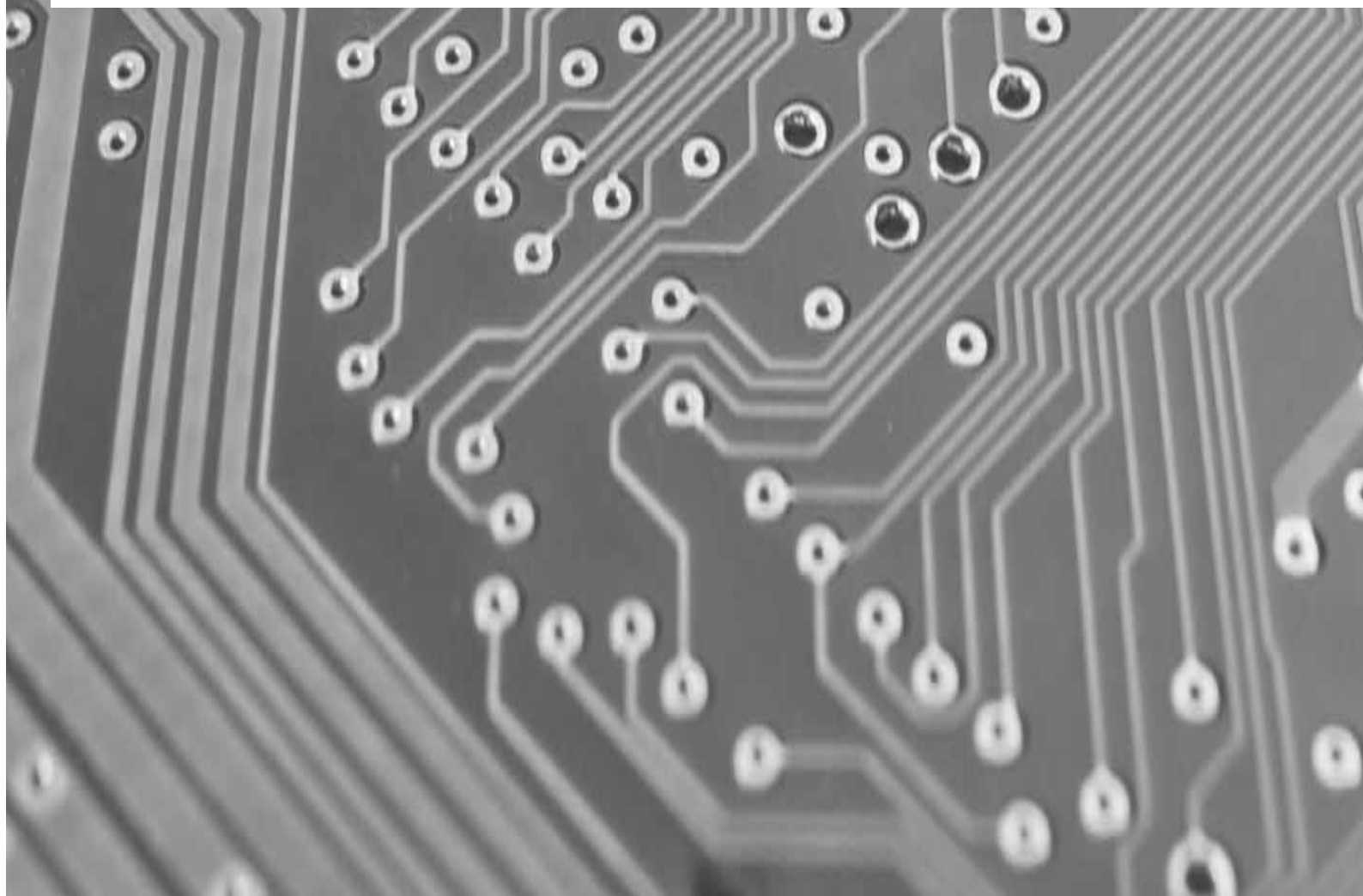


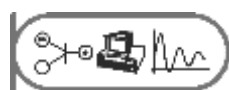
Tagungsband

ASIM 2016 - 23. Symposium Simulationstechnik

Zusammenfassung der Beiträge
Thomas Wiedemann (Hrsg.)

**7. - 9. September 2016
HTW Dresden**





ARGESIM Reports

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

Series Editor:

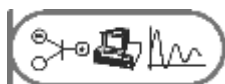
Felix Breitenecker (ARGESIM / ASIM)
Div. Simulation, Vienna University of Technology
Wiedner Hauptstrasse 8 - 10, A - 1040 Vienna
Tel: +43-1-58801-10115, Fax: +43-1-58801-10199
Email: Felix.Breitenecker@tuwien.ac.at

ARGESIM Report 52 ASIM Mitteilung AM 160

Titel: Tagungsband
ASIM 2016
23. Symposium Simulationstechnik

Herausgeber: Thomas Wiedemann
HTW Dresden F.-List-Platz 1
01069 Dresden
Email: wiedem@informatik.htw-dresden.de

ISBN print 978-3-901608-49-0
ISBN ebook 978-3-901608-89-6
DOI 10.11128/ arep.52



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

© by ARGESIM / ASIM, Wien, 2016

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

Druck:

Manufaktur für Gestaltung und Druck
Wiener Str.76 01219 Dresden

Tagungsband

ASIM 2016

23. Symposium Simulationstechnik

7. bis 9. September 2016
HTW Dresden

Zusammenfassung der Beiträge
Thomas Wiedemann (Hrsg.)



Arbeitsgemeinschaft Simulation ASIM in der Gesellschaft für Informatik GI



Hochschule für
Technik und Wirtschaft
Dresden
University of Applied Sciences

Tagungsleitung:

Thomas Wiedemann, HTW Berlin

Programmkomitee:

Fernando Barros, Universidade de Coimbra

Felix Breitenecker, TU Wien

Walter Commerell, Hochschule Ulm

Christina Deatcu, Hochschule Wismar

Umut Durak, DLR, Braunschweig

Joachim Fischer, Humboldt-Universität Berlin

Leo Gall, LTX Simulation GmbH

Torsten Gerlach, DR, Braunschweig

Joachim Haase, Fraunhofer IIS/EAS Dresden

Daniel Lückerath, Universität zu Köln

Heinz-Theo Mammen, Hella KGaA Hueck & Co., Lippstadt

Werner Maurer, ZHAW, Winterthur

Gottfried Mayer, BMW, München

Mike Müller, hydrocomputing GmbH & Co. KG, Leipzig

Torsten Munkelt, HTW Dresden

Katharina Nöh, Forschungszentrum Jülich

Klaus Panreck, Fachhochschule Bielefeld

Thorsten Pawletta, Hochschule Wismar

Nikolas Popper, dwh GmbH, Wien

Markus Rabe, TU Dortmund

Dirk Reichelt, HTW Dresden

Oliver Rose, Universität der Bundeswehr München

Thomas Schramm, HafenCity Universität Hamburg

Dirk Steinhauer, Flensburger Schiffbau-Gesellschaft mbH & Co. KG

Michael Striebel, ZF Lenksysteme GmbH, Schwäbisch Gmünd

Gabriel Wainer, Carleton University Ottawa

Siegfried Wassertheurer, AIT Austrian Institute of Technology

Jochen Wittmann, HTW Berlin

Sigrid Wenzel, Universität Kassel

Tagungsort:

Hochschule für Technik und Wirtschaft Dresden

Friedrich-List-Platz 1 01069 Dresden

Inhaltsverzeichnis

Produktion und Logistik (Production and Logistics)

Verbessertes Risikomanagement durch ein simulationsgestütztes logistikintegriertes Projektmanagement im Anlagenbau <i>Christoph Laroque and Sigrid Wenzel</i>	1
Modulares Modellkonzept für die planungsbegleitende Simulation von Luftfrachtterminals <i>Karsten Turek, Armin Siegel and Thorsten Schmidt</i>	9
Modellierung von Arbeit an bewegten Objekten in kollaborativen Betriebsformen <i>Titanilla Komenda and Felix Breitenecker</i>	17
A model factory in augmented reality as an eye-catcher at exhibitions and fairs <i>Torsten Munkelt, Steven Behne, Markus Wacker and Sven Völker</i>	25
Iterative Entwicklung und Verbesserung eines dynamischen Prozesssimulationsmodells eines Aluminiumschachtschmelzofens anhand globaler Prozessgrößen <i>Matthias Henninger and Wolfgang Schlüter</i>	33
Combining Capacity Planning with Simulation <i>Paul Barthel, Tobias Eckert and Ralf Sprenger</i>	41
Simulative Untersuchung energie- und ressourceneffizienter Betriebsstrategien in der E ³ -Forschungsfabrik <i>Johannes Stoldt, Marc Münnich, Enrico Franz, Andreas Schlegel and Matthias Putz</i>	45
Rechenbeispiel der Makro- und Mikroperioden-Bestellmengenplanung in Lebensmittelfilialen zwecks Abfallreduktion <i>Larissa Janssen, Thorsten Claus and Jürgen Sauer</i>	53

Methoden (Methods)

Autokorrelation: Die große Unbekannte in der Systemplanung und – analyse <i>Sebastian Rank, Frank Schulze and Thorsten Schmidt</i>	61
Simulation sozioökonomischer Prozesse mit zellulären Automaten – räumliche Analyse des Kundenverhaltens konkurrierender Apotheken <i>Thomas Wiedemann, Anja Voß-Böhme and Marcel Stasik</i>	67
An Explicit Approach for Asynchronous Step Size Control in Co- Simulation <i>Wolfgang Müller and Felix Breitenecker</i>	75
Automatic Layout of Scilab/Xcos Diagrams <i>Chenfeng Zhu, Umut Durak, Sven Hartmann and Clément David</i>	81
A new approach for integrating discrete element method into component- oriented system simulations <i>Christian Richter</i>	91

Komplexität beim Modellieren und Simulieren: Eine Analyse und ein Plädoyer für schlanke Modelle <i>Jochen Wittmann</i>	99
Eine Machbarkeitsstudie zur modellbasierten Beurteilung stadtklimatischer Indikatoren auf der Basis von Bebauungsplänen im GIS <i>Jochen Wittmann</i>	107
Optimierung/Scheduling (Optimization/Scheduling)	
Vergleich der Wirksamkeit von Steuerungsregeln zur Distribution von Flüssigaluminium in Nicht Eisen-Schmelz- und Druckgussbetrieben durch eine hybride Fertigungssimulation <i>Dominik Jeckle, Wolfgang Schlüter and Ansgar Ringleb</i>	113
OptPlanEnergie - AN OPTIMIZATION AND SCHEDULING PLATFORM FOR THE ENERGY-EFFICIENT PRODUCTION OF TEMPERED GLASS <i>Frank Baumann, Heike Wilson, Stephan Seidel, Matthias Franke and Ulrike Gromnitza</i>	121
Web-basierte Simulation zur Optimierung von Logistikstrukturen <i>Mathias Bös</i>	125
A Framework for the Metamodeling of Multi-variant Systems and Reactive Simulation Model Generation and Execution <i>Thorsten Pawletta, Artur Schmidt, Umut Durak and Bernard P. Zeigler</i>	131
Verkehrssimulation (Traffic simulation)	
A Framework for Parametric Aircraft, Engine and Sub-Systems Models <i>Michael Sielemann, Olga Silantyeva and Majed Sammak</i>	139
Behaviour of Rotor Blade with Double-Sweep in a Whirl Tower <i>M Rohin Kumar and Sriram Palika</i>	151
Model-based Development of Enhanced Ground Proximity Warning System for Heterogeneous Multi-Core Architectures <i>Umut Durak, David Müller, Jürgen Becker, Nikolaos S. Voros, Panayiotis Alefragis, Timo Stripf, Pierre-aimé Agnel, Gerard Rauwerda and Kim Sunesen</i>	159
Numerische Lösung eines mathematischen Modells für eine optimale Krebskombinationstherapie aus Anti-Angiogenese und Strahlentherapie <i>Kurt Chudej, Dominik Huebner and Hans Josef Pesch</i>	169
Simulation von Stützprozessen und Logistik in der Klinikplanung <i>Carsten Matysczok, Markus Kühn, Moritz Schele and Gime Batija</i>	177
Automatic classification of heartbeats in the electrocardiogram using support vector machines and self-organizing maps <i>Natalie Gschaidler, Martin Bachler, Christopher Mayer and Siegfried Wassertheurer</i>	185

Technische Systeme (Technical System)

Design, Simulation and Operation of Task-oriented Multi-Robot Applications with MATLAB/Stateflow <i>Birger Freymann, Sven Pawletta, Artur Schmidt and Thorsten Pawletta</i>	193
Simulation von Strahlströmungen mittels des SST-Turbulenzmodells <i>Ansgar Ringleb, Wolfgang Schlüter, Günter Wozniak and Oliver Sommer</i>	201
Towards Safe Robotics - Modellbasierte Entwicklung von High Integrity Robotern <i>Maximilian Apfelbeck, Manuel Fédou and Stephan Myschik</i>	207
Verteilte Simulation mehrachsiger magnetischer Positionssensorsysteme <i>Jörg Bretschneider, Xuehao Wang and Neudeck Willi</i>	215

Verteilt/Parallel (Distributed/Parallel)

Parallel multi-agent smart grid simulation <i>Christian Kuschel and Ulrich Rüde</i>	223
VOMAS for Validation of Agent-based Models - Requirements and Application <i>Julian Ruths, Niki Popper and Florian Miksch</i>	231
Calibration Strategies for Agent-based Simulation Models with Variability <i>Claire Rippinger, Martin Bicher and Florian Miksch</i>	239

Lehre (Education)

Modellierungsansätze und Fallbeispiele in der Ausbildung für Modellbildung und Simulation <i>Andreas Körner, Stefanie Nadine Winkler and Felix Breitenecker</i>	247
Vergleich von Blended Learning Tools für Modellbildung und Simulation <i>Stefanie Winkler, Martin Bicher and Andreas Körner</i>	249
Integration of Simulation-based Training for Welders <i>Benjamin Knoke and Klaus-Dieter Thoben</i>	253
Physical Simulation Related Exercises for the Education in the STEM Field – Approaches Based on the Physolator Framework <i>Dirk Eisenbiegler, Dietmar Gruber and Thomas Jörg</i>	261
Modellierung und Simulation der langfristigen Bildungspolitik in Deutschland und Europa <i>Thomas Wiedemann</i>	269

Vorwort

Die Anwendung von Modellierung und Simulation als allgemeine Methodik zur Planung, Analyse und Optimierung von komplexen Systemen hat sich in den vergangenen Jahren zu einer etablierten Technik zur Problemlösung und Entscheidungsfindung entwickelt. In immer mehr wissenschaftlichen Publikationen und Projekten werden Simulationen als das Mittel zur genauen Analyse und darauf aufsetzenden Synthese neuer Lösungen erfolgreich eingesetzt.

Bei einer kritischen Betrachtung der Relevanz der eingesetzten Modelle und Simulationen für den jeweiligen Anwendungsfall fallen trotz des prinzipiell sehr erfolgreichen Entwicklungsstandes deutliche Unterschiede in verschiedenen Bereichen auf.

Im Bereich der Simulation technischer Systeme sind Simulationsmodelle unabdingbar für den Erfolg. Ich wage sogar zu behaupten, dass ein unberechtigter Abgriff eines Simulationsmodells eines für 2018 angekündigten Prozessors oder Autos kritischer ist als die Konstruktionszeichnung oder das Auto selbst, weil im Simulationsmodell auch alle internen, sonst unsichtbaren Abhängigkeiten dokumentiert sind.

Im Bereich der Produktionsplanung und Logistik haben sich Simulationsmodelle ebenfalls dauerhaft etabliert. Es ist dabei zu unterscheiden zwischen einmaligen Simulationen von Investitionen und einer kontinuierlichen Simulation im operativen Betrieb. Professionelle Logistiktechnikanbieter gehen keinen Vertrag mehr ein, ohne dass das Großregallager nicht genau bzgl. seiner maximalen Auslieferkapazität berechnet und nachgewiesen ist. Bei der kontinuierlichen Planung können durch ständigen Abgleich mit dem Istzustand Genauigkeiten der Voraussagen von weniger als 1% Abweichung generiert werden. In beiden Fällen schaffen diese Ergebnisse Vertrauen und erzeugen auch die Akzeptanz bei der Geschäftsführung bzgl. der notwendigen Ausgaben.

In gleicher Weise sind auch in den Bereichen der medizinischen Versorgung, der Verkehrsplanung und der Umwelt- und Geowissenschaften zunehmend sehr effektive Simulationsanwendungen zu beobachten und auch in diesem Tagungsband dokumentiert.

Aus meiner persönlichen Sicht und aus einigen Berichten von internationalen Simulationskollegen sieht dies im Bereich großer Modelle zur Vorhersage von globalen Entwicklungen im politischen und finanztechnischen Bereich leider noch häufig anders aus.

Trotz korrekter Modelle und wahrscheinlich auch valider Simulationsergebnissen werden im politischen Alltagsgeschäft Simulationsergebnisse kaum als entscheidungsunterstützende Mittel eingesetzt. In einem Fall wurden sogar kritische Simulationsergebnisse zu neuen Steuerplänen bewusst ignoriert und einige Jahre später wundert man sich über leere Kassen und zusammenbrechende Infrastrukturen.

Die Ursachen für diese geringe Akzeptanz im politischen Bereich liegen aus meiner Sicht in zwei Punkten. Zuerst werden im politischen Bereich meist Einmalereignisse wie z.B. der Börsencrash 2008 als besonders relevant wahrgenommen und derartige „Schwarze Schwäne“ lassen sich zwar prinzipiell, aber nicht tagesgenau vorhersagen, da sie stark von menschlichen Akteuren abhängen. Und die Modellierung von menschlichem Entscheidungsverhalten ist schwierig, trotz aller Fortschritte bei Multi-Agentensystemen und neuronalen Netzwerken.

Der zweite Problempunkt ist die Akzeptanz der Simulationsmethodik an sich. Ohne Politiker pauschal beurteilen zu wollen, sind Simulationsmodelle in deren Ausbildungslauf eher unterrepräsentiert. Auch ältere Entscheidungsträger in großen Wirtschaftseinheiten haben in ihrem Jahrzehnte zurückliegenden Studium natürlich noch nicht den heutigen Stand der Simulationstechnik erfahren können. Aufgrund dieser beobachteten Tatsachen und auch der vor uns stehenden großen europäischen Aufgaben sehe ich einen großen Bedarf an einer berufsbegleitenden Schulung von Entscheidungsträgern aus Wirtschaft, Politik und Gesellschaft zu modernen Methoden der Modellierung und Simulation komplexer Systeme.

Es ist unter diesem Gesichtspunkt sehr positiv, dass auf dem 23. ASIM Symposium zur Simulationstechnik auch eine Sitzung der neu gegründeten Fachgruppe zum Einsatz der Simulation in und für die Lehre stattfinden wird. Ich wünsche den Kollegen viel Erfolg bei der Diskussion neuer Methoden bei der Ausbildung mit und im Simulationsbereich, sowohl in wie außerhalb der Universitäten.

Natürlich wünsche ich auch allen anderen teilnehmenden Fachkollegen eine überaus interessante Konferenz und viele neue Erkenntnisse zum erfolgreichen Einsatz der Simulationstechnik in den nächsten, sicher sehr herausfordernden Jahren.

Ihr Thomas Wiedemann

Dresden im September 2016

Verbessertes Risikomanagement durch ein simulationsgestütztes logistikintegriertes Projektmanagement im Anlagenbau

Sigrid Wenzel¹, Christoph Laroque¹

¹ Universität Kassel, Kassel (Germany)

² Westsächsische Hochschule Zwickau, Zwickau (Germany)

christoph.laroque@fh-zwickau.de

Der deutsche Maschinen- und Anlagenbau ist bekannt für innovative und hochwertige Lösungen. Neben technischem Know-how zählt vor allem die termingerechte Inbetriebnahme des Produktes. Mit der heute immer noch gängigen Praxis, Unsicherheiten hauptsächlich durch zusätzliche Pufferzeiten zu berücksichtigen, verlieren kleine und mittlere Unternehmen (KMU) jedoch wichtige Punkte im globalen Wettbewerb um Aufträge. Ansatz des hier beschriebenen Forschungsprojektes „simject“ war es deshalb, die ohnehin vorhandenen Daten der herkömmlichen Planungs- und Projektmanagementtools durch die Anwendung von Simulations-, Optimierungs-, Analyse- und Visualisierungsverfahren so zu ergänzen, dass die Pläne auch unter zufälligen Einflüssen getestet und validiert werden. Da somit die existierenden Unsicherheiten der Realisierungsphase bereits in der Planungsphase simuliert werden, stehen als Basis für die Realisierung mit Wahrscheinlichkeiten hinterlegte Zeitpunkte oder Zielkorridore zur Verfügung. Diese Daten dienen zur Verbesserung des Risikomanagements im Projekt und werden fortlaufend aktualisiert.¹

1 Einleitung

Der kundenindividuelle Anlagenbau unterscheidet sich in der betrieblichen Praxis deutlich von der stationären Serienfertigung (vgl. [1] und [2]). Die Planung eines derartigen Projektes ist hier „in Abhängigkeit von den systemtechnischen und konstruktiven Randbedingungen der Anlage, den lokalen Standortgegebenheiten, den organisatorischen Projektvorgaben (z. B. Bauabschnitte, Produktionsschritte oder Ressourcendisposition) sowie den damit verbundenen logistischen Restriktionen durchzuführen“ [1]. Zur Absicherung der logistischen Restriktionen und Prozesse wird bei der Realisierung von kundenindividuellen Einmalprodukten in einzelnen Branchen wie dem Bauwesen (vgl. [3], [4], [5], [6], [7]) und dem Schiffbau (vgl. [8], [9], [10], [11]) bereits die ereignisdiskrete Simulation eingesetzt. Zudem finden sich erste Werkzeuge zum Einsatz der Simulation im Projektmanagement (vgl. [12] und [13]). Allerdings wird eine Betrachtung logistischer Fragestellungen in Kombination mit der Projektplanung nicht umgesetzt.

Das gemeinsam von den Universitäten Kassel und Paderborn durchgeführte Forschungsprojekt „simject“²

(17725 N) der Forschungsvereinigung Bundesvereinigung Logistik (BVL) widmete sich in der Zeit vom 01.04.2013 bis zum 31.03.2015 dieser Verknüpfung und erarbeitete eine Lösungskonzept für ein simulationsgestütztes logistikintegriertes Projektmanagement. Mit diesem Konzept soll die zeitliche Machbarkeit einer Baumaßnahme und die Robustheit der Projektpläne unter Berücksichtigung aller kundenindividuellen Restriktionen auch bei unvermeidbaren Störungen (z. B. aufgrund von fehlendem Material) sichergestellt und durch den Einsatz verschiedener Simulations- und Optimierungsmethoden das Risikomanagement innerhalb dieser Projekte verbessert werden. Das Vorhaben wurde neben Unternehmen der Automobilindustrie und des Schiffbaus vor allem von Partnern aus den Bereichen Energie-, Kraftwerk- und Umwelttechnik über einen projektbegleitenden Ausschuss (PA) unterstützt. Diese Partner unterscheiden sich nicht nur hinsichtlich ihrer Unternehmensgröße, sondern bedingt durch ihre Produkte (von Windkraftanlage über Biogasanlage bis zum Kraftwerk) auch hinsichtlich ihrer Projektstrukturen.

¹ Dieser Beitrag stellt wesentliche Erkenntnisse des Forschungsvorhabens als Auszug aus dem Forschungsendbericht [14] zusammen.

² Gefördert über die AiF im Rahmen des Programms zur Förderung der Industriellen Gemeinschaftsforschung (IGF) vom Bundesministerium für Wirtschaft und Technologie aufgrund eines Beschlusses des Deutschen Bundestages

2 Projektablauf

Der erste Arbeitsschritt im Projekt diente der Aufnahme von KMU-spezifischen Prozessen, Vorgehensweisen und Methoden im Anlagenbau. Hierzu wurden zu Beginn des Projektes im Rahmen einer ersten Sitzung mit den Industrievertretern die Projektbesonderheiten im Anlagenbau in Bezug auf Planung und Logistik bei den beteiligten Unternehmen zunächst diskutiert und nach Analyse einschlägiger Literatur ergänzt. In den Gesprächen mit den beteiligten Unternehmen während dieser Sitzung stellte sich heraus, dass aufgrund der unternehmensspezifischen Projektbelastung durch Interviewleitfäden vorbereitete Einzelinterviews gewünscht waren. Aus diesem Grund wurde ein Interviewleitfaden so ausgearbeitet, dass er auch zur Erfassung von Anforderungen genutzt werden konnte. Dies erlaubte zum einen, vertrauliche Unternehmensinterna an die Forschungsstellen, nicht aber an den PA zu kommunizieren, zum anderen konnte die Belastung der Unternehmen für Einzelinterviews konzentriert werden. Um die Bandbreite möglicher Entwicklungen aufzuzeigen, verdeutlichten Partnerunternehmen ihre unternehmensspezifischen Entwicklungen im Bereich des Projektmanagements. Aus den Erkenntnissen der Ist-Aufnahme wurde ein neutrales Referenzmodell für das Projektmanagement des Anlagenbaus abgeleitet und in einer zweiten Sitzung vorgestellt, diskutiert und ergänzt. Die Diskussion unternehmensspezifischer Referenzmodelle wurde grundsätzlich nur in den Interviews geführt; Erkenntnisse aber in das neutrale Referenzmodell überführt.

simject-User Stories		Version 1.0 Stand: 26.08.2014	
lfd. Nr.	Name der "Story"	User Story	zu LH-Nr.
1	Projektverwaltung	Als Anwender möchte ich unbedingt eine zentrale Projektverwaltung, die Projektname, Properties, Verweise auf zugehörige Dateien enthält, um effektives Arbeiten zu ermöglichen.	ABE001
2	Suche nach Projekten	Als Anwender will ich eine einfache Suche nach Worten wie Name, Ersteller, Keywords nutzen, um effektives Arbeiten zu ermöglichen.	ABE002
3	Andere MS-Produkte nutzen	Als Anwender will ich den Datenimport aus anderen MS-Produkten nutzen, um flexibel zu bleiben.	ABE003
4	Projektpläne editieren	Als Anwender will ich die Korrektur von Start-/Endzeitpunkt, Nachführen gem. FSP, SEP usw. durchführen, um flexibel zu bleiben.	ABE004
5	Ist-Dateneingabe	Als Anwender will ich die Eingabe der Ist-Dauer durchführen, um Medienbrüche zu vermeiden.	ABE005
6	Projektberichte anlegen	Als Anwender möchte ich Fenster mit Text-Templates für Projektberichte, Aufgabenverteilung zu Plänen (Kommentare) haben, um Medienbrüche zu vermeiden.	ABE006
7	Kommentare anlegen	Als Anwender möchte ich Fenster mit Text-Templates für Kommunikationsprotokolle zu Plänen (Kommentare) haben, um Medienbrüche zu vermeiden.	ABE007
8	Einsatzpläne einsehen	Als Anwender möchte ich Einsicht von Einsatzplänen von	

Abbildung 1: Ausschnitt der Anforderungsliste

Im zweiten Arbeitsschritt erfolgte unter Nutzung des erwähnten Interviewleitfadens die Erhebung von technischen, funktionalen und organisatorischen Anforderungen an ein Projektmanagement; die gewählte Vorgehensweise entsprach der des ersten Arbeitsschrittes. Neben den im Fokus des Projektes stehenden Anforderungen wurden auch über den inhaltlichen Projektkern hinausgehende Wünsche diskutiert. Alle Anforderungen wurden in Form von User-Stories beschrieben und priorisiert (vgl. Abbildung 1). Zusätzlich wurde eine umfassende Analyse bestehender Forschungsarbeiten und wissenschaftlicher Ergebnisse Dritter durchgeführt (vgl. auch [1]). In diesem Zusammenhang wurden auch Ähnlichkeiten mit dem klassischen Hoch- und Tiefbau geprüft. Die abschließende Diskussion der Ergebnisse erfolgte auf der dritten PA-Sitzung im März 2014.

Basierend auf den Ergebnissen der ersten beiden Arbeitsschritte wurde dann in den nachfolgenden Arbeitsschritten mit der Konzeption einzelner Lösungsansätze für Logistiks simulation, Projektplanerstellung, -simulation und -optimierung sowie 3D-Visualisierung des Projektfortschrittes begonnen. Hierbei wurden zunächst die einzelnen Fragestellungen zur durchgängigen Nutzung einer Plattform aus den PA-Sitzungen aufgegriffen, um anhand des Standardreferenzszenarios Ansätze zu deren Lösung zu entwickeln.

Das Hauptaugenmerk lag auf einer schnellen Entwicklung von Teillösungsansätzen, anhand derer eine mögliche Integration in eine Gesamtumgebung sowie daraus abgeleitet die Formulierung von Schnittstellen diskutiert wurde. Diese Teillösungen dienten als Kommunikationsbasis gegenüber den PA-Mitgliedern, um deren Anregungen in die eigentliche Gesamtlösung zeitnah aufzunehmen. Bei der Bearbeitung zeigte sich aufgrund des gewählten agilen Entwicklungsprozesses in enger Zusammenarbeit auch mit den Industrievertretern, dass die Einbindung der Lösungsansätze für die Teilaufgaben in einer Gesamtarchitektur auch durch die Gesamtarchitektur selbst bestimmt wird. Die konzipierte grobe Gesamtarchitektur wurde nachfolgend weiter detailliert und in Bezug auf die seitens der Firma SimPlan AG zur Verfügung gestellte Entwicklungsplattform SimAssist zu einem abgestimmten Gesamtkonzept erweitert, so dass projektspezifische Plug-ins sukzessive eingebunden werden konnten (siehe auch Abbildung 4).

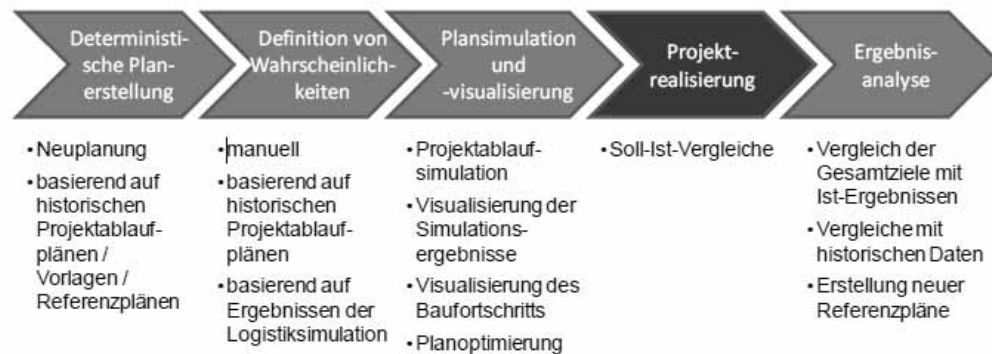


Abbildung 2: Hauptprozesse

Anschließend wurden Teile des Lösungskonzeptes in Form eines ablauffähigen Demonstrators implementiert. Hierzu zählten Plug-ins zum Importieren vorhandener Projektpläne aus der Standardsoftware MS Projekt, zur Simulation von Projektplänen, zur heuristischen Optimierung von Projektplänen sowie zur Kopplung verschiedener Simulationsmodelle innerhalb eines Planungsprojektes. Parallel zur Realisierung einzelner Plug-ins wurde ein abgestimmtes Daten- und Informationsmodell entwickelt, so dass die in den verschiedenen Teilsystemen benötigten Daten möglichst gut auf einer gemeinsamen Ontologie aufbauen und relevante Planungsinformationen an genau einer Stelle im integrierten Datenmodell verwendet werden.

Im Anschluss wurden einzelne, konkrete Planungsmodelle für Windkraftanlagen sowie Windkraftanlagenparks umgesetzt und validiert, um zum einen den Demonstrator zu testen, zum anderen die Durchgängigkeit der Entwicklung an einem praxisnahen Beispiel aus dem Anlagenbau aufzuzeigen. Bei diesen Beispielen konnten einzelne Bauprojekte, aber auch Multiprojektpläne mit der entwickelten Vorgehensweise bewertet werden. Für die Kopplung verschiedenartiger Simulationsmodelle innerhalb einer Anlagenplanung ist besonders die Frage interessant, inwieweit die Simulation eines Logistikszenarios (diskret, ereignisgesteuertes Simulationsmodells) die Generierung einer Plansimulation (Monte-Carlo Simulation eines Projektplans) unterstützen kann. Hierzu wurde für die Montage einer Windenergieanlage ein Modell erstellt, das ausgehend von einem deterministischen Projektplan als Eingabedatei alle logistikrelevanten Prozesse bis hin zu Transport und Montage der Bauteile diskret simuliert. Die hierbei durch Wechselwirkungen zwischen den Logistikprozessen entstehenden Abweichungen von der Planvorgabe werden in den zu simulierenden Projektplan als Verteilungsfunktionen zurückgespielt, so dass

Daten für dessen Simulation zur Verfügung stehen (Verteilungsidentifikation für die Kennzahlen des Logistik-Simulationsmodells als stochastischen Output).

Die Evaluation der Projektergebnisse wurde als zweistufiger Prozess umgesetzt. Auf der 5. PA-Sitzung erfolgte zunächst die Evaluation des Nutzungsprozesses des Lösungskonzeptes. Dieser wurde gemeinsam von den beiden Forschungsstellen in einer ausführlichen Präsentation vorgestellt und diskutiert; Anmerkungen seitens der Industrievertreter flossen direkt in die Überarbeitung ein. In einem zweiten Schritt wurden die Ansätze auf Basis des Demonstrators intensiv evaluiert. Hierzu wurde zunächst ein Evaluationsleitfaden entwickelt, der als Basis für die Evaluation im Rahmen des PA-Meetings sowie individueller Evaluationsgespräche bei den Industrievertretern diente. Der Leitfaden bezog sich insbesondere auf die Überprüfung der Einsetzbarkeit für KMU in Bezug auf Plausibilität, Anwendbarkeit und Relevanz der Plug-ins.

3 Projektergebnisse

Ein erstes wichtiges Projektergebnis ist das Projektreferenzmodell, das alle wesentlichen Arbeitsschritte der unternehmensindividuellen Vorgehensweisen in einem neutralen Prozessmodell vereint. Als ein Beispielerzeugnis wurde im Rahmen des Projektes ein fiktiver Fluxkompensator verwendet, um die Produktneutralität im Konsortium sicherzustellen.

Das Prozessreferenzmodell umfasst insbesondere auch die logistikrelevanten Prozesse sowie ein Evaluations-szenario, das die unterschiedlichen Facetten der beteiligten Unternehmen enthält. In einer ergänzenden Arbeit wurde zudem ein Referenzprozess mit Hilfe von BPMN 2.0 (Business Process Model and Notation) modelliert, um aus einer Identifikation und Analyse von für ein Planungsprojekt im Anlagenbau charakteristischer Risiken Erkenntnisse zu gewinnen, die in der

Praxis eine störungsfreie Durchführung eines Entwicklungsprojektes ermöglichen (Schnittstellen zwischen Prozessverantwortlichkeiten, Abhängigkeiten von Logistik, Verkehr, Wetter, etc.).

Ein weiteres Projektergebnis ist die entwickelte Vorgehensweise zur Unterstützung des Projektmanagements. Ihr Grundanliegen ist es, die im Planungsprozess enthaltenen Risiken besser zu berücksichtigen. Hierzu wird das klassische Vorgehen der Planung um Teilschritte ergänzt, in denen einzelnen Risikoparametern in Planung und Steuerung eines Projektes möglichst adäquate, stochastische Wahrscheinlichkeitsverteilungen zugeordnet werden. Datenquellen dieser Wahrscheinlichkeitsverteilungen können sowohl historische Projektdaten, Einschätzungen des Planers als auch Ergebnisse aus detaillierteren Planungsmodellen sein, beispielsweise einer Logistiksimulation. Die heute deterministische Planung wird dann durch verschiedene Simulations-, Optimierungs- und Visualisierungsmethoden unterstützt, die diese stochastischen Einflüsse berücksichtigen und das jeweilige Planungsergebnis um die zusätzlich gewonnenen Informationen ergänzen. Im Ergebnis erhält der Planer eine realistischere Einschätzung der in seiner Planung enthaltenen Projektrisiken und deren Auswirkungen auf die Gesamtplanung.

