynamics of the arterial system [3]. In this work a three- and a four-element model (WK3 and WK4) are used, consisting of the total peripheral resistance R , the total arterial compliance C , the characteristic impedance of the aorta Z_c and (in case of the four-element model) the total inertia of the arterial system L .

Segers et al. introduced and validated the heart-arterial interaction model with a four-element Windkessel illustrated in figure 1 [6]. The cardiac valves are modeled as perfectly closing devices allowing only forward flow and acting as simple resistances R_{A-V} for the mitral valve and R_{V-art} for the aortic valve when opened. The preload or filling rate of the heart is represented by the venous filling pressure P_{venous} .

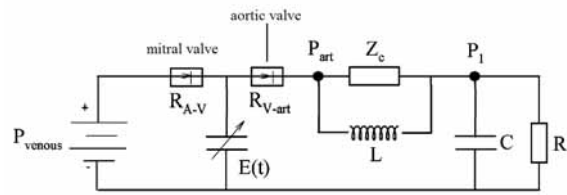


Figure 1: Electrical analog representation of the heart-arterial interaction model with WK4 (adapted from [6]).

The combination of Windkessel and elastance equations yields a system of three ordinary differential equations for the unknown ventricular blood volume V_v , Windkessel pressure P_1 and aortal pressure P_{art} :

$$\begin{aligned} \frac{dV_v}{dt} &= \frac{P_{venous} - E(V_v - V_0)}{R_{A-V}} - \frac{E(V_v - V_0) - P_{art}}{R_{V-art}} \\ \frac{dP_1}{dt} &= \frac{1}{C} \frac{E(V_v - V_0) - P_{art}}{R_{V-art}} - \frac{P_1}{RC} \\ \frac{dP_{art}}{dt} &= \left(1 + \frac{Z_c}{R_{V-art}}\right)^{-1} \left[\frac{Z_c}{R_{V-art}} \left(\frac{dE}{dt} (V_v - V_0) + E \frac{dV_v}{dt} \right) + \frac{dP_1}{dt} + \frac{Z_c}{L} (P_1 - P_{art}) \right]. \end{aligned} \quad (2)$$

These equations are solved throughout the whole cardiac cycle with R_{A-V} and R_{V-art} changing from infinity to an appropriate finite value according to the sign of $P_{venous} - P_v$ and $P_v - P_{art}$, respectively. Ventricular pressure P_v and aortic blood flow Q_{art} can be calculated using formula (1) and $\frac{dV_v}{dt} = -Q_{art}$.

In order to solve the model equations (2), the time-varying elastance function has to be known. However, in practice it is very difficult to assess the parameters defining the function $E(t)$, but comparatively easy to measure the aortic blood pressure P_{art} . Therefore the model was modified in a way to simulate E using measured P_{art} -curves. It must be pointed out that only during the ejection phase, when the aortic valve

is opened, an actual connection between the ventricle and the aorta exists. Hence only in this phase the elastance curve can really be calculated.

In order to find an equation for E during the ejection phase, $E(V_v - V_0)$ is replaced by P_v in the model (2). Rearranging the third equation, the new model with unknowns P_1 , P_v and V_v is given by

$$\begin{aligned} \frac{dV_v}{dt} &= -\frac{P_v - P_{art}}{R_{V-art}} \\ \frac{dP_1}{dt} &= \frac{1}{C} \frac{P_v - P_{art}}{R_{V-art}} - \frac{P_1}{RC} \\ \frac{dP_v}{dt} &= \frac{dP_{art}}{dt} + \frac{R_{V-art}}{Z_c} \left(\frac{dP_{art}}{dt} - \frac{1}{C} \frac{P_v - P_{art}}{R_{V-art}} \right. \\ &\quad \left. + \frac{P_1}{RC} - \frac{Z_c}{L} (P_1 - P_{art}) \right). \end{aligned} \quad (3)$$

E can then be calculated from formula (1). Note that this model also requires the time derivative of P_{art} which has to be approximated numerically.

For the rest of the cardiac cycle E needs to be extended in a suitable way. This was achieved by assuming that the elastance function is of a form similar to the one used by Danielsen et al. [8] but with an additional sinus term to obtain a faster increase of E in early systole. The elastance function used in this work (figure 2) is given by

$$E(t) = E_{min} \cdot (1 - \phi(t)) + E_{max} \cdot \phi(t), \quad (4)$$

where the function ϕ is defined as

$$\phi(t) = \begin{cases} a_\phi \sin\left(\frac{\pi t}{t_{ce}}\right) - b_\phi \sin\left(\frac{2\pi t}{t_{ce}}\right) + 0.1 \cdot \sin\left(\frac{3\pi t}{t_{ce}}\right) & 0 \leq t < t_{ce} \\ 0 & t_{ce} \leq t \leq T \end{cases} \quad (5)$$

and t_{ce} denotes the time for the onset of constant elastance.

The values of E determined during the ejection phase are used to estimate the parameters a_ϕ , b_ϕ , E_{max} , E_{min} and t_{ce} in a way that the function given by (4) and (5)

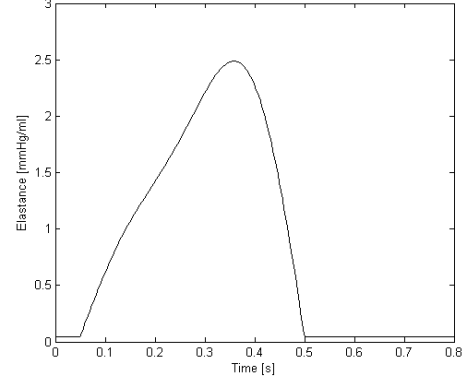


Figure 2: Time-varying elastance $E(t)$ given by (4) and (5) with $a_\phi = 0.9$, $b_\phi = 0.25$, $t_{ce} = 0.45$ s, $E_{max} = 2.49$ mmHg/ml, $E_{min} = 0.049$ mmHg/ml.

approximates the calculated curve E as well as possible during the ejection phase. As maximal elastance occurs during the ejection phase, E_{max} is determined by the calculated values. The other four parameters are each varied over an appropriate range of values to minimize the difference between the analytical function and the simulated values.

In a similar way, model equations for a coupling of the elastance concept and the three-element Windkessel model can be deduced (figure 3).

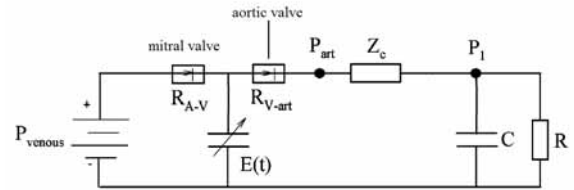


Figure 3: Electrical analog representation of the heart-arterial interaction model with WK3 (adapted from [6]).

The use of the three-element Windkessel leads to a system of only two differential equations for P_1 and V_v . The equation for V_v follows from $\frac{dV_v}{dt} = -Q_{art}$ and the fact that the blood flow through the aortic valve Q_{art} equals the flow through the resistance Z_c :

$$\frac{dV_v}{dt} = -Q_{art} = -Q_{Z_c} = \frac{P_1 - P_{art}}{Z_c}. \quad (6)$$

An equation for P_1 results from the equality $Q_{Z_c} =$

$Q_C + Q_R$ where the flow through R is given by $Q_R = \frac{P_1}{R}$ and the flow through C by $Q_C = C \frac{dP_1}{dt}$. Using $Q_{Z_c} = \frac{P_{art} - P_1}{Z_c}$ yields

$$\frac{dP_1}{dt} = \frac{P_{art} - P_1}{CZ_c} - \frac{P_1}{RC}. \quad (7)$$

Having solved these two equations, the equality $Q_{Z_c} = Q_{art}$ can be used to obtain

$$\frac{E(V_v - V_0) - P_{art}}{R_{V-art}} = \frac{P_{art} - P_1}{Z_c}.$$

Consequently, it is possible to calculate E from the measured aortic pressure P_{art} and the solutions V_v and P_1 from equations (6) and (7) by the formula

$$E = \frac{1}{V_v - V_0} \left(\frac{R_{V-art}}{Z_c} (P_{art} - P_1) + P_{art} \right). \quad (8)$$

3 Results

In order to test the previously described model, a measured aortic and left ventricular pressure data set was used for a simulation. The measured pressure curves are shown in figure 4. Pressure was measured invasively applying a 5F Millar SPC-454D catheter (see [9] for details). The study was approved by the regional ethics committee, and all of the participants gave written informed consent.

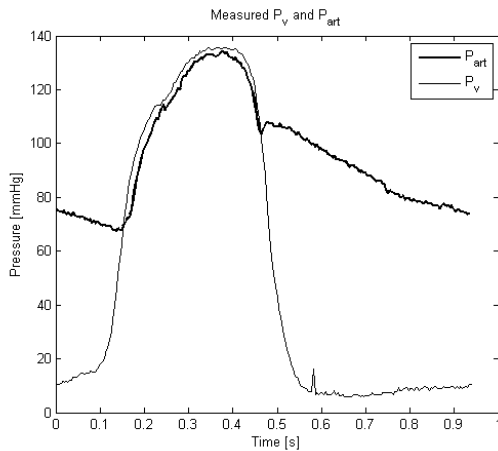


Figure 4: Measured aortic and ventricular pressure curves.

The measured aortic pressure curve was taken as an input for the model (3) with WK4 and the model (6)-(8) with WK3 to determine two sets of parameters for the elastance function (4)-(5). Then these elastance parameter sets for WK4 and WK3 were again used as an input for the corresponding original models represented in figure 1 and 3 in order to compute ventricular and aortic pressure and volume. The resulting simulated ventricular pressure curves were compared to the measured data.

In addition to the aortic pressure, the model for calculating the elastance curve and the original model of Segers et al. need some more parameters, namely the Windkessel parameters R , C , Z_c and L , the duration of one heartbeat T , the filling pressure P_{venous} , the enddiastolic volume EDV , the correction volume V_0 and the valve resistances R_{V-art} and R_{A-v} . The Windkessel parameters were estimated empirically in a way to fit the aortic pressure curve. The values used for this simulation are given in table 1. T was deduced from the P_{art} -curve as $T = 0.94$ s. As no ventricular volume data was available, EDV and V_0 were given appropriate values. For this simulation $EDV = 120$ ml and $V_0 = -10$ ml were chosen. The venous filling pressure, which is only used for simulating the original model, was calculated from the relation $E_{min} = \frac{P_{venous}}{EDV - V_0}$ with the minimal elastance value found before. The valve resistances were set to $R_{V-art} = R_{A-v} = 0.007$ mmHg/(ml/s). The parameter values used here are in the same range as those specified in [10].

	WK4	WK3
R [mmHg/(ml/s)]	0.9	0.9
C [ml/mmHg]	0.8	1.0
Z_c [mmHg/(ml/s)]	0.09	0.08
L [mmHg/(ml/s ²)]	0.002	-

Tabelle 1: Values for the Windkessel parameters.

Figure 5 shows the elastance curves simulated with the parameters given above for the WK4 model (3) in the upper panel and the WK3 model (6)-(8) in the lower panel. The elastance values calculated from the differential equations for the ejection phase are illustrated (asterisks) as well as the approximating functions of the form (4) (solid line).

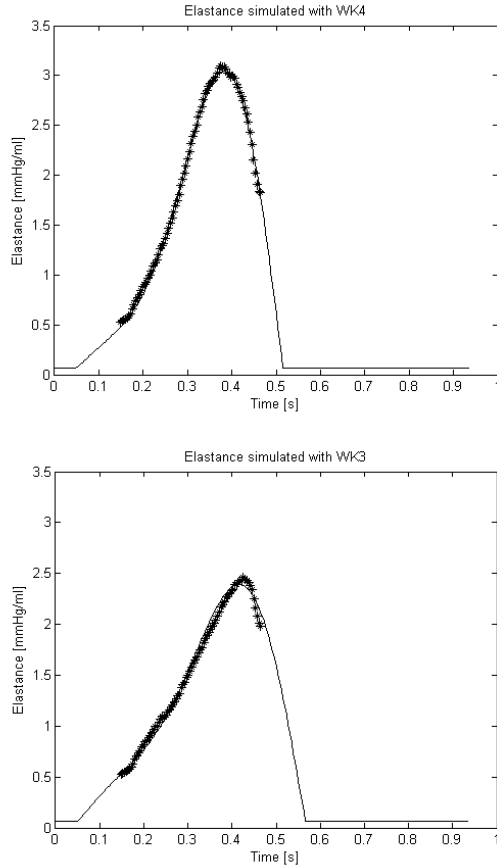


Figure 5: Elastance curves simulated with WK4 (upper panel) and WK3 (lower panel). The asterisks denote the simulated values during ejection phase, the approximating elastance function of form (4) is given by the solid line.

Figure 6 shows the simulation results of the original models using the corresponding elastance parameter sets found before. The simulation with the WK4 model given by equations (2) is shown in the upper panel, the WK3 model results can be seen in the lower panel.

Finally, in figure 7 the outcome of the simulations with WK4 and WK3 and the real ventricular pressure data are compared.