Da ein simulationsgestütztes Projektmanagement mit integrierter Betrachtung der Logistik den erweiterten Planungsprozess als Ganzes unterstützen muss, werden in dem zugrunde gelegten erweiterten Planungsprozess fünf Hauptprozesse festgelegt und über ihre elementaren Funktionen beschrieben (Abbildung 2). Jeder Hauptprozess ist mittels der Prozessbeschreibungssprache BPMN modelliert, so dass die einzelnen Funktionen in einen logischen Ablauf gebracht sowie die jeweilige Unterstützung des Prozesses beschrieben werden. Basierend auf den abgeleiteten Arbeitsprozessen liegt ein konkreter Architekturentwurf mit konzeptioneller Einbindung unterschiedliche Werkzeuge vor, um die einzelnen Teilaufgaben zu bewältigen. Diese umfassen im Wesentlichen folgende Funktionen:

- Schnittstellen zu externen Datenquellen (Zeitpläne, Daten aus PDM- und ERP-Systemen, GIS und Wetterdaten)
- Projektbearbeitung (Editor zum Verwalten und Ändern von Projektplänen)
- Logistiksimulation (ereignisdiskreter Simulator für Logistikmodelle)

- Projektsimulation (Simulator für Projektpläne)
- Projektoptimierung (Optimierungshilfen für Projektpläne)
- Visualisierung (Planvisualisierung, 2D-/3D-Baufortschrittsvisualisierung)

Die jeweils zu nutzenden Funktionen variieren dabei in Abhängigkeit von der gestellten Planungsaufgabe, so dass nicht von einem fest vorgegebenen Planungsablauf im Detail ausgegangen werden kann. Somit sind die Funktionen und die hierzu zu nutzenden Werkzeuge für die jeweilige Planungsaufgabe geeignet zu orchestrieren. So können beispielsweise Simulationsläufe eines Projektplans ohne zugehörige, detaillierte Betrachtung der Logistikprozesse erfolgen oder aber mit einem detaillierten Modell der Logistik dynamisch gekoppelt werden. Abbildung 4 zeigt einen beispielhaften Nutzungsablauf der einzelnen Funktionsmodule.

Eines der spezifisch entwickelten Plug-ins erzeugt auf Basis historischer Prozessdaten in den Unternehmen eine sogenannte Referenzdatenbank, um Projekterfahrungen bei neu zu planenden Projekten zur Verfügung zu stellen. Die Implementierung der Referenzdatenbank erfolgte direkt in MS Project 2013. Der Benutzer hat die Möglichkeit, einen Projektplan direkt aus der Anwendung in die Datenbank abzulegen. Der Projektplan wird über einen Namen, eine Versionsnummer und eine Beschreibung des Projektes charakterisiert und referenziert. Ist ein Projektplan in der Datenbank gespeichert, können alle Teilprozesse aus diesem Projektplan als neue Referenzprozesse abgelegt werden oder als Realisierungswert zu einem bestehenden Referenzprozess hinzugefügt werden (vgl. Abbildung 3).

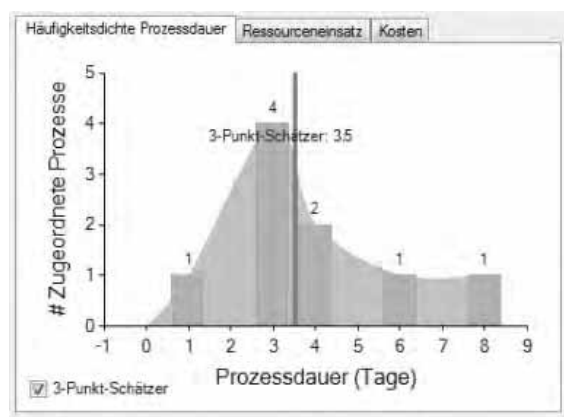


Abbildung 3: Historische Daten eines Referenzprozesses

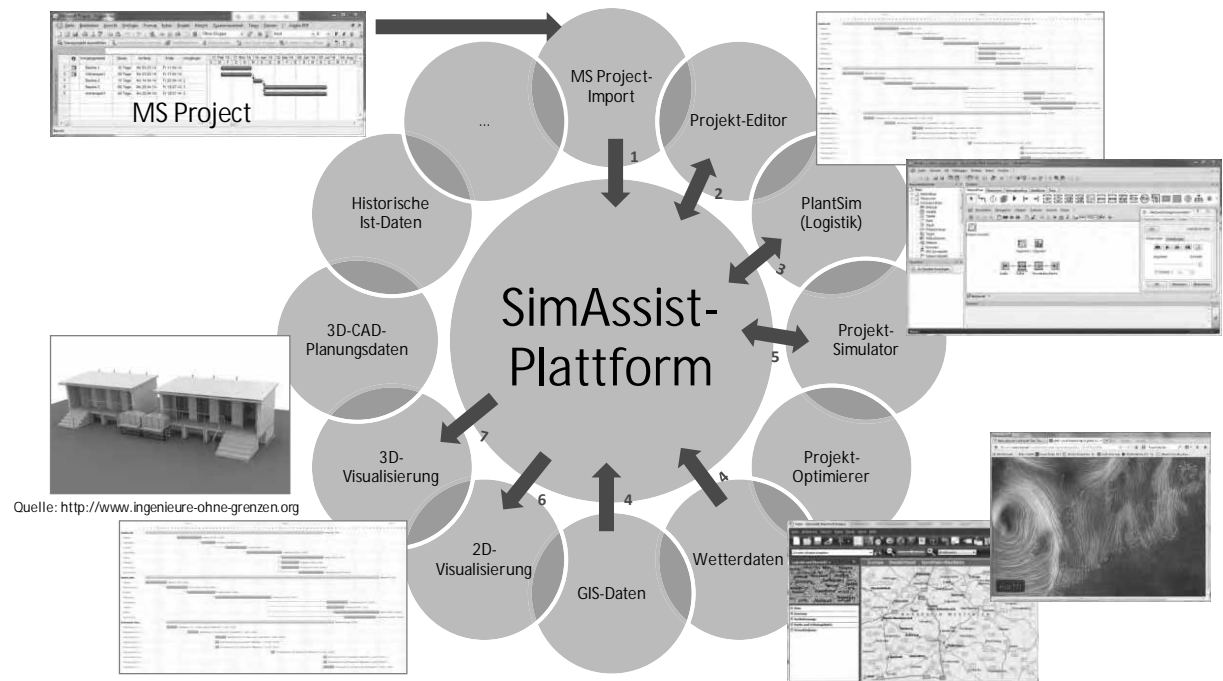


Abbildung 4: Systemkomponenten und Datenfluss unter Verwendung der Software SimAssist der Firma SimPlan AG

4 Diskussion und Ausblick

Die Evaluation des Projektes bei den beteiligten Projektpartnern ergab eine hohe Relevanz und Anwendbarkeit der erzielten Ergebnisse. Mit dem Einsatz der entwickelten Vorgehensweise werden eine abgesicherte Planung der logistischen Prozesse sowie eine stabilere Projektplanung gewährleistet. Durch die Verbesserung der konkreten Planungsleistung über eine genauere Abschätzung werden Projektdauer und –kosten durch das KMU kalkulierbarer; und es erfolgt schon in der Angebotsphase eine verbesserte Planung. Durch die Reduzierung der einzuplanenden Puffer, die heute noch aus der Unsicherheit der Planung entstehen, kann das Angebot zu realistischen Kalkulationspreisen unterbreitet werden und eröffnet den KMU daher eine gute Ausgangsposition im Bieterwettbewerb im Rahmen der Ausschreibungen von Investitionsprojekten.

Ein wesentlicher Vorteil kommt aber nach der Auftragserteilung durch den Kunden beim KMU zu tragen. Das entwickelte Vorgehen unterstützt eine robuste Steuerung des Projektes; projekthäufige Risiken werden durch den verantwortlichen Projektmanager besser quantifiziert und qualifiziert. Hierdurch ist die punktgenaue Steuerung auf den mit dem Kunden ver-

einbarten Auslieferungstermin bzw. Inbetriebnahmetermine gewährleistet (Liefertermintreue), was im Anlagenbau gerade auch im Vergleich zu preisgünstigeren Anbietern aus Asien ein entscheidender Wettbewerbsvorteil des deutschen Maschinen- und Anlagenbaus ist.

Der konkrete Mehrwert bei Anwendung lässt sich nur für einen konkreten Anwendungsfall bzw. ein konkretes KMU quantifizieren. Die Experten der Forschungsstellen schätzen, dass bis zu 30% der eingeplanten Projektlaufzeit, 15% der Änderungskosten und 10% der Investitionskosten reduziert werden. Zudem lassen sich über die Kenntnis der projekthäufigen Risiken und deren Auswirkungen im Eintrittsfall (ermittelbar über den Einsatz der Simulation) potenzielle Steuerungsfehler des Projektmanagers im Krisenmanagement in signifikanter Weise reduzieren.

Aufgrund der Verwendung von historischen Referenzprozessen bzw. kompletten Referenzprojektplänen wird erwartet, dass die Aufwände für das Erstellen und Neuerstellen von Plänen und für nicht-wertschöpfende Arbeiten auf der Baustelle deutlich gesenkt werden. Diese Nutzenpotenziale entstehen für die KMU im Wesentlichen durch die Tatsache, dass durch Vernetzung von Projektmanagement, simulationsgestützter Planbewertung und –optimierung sowie Visualisierung in einem praktikablen Ansatz eine umfassende

Entscheidungsunterstützung und damit eine permanente Entscheidungssicherheit gegeben wird. Sie entstehen aber nur dann, wenn die in der Projektsteuerung realisierten Ist-Werte auch systematisch vom Planer wieder in die Projektpläne zurückgespielt werden und sich somit die Basis historischer Daten zu den definierten Referenzprojektprozessen sukzessive erweitert.

Des Weiteren lässt das Vorgehensmodell auch eine Multi-Projektplanung unter Berücksichtigung von logistischen Restriktionen zu und erlaubt den KMU, eine deutlich höhere Planungsqualität als bisher zu erreichen. Abgesehen von der Tatsache, dass ein verlässlicher Projektplan Sicherheit für die eigene Planung aber auch die der zuliefernden Prozesse bietet, stellt die termingerechte Aufnahme des Anlagenbetriebs den entscheidenden Wettbewerbsvorteil für den Anlagenbetreiber dar. Detailliertere Darstellungen der Projektergebnisse finden sich auch unter [14].

Insbesondere für Kleinserien- und Unikatfertiger stellt sich eine weitere Herausforderung, die durch zukünftige Forschungsanstrengungen noch zu lösen ist. Ihnen fehlt durch den Charakter des Unikatproduktes zumeist eine belastbare Anzahl historischer Daten. Ähnliche Produkt- oder Prozesselemente wurden zwar in der Vergangenheit realisiert, unterscheiden sich aber in einem oder mehreren Parametern entscheidend von der konkreten aktuellen Kundenanforderung. Offen ist hier, wie auf Basis solcher Informationen die konkrete Prozessdauer eines kundenspezifischen Bauteils dennoch bestmöglich prognostiziert werden kann, um eine vernünftige Projektplanung zu realisieren. Hier Lösungen zu entwickeln wird eine zukünftige Aufgabe der beteiligten Forschungsstellen sein.

Literatur

- [1] Wenzel, S.; Laroque, C.: Methodik für ein simulationsgestütztes logistikintegriertes Projektmanagement im Anlagenbau. In: Dangelmaier, W.; Laroque, C.; Klaas, A. (Hrsg.): Simulation in Produktion und Logistik 2013. Paderborn: W.V. Westfalia Druck GmbH, S. 537-547.S
- [2] Gutfeld, T.; Jessen, U.; Wenzel, S.; Laroque, C.; Weber, J.: A Technical Concept for Plant Engineering by Simulation-Based and Logistic-Integrated Project Management. In: Tolk, A.; D'Allo, S. Y.; Ryzhov, I. O.; Yilmaz, L.; Buckley, S.; Miller, J. A. (Hrsg.): Proceedings of the 2014 Winter Simulation Conference; Savannah: Omnipress, S. 3423-3434.
- [3] Chahrour, R.: Integration von CAD und Simulation auf Basis von Produktmodellen im Erdbau. Dissertation, Bauingenieurwesen, Universität Kassel, 2007.
- [4] Weber, J.: Simulation von Logistikprozessen auf Baustellen auf Basis von 3D-CAD-Daten. Dissertation, Maschinenbau, Universität Dortmund. 2007.
- [5] Günthner, W. A.; Bormann, A. (Hrsg.): Digitale Baustelle – innovativer Planen, effizienter Ausführen: Werkzeuge und Methoden für das Bauen im 21. Jahrhundert. Berlin, Heidelberg: Springer, 2011.
- [6] Kugler, M.: CAD-integrierte Modellierung von agentenbasierten Simulationsmodellen für die Bauablaufsimulation im Hochbau. Dissertation Bauingenieurwesen, Universität Kassel, 2012.
- [7] Mefisto: Homepage des im Rahmen des BMBF-Förderprogramms IKT 2020 geförderten Leitprojektes Mefisto. <http://www.mefisto-bau.de>; zuletzt zugegriffen am 10.06.2016.
- [8] Steinhauer, D.: The Simulation Toolkit Shipbuilding (STS) – 10 Years of Cooperative Development and Interbranch Applications. In: Proceedings of the 10th Euro-Conference on Computer and IT Applications in the Maritime Industries (COMPIT), Berlin, 2011, S. 453-465.
- [9] Beißert, U.; König, M.; Bargstädt, H.-J.: Soft Constraint-based simulation of execution strategies in building engineering. In: Journal of Simulation (JOS) 4 (2010), S. 222-231 (doi:10.1057/jos.2010.8).
- [10] SimCoMar: Homepage der internationalen Kooperationsgemeinschaft SimCoMar. <http://www.simcomar.com/>; zuletzt zugegriffen am 10.06.2016.
- [11] SIMoFit: Homepage zur Kooperationsvorhaben SIMoFit. <http://www.simofit.com>; zuletzt zugegriffen am 10.06.2016.
- [12] Afinion: Homepage zum Afinion Project-Simulator: <http://www.afinion.ch/afinion-de/aktuelles/meldungen/Project-Simulator.php>; zuletzt zugegriffen am 10.06.16.

- [13] ProModel: Homepage zum Projektsimulator auf Basis von ProModel: <http://www.promodel.com/products/projectsimulator/management.asp>; zuletzt zugegriffen am 10.06.2016.
- [14] Gutfeld, T.; Jessen, U.; Wenzel, S.; Akbulut, A.; Laroque, C.; Weber, J.: simject - Simulationsgestütztes logistikintegriertes Projektmanagement im Anlagenbau. Hrsg.: Wenzel, S.; Laroque, C. ISBN 978-3-00-050113-5, 2015.

Modulares Modellkonzept für die planungsbegleitende Simulation eines Luftfrachtterminals

Karsten Turek, Armin Siegel, Thorsten Schmidt
TU Dresden, Institut für Technische Logistik und Arbeitssysteme
karsten.turek@tu-dresden.de

Der Artikel stellt einen strukturierten Ansatz zur planungsbegleitenden Simulation großer Materialflusssysteme am Beispiel eines Luftfrachtterminals vor. Luftfrachtterminals arbeiten als Verteiler von Frachtgut für Import und Export über die Land- und Luftseite. Typischerweise sind in diesen Systemen mehrere hundert Transportkomponenten installiert. Simulationsvorhaben dieser Systeme weisen Herausforderungen auf, die sich einerseits aus den verschiedenen Perspektiven der am Planungsprojekt beteiligten Partner und andererseits aus der Vielfalt und der Verzahnung der Logistikprozesse ergeben. Der Fokus des Artikels liegt auf einer Vorgehensweise zur adaptiven Detaillierung des Modells an einen fortschreitenden Planungs- und Erkenntnisstand. Die entsprechenden Anpassungen und Strukturen innerhalb des Simulationsmodells werden vorgestellt. Im Ergebnis ist eine Vorgehensweise entstanden, die eine schrittweise logische und visuelle Verfeinerung der Transportmittel sowie der Transport- und Bearbeitungsprozesse innerhalb eines bestehenden Simulationsmodells erlaubt. Zusätzlich erleichtert der modulare Modellaufbau die kollaborative Modellerstellung im Simulationsprojekt. Als Materialflusssimulator kommt die Software AutoMod zum Einsatz.

1 Einleitung

Die ereignisdiskrete Simulation (DES) wird bei der Planung neuer Logistiksysteme eingesetzt, um die neue Technik und Steuerung in alternativen Ausgestaltungen hinsichtlich ihrer Leistungsfähigkeit vorab zu bewerten [1, 2]. Typisch für Planungsprojekte ist die Unvollständigkeit in den Technikdetails und den Steuerungskonzepten während der Simulationsmodellerstellung. Deren Präzisierung ist eine erforderliche, im Aufwand oft unterschätzte Teilaufgabe in Simulationsprojekten.

Materialflusssimulationsmodelle bestehen aus drei wesentlichen Bereichen (vgl. [3]). Die Modelle weisen zum einen eine Repräsentation des räumlichen Layouts der materialflusstechnischen Komponenten auf. Dabei werden die technischen Parameter und die lokale Steuerung der Komponenten abgebildet (Anlageninformation). Ergänzt wird das Modell durch eine Logik, in der die Prozessfolgen und -abläufe sowie die Transport- und Lagerstrategien abgebildet werden

(Ablauforganisatorische Information). Drittens muss das Modell um Daten ergänzt werden, welche die vorgesehene Last für das Modell generieren (Produktionsinformation).

Auf dem Gebiet der ereignisdiskreten Simulation sind eine Vielzahl Werkzeuge verfügbar. Eine geeignete Übersicht wird in [4] dargestellt. Die Autoren verwenden AutoMod als Simulationswerkzeug. Dieses Tool ist eines der schnellsten kommerziellen Simulationssysteme in diesem Bereich und gilt daher als de-facto Standard für die Modellierung großer Materialflusssysteme (vgl. [5, 6]). Gleichwohl können auch andere Simulatoren erfolgreich eingesetzt werden [7].

Im Rahmen eines zurückliegenden Simulationsprojekts für ein Luftfrachtterminal wurde in Zusammenarbeit mit einem Projektpartner ein modulares Modellkonzept entwickelt und getestet. Details und Ergebnisse zu diesem Projekt können aufgrund interner Vereinbarungen hier jedoch nicht dargestellt werden. Grundprinzipien und Vorgehensweise innerhalb des Konzepts werden nachfolgend erläutert.

Der Artikel ist wie folgt strukturiert. Zuerst wird die Intralogistik in Luftfrachtterminals charakterisiert. Anschließend wird die Grundidee des modularen Simulationsmodellkonzepts vorgestellt und motiviert. Danach wird auf Details und Implementierungsaspekte des Konzepts eingegangen. Die Funktionsweise der implementierten generischen Transportmittelauswahl wird erläutert. Abschließend wird die gewählte Vorgehensweise bewertet und eingeordnet.

2 Intralogistik in Luftfrachtterminals

Luftfrachtterminals (auch: Luftfrachthub) arbeiten als Verteiler für Frachtgut im Luftverkehr. Sie realisieren den Import und Export von Gütern. Sie besitzen eine Anbindung über die Landseite via LKW und über die Luftseite via Flugzeug. Das Luftfrachtgut wird entweder in standardisierten Containern und Paletten (ULD - Unit Load Device) oder als sogenannte lose Ware angeliefert. Im Terminal muss das Frachtgut identifiziert, sortiert, zwischengespeichert und neu zusammengestellt werden. Die Intralogistik in Luftfrachtterminals ist geprägt durch eine starke Verzahnung und hohe Vielfalt der Logistikprozesse.

Auf der Luftseite werden die Transportgüter zunächst in Containern oder auf Paletten aus dem Flugzeug entladen und mittels Dolly-Zügen zum Terminal gebracht. Die Weitergabe zur Fördertechnik im Terminal erfolgt an speziellen Übergabestellen. Von dort werden die ULD abhängig von ihrem Frachtziel und den zeitlichen Anforderungen zur Zwischenlagerung, zu Entladestationen oder direkt zur Landseite transportiert. An den Arbeitsstationen (Break down) wird das Frachtgut aus den ULD manuell entladen, identifiziert, auf Paletten gestellt und zur Zwischenlagerung in verschiedene Lagerorte transportiert. Der Transport im Terminal erfolgt mittels Verteilwagen (TV - Transfer Vehicle), Liften, Hebern und Stetigförderern. Warentransporte erfolgen hauptsächlich automatisch, werden jedoch durch manuelle Transporte ergänzt. Entsprechend der Größenklassen der Transportgüter kommen unterschiedliche Systemtechniken beim Transport zum Einsatz.

Der Prozess der Bereitstellung des Frachtgutes wird in einer definierte Zeitspanne vor Abflug bzw. Abtrans-

port gestartet. Die zwischengelagerten Güter werden aus ihren Lagerbereichen angefordert und zu den Arbeitsstationen (Build up) transportiert. Parallel wird eine leere ULD bereitgestellt. Nach Abschluss der Beladung erfolgt der Transport der ULD über die Übergabestellen via Dolly-Zug zum Flugzeug.

Die logistische Systemlast wird in einem Flugplan für das Terminal definiert. Der Flugplan gibt Zeitpunkte und Beladung der ankommenden und abgehenden Flugzeuge und LKW vor und setzt dadurch Mengen und Abhängigkeiten der umzuschlagenden Frachtgüter fest. Die zeitliche Abfolge der logistischen Prozesse, wie z. B. der ULD Entladung und ULD Beladung, ergibt sich in Konsequenz daraus. Dabei kommt es zur Überschneidung der konkurrierenden Transportprozesse von Import und Export. (vgl. [8, 9])

3 Modellkonzept

Die Systemplanung eines Luftfrachtterminals ist eine komplexe Aufgabe mit erheblichen Herausforderungen für die Simulationsentwicklung [10]. Einerseits soll das Simulationsmodell in der Planungsphase schnell erste Erkenntnisse über das Zusammenwirken innerhalb des Gesamtsystems liefern. Andererseits entstehen etliche Details der Teilsysteme sowie wesentliche Steuerungsalgorithmen erst mit Planungsfortschritt und können deshalb erst während des Modellaufbaus hinzugefügt werden.

Ein weiterer Aspekt der Konzeptentwicklung war das Ziel einer einfachen Konfigurierung und Validierung des Modelllayouts und der geplanten Transportrouten. Die Prüfung auf Plausibilität und Vollständigkeit der abgebildeten Materialflüsse stellt ein wichtiges Teilerziel in der Modellentwicklung dar. Flexible Anpassungen im Layout sollen auch durch Projektbeteiligte mit nur geringer AutoMod Erfahrung durchgeführt werden können. Dabei war schnell klar, dass diese beiden Ziele nur über eine geeignete parametrierbare Transportsteuerungslogik erreicht werden können (vgl. [11]). Die Problemlösung soll deshalb auf einem generischen Modellansatz basieren.

Die Grundidee des Konzeptes besteht darin, das Gesamtmodell in ein Transportmodell, ein Dispositionssystem und einen Routenkonfigurator aufzutrennen (vgl. Abbildung 1). Diese Modularisierung ermöglicht

eine parallele Entwicklung in einem flexiblen Änderungsprozess. Ergänzende Ausführungen zu den Modulen des Modellkonzepts sind in [7] dargelegt.

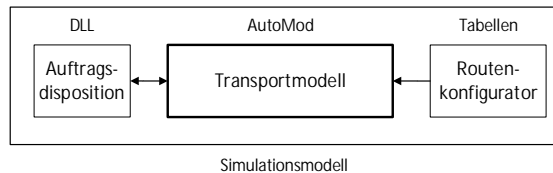


Abbildung 1: Modellarchitektur

Die Lastdatengenerierung wird in einem eigenständigen Datengenerator durchgeführt. Der Datengenerator bildet sämtliche Anforderungen und Restriktionen ab, um eine vorgegebene Menge an Luftfracht auf Flüge eines Flugplans zu verteilen. Die Fracht (Inbound, Outbound, Export und Import) wird gemäß parametrierbarer Verteilung auf den Simulationszeitraum verteilt. Entscheidend hierbei ist die Beachtung aller abhängigen Bilanzen der einzelnen Gutströme. So muss die Gesamtbilanz je Frachtklasse eingehalten werden. Gleiches gilt für die Bilanzen der Gewichte je Flugzeugtyp, der Charakteristiken der Lieferungen.

Das Dispositionssystem verantwortet die Verteilung der Fracht auf Arbeitsstationen und Lagerbereiche. Die Forderung nach einer Synchronisation aller Abläufe im Luftfrachtzentrum detailgenau für jedes einzelne Paket erfordert eine übergeordnete Steuerungsinstanz. Diese verfolgt für jedes Paket, ebenso für die übergeordneten Ladeeinheiten wie Sendungen und Transporteinheiten, den aktuellen Status sowie den Aufenthaltsort im System.

Das Dispositionsmodul ist in C++ realisiert und als DLL mit dem Simulationsmodell in AutoMod über Schnittstellenfunktionen verbunden. Es erhält Konfigurationsdaten und Ereignismeldungen aus dem Transportmodell mitgeteilt und übergibt Transportaufträge an das Transportmodell. Über die zeitliche Zuordnung der Transportaufträge realisiert die Steuerung die dispositive Zuweisung von Arbeitsstationen, Pufferplätzen, Lagerstellplätzen. Die Kommunikation wird durch das AutoMod-Transportmodell gesteuert und erfolgt ereignisbasiert als auch im Polling-Verfahren.

Das Luftfrachtterminal wurde in transportlogische Teilbereiche (z. B. Bearbeitungsbereiche, Puffer, Lager) aufgeteilt. Innerhalb dieser Bereiche sind logische Orte definierbar, die konkreten Orten im Sys-

tem entsprechen. Die Orte haben in Abhängigkeit von ihrer konkreten Funktion (Bearbeiten, Lagern, Puffern) spezifische Eigenschaften wie Temperaturklasse, geeignete Gut-Größe, Kapazität. Anzahl und Eigenschaften der Orte sind für Simulationsexperimente flexibel parametrierbar und bestimmen so die Funktionsweise der Teilbereiche.

Zu Beginn eines Simulationslaufes werden die Orte der Bereiche und ihre Parameter an die Steuerung gemeldet und dienen als Grundlage für Dispositionsentscheidungen. Innerhalb einer gegebenen Grundstruktur wird über diese Parametrierbarkeit eine Flexibilität in der modellierbaren Systemstruktur erreicht.

Das Transportmodell realisiert die Transportdurchführung zwischen den Orten auf der Basis der Tabellen des Routenkonfigurators. Die Beziehungen der Transportmittel im Terminal können als strukturiertes Netzwerk dargestellt werden (vgl. [12]). Nachfolgend wird das Zusammenspiel von Transportmodell und Dispositionsmodul an einem Beispiel anhand des vereinfachten Transportnetzwerks in Abbildung 2 erläutert.

Eine ULD, zu einem ankommenden Flugzeug gehörend, wird entsprechend des Flugplans an der Quelle Airside (S1) generiert. Für diese ULD erzeugt das Dispositionsmodul eine Transportanweisung zur Übergabestelle (Inbound Airside). Die Transportanweisung wird im Transportmodell durch das Transportmittel Dolly-Zug durchgeführt. Die Verwaltung der Dolly-Züge obliegt dem Transportmodell. Die Ankunft der ULD an der Inbound Airside Übergabestelle wird als Ereignis an das Dispositionsmodul gemeldet. Die ULD wartet danach auf die nächste Transportanweisung.

In Abhängigkeit vom Transportinhalt der ULD entscheidet das Dispositionsmodul über den nächsten Zielbereich, beispielsweise der Arbeitsstationsbereich A2xx. Andere mögliche Zielbereiche sind das ULD Lager oder Zwischenpuffer vor den Arbeitsstationen (zur Vereinfachung nicht in der Abbildung dargestellt). Die Steuerungsentscheidung beinhaltet die Prüfung der Verfügbarkeit eines Zielplatzes (z. B. A2L2) im Zielbereich. Das Transportmodell setzt den Transportauftrag mit den Transportmitteln Verteilwagen (TV IF 1), ULD-Förderer 1, Lift 1 oder Lift 2 und Verteilwagen (TV 1) um. Gleichzeitig überwacht das Dispositionsmodul intern die Transportaufträge, um eine Überbuchung der Zielbereiche

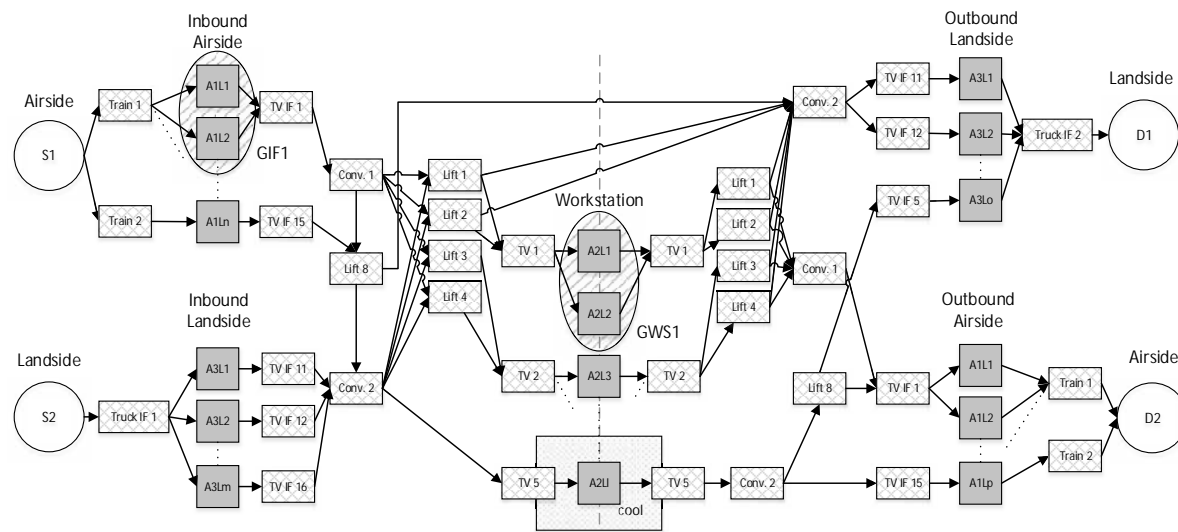


Abbildung 2: Vereinfachtes Transportnetzwerk

zu vermeiden. Bei der Zuweisung der Zielbereiche und Zielorte sind teilweise spezifische Eigenschaften des Transportgutes zu beachten, so z. B. sind für Cool-ULD spezielle Arbeitsstationen und Lagerbereiche vorgesehen. Diese Cool-Stationen sind räumlich getrennt angeordnet und werden über andere Transportmittel erreicht, wie in Abbildung 2 angedeutet.

Nach Ankunft an der Bearbeitungsstation, dem Transportziel, meldet die ULD aus dem Transportmodell ihren geänderten Status an das Dispositionsmodul. Der Entladevorgang wird im Transportmodell über einen parametrierbaren Zeitbedarf je Größe und Art des Transportgutes umgesetzt. Das entladene Frachtgut erhält vom Dispositionsmodul die entsprechenden Transportziele zugewiesen, das Transportmodell übernimmt dann deren Weitertransport. Das Dispositionsmodul verwaltet die Inhalte der Lager- und Pufferbereiche. Die leeren Transporthilfsmittel werden ebenfalls vom Dispositionsmodul verwaltet und disponiert.

Im Fall der Frachtauslieferung stößt das Dispositionsmodul entsprechend den Anforderungen des Flugplans die Bereitstellung von Transportgut und Transporthilfsmittel aus den Speicherbereichen an. Die Synchronisation der Transportströme innerhalb eines Bearbeitungsprozesses, z. B. im Build up, findet im Transportmodell statt.

Die Konfiguration der geplanten Transportmittelnutzungsabfolgen zwischen den definierten Orten im Netzwerk erfolgt in Tabellen des externen Routen-

konfigurator. In einem ersten Schritt lassen sich die Transportmittelfolgen aller Ort-zu-Ort-Beziehungen wie in Tabelle 1 abbilden.

In der Tabelle sind der Start und das Ziel des Transportes sowie die Reihenfolge der nacheinander genutzten Transportmittel notiert. Es wurden Folgen mit bis zu acht Zwischenschritten erstellt. In der Folge alternativ benutzbare Transportmittel werden in einer Zelle mit einem / getrennt.

Tabelle 1: Notation Transportmittelfolgen

From	To	Step 1	Step 2	Step 3	Step 4
A1L1	A2L1	TV IF 1	Conv. 1	Lift 1/Lift 2	TV 1
A1L1	A2L2	TV IF 1	Conv. 1	Lift 1/Lift 2	TV 1
⋮	⋮	⋮	⋮	⋮	⋮

Bei etwa 500 Orten im Beispielmmodell mit nahezu allen möglichen Transportrelationen dazwischen ist der manuelle Erstellungsaufwand und die potentielle Fehlerrate entsprechend hoch. Gleichfalls enthält die Tabelle eine Vielzahl redundanter Einträge, so ist z. B. die Ressourcennutzung vom Dolly-Interface (A1L1) zu den Arbeitsstationen eins und zwei (A2L1 und A2L2) identisch. Aus diesen Gründen wurde mit Gruppen (G) eine weitere Hierarchieebene (Abbildung 3) eingeführt, um die Notation der Relationen deutlich kompakter und übersichtlicher zu gestalten.

Auf der Ebene von Gruppen (G) werden Orte eines Bereichs zusammengefasst, die sich dadurch aus-

zeichnen, dass sie mit demselben Transportmittel bedient werden. Für das betrachtete Beispiel in Tabelle 1 zeigt Tabelle 2 die veränderte Notation mit Gruppen. Im betrachteten Simulationsmodell konnten bis zu 20 Orte zu einer Gruppe zusammengefasst werden. Im Ergebnis entstanden 120 Gruppen. Dadurch verringert sich der Umfang der Konfigurationstabelle erheblich.

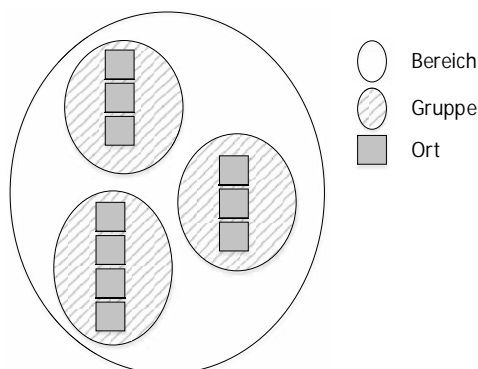


Abbildung 3: Hierarchische Struktur der Transportziele

Im ersten Modellierungsschritt werden die Transportmittel im Simulationsmodell durch Ressourcen mit begrenzter Kapazität sowie die einzelnen Orte durch Warteräume (Queues) abgebildet. Die Erstellung dieses sogenannten Q-R-Modells ohne präzise Layoutrepräsentation ist vergleichsweise schnell möglich. So kann frühzeitig mit dem Test und der Validierung der Steuerungsfunktionen und der Modellstruktur begonnen werden. Bereits das Q-R-Modell erlaubt Aussagen zu Engpässen und Schwachstellen im System. Eine Herausforderung bei dieser Herangehensweise ist es, dem Auftraggeber zu vermitteln, dass das Q-R-Modell auch ohne Visualisierung der Transporte für Bewertungen sinnvoll eingesetzt werden kann.