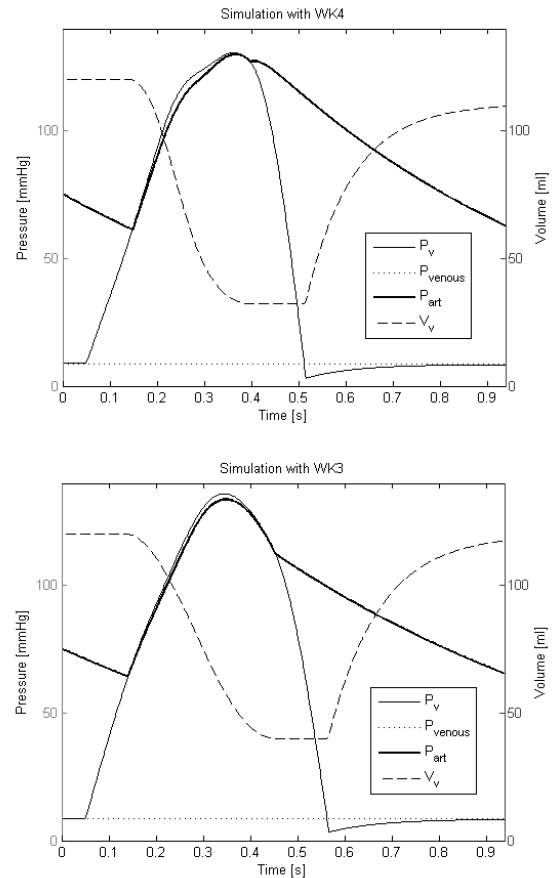


Figure 6: Simulation of the original model with WK4 (upper panel) and WK3 (lower panel).

4 Discussion

The main objective of this work was to find a simple model based on the interaction of ventricle and arteries that is able to simulate properties of the heart using arterial pressure as an input. The simulation results given in figures 5-7 indeed suggest that the heart-arterial interaction models represented in figure 1 and 3 achieve this goal.

A first observation taken from figure 5 is that the elastance values calculated for the ejection phase are in the expected range of values for human hearts and resemble the form found in measurements [7]. Furthermore it can be seen that the functions given in (4) and (5) are suitable for approximating these values and provide an extension to the whole cardiac cycle that resembles

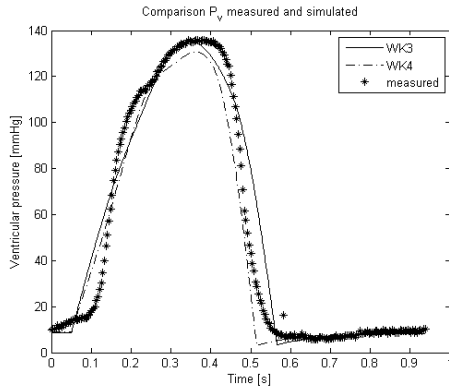


Figure 7: Comparison between measured ventricular pressure and P_v simulated with WK4 and WK3.

the curves found in [7].

Comparing the results for the four-element Windkessel to those obtained with the three-element model in figure 5, it is evident that both models qualitatively yield very similar elastance functions. The maximal elastance E_{max} calculated with the WK4 model is higher than the maximal value retrieved from the WK3 simulation. Also, the time to reach this maximal value is shorter in the WK4 model than in the WK3 version.

Considering the simulation results for P_{art} , P_v and V_v obtained through simulating the original models using the previously calculated elastance parameters (figure 6), one notable difference is that in the WK4 simulation the aortic valve closes much earlier than in the WK3 simulation and the real data. However, unlike the WK3 model, the four-element Windkessel is able to model the short increase in aortic pressure in early diastole that is also seen in the measured P_{art} curve in figure 4, whereas use of the WK3 model leads to a simple exponential decay in aortic pressure during diastole.

As can be seen from figure 7, there is a good agreement between measured and simulated ventricular pressure. The shape of all curves is similar and important values like minimal and maximal pressure hardly differ. During the isovolumic contraction phase though, the simulated pressure does not increase as fast as the measured pressure, and at the beginning of the filling phase simulated pressure decreases a bit too far. Although the WK4 model yields higher elastance

values than the WK3 model, there is not as much difference in the resulting ventricular pressure curves.

Although the model yields promising simulation results, several limitations need to be acknowledged. First, in this work the model was only applied to one specific data set. Of course, to thoroughly validate the proposed model, testing with a larger cohort of patient data is necessary. Also the aortic and ventricular pressure curves used for the simulation were measured at the same patient but not at the same time. Another source of weakness is that several of the necessary parameters were given appropriate values instead of being measured or determined algorithmically. The findings are also limited by the lack of information on ventricular blood volume, which would be necessary to directly compare simulated elastance curves to measurements. Naturally, measuring errors may occur when using catheters, which means that simulation results cannot be compared to real values, only to these measured values. However, in the study that was used here, pressure-tip catheters were applied, which currently provide one of the most accurate measuring methods for blood pressure [11]. One of the most important advantages of the model is the small number of parameters required and their straightforward physiological interpretation. On the other hand, the low number of model parameters at the same time provides a limitation, as already pointed out for the original model in [6].

In conclusion, a simple and physiologically relevant heart-arterial interaction model was elaborated and the simulation results support the idea that this model can be applied to assess ventricular properties. However, further research and testing with a larger cohort with additional measurements is required to decide whether the model has practical relevance.

As a perspective for possible future research, the heart-arterial interaction model could be combined with other modeling techniques to use non-invasively measured pressure curves as an input. Aortic pressure curves can be determined from peripheral measurements by using transfer functions, for example with the algorithm used in [9]. Coupling such an algorithm to this model could make it possible to assess important ventricular parameters only by carrying out a simple brachial cuff based measurement. In this way the model could support the analysis of cardiovascular function in clinical conditions.

References

- [1] F.N. van de Vosse and N. Stergiopulos. Pulse Wave Propagation in the Arterial Tree. *Annual Review of Fluid Mechanics* 43, p. 467-499, 2011.
- [2] B.W. Smith, J.G. Chase, R.I. Nokes, G.M. Shaw and G. Waked. Minimal haemodynamic system model including ventricular interaction and valve dynamics. *Medical Engineering & Physics* 26, p. 131-139, 2004.
- [3] N. Westerhof, J. Lankhaar and B. Westerhof. The arterial Windkessel. *Medical and Biological Engineering and Computing* 47(2), p. 131-141, 2009.
- [4] Y. Shi, P. Lawford and R. Hose. Review of Zero-D and 1-D Models of Blood Flow in the Cardiovascular System. *BioMedical Engineering On-Line* 10, article 33, 2011.
- [5] H. Suga, K. Sagawa and A. Shoukas. Load Independence of the Instantaneous Pressure-Volume Ratio of the Canine Left Ventricle and Effects of Epinephrine and Heart Rate on the Ratio. *Circulation Research* 32(3), p. 314-322, 1973.
- [6] P. Segers, N. Stergiopulos, N. Westerhof, P. Wouters, P. Kolh and P. Verdonck. Systemic and pulmonary hemodynamics assessed with a lumped-parameter heart-arterial interaction model. *Journal of Engineering Mathematics* 47, p. 185-199, 2013.
- [7] H. Senzaki, C.-H. Chen and D. A. Kass. Single-beat estimation of end-systolic pressure-volume relation in humans. A new method with the potential for noninvasive application. *Circulation* 94(10), p. 2497-2506, 1996.
- [8] M. Danielsen and J. Ottesen. *A Cardiovascular Model*. In: *Applied Mathematical Models in Human Physiology*. By: J. Ottesen, M. Olufsen and J. Larsen, Roskilde University, Denmark, p. 113-126, 2003.
- [9] T. Weber, S. Wassertheurer, M. Rammer, E. Maurer, B. Hametner, C. Mayer, J. Kropf and B. Eber. Validation of a Brachial Cuff-Based Method for Estimating Central Systolic Blood Pressure. *Hypertension*, p. 825-832, 2011.
- [10] P. Reymond, F. Merenda, F. Perren, D. Rüfenacht, and N. Stergiopulos. Validation of a one-dimensional model of the systemic arterial tree. *American Journal of Physiology-Heart and Circulatory Physiology* 297, p. H208-H222, 2009.
- [11] A. de Vecchi, R.E. Clough, N.R. Gaddum, M.C.M. Rutten, P. Lamata, T. Schaeffter, D.A. Nordsletten, and N.P. Smith. Catheter-induced Errors in Pressure Measurements in Vessels: an In-vitro and Numerical Study. *IEEE Transactions on Biomedical Engineering*, in press, DOI 10.1109/TBME.2014.2308594, 2014.

Evaluation of Risk Factors for Parametrization of Cancer Models

Andreas Bauer¹, Günther Zauner², Christoph Urach², Felix Breiteneker¹

¹Vienna University of Technology, Institute for Analysis and Scientific Computing, Austria

²dwh Simulation Services, Vienna, Austria

andreas.e101.bauer@tuwien.ac.at

Cancer is the second most cause of death in Austria and around 38000 people are diagnosed with cancer each year [1]. The goal of this paper is to analyze methods for evaluation of risk factors in order to parametrize a micro simulation model for cancer prevalence. The focus of this paper is on modeling the survival time. This is done by the methods of survival analysis and model selection. Firstly, the survival function is estimated by the Kaplan-Meier estimate. Afterwards, a Cox proportional hazards regression is performed with all possible sets of parameters. These models are tested by twos with the likelihood ratio test in order to compare them. Another approach is the so-called Lasso method. This method puts a constraint on the sum of the absolute values of the regression coefficients and in most cases forces some of the coefficients to go to zero. The Akaike Information Criterion is also applied. All three methods are compared and the parameters which are supported, at least to a certain extent, by all of them are included in the estimation of the survival time of the prevalence model.

1 Introduction

Cancer is the second most cause of death in Austria and around 38000 people are diagnosed with cancer each year [1]. So, it is of great importance to find out the risk factors on one side, but also to model the incidence and prevalence to be able to evaluate health policy measures on the other side. An important step for doing a simulation is to find out the potential influences on the course of the disease and to quantify them in order to parametrize the model. The goal of this paper is to test methods for identification of possible influence factors on the course of cancer and to do a survival analysis for finding out the factors on which the course of the disease depends. Also, methods of model selection are used. These analyses will be used for the parametrization of a micro-simulation model for cancer prevalence later on.

2 Data

The following six categories are examined to find out, if they are possible influences on the development or the course of cancer: sex, age at the diagnosis date, chronic diseases X, Y and Z and the stage of cancer at the date of the diagnosis. In Table 1, an overview of these categories with according types and ranges is presented.

Number	Category	Type	Range
1	Age	ratio	23-83
2	Sex	nominal	0,1
3	Chronic disease Y	ordinal	0,1
4	Chronic disease Z	ordinal	0,1
5	Chronic disease X	ordinal	0,1
6	Stage of disease	ordinal	2-4

Table 1. Overview of categories with according types and ranges

3 Methods

3.1 Survival Analysis

In order to examine the survival time of the individuals depending on the possible influence factors, methods of survival analysis are applied. These methods allow the estimation and the analysis of the survival function and the hazard function. The survival function $S(t)$ is defined as the probability that an individual will survive up to time t and the hazard function $h(t)$ is defined as the instantaneous rate of death at time t .

Another important aspect regarding survival analysis is censoring. In the field of survival analysis often the data collection ends before the event of interest has

occurred for all individuals. For those individuals, the survival time cannot be determined. The only thing that is known is that the survival time exceeds the time of the observation of the particular patient.

The Kaplan-Meier estimate is an estimate for the survival function $S(t)$. It makes use of the information of the exact date of the occurrence of death. The estimated survival probability s_t at time t is:

$$s_t = \frac{n_t - d_t}{n_t} \quad (1)$$

n_t is the number of people alive at time t and d_t is the number of people that died at time t . So, s_t is simply the ratio of the people alive who survive time t . Thus, the probability of surviving up to a certain point of time t_j is calculated with the so-called product-limit formula [2]:

$$S(t_j) = \prod_{i=1}^j S(t_i) \quad (2)$$

A common approach to do regression analysis on survival data is the so-called Cox regression, also known as proportional hazards regression. It assumes that the ratio of the hazards comparing different exposure groups remains constant over time. This is called the proportional hazards assumption. The mathematical form of the proportional hazards model is:

$$h(t) = h_0(t) * \exp(\sum_{i=1}^n b_i * x_i) \quad (3)$$

$h_0(t)$ denotes the baseline hazard which refers to a particular group of individuals (for example, the individuals with value zero in all binary categories, with mean age and with stage of illness two), n is the number of covariates, x_i is the value of the i th covariate and b_i is the corresponding regression coefficients [3,4].

3.2 Model Selection

The methods of model selection can be used to find the significant covariates for our model depending on given data. The goal of model selection is to eliminate some of the covariates from the full model with six covariates to get a simpler model which still explains most of the effects correctly. In order to find an appropriate model, three approaches are considered: Likelihood ratio tests, Lasso – Method and Akaike Information Criterion (AIC).

Firstly, the Cox regression is performed with all possible sets of parameters. That means the parameter

sets of the models are all possible subsets of the full set with six parameters.

For each two nested models the likelihood ratio test is applied. With this test we examine, if the bigger model of the two significantly provides additional information in comparison to the smaller nested model. The significance level is set to 0.05.

Another approach to select a model is to use the Lasso-method. The regression coefficients of the Cox regression are calculated as usual by minimizing the partial log-likelihood, but additionally the sum of the absolute values of the regression coefficients is bounded by a constant in order to force some of the coefficient to shrink to zero. This results in a sequence of models depending on the size of the constraint. There are various ways to determine the “best” size of the constraint. It can be either chosen arbitrarily or automatically based on the data. For instance, the use of an approximate generalized cross-validation (GCV) statistic is a common tool for automatic constraint selection [5].