Tabelle 2: Notation Transportmittelfolgen mit Gruppen

From	To	Step 1	Step 2	Step 3	Step 4
GIF1	GWS1	TV IF 1	Conv. 1	Lift 1/Lift 2	TV 1
⋮	⋮	⋮	⋮	⋮	⋮

In den nachfolgenden Modellierungsschritten wird das AutoMod-Modell um Teilmodelle dynamischer Transportsysteme ergänzt. Es entsteht eine Detaillierung und Visualisierung der Transportprozesse. Dafür sind die Layoutrepräsentationen der Transportsysteme (Fahrzeuge, Fahrwege und Stoppunkte) im Modell abzubilden. In AutoMod bieten sich für die Repräsentationen von Verteilwagen, Liften und Dolly-Zügen

das Teilsystem Path Mover mit Controlpoints an sowie das Teilsystem Conveyor mit Stations für Förderer. Die Informationen aus Tabelle 2 sind um Spezifizierungen der Übergänge zwischen den Transportmitteln der Folge zu ergänzen. Hierbei sind drei Fälle zu unterscheiden:

- der Übergang von einer Ortsgruppe zum ersten Transportmittel
- der Übergang von einem Transportmittel zum nächsten
- der Übergang vom letzten Transportmittel zur Ortsgruppe

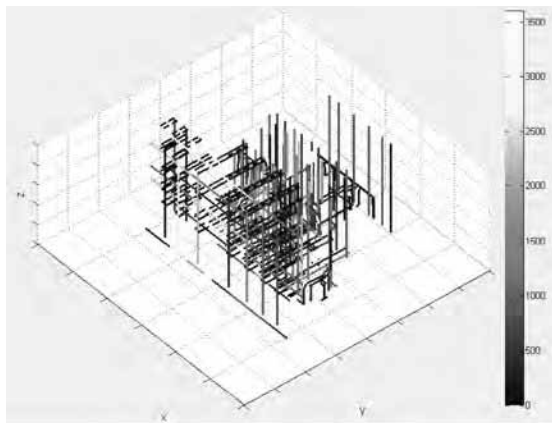
Somit werden Start, Ende und die Zwischenschritten in der Transportfolge getrennt notiert. Der Aufbau der drei entsprechenden Konfigurationstabellen ist strukturgleich gestaltet. Nachfolgend wird die Umsetzung für den ersten Fall beispielhaft erläutert. Die Tabelle 3 zeigt den Übergang von der Gruppe Interface 1 (GIF1) zum anschließenden Verteilwagen (TV IF1) und von der Gruppe Arbeitsstation 1 (GWS1) zum Verteilwagen TV 1. Dabei werden immer das AutoMod-Teilsystem in dem die ULD sich befindet und der Modellpunkt (Controlpoint, Station oder Queue) über den die ULD befördert werden soll angegeben. Die Notation erfordert bis zu vier Schritte, um einen Übergang von beispielsweise einem Verteilwagen (TV) zu einem Lift mit einem kurzen Zwischenförderer abbilden zu können. Dabei wird zuerst der Stoppunkt auf dem TV beschrieben. Im zweiten und dritten Schritt werden die Ein- und Ausgangsstationen auf dem Zwischenförderer notiert und im letzten Schritt der Startpunkt des Liftes. Die Umsetzung dieser Abfolge erfolgt innerhalb von AutoMod dann mit den Befehlen `move into` und `travel to`. Als Besonderheiten sind in Tabelle 3 Einträge gezeigt, die mit `[loc]` eine dynamische Zuordnung des Startortes und mit `[1..4]` eine zufällige Wahl des Ortes in der konkreten Transportausführung gestatten. Die vorgegebene Transportfolge wird auf diese Weise Schritt für Schritt vom Transportmodell in AutoMod abgearbeitet bis der Zielort erreicht ist. Die Bezeichnungen in diesen Tabellen müssen mit den Komponenten im AutoMod-Modell übereinstimmen.

Auf der Basis dieser Implementierung kann das Modell schrittweise um eine detaillierte Repräsentation der Transportmittel ergänzt werden. Pufferförderer zwischen zwei Fahrzeugsystemen können bei Bedarf hinzugefügt werden. Die Konfigurationstabellen

Tabelle 3: Transportmittelübergänge

Gruppe	Res.	Sub Sys. 1	Loc.	Sub Sys. 2	Loc.	Sub Sys. 3	Loc.	Sub Sys. 4	Loc.
GIF1	TV IF 1	(main system)	qTV_IF_(1)	Airside.TV_IF	cp1_1_[1..4]				
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
GWS1	TV 1	subsys2.cnv	st_WS_[loc]	subsys2.TV_1	cp12_[loc]				
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

(und deren Verarbeitung im AutoMod-Modell) erlauben das Eintragen sowohl von Elementen modellierter dynamischer Transportsysteme als auch von statischen Ressourcen mit den Parametern Kapazität und Zeitverbrauch. Orte können durch Stoppunkte eines dynamischen Transportsystems als auch über statischen Warteräume (Queues) abgebildet werden. So können Abbildungen mittels Q-R-Repräsentation parallel zu Elementen dynamischer Transportsysteme in der Transportmittelfolge verwendet werden. Die Art der Transportdurchführung im Simulationsmodell ist entkoppelt von der vorab logisch definierten Transportmittelkette. Nach jedem Detaillierungsschritt können die weiteren Maßnahmen mit den beteiligten Partnern abgestimmt werden.

**Abbildung 4:** Validierung von Transportflüssen

Eine Zuordnung von Raumkoordinaten zu den Orten und den Transportmitteln im Routenkonfigurator ermöglicht eine aussagekräftige räumliche Darstellung der Transportabfolgen aller Ort-zu-Ort-Beziehungen auch außerhalb des Modells, exemplarisch siehe Abbildung 4. Diese Darstellung bietet eine zusätzliche Überprüfung der Transportflüsse und ihrer Implementierung. Asymmetrische Verteilungen als auch Lücken in der Abfolge können erkannt werden.

4 Zusammenfassung

Innerhalb des Planungsprozesses komplexer Materialflusssysteme werden Layout, Technikconfiguration und Steuerungsalgorithmen ausgehend von provisorischen Lösungen stetig weiterentwickelt. Deshalb erfordert die planungsbegleitende Simulation eine starke Flexibilität im Simulationsmodell, um mit den Planungsänderungen Schritt halten zu können.

Der im Beitrag skizzierte modulare Ansatz beruht auf der getrennten Implementierung von Auftragsdisposition, Routenconfiguration und Transportdurchführung. Die Auftragsdisposition verwaltet konfigurierbare Systemorte als Start- und Zielpunkte von Transporten. Die Routenconfiguration legt die strukturelle Transportmittelabfolge zwischen den Systemorten fest. Die Transportdurchführung einschließlich lokaler Transportsteuerung übernimmt das AutoMod-Modell mit inhärenter Visualisierung.

Ausgehend von einem einfachen Q-R-Modell können die Teilbereiche und Transportmittel des Logistiksystems in unterschiedlichem Grade bis zur erforderlichen Visualisierung modelliert werden. Das Modellkonzept erlaubt die schrittweise logische und visuelle Verfeinerung der Transportmittel sowie der Transport- und Bearbeitungsprozesse innerhalb eines gewählten Rahmens im Simulationsmodell unter Beibehaltung der Dispositionsalgorithmen. Mit Abschluss des Simulationsprojekts steht ein Modell zur Leistungsbewertung mit dem zur Vorbereitung der Installation benötigten Detaillierungsgrad zur Verfügung.

Der modulare Modellaufbau erleichtert wesentlich die Zusammenarbeit bei der Modellerstellung und verbessert die Validierungsmöglichkeiten durch Einzelbetrachtung der Module. Darüber hinaus bietet das Konzept einen Ausgangspunkt zur Entwicklung einer Modellierungsvorgehensweise, mit der Simulationsmodelle in AutoMod zukünftig effizienter und agiler erstellt werden können.

Literatur

- [1] J. Banks, J. S. Carson und B. L. Nelson. *Discrete-event system simulation*. Prentice-Hall international series in industrial and systems engineering. Prentice Hall, Upper Saddle River, N.J, 2nd edition, 1996.
- [2] A. M. Law und W. D. Kelton. *Simulation modeling and analysis*. McGraw-Hill series in industrial engineering and management science. McGraw-Hill, 3rd edition, 2000.
- [3] 3633 VDI. Simulation von Logistik-, Materialfluss- und Produktionssystemen Grundlagen, Beuth Verlag Berlin, 2010.
- [4] J. J. Swain. Simulation software survey – simulation: a better reality? *OR/MS Today*, 40(5), 2013.
- [5] K. Hallenborg und Y. Demazeau. Decide: applying multi-agent design and decision logic to a baggage handling system. In *Engineering Environment-Mediated Multi-Agent Systems*, S. 148 – 165. Springer, 2008.
- [6] D. Muller. AutoMod®: Modeling complex manufacturing, distribution, and logistics systems for over 30 years. In *Proceedings of the 2013 Winter Simulation Conference*, S. 4037 – 4051. IEEE Press, 2013.
- [7] M. Bös, K. Turek und A. Siegel. Moderne Anwendungskonzepte bei logistischen Simulationsvorhaben. In *Tagungsband zum Forum Technische Logistik und Arbeitssysteme 2016*, S. 29 – 40. Selbstverlag der Technischen Universität Dresden, 2016.
- [8] P. DeLorme, J. Procter, S. Swaminathan und T. Tillinghast. Simulation of a combination carrier air cargo hub. In *Proceedings of the 1992 Winter Simulation Conference*, S. 1325 – 1331. ACM Press, 1992.
- [9] A.L. Nsakanda, M. Turcotte und M. Diaby. Air cargo operations evaluation and analysis through simulation. In *Proceedings of the 2004 Winter Simulation Conference*, S. 711 – 719. IEEE, 2004.
- [10] C. Versteegt, S. Vermeulen und E. van Duin. Joint simulation modeling to support strategic decision-making process. In *Proceedings 15th European Simulation Symposium*, S. 284 – 290. SCS, 2003.
- [11] S. Rank, C. Hammel, T. Schmidt und G. Schneider. Reducing simulation model complexity by using an adjustable base model for path-based automated material handling systems - a case study in the semiconductor industry. In *Proceedings of the 2015 Winter Simulation Conference*, S. 2896 – 2907. IEEE, 2015.
- [12] C. Lee, Huei C. Huang, B. Liu und Z. Xu. Development of timed colour petri net simulation models for air cargo terminal operations. *Computers & Industrial Engineering*, 51(1):102 – 110, 2006.

MODELLIERUNG VON ARBEIT AN BEWEGTEN OBJEKTEN IN KOLLABORATIVEN BETRIEBSFORMEN

Titanilla Komenda¹, Felix Breitenecker²

¹Centauro GmbH

²Technische Universität Wien, Institut für Analysis und Scientific Computing

komenda@centauro.at

Die Mensch-Maschine-Kollaboration wird von Parametern beeinflusst, die in Konflikt zu Zykluszeit und Produktivität stehen. Ein optimales Set an Einflussparametern kann jedoch mithilfe von Simulationsmodellen bestimmt werden, die die physikalischen, psychophysiologischen und operativen Faktoren sowie relevante Sicherheitsaspekte des Gesamtsystems berücksichtigen. In diesem Paper, werden anhand einer systemischen Analyse kollaborativer Szenarien, Einflussfaktoren von Arbeiten an bewegten Objekten definiert, die zur Systembeschreibung unter Beachtung von Nichtlinearitäten dienen. Auf Basis der Systemanalyse lässt sich in weiterer Folge ein Beschreibungsformalismus für die Mensch-Maschine-Kollaboration als hybrides Modell ableiten, der zur Auslegung und Optimierung von kollaborativen Szenarien herangezogen werden kann.

1 Einleitung

Ein Produktionsbetrieb, bei dem ein Mensch und eine hierfür konstruierte Maschine innerhalb eines festgelegten Arbeitsraums (geschützter Bereich als so genannter Kollaborationsraum) direkt zusammenarbeiten, wird nach DIN EN ISO 10218-1&2:2012 [1] [2] sowie nach ISO/TS 15066:2016 [3] als kollaborierender Betrieb bezeichnet. Diese Interaktion bzw. Zusammenarbeit ist dabei von Parametern bestimmt, die aus operativen Zielvorgaben, maschinellen sowie menschlichen Faktoren und der durchzuführenden Arbeitstätigkeit resultieren. Im Allgemeinen sind Maschinen so programmiert, dass sie eine minimale Zykluszeit bei gleichzeitig hoher Produktivität erreichen. In der Zusammenarbeit mit Menschen müssen jedoch zusätzliche Faktoren, wie Sicherheit, Ergonomie, Anthropometrie, Psychophysiologie und physische Grenzen mitberücksichtigt werden. Bei der Validierung von Mensch-Maschine-Modellen sind dabei im Speziellen physische Grenzen sowie psychophysiologische Einflüsse von entscheidender Bedeutung. In manchen Fällen kann es nämlich dazu kommen, dass der Mensch die gemeinsame Tätigkeit nicht fertigstellen konnte, weil die Maschine das Objekt zu schnell aus dem Kollaborationsbereich befördert oder der Mensch einen Fehler gemacht hat.

2 Kollaborative Betriebsformen

Anwenderstudien von Mensch-Roboter-Kollaborationen zeigen bereits mögliche kollaborative Betriebsformen [4] [5]. Der Fokus jener Studien liegt dabei oft

auf einer anwendbaren und vor allem sicheren Auslegung bzw. Programmierung der Mensch-Roboter-Zusammenarbeit [6] [7] oder auf der psychologischen Verhaltensanalyse des Menschen [8]. Die Evaluierung des Einflusses ergonomischer Bedingungen, physischer Grenzen oder psychophysiologischer Eigenschaften auf die Produktivität des Mensch-Maschine-Systems (MMS) bleibt hier meist unberücksichtigt.

2.1 Klassifizierung

Kollaborierende Betriebsformen im Sinne einer direkten Interaktion zwischen Mensch und Maschine lassen sich grundsätzlich in drei Arten unterscheiden (Abbildung 1):

- Mensch – Bearbeitungsmaschine (Bedienung, sich daraus ergebende Zeitregime, z.B. Arbeiten an Fördersystemen, Befüllen/Entleeren von Bearbeitungsmaschinen)
- Mensch – Logistik (Sicherheitsaspekte bei der Raum- und Arbeitsteilung, z.B. mit Flurförderzeugen oder autonomen mobilen Plattformen)
- Mensch – Roboter (Sicherheitsaspekte bei Arbeiten ohne trennende Schutzeinrichtungen, z.B. gemeinsames Handling, gemeinsame Bearbeitung)

2.2 Faktoren kollaborativer Arbeit

Während des kollaborativen Betriebes kommt es im Kollaborationsraum somit auch zu einer Annäherung und gegebenenfalls zu einem direkten Kontakt zweier

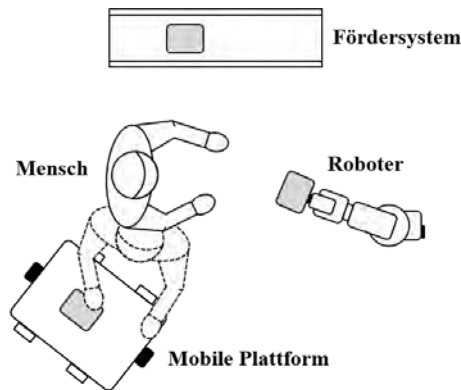


Abbildung 1. Kollaborative Betriebsformen

Kollaborationspartner. Dies führt einerseits zur Berücksichtigung maschinenspezifischer Parameter:

- Maschine (Arbeitsbereich, Manipulationslast, Geschwindigkeit, Kraft, Energie)
- Endeffektor inkl. Werkstück (Größe, Gewicht, ergonomische Gestaltung, scharfe Kanten, überstehende Teile)
- Maschinensystem (Anordnung und Sequenz)
- Verfahrenswege, Geschwindigkeiten und Beschleunigungen (Abstand und Annäherungsgeschwindigkeit der Kollaborationspartner)
- Befestigungen, Spannvorrichtungen bzw. Peripherie (Gestaltung, Platzierung und Betrieb v.a. in Hinblick auf Kollisionen)
- Schutzeinrichtungen (Arbeitsraumüberwachung im Sinne einer Positions-, Geschwindigkeits- und Kraftüberwachung)

und andererseits zur Berücksichtigung menschlicher Faktoren [9], wie:

- Bedienperson (Position in Hinblick auf die Nähe zur Maschine, Informationsaustausch)
- Anthropometrie und Ergonomie (Körperhaltungen, Geometrie der Arbeitsumgebung, Störkonturen)
- Physische Grenzen (Reaktions-, Bewegungs- und Vorgabezeiten)
- Psychophysiologie (Stressreaktionen, nicht normgerechtes Verhalten)

2.3 Zielfunktion

Eine kollaborative Tätigkeit **CO** (*Collaborative Operation*) lässt sich mithilfe menschspezifischer sowie maschinenspezifischer Parameter definieren. Insbesondere sind die genannten Parameter auch abhängig von der jeweiligen Bedienperson $CO_H(i)$ und

der eingesetzten Maschine $CO_M(j)$. In einem kollaborierenden System können dabei eine unterschiedliche Anzahl an Menschen h sowie Maschinen m miteinander arbeiten. Zur Bewertung des kollaborativen Systems, werden nun die menschspezifischen und die maschinenspezifischen Parameter mithilfe einer Gewichtungsmatrix \mathbf{W} gewichtet. Das führt zu der Annahme, dass ein optimales Set an kollaborativen Parametern für eine bestimmte Kollaborationsaufgabe existiert, das einer spezifischen Zielfunktion, beispielsweise der einer minimalen Taktzeit, gerecht wird. Insofern lässt sich die Parametrisierung eines kollaborativen Systems als multi-kriterielles Optimierungsproblem mit einer gewichteten Zielfunktion beschreiben.

$$f_{obj} = \min \left[\sum_{i=0}^h \mathbf{W}_H(i) \cdot \mathbf{CO}_H(i) + \sum_{j=0}^m \mathbf{W}_M(j) \cdot \mathbf{CO}_M(j) \right] \quad (1)$$

Der Kollaborationsbereich, die Objektpose bzw. der Bewegungspfad des Objektes sowie die Objektgeschwindigkeit und -beschleunigung wird dabei durch die jeweilige Maschinensteuerung bestimmt [10] [11]. Ergonomische Richtlinien für kollaborative Arbeiten können aktuellen Richtlinien zur Gestaltung manueller Arbeitsplätze entnommen werden (wie z.B. ergonomische Arbeitshöhen oder freie unverstellte Flächen rund um einen Steharbeitsplatz). Physische Grenzen, die u.U. die Produktivität kollaborativer Tätigkeiten beeinflussen, müssen jedoch im Rahmen von Experimenten ermittelt werden. Ebenso ist der Zusammenhang zwischen Produktivität und Psychophysiologie empirisch zu ermitteln und noch Teil der Forschung.

3 Arbeit an bewegten Objekten

Zur Ermittlung physischer Grenzen und des adaptiven Verhaltens des Menschen bei der Arbeit an bewegten Objekten, wurden 20 verschiedenen Tätigkeiten an bewegten Objekten durchgeführt und analysiert. Die kollaborativen Tätigkeiten wurden auf Basis aktueller Anwendungsfälle [12] [13] und auf Basis von Verrichtungen, die sowohl im Methods-Time Measurement System (MTM) als auch in der Simulationsumgebung Editor Menschlicher Arbeit (ema) enthalten sind, definiert. MTM ist ein standardisiertes Verfahren zur Analyse von manuellen Arbeitsabläufen sowie zur Ermittlung von Plan- und Vorgabezeiten und umfasst eine Vielzahl an Verrichtungen mit empirisch ermittelten Zeitwerten als Normleistungsbezug [14]. ema ist die bislang am weitest entwickelte Simulationsumgebung zur virtu-

ellen Planung menschlicher Tätigkeit und basiert auf MTM, d.h. den in ema implementierten Verrichtungen sind die MTM-Normzeitwerte hinterlegt [15]. Da Menschen im Rahmen der Arbeit an bewegten Objekten jedoch in der Lage sind, sich der Geschwindigkeit der bewegten Objekte anzupassen, würde das auch andere Zeitwerte für einzelne Verrichtungen mit sich führen. Aus diesem Grund können bislang keine dynamischen Tätigkeiten im Sinne von Arbeiten an bewegten Objekten planerisch dargestellt werden – besonders auch in Hinblick auf physische Grenzen.

3.1 Hypothesen und Versuchsplanung

Auf Basis der Analyse kollaborativer Betriebsformen im Rahmen einer Mensch-Maschine-Zusammenarbeit, lassen sich folgende Hypothesen aufstellen:

- H1 – Arbeiten an bewegten Objekten sind durch eine Minimal- und eine Maximalgeschwindigkeit begrenzt, bei der die Bedienungsperson entweder unter- oder überfordert ist.
- H2 – Die Fehlerrate von Arbeiten an bewegten Objekten steigt mit erhöhter Bewegungsgeschwindigkeit der Objekte.
- H3 – Es existiert ein optimales Set an Bewegungsparametern, bei dem das Verhältnis von Anforderung und Fehlerrate minimal ist.

Die Experimente wurden mit 3 verschiedenen Maschinen – Förderband, Industrieroboter, Mobiler Roboter (im Sinne der oben definierten Klassifizierung) – und mit 3 Versuchspersonen unterschiedlicher Größe durchgeführt. Die analysierten Verrichtungen umfassten Merkmalprüfung, Montage-, Schraub- und Andrücktätigkeiten, Objekthandhabungen im Sinne von Aufnehmen und Platzieren, Werkzeughandhabungen sowie Körperbewegungen. Einflussparameter waren Objektgeschwindigkeit, Bewegungspfad, Distanz zwischen Mensch und bewegtem Objekt und Anthropometrie der Versuchsperson.

3.2 Versuchsdurchführung

Die kollaborativen Tätigkeiten wurden jeweils nach einer 1-tägigen Übungsphase durchgeführt, um den Einfluss des Lernprozesses zu minimieren. Die Objektgeschwindigkeiten reichten von 50 – 400 mm/s in 50 mm/s-Schritten. Die Bewegungspfade wurden jeweils auf die Bewegungsdimension der Maschine angepasst. D.h. Bewegungspfade am Förderband umfassten eine 1-dimensionale Bewegung in der xy-Ebene bei gleichbleibender Objektorientierung,

Bewegungspfade des Mobilen Roboters umfassten eine 2-dimensionale Bewegung in der xy-Ebene bei einer möglichen geänderten Objektorientierung um die z-Achse, und Bewegungspfade des Industrieroboters umfassten 3-dimensionale Bewegungen im Raum bei ebenfalls möglicher geänderter Orientierung im Raum. Jedes Bewegungsset (mit entsprechender Maschine, Geschwindigkeit, Distanz, Pfad und Versuchsperson) wurde 10-mal wiederholt und mit 4 HD Kameras aus unterschiedlichen Positionen gefilmt.

3.3 Ergebnisse

Die Ergebnisse der durchgeführten Experimente zeigten deutlich einen Zielkonflikt zwischen operativen Faktoren und physischen Grenzen:

- Die Fehlerrate steigt mit steigender Objektgeschwindigkeit.
- Für jede kollaborative Tätigkeit existiert eine optimale Bewegungsgeschwindigkeit des Objektes.
- Die Position, der Bewegungspfad sowie die Bewegungsart (im Speziellen bei Bewegungen durch den Industrieroboter) des Objektes haben keinen Einfluss auf die kollaborative Tätigkeit im Sinne von Ausführungszeit und Fehlerrate.

Des Weiteren zeigten die Ergebnisse eine deutliche Änderung der Arbeitsgeschwindigkeit des Menschen in Abhängigkeit der Objektgeschwindigkeit. Abbildung 2 visualisiert das adaptive Verhalten des Menschen am Beispiel der Bauteilentnahme von einem Förderband. Nach MTM würde die Bauteilentnahme 2,717 Sekunden dauern, was in einer Objektgeschwindigkeit von rund 230 mm/s resultiert. Ändert man nun die Objektgeschwindigkeit, resultiert dies in einer veränderten Bewegungsgeschwindigkeit des Menschen. Bei größeren Objektgeschwindigkeiten muss der Mensch noch schneller reagieren und handeln, um das Objekt noch rechtzeitig aufnehmen zu können.

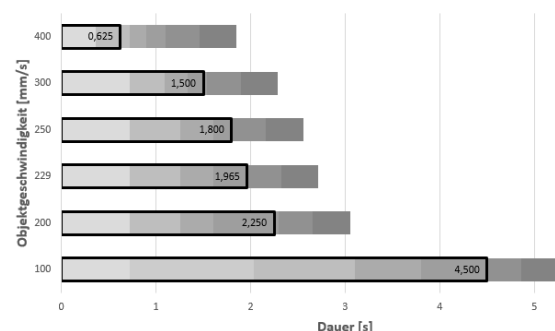


Abbildung 2. Dauer der Verrichtung ‚Bewegtes Objekt aufnehmen‘ bei veränderter Objektgeschwindigkeit

Auf Basis der ermittelten Zeitwerte wurden anschließend die einzelnen dynamischen Verrichtungen in der Simulationsumgebung *ema* abgebildet und mit den empirisch ermittelten Zeitwerten bzw. Grenzparametern hinterlegt (siehe dazu Abbildung 3).

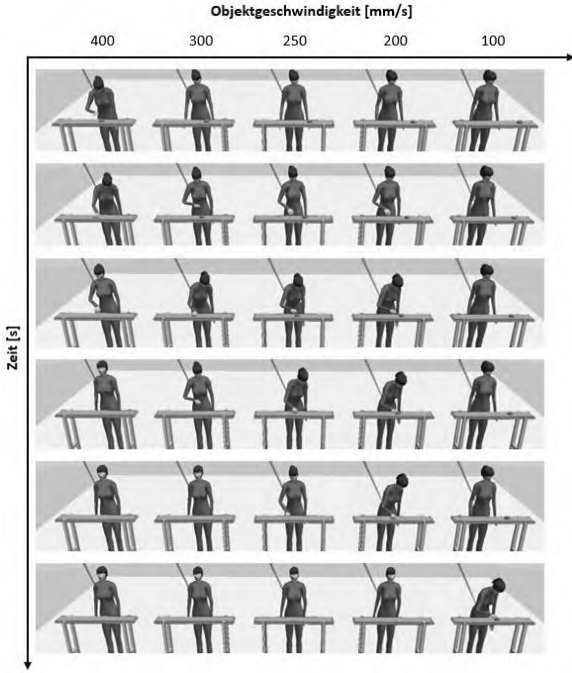


Abbildung 3. Modellierung der dynamischen Verrichtung ‚Bewegtes Objekt aufnehmen‘ in *ema*

4 Durchgängiger Modellentwurf

Im Vergleich zu bisherigen Parametern, die im Rahmen einer menschengerechten Arbeitsplatzgestaltung Berücksichtigung finden, müssen für Mensch-Maschine-Kollaborationen im Besonderen Parameter zur Arbeitsraumüberwachung, zur Funktion des Steuer- und Regelkreises sowie zu psychophysiologischen Wirkungen hinzugefügt werden.

4.1 Arbeitsraumüberwachung und Sicherheit

Wird die räumliche Trennung zwischen Mensch und Maschine im Rahmen der Kollaboration reduziert, müssen folgende Bewegungsparameter der Kollaborationspartner ortsunabhängig überwacht werden:

- Position des Menschen \mathbf{p}_H
- Bewegungsgeschwindigkeit des Menschen \mathbf{v}_H
- Beschleunigung des Menschen \mathbf{a}_H
- Position der Maschine bzw. des bewegten Objektes \mathbf{p}_M
- Geschwindigkeit der Maschine \mathbf{v}_M
- Beschleunigung der Maschine \mathbf{a}_M

Abbildung 4 zeigt den Positionsvektor \mathbf{p}_M des Objektes in Abhängigkeit der Manipulation durch eine der oben genannten Maschinen M . Je nachdem, ob das Objekt durch ein Förderband C , eine Mobile Plattform P oder einen Industrieroboter R bewegt wird, besitzt \mathbf{p}_M zwei, drei oder sechs Parameter. Der Positionsvektor des Menschen \mathbf{p}_H umfasst hingegen drei Parameter für die Position und einen für die Orientierung.

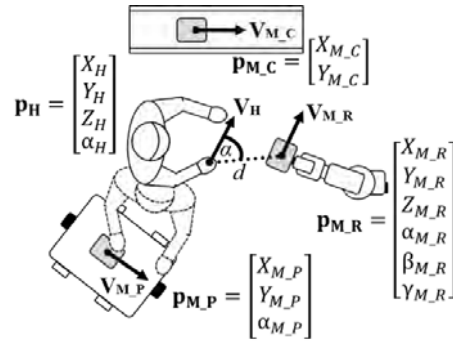


Abbildung 4. Prozessgrößen

Die resultierenden Prozessgrößen sind demnach:

- Relativer Abstand zwischen Mensch und Maschine d
- Relativer Annäherungswinkel zwischen Mensch und Maschine α
- Relativgeschwindigkeit \mathbf{v}_{MH}
- Relativbeschleunigung \mathbf{a}_{MH}

Der momentane Abstand zwischen Mensch und Maschine lässt sich über die Positionsvektoren der jeweiligen Kollaborationspartner berechnen:

$$d = |\mathbf{p}_{MH}| = |\mathbf{p}_H - \mathbf{p}_M| \quad (2)$$

Der Annäherungswinkel α wird über den momentanen Abstandsvektor und den momentanen Vektor der Relativgeschwindigkeit aufgespannt.

$$\alpha = \arccos \left(\frac{|\mathbf{p}_{MH} \cdot \mathbf{v}_{MH}|}{|\mathbf{p}_{MH}| \cdot |\mathbf{v}_{MH}|} \right) \quad (3)$$

Die momentane Geschwindigkeit eines Objektes, das durch ein Förderband manipuliert wird, wird über die Ableitung der momentanen Positionsdaten ermittelt.

$$\mathbf{v}_{M_C} = \frac{d\mathbf{p}_{M_C}}{dt} \quad (4)$$

Im Falle einer radgeführten Mobilen Plattform lässt sich der Geschwindigkeitsvektor eines Rades i mithilfe der translatorischen Geschwindigkeit der Plattform

\mathbf{v}_{M_P} und der Winkelgeschwindigkeit $\boldsymbol{\omega}_{M_P}$ um die jeweilige vertikale Radachse beschreiben, wobei \mathbf{w}_i dem jeweiligen Lagevektor des i -ten Radschwerpunktes entspricht.

$$\mathbf{v}_{M_{Pi}} = \mathbf{v}_{M_P} + \boldsymbol{\omega}_{M_P} \times \mathbf{w}_i \quad (5)$$

Im Falle des Industrieroboters wird die Änderung der Position und Orientierung \mathbf{p}_{M_R} des Objektes mithilfe der sogenannten Jacobi-Matrix berechnet. Die Jacobi-Matrix stellt dabei einen Zusammenhang zwischen der Änderung einzelner Gelenkvariablen \mathbf{q} des Roboters und der Geschwindigkeit des Maschinenarbeitspunktes her.

$$\mathbf{v}_{M_R} = \frac{d\mathbf{p}_{M_R}}{dt} = \mathbf{J}(\mathbf{q}) \frac{d\mathbf{q}}{dt} = \mathbf{J}(\mathbf{q}) \cdot \dot{\mathbf{q}} \quad (6)$$

Die momentane Beschleunigung des Roboters wird über die Bewegungsgleichung einer kinematischen Kette bestimmt, wobei \mathbf{M} die Massenmatrix und $\boldsymbol{\tau}$ die in den Gelenken wirkenden Drehmomente beinhaltet. Des Weiteren beinhaltet \mathbf{b} mehrere Summanden, die die von der Gravitation verursachenden Gelenkdrehmomente bzw. -kräfte, Reibungsverluste sowie Zentripetal- und Corioliskräfte beschreibt.

$$\mathbf{a}_{M_R} = \dot{\mathbf{v}}_{M_R} = \mathbf{M}(\mathbf{q})^{-1} \cdot [\boldsymbol{\tau} - \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})] \quad (7)$$

Des Weiteren ist die Funktion des Steuer- und Regelkreises im Sinne einer sicherheitstechnischen Überwachung der Bedienperson als boole'sche Logik zu implementieren. Beispielsweise soll in Abhängigkeit der Distanz zwischen Mensch und Maschine die Maschinengeschwindigkeit reduziert werden. Bei Fehlverhalten der Bedienperson (Vergessen eines Arbeitsschrittes oder Unachtsamkeit) und eines darauf folgenden abrupten Eintretens in den Maschinenarbeitsbereich, soll die Maschine gestoppt werden.

4.2 Fehlverhalten und Psychophysiologie

Nach Leidholdt [16] lässt sich die simulative Absicherung von kollaborativen Arbeitsformen eines Mensch-Maschine-Systems entweder auf Basis von FMEA (Failure Mode and Effects Analysis) oder auf Basis von Fehlerwahrscheinlichkeiten durchführen. Im Rahmen einer Fehlermöglichkeiten- sowie einer Fehlereinflussanalyse werden bewusst Fehler und Fehlerkombinationen in den Fertigungsprozess integriert. Der Nachteil liegt allerdings darin, dass die FMEA auf die menschliche Vorstellungskraft in Bezug auf auftretbare Fehler beschränkt ist. Im Gegensatz dazu können Fehlerszenarien auch zufällig, einer

vorausgesetzten Wahrscheinlichkeit folgend, initiiert werden. Obwohl bei dieser Methode sehr vielfältige Handlungsszenarien berücksichtigt werden können, ist die sehr lange Simulationsdauer nachteilig.

In Bezug auf die psychophysiologische Wirkung menschlichen Verhaltens stellt Leidholdt und Wohlschläger [17] ein numerisches Verhaltensmodell des Menschen vor, das eine Verhaltensänderung bei erhöhter Beanspruchung abbildet. Das Modell fasst dabei Belastungen, Beanspruchungen und Menschentypen in Zahlen zusammen. Die Zahlenwerte wurden indirekt oder durch Schwarmintelligenz, die dahinterstehenden Formeln empirisch, ermittelt. Das Modell ist zwar plausibel, da die digital berechnete globale Beanspruchung des Menschmodells in der Simulation auf Arbeitsgeschwindigkeit, -genauigkeit, Fehlerwahrscheinlichkeit und Verweigerungswahrscheinlichkeit wirkt, aber noch nicht vollständig validiert.

4.3 Datenfusionierung im hybriden Modell

Das Modell zur Abbildung eines komplexen Mensch-Maschine-Systems lässt sich als hybrides Modell mithilfe des von Zeigler et al. [18] entwickelten Formalismus beschreiben. Hybride Systeme sind dabei Systeme, die sowohl einen diskreten als auch einen kontinuierlichen Anteil besitzen. In Bezug auf Mensch-Maschine-Systeme sind die Bewegungen der einzelnen Kollaborationspartner bzw. des durch die jeweilige Maschine manipulierten Objektes kontinuierliche Prozesse. Die Ein- und Ausgangswerte der Steuerung sowie Regel- und Grenzparameter sind allerdings diskret. Das hybride System der kollaborativen Tätigkeit wechselt also in Abhängigkeit des diskreten Eingangs und der Überwachungskenngrößen seinen Zustand.

So genannte DEV&DESS (Discrete Event and Differential Equation Specified Systems) werden mithilfe eines 11-Tupel beschrieben, das folgende Parameter beinhaltet:

- diskrete Eingangs- und Ausgangsparameter X^{discr} , Y^{discr} , wie z.B. Art der Schutzeinrichtung und des daraus resultierenden Handlungsregimes
- kontinuierliche Eingangs- und Ausgangsparameter X^{cont} , Y^{cont} , wie z.B. aktuelle Posen von Mensch und Maschine
- mögliche Systemzustände $S = S^{discr} \times S^{cont}$, wie z.B. Annäherung des Menschen an die Maschine, synchrone Bewegung, Fehlerzustand, Abstandshaltung

- externe und interne Zustandsänderungsfunktionen δ_{ext} , δ_{int} , wie z.B. Auslösen spezifischer Sicherheitsfunktionen aufgrund einer aktiven Schutzeinrichtung oder der Generierung einer Kollaborationsaufgabe
- diskrete und kontinuierliche Ausgangsfunktionen λ^{disc} , λ^{cont} , wie z.B. Bewegungsdurchführung, Durchführung der kollaborativen Tätigkeit oder Aufgabenerfüllung
- Änderungsfunktionen f , z.B. Änderung der Bewegungssteuerung aufgrund einer aktiven Sicherheitsfunktion sowie
- Zustandsfunktionen C_{int} , z.B. der Überprüfung, ob sich beide Kollaborationspartner im Kollaborationsraum befinden.

Die Arbeitsraumüberwachung erfolgt dabei in Form einer Bewertung der relativen Bewegungsparameter zwischen den einzelnen Kollaborationspartnern. Des Weiteren werden ergonomische Grenzparameter für kollaborative Betriebsformen von Richtlinien für ergonomische Arbeitsplatzgestaltung abgeleitet. Die in den Versuchsreihen empirisch ermittelten physischen Grenzparameter fließen neben den ergonomischen Richtwerten ebenfalls als Nichtlinearitäten in die hybride Systembeschreibung ein.

Der DEV&DESS-Formalismus, lässt sich, wie in Abbildung 5 dargestellt, auf ein kollaboratives System übertragen und in weiterer Folge als simulationsfähiges Modell implementieren. Aufgrund der aktuell noch laufenden Implementierung des Modells, können hierzu noch keine Ergebnisse veröffentlicht werden.

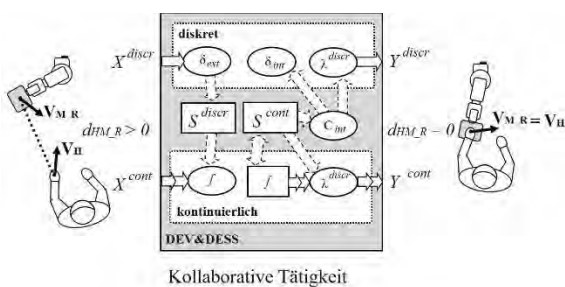


Abbildung 5. Hybride Systembeschreibung

5 Zusammenfassung und Diskussion

Besonders in Bezug auf die Sicherheit des Bedienpersonals ist eine simulationsgestützte Validierung und Verifizierung von kollaborativen Betriebsformen essentiell. Hinzu kommen zunehmende gesetzliche und normative Reglementierungen, die eine Berücksichtigung dieser Arbeitsform in Planungswerkzeugen

mit sich führt. Obwohl digitale Menschenmodelle bereits zur Planung manueller Tätigkeiten verwendet werden, sind Analyseszenarien auf die Handhabung statischer Objekte beschränkt. Zur Simulation kollaborativer Arbeiten müssen bestehende Modelle erweitert werden, sodass speziell gemeinsam mit Maschinen durchzuführende Arbeiten validierbar werden. Neben der Aufgabenaufteilung zwischen Mensch und Roboter zieht dies jedoch auch eine Berücksichtigung von bislang nicht implementierten Modellparametern mit sich, wie der Überwachung des Kollaborationsraums, der Richtigkeit der Arbeitsabfolge oder der eines Verhaltensmodells des Menschen.

Weiters lässt sich die Auslegung eines kollaborierenden Mensch-Maschine-Systems als multi-kriterielles Optimierungsproblem mit einer gewichteten Zielfunktion formulieren, was die Festlegung einzelner Modellparameter erschwert. Wird beispielsweise die Geschwindigkeit der Maschine erhöht, um eine höhere Taktzeit zu erhalten, steigt die Fehlerwahrscheinlichkeit des Menschen bei der Durchführung der kollaborativen Aufgabe. Offen bleibt in dem Beitrag, wie die systemspezifischen Parameter zu gewichten sind bzw. wie eine Normierung von menschen- und maschinenspezifischen Faktoren zu erfolgen hat, um eine Zielfunktion, wie der einer minimalen Taktzeit, zu erreichen.

Die Auswertung der empirischen Studie hat jedenfalls gezeigt, dass Menschen ihre Arbeitsgeschwindigkeiten auf die Objektgeschwindigkeiten anpassen und dass somit die aktuell noch hinterlegten MTM-Normzeitwerte für die Modellierung manueller Verrichtungen im Rahmen von dynamischen Arbeiten adaptiert werden müssen. Die aus den Studien ermittelten Grenzwerte für manuelle Bewegungsgeschwindigkeiten können folglich als Parameter für ein Systemmodell kollaborativer Arbeit verwendet werden.

Des Weiteren können durch die hybride Systembeschreibung, weitere Modellparameter für das Optimierungsproblem ermittelt werden. Der in diesem Paper vorgestellte hybride Beschreibungsformalismus muss jedoch noch auf Gültigkeit und Zuverlässigkeit geprüft werden. Beispielsweise muss das Modell in der Lage sein, zu entscheiden, ob die durch den Menschen vorgesehene Arbeit am bewegten Objekt, bei der aktuellen Maschinengeschwindigkeit und der relativen Position des Menschen zur Maschine, überhaupt physikalisch möglich ist. Des Weiteren darf die Maschine die Person auch bei einer fehlerhaften Prozessreihenfolge nicht verletzen. Am Anfang der

Forschung steht auch noch die digitale Modellierung des menschlichen Verhaltens. Es ist aber davon auszugehen, dass das vorgestellte hybride Modellierungsverfahren für kollaborative Betriebsformen valide genug ist, um einen sicheren und physikalisch sinnvollen Betrieb zu planen. Des Weiteren kann auf Basis der durchgeführten Experimente auch ein System entwickelt werden, das den Normleistungsbezug für Arbeiten an bewegten Objekten abbildet.

6 Referenzen

- [1] DIN EN ISO 10218-1: Industrieroboter – Sicherheitsanforderungen – Teil 1: Roboter, 2012.
- [2] DIN EN ISO 10218-2: Industrieroboter – Sicherheitsanforderungen – Teil 2: Robotersysteme und Integration, 2012.
- [3] ISO/TS 15066: Robots and robotic devices – Collaborative robots, 2016.
- [4] A. Bauer, D. Wollherr und M. Buss. *Human-Robot Collaboration: A Survey*. International Journal of Humanoid Robotics, S.47–66, 2008.
- [5] M.A. Goodrich, und A.C. Schultz. *Human-Robot Interaction: A Survey*. Foundations and Trends in Human-Computer Interaction, S. 203–275, 2007.
- [6] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier und P. Fraisse. *Collaborative manufacturing with physical human-robot interaction*. Robotics and Computer-Integrated Manufacturing, S. 1–13, 2016.
- [7] J. Kim, S. You, V. Kamat, S. Sanghyun Lee und L.P. Robert. *Evaluation of Human Robot Collaboration in Masonry Work Using Immersive Virtual Environments*. Proceedings of the 15th International Conference on Construction Applications of Virtual Reality, in Canada, S. 132–141, 2015.
- [8] M. Giuliani, N. Mirnig, G. Sollnberger, S. Stadler, R. Buchner und M. Tscheligi. *Systematic analysis of video data from different human-robot interaction studies: a categorization of social signals during error situations*. Frontiers in Psychology, 2015.
- [9] S. Stadler, A. Weiss, N. Mirnig, und M. Tscheligi. *Anthropomorphism in the factory – a paradigm change?* Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction, in Japan, S. 231–232, 2013.
- [10] S. Puls. *Situationsverstehen für die Risiko-beurteilung bei der Mensch-Roboter-Kooperation*. Dissertation, Karlsruher Institut für Technologie, Deutschland, 2015.
- [11] S. Thiernemann. *Direkte Mensch-Roboter-Kooperation in der Kleinteilmontage mit einem SCARA-Roboter*. Dissertation, Universität Stuttgart, Deutschland, 2005.
- [12] M. Bélanger-Barrette. *Collaborative Robot Ebook*. Robotiq, Canada, 2015.
- [13] J.T.C. Tan, F. Duan, R. Kato und T. Arai. *Collaboration Planning by Task Analysis in Human-Robot Collaborative Manufacturing System*. Advances in Robotic Manipulators, S. 113–132, 2010.
- [14] R. Bokranz und K. Landau. *Handbuch Industrial Engineering. Produktivitätsmanagement mit MTM*. Schäffer-Poeschel, Deutschland, 2012.
- [15] L. Fritzsche, R. Jendrusch, W. Leidholdt, S. Bauer, T. Jäckel und A. Pirger. *Introducing ema (Editor for Manual Work Activities) - A New Tool for Enhancing Accuracy and Efficiency of Human Simulations in Digital Production Planning*. Digital Human Modeling, S. 272–281, 2011.
- [16] W. Leidholdt. *Die Simulation von Mensch-Roboter-Kollaboration – unabdingbar für die Prozessgestaltung*. Produktion und Arbeitswelt 4.0, S. 109–115, 2014.
- [17] W. Leidholdt und C. Wohlschläger. *Das Numerische Verhaltensmodell des Menschen – Simulation von Werkern unter erhöhter Beanspruchung*. Tagungsband der 9. Fachtagung Digitales Engineering zum Planen, Testen und Betreiben technische Systeme, in Deutschland, S. 229–233, 2012.
- [18] B.P. Zeigler, H. Praehofer und T.G. Kim. *Theory of Modeling and Simulation. Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, USA, 2000.

A model factory in augmented reality as an eye-catcher at exhibitions and fairs

Torsten Munkelt¹, Steven Behne¹, Markus Wacker¹, Sven Völker²

¹Dresden University of Applied Sciences

²Ulm University of Applied Sciences

munkelt@informatik.htw-dresden.de

Our universities required an eye-catcher at exhibitions, conferences, open houses etc. We decided in favor of a model factory in augmented reality demonstrating a queueing system, material flow, and aspects of production planning and control. When visitors look at an empty table “through” their personal mobile devices, they can watch a factory producing on the table. The visitors can alter control variables and watch the factory change its behavior accordingly. We explain the underlying simulation model and its configuration, the composition and functionality of the mobile app for augmented reality, and the communication via database. Furthermore, we present a solution for the automatic generation of the model factory’s layout, and we explicate how to eliminate the blurring of the model factory in augmented reality. Finally, we share experiences and user feedback from first exhibitions of the model factory and outline our plans for its future development.

1 Motivation for an AR model factory

At industrial trade fairs, recruitment events and similar occasions, exhibitors compete for visitors’ attention. Most exhibitors try to contact as many visitors as possible because the overall number of contacts correlates with the number of leads. To initiate contact, an eye-catcher is needed which attracts visitors to the booth. Posters, on-screen presentations and even video clips do not serve this purpose well, since these types of media are omnipresent at fairs. The ideal eye-catcher should be physical, moving, and sending optical and acoustic signals (e.g. it should blink and beep), thus, appealing to multiple senses. According to experience, the eye-catcher often does not relate to the products or services of the exhibitor and attracts visitors nonetheless.

The Faculty of Computer Science/Mathematics of Dresden University of Applied Sciences inspired this work. The Faculty required an eye-catcher at trade fairs, exhibitions, conferences, open houses etc. The ultimate goal is to acquire new students for our universities, faculties, and courses of study.

The original idea was to build a transportable model factory. Since many universities already possess model factories illustrating industrial engineering, the new model factory was to focus on queueing systems and material flow instead. Hence, our eye-catcher illustrates some principles of production planning and control which relates the eye-catcher to our services,

and consequently makes it even more appealing to visitors.

Furthermore, it is required that the model factory

- can be influenced by setting structural and behavioral parameters, e.g. the number of machines, the routings, and the control strategy,
- produces perpetually, e.g. does not run empty,
- blinks and beeps occasionally,
- is easy to transport and to set up, and
- fits into the trunk of a station wagon.

When developing a concept for the model factory, the focus shifted from a physical model to a model factory in augmented reality (AR) due to several reasons: State-of-the-art technology of mobile devices and augmented reality frameworks allow the creation of an AR model factory with reasonable effort. Such a model factory is configurable according to the specific exhibition, and visitors can manipulate it directly without the risk of physical damage. Furthermore, there is currently no comparable AR model factory known, which gives us a “unique selling point.”

A realization of an AR model factory must fulfill the following basic requirements:

- Visitors can see the factory (machines and production orders in process) through their personal mobile devices (smartphones or tablet computers) but not with bare eyes.

- Mobile devices project the factory on the spatially restricted top of a conventional table.
- Production orders are visualized during their entire lifecycle, e.g. while they are moving from station to station, waiting in queues, and being processed at the machines.
- Visitors may set values of control parameters online on their mobile devices and observe the resulting effect on key performance indicators like adherence to delivery dates.

2 Types and applications of model factories

Regarding their technical basis, there are different types of model factories: virtual factories, downscaled physical factories, and life-size physical factories. Model factories serve different purposes, e.g. evaluation, demonstration, and education.

Virtual factories are digital simulation models that do not contain any physical components. They are widely used for evaluating alternative production concepts (e.g. [1]) as well as for educational case studies (e.g. [2]), since they can represent arbitrary complex production systems and are easy to modify.

Downscaled physical model factories consist of small physical models of factories' resources. The resource models are either simple blocks or functional resources that handle simplified "products" like small bricks. Components for creating this type of model factories are commercially available (e.g. [3]). Downscaled physical model factories typically serve educational purposes.

Life-size model factories are equipped with machines also used in real factories. Although they are usually restricted to a few workstations, these model factories require considerable space and are quite expensive. Their purpose is to demonstrate state-of-the-art technology (e.g. [4]), professional training (e.g. [5]), and research (e.g. [6]). Some of the life-size model factories use AR applications to provide manufacturing information to workers.

An investigation concerning AR based model factories did not come up with any existing solution similar to the one described in this paper.

3 Overview of the system architecture of the AR model factory

Figure 1 shows an overview of the system architecture of the AR model factory.

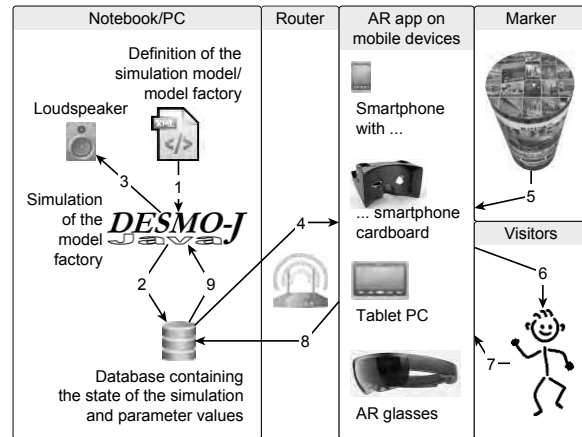


Figure 1: Overview of the system architecture

The system operates as follows: The simulation environment, DESMO-J, loads an XML configuration file (1), which contains all relevant parameters of the factory, e.g. machines, routings, and the product mix. DESMO-J automatically configures an executable simulation model based on these parameters. Then the simulation starts. Initially, the simulation writes master data to the database. As the simulation progresses, DESMO-J writes the current state of the model factory to the database and updates it at every change in state (2). Acoustic signals mark certain events during the simulation (3). A WLAN router connects the mobile device to the notebook/PC. The mobile device periodically polls the current state of the model factory from the database (4). The visitor points the camera of the mobile device to the marker standing on top of the table. Now, the app on the mobile device receives the camera's picture (5) and recognizes the marker's position in relation to the position of the camera. The app overlays the picture the camera is receiving with the visualization of the model factory according to the position of the marker in relation to the position of the camera. If the visitor looks at the table "through" the mobile device, she or he will see the model factory in its current state on top of the table (6). Figure 2 shows the visitor's view of the AR model factory.

The app updates the view in short intervals resulting in an animation of the production process. The visitor may change control parameters of the model factory

via the mobile device (7). The app writes the new parameter values to the database (8), the simulation system reads the new parameter values (9) and adopts its behavior accordingly. Then, the process starts anew, and the visitor watches the changed behavior of the AR model factory on the mobile device. Several visitors may view the model factory at the same time using their personal mobile devices.

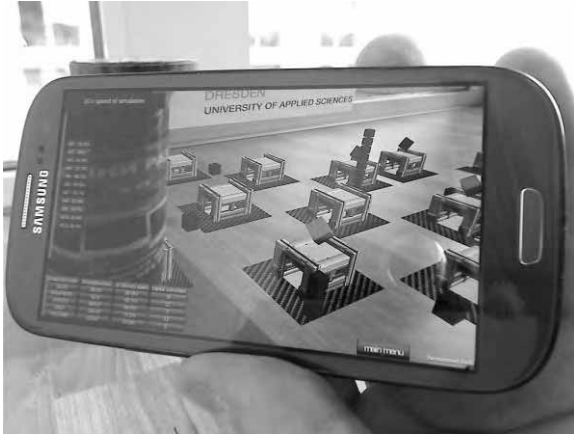


Figure 2: Visitor's view of the AR model factory

The following three sections describe the components of the system and their functionality in detail.

4 Event-based simulation of the model factory

The model factory is a job shop system (see [7]): An infinite stream of production orders enters the factory. Each production order requires a sequence of operations to be processed. Each operation is performed by a specific machine and requires a certain processing time. Each production order belongs to a certain product type assigned to a specific routing (i.e., a specific sequence in which the production order seizes the machines). A machine can process only one production order at a time, and one production order can only seize one machine at a time. While a machine is processing an order, other incoming orders enqueue in front of the machine. Every machine has its own queue with unlimited capacity. After an operation is completed, the production order moves on to the next machine. In our model, only the transportation distance influences the required transportation time. Transportation resources are not part of the model. For demonstrating the model factory, we applied a production system consisting of eleven machines and producing five types of products (see [8] und [9]). Figure 3 pictures the production system.

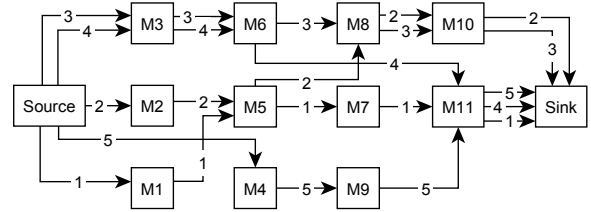


Figure 3: Production system containing eleven machines and the routings of five different products

Typically, production planning and control pursues four objectives, namely [10]

- adherence to delivery dates,
- short throughput times,
- high utilization of the machines, and
- low level of work in process.

Four key performance indicators (KPI) operationalize these objectives. The KPIs are calculated and updated during the simulation.

A number of parameters is necessary to describe the model factory:

- number of machines,
- distribution of interarrival times,
- product types, their probability, and their specific routings (sequences of operations processed on certain machines and requiring certain processing times),
- velocity of transport, and
- control strategy applied to select orders from waiting queues.

An administrator writes the values of these parameters to an XML file. This configuration file serves as input for generating an event-based simulation model.

A variety of simulation systems is able to simulate job shop systems. We chose DESMO-J (see [11] and [12]), version 2.5.1c, developed by the Department of Computer Science of Hamburg University. DESMO-J is an open-source object-oriented simulation framework written in JAVA™. JAVA™ is also used to realize the simulation models itself. DESMO-J supports process-oriented and event-oriented modeling [11]. We chose the latter because of its global view, its simplicity, and much lower computational costs.

While data-driven model generation is a well-known technique in simulation [13], there is one aspect that is worth mentioning: For sake of simplicity, the configuration file does not contain information about the physical location of the machines on the shop floor or a geometric description of the network of transporta-

tion routes. Hence, the factory layout has to be created automatically: An algorithm for layered drawing of directed acyclic graphs places the virtual machines relative to each other on the factory floor. The machines are the nodes of the graph, the transportation routes between consecutive machines are the directed edges of the graph. The algorithm aims at as few crossings of transportation routes as possible and at as little area as possible for the layout of the factory floor (see [14]). In order to achieve both, the algorithm sorts the machines topologically and assigns them to virtual layers (see [15]). Afterwards, the algorithm heuristically exchanges the machines at each layer to minimize the number of crossings of transportation routes (see [16]). All transportation routes are either parallel or perpendicular to each other and they may have perpendicular bends.

The model generator writes the coordinates of the machines and the transportation routes to the database (see section 6). This way, the AR app can access the geometric information regarding the factory layout.

Once the model generation is completed, the simulation starts and runs until manually terminated. Visitors can not only view the operating model factory but also change some parameters at runtime:

One parameter is the utilization of the production indicated by the utilization of the bottleneck machine. The system limits the utilization to 85 % because the factory and the queues should not overflow with production orders. When a visitor enters a new utilization, the system calculates an appropriate interarrival time of the production orders applying the probabilities of the product types and the routings. Thus, the system generates production orders more or less often, and after a certain time the utilization converges to the required value. [17] gives an example of the above calculations.

The visitor may also change the product mix. She or he can choose one of three options:

- the initial distribution read from the configuration,
- a uniform distribution (all product types appear with the same probability), or
- a distribution where the next product type is twice as probable as the current one, i.e., if there are product types $i=1, \dots, n$, the next arriving order will require product i with probability $p_i=2^{i-1}/(2^n-1)$.

Furthermore, the visitor can choose from several queueing strategies [18]:

- first come, first served,
- shortest processing time first,
- shortest slack time first, and
- highest value first.

All queues will be re-sorted when the global queueing strategy changes. The slack time and the values of the production orders are calculated and updated during the course of the simulation.

When a visitor changes a parameter value, the model factory needs time until it reaches a steady state again. The method “crossing of the means” (see [19]) indicates when this is the case. Until then, visitors cannot change parameter values further. This way, the visitor can experience the effects of her or his change on the KPIs representing the four objectives of production planning and control.

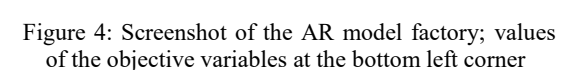
5 The composition and the functionality of the AR mobile app

Two frameworks support the visualization of the AR model factory. Unity is a runtime and development environment for 3D visualization and interaction (see [20] and [21]). It is a product of Unity Technologies, Inc. We applied free Unity Personal, Version 5.2.2. Vuforia is a software development kit for mobile devices that enables the creation of AR applications (see [22] and [23]). Vuforia was developed by the Qualcomm Developer Network and is now product of the Parametric Technology Corporation, Inc. We applied the non-commercial version Vuforia 5.5.9.

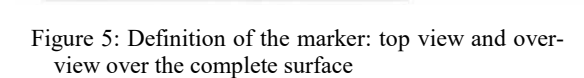
The 3D visualization of the model factory in Unity is straightforward: The machines are ready-made visual objects. The production orders are simple cubes differently colored in order to distinguish their routings/products. If necessary, they are stacked in front of the machines in order to resemble waiting queues. If a stack becomes too high, its order cubes are scaled down proportionally. If a machine is processing a production order, a spinning cube above the machine indicates that the machine is working.

The transportation routes are drawn as parallel or perpendicular lines with perpendicular bends if necessary. The transportation routes are colored according to the color of the routings/products. The data for the visualization stems from the simulation (see section 4). Unity calculates the movement of the production orders on the transportation routes, which results in a visualization of a smooth flow of the orders along the routes. Figure 4 shows the AR model factory

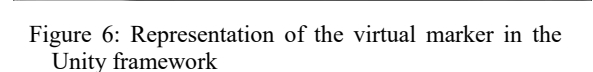
Unity also provides control widgets. We applied widgets to change the utilization of the model factory, to change the product mix, and to change the queueing strategies (see section 4). The widgets do not appear inside the 3D visualization of the factory but at the 2D graphical user interface (GUI) of the mobile app. Furthermore, the GUI displays the current values of the objective variables. This way, the visitor can observe how the change of the parameter values effect the objectives. Figure 4 shows the widgets and the display of the values of the objective variables.



When we developed the AR app, we decided on marker-based augmented reality. The app knows the marker beforehand, and thus, the app identifies the marker more easily. In marker-less AR settings, the app has to identify certain objects, like faces, without knowing every object beforehand, which is difficult and therefore computationally expensive [24].



The screenshot shows the top of the Unity interface. The title bar reads "Unity Personal - factoryvisualisation.unity - factoryvisualisation - Android* <DX11 on DX9 GPU>". Below the title bar is a menu bar with "File", "Edit", "Assets", "GameObject", "Component", "Hierarchy", "Window", and "Help". Below the menu bar is a toolbar with icons for selection, rotation, translation, and other tools. The "Pivot" and "Local" buttons are highlighted.



coordinates as reference. With the virtual camera, the Vuforia-part of the app then projects the virtual Unity model on the video stream, i.e. on the table (and the marker). The app finally delivers the augmented video stream to the visitor on the screen of her or his mobile device in real-time. The visitor gets the impression that she or he is watching the model factory on top of the table through the mobile device.

One major problem occurred when Vuforia overlaid the real-time picture with the 3D visualization: The visualization alternated abruptly and noticeably between two positions in consecutive resulting pictures. The reason for the undesirable blurring is a picture downscaling effect: For faster real-time pattern recognition of the marker in the real-world picture, Vuforia scales the input picture down to a size of 640×480 pixels. The remaining pixel array serves the detection of the edges of the marker. If the mobile device and its camera move only slightly, their position will repeatedly alter marginally in relation to the physical tracker and the detected edges of the marker in the picture will jump between adjacent pixels. Scaled up again, the distance between the marker's edges detected in one picture differs considerably from the edges detected in the next picture. Thus, Vuforia overlays the 3D visualization on the real-world pictures at different positions in consecutive pictures. Consequently, the 3D visualization jumps noticeably from one resulting picture to the next. We solved the problem by using the moving averages of the last five detected positions and orientations of the marker in the Vuforia software module when overlaying the real-world picture with the 3D visualization. This proved to be sufficient to avoid jumps of the 3D model in consecutive resulting pictures.

After compiling the app in Unity, it can be exported as Android Package (APK) for the Android operating system or as XCode for iOS.

6 The database for the communication between simulation and mobile app

The database is stored on the notebook and managed by MySQL. Both the simulator and the app are connected to the database via JDBC.

The database serves the bidirectional communication between the simulation and the AR app. In brief, the simulation writes master data and events to the database while the mobile app reads the master data and the events from the database and applies them to the

visualization. On the reverse channel, the mobile app writes control data to the database while the simulation reads the data from the database and adopts the simulation accordingly.

Figure 7 shows a simplified data model in crow's foot notation (see [26]). Initially, the simulation model generator writes master data to the database: machines and their locations, transportation routes with their starts, stops, and bends. As a simulation run proceeds, the relevant events of the order lifecycle are written to the database: entering the factory, enqueuing in a waiting queue, seizing a machine, releasing a machine, and leaving the factory. Moreover, the simulation writes objective values to the database. The simulation adds timestamps to all records. Database triggers aggregate, simplify, and prepare all data relevant for the visualization and distribute the data to small tables containing only very few records. Reading these tables is very fast. The mobile app reads the master data at the beginning of the simulation and creates the 3D visualization of the factory layout. Then, the mobile app periodically polls events from the database. Every period comprises a few frames. According to the events, the mobile app updates the 3D visualization of the production orders. As mentioned in section 5, the only data not provided by the database are the locations of the production orders on the transportation routes during transport. The mobile app continuously calculates these locations in relation to the length of the routes and the given velocity.

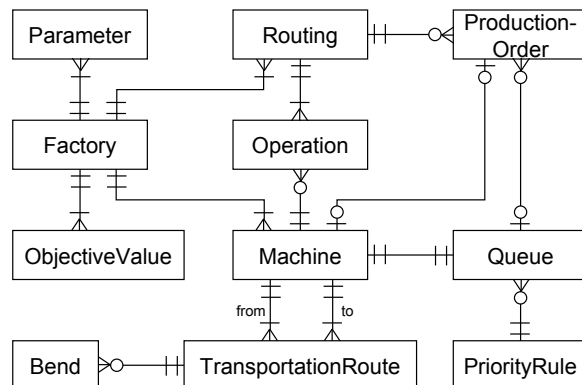


Figure 7: Simplified data model

When a visitor changes the value of a control parameter, the mobile app writes the change and a timestamp to the database. All connected apps read the new parameter values from the database, update the parameter values in their graphical user interface, and disable the controls until the simulation reaches a

steady state again. Thus, no visitor can change parameter values until then. If another visitor changes the parameter values in the meantime, the database will discard the changes because of the later timestamp. In parallel, the simulation also reads the new parameter values and adopts its behavior accordingly. Thereupon, the simulation writes newly occurring events to the database and updates the objective values. The mobile app reads the updated objective values from the database and displays them to the visitor. When the simulation reaches a steady state again, the simulation writes the information to the database and the mobile app reads the information and enables the controls.

7 Watch and operating the model factory through a personal mobile device

When exhibiting the AR model factory, we provide mobile devices, so visitors can watch the factory without their personal mobile devices. However, there may not be enough mobile devices for all visitors and some visitors might want to watch the factory through their personal devices.

In order to install the AR app on his or her own device, the visitor has to scan a QR code printed on a poster provided at the booth. The QR code encodes an URL referring to a website located on the local notebook/PC at the booth. Using the URL, visitors can download the installation files of the mobile app for different versions of different operating systems. Once the app has been installed, it can be used immediately to view and control the AR model factory.

8 Recent and future use of the AR model factory

We already exhibited prototypes of the AR model factory at the career-start fair (KarriereStart) in Dresden, during open houses at Dresden University of Applied Sciences in January and April 2016 respectively, and during Lange Nacht der Wissenschaften (Long Night of Sciences) in Dresden 2016. Visitors were very interested in the model factory. More than 150 of them watched the model factory on the table “through” our prepared tablet computers; more than 40 of them installed the mobile app on their own mobile devices. All visitors changed at least the utilization of the factory and watched the waiting queues lengthen or shorten. The feedback was positive without exception. The most frequent question asked by

visitors was if we could also visualize and control their real-life factories in AR in real-time.

The exhibition scheduled next is the career-start fair in Dresden in January 2017. When introducing Dresden University of Applied Sciences at high schools, we will also exhibit the AR model factory. We also plan to present the model factory during poster sessions at scientific conferences.

9 Summary and outlook

We described the development of an AR model factory. The overall goal was to build an eye-catcher we can apply at trade fairs, exhibitions, conferences, open houses etc. We achieved the goal in full extent by building the AR model factory. Evidently, the model factory stimulates the visitors to approach the booth and to gather information of the courses of study offered at our faculties/universities. Moreover, the model factory has a strong relation to a specific domain of business informatics, namely production planning and control: We can use the model factory to demonstrate some principles of production planning and control. The eye-catcher is not physical, but some of its parts are moving, and it sends optical and acoustic signals when certain events occur in the factory. In essence, the visitor can watch the model factory “through” her or his smartphone. The model factory seems to stand on top of a physically empty table and it is visibly processing production orders of different types. Visitors may manipulate the behavior of the model factory by changing the utilization, the product mix, and the queueing strategy. The changes result in different values of the objectives (like adherence to delivery dates or work in process), which the visitor can also monitor.

At a next stage of extension, the visitor could interact with the mobile app via speech and gestures but speech and gesture recognition might slow down the visualization. If the simulation and the database run on a remote server, the notebook/PC will become obsolete at the booth and we will need the marker only. The internet connection should be broad and stable for this scenario, which is often not the case at trade fairs. The visualization could display the number of waiting orders above the queues. Furthermore, the visualization could display the changes of the objective values in diagrams. A printout of the factory layout and physical building blocks representing the machines could serve as a physical model of the factory. This way, the model factory would interlace

reality and AR even more and it would more closely resemble a physical model factory.

10 References

- [1] L. März et al.: *Simulation und Optimierung in Produktion und Logistik*. Springer, 2011.
- [2] S. Bickel et al.: *The e-Beer Game*. In: I. Mayer et al.: *Organizing and Learning Through Gaming and Simulation*, Eburon, Netherlands, pp. 39-46, 2007.
- [3] <http://www.fischertechnik.de/en/Home.aspx>
- [4] F. Mayer et al.: *Deutschlandweiter I4.0-Demonstrator*. Lehrstuhl für Automatisierung und Informationssysteme, TU München, 2013.
- [5] <http://www.festo-didactic.com>
- [6] <http://www.arena2036.de>
- [7] J. Blazewicz, W. Domschke, E. Pesch: *The Job Shop Scheduling Problem: Conventional and New Solution Techniques*. European Journal of Operational Research 93, pp. 1-33, 1996.
- [8] R. Schulz: *Parallele und Verteilte Simulation bei der Steuerung komplexer Produktionssysteme*. Technische Universität Ilmenau, Diss., 2002.
- [9] O. Bube: *Architektur und Implementierung von Prozeßidentifikatoren*. Ilmenau, Technische Hochschule Ilmenau, Diss., 1993.
- [10] J. A. Buzacott, H. Corsten, R. Gössinger, H. M. Schneider: *Production Planning and Control: Basics and Concepts*. Oldenbourg, 2012.
- [11] B. Page et al.: *The Java Simulation Handbook: Simulating Discrete Event Systems with UML and Java*, Shaker, Aachen, 2005.
- [12] B. Page et al.: *Objektorientierte Simulation in Java mit dem Framework DESMO-J*, Books on Demand, 2000.
- [13] S. Bergmann: *Automatische Generierung adaptiver Modelle zur Simulation von Produktionssystemen*. Technische Universität Ilmenau, Diss., 2013.
- [14] G. D. Battista et al.: *Graph Drawing: Algorithms for the Visualization of Graphs*. Upper Saddle River, NJ: Prentice-Hall, 1998.
- [15] E. G. Coffman, R. L. Graham: *Optimal Scheduling for Two-Processor Systems*. In: Acta Informatica 1, p. 200–213, 1972.
- [16] W. Barth, M. Jünger, P. Mutzel: *Simple and Efficient Bilayer Cross Counting*. In: Journal of Graph Algorithms and Applications 8 (2004) 2, pp. 179-194, 2004.
- [17] T. Munkelt: *Potenzial Bayes'scher Netze zur Unterstützung der Produktionsplanung und -steuerung*. Technische Universität Ilmenau, Diss., 2009.
- [18] A. Shtub, R. Karni: *ERP: The Dynamics of Supply Chain and Process Management*, Springer, 2010.
- [19] P. S. Mahajan, R. G. Ingalls: *Evaluation of Methods Used to Detect Warm-up Period in Steady State Simulation*. In: R. G. Ingalls et al. (eds.): *Proceedings of the Winter Simulation Conference*, pp 663-671, 2004.
- [20] W. Goldstein: *Unity game development essentials*, Packt Publishing Ltd, 2009.
- [21] R. H. Creighton: *Unity 3D Game Development by Example: A Seat-of-Your-Pants Manual for Building Fun, Groovy Little Games Quickly*. Packt Publishing Ltd, 2010.
- [22] F. Kwik, R. Bahana: *Using Augmented Reality to Enhance Aetherpet, a Prototype of a Social Game*. ICCSCI 2015, pp. 282-290, 2015.
- [23] N. Imbert et al.: *Adding Physical Properties to 3D Models in Augmented Reality for Realistic Interactions Experiments*. International Conference on Virtual and Augmented Reality in Education, pp. 364-369, 2013.
- [24] H.-L. Chi et al: *Research trends and opportunities of augmented reality applications in architecture, engineering, and construction*. Automation in construction, pp. 116-122, 2013.
- [25] <https://developer.vuforia.com/target-manager>
- [26] P. Rob et al.: *Database Systems: Design, Implementation & Management*. Cengage Learning EMEA, London, 2008.

Iterative Entwicklung und Verbesserung eines dynamischen Prozesssimulationsmodells eines Aluminiumschachtschmelzofens anhand globaler Prozessgrößen

Matthias Henninger¹, Wolfgang Schlüter¹

¹Hochschule Ansbach

matthias.henninger@hs-ansbach.de

Im Green Factory Bavaria-Teilprojekt „Smart Melting“ werden Möglichkeiten zu Energieeffizienzsteigerung in der Schmelz- und Druckgussindustrie untersucht. Für eine Erhöhung des Prozessverständnisses und Überprüfung von Verbesserungskonzepten wird ein Energiemodell der Schachtschmelzöfen entwickelt.

Als Grundlage der Modellbildung werden der Aufbau und die Funktionsweise des Schachtschmelzofens, sowie die messbaren Größen erläutert. Aufbauend darauf wird die Auswahl des dynamischen Prozesssimulationsansatzes, welcher einer Kombination aus Wärmetauscheransatz und Methode der Blockkapazität entspricht, beschrieben. Im Rahmen der Modellbildung werden diese Ansätze, sowie deren Vor- und Nachteile erklärt. Zur Verbesserung des Grundansatzes wird ein Verbesserungsansatz erarbeitet, welcher auf einem Vergleich zwischen den Mess- und Simulationsergebnissen basiert. Die Vergleichsergebnisse werden anschließend analysiert und daraus fehlende Zusammenhänge zwischen Prozessgrößen abgeleitet. Die Quantifizierung der Zusammenhänge basiert auf Ergebnissen der Strömungssimulation, thermodynamischen Grundlagen und der iterativen Anpassung auf Grundlage von Messwerten.

Die Qualität des Modells wird anhand eines qualitativen und quantitativen Vergleichs zwischen Mess- und Simulationsergebnissen untersucht und bestätigt. Die mit dem validierten Modell durchgeführte Simulationsstudie zur Auswirkung der Schachtfüllung auf die Energieeffizienz zeigt, dass der Schachtfüllstand großen Einfluss auf die Effizienz des Ofens besitzt. Durch eine Erhöhung des Schachtfüllstandes kann die Schmelzleistung um bis zu 50 % erhöht und der spezifische Energieverbrauch um bis zu 28 % verringert werden.

1 Einleitung

Um am Weltmarkt bestehen zu können ist es insbesondere für energieintensive Unternehmen, wie zum Beispiel Druckgussbetrieben, überlebensnotwendig einen hohen Automatisierungsgrad und eine hohe Energieeffizienz zu erreichen. Dies gilt auch für die Nicht-Eisen-Gussindustrie, welche im Fokus der vorliegenden Studie steht. In Deutschland werden jährlich ungefähr 5,9 Millionen Tonnen Metall eingeschmolzen und vergossen, davon sind ca. 20 % Nicht-Eisen-Werkstoffe. Bei einem abgeschätzten Energieeinsatz von 2000 kWh pro Tonne gutem Guss ergibt sich dafür ein jährlicher Energieverbrauch von 2,2 Mrd. kWh [1]. Nach Daten des Statistischen Bundesamtes betragen die Kosten für Energie in der Gießerei-Industrie 25 % der Bruttowertschöpfung.