Another criterion to select a set of variables is the so-called “Akaike Information Criterion” (AIC). The AIC value is calculated as follows:

$$AIC = -2 \log \mathcal{L}(t_0|y) + 2K \quad (4)$$

The first summand is the negative of twice the numerical value of the log-likelihood at its maximum point t_0 given data y and the second summand is twice the number of parameters of the model. The smaller the AIC value of a model is, the better it is, because the AIC value can be interpreted as a kind of information loss [6]. The AIC can also be used for automatic choice of a constraint for the abovementioned Lasso-method.

4 Results

The Kaplan-Meier estimate was calculated for various groups of the population. Figure 1 shows the Kaplan-Meier estimates for male and female individuals in comparison. We can see that the estimate for the males is lower than the estimate for the females until about 800 days after the diagnosis, when only 10 percent of the individuals are left alive.

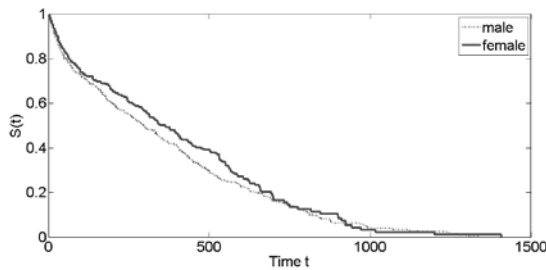


Figure 1. Comparison of Kaplan-Meier estimates for male and female individuals

The coefficients for the Cox regression with all six covariates included are shown in Table 2.

b_1	b_2	b_3	b_4	b_5	b_6
-0.11	0.02	-0.05	0.09	-0.25	0.02

Table 2. Coefficients of Cox regression in order: sex, age, chronic disease Y, chronic disease Z, chronic disease X, stage of illness

The p-values of the Cox regression show that the p-value of the term age is the lowest, so we start with the model with only term age. The likelihood ratio test shows that the model with added terms chronic disease X and sex and the model with added term chronic disease X, both given the term age, are statistically significant, while any other extension given the term age is not significant.

Figure 2 shows the values of the six regression coefficients of the Cox model plotted over the l_1 -norm of the coefficient vector. On the x-axis above the plot also the number of non-zero coefficients is displayed.

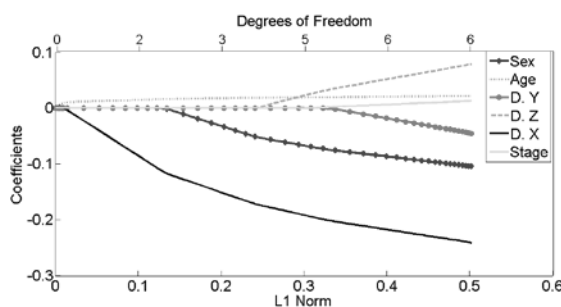


Figure 2. Cox regression coefficients over the norm of coefficients vector.

In Figure 2, we see that the smaller the norm of the vector gets, the smaller is the number of non-zero coefficients. The coefficient that vanishes at last, when the norm of the coefficient vector goes to zero, is the coefficient of the parameter age, right after the coefficients of the parameters chronic disease X and

sex. The other three coefficients are eliminated earlier.

The AIC value is calculated for the models with all 64 possible sets of variables. In order to compare different models, AIC differences are computed, because the relative values of the AIC are more meaningful than the absolute values. The AIC differences are computed by subtracting the AIC value of the model with the least AIC value from the AIC values of each model.

In Table 3, the five models with the least AIC values and the AIC differences are listed.

Parameters	AIC	AIC difference
1; 2; 5	6788.75	0
2; 5	6789.15	0.40
1; 2; 3; 5	6790.27	1.52
1; 2; 4; 5	6790.41	1.66
1; 2; 5; 6	6790.59	1.84

Table 3. Parameters sets with lowest AIC values and AIC differences, numbers referred to numbering of categories in Table 1

Table 3 shows that the model with parameters sex, age and chronic disease X has the lowest AIC-value followed by the model with parameters age and chronic disease X. The other three listed models are also substantially supported by the AIC.

5 Conclusion and Outlook

The categories age, sex and chronic disease X are found to be significant by likelihood ratio tests using the Cox regression. The AIC ranks this model as first too. The Lasso-Method shows that these three parameters are the last three parameters that are left, when the norm of the coefficient vector declines. Since the set of parameters is not very big, in uncertain situations, where it is not obvious, if a certain parameter should be included or not, the parameter will be included in the model to avoid the situation that a substantial effect is eliminated from the model by accident. So all used methods suggest that these three parameters definitely should be included in the estimation of the survival time for the prevalence model. For the other categories, further analysis will be done to determine, if they also will be included in the future model.

6 References

- [1] *Jahrbuch der Gesundheitsstatistik*. Statistik Austria, Austria, 2012.
- [2] B. R. Kirkwood and J. A. C. Sterne. *Essential Medical Statistics*, 2nd edition. Wiley-Blackwell, United Kingdom, 2003.
- [3] G. Rodriguez. *Lecture Notes on Generalized Linear Models*. Princeton University, United Kingdom, 2007.
- [4] D. R. Cox and D. Oakes. *Analysis of Survival Data*. Chapman & Hall, United Kingdom, 1984.
- [5] R. Tibshirani. *The Lasso Method for Variable Selection in the Cox Model*. Statistics in Medicine Volume 16, p. 385-395, 1997.
- [6] K. P. Burnham and D. R. Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*, 2nd edition. Springer, United States of America, 2002.

Implementation, testing, and evaluation of center selection methods for supervised radial basis networks to enhance breast cancer analysis

Manan Nanavati, Boris R. Bracio

Anhalt University of Applied Sciences, Koethen, Germany

manan.nanavati@live.com

Breast cancer is a chronic disease which has been classified as a cancer type having one of the highest mortality rates. An early and accurate diagnosis of any chronic disease plays an important role and can be lifesaving. Numerous research articles state that the role of computerized diagnostic tools supporting the decision making in diagnosis of chronic diseases has increased over the past decade. The presented study implements and evaluates three different artificial neural networks in form of supervised radial basis networks (RBN). The performance of the RBN's in regards to different center selection methods using different clustering algorithm are evaluated with the help of the Wisconsin breast cancer dataset (WBCD) by UCI machine learning repository.