Im Zuge dessen werden im Forschungsprojekt „Smart-Melting“, einem Teilprojekt des bayerischen Forschungsverbundes „Green Factory Bavaria“, neue Wege zur Steigerung der Energieeffizienz in der Aluminium Schmelz- und Druckgussindustrie untersucht. Der Schwerpunkt des Projektes liegt auf einer

Verringerung des Energieverbrauchs von gasbetriebenen Schmelzöfen durch eine verbesserte Vernetzung und Abstimmung der Betriebsteilbereiche und einer damit verbundenen gleichmäßigeren Auslastung der Schmelzöfen. Um dies realisieren zu können muss der komplette Schmelz- und Druckgussbetrieb erfasst und analysiert werden. Die wesentlichen Betriebsbereiche sind die Schmelzerei, in dieser wird das flüssige Aluminium zu Verfügung gestellt, und die Gießerei, in welcher das Aluminium verarbeitet wird. Die Druckgussanlagen sind somit entscheidend für die Nachfragesituation und die Schmelzöfen entscheidend für die Angebotssituation des Flüssigaluminiums. Der Fokus der Betriebsführung liegt im Allgemeinen sehr stark auf der Produktion und kaum auf dem energieeffizienten Schmelzen des Aluminiums. Um die Zusammenhänge zwischen den verschiedenen Prozessschritten erfassen und verstehen zu können und eine verbesserte Betriebsweise der Schmelzöfen zu ermöglichen wird ein Simulationsmodell des gesamten Druckgussbetriebs erarbeitet. Die detaillierte Simulation der Vorgänge in den Schmelzöfen erfolgt durch das sogenannte Energiemodell. Auf die-

sem Modell liegt der Schwerpunkt in der vorliegenden Arbeit.

2 Grundlagen

2.1 Schmelzöfen

Eine essentielle Voraussetzung für die Entwicklung des Modells ist ein Grundverständnis des Aufbaus und der Funktionsweise des abzubildenden Schachtschmelzofens. Die Einführung erfolgt anhand des in Abbildung 1 dargestellten Schmelzofens mit kurzem Schacht. Über die Chargiereinrichtung wird das feste Aluminium in den sogenannten Schmelzschacht eingebracht. Dieser Vorgang, der durch eine Vorrichtung in der Schachtdecke erfolgt, wird häufig als Beschickung bezeichnet. Damit während der Beschickung das Rauchgas nicht darüber entweicht, müssen die Brenner während einer Beschickung abgeschaltet werden. Im Schmelzschacht wird das feste Aluminium aufgewärmt und geschmolzen. Die dafür benötigte Energie wird durch die Verbrennung von Erdgas und dem daraus resultierenden Rauchgas zu Verfügung gestellt. Das Rauchgas tritt mit maximaler Temperatur aus den Schmelzbrennern in den Schmelzschacht ein. Aufgrund von Wärmeübertragungsvorgängen nimmt der Energieinhalt des Rauchgases während des Umströmens des Aluminiums ab, bis dieses mit minimaler Temperatur in Richtung Ofenwanne strömt. Das Aluminium hingegen wird mit Umgebungstemperatur eingebracht und verlässt den Schmelzschacht in flüssiger Form mit maximaler Temperatur. Das geschmolzene Aluminium fließt aus dem Schmelzschacht in den Warmhaltebereich. Hier wird das Aluminium durch Wärmezufuhr auf eine definierte Temperatur überhitzt und auf dieser gehalten. Dafür wird das Rauchgas der Schmelz- und Warmhaltebrenner verwendet. Das beschriebene Verhalten von Aluminium und Rauchgas entspricht grob dem Verhalten der Medien in einem Gegenstromwärmeübertrager.

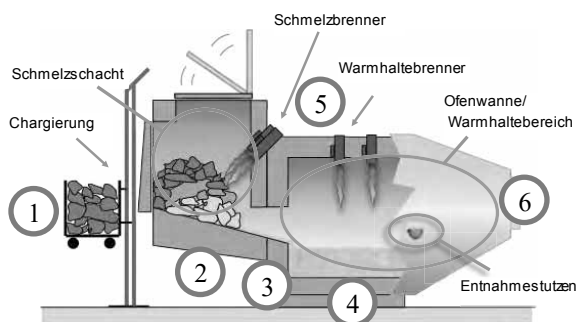


Abbildung 1. Schema Aluminium-Schmelzofen

2.2 Messgrößen und Messstellen

Für die Modellerstellung und deren Validierung wird zunächst überprüft, welche Größen messtechnisch erfasst werden können. In Abbildung 1 sind die sechs vorhanden/möglichen Messstellen eingetragen. In Tabelle 1 sind die messbaren Größen mit dem entsprechenden Messintervall angegeben. Die Messungen an den Stellen 2 bis 6 sind automatisiert und werden für die interne Steuerung des Ofens herangezogen. Die Messung der Zustandsgrößen an Messpunkt 1 ist nicht automatisiert, sondern muss manuell aufgenommen und abgeschätzt werden. Diese Größen werden im aktuellen Ofenbetrieb nicht berücksichtigt sondern werden ausschließlich für die Simulation benötigt. Neben den erfassten Größen sind feste Einstellungen/Rahmenbedingungen wie zum Beispiel das Luftverhältnis der Verbrennung oder die Ofenkonstruktion vorhanden.

Für die Simulation wären insbesondere exaktere Messungen der Brenner und der zugehörigen Brennersteuerung hilfreich. Diese muss aktuell aufgrund der erfassten Messwerte, wie dem Erdgasverbrauch der Schmelz- und Warmhaltebrenner und der Rauchgastemperaturen, angenähert werden.

Tabelle 1: Verfügbare Messgrößen

Messpunkt	gemessene Größe	Intervall [s]
1	Temperatur der Beschickung	--* ¹
1	Masse Beschickung	--* ¹
1	Spezifische Oberfläche	--* ¹
1	Zeitpunkt/Zeitdauer Beschickung	--* ¹
2	Aluminiummasse im Schmelzschacht	300
3	Rauchgastemperatur am Schachtaustritt	120
4	Aluminiummasse Ofenwanne	300
4	Aluminiumtemperatur Ofenwanne	300
5	Erdgasverbrauch der Schmelz- und Warmhaltebrenner	300
6	Rauchgastemperatur Ofenaustritt	120

*¹Die Erfassung erfolgt händisch und nur bei Bedarf

3 Modellbildung

3.1 Auswahl des Simulationskonzeptes

Mit dem Simulationsmodell müssen unterschiedliche Rahmenbedingungen und Zeitabschnitte von bis zu einer Kalenderwoche simuliert werden können. Dies führt dazu, dass eine hohe Flexibilität in Bezug auf die äußeren Rahmenbedingungen sowie ein geringer Rechenaufwand elementare Voraussetzungen des zu entwickelnden Modells sind.

Weitere wichtige Punkte für die Modellauswahl sind die verfügbaren Messwerte und die zu simulierende Problematik. Wie in Tabelle 1 gezeigt, sind lediglich globale Messwerte und keine Detailwerte vorhanden. Eine physikalische Modellierung ist aufgrund der vielen komplexen Teilvorgängen, wie der Verbrennung des Erdgases, der Strömung des Rauchgases, der Bewegung des festen Aluminiums, der Wärmeübertragung (Leitung, Konvektion, Strahlung) und der Beschickung, hochkomplex und mit der geforderten hohen Flexibilität und dem geringen Rechenaufwand nur bei einer globalen Betrachtung der Problematik möglich.

Verringert man die Komplexität des Problems, indem man stationäre Rahmenbedingungen vorgibt können Strömungssimulationsmodelle aufgebaut und in akzeptabler Zeit durchgeführt werden. Anhand dieser Simulationen können bestimmte Parameter wie z.B. der konvektive Wärmeübergang in Abhängigkeit der Brennerleistung und der Materialschichtung bestimmt werden. Diese Ergebnisse bilden dann in Kombination mit den Messwerten die Grundlage der Entwicklung und Parametrierung des dynamischen Prozesssimulationsmodells.

Gegen den Einsatz eines CFD-Tools als Primärtool sprechen die Nachteile einer sehr hohen Rechenzeit, einer hohen Modellkomplexität, ein hoher Implementierungsaufwand und geringe Flexibilität in Bezug auf die äußeren Rahmenbedingungen. Für die Implementierung wird aufgrund der Vielseitigkeit Matlab mit den Zusätzen Simulink und State-Flow gewählt.

3.2 Grundlagen Schmelzofenmodell

Als Grundlage der Modellierung in der vorliegenden Arbeit dient das von Schmidt erarbeitete Modell [2]. Für die Modellierung wird der Schmelzofen in die zwei Bereiche Schmelzschacht (Region 1) und Ofenwanne (Region 2) eingeteilt (Abbildung 2). In der Abbildung sind die Ein- und Ausgänge der Massenströme des Rauchgases (gelb) und des Aluminiums

(blau) gekennzeichnet. In der vorliegenden Arbeit wird nur die erweiterte Modellierung des Schmelzschachtes beschrieben, für welchen ein kombinierter Modellansatz verwendet wird.

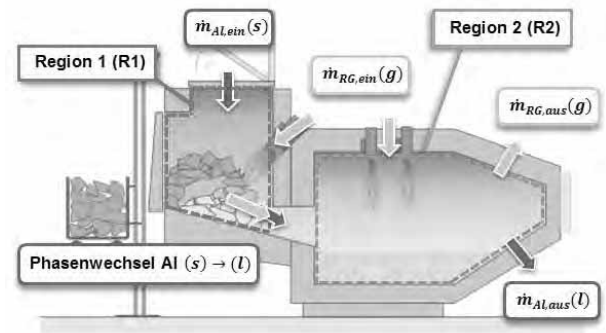


Abbildung 2: Einteilung Ofenmodell

3.3 Annahmen und Vereinfachungen

Um den Schmelzschacht mit dem gewählten Ansatz der dynamischen Prozesssimulation modellieren zu können und die geforderten Kriterien zu erfüllen müssen Annahmen und Vereinfachungen des realen Problems vorgenommen werden. Die wichtigsten sind:

- Idealisierte Temperaturverteilung des Aluminiums
- Schmelzvorgang wird nur energetisch erfasst
- Idealisierte Verbrennung
- Mittelwertbildung für Materialparameter
- Vernachlässigung der Wärmespeicherfähigkeit der Ofenwände
- Idealisierte Berechnung des Wärmestroms zwischen den Medien

3.4 Modellansatz

Die Modellierung erfolgt über eine Kombination aus einem Gegenstromwärmetauscheransatz [3] und der Methode der Blockkapazität, bei der für die betrachtete Größe im gesamten Bereich ein identischer Wert vorliegt [4]. Aufgrund der Temperaturdifferenz und der vorliegenden Grenzfläche zwischen Rauchgas und Aluminium findet Wärmeübertragung durch Konvektion und Strahlung statt. Als Konsequenz aus dem auftretenden Wärmestrom wird das kühlere Medium bei Durchströmen erwärmt und das wärmere abgekühlt. In Abbildung 4 (links) ist der Rauchgasstrom von Eintritt in Richtung Austritt und die

damit verbundene Temperaturabnahme dargestellt. Im Schmelzschacht wird das Aluminium vom Rauchgas von rechts angeströmt, bis dieses auf der linken Seite austritt. Im rechten Bild sind die wichtigsten Größen des Wärmetauscheransatzes eingetragen. Über den Wärmetauscheransatz werden die Energie- und Massenbilanzen des Aluminiums und des Rauchgases eingeführt.

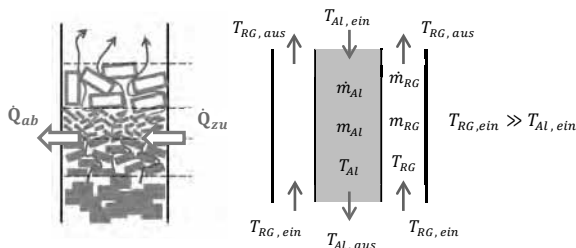


Abbildung 3. Modell Schmelzschacht

Für die Berechnung des Wärmestroms zwischen Aluminium und Rauchgas wird die mittlere logarithmische Temperaturdifferenz verwendet. Diese stellt anhand eines idealisierten und gemittelten Temperaturprofils eine Verbindung zwischen den Ein- und Austrittstemperaturen des Aluminiums und des Rauchgases her.

Durch die flexible Modellierung können unterschiedliche Materialgeometrien mit verschiedenen Temperaturen ohne Änderungen am Modell betrachtet werden.

Die Nachteile des räumlich gemittelten Ansatzes sind:

- keine örtliche Auflösung der Prozessgrößen
- physikalisch ungenaue Abbildung
- zwingend erforderliche Validierung
- für Detailbetrachtungen ungeeignet

Aufgrund der vorher beschriebenen Nachteile und der getroffenen Annahmen und Vereinfachungen ist eine Anpassung und Überprüfung des Modells anhand von Messwerten obligatorisch.

4 Weiterentwicklung des Modells

Um die beschriebenen Nachteile des kombinierten Ansatzes, wie die fehlende örtliche Auflösung einzelner Prozessgrößen und die unzureichende physikalische Abbildung zu verringern, muss dieser erweitert werden. Mit der Erweiterung werden im Grundansatz vernachlässigte Verknüpfungen zwischen Prozessgrößen und Rahmenbedingungen eingeführt. Ziel ist eine

verbesserte Erfassung der realen Bedingungen/Vorgänge innerhalb des Schmelzofens. Grundlage des Ansatzes ist die Analyse vorhandener Messdaten sowie der Ergebnisse der Strömungssimulationen. Aus den Erkenntnissen der Analysen werden Zusammenhänge zwischen Prozessgrößen hergestellt und iterativ angepasst. Für die iterative Anpassung werden die vorhandenen Messergebnisse übergeordneter, auch als global bezeichnete, Prozessgrößen wie zum Beispiel die Rauchgastemperatur am Schachtaustritt (Messpunkt 3) verwendet. Diese Größen werden als global (übergeordnet) bezeichnet, da diese das Ergebnis einer Vielzahl von Teilfunktion abbilden. Somit kann anhand dieser Prozessgrößen die Genauigkeit modellierter Teilfunktionen abgeschätzt und verbessert werden. Bei der genannten Rauchgastemperatur am Schachtaustritt können zum Beispiel Rückschlüsse zur Brennersteuerung und der Wärmeübertragung gezogen werden.

4.1 Modellverbesserung

Der Ablauf der Verbesserung im vorliegenden Projekt ist in Abbildung 5 dargestellt. Es ist zu erkennen, dass jeder Verbesserung eine Validierung und Analyse des Modells voraus geht. Anhand der Analyse werden Schwachstellen aufgedeckt und eine Verbesserung entwickelt und implementiert.

Für die Implementierung im Modell müssen die physikalischen Zusammenhänge quantitativ erfasst werden. Dies erfolgt anhand einer Abschätzung der realen Verhältnisse auf Grundlage der Messergebnisse, der Strömungssimulationsergebnisse und wissenschaftlicher Veröffentlichungen. Dies ist nötig, da die Kopplungen zwischen den Größen nicht oder schwer messbar sind. Die Auswirkungen der eingeführten Zusammenhänge werden anhand des Vergleichs der globalen Prozessgrößen zwischen Simulation und Messung betrachtet. Anhand der daraus gewonnen Erkenntnisse kann iterativ nachjustiert werden, bis die ideale Einstellung für die betrachtete Kopplung erreicht wird.

Das Einführen der physikalischen Zusammenhänge wird genutzt um die Problematik der fehlenden örtlichen Auflösung des Simulationsansatzes zu kompensieren. So wird beispielsweise eine idealisierte Temperaturverteilung innerhalb des Aluminiums eingeführt. Das Vorgehen wird nun beispielhaft anhand der realisierten Berücksichtigung der Strömungsverhältnisse erklärt.

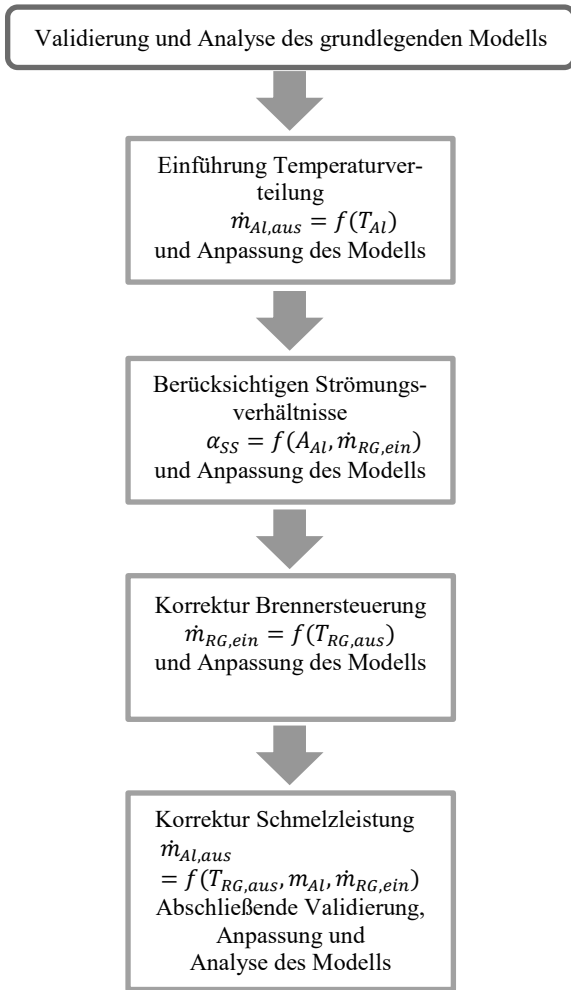


Abbildung 4: Ablauf der Modellverbesserung

4.2 Umsetzung am Beispiel Wärmeübergangskoeffizient

Es werden die genannten Prozessschritte anhand des mittleren Wärmeübergangskoeffizienten (α_{SS}) beschrieben. Dieser nimmt großen Einfluss auf die Wärmeübertragung und somit die messbare Rauchgastemperatur am Schachtende (Messpunkt 3). Im Grundansatz ist für α ein konstanter Wert vorgegeben.

Der Vergleich zwischen der gemessenen Rauchgastemperatur am Schachtende und der simulierten differieren insbesondere bei hohen und niedrigen Schachtfüllständen stark. Dies liegt daran, dass der konstante Wärmeübergangskoeffizient nur für bestimmte Strömungsverhältnisse korrekt ist. Die Strömungsverhältnisse werden insbesondere durch die Brennerleistung und den Schachtfüllstand beeinflusst. Deshalb wird im Folgenden ein Zusammenhang zwischen Wärmeübergangskoeffizient, Rauchgasmassenstrom $\dot{m}_{RG,ein}$

und wärmetauschender Oberfläche A_{Al} hergestellt (Formel 1).

$$\alpha_{SS} = f(A_{Al}, \dot{m}_{RG,ein}) \quad (1)$$

Dieser Zusammenhang muss auch quantifiziert werden. Problematisch ist in diesem Fall, dass dieser messtechnisch nicht direkt erfasst werden kann. Es können jedoch Annahmen und Annäherungen auf Grundlage von stationären Strömungssimulationen und thermodynamischen Grundlagen getroffen werden. Auf Basis dieser Erkenntnisse werden die Zusammenhänge zwischen α_{SS} und A_{Al} , sowie α_{SS} und $\dot{m}_{RG,ein}$ mathematisch beschrieben.

Das Ergebnis der Modellierung von α_{SS} ist in Abbildung 5 dargestellt. α_{SS} ist besonders hoch bei geringem Schachtfüllstand und hohem Rauchgasmassenstrom (hohe Brennerleistung). Um das Modell zu verbessern und eine erhöhte Übereinstimmung mit dem realen Prozess zu erhalten, werden die Simulationsergebnisse mit gemessenen Werten verglichen. Aufbauend auf diesem Vergleich wird die mathematische Erfassung des Zusammenhangs nachjustiert. Dies wird in iterativer Weise wiederholt bis eine ausreichend genaue Übereinstimmung erreicht wird.

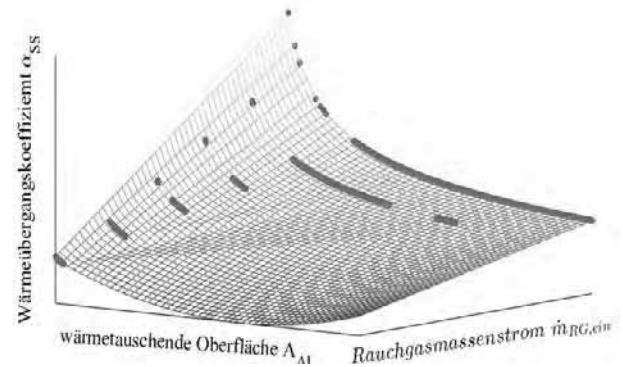


Abbildung 5: mittlerer Wärmeübergangskoeffizient

5 Validierung Simulationsmodell

Um die Anwendbarkeit des resultierenden Simulationsmodells überprüfen zu können, wird ein Vergleich zwischen den Ergebnissen des resultierenden Simulationsmodells und den Messungen aus dem realen Betrieb durchgeführt. Dazu werden die gemessenen Beschickungs- und Entnahmedaten verwendet. Der Vergleich erfolgt anhand der wichtigsten Prozessparameter. Diese sind:

- Schachtmasse (Parameter $m_{Al,SS}$)
- Schmelzleistung (Parameter $\int \dot{m}_{Al,aus,SS}$)

- Badmasse (Parameter $m_{Al,OW}$)
- Abgastemperatur Schacht ($T_{RG,aus,SS}$)
- Gasverbrauch (V_{EG})

Für die Validierung wird eine Messreihe mit ca. 5 Stunden verwendet. Die Auswertung des Vergleichs zwischen Simulationswerten und Messergebnissen erfolgt qualitativ und quantitativ.

5.1 Qualitativer Vergleich

In Abbildung 6 sind die Verläufe für die Aluminiummasse im Schmelzschacht des Ofens für die Simulations- und Messwerte dargestellt. Ein Anstieg der Masse ist immer mit einem Beschickungsvorgang verbunden. Die Abnahme entspricht der Masse an Aluminium, welche durch abgeschmolzen ist und aus dem Schacht in den Warmhaltebereich fließt. Es ist zu erkennen, dass mit der Simulation die reale Situation mit Ausnahme einiger Teilbereiche gut abgebildet werden kann.

Ein Teil der Abweichung resultiert aus der unpräzisen Erfassung der Messdaten und der dazugehörigen Rahmenbedingungen. Dies gilt insbesondere für die spezifische Oberfläche und mittlere Temperatur des chargierten Materials. Zusätzlich können externe Effekte auftreten, wie die manuelle Verringerung der Brennerleistung durch den Schmelzleiter um Überkapazitäten zu vermeiden. Dieser Fall tritt nach ca. 3.8 Stunden auf und kann aufgrund nicht vorhandener Erfassung nicht berücksichtigt werden.

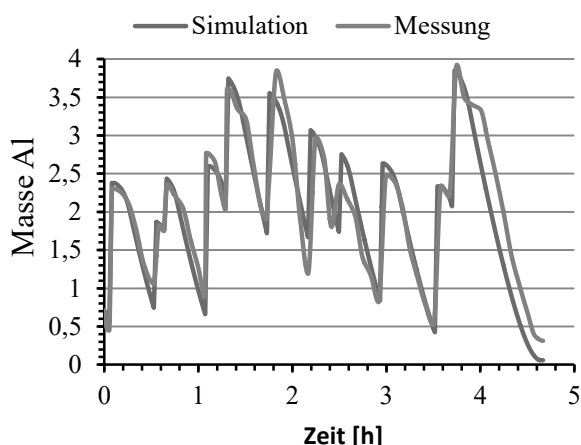


Abbildung 6. Verlauf Aluminium Schmelzschacht

Neben der Aluminiummasse im Schmelzschacht wird die Rauchgastemperatur am Austritt des Schmelzschachtes für die qualitative Bewertung herangezogen. Diese Größe ist besonders gut für den Vergleich

geeignet, da mit dieser die Abbildung des Wärmeübergangs innerhalb des Ofens bewertet werden kann. Wird hier eine gute Übereinstimmung erreicht bedeutet dies, dass die Wärmeübertragungscharakteristik gut erfasst wird.

Der Vergleich in Abbildung 7 zeigt, dass mit der Simulation der Temperaturverlauf des Rauchgases gut abgebildet werden kann. Die nahezu senkrechten, großen Temperaturabnahmen resultieren immer aus einem Chargiervorgang und der daran gekoppelten Erhöhung des Wärmeübergangs aufgrund einer steigenden Temperaturdifferenz und wärmetauschenden Oberfläche zwischen Aluminium und Rauchgas. Die Temperaturabnahmen von ca. 1100 °C auf 1050 °C ergeben sich aus der internen Brennerregelung. Mit dem Erreichen der Grenztemperatur wird die Brennerregelung aktiv und verringert die Leistung der Schmelzbrenner. Dies führt zu einem geringeren Rauchgasmassenstrom und einer erhöhten Abnahme der Rauchgastemperatur innerhalb des Ofens. In der Simulation ist die Temperaturabnahme aufgrund der angenäherten internen Brennerregelung etwas höher als im realen Ofen. Es ist trotzdem eine ausreichend gute Übereinstimmung in diesem Punkt vorhanden.

Die größten Unterschiede treten zu Beginn und nach ca. 3.8 Stunden auf. Die Abweichungen zu Beginn sind auf die unbekannte Betriebssituation des Schmelzofens am Anfang zurückzuführen. Die Abweichung nach 3.8 Stunden resultiert aus der Reduktion der Brennerleistung durch den Schmelzleiter, welche in der Simulation nicht berücksichtigt werden kann.

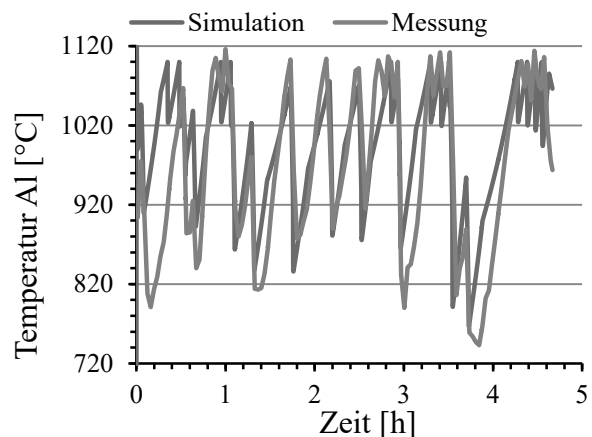


Abbildung 7. Temperatur Rauchgas am Schachaustritt

5.2 Quantitativer Vergleich

Für einen quantitativen Vergleich werden durchschnittlichen Abweichungen zwischen den interpolierten Messwerten und den Simulationsergebnisse bestimmt. Die Ergebnisse sind als Absolut- und Relativwerte in Tabelle 2 eingetragen. Mit 10 % ist die

relative Abweichung für die Aluminiummasse im Schmelzschacht am Größten. Die Abweichungen der verbleibenden Größen sind deutlich geringer, insbesondere der Gasverbrauch wird sehr genau wiedergegeben. Aufgrund der positiven Ergebnisse der qualitativen und quantitativen Betrachtung der Validierung kann geschlossen werden, dass die reale Situation mit dem Modell, trotz der Annahmen und Vereinfachungen, ausreichend genau erfasst wird.

Tabelle 2: Abw. Simulation und Messwerte

Parameter	Abweichung Mittelwerte	
	Abs.	Rel.
Masse Aluminium Ofenwanne	220 kg	2,6 %
Abgastemperatur Schmelzschacht	54 °C	6 %
Gasverbrauch	9 m³	0,5 %
Masse Aluminium Schmelzschacht	180 kg	10 %
Masse geschmolzen	250 kg	1,5 %

5.3 Einschränkungen

Das Simulationsmodell wurde anhand weniger Messreihen validiert. In diesen werden verschiedene Betriebssituationen nicht erfasst. So tritt dabei kein permanent sehr voller bzw. leerer Schmelzschacht auf. Diese Fälle werden in der folgenden Untersuchung auch betrachtet, so dass hier stärkere Abweichungen zur Realität möglich sind.

6 Energieeffizienzuntersuchungen

Mit dem validierten Simulationsmodell wird eine Studie mit unterschiedlichen Beschickungsgrößen und daran gekoppelten Schachtfüllständen ausgeführt. Anhand dieser werden die Auswirkungen der Beschickung auf den spezifischen Energieverbrauch und die Schmelzleistung des Schmelzofens untersucht.

In der Studie erfolgt eine Beschickung alle 1000 Sekunden. Die Masse pro Beschickung wird pro Simulationsdurchlauf um 100 kg vom Ausgangswert 600 kg auf den Maximalwert 1400 kg erhöht. Die Konsequenzen auf die Leistungsdaten des Ofens

werden anhand des Mittelwertes für die Aluminiummasse im Schmelzschacht, der stündlichen Schmelzleistung und des spezifischen Energieverbrauchs betrachtet. Die Ergebnisse sind in Abbildung 8 dargestellt. Mit steigender Masse pro Beschickung steigt die Aluminiummasse nicht linear an. Die Nichtlinearität ergibt sich aus der mit der Masse steigenden Schmelzleistung. Diese steigt mit der zunehmenden Aluminiummasse und Beschickungsgröße von 2800 kg/h auf 4800 kg/h an. Dieser Anstieg der Schmelzleistung ergibt sich aus einer verbesserten internen Vorwärmung des Aluminiums und einer erhöhten Wärmeübertragung. Die Schmelzleistung ist entscheidend für die Masse an geschmolzenem Aluminium und die Speicherstände der Schmelzöfen. Somit sind die Schmelzleistung und auch die daran gekoppelte Betriebsweise der Schmelzöfen entscheidend für die Versorgungssicherheit der Druckgussanlagen.

Der spezifische Energieverbrauch gibt die benötigte Energiemenge für das Schmelzen und Warmhalten einer bestimmten Aluminiummasse an. Dieser nimmt mit steigender Aluminiummasse und Schmelzleistung von 890 kWh/t auf 640 kWh/t ab. Dies entspricht einer Reduktion des Gasverbrauchs um 28 % und zeigt, dass mit höherem Schachtfüllstand ein energieeffizienterer Betrieb des Ofens erreicht wird.

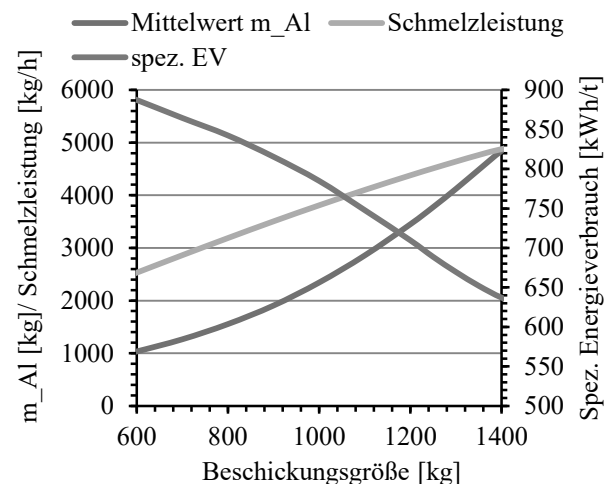


Abbildung 8: Ergebnisse Beschickungsstudie

7 Zusammenfassung und Ausblick

Im vorliegenden Beitrag sind die Modellbildung, Weiterentwicklung und Validierung eines dynamischen Prozesssimulationsmodells eines Schachtschmelzofens beschrieben. Als Grundlage der Modellbildung wird die Funktionsweise des Ofens beschrieben und die vorhandenen Messgrößen aufgelistet. Aufgrund der wenigen globalen Messgrößen, der hohen geforderten Flexibilität und der erforderlichen

niedrigen Rechenzeit wird der Ansatz der dynamischen Prozesssimulation verwendet. Dieser basiert auf einer Kombination des Wärmetauscheransatzes und der Methode der Blockkapazität.

Um die Nachteile dieses Ansatzes zu verringern wird ein mehrstufiges Verfahren zur Verbesserung des Simulationsmodells beschrieben. Dieses basiert auf der Einführung ergänzender Zusammenhänge zwischen verschiedenen Prozessgrößen. Grundlage der Zusammenhänge ist die Analyse vorhandener Messdaten, sowie der Ergebnisse von Strömungssimulationen. Die Qualität des Modells wird anhand eines abschließenden Vergleichs zwischen Mess- und Simulationsergebnissen qualitativ und quantitativ untersucht und bestätigt. Die mit dem validierten Modell durchgeführte Simulationstudie zur Auswirkung der Schachtfüllung auf die Energieeffizienz zeigt, dass die Schachtfüllung sehr großen Einfluss auf die Schmelzleistung und den spezifischen Energieverbrauch besitzt. Durch eine Erhöhung des Schachtfüllstandes kann die Schmelzleistung um bis zu 50 % erhöht und der spezifische Energieverbrauch um bis zu 28 % reduziert werden.

Für die Validierung des Modells und die Bestätigung der Simulationsergebnisse sollen weitere Messungen im Referenzbetrieb durchgeführt werden. Ziel ist es mit den Messungen auch die Grenzfälle, wie eine sehr hohe bzw. niedrige Schachtfüllung zu erfassen. Im Bereich des Modells sollen weitere Schwachstellen beseitigt oder verringert werden. Zum Beispiel soll ein Energiemodell für die Ofenwand eingeführt werden. Des Weiteren sollen die Wärmeverluste während einer Beschickung zunächst mit einer Strömungssimulation untersucht und anschließend in die Prozesssimulation implementiert werden.

8 Quellen

- [1] C. Herrmann, H. Pries und G. Hartmann. *Energie- und ressourceneffiziente Produktion von Aluminiumdruckguss*. Springer Vieweg, Berlin, 2013.
- [2] J. Schmidt und W. Schlüter. Ein dynamisches Prozesssimulationsmodell für die energetische Betrachtung von Aluminium – Schmelzöfen in einer betriebsumfassenden Materialflusssimulation. Tagungsband Konferenz ASIM/GI-Fachgruppen: Simulation thermischer Systeme, in Lippstadt, S. 29-36, 2016.
- [3] H. E. Scherf. *Modellbildung und Simulation dynamischer Systeme: Eine Sammlung von Simulink-Beispielen*. Oldenbourg, München, 2004.
- [4] W. Polifke und J. Kopitz. *Wärmeübertragung: Grundlagen, analytische und numerische Methoden*. Pearson. München. 2009

Combining Capacity Planning with Simulation

Paul Barthel, Tobias Eckert, Ralf Sprenger

GLOBALFOUNDRIES Dresden Module One LLC & Co. KG

Ralf.Sprenger@globalfoundries.com

This paper discusses a methodology to overcome future capacity issues in semiconductor manufacturing. The required capacity is compared with the available tools and critical issues are highlighted. This enables for a manual evaluation of upcoming problems that can be addressed. A plan for improving some of these issues is made and fed into a simulation model to assess its performance. This approach prevents adding cycle time by handling upcoming problems before they become imminent. Strategical and tactical decision making processes using this approach within GLOBALFOUNDRIES are discussed.

1 Introduction

Short-term changes in demand become more and more important in semiconductor manufacturing. Fabs with only a few products and long demand plans turned into foundries that produce hundreds of different products for dozens of customers and have to react fast to demand changes.

While a decade ago, fabs were built according to a stable demand plan and machine parks were purchased accordingly, highly dynamic adaption to the demand is required now. This begins at process engineering level, where the processes have to use existing machines and their configuration. At the same time to prevent yield loss in wafer production, the number of available tools per process step is reduced by very specific constraints. This makes simple calculations, which assume that all machines of the same type can be used for the same process step as it is often done for long-term demand planning unusable.

Expected future changes to tool dedications have to be kept in mind to evaluate if the required fab capacity will be available at the right time. This means that there is no static fab model that can be used for simulation. Simplifications have to be made that might hide critical problems. Many future changes, even for the next few weeks, have to be considered. This often leads to the situation where it is unknown which change is really required and at which time.

Many different systems are used for defining process related limitations. Many of them reduce the capacity of the fab. In addition to these runpath issues, current availability of certain tools has to be considered.

Typically, capacity planning only focuses on technology variants and makes assumptions about usable

machines, e.g. by assuming all machines of the same type have the same dedications, whereas reality is often different. Simulation can help but one of its weaknesses is that if the current setup is not able to process the material, it will get stuck at the first bottleneck in the flow. Certainly, this is an indicator for an upcoming problem. It could be addressed, solved in reality and the solution can be put back into the simulation model to find and proceed with the next bottleneck. However, it is very time consuming to solve each problem step-by-step.

Setup details change very often. Any kind of runpath limitation can be set within seconds in the system and also the upcoming demand can change once or twice per week. One of the major requirements for an analysis approach is that it can provide results within a few hours.

In this paper, we will describe a methodology how an evaluation of all potentially upcoming problems can be done at the same time. We will describe how it is used to identify all problems and assess if they will become critical. Then we remodel them - including future solutions - back into a simulation model that allows a cycle time estimation of the new plan.

2 Methodology

In this section, we will discuss the constraints and, afterwards, describe the calculation approach.

2.1 Constraints

Many constraints have to be considered to assess the capacity available to fulfill a demand plan:

- a tool needs to be able to process a recipe,
- yield data indicates that it should be used for volume production,

- it is not inhibited. There are several reasons why a machine can be inhibited such as process related limitations, momentary bad signals for similar products, setup change considerations based on upcoming material or labor-related issues to reduce the workload on the shop floor and
- reticle limitations for photo lithography operations.

In the first step, we assume that all of these restrictions persist as-is because we cannot be sure if they are set only temporarily and can be released instantly, if we have to address them as necessary to being solved or if they need to be considered as mid or long-term restrictions. As a result, for each process step we have a given set of usable tools that are dedicated and not restricted according to the described limitations.

For capacity calculations, the following additional data is required:

- process flows,
- current availability of machines,
- throughput rates of machines and recipes,
- current positions of material in the line and upcoming wafer starts plan and
- a target speed at which material should be moved through the process flow. This may be provided in terms of plan values for different products or as customer orders with known due dates.

2.2 Calculation Approach

Step 1: Determine all lots i in the line and their current position p_m .

Step 2: Create and add all lots that are scheduled to be started within the planning period h to the beginning of their flow (p_0).

Step 3: Move each lot i along its flow $p_0 \dots p_n$ with a given turn rate tr_i over a given planning period h (e.g. one week):

$$tr_i = \frac{\text{remaining steps to end of line}_i}{\text{remaining time until desired fab out}_i}$$

We assign the lot to each operation $p_m \dots p_{m+tr_i \cdot h}$ that it passes within the planning period.

Step 4: For each operation, distribute the lots to the available tools using the following balanced allocation heuristic:

- Determine the tools and sort them according to their flexibility.
- Take the least flexible tool and distribute the lots to it until it is loaded 50%.
- Using the next tool repeat step *a* until there are no remaining tools.
- Distribute the remaining lots evenly. Load beyond 100% is allowed.

Note that this can also be done using a MIP solver but the runtime is much higher so that we use this heuristic for most scenarios.

Step 5: Go back to Step 1 but use the forecasted position of each lot as its new starting position in the flow and evaluate for the next planning period.

There are different possibilities to determine the number of operations that should be passed within a given horizon as done in Step 3. [1] and [2] describe an approach that determines the cycle time per operation by multiplying the raw processing time with a so-called x-factor. Alternatively, using estimations based on historic, measured cycle times is possible. For our use cases a strictly uniform approach as described in Step 3 leads to the best results. This mainly has the following reasons:

- Historic cycle time approaches turned out not to be very stable. The forecasts change too much over time due to changed real cycle times. For example, if a bottleneck existed in the past in one metal layer causing high cycle time but there is no indication that this will appear again in the future.
- Process time related approaches assume that high process time also leads to high cycle time. Sometimes this is correct as shown in [1] but for some operations with long process times like at furnaces this leads to unreasonably long cycle times making up for large amounts of the overall cycle time. Consequently, layers with many furnace operations are much slower than in reality.

2.3 Load situation

One of the results of the described calculation approach gives the utilization for each tool and period.

Defining groups of tools according to unique sets of process dedications is essential for further investigation. Figure 1 shows an example output. ToolGroup and SubGroup levels show loading average:

ToolGroup A: Major capacity gap. Overall capacity in the machine group is insufficient.

ToolGroup J: Temporary overload situation that might increase cycle time slightly.

ToolGroup M: Situation looks good on average ToolGroup level but insufficient releases cause imbalance and individual tool overloads.

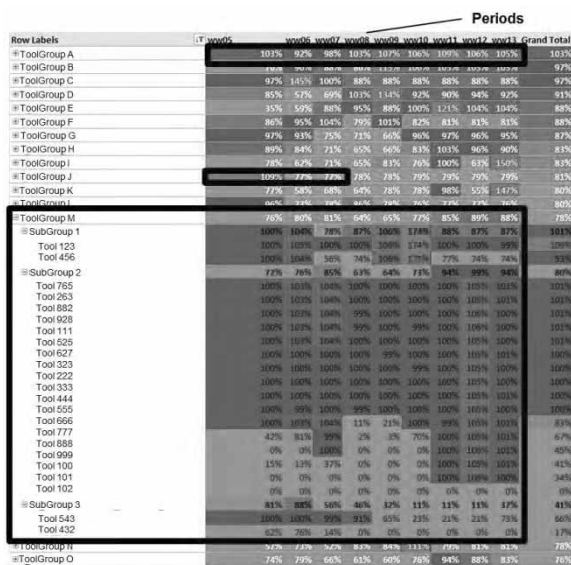


Figure 1. Utilization overview

We follow a two-step approach:

1. analyze situation on machine group level, then
2. analyze release situation within the machine groups.

2.4 Required moves level

As tool overload is permitted and lots are moved with target speed the resulting tool loading are *required moves*. Figure 2 shows an example grouped per metal layer. However, aggregating per customer, recipe or operation description is also possible.

2.5 Outs profile

The outs profile is a direct result of the calculation. Normally, a fab should have a smooth profile. High peaks are not supported by any production line. De-

sired fab out dates for the lots should be reviewed if the profile is not smooth.

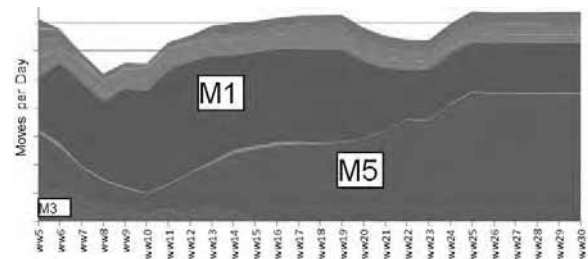


Figure 2. Required moves per layer for one tool group

2.6 Analysis of releases

The way release related limitations are visualized is shown in Figure 3. Releases with many moves are at the top whereas in lower rows the less important ones are shown. This makes clear which problems lead to this overload situation and need to be solved.

ToolGroup	Flow	Product	Operation	Moves	Tools released	Inhibited Tools with Reason
LITHO1	1234	P1	LIT-M4	3333	LITA1;LITA2	LITA3 (Maintenance)
POLISH3	4321	P1	POL-M0	321	POL50	POL61 (bad yield); POL62 (down)
FVX2	3333	P2	FVX-123	10	FVX456;FVX567	

Figure 3. Overview of releases

In daily business, we use the QlikView reporting system instead of Excel. Analysis for the whole fab can be done in around one hour. Note that all data is scrambled and does not show real industry data.

3 Work mode

At GLOBALFOUNDRIES, capacity planners are analyzing this data every week. They start with the utilization profile and analyze if there are problems with the utilization on tool group level. If the analysis shows overload situations but actual performance is assumed to improve in the future, improvement schedules are fed to a simulation model by adding them to a list as shown in Table 1.

After that, the planners analyze at the tool level for single overloaded tools. They analyze tool releases and try to figure out which of them are persistent, which constraints can be solved and which ones cannot. Some of the findings lead to action plans with other departments that commit a date by when the improvement is made. Additional releases are added to Table 1.

For analyzing and comparing performance with reality, additional data like cycle time and number of

average passes for a machine group based on current material mix is provided to the planner (Figure 4). In addition, the required maximum and average moves level is compared to reality.

Entity	Change	Owner
M2-LIT for customer xy	add release of tool LIT123 to this operation on 12/07/16	LIT
Tool POL321	Increase assumed capacity by 8% immediately.	CMP
Reticle 321AB	Can be used for all operations that use all tools of group Immersion-M starting Sep 2016.	LIT

Table 1. Change list generated by the planner

3.1 Feeding data back to the simulation

Table 1 shows the outcome of the manual analysis of the planner. This data is fed to a simulation model that implements the changes at the given time. Cycle time and wafer outs predictions are used to verify the capacity plan and estimate a realistic cycle time based on the improvements.

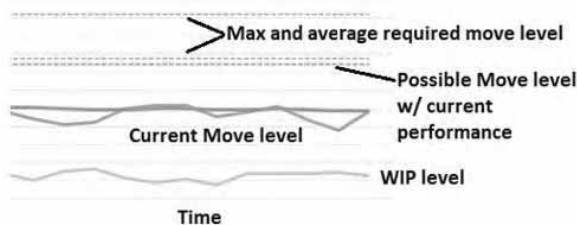


Figure 4. Comparison with current moves level

3.2 Decision making

Outcomes from the described calculation are used by GLOBALFOUNDRIES for decision making at many additional points in the line. Some examples for usage in tactical decision making are:

- Line control aligns moves targets, for example, per metal layer based on this data and analyzes the runpath to handle potential weak points.
- Modules prepare their tools for the upcoming load and check all limitations that are set by operational requirements.
- Integration uses a list of inhibits that are critical in the future and have to be solved.

For strategic decision making:

- Supply chain planners verify if the wafer starts plan can be fulfilled, vary product starts if required and check if additional demand can be supported.
- Line control ranks material at the highly loaded points in the line to overcome capacity issues by processing important material first.
- Consumption of consumables is assessed by the modules.

4 Outlook

Approaches like described in [3] cover the problem of increased complexity in semiconductor manufacturing by focusing on the most important topics. However, any obstacle can cause high cycle time in the line and has to be avoided.

We have been using the described approach for several years now and see great benefits. The complexity and urgency of runpath clearance cannot be handled anymore by a simulation model alone. Switching to a work mode where all obstacles are automatically analyzed and addressed to the right owners afterwards is essential.

At this time, we use a static cycle time for moving the material along the flow. Using cycle time estimations based on characteristic curves might improve the approach in the future.

5 References

- [1] K. Ozawa, H. Wada and T. Yamaguchi, *Optimum tool planning using the X-factor theory*. In IEEE International Symposium on Semiconductor Manufacturing Conference Proceedings, Santa Clara, CA, USA, pp. 49-52, 1999.
- [2] M. Kishimoto, K. Ozawa, K. Watanabe and D. Martin, *Optimized operations by extended X-factor theory including unit hours concept*. In IEEE Transactions on Semiconductor Manufacturing 14(3), pp. 187-195, 2001.
- [3] R. Sprenger and O. Rose. *On the Simplification of Semiconductor Wafer Factory Simulation Models*. In Conceptual Modeling for Discrete-Event Simulation, Taylor&Francis, Boca Raton, FL, USA, pp. 451-470, 2010.

Simulative Untersuchung energie- und ressourceneffizienter Betriebsstrategien in der E³-Forschungsfabrik

Johannes Stoldt¹, Marc Münnich^{1,2}, Enrico Franz¹, Andreas Schlegel¹, Matthias Putz^{1,2}

¹Fraunhofer-Institut für Werkzeugmaschinen und Umformtechnik IWU, Chemnitz

²Technische Universität Chemnitz, Fakultät für Maschinenbau, Chemnitz

johannes.stoldt@iwu.fraunhofer.de

Gesellschaftlicher, politischer und wirtschaftlicher Druck erfordern es, dass produzierende Unternehmen auf die Energiewende reagieren. Hierzu werden große Anstrengungen zur Entwicklung effizienter Produkte, Prozesse und Betriebsstrategien unternommen. In der E³-Forschungsfabrik des Fraunhofer IWU wird an Lösungen für dieses Spannungsfeld geforscht. Um insbesondere Ansätze des energieeffizienten Betriebsmanagements untersuchen zu können, wurde ein Simulationsmodell auf Basis des realen Systems entwickelt. Ausgehend von der ultrakurzen Prozesskette einer Antriebshohlwelle wurde ein Produktionsprogramm entwickelt. Für dessen Fertigung notwendige Maschinen wurden vermessen und in Siemens Tecnomatix Plant Simulation modelliert. Dieser Beitrag berichtet über die zugrundeliegende Systemanalyse sowie die Implementierung und gibt einen Ausblick über die geplanten Untersuchungen.

1 Einleitung

Die stetige Verschärfung von Richtlinien zur Energieeffizienz und aktuelle Herausforderungen wie die steigenden Preise für Energie, strengere Regularien zur Schonung der Umwelt und die Energiewende zwingen Unternehmen zu Anstrengungen im Bereich der effizienteren Energienutzung. Gerade in Deutschland ist der Anteil energieintensiver Industrie mit den starken Sektoren Maschinenbau und Automobilproduktion sehr groß. Lösungen zur Bewältigung dieser aktuellen Herausforderungen werden dringend benötigt. Nicht nur der Umgang mit verfügbaren Ressourcen, sondern auch die Nutzung von erneuerbaren Energien in der Produktionsumgebung spielt dabei eine zentrale Rolle. Eine veränderte Denkweise in Hinblick auf Energiegewinnung und Ressourcenschonung im Fabrikbetrieb sind deshalb unabdingbar, um diesen Anforderungen gerecht zu werden [1].

Am Fraunhofer-Institut für Werkzeugmaschinen und Umformtechnik IWU in Chemnitz entstand mit der E³-Forschungsfabrik ein Ort, in dem an der autarken und emissionsneutralen Produktion der Zukunft geforscht wird. Schwerpunkt der Arbeiten in Chemnitz sind die Konzeptionierung ultrakurzer Prozessketten in der Fertigung von Antriebsstrangkomponenten, der Einsatz ressourceneffizienter und flexibler Technologien im Karosseriebau sowie die Integration innovativer Lösungen im Daten- und Energiemanagement [2]. Letzteres umfasst neben der ressourceneffizienten Fertigungssteuerung auch Ansätze zur dezentralen

Einbindung von erneuerbaren Energiequellen sowie Energiespeichern in die Produktion.

Eine zentrale Schwierigkeit in diesem Zusammenhang stellt die Erprobung neuer Strategien und Verfahren zum Energie- und Produktionsmanagement, kurz Betriebsmanagement, dar. Mit der energetisch optimierten Fertigung einer Antriebshohlwelle besteht in der E³-Forschungsfabrik bereits eine innovative Referenzprozesskette. An dieser können einerseits weitere Prozessanpassungen erprobt und andererseits verschiedene Prozesskennwerte ermittelt werden. Zugleich besteht im realen System jedoch kein durchgängiger Produktionsablauf, welcher die unmittelbare Untersuchung neuer Betriebsstrategien zuließe. Durch den Einsatz von Simulation kann dieser jedoch nachempfunden werden, so dass neuste Entwicklungen auf Grundlage real gemessener Daten untersucht und validiert werden können.

Dieser Beitrag stellt ein Simulationsmodell der E³-Forschungsfabrik vor und gibt einen Ausblick auf damit geplante Untersuchungen. Hierzu wird zunächst ein Überblick zu simulationsgestützten Untersuchungen von Betriebsmanagementstrategien aus der Literatur präsentiert. Eine Systemanalyse zur E³-Forschungsfabrik erläutert Struktur, Produktionsprogramm und Dimensionierung sowie Energieversorgungssystem des realen Systems. Im Anschluss wird die Implementierung in Siemens Plant Simulation diskutiert, gefolgt von der Vorstellung zu untersuchender Betriebsstrategien bzw. Forschungsansätze.

2 Stand der Technik

Die Erprobung und unternehmensweite Implementierung von Kanban wurde einst durch praktische Versuche getrieben, wobei immer weitere Bereiche auf diese Weise gesteuert wurden [3]. Mit der Verfügbarkeit von Simulationssoftware können derartig tiefgreifende Veränderungen im Betriebsmanagement zunächst in einer „sicheren“ Umgebung untersucht werden. Entsprechend findet sich in der Literatur eine Vielzahl von Ansätzen und Fallbeispielen, in denen Energieeffizienzmaßnahmen mittels Simulation gestaltet oder validiert wurden. Literaturanalysen zu dazu eingesetzten Werkzeugen bzw. Ansätzen finden sich beispielsweise in [4] und [5].

Eingriffe in die Produktionsplanung und -steuerung (PPS) sind ein häufig gewähltes Mittel um den Energieeinsatz in der Fertigung zu senken. Dabei kann grob in Steuerungsstrategien [6], welche die Abarbeitung von Aufträgen beeinflussen, Losgrößenplanung [7], Kapazitätsplanung [5] und Sequenzierung bzw. Reihenfolgeplanung [8] unterschieden werden.

Auch die Systemplanung mit unterschiedlichem Fokus stellt einen zentralen Fokus bisheriger Arbeiten dar. Die Bestimmung des ökologischen Fußabdrucks [9] (häufig in Anlehnung bzw. auf Basis von Lifecycle Assessment – LCA) dient vordergründig der Analyse von Prozessen. Die Veränderung von Systemen wurde bislang hinsichtlich Prozess-/Systemgestaltung [10], Infrastrukturplanung [11] und Betriebsoptimierung [12] (Anpassung von Maschinensteuerungen zur Effizienzsteigerung, z. B. Vermeidung von Lastspitzen durch veränderte Kinematik) untersucht.

Das Modell der E³-Forschungsfabrik soll zukünftig gleichermaßen der Untersuchung von PPS- und Systemplanungsansätzen dienen. Durch die explizite Beachtung erneuerbarer Energiequellen und der Effekte ihrer Einbindung werden die Arbeiten über den bisherigen Stand der Technik hinausgehen.

3 Systemanalyse

Der vorliegende Beitrag stellt einen Teil einer Simulationsstudie vor, welche in Anlehnung an die VDI 3633 Blatt 1 [13] konzipiert wurde. Entsprechend werden im Folgenden die Aufbaustruktur und Systemgrenzen sowie das Produktionsprogramm und die Dimensionierung thematisiert. Abschließend wird das Energiesystem der E³-Forschungsfabrik herausgehoben, welches Ausgangspunkt für verschiedene geplante Untersuchungen bildet.

3.1 Aufbaustruktur und Systemgrenzen

Voraussetzung für die simulative Abbildung der E³-Forschungsfabrik war die zweckmäßige Analyse selbiger sowie die Abgrenzung ihrer einzelnen Bestandteile. Von übergeordnetem Interesse war es, die räumliche Struktur der Fabrik zu erfassen und zudem auch eine Vereinfachung des komplexen Produktionsraumes vorzunehmen. Dazu erfolgte einleitend eine Aufteilung des Fabrikkomplexes in mehrere Hierarchieebenen [1]. Diese bilden ein Gefüge sich gegenseitig beeinflussender Subsysteme und enthalten zudem alle für die folgende simulative Betrachtung relevanten Elemente. Demgemäß wurden insgesamt vier Ebenen definiert und der Untersuchungsraum für alle weiteren Betrachtungen abgegrenzt.

Der Fabrikkomplex als Gesamtheit der einzelnen Produktionsbereiche und Standort aller benötigten Anlagen und Einrichtungen stellte die oberste Betrachtungsebene und zugleich auch die Systemgrenze zur angrenzenden Umwelt dar. Eine Interaktion mit dieser Umwelt besteht beispielsweise für die Anlieferung von Rohmaterial oder aber auch für die Auslieferung von Endprodukten. Ausgehend von dieser Gegenseitigkeit der Einflussnahme wird auch von einem offenen System gesprochen [14].

Aufbauend auf dieser Abgrenzung des Betrachtungsgegenstandes wurde der Fabrikkomplex in seine Gebäudekomplexe differenziert. Dies ist zum einen die E³-Forschungsfabrik selbst und zum anderen das daran angrenzende Versuchsfeld für umformtechnische Materialbearbeitungen. Erstere bildet dabei den zentralen Betrachtungsgegenstand der Systemanalyse und wurde im nächsten Schritt in die eingangs bereits erwähnten Kompetenzbereiche flexibler Karosseriebau und ultrakurze Prozesskette für Antriebsstrangkomponenten gegliedert. Aufgrund der Fokussierung auf letzteren Fertigungsabschnitt erfolgte die Analyse des Bereichs Karosseriebaus lediglich im Sinne eines Black-Box-Modells. Dieses ist gekennzeichnet durch eindimensionalen Input, Produktionsfunktion sowie ebenso vereinfachten Output und wurde weiterführend nicht genauer in die Systemanalyse einbezogen.

Demgegenüber stellten die Betriebsmittel der Antriebskomponentenfertigung die niedrigste Abstraktionsebene der Systemanalyse dar. Ausgehend von der voranstehenden Betrachtung der einzelnen Systembereiche bietet Abbildung 1 eine vereinfachte Darstellung der E³-Forschungsfabrik. Gezeigt sind die darin befindlichen Systemelemente (grau schraffiert mit abgekürzten Bezeichnungen), deren räumlichen An-

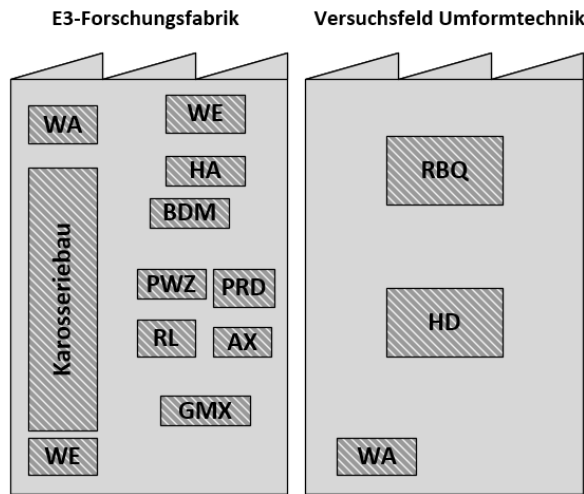


Abbildung 1 - Vereinfachte Darstellung des Systems.

ordnung und der Wareneingang (WE) sowie Waren-
ausgang (WA) als Systemgrenzen zur Umwelt.

Das Gros der Betriebsmittel ist auf dem Areal der E³-Forschungsfabrik angeordnet. Im Sinne einer hohen Vielfalt an durchführbaren Fertigungsverfahren wurden überdies weitere Betriebsmittel des Versuchsfeldes für Umformtechnik einbezogen. Dies geschah vorausschauend auf die Definition des Produktionsprogramms.

Der folgende Abschnitt beschäftigt sich mit der Identifikation von Produktvarianten und der Produktionsprogrammplanung. Weiterhin wird die Auslegung der zu simulierenden E³-Forschungsfabrik im Hinblick auf die Bestimmung von Betriebsmittellanzahlen und deren zeitlichen Einsatzbedingungen thematisiert.

3.2 Produktionsprogramm und Dimensionierung

Bereits einleitend wurde auf die Schwierigkeiten in der Forschung im Betriebsmanagement mit Hinblick auf die Erprobung neuer Betriebsstrategien aufgrund eines fehlenden durchgängigen Produktionsablaufes hingewiesen. Zugleich besteht mit der Prozesskette einer energetisch optimierten Antriebs-hohlwelle ein Demonstrator, der Eingangsdaten für die Validierung fertigungsstrategischer Anpassungen liefern kann. Um die Potenziale der bisher erforschten Prozessanpassungen auch unter praxisnahen Bedingungen untersuchen zu können, bedarf es der Generierung und Auswertung von Produktionsdaten. Simulationsexperimente bieten eine probate Möglichkeit zur Validierung neu konzipierter Steuerungsstrategien.

Zu diesem Zwecke wurde ausgehend von den Kernkompetenzen des Fertigungsbereiches ein virtuelles Produktionsprogramm für den untersuchten Fertigungsabschnitt entwickelt gefolgt von einer statischen Neudimensionierung des Produktionsbereiches. Damit wurde einerseits erreicht, dass ein industrieorientiertes Spektrum an Produkten abgebildet wird und andererseits wurde auch die Bewältigung der geplanten Systemlast durch die Anpassung der Betriebsmittelzahlen und deren Arbeitszeitfonds sichergestellt.

Im Rahmen der Produktfindung wurden dafür zunächst die Schlüsseltechnologien der Fertigung identifiziert. Diese liegen besonders im Bereich der Herstellung von Antriebsstrangkomponenten mit der Fokussierung auf den Einsatz überwiegend umformbasierter Fertigungsverfahren wie Rundkneten, Bohrungsdrücken oder Verzahnungswalzen. Der Vorteil dieser Verfahren liegt dabei besonders in der Einsparung an Bauteilgewicht und Material gegenüber abtragenden Verfahren wie Abwälzfräsen oder Bohren.

Ausgehend von der bereits optimierten Antriebs-hohlwelle wurde eine Diversifizierung in verschiedene Varianten rotationssymmetrischer Bauteile vorgenommen. Insgesamt konnten dadurch neun Produktvarianten erarbeitet werden. Für diese wurden entsprechende Arbeitsfolgen und Fertigungszeiten ermittelt. Das Spektrum erstreckt sich dabei von verschiedenen Hohl- und Vollwellen mit oder ohne Gangrädern bis hin zu Zahnrädern mit geraden oder schrägen Verzahnungen. Abbildung 2 zeigt einen Überblick über die dabei gewählten Fertigungsfolgen durch das Arbeitsablaufschaema (vertikal Arbeitsschritte und dafür notwendige Maschinen, horizontal Produktvarianten).

Im Anschluss wurde das Produktionsspektrums um eine mengenmäßige und zeitliche Komponente erwei-

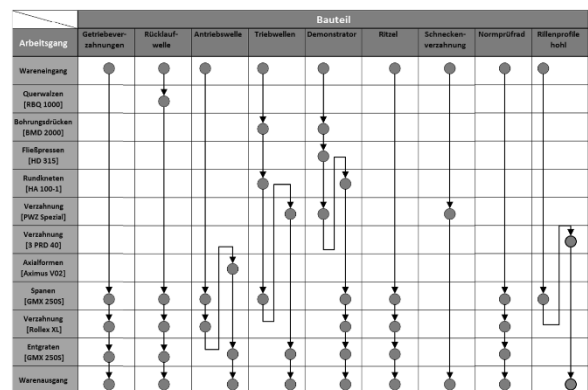


Abbildung 2 - Arbeitsablaufschaema der Produkte.

tert um das Produktionsprogramm zu definieren [15]. Bei der Betrachtung der zu fertigenden Stückzahlen wurden Annahmen getroffen, die eine Vergleichbarkeit mit einem typischen, mittelständigen Lohnfertiger ermöglichen sollen. Die dabei ermittelten Gesamtstückzahlen für eine Planungsperiode von einem Jahr wurden auf gleiche Losgrößen aufgeteilt und somit auch die Auftragsanzahl je Produktvariante und Planungsperiode festgelegt.

Um sicherzustellen, dass das geplante Produktionsprogramm auch vom Produktionssystem innerhalb der Planungsperiode bewältigt werden kann, fand ein Abgleich der vorhandenen mit der benötigten Betriebsmittelkapazität statt. Diesem Abgleich lag die Annahme zugrunde, dass die installierte Betriebsmittelkapazität mindestens so groß sein musste, wie die benötigte Kapazität zur Abarbeitung aller Fertigungsaufträge [16]. Dafür wurde der Bilanzansatz auf Basis der zeitlichen Belastung entlang des Ein-, Zwei- und Dreischichtbetriebes der Produktion herangezogen. Diesem Abgleich entsprechend wurden die Anzahlen der benötigten Betriebsmittel angepasst und somit die Bewältigung des Produktionsprogramms unter Missachtung eventueller Bedarfsschwankungen über den zeitlichen Verlauf hinweg sichergestellt.

Bei der Auswahl der Anzahl von Betriebsmitteln für die simulierte E³-Forschungsfabrik wurde einerseits die Anzahl an Maschinen minimiert und andererseits eine Begrenzung auf maximal zwei verschiedene Schichtsysteme für alle Betriebsmittel vorgenommen. Der Erkenntnisgewinn aus diesem Prozess diente als Eingangsgröße für die Auslegung des zu implementierenden Simulationsmodells. Da für die eigentliche Modellerstellung das Energiesystem der E³-Forschungsfabrik eine zentrale Rolle spielt, wird dies im Folgenden zunächst vorgestellt.

3.3 Energiesystem der E³-Forschungsfabrik

Die E³-Forschungsfabrik "Ressourceneffiziente Produktion" am Fraunhofer IWU in Chemnitz, ist nicht nur Forschungs- und Versuchslabor für innovative Technologien, Prozessketten und Produktionssysteme zur energie- und materialsparenden Herstellung von Powertrain-Komponenten und Karosseriebaugruppen. Als Prototyp und Demonstrator energieeffizienter Ansätze des Industriebaus und der technischen Gebäudeausrüstung sowie innovativer Konzepte zur Energieversorgung und zum Energiedatenmanagement ist die E³-Forschungsfabrik selbst Gegenstand einer intensiven Forschung auf diesen Gebieten.

Ein wesentliches Element des Energiesystems der E³-Forschungsfabrik ist die lokal vorgehaltene Infrastruktur zur Erzeugung aller benötigten Energien und Prozessmedien. Integriert wurden hierzu regenerativer Energiequellen (Photovoltaikanlage mit 58,5 kW_p Spitzenleistung, zusätzlich ist die Errichtung eines Vertikalachsenwindrads geplant) und ein gasbetriebenes Blockheizkraftwerk (238 kW_{el}/263 kW_{th}) sowie eine Absorptionskältemaschine (210 kW_{th}) zur gekoppelten Erzeugung von Wärme, Elektrizität und Prozesskälte. Mit diesen sollen die Produktionsanlagen und weiteren Einrichtungen der E³-Forschungsfabrik stets bedarfsorientiert und nach Kosten- bzw. Emissionszielen optimiert versorgt werden können. Geplant ist die zeitnahe Ergänzung dieser technischen Infrastruktur mit Energiespeichern. Diese sollen vorzugsweise für die Glättung des Strombedarfs im Kurzfristbereich (Sekunden bis Minuten) sowie für die Speicherung der aus regenerativen Quellen gewonnenen Energie im mittelfristigen Bereich ihre Anwendung finden. Darüber hinaus ist die Einbindung thermischer Speicher zur verbesserten Abwärmenutzung bei Einsatz des BHKW, auch in Zeiten reduzierten Wärme- und Kältebedarfs, angestrebt.

Das Energiesystem der E³-Forschungsfabrik basiert neben dem oben dargestellten Energieversorgungskonzept auf einem fortschrittlichen Energiedatenmanagement. Dieses umfasst das weitgefächerte Mess- und Erfassungsnetzwerk mit über 150 Messstellen und mehr als 1500 Datenpunkten sowie weitere Lösungen und Komponenten für ein integriertes Energie- und Produktionsmanagement. Leitstandsfunktionen dienen der permanenten Erfassung, Bewertung und Beeinflussung energieverbrauchsbezogener Größen.

Für die Verbindung zur Maschine selbst dienen Informationsschnittstellen. Über diese können die relevanten Messgrößen des Energieverbrauchs, Betriebsparameter der Anlagen und produktqualitätsbezogene Daten übermittelt werden. Zugleich lassen sich Steuersignale zum gezielten An- und Abschalten oder Regeln von Anlagen und Baugruppen erproben und umsetzen. Mittels der vorhandenen Schnittstellen konnten die simulationsrelevanten Maschinen hinsichtlich ihres Energieverbrauchs weitgehend vermessen werden. Für die Modellerstellung wurden die Daten dabei so aufgearbeitet, dass für jede Maschine zunächst Betriebszustände identifiziert und für diese anschließend ein durchschnittlicher Elektrizitätsverbrauch ermittelt wurden.

4 Implementierung

Für die Abbildung von Energieflüssen in der Simulation wurde die Eigenentwicklung eniBRIC genutzt. Diese wird eingangs kurz erläutert, bevor die eigentliche Modellstruktur und die darin implementierte Ablaufsteuerung vorgestellt werden.

4.1 eniBRIC

Die Klassenbibliothek eniBRIC (siehe auch [17][18]) stellt eine Erweiterung für Siemens Tecnomatix Plant Simulation dar. Mit dieser kann in flexibler Weise der Verbrauch und die Bereitstellung von Energie in der Materialflusssimulation mitbetrachtet werden. Ihr zentraler Bestandteil ist die Objektklasse bzw. das Modul eniBRIC. Dieses ist für jedes Simulationsobjekt zu instanzieren, dessen Energieverbrauch oder -abgabe simuliert werden soll. Demfolgend kann energieversorgende Infrastruktur mit eniBRIC explizit mitbetrachtet werden.

Die entstandene Lösung zeichnet sich dadurch aus, dass sich mit ihr parallel quasi beliebig viele Energieträger betrachten lassen. Hierzu findet keine exakte Abbildung von physikalischen Energiewandlungs- oder auch -übertragungsprozessen Anwendung. Stattdessen werden Betriebszustände mit durchschnittlichen Aufnahme- bzw. Abgabeströmen zur Modellierung von Energieverbrauch bzw. -bereitstellung genutzt. Änderungen des Betriebszustandes werden über Materialflussereignisse (Ein-/Ausreten von Teilen) oder gezielte Steuerungsbefehle realisiert. Letztere lassen sich über eine definierte Schnittstelle an das Modul eniBRIC übermitteln. Hiermit lassen sich auch komplexere Energieverbrauchsprofile als Folge mehrerer Betriebszustände realisieren.

Während der Simulation kommunizieren einzelne Instanzen von eniBRIC direkt miteinander, um die Verfügbarkeit von Energieträgern zu prüfen oder Veränderungen im Verbrauch zu signalisieren. Die zugrundeliegende Objektklasse vereint dazu alle Funktionalitäten von Verbrauchern und Erzeugern, so dass sich einzelne Instanzen lediglich in ihrer Parametrierung (z.B. aufgenommene/abgegebene Energieträger, Betriebszustände, etc.) unterscheiden. Ein Logbuch über Veränderungen in der Höhe der aufgenommenen oder abgegebenen Energieträger wird für instanzspezifische Auswertungen lokal mitgeführt.

Ergänzend beinhaltet die Klassenbibliothek eniBRIC noch ein Konfigurationsmodul und ein Auswertemodul. Ersteres wird verwendet, um globale Einstel-

lungsvorlagen zu verwalten und an Instanzen von eniBRIC zu propagieren. Darüber hinaus enthält es eine Zuordnungsmatrix, in welcher sich jedem Verbraucher die Versorger für die jeweils aufgenommenen Energieträger zuweisen lassen. Die einzelnen eniBRIC-Instanzen kommunizieren Zustandveränderungen ebenfalls an das Auswertemodul. Somit können mit diesem modellweite Auswertungen zum Energieverbrauch realisiert werden.

4.2 Modellstruktur

Die implementierte Modellstruktur in Siemens Tecnomatix Plant Simulation lehnt sich direkt an die Ergebnisse der Systemanalyse an. Dabei wurde ausgehend von den definierten Hierarchieebenen des realen Systems eine Abstraktion dieser auf das Simulationsmodell vorgenommen und mittels Plant Simulations Netzwerkobjekten realisiert. Für die Abbildung aller identifizierten Strukturen wurden die in der Software bereitgestellten Bausteine genutzt und teilweise unter Nutzung geeigneter Steuerungen (Prozeduren) zu Bausteinen höherer Logik zusammengefasst, wie dies beispielsweise auf eniBRIC zutrifft.

Das Hauptnetzwerk (siehe Abbildung 3 links) beinhaltet dabei den informationstechnischen Kern und einen großen Teil der benötigten Logik für die Ablaufsteuerung des Materialflusses. Dies betrifft insbesondere die Auftragseinlastung und -freigabe, aber auch Instanzen des Auswerte- und Konfigurationsmoduls von eniBRIC sowie einer Erzeugerinstanz für Elektrizität.

In der E³-Forschungsfabrik, als nächstniedrigere Struktureinheit (siehe Abbildung 3 rechts), findet der eigentliche Materialfluss statt. Somit kommt dieser eine zum großen Teil veranschaulichende Rolle innerhalb des gesamten Simulationsmodells zu. In diesem Netzwerk wurden die Betriebsmittel instanziiert und zudem die Steuerungslogik zur Bauteilweitergabe zwischen diesen implementiert. Neben dem ausführlich modellierten Fertigungsbereich für Antriebsstrangkomponenten wurde die Karosseriebauzelle lediglich als Einzelstation implementiert, sodass diese nur für die Bewertung des Gesamtenergieverbrauches des Systems zweckdienlich ist.