1 Introduction

Cancer is a disease where the neoplasm or the tumor cells show uncontrolled death which leads to formation of a mass or lumps [1], [2]. Today clinicians face the challenge to screen almost 200 different types of cancer [3]. In regards to breast cancer the World Cancer fact-sheet [4] lists it as the second most diagnosed cancer across the globe along with the second highest mortality rate. The regain of a healthy state of patients in case of chronic illness depends on an early detection as well as on a proper treatment. The most common methods for breast cancer diagnosis are: (1) surgical biopsy and (2) fine needle aspiration cytology (FNAC). The physician experience and analytical skills are part of the subset for accurate diagnosis of breast cancer. One of the earliest computerized implementation for cancer diagnosis dates back to 1995 using a linear programming approach [5]. During the following decade several machine learning and data mining implementation followed. The positive results of those implementations led to an increase in usage of such computerized tools and underlined the promising nature of computer assisted diagnosis for chronic disease.

Machine learning like artificial neural networks (ANN's) and support vector machine (SVM) have been proven accurate and fast enough for the disease diagnosis. ANN is a soft computing approach, which processes the input data in an adaptive way, i.e. the designed algorithm involves "learning" from the past information. After learning, the designed ANN can be specifically used for classification of patterns or prediction or forecasting. ANN has become an accurate

method for analysis of clinical data for diagnosis purposes in the linear and non-linear domain [6], [7], [8], [9].

In this research study, RBNs were used for the analysis of breast cancer. The fast learning rate and unique design in its own class makes RBN more dominant in some of the applications than conventional multilayer perceptron networks (MLP). This study involves implementation and evaluation of three different RBN's using supervised learning methods in regards to their centers selection methods:

1. Fixed selection of centers at random [10]
2. Selection of centers using the default k-means algorithm of MATLAB – proposed method
3. Selection of centers using "customized" k-means algorithm – proposed method

The main reason behind choosing the supervised learning algorithm is the accuracy obtained in the end stage when compared to that of an unsupervised method. One of the examples representing dominance of supervised RBN over unsupervised RBN was described in [11].

The breast cancer data used for this study is the Wisconsin Breast Cancer Dataset (WBCD) from the University of California Irvin (UCI) Machine Learning Repository [12].

This paper is organized as follows: Section-2 gives the background information on radial basis networks, WBCD data and the previous work using it. Section-3 describes the methods used for this study with subsections explaining all three designs and its imple-

mentation. The results obtained are discussed in Section-4 followed by summary of this study in Section-5.

2 Background

2.1 Wisconsin breast cancer dataset:

In this research study, the Wisconsin breast cancer dataset available on [12] was used for breast cancer analysis using a RBN. The original data consists of recordings from 699 patients towards their FNAC findings which accumulates 9 different attributes on a scale of 0-10. In each of the data 10 was classified as most abnormal value and 1 as most normal value, also the class labeled for diagnosis was assigned, “2” stands for benign and “4” for malignant breast cancer. According to WBCD original data out of 699 patients, 458 patients were classified into the benign class cancer and 241 as malignant. The FNAC recorded attributes are as follows:

- Clump thickness
- Uniformity of cell size
- Uniformity of cell shape
- Marginal adhesion
- Single epithelial cell size
- Bar nucleoli
- Bland chromatin
- Normal nucleoli
- Mitoses

In the pre-processing part of the data, the original WBCD data available on [12] contains a total of 16 instances having missing attributes value, thus they have been eliminated from the database before implementing of the RBN. The output class of “4” as malignant breast cancer was changed to “1”, which states a new set of class values i.e. “1” representing malignant and “2” representing benign. The same data was also normalized from 0-1 scale using a min-max method.

2.2 Radial Basis Networks:

A radial basis function is a function whose output value depends on its distance from a center “c”.

The activation function of RBN is given as:

$$\varphi = f \|x - c\| \quad (1)$$

φ = activation function

x = input value

c = center of the radial neuron

Radial basis function networks (RBFN) possess a radial symmetric property in regards to their own centers and are a subset network of MLP's. RBFN have a different design and algorithm, it works on analysis of the data during a learning process and applying a “best fit” approach during a testing phase.

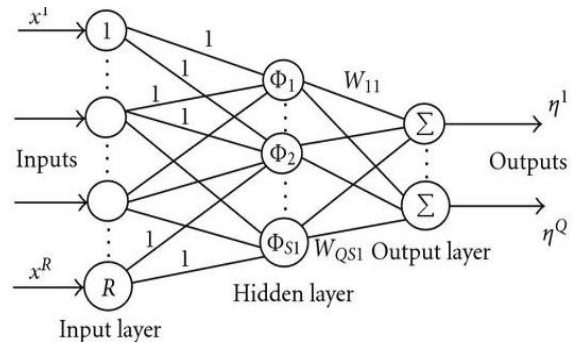


Figure 1: Basic schematic of Radial basis function network [13].

As shown in Fig. 1 the construction of a basic RBFN consist of the three different layers:

1. **Input Layer:** The layer which acts as the source node for the input data.
2. **Hidden Layer:** It is layer having high number of neurons with a basis function working as the network activation function.
3. **Output Layer:** It acts as a sink in the network, giving an output response for the given input data.

Considering all above design constraints, the output response of the radial basis network is given as:

$$y = \sum w_{ij} \cdot \varphi_i(x) \quad (2)$$

y = output response of network

φ = activation function

x = input value

c = center of radial neuron

w_{ij} = weight of output layer

For a Gaussian RBF the activation function is given for a spread value σ of the radial neuron as:

$$\varphi = e^{-\frac{\|x-c\|^2}{2\sigma^2}} \quad (3)$$

The most versatile characteristic of RBFN is that the mapping between input to hidden layer is nonlinear whereas it is linear between the hidden-output layer [10].

2.3 Previous works on WBCD data:

The breast cancer diagnosis on WBCD data performed at the University Of Sydney, Australia achieved a 94.74% accuracy using a C4.5 decision tree algorithm [14]. Researchers at University of Veszprem, Hungary have obtained an overall accuracy of 95.57% using a fuzzy clustering method [15] whereas researchers at Swiss Federal Institute of Technology, Switzerland have obtained accuracy 97.36% using a fuzzy genetic algorithm [16]. Researcher at the Tobbi University of Economics and Technology, Turkey has demonstrated a SVM machine learning approach which yielded an overall accuracy of 99.54% [17] where on other side a 100% accuracy was obtained using a Rough subset theory by SVM at University Changchun, China [18]. When focusing on RBN methods, researcher from the Indian Institute of Management and Technology, India obtained an accuracy of 49.79% [19], on the other hand researchers at the Yildiz Technical University, Turkey achieved an accuracy of 96.18% [20].

3 Methods and its Implementation

As mentioned in Section 2.2, radial basis networks are symmetric to its own center(s), and the output value depends on the distance between input value and the respective center. Thus, the determination of the center value plays a critical role in the performance of supervised RBN's. In this research study three methods regarding the selection of centers of radial neurons were implemented and their obtained results were compared to each other.

3.1 Algorithm 1: Fixed selection of centers at random

In this supervised learning method the random selection of center(s) method described in [10] was used. This was implemented in the form of a radial basis network supervised algorithm present in MATLAB neural network toolbox. The basic steps involved in this method are summarized as follows.

1. Selection of RBN architecture
2. Initialization of network
3. Training of network
4. Validation of the network
5. Test of network
6. Presentation of results

Here the WBCD data for supervised learning were randomly divided for analysis into 70% training, 15% validation and 15% test dataset.

3.2 Algorithm 2: Selection of centers by default k-means algorithm

The second RBN implementation was also done in MATLAB, but without the neural network toolbox. First a statistical test was used for the determination of a "Silhouette Index" to calculate the best possible "k" value for a k-means clustering on the WBCD data. For this study "k=2" was obtained, so two clusters of data will be implemented for this algorithm. The main aim of using the k-means algorithm in this method is to determine the cluster centers for each category input. In this method the default k-means algorithm of MATLAB to determine center values is used. It was followed by the design of a supervised RBN for the WBCD analysis. The steps involved in this method are listed as follows:

1. Determination of Silhouette index
2. Perform k-means clustering to determine the centers of input value
3. Calculation of $\frac{1}{2\sigma^2}$ term in activation function, where σ represents the spread value of radial part
4. Calculate the output activation of radial neurons over inputs
5. Determination of output weights using pseudoinverse method
6. Evaluation of RBN response
7. Presentation of results.

3.3 Algorithm 3: Selection of centers by modified k-means algorithm

This last RBN was also implemented using MATLAB without using the neural network toolbox. The main aim of using a k-means algorithmic approach in this

method was to determine the centers for the input categorical data. Furthermore, after a successful determination of a center value using a modified k-means approach, a supervised RBN was implemented. The difference between “Algorithm 2” and “Algorithm 3” is the method used to calculate the k-means. In Algorithm 2, the given algorithm of MATLAB was used, whereas in “Algorithm 3” a customized k-means approach was designed for the determination of the centers value. The methodological step number 2 to 7 presented in Algorithm 2 in Section 3.2 are same to analyze the WBCD cancer data. The following steps represent the customized k-means algorithm for determination of centers:

1. Determination of number of unique category in target for determining value for k-means
2. Selection of Centers per categorial data
3. Selection of intial centroids
4. Perform k-means over iterative loop
5. Remove, if any empty clusters present
6. Find the closest centroid to determine membership class.

The performance measure for all three implemented supervised RBN algorithms was the overall accuracy obtained in regards to the total classification rate. The overall accuracy obtained for all three supervised algorithms/methods for centers' determination were compared.

4 Results and Discussion

4.1 Results of Algorithm 1: Fixed selection of centers at random

The used RBN algorithm in this method for selection of centers is described in [10]. The fixed centers from the categorical input data at random are selected in this method. The algorithm was implemented in MATLAB 7.10 with use of neural network toolbox. The nine different categorical inputs presented in Section 2.1 were taken as the input to the RBN model. The key factor which influences the performance of RBN model is the “spread” value and centers of the given input data. Here in this method, we have selected centers randomly as mentioned and best

spread value was determined using “spread over loop” method. The following response was obtained when network was simulated with “loop over spread” method.

The results of implemented model obtained highest overall accuracy rate at spread value of **0.7** represented in Fig.2.

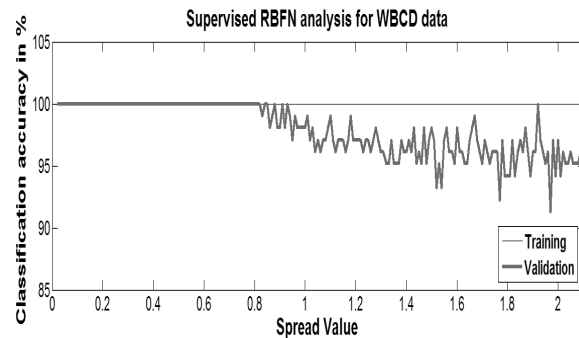


Figure 2: Classification analysis v/s spread values of RBN model

The following tables, Table: 1 and Table: 2 represent the confusion matrix obtained through Algorithm 1 during training and testing phase.

Training Dataset

Actual class	Predicted class	
	Positive (Malignant)	Negative (Benign)
Positive (Malignant) = 195	195	0
Negative (Benign) = 285	0	285

Table: 1 Results of Training phase- Algorithm 1

Test Dataset

Actual class	Predicted class	
	Positive (Malignant)	Negative (Benign)
Positive (Malignant) = 21	20	1
Negative (Benign) = 79	1	78

Table: 2 Results of Testing phase- Algorithm 1

The total accuracy for the analysis of results obtained is calculated as:

$$T.A = \frac{TP + TN}{TP + FP + FN + TN} * 100\% \quad (4)$$

TA = Total accuracy

TP = True Positive value

TN = True Negative value

FP = False Positive value

FN = False Negative value

The overall accuracy of 100% for the training and validation and 98% for the testing data set was obtained.

4.2 Results of Algorithm 2: Selection of centers using default k-means algorithm of MATLAB

In this second method, the RBN based diagnostic tool for breast cancer was also implemented and tested on MATLAB 7.10, but here the neural network toolbox was not used. The same set of input parameters as taken in Algorithm 1 were taken as end point and to compare and analyze these algorithms. But here no partition of data was done prior before giving input (i.e. 100% training dataset). Prior to the design and implementation of supervised RBN model for breast cancer two steps were done: (1) Determination of Silhouette index¹ (2) k-means clustering for determination of centers. The maximum silhouette index of “0.75” was obtained at “k=2”. Later on the obtained “k” value was adapted for default k-means algorithm. Then in the same way an implementation of the RBN model and activation function was determined based on the input values.

Finally the learning weights were determined using pseudoinverse method. The result of the same method were implemented and analyzed by calculation of the total accuracy obtained as mentioned in above method. The following table represents the results obtained through implementation of algorithm-2.

The overall accuracy of “Algorithm 2” was found to be 96.77%. Here in this method, 2 centers per categorical data were used as the outcome of the default k-means algorithm.

¹The silhouette value for input data is a measure of similarity of points within in its own cluster, compared to the other points in the cluster [21].