Die Betriebsmittel bilden die niedrigste Netzwerkebene des Modells. Innerhalb dieser erfolgt eine große Anzahl an Steuerungsvorgängen (etwa zur Realisierung losweiser Bearbeitung), welche maschinenübergreifend identisch sind. Durch die Bildung einer eigenen Objektklasse, welche entsprechend mehrfach

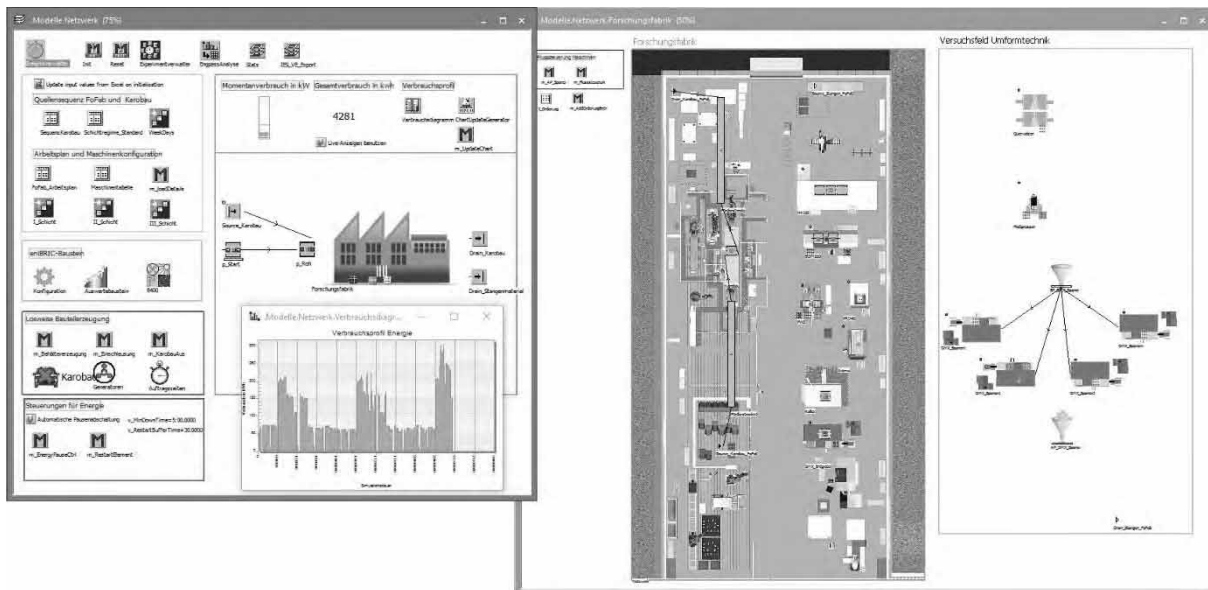


Abbildung 3 - Screenshot des Modells während einer Simulation (Hauptnetzwerk links, E³-Forschungsfabrik rechts).

im Netzwerk E³-Forschungsfabrik instanziiert wurde, konnten alle Betriebsmittel mit der gleichen Struktur und Funktionalität modelliert werden. Einzelne Instanzen unterscheiden sich somit lediglich in ihrer Parametrierung und Bezeichnung. Die Struktur der Betriebsmittel-Objektklasse beinhaltet mehrere Pufferrelamente, welche einer Einzelstation vor- bzw. nachgelagert sind. Hierunter fallen Ein- und Ausgangspuffer sowie Ab- und Aufstapelplatz. Ergänzend zu diesen sind Methoden (Prozeduren) zur Realisierung der Betriebsmittellogik Bestandteil der Klasse. Überdies wurde die eniBRIC-Objektklasse integriert. Dessen Einbindung in die Logik der Betriebsmittel ermöglicht die Auswertung von, aber auch die Beeinflussung des Materialflusses nach energetischen Aspekten. Die Charakteristika des Materialflusses, auch unter Einbezug der energetischen Analyse von diesem, werden nachfolgend genauer dargestellt.

4.3 Ablaufsteuerung

Die Ablaufsteuerung des Modells bestimmt den Materialfluss während der Simulation und somit die Abarbeitung der Produktionsaufträge gemäß implementierter Steuerungsstrategie. Dieser Vorgang ist an den Informations- und Energiefluss gekoppelt und findet innerhalb der definierten Systemgrenzen statt.

Der simulierte Materialfluss der E³-Forschungsfabrik wurde nach dem Push-Prinzip realisiert und es wurde eine losweise Bearbeitung und Weitergabe von Produktionsaufträgen definiert. Die Erzeugung von Fertigungsaufträgen auf Basis des Produktionspro-

gramms erfolgt durch den Einsatz von Zufallsgeneratoren. Diese generieren entsprechend einer zuvor definierten Wahrscheinlichkeitsverteilung (hier: Normalverteilung) Ereignisse zu denen produktvarianten-spezifische Aufträge erzeugt werden. Der mittlere Abstand μ zwischen diesen Ereignissen entspricht dabei einer gleichmäßigen Verteilung über die Planungsperiode (z.B. 12 Lose pro Jahr bedeutet $\mu = 1$ Monat). Die Standardabweichung σ wird als 10 % von μ angenommen (im Beispiel ≈ 3 Tage). Somit werden im Rahmen dieser Grenzen zufallsgesteuerte Auftragseingänge abgebildet.

Verschiedene Steuerungsstrategien regulieren die Zeitpunkte für die nachfolgende Einlastung der Aufträge in die Produktionsumgebung selbst. Hier lassen sich gleichwohl klassische Prioritätsregeln, aber auch komplexere Mechanismen der Reihenfolge- und Freigabeplanung integrieren.

Der Materialfluss findet zwischen den Betriebsmitteln statt und bezieht die dazu benötigten Informationen aus dem Arbeitsplan. Die dazu notwendigen Transportprozesse wurden hierbei nicht modelliert, da diese im Vergleich zu den Bearbeitungs- und Rüstzeiten vernachlässigbar sind. Der Produktvariantenspezifische Arbeitsplan enthält neben der Arbeitsfolge (Bearbeitungsschritte und Maschinenzuordnung) auch weitere Informationen wie Bearbeitungs- und Rüstzeiten.

Innerhalb der Betriebsmittel-Instanzen findet die eigentliche Abarbeitung der Fertigungslose statt.

Dazu werden wartende Aufträge nach ihrem Eintreffen zunächst auf den Abstapelplatz umgelagert. Hierbei erfolgt die Auswahl des nächsten Auftrags auf Grundlage definierter Strategien (z.B. First-In-First-Out oder auch nach Planreihenfolge). Bezugnehmend auf energetisch günstige Auftragsbearbeitung kann beispielsweise Einfluss auf die Priorisierung der Aufträge genommen werden. Eine Möglichkeit dafür wäre die Variation der Abarbeitungsreihenfolge auf Basis energieeffizienter Fahrweisen der Betriebsmittel. Vom Abstapelplatz aus wird das Los stückweise auf der Einzelstation bearbeitet und im Anschluss auf dem Aufstapelplatz wieder zusammengefasst. Von dort aus geht das fertige Los auf den Ausgangspuffer und ein neues Los kann begonnen werden.

Nach abgeschlossener Wertschöpfung, d.h. nach Abarbeitung aller Stationen im Arbeitsplan findet die Umlagerung der Fertigungslose auf den Warenausgang statt. An dieser Stelle verlassen die Lose das Modell. In diesem Schritt erfolgt die Erfassung von zeitlichen Kenngrößen wie Durchlaufzeit, Bearbeitungsbeginn, Bearbeitungsende oder aber Zykluszeit der Aufträge. Diese Kenngrößen können der Analyse und dem Vergleich unterschiedlicher Steuerungsansätze dienen. Ein Überblick hierzu wird im folgenden Abschnitt präsentiert.

5 Verfolgte Untersuchungsansätze

Die aktuellen Forschungsarbeiten, in die sich die Arbeiten am hier vorgestellten Simulationsmodell einordnen, verfolgen das Ziel den Anteil erneuerbarer Energien im Einsatz in Produktionsunternehmen zu erhöhen. Dabei geht es primär darum, Energiequellen wie Photovoltaik- oder Windkraftanlagen dezentral in das vorhandene Energieangebot zu integrieren und die Verwendung konventionell bezogener Energie zu minimieren. Eine Kernforschungsfrage, die sich hieraus ableitet, ist die Flexibilisierung des Energiebedarfs durch produktionsorgansiatorische und technische Maßnahmen. Insbesondere organisatorisch lassen sich aufgrund von Warte-, Liege- und Rüstzeiten, Wartungs- und Instandhaltungsmanagement oder auch fertigungsentkoppelnden Produktpuffern Flexibilitätspotentiale durch die Möglichkeit einer zeitlichen Entkopplung von Energieerzeugung zu Energieverbrauch identifizieren.

Um geeignete Betriebsstrategien für die diese Flexibilisierung zu entwickeln, ist es zunächst notwendig, Abhängigkeiten von Produktionsanlagen, versorgender Medien-Infrastruktur, Produktionslogistik als

auch Gebäudetechnik zu modellieren. Somit sollen sich Eingriffshorizonte (zur Verfügung stehende Pausenzeiten, Pufferdimensionen, Zeitfenster für Anlagenabschaltung, Zuschaltung dezentraler Energiequellen wie ein BHKW, etc.) definieren lassen.

Mittels virtueller Restrukturierung von Produktionsbereichen sollen flexiblere, steuerbarere Einheiten unter vorgenannten Zielbedingungen geschaffen und untersucht werden. Damit die sich hieraus ergebenden Steuerungsszenarien unter Berücksichtigung logistischer Zielstellungen (Durchlaufzeit, Termitreue, etc.) analysieren lassen, wird das vorgestellte Simulationsmodell genutzt. In diesem können verschiedene Flexibilitäten (Anlagenabschalt- und Pufferauslastungsszenarien, energieorientierte Sequenzierungs- und Terminierungspläne, predictive Prozessmedienherzeugung) abgebildet, analysiert und bewertet werden.

Überdies soll das entstandene Modell für die Weiterentwicklung von eniBRIC genutzt werden. Konkret soll die Klassenbibliothek so erweitert werden, dass mit ihr auch erneuerbare Energiequellen sowie Energiespeicher mitsimuliert werden können. Dabei hilft das Simulationsmodell insbesondere Funktionalitäten für das Energie- und Speichermanagement zu erproben. Hierbei stellt das Zusammenspiel zwischen konventionellen und volatilen Energiequellen eine zentrale Herausforderung für die Modellierung dar. Überdies lassen sich auf diesem Wege neue Ansätze zur Produktionssystemplanung unter Einbezug volatiler, dezentraler Energiequellen simulativ untersuchen.

6 Zusammenfassung und Ausblick

Dieser Beitrag behandelt die Systemanalyse für und die Implementierung eines Simulationsmodells der E³-Forschungsfabrik des Fraunhofer IWU. Hierzu wurde zunächst diskutiert, wie die um energetische Betrachtungen erweiterte Materialflusssimulation bereits zur Untersuchung von energie- und ressourceneffizienten Betriebsstrategien genutzt wird. Das letztlich entwickelte und vorgestellte Modell erlaubt es einen Produktionsfluss in der E³-Forschungsfabrik zu simulieren, welcher aufbauend auf Realdaten die Untersuchung neuer Ansätze zum Betriebsmanagement erlaubt. Erste geplante Ansätze dazu wurden ebenfalls dargestellt. Weitere Arbeiten werden sich insbesondere auf deren Integration in das Simulationsmodell beziehen, wobei Algorithmen zur energie-sensitiven Produktionsplanung einen ersten Fokus darstellen.

Die diesem Artikel zugrundeliegenden Vorhaben wurden durch das Fraunhofer-Leitprojekt »E³-Produktion« gefördert. Die Verantwortung für den Inhalt liegt bei den Autoren.

7 References

- [1] E. Müller, J. Engelmann, T. Löffler, und J. Strauch. *Energieeffiziente Fabriken planen und betreiben*. Springer Berlin Heidelberg, Deutschland, 2009.
- [2] Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V. *E3-Forschungsfabrik*. <http://www.e3-fabrik.de/> [letzter Zugriff 12.04.2016].
- [3] T. Ōno. *Toyota production system: beyond large-scale production*. Productivity Press, USA, 1988.
- [4] S. Thiede, C. Herrmann, und S. Kara. *State of Research and an innovative Approach for simulating Energy Flows of Manufacturing Systems*. In: *Glocalized Solutions for Sustainability in Manufacturing*, Herausgeber: J. Hesselbach und C. Herrmann, Springer Berlin Heidelberg, Deutschland, S. 335–340, 2011.
- [5] A. Sproedt, J. Plehn, P. Schönsleben, und C. Herrmann. *A simulation-based decision support for eco-efficiency improvements in production systems*. *Journal of Cleaner Production*. Bd. 105, S. 389–405, 2015.
- [6] E. Fuss, S. Drvendzija, und A. Krewald. *Transparenz als Schlüssel zur energieeffizienten Produktion*. *Productivity management*, Bd. 18 (2013), Nr. 3, S. 35–38, 2013.
- [7] C. Herrmann und S. Thiede. *Process chain simulation to foster energy efficiency in manufacturing*. *CIRP Journal of Manufacturing Science and Technology*. Bd. 1, Nr. 4, S. 221–229, 2009.
- [8] M. Selmair, F. Herrmann, T. Claus, und M. Teich. *Potentiale in der Reduzierung des Gesamtenergieverbrauchs einer Werkstattfertigung in der Maschinenbelegungsplanung*. In: *Simulation in production und logistics 2015*, Herausgeber: M. Rabe und U. Clausen, Fraunhofer Verlag, Deutschland, S. 575–584, 2015.
- [9] J. Andersson, A. Skoogh, und B. Johansson. *Environmental Activity Based Cost Using Discrete Event Simulation*. Tagungsband 2011 Winter Simulation Conference, Phoenix, S. 891–902, 2011.
- [10] I. Barletta, J. Andersson, B. Johansson, G. May, und M. Taisch. *Assessing a proposal for an energy-based Overall Equipment Effectiveness indicator through Discrete Event Simulation*. Tagungsband 2014 Winter Simulation Conference, Savannah, S. 1096–1107, 2014.
- [11] F. Bleicher, F. Duer, I. Leobner, I. Kovacic, B. Heinzl, und W. Kastner. *Co-simulation environment for optimizing energy efficiency in production systems*. *CIRP Annals - Manufacturing Technology*. Bd. 63, Nr. 1, S. 441–444, 2014.
- [12] S. Lorenz, M. Hesse, und A. Fischer. *Simulation and optimization of robot driven production systems for peak-load reduction*. Tagungsband 2012 Winter Simulation Conference, Berlin, 2012.
- [13] VDI-Gesellschaft Produktion und Logistik (GPL). *VDI 3633, Blatt 1: Simulation von Logistik-, Materialfluß- und Produktionssystemen—Grundlagen*. Beuth, Deutschland, 2014.
- [14] A. Büchel, M. Becker, W. F. Daenzer u.a. *Systems Engineering*. Verlag Industrielle Organisation, Schweiz, 1994.
- [15] C.-G. Grundig. *Fabrikplanung*. Carl Hanser Verlag, Deutschland, 2009.
- [16] H. Schmigalla. *Fabrikplanung*. Carl Hanser Verlag, Deutschland/Österreich, 1995.
- [17] A. Schlegel, J. Stoldt und M. Putz. *Erweiterte Integration energetischer Betrachtungen in der Materialflusssimulation*. In: *Simulation in Produktion und Logistik 2013: [Entscheidungsunterstützung von der Planung bis zur Steuerung; 15. ASIM Fachtagung]*, Herausgeber: W. Dangelmaier, C. Laroque und A. Klaas, Heinz-Nixdorf-Inst./Univ. Paderborn, Deutschland, S. 187–196, 2013.
- [18] J. Stoldt, A. Schlegel und M. Putz. *Enhanced integration of energy-related considerations in discrete event simulation for manufacturing applications*. *Journal of Simulation*, Bd. 10, Nr. 2, S. 113–122, 2016.

Rechenbeispiel der Makro- und Mikroperioden-Bestellmengenplanung in Lebensmittelfilialen zwecks Abfallreduktion

Larissa Janssen¹, Thorsten Claus¹, Jürgen Sauer²

¹ TU Dresden / IHI Zittau, Dresden (Germany)

² Carl von Ossietzky Universität Oldenburg, Oldenburg (Germany)

larissa.janssen@mailbox.tu-dresden.de

In Lebensmittelfilialen besteht das Problem der Abfallreduktion. Wir haben als Lösungsweg eine mehrmals tägliche Bestellmengenplanung mit Belieferung vorgeschlagen. In dieser Arbeit wird ein Rechenbeispiel gezeigt, wie mithilfe unseres Bestellmengenmodells in Filialen geplant werden kann und wo der Unterschied zu der Planung auf Tagesbasis liegt. Wir planen für vorgepackte schnell verderbliche Fertigsalate mit fixer Haltbarkeit. Wir zeigen in dieser Arbeit, dass die zweimal tägliche Planung insgesamt zu niedrigeren Lagerbeständen (inklusive Sicherheitsbeständen) und Abfallmengen unter Beachtung von Kosten in mehreren Lebensmittelfilialen und einem Regionallager führt.

1 Einleitung

Laut der Studie [7] des Instituts für Siedlungswasserbau, Wassergüte- und Abfallwirtschaft der Universität Stuttgart entstehen allein im Handel (Großmärkte und Lebensmitteleinzelhandel) ca. 5% Abfälle von Lebensmitteln jährlich. Von diesen Abfällen beläuft sich der Anteil der Großmärkte laut der zitierten Studie auf 12,4% (61.000 t) und des Lebensmitteleinzelhandels (LEH) auf 87,6% (490.000 t) pro Jahr. Das zeigt, dass besonders im Lebensmitteleinzelhandel ein Bedarf zur Abfallreduktion besteht. Zu den Abfällen gehören hauptsächlich *Frischware* mit kurzer Haltbarkeit (3-5 Tage) wie beispielsweise vorgepackte Frischsalate, Obst, Gemüse, Fleisch, Fisch, Molkereiprodukte und Backwaren.

Das Abfallproblem hängt in Lebensmittelfilialen meistens mit der Bestellmengenplanung zusammen [7]. Die *Bestellmengenplanung* bezieht sich im Wesentlichen auf Entscheidungen hinsichtlich der Bestellmenge und der Bestell- und Lieferzeitpunkte [3, 4]. Die Bestellmengenplanung wird durch die Nachfrage-Unsicherheit erschwert.

Die Belieferung von Filialen mit Frischware geschieht im Lebensmitteleinzelhandel typischerweise einmal pro Tag [12, 4, 8]. Wir haben in [5] gezeigt, dass eine mehrmals tägliche Bestellmengenplanung und Belieferung für die schnell verderbliche Frischware aus Sicht der Abfallreduktion besser sein können. Das Ziel dieses Beitrags ist die exemplarische Darstellung einer detaillierten Mikro- und Makroperioden-Planung und nicht die Vorstellung des Optimierungsmodells oder der Simulationsstudie selbst. Diese Beispielrechnung verdeutlicht, wie sich die beiden Planungsalternativen (einmal versus zweimal tägliche Planung) auf Abfallmengen und Kosten in Filialen auswirken.

2 Fallstudie - frische Fertigsalate

Wir betrachten einen realistischen Fall der Bestellmengenplanung für vorgepackte frische Fertigsalate in Filialen des Lebensmitteleinzelhandels (LEH), die ganzjährig in Filialen angeboten werden. Die Fertigsalate sind dauerhaft gefragt, wobei die Nachfrage in Sommermonaten am höchsten und in Wintermonaten am geringsten ist. Die Haltbarkeit H der

Salate ist begrenzt. Meistens kommen die Salate in Filialen mit der Haltbarkeit von 3-5 Tagen (je nach Salatsorte) an. Die erwartete Nachfrage D ist zufällig und normalverteilt. Sie ist in diesem Beispiel durch den Mittelwert $\mu = 60$ und die Standardabweichung $\sigma = 30$ genau beschrieben. Ihre Werte d_{ft} können sich wegen der Saisonalität der Nachfrage von Periode zu Periode unterscheiden.

Eine Lieferung von Fertigsalaten führt der Hersteller in den vereinbarten Mengen zu den gewünschten Terminen an ein Zentrallager oder auch an mehrere Regionallager durch. Das Regionallager verteilt anschließend die Ware an seine Filialen. Die gelieferte Ware wird in Filialen i.d.R. sofort in Kühlregale eingeräumt [4]. Die Regalkapazität G für die Fertigsalate ist begrenzt, jedoch ausreichend groß, um die Nachfrage zum gewünschten Servicegrad für mehrere Planungsperioden decken zu können.

3 Bestellmengenmodell

Unser stochastisches altersbasiertes Mikroperioden-Bestellmengenmodell für verderbliche Güter mit fixer (deterministischer) Haltbarkeit wurde bereits publiziert [5]. Die Bestellmengenplanung auf Tagesbasis wird als *Makroperioden*-Planung und eine mehrmals tägliche Planung und Belieferung wird als *Mikroperioden*-Planung bezeichnet. Es gibt in einer Makroperiode vier Mikroperioden ($V = 4$): Morgen, Mittag, Nachmittag und Abend.

Die Nachfrage sei normalverteilt. Fehlmengen sind nicht erlaubt. Der Umsatz aus der nicht befriedigten Nachfrage geht unberücksichtigt verloren. Der Bestellabstand R ist über die fixen Bestellkosten definiert. Die Wiederbeschaffungszeit L ist deterministisch. Es wird rollierend geplant. Nur die Bestellentscheidung der Periode t ist verbindlich und wird realisiert. Der verbliebene Restlagerbestand x_{fh} und die Lieferung y_{ft} aus dem vergangenen rollierenden Planungshorizont gehen in das Optimierungsmodell als Parameter ein.

Um den gewünschten Kundenservicegrad unter Nachfrage-Unsicherheit zu realisieren, wird ein Sicherheitsbestand in Filialen gehalten. Es wird von dem β -Servicegrad nach [10] ausgegangen. Der Sicherheitsbestand wird für den Risikozeitraum $r =$

$R + L$ nach [10] ermittelt. Nach R -Perioden wird die Bestellentscheidung erneut getroffen. Der Sicherheitsbestand wird direkt zu der Nachfrage aufaddiert und damit bei Bedarf mitbestellt. Zur Abbildung des altersbasierten Bestandes wird das deterministische Bestellmengenmodell für verderbliche Güter von Häselbarth und Scholl (vgl. [2]) verwendet, welches das klassische dynamische Bestellmengenmodell von Wagner und Whitin erweitert. Zu unseren wichtigsten Erweiterungen des Modells [2] zählen: Mikroperioden-Planung, Berücksichtigung der stochastischen Nachfrage, positive Wiederbeschaffungszeit, LIFO- und FIFO-Entnahmeart sowie Einbeziehung der Transportkosten eines Regionallagers (RL) und der Filialen (Fil.).

Der Geltungsbereich von f , t und h ist in allen Gleichungen, falls nicht anders angegeben, identisch:

$$\begin{aligned} f: & \text{Filiale} & F: & \text{Filialenanzahl} & \forall f \in 1..F \\ h: & \text{Alter} & H: & \text{max. Alter} & \forall h \in 0..H \\ t: & \text{Periode} & T: & \text{Planungshorizont} & \forall t \in 1..T \end{aligned}$$

Parameter:

BK_{ft} : fixe Bestellkosten
 c_{ft} : variable Einkaufskosten
 d_{ft} : Nachfragemenge
 $d_{ft}^{k \in \tau}$: Nachfrage inkl. vorzeit. Abfallmenge
 \tilde{D}_{ft}^{τ} : kumulierte Nachfrage
 $\varepsilon_{fh}, \varepsilon_{f,j \in h}$: Verbleibfaktor der Frischware
 ϕ^{LIFO} : Nachfrageanteil, LIFO-Prinzip
 G_{ft} : maximale Lagerkapazität
 LK_{ft} : Lagerhaltungskosten
 L : Wiederbeschaffungszeit
 ω_{ft}^{fix} : fixe Transportkosten in Fil.
 ω_{ft}^{var} : variable Transportkosten in Fil.
 ωRL_t^{fix} : fixe Transportkosten des RL pro Tour
 ωRL_t^{var} : variable Transportkosten des RL
 Q_t^{max} : max. Kapazität eines LKW
 t^{makro} : Makroperiode
 sb_{ft} : Sicherheitsbestand für den Zeitraum $R + L$
 τ : eine Folgeperiode von t
Morgen: erste Periode an einem Tag
 x_{fh} : Restlagerbestand nach Alter h
 y_{ft} : Lieferung in Periode t

Entscheidungs- und Zustandsvariablen:

d_{fth}^{FIFO} : FIFO-Nachfrage vom Alter h
 d_{fth}^{LIFO} : LIFO-Nachfrage vom Alter h

q_{ft} : Bestellmenge in Periode t der Filiale f
 X_{fth} : Lagerbestand in Periode t nach Alter h
 X_{fth}^{free} : abfallfreier Lagerbestand der Filiale f
 z_{ft} : binärer Bestellflag in Periode t

Die Zielfunktion (1) enthält fixe Bestellkosten, Lagerhaltungskosten, Einkaufskosten, Lagerkosten für den Sicherheitsbestand des ersten Bestellzyklus und Transportkosten von Filialen und einem Regionallager in allen Perioden [5].

$$\begin{aligned}
 \text{Min } C = & \sum_{f=1}^F \sum_{t=1}^T BK_{ft} \cdot z_{ft} + LK_{ft} \cdot X_{ft} + c_{ft} \cdot q_{ft} \quad (1) \\
 & + \sum_{t=L+1}^T LK_{ft} \cdot sb_{ft} + \sum_{f=1}^F \sum_{t=1}^T \omega_{ft}^{fix} \cdot z_{ft} + \omega_{ft}^{var} \cdot q_{ft} \\
 & + \sum_{f=1}^F \sum_{t=1}^T z_{ft} \cdot \omega RL_t^{fix} + q_{ft} \cdot \omega RL_t^{var}
 \end{aligned}$$

Mit Geltung der folgenden Restriktionen:

$$X_{ft,h=0}^{free} = 0, t = \text{Morgen} \quad (2)$$

$$X_{ft,h>0}^{free} = \varepsilon_{fh} \cdot X_{f,t-1,h-1}, t = \text{Morgen} \quad (3)$$

$$X_{fth}^{free} = \varepsilon_{fh} \cdot X_{f,t-1,h}, t \neq \text{Morgen} \quad (4)$$

$$X_{f,t=0,h} = x_{fh} \quad (5)$$

$$X_{ft,h=0} = X_{ft,h=0}^{free} + y_{ft} \quad (6)$$

$$-d_{ft,h=0}^{LIFO} - d_{ft,h=0}^{FIFO}, t = 1, \dots, L$$

$$X_{ft,h=0} = X_{ft,h=0}^{free} + q_{f,t-L} + y_{ft} \quad (7)$$

$$-d_{ft,h=0}^{LIFO} - d_{ft,h=0}^{FIFO}, t = L+1, \dots, T$$

$$X_{ft,h>0} = X_{ft,h>0}^{free} - d_{ft,h>0}^{LIFO} - d_{ft,h>0}^{FIFO} \quad (8)$$

$$d_{ft}^{LIFO} = d_{ft} \cdot \phi^{LIFO} \quad (9)$$

$$d_{ft}^{FIFO} = d_{ft} - d_{ft}^{LIFO} \quad (10)$$

$$X_{fth} = 0 \vee d_{fth}^{LIFO} = d_{ft}^{LIFO} - \sum_{j=0}^{h-1} d_{ftj}^{LIFO} \quad (11)$$

$$X_{ft,H-h} = 0 \vee d_{ft,H-h}^{FIFO} = d_{ft}^{FIFO} - \sum_{j=H-h+1}^H d_{ftj}^{FIFO} \quad (12)$$

$$\sum_{h=0}^H d_{fth}^{LIFO} = d_{ft}^{LIFO}, \sum_{h=0}^H d_{fth}^{FIFO} = d_{ft}^{FIFO} \quad (13)$$

$$\sum_{h=0}^H X_{fth} \leq G_f \quad (14)$$

$$X_{fth}^{free} \geq 0, d_{fth}^{LIFO} \geq 0, d_{fth}^{FIFO} \geq 0 \quad (15)$$

$$q_{ft} - \tilde{D}_{ft}^{\tau(t,L)} \cdot z_{ft} \leq 0 \quad (16)$$

$$\begin{aligned}
 \tilde{D}_{ft}^{\tau} &= d_{f,t+L} + \sum_{k=t+L+1}^{\tau} \tilde{d}_{ft}^k, \quad (17) \\
 \tau &\in [t+L+1, \dots, \bar{T}(t,L)]
 \end{aligned}$$

$$\tilde{d}_{ft}^k = d_{f\tau} / \prod_{j=0}^{h'} \lambda_j, h' = \max(0, t_{\tau}^{macro} - t_{t+L}^{macro}) \quad (18)$$

Den abfallfreien Lagerbestand bezeichnen wir als X_{fth}^{free} . Am Morgen gibt es bei der ganz frischen Ware ($h=0$) keine Abfälle (Gl. 2). Ansonsten ($h>0$) wird am Morgen nach der Alterung um einen Tag ($h-1$) der abfallfreie Lagerbestand ermittelt (Gl. 3). Am Mittag, Nachmittag und Abend wird X_{fth}^{free} aus der vorigen Mikroperiode $t-1$ des gleichen Alters ermittelt (Gl. 4).

Um zu erfahren, ob es ein Morgen, Mittag, Nachmittag oder Abend ist (Gl.2 - Gl. 4) wird eine Mikroperiode relativ zu einer Makroperiode betrachtet. Die relative Mikroperiode eines Tages wird aus dem Modulo bestimmt: *Ergebnis* = $t \bmod V$, wobei V die Anzahl von Mikroperioden in einer Makroperiode ist. In unserem Fall ist $V=4$. Zum Beispiel $t=9$ *daytime=Morgen* ($9 \bmod 4=1$), $t=10$ *Mittag*, $t=11$ *Nachmittag*, $t=12$ *Abend*.

Der Restlagerbestand x_{fh} aus dem vergangenen Planungshorizont wird in $t=1$ angerechnet (Gl. 5). Ist die Wiederbeschaffungszeit positiv ($L>0$) und $h=0$, trifft in $t \in [1, \dots, L]$ keine Lieferung q_{ft} ein (Gl. 6). In Periode $t \in [L+1, \dots, T]$ kann eine Lieferung mit $h=0$ aus der Bestellperiode $t-L$ eintreffen (Gl. 7). Ist das Alter $h>0$, so entfällt aus der Lagerbilanz die Bestellung q_{ft} und die Lieferung y_{ft} (Gl. 8).

Die Nachfrage d_{ft} wird nach Entnahmekategorien in zwei Nachfragemengen d_{ft}^{LIFO} und d_{ft}^{FIFO} zerlegt (Gl. 9 und Gl. 10). Falls der Lagerbestand X_{fth} mit der Frischware des Alters h geringer als oder gleich wie die benötigte Nachfragemenge aus d_{ft}^{LIFO} ist, wird dieser Bestand ganz entnommen (Gl. 11, Term links). Falls der Lagerbestand X_{fth} größer ist, wird aus dem Lager so viel entnommen, wie für die Nachfrage d_{ft}^{LIFO} noch fehlt (Gl. 11, Term rechts). Das Gleiche gilt auch für die Nachfrage d_{ft}^{FIFO} (Gl. 12).

d_{fth}^{LIFO} und d_{fth}^{FIFO} sind jeweils nach Alter h aufgeteilte Nachfragemengen, die je nach verfügbarem Lagerbestand die Nachfrage d_{ft}^{LIFO} und d_{ft}^{FIFO} wiedergeben (Gl. 13). Die Lagerkapazität ist begrenzt (Gl. 14). Nichtnegativitätsbedingungen sind in Gleichung (15)

zusammengefasst.

Gleichung (16) ergibt sich aus der Beachtung der spätesten Verbrauchsperiode $\bar{T}(t, L) = \min(T, t + L + (H + 1) \cdot V - \bar{t})$. Da die Alterung der Frischware um einen Tag nur am Morgen angenommen wird, ergibt sich die Korrektur der Verbrauchsperiode um $\bar{t} = (t + L) \bmod V$ falls $(t + L) \bmod V \neq 0$, sonst $\bar{t} = V - ((t + L) \bmod V)$ Mikroperioden. $\bar{D}_{ft}^{\bar{T}(t, L)}$ ist die kumulierte modifizierte Bedarfsmenge, die wie folgt in (Gl. 17) berechnet wird. Die Bedarfsmenge \bar{d}_{ft}^k enthält auch die Abfallmenge aus Qualitätsgründen (Mortalität etc.), wobei h' das Alter der Frischware von Periode t bis τ (Gl. 18) ist. Die kumulierte erwartete Nachfrage $\bar{D}_{ft}^{\bar{T}(t, L)}$ ist 0 während der Wiederbeschaffungszeit (d.h. $\tau - t < L$) oder die letzte Verbrauchsperiode $\bar{T}(t, L)$ ist überschritten.

Eine Makroperiode t_t^{macro} wird nach dieser Gleichung bestimmt: $t_t^{macro} = \lfloor (t - 1)/V \rfloor + 1$. Wenn die Mikroperiode $t = 9$ ist, dann liegt sie in der Makroperiode 3: $t_9^{macro} = \lfloor (9 - 1)/4 \rfloor + 1 = 3$.

4 Rechenbeispiel

Die Unterschiede in der Makro- versus Mikroperioden-Planung verdeutlichen wir in diesem Abschnitt anhand eines kleinen Rechenbeispiels. Zur Simulation wird das Tool Tecnomatix Plant Simulation 11 eingesetzt. Es handelt sich um ein zeitdiskretes Simulationsmodell mit vier Zeitscheiben (Mikroperioden) pro Tag. Das Bestellmengenproblem (Abschnitt 3) wird mittels IBM ILOG Solver gelöst. Nachfragewerte (Inputdaten des Modells) werden mittels der Monte-Carlo-Methode [6] generiert. Die generierten Zufallszahlen sind voneinander unabhängig und folgen der Verteilung der Nachfrage. Pro Experiment gibt es 70 Replikationen. Die Einschwingphase beträgt 3 Monate. Jedes Experiment endet nach ca. 3.000 (2 Jahren) Planungsperioden. In Experimenten wird der Präzisionsgrad für den β -Servicegrad $\pm 0,002$ beim 95%-Konfidenzniveau erreicht.

Die beiden Planungsmethoden unterscheiden sich durch die Parameter, die in Tabelle 1 gezeigt sind. Der *Bestellabstand* (Anzahl der Perioden, deren Nachfragemenge gemeinsam bestellt wird [9]) R beträgt bei der Makroperioden-Planung einen Tag ($R = 4$). R wird über die fixen Bestellkosten nach

dem klassischen Losgrößenproblem von Harris [1] ($c_{ft} = R^2/2 \cdot 1/R \cdot \sum_{t=1}^{t+R-1} d_{ft} \cdot a_{ft}$) eingestellt. Die fixen Transportkosten verändern den Bestellabstand R in diesem Beispiel unwesentlich. Frischware wird bei der Makroperioden-Planung maximal einmal täglich geliefert (Lieferzeit $L = 4$). Die Planung im Mikroperioden-Fall wird dagegen zweimal täglich durchgeführt ($R = 2$). Die Lieferzeit beträgt in diesem Fall nur eine Mikroperiode ($L = 1$), deshalb kann die Lieferung am selben Tag erfolgen. Parameter sb_{ft} und $X_{f,1,0}^{free}$ werden weiter unten erläutert.

	R	L	c_{ft}	sb_{ft}	$X_{f,1,0}^{free}$
Mikro	2	1	360	88	268
Makro	4	4	1440	137	617

Tabelle 1: Einstellungen der Mikro- und Makro-Optimierungsfälle

Die Nachfrageparameter sind in Tab. 2 dargestellt. In Abhängigkeit von der Wiederbeschaffungszeit L enthält die Nachfrage d_{ft} in Periode $L + 1$ den ex ante berechneten Sicherheitsbestand sb_{ft} für den Risikozeitraum $R + L$. Die erwartete Nachfrage inklusive Sicherheitsbestand beträgt in Periode t_2 ($L + 1$) $d_{f,t=2} = 148$ ME (bei $L=1$) bzw. $d_{f,t=5} = 197$ ME (bei $L=4$), sonst ist $d_{ft} = 60$ ME.

t	1	2	3	4	5	6	7	8	9	10	11	12
$L=1$	60	60	60	60	60	60	60	60	60	60	60	60
		+88										
$L=4$	60	60	60	60	60	60	60	60	60	60	60	60
				+137								

Tabelle 2: Nachfrageparameter d_{ft} für das Optimierungsmodell

Der Sicherheitsbestand sb_{ft} ist für die beiden Fälle nach [10] berechnet. Der Ziel- β -Servicegrad ist 99,20%. Der Sicherheitsbestand für den Makroperioden-Fall wird wie folgt ermittelt: Im Zeitraum $R + L$ ist die kumulierte Nachfrage $d_t^{R+L} = \sum_{i=t}^{t+R+L-1} d_{f,i+L} = \sum_{i=1}^8 60 = 480$ ME und im Zeitraum von R -Perioden ist sie $d_t^R = \sum_{i=1}^4 60 = 240$. Die kumulierte Standardabweichung im Zeitraum $R + L$ ist $\sigma_t^{R+L} = \sqrt{\sum_{i=t}^{t+R+L-1} \sigma_{f,i+L}^2} = \sqrt{\sum_{i=1}^8 30^2} = 84,85$ (ME). Die erlaubte standardisierte Fehlmenge für den Risikozeitraum $R + L$ ergibt sich aus: $F_t^{R+L} = ((1 - \beta) \cdot d_t^{R+L}) / \sigma_t^{R+L} = (1 - 0,992) \cdot 240 / 84,85 \approx 0,0226$. Anschließend wird mithilfe der Inverse der Verlust-

funktion G_{R+L}^{-1} erster Ordnung der Sicherheitsfaktor bestimmt: $sf_t^{R+L} = G_{R+L}^{-1}(F_t^{R+L} = 0,0226) = 1,61$. Anschließend wird der Sicherheitsbestand bestimmt: $sb_{ft} = \sigma_t^{R+L} \cdot sf_t^{R+L} = 84,85 \cdot 1,61 \approx 137$. Analog dazu wird der Sicherheitsbestand für den Mikroperioden-Fall berechnet, wobei $R = 2$ und $L = 1$.