Training Dataset

	Number of predicted value (Benign + Malignant)
Right predicted values	661
Wrong predicted values	22

Table: 3 Results of - Algorithm 2

4.3 Algorithm 3: Selection of centers using “customized” k-means algorithm

In this last method, the RBN model for breast cancer diagnosis was also implemented on MATLAB 7.10 but here also no neural network toolbox is used. The same input dataset used in “Algorithm 1” and “Algorithm 2” is also used in this method, but likewise no partition of data as mentioned in Section 4.2 is used i.e. 100% training dataset. Prior before implementation of RBN model for diagnosis, two steps were done: (1) Determination of number of unique categories in target data (2) Perform “customized” k-means for determination of centers. The only difference between both k-means approach is the involved sub-steps of it. Here in this method, there is feature added to decide the “number of centers per category”. Also, empty clusters were also removed, which may lead to higher accuracy, was also implemented in this method. In this method, the total accuracy was recorded over different values of “number of centers per category”. The other steps like determination of activation function and determination of output weights were same as used in “Algorithm 2”. The following Table.4 describes the accuracy analysis over different values of “number of category”. Here in this method numbers of centers ranges from 1 to 10 were evaluated for accuracy analysis.

At the end of all implementation involved in this research study following few things can be summarized when all results are compared in regards to their overall accuracy. All the methods implemented in this study have achieved noticeable performance of 90% and higher accuracy in the end results obtained. One of our proposed method i.e. Algorithm 3 i.e. “Selection of centers using customized k-means algorithm” has achieved accuracy of 97.07% at 3 selected centers per category. On other side no significant difference in overall accuracy was found in higher number of centers per category. Likewise on other side, by use of sophisticated neural network toolbox gave almost same level of accuracy.

Number of centers per category	Total over-all accuracy	Number of centers per category	Total over-all accuracy
1	96.48%	6	96.92%
2	96.92%	7	96.92%
3	97.10%	8	96.92%
4	96.77%	9	96.92%
5	96.92%	10	96.92%

Table 3: Accuracy outcome obtained in Algorithm 3

The following Table 5 represents the analysis of correct predicted values and wrong predicted values for “number of selected centers = 3”.

Training Dataset

	Number of predicted value (Benign + Malignant)
Right predicted values	663
Wrong predicted values	20

Table: 3 Results for “number of selected centers = 3” (Algorithm – 3)

5 Conclusions

The main purpose of this research work was to compare performance of several centers selection methods for developing RBN models. The overall classification rate obtained at the end of this study stated that, all the implemented supervised RBN models show higher accuracy rate of 90% and above. The noticeable performance of 97.07% was obtained in last method based on “number of centers per category” choice. On the other side the other two methods i.e. basic supervised RBN algorithm i.e. method 1 showed total accuracy of 98% and method 2 with basic k-means showed total accuracy of 96.77%. When comparing all above results obtained, significant differences were found. This states that there is dominance of centers selection for performance of RBN model. The noticeable result obtained in this study also states: the algorithm without neural network toolbox showed almost the same results as that of the algorithm involving sophisticated toolbox.

Thus, in the end to summarize the research work we conclude that there is a critical role of centers determination in performance of RBN when accuracies of all methods are compared to each other, as it is clear from the definition that RBN are radially symmetric in regards to their own centers. In future sophisticated algorithms to determine centers can be developed for evaluation of RBN which further can be compared to standard supervised RBN algorithm of MATLAB which may produce noticeable results.

6 References

- [1] "Cancer Glossary". cancer.org. American Cancer Society. Retrieved September 11, 2013.
- [2] "What is cancer?". cancer.gov. National Cancer Institute. Retrieved September 11, 2013.
- [3] "How many different types of cancer are there?: Cancer Research UK: CancerHelp UK". Retrieved 11 May 2012.
- [4] International Agency for Research on Cancer and Cancer Research UK. *World Cancer Factsheet*. Cancer Research UK, London, 2014.
- [5] O. L. Mangasarian and W. H. Wolberg, *Cancer diagnosis via linear programming*, SIAM News, vol. 23, no. 5, pp. 1 - 18, 1990.
- [6] A. Cichocki and R. Unbehauen, "Neural Networks for optimisation and signal processing," J. Wiley, Sons Ltd. And B.G. Teubner, Stuttgart, 1993.
- [7] Abdelaal Ahmed, Mohamed Medhat and FarouqWaelMuhammed, "Using data mining for assessing diagnosis of breast cancer," in Proc. International multiconference on computer science and information Technology, 2010, pp. 11-17.
- [8] Burke H. B. Et al, "Artificial Neural Networks Improve the Accuracy of Cancer Survival Prediction", Cancer, 1997, vol.79, pp.857-862.
- [9] Kenneth Revett, Florin Gorunescu, Marina Gorunescu, Elia El-Darzi and Marius Ene "A Breast Cancer Diagnosis System: A Combined Approach Using Rough Sets and Probabilistic Neural Networks ", EUROCON 2005, Serbia & Montenegro, Belgrade, November, 22- 24, 2005,pp 1124-1127.
- [10] Haykin, S. (1994). *Neural networks: a comprehensive foundation*. New York: Macmillan.
- [11] Wilamowski, Bogdan M., and Karina Vieira. "Clustering of patterns using radial base

- function networks.*" In Artificial Neural Networks in Engineering ANNIE'95, pp. 109-115. 1995.
- [12] UCI Repository of Machine Learning Databases. www.archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/
- [13] M. N. Ab Malek and M. S. Mohamed Ali, "Evolutionary Tuning Method for PID Controller Parameters of a Cruise Control System Using Metamodeling," *Modelling and Simulation in Engineering*, vol. 2009, Article ID 234529, 8 pages, 2009. doi:10.1155/2009/234529
- [14] J.R. Quinlan, Improved use of continuous attributes in C4.5, *J. Artif. Intell. Res.* 4 (1996) 77–90.
- [15] J. Abonyi, F. Szeifert, *Supervised fuzzy clustering for the identification of fuzzy classifiers*, *Pattern Recognition Lett.* 24 (2003) 2195–2207.
- [16] C.A. Pena-Reyes, M. Sipper, *A fuzzy-genetic approach to breast cancer diagnosis*, *Artif. Intell. Med.* 17 (1999) 131–155.
- [17] Übeyli, Elif Derya. "Implementing automated diagnostic systems for breast cancer detection." *Expert Systems with Applications* 33, no. 4 (2007): 1054-1062.
- [18] Chen, Hui-Ling, Bo Yang, Jie Liu, and Da-You Liu. "A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis." *Expert Systems with Applications* 38, no. 7 (2011): 9014-9022.
- [19] Janghel, R. R., Anupam Shukla, Ritu Tiwari, and Rahul Kala. "Breast cancer diagnosis using artificial neural network models." In *Information Sciences and Interaction Sciences (ICIS)*, 2010 3rd International Conference on, pp. 89-94. IEEE, 2010.
- [20] Kiyan, Tüba, and Tülay Yildirim. "Breast cancer diagnosis using statistical neural networks." *IU-Journal of Electrical & Electronics Engineering* 4, no. 2 (2011): 1149-1153.
- [21] Kaufman L., and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ: John Wiley & Sons, Inc., 1990.