Der Anfangslagerbestand $X_{f,1,0}^{free}$ (siehe Tab. 1) wird für Periode $t = 0$ wie folgt berechnet. Im Mikroperioden-Fall ist $X_{f,1,0}^{free} = sb_{ft} + d_t^{R+L} = 88 + 60 \cdot 3 = 268$ und im Makroperioden-Fall ist $X_{f,1,0}^{free} = 137 + 60 \cdot 8 = 617$.

Die gemeinsamen Modellparameter der Mikro- und Makroperioden-Planung sind in Tab. 3 dargestellt. Die Planung erfolgt in beiden Fällen rollierend [9]. Der rollierende Planungshorizont beträgt 12 Mikroperioden (3 Tage). Die Warenentnahme in Filialen erfolgt in diesem Beispiel zu 60% nach dem LIFO- und zu 40% nach dem FIFO-Prinzip.

Monetäre Parameter	
Fixe Transp.kosten (Fil.) ω_{ft}^{fix}	188 GE/Liefer.
Var. Transp.kosten (Fil.) ω_{ft}^{var}	7 GE
Fixe Transp.kosten (RL) ωRL_t^{fix}	540 GE/Fil.
Var. Transp.kosten (RL) ωRL_t^{var}	4 GE
Lagerhaltungskosten a_{ft}	2 GE/ME
Fixe Bestellkosten c_{ft} bei $L=1$	360 GE
Fixe Bestellkosten c_{ft} bei $L=4$	1440 GE
Var. Bestellkosten b_{ft}	60 GE
Abfallkosten w_{ft}	64 GE/ME
Nicht monetäre Parameter	
Planungshorizont T	12
Max. Haltbarkeit H	2 (3 Tage)
Verbleibfaktor der Ware ε_{fh}	$\{1, 1, 1\}$
Erwartete Nachfrage d_{ft}	$\mu=60, \sigma=30$
Max. Lagerkapazität G_{fk}	300/700
Max. LKW-Kapazität	14400 QU
FIFO-/LIFO-Nachfrageanteil	60%/40%
β -Servicegrad	99,20%

Tabelle 3: Parameter der Mikro- und Makroperioden-Planung

Die Werte aus Tabelle 5 dienen zur Einschränkung der maximalen Bestellmenge für die Makroperioden-Planung. Die kumulierte Nachfrage \tilde{D}_{ft}^{τ} wird unmittelbar vor der Optimierung generiert. Die Berechnung von \tilde{D}_{ft}^{τ} (Tab. 5 und Tab. 6) geschieht nach Gl. (17) und (18). Die Vorgehensweise wird anhand eines

\tilde{D}_{ft}^{τ}	Berechnungsdetails der Makroperioden-Planung mit $L = 4$ (siehe Tab. 5)
0	$t = 1, \tau = 1, \dots, 4$ wegen $\tau - t < L$
197	$t = 1, \tau = 5$ nach Gl. (17): $\tilde{D}_{f1}^5 = d_{f,t+L} + \sum_{k=t+L+1}^{\tau} \tilde{d}_{ft}^k = d_{f,1+4} + \sum_{k=6}^5 \tilde{d}_{ft}^k = d_{f5} + 0 = 60 + 137 = 197$. $d_{f5} = 60 + 137$ (siehe Tab. 2 für $L = 4$).
257	$t = 1, \tau = 6$ nach Gl. (17): $\tilde{D}_{f1}^6 = d_{f,1+4} + \sum_{k=6}^6 \tilde{d}_{f1}^k = d_{f5} + \tilde{d}_{f1}^6 = 197 + 60 = 257$, wo $\tilde{d}_{f1}^6 = d_{f6} / \prod_{j=0}^{h'} \varepsilon_{fj}$. Das fortschreitende Alter berücksichtigen: $h' = \max(0, t_6^{macro} - t_5^{macro}) = \max(0, 2 - 2) = 0$. Das ergibt $\varepsilon_{f0} = 1$. Die Umrechnung der Mikro- in Makroperiode: $t_6^{macro} = \lfloor (6-1)/4 \rfloor + 1 = 2$ und $t_5^{macro} = \lfloor (5-1)/4 \rfloor + 1 = 2$. Beim Einsatz von ε_{f0} ergibt sich die Menge $\tilde{d}_{f1}^6 = d_{f6}/1 = d_{f6} = 60$.

Tabelle 4: Berechnung von \tilde{D}_{ft}^{τ} für $t = 1, \tau = 5$ und $t = 1, \tau = 6$ der Tab. 5

Beispiels in Tab. 4 geschildert. Es wird gezeigt, wie die Werte 197 und 257 für $t = 1, \tau = 5$ und $t = 1, \tau = 6$ der Tab. 5 ermittelt werden.

Die Interpretation der Daten aus Tab. 5 ist wie folgt: In Periode $t = 1$ können für Folgeperioden $\tau = 5, \dots, 12$ bis zu 617 ME mitbestellt werden. Wäre die Haltbarkeit der Frischware 1 Tag, so hätte man maximal 377 ($\tilde{D}_{f,t=1}^{\tau=8} = 377$) bestellen können. Dann wären $\tilde{D}_{f,t=1}^{\tau=9, \dots, 12} = 0$.

Die Berechnung der Werte der Tab. 6 geschieht analog, jedoch wird die Wiederbeschaffungszeit $L = 1$ und die Nachfrage für $L = 1$ aus Tab. 2 eingesetzt. Hätten wir in unserem Rechenbeispiel den Abfall aus Qualitätsgründen (Mortalität, Schimmelbefall etc.) zugelassen, wären diese Mengen in \tilde{D}_{ft}^{τ} ebenfalls enthalten.

Tabelle 8 zeigt beispielhaft, wie auf Basis von Makro- und Mikroperioden rollierend geplant wird.

Einfachheitshalber wird hier die rollierende Planung von 13 Mikroperioden für eine Lebensmittelfiliale gezeigt. Die erste Mikroperiode ($t = 1$) ist ein Donnerstag und die letzte Periode ($t = 13$) ist ein Sonntag. Am Sonntag ist die Filiale geschlossen, deshalb findet hier keine Planung statt und die Nachfrage ist null. Die vorzeitige Verschlechterung der Qualität bei Frischware schließen wir einfachheitshalber aus, um

t	1	2	3	4	5	6	7	8	9	10	11	12
1	L	L	L	L	197	257	317	377	437	497	557	617
2	-	L	L	L	L	60	120	180	240	300	360	420
3	-	-	L	L	L	L	60	120	180	240	300	360
4	-	-	-	L	L	L	L	60	120	180	240	300
5	-	-	-	-	L	L	L	L	60	120	180	240
6	-	-	-	-	-	L	L	L	L	60	120	180
7	-	-	-	-	-	-	L	L	L	L	60	120
8	-	-	-	-	-	-	-	L	L	L	L	60
9	-	-	-	-	-	-	-	-	L	L	L	L
10	-	-	-	-	-	-	-	-	-	L	L	L
11	-	-	-	-	-	-	-	-	-	-	L	L
12	-	-	-	-	-	-	-	-	-	-	-	L

Tabelle 5: Kumulierte Nachfrage \tilde{D}_{ft}^τ bei der Wiederbeschaffungszeit $L=4$ (Makroplanung)

t	1	2	3	4	5	6	7	8	9	10	11	12
1	L	148	208	268	328	388	448	508	568	628	688	748
2	-	L	60	120	180	240	300	360	420	480	540	600
3	-	-	L	60	120	180	240	300	360	420	480	540
4	-	-	-	L	60	120	180	240	300	360	420	480
5	-	-	-	-	L	60	120	180	240	300	360	420
6	-	-	-	-	-	L	60	120	180	240	300	360
7	-	-	-	-	-	-	L	60	120	180	240	300
8	-	-	-	-	-	-	-	L	60	120	180	240
9	-	-	-	-	-	-	-	-	L	60	120	180
10	-	-	-	-	-	-	-	-	-	L	60	120
11	-	-	-	-	-	-	-	-	-	-	L	60
12	-	-	-	-	-	-	-	-	-	-	-	L

Tabelle 6: Kumulierte Nachfrage \tilde{D}_{ft}^τ bei der Wiederbeschaffungszeit $L=1$ (Mikroplanung)

die Unterschiede der beiden Planungen zu verdeutlichen.

Die realisierte Nachfrage d'_{ft} in Perioden 1 bis 12 ist in Tab. 7 gezeigt. Diese Nachfrage wird für die FIFO- und LIFO-Warenentnahme in die Nachfragen d_{ft}^{FIFO} und d_{ft}^{LIFO} eingeteilt und in der Beispielrechnung der Tab. 8 verwendet.

t	1	2	3	4	5	6	7	8	9	10	11	12
d'_{ft}	66	67	67	48	96	81	57	78	15	80	15	3

Tabelle 7: Die realisierte Nachfrage d'_{ft} des Rechenbeispiels

Die Einzelheiten der Planung sind in Tabelle 8 dargestellt. In der Tabelle wird zuerst die Makro- und darunter die Mikroperioden-Planung gezeigt. Es gibt insgesamt 8 Planungsschritte, die sich in jeder Periode wiederholen. Im 1. Schritt wird die Abfallmenge w_{ft}

bestimmt und der Lagerbestand von Abfällen bereinigt. Im 2. Schritt wird gegebenenfalls eine Lieferung y_{ft} empfangen und sofort in das Lager (oder in Verkaufsregale) eingeräumt. Im 3. Schritt wird der initiale Lagerbestand (X_{fth}^{free}) einer Mikroperiode bestimmt, der von Abfällen bereits bereinigt ist. Da der Lagerbestand altersbasiert ist, sind die Frischwaremengen im Lager nach Alter ($h=0, h=1$ und $h=2$) bekannt. Im 4. Schritt wird geprüft, ob eine Bestandskontrolle R in der Periode t erfolgen soll. Falls ja, wird im 5. Schritt, falls notwendig, die Frischware bestellt. Die optimale Bestellmenge q_{ft} wird mittels unseres Bestellmengenmodells aus Abschnitt 3 in beiden Planungsfällen bestimmt. Im 6. Schritt wird die Nachfrage d'_{ft} erfüllt, die aus den FIFO- und LIFO-Anteilen (d_{ft}^{FIFO} und d_{ft}^{LIFO}) besteht. Am Ende der Periode t steht im 7. Schritt der Endlagerbestand fest. Die Kosten der Periode t werden im 8. Schritt ermittelt. Zu den Kosten gehören die Filialenkosten C_{fk} (inklusive Transportkosten) und die Transportkosten des Regionallagers C_{fk}^{RL} . Am Sonntag, in Periode $t=13$, können nur Abfallkosten und Lagerhaltungskosten in den Filialen entstehen.

Die durchschnittlichen Ergebnisse von 12 Perioden des Rechenbeispiels sind in Tabelle 9 zusammengefasst. Die Bestellmenge im Makroperioden-Fall ist im Durchschnitt doppelt so groß. Der durchschnittliche Lagerbestand ist ebenfalls bei der Makroperioden-Planung um 94% höher. Die durchschnittlichen Kosten der Filiale und des Regionallagers liegen im Mikroperioden-Fall leicht (um 1,20%) niedriger. Für die höheren Kosten des Makroperioden-Falls sind hauptsächlich die größeren Lagerbestände verantwortlich. Die Lieferanzahl ist bei der Mikroperioden-Planung stark angestiegen, was auch erwartet wird. Die Abfallmenge beträgt in der 13. Periode im Makroperioden-Fall $w_{ft}=12$, dagegen gab es im Mikroperioden-Fall keine Abfälle.

5 Zusammenfassung

In dieser Arbeit haben wir an einem Rechenbeispiel die rollierende Makro- und Mikroperioden-Bestellmengenplanung in Lebensmittelfilialen gezeigt. Schon in diesem einfachen Rechenbeispiel sind Auswirkungen der Mikroperioden-Planung zu beobachten. Die Planung in Makroperioden

Makroperioden-Planung														
Schritt	t:	1	2	3	4	5	6	7	8	9	10	11	12	13
1.	W_{ft}	0	0	0	0	0	0	0	0	0	0	0	0	12
2.	y_{ft}	0	0	0	0	0	0	0	0	248	0	0	0	-
3.	$X_{ft,h=0}^{free}$	617	551	484	417	0	0	0	0	248	239	191	184	0
	$X_{ft,h=1}^{free}$	0	0	0	0	369	273	192	135	0	0	0	0	183
	$X_{ft,h=2}^{free}$	0	0	0	0	0	0	0	0	57	51	19	14	0
4.	R	ja	-	-	-	ja	-	-	-	ja	-	-	-	-
5.	q_{ft}	0	-	-	-	248	-	-	-	312	-	-	107	-
6.	d_{ft}^{FIFO}	27	27	27	20	39	33	23	32	6	32	5	2	-
	d_{ft}^{LIFO}	39	40	40	28	57	48	34	46	9	48	7	1	-
7.	$X_{ft,h=0}$	551	484	417	369	0	0	0	0	239	191	184	183	-
	$X_{ft,h=1}$	0	0	0	0	273	192	135	57	0	0	0	0	-
	$X_{ft,h=2}$	0	0	0	0	0	0	0	0	51	19	14	12	-
8.	C_{fk}	1653	1452	1251	1107	17139	576	405	171	22954	630	594	8445	1317
	C_{fk}^{RL}	0	0	0	0	0	0	0	0	1532	0	0	0	0
Mikroperioden-Planung														
Schritt	t:	1	2	3	4	5	6	7	8	9	10	11	12	13
1.	W_{ft}	0	0	0	0	0	0	0	0	0	0	0	0	0
2.	y_{ft}	0	0	0	133	0	115	0	177	0	135	0	95	-
3.	$X_{ft,h=0}^{free}$	268	202	135	201	0	115	67	210	0	135	87	176	0
	$X_{ft,h=1}^{free}$	0	0	0	0	153	57	24	1	133	118	86	80	175
	$X_{ft,h=2}^{free}$	0	0	0	0	0	0	0	0	0	0	0	0	78
4.	R	ja	-	ja	-	ja	-	ja	-	ja	-	ja	-	-
5.	q_{ft}	0	-	133	-	115	-	177	-	135	-	95	12	-
6.	d_{ft}^{FIFO}	27	27	27	20	39	33	23	32	6	32	5	2	-
	d_{ft}^{LIFO}	39	40	40	28	57	48	34	46	9	48	7	1	-
7.	$X_{ft,h=0}$	202	135	68	153	0	67	33	133	0	87	81	175	-
	$X_{ft,h=1}$	0	0	0	0	57	24	1	0	118	86	80	78	-
	$X_{ft,h=2}$	0	0	0	0	0	0	0	0	0	0	0	0	-
8.	C_{fk}	606	405	8544	1578	7431	1266	11082	1826	8814	1652	6543	2692	759
	C_{fk}^{RL}	0	0	0	1072	0	1000	0	1248	0	1080	0	920	0

Tabelle 8: Planungsschritte (1.-8.) je Periode der rollierenden Mikro- und Makroperioden-Planung (mit 13 Iterationen von $t=1$ bis $t=13$) unter Verwendung unseres Bestellmengenmodells (Abschnitt 3) im 5. Schritt

unterscheidet sich von der Planung in Mikroperioden hauptsächlich dadurch, dass Filialen zweimal statt einmal täglich beliefert werden. Die Stärke der Mikroperioden-Planung liegt in der Reduzierung des mittleren Lagerbestands, ohne den Servicegrad zu verringern. Die Senkung der mittleren Lagerbestände führt zur Reduzierung von Abfallmengen, Lagerhaltungskosten und Abfallkosten. Die Idee der Mikroperioden-Planung für schnell verderbliche Güter basiert auf [11], wo gezeigt wird, dass der Kundenservicegrad und der Sicherheitsbestand mit Abfallmengen zusammenhängen.

Das Nachhaltigkeitsmanagement, inklusive Abfallre-

duktion, wird für den Lebensmitteleinzelhandel aufgrund des öffentlichen Interesses und des Imagefaktors immer wichtiger. Auch die EU-Kommission strebt an, die Abfälle in Nahrungsketten dauerhaft zu reduzieren [7], weil das Ausmaß der Verschwendung an Umwelt- und Naturressourcen entlang von Nahrungsketten weltweit sehr hoch ist. Im LEH liegen Umsatzverluste aus Bruch und Verderb bei Obst und Gemüse im Durchschnitt bei 5,12% [7]. Daher ist die zweimal tägliche Belieferung praxisrelevant, weil sie nicht nur eine Abfallreduktion, sondern auch finanzielle Vorteile bringen kann.

Der praktische Einsatz der Mikroperiodenpla-

Beispiel-Ergebnisse	Makroperioden-Planung	Mikroper.-Planung
Bestellmenge	111 ME $1/3 \sum_{t=1}^{12} q_{ft}$	222 ME $1/6 \sum_{t=1}^{12} q_{ft}$
Lagerbestand $1/13 \sum_{t=1}^{13} \sum_{h=0}^H X_{fth}$	273 ME	141 ME
Kosten (Fil.+RL) $\sum_{t=1}^{13} (C_{ft} + C_{ft}^{RL})$	59226 GE	58518 GE
Lieferanzahl Zählen falls $y_{ft} > 0$	1 Mal	5 Mal
Abfallmengen $\sum_{t=1}^{13} W_{ft}$	12 ME	0

Tabelle 9: Ergebnisse des Rechenbeispiels von 12 Perioden (3 Tage)

nung (mit einer Wiederbeschaffungszeit von 1-2 Mikroperioden) für Frischware ist jedoch durch die Wiederbeschaffungszeit von 1 Tag [8] (4 Mikroperioden) aus Regionallägern stark eingeschränkt. Es gibt dennoch Möglichkeiten zum Umgehen dieser Einschränkung im LEH, um eine Mikroperiodenplanung verwenden zu können. In weiteren Publikationen wollen wir unseren Lösungsansatz dazu vorstellen.

References

- [1] F. W. Harris. *How Many Parts to Make at Once*. Operations Research, 38(6):947–950, 1990.
- [2] L. Häselbarth und A. Scholl. *Dynamische Bestellmengenplanung für verderbliche Luxusgüter*. Wirtschaftswissenschaftliche Fakultät / Friedrich-Schiller-Universität Jena, 13(ISSN 1611-1311), 2003.
- [3] J. Hertel, J. Zentes und H. Schramm-Klein. *Supply-Chain-Management und Warenwirtschaftssysteme im Handel*, Springer, Berlin, Heidelberg, 2011.
- [4] F. Hofer. *Management der Filiallogistik im Lebensmitteleinzelhandel, Gestaltungsempfehlungen zur Vermeidung von Out-of-Stocks*, Dissertation, Universität St. Gallen, Deutschland, 2009.
- [5] L. Janssen, T. Claus und F. Herrmann. *Beitrag zur Abfallreduktion mittels mathematischer Optimierung und simulationsbasierter Validierung*. In: Simulation in Production und Logistics, Herausgeber: M. Rabe und U. Clausen, Fraunhofer IRB, Stuttgart, S. 177–186, 2015.
- [6] M.A. Law, W.D. Kelton. *Simulation Modeling and Analysis*, McGraw-Hill Higher Education, New York, 2000.
- [7] M. Kranert, G. Hafner, J. Barabosz, H. Schuller, D. Leverenz, A. Kölbig, F. Schneider, S. Lebersorger und S. Scherhauser. *Ermittlung der weggeworfenen Lebensmittelmengen und Vorschläge zur Verminderung der Wegwerfrate bei Lebensmitteln in Deutschland*, <http://bit.ly/1Vlcgoy>, Abruf am 05.04.2016, 2012.
- [8] H. Kuhn und M. Sternbeck. *Logistik im Lebensmittelhandel - Eine empirische Untersuchung zur Ausgestaltung handelsinterner Liefernetzwerke*. Forschungsbericht. Katholische Universität, Ingolstadt-Donau, 2011.
- [9] A. Scholl und L. Häselbarth. *Bestellmengenplanung im zeitlich offenen Entscheidungsfeld Bestellmengenplanung im zeitlich offenen Entscheidungsfeld*. Wirtschaftswissenschaftliche Fakultät / Friedrich-Schiller-Universität Jena, 16(ISSN 1611-1311), 2003.
- [10] H. Tempelmeier. *Bestandsmanagement in Supply Chains*. Books on Demand GmbH, Norderstedt, S. 166ff, 2012.
- [11] K.H. van Donselaar und R.A.C.M. Broekmeulen. *Approximations for the relative outdating of perishable products by combining stochastic modeling, simulation und regression modeling*. International Journal of Production Economics, 140(2), S. 660–669, 2012.
- [12] R.V. Weteling. *Improving Freshness The effects of week patterns und opening terms on outdating at Albert Heijn*, Masterarbeit, Eindhoven University of Technology, Niederlande, 2013.

Autokorrelation: Die große Unbekannte in der Systemplanung und -analyse

Sebastian Rank, Frank Schulze, Thorsten Schmidt
Technische Universität Dresden, Institut für Technische Logistik
sebastian.rank@tu-dresden.de

Autokorrelierte Prozesse können in nahezu jedem System sämtlicher Bereiche festgestellt werden. Diese haben in der Regel negativen Einfluss auf die Systemleistung und verursachen bspw. längere Warteschlangen und Durchlaufzeiten oder erhöhen die Volatilität solcher Kennzahlen. Dies steht im Kontrast zu den Möglichkeiten aktueller Programmiersprachen oder Simulatoren, die lediglich die Erzeugung unkorrelierter Zufallszahlen und damit unkorrelierter Prozesse erlauben. Entsprechende Modelle sind nicht valide und zeigen ein Systemverhalten, was nicht der Realität entspricht. Vor diesem Hintergrund diskutiert der Artikel das Vorkommen und die Wirkung von Autokorrelation in Bediensystemen und stellt mit *JARTA* ein nutzerfreundliches Tool vor, das die Erzeugung autokorrelierter Zufallszahlen erlaubt und gut in vorhandene Simulationsumgebungen integriert werden kann.

1 Einleitung, Motivation, Grundlagen

Ein wesentlicher Anspruch der ereignisdiskreten Simulation besteht im Gewinnen von Erkenntnissen über das dynamische Verhalten des zugrundeliegenden Systems. Hieraus leitet sich unmittelbar die Forderung nach dem Einsatz valider Modelle ab. Neben der Abbildung statischer Parameter wie bspw. dem Layout eines Transportsystems, bestehen große Herausforderungen bei der Modellierung von Prozesscharakteristiken wie z. B. schwankenden Bedienzeiten oder Auftragsparametern. Idealerweise werden hierbei Realdaten eingesetzt. Häufig kommt die Simulation aber bei Neu- oder Umbauprojekten zum Einsatz. Ohne vorhandene Einlastdaten sind dann entsprechende Parameter und Daten zu schätzen sowie passende Verteilungsfunktionen zu wählen.

Dieser Artikel beschäftigt sich zunächst mit der Frage, ob die Festlegung einer statistischen Verteilung, deren Mittelwert und Varianz für eine valide Modellierung dabei ausreichend sind. Hintergrund ist die Vermutung, dass mit der Autokorrelation eine weitere Eigenschaft zu berücksichtigen ist. In diesem Zusammenhang wird ein Generator zur Erzeugung autokorrelier-

ter Zufallszahlen vorgestellt. Dieser ist für den Fachmann *einfach* zu parametrieren und gut in bestehende Simulationsumgebungen zu integrieren.

Der Beitrag ist wie folgt aufgebaut. Die Diskussionen erfolgen allgemein anhand von Bediensystemen, also Modellen, mit einer Bearbeitungsstation und vorgelagertem Warteraum zur Möglichkeit der Bildung von Warteschlangen. Der Schwerpunkt liegt auf intralogistischen Systemen. Nach kurzer Einleitung und wichtigen Definitionen sowie Grundlagen wird anhand einer Literaturanalyse und eigener empirischer Untersuchungen gezeigt, dass autokorrelierte Prozesse in nahezu jedem logistischen System nachgewiesen werden können und dabei erheblichen Einfluss auf dessen Verhalten/Performance haben. Die Erkenntnisse werden Grundlage für die Forderung der Abbildung autokorrelierter Prozessereignisse und des Einsatzes entsprechender Zufallszahlengeneratoren sein. Es werden grundlegende, geeignete Verfahren umrissen und eine Vorzugsvariante diskutiert. Der Beitrag wird mit Zusammenfassung, Fazit und Ausblick schließen.

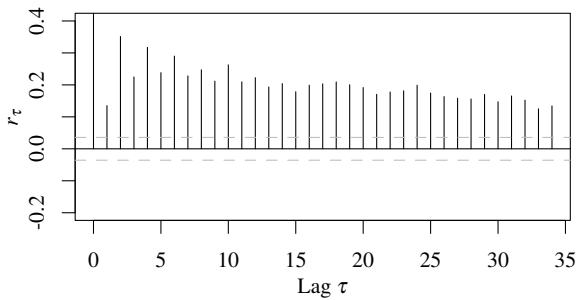


Abbildung 1: Zeitreihe mit positiven Autokorrelationskoeffizienten – Horizontal gestrichelte Linie: Signifikanzband für eine Irrtumswahrscheinlichkeit von 5 %

1.1 Autokorrelation

Nachfolgende Ausführungen sind im Detail in [1] nachzuvollziehen. Danach sind die Autokorrelationskoeffizienten r_τ Kennzahlen, die für jede Zeitreihe X_t errechnet werden können. Autokorrelation beschreibt den linearen Zusammenhang eines Merkmals mit sich selbst, verschoben um einen Zeitversatz (Lag) τ . Übertragen auf logistische Systeme wird also bspw. beurteilt, ob das Auftreten eines Ereignisses x_t das Auftreten eines Folgeereignis x_{t+1} erwarten lässt, also der Prozess selbstähnlich ist. Mathematisch ist Autokorrelation die auf das Intervall $[-1, +1]$ normierte Autokovarianz. Hohe positive Ausprägungen suggerieren statistische Ähnlichkeit, negative Koeffizienten drücken eine statistische Unähnlichkeit aus. Die Bildungsvorschrift lautet

$$r_\tau = \frac{\sum_{t=1}^{N-\tau} (x_t - \bar{x})(x_{t+\tau} - \bar{x})}{\sum_{t=1}^N (x_t - \bar{x})^2} \quad (1)$$

mit $\bar{x} = \sum_i \frac{x_i}{N}$ für $\tau = 0, 1, 2, \dots, N-1$.

Die Koeffizienten mehrerer Lags τ werden in Autokorrelogrammen dargestellt. Abbildung 1 zeigt beispielhaft eine Zeitreihe mit positiven Autokorrelationskoeffizienten r_τ .

Insbesondere bei der Erzeugung autokorrelierter Zeitreihen sind *partielle* Autokorrelationskoeffizienten α_τ von Relevanz. Sie können aus den r_τ abgeleitet werden und geben den Zusammenhang zwischen Beobachtungswerten $(x_t, x_{t+\tau})$ an, ohne die indirekten („vererbte“) Korrelationen von x_u für $t < u < t + \tau$ auf x_t zu berücksichtigen.

Ein autokorrelierter Prozess mit einem Koeffizienten $r_1 > 0$ sei nachfolgend als positiv autokorreliert bezeichnet und analog bei $r_1 < 0$ als negativ autokorreliert.

1.2 Zufallszahlengeneratoren

Die Beschreibung dynamischer und stochastischer Prozesse in Bediensystemen erfolgt üblicherweise durch Zufallszahlen. Deren Erzeugung erfolgt mittels Zufallszahlengeneratoren (engl. auch Random Number Generators, RNG). Umfangreiche Abhandlungen des Themas sind bspw. [4] oder [5] zu entnehmen. Als Quasistandard zur Erzeugung von Zufallszahlen haben sich Kongruenzgeneratoren nach [6] herauskristallisiert. Im einfachsten Fall basieren diese auf einer linear rekurrenten Bildungsvorschrift. Eine Reihe natürlicher Zufallszahlen Z_1, Z_2, \dots im Intervall $0 \leq Z_i \leq m$ ist definiert durch

$$Z_i = (aZ_{i-1} + c) \pmod{m} \quad (2)$$

wobei durch die Festlegung des Multiplikators a , des Startwertes Z_0 , der Verschiebung c und des Moduls m die Güte der Sequenz festgelegt wird. Die Güte zeichnet sich hierbei nach [5] durch u. a. stochastisch unabhängige, identisch verteilte Folgeglieder (auch i.i.d. oder IID – engl. für independent and identically distributed) aus. Die Transformation $U_i = Z_i/m$ erlaubt das Generieren gleichverteilter Zufallszahlen $\mathcal{U}(0, 1)$ aus dem Intervall $[0, 1)$. Diese dienen wiederum als Basis für die Erzeugung von Zufallszahlen mit beliebiger Randverteilung und IID-Eigenschaft [4]. An dieser Stelle wird bereits ein Widerspruch deutlich: Wie angedeutet, sind viele Prozesse durch Autokorrelation gekennzeichnet. Gängige, populäre Software lässt aber nur das Generieren unabhängiger Ereignisse zu. Entsprechende Modelle sind folglich invalide. Wie hoch hierbei die Diskrepanz ist und wie durch Anwendung spezieller Generatoren Abhilfe geschaffen werden kann, beleuchten die nachfolgenden Kapitel.

2 Autokorrelation in der Praxis und ihre Wirkung

Im Rahmen der Simulation intralogistischer Systeme werden Prozesse in der Regel als unkorreliert ange-

nommen. Einfache Überlegungen bzgl. Batchbildung oder Prioritätsregeln bei der Steuerung lassen aber annehmen, dass aufeinanderfolgende Ereignisse nicht zwangsläufig unabhängig sind, sondern ggf. autokorrelieren. Ein einfaches Experiment verdeutlicht die Tragweite dieser Annahme: Modelliert sind drei Bearbeitungsstationen, die sich nur in der Form der Batchbildung im Ankunftsstrom unterscheiden. Andere Parameter wie Randverteilungen oder Auslastungsgrade sind identisch. Tabelle 1 zeigt die Ergebnisse in Form durchschnittlicher Warteschlangenlängen.

	Grad der Batchbildung		
	keine	mäßig	ausgeprägt
Ø Warteschlange	8	25	68

Tabelle 1: Durchschnittliche Warteschlangenlänge bei unterschiedlicher Batchbildung – Untersuchte Kennzahl: Warteschlangenlänge vor einer M|M|1-Bedienstation mit einem Auslastungsgrad von 0.9

In Tabelle 1 ist zu erkennen, dass sich bei größeren Batches infolge der zunehmenden positiven Autokorrelation erheblich längere Warteschlangen ausbilden. Ein Modell ohne Beachtung dieses Effektes hätte zu falschen Simulationsergebnissen sowie ggf. einer nicht korrekt ausgelegten Anlage und erheblichen Fehlerfolgekosten geführt. Eine Reihe weiterer Autoren kommen zu gleichen oder ähnlichen Ergebnissen – eine umfangreiche Literaturübersicht wird in [11] gegeben.

Diese Überlegungen sind nicht rein theoretischer Natur. Durch die Autoren dieses Beitrages wurde eine umfangreiche empirische Studie durchgeführt. Hierbei wurden in Zusammenarbeit mit sechs Anlagenbetreibern aus der Intralogistikbranche insgesamt 68 Prozesse bzw. zugehörige Datensätze auf Autokorrelation geprüft wobei diese in 64 Fällen signifikant nachgewiesen werden konnte. Im direkten Vergleich mit einer hypothetischen Falschannahme unkorrelierter Prozesse zeigen sich jeweils erhebliche Unterschiede in den Simulationsergebnissen. Teilweise können Abweichungen der untersuchten Warteschlange von mehreren hundert Prozent festgestellt werden – Tabelle 2 zeigt auszugsweise die Analyseergebnisse drei repräsentativer Datensätze. Noch einmal sei betont, dass der Fehler bzw. Abweichung der Ergebnisse kausal auf Autokorrelation zurückge-

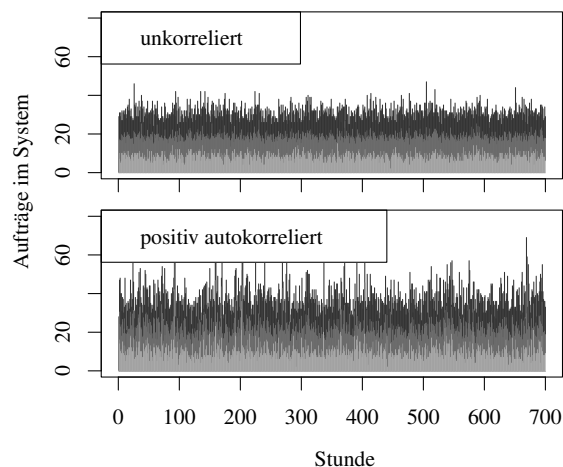


Abbildung 2: Im Gesamtsystem befindliche Aufträge stündlich aufgelöst; hellgrau: minimale Anzahl, dunkelgrau: mittlere Anzahl, schwarz: maximale Anzahl

führt werden kann. Alle anderen Modellparameter wie bspw. die Form der Randverteilung des Ankunftsprozesses sowie deren Mittelwert und Varianz wurden als identisch angenommen.

Daten-satz	r_1	r_2	r_3	Fehler durch Annahme unkorrelierter Prozesse
I	0.23	0.15	0.12	+40 %
II	0.46	0.42	0.34	+1950 %
III	-0.23	-0.20	-0.22	-80 %

Tabelle 2: Empirische Studie zum Auftreten und Wirken von Autokorrelation – r_τ : Autokorrelationskoeffizient zum Lag τ ; Untersuchte Kennzahl: Warteschlangenlänge vor einer M|M|1-Bedienstation mit einem Auslastungsgrad von 0.9

Bei näherer Untersuchung zweier Kommissionierstationen mit angebundenem, automatischem Lager wird ferner deutlich, dass autokorrelierte Prozesse zu einem im Vergleich deutlichen Anstieg der Schwankung von Systemkennzahlen wie bspw. Durchsatz oder Auslastungsgrad führen. Dies wird in Abbildung 2 deutlich. Dargestellt sind in stündlicher Auflösung Zeitreihen über die im System befindlichen Aufträge – also im übertragenem Sinne der Füllgrad des Systems.

In Konsequenz wurde trotz identischem Durchsatz bzw. Auslastungsgrad des Systems ein Anstieg der mittleren Durchlaufzeit von bis zu 45 % und eine um

250 % erhöhte Wahrscheinlichkeit überfüllter Puffer beobachtet. Es bleibt festzuhalten, dass Modelle mit nicht autokorrelierten Ereignissen das sich in Realität einstellende Systemverhalten deutlich falsch dargestellt hätten. Relevant sind hierbei vor allem positiv autokorrelierte Prozesse. Treten diese auf, werden planungs- und auslegungsrelevante Kennzahlen regelmäßig erheblich unterschätzt. Dies gilt umgekehrt, in deutlich abgeschwächter Form, für negativ autokorrelierte Prozesse. In Konsequenz müssen spezielle Zufallszahlengeneratoren zum Einsatz kommen. Eine Auswahl geeigneter Verfahren wird im nachfolgenden Kapitel vorgestellt und diskutiert.

3 Erzeugung autokorrelierter Zufallszahlen

Zur Abbildung autokorrelierter Prozesse werden in der Literatur am häufigsten folgende Verfahren diskutiert und angewendet:

- *Markovsche Ankunftsprozesse* induzieren Korrelationen zwischen Zufallszahlen durch Festlegung von Zuständen und Übergangsraten einer zugrundeliegenden Markov-Kette [3].
- *Copulas* beschreiben Abhängigkeiten zwischen univariaten Randverteilungen in einer gemeinsamen Randverteilung [10].
- Distorsionsansätze wie bspw. *Transform-Expand-Sample* oder *Minification* und *Maxification* erzeugen aus unkorrelierten Zufallszahlen durch spezielle Bildungsvorschriften autokorrelierte Werte [9, 7].
- *Autoregressive Modelle* basieren auf dem Prinzip der Beschreibung von Zufallszahlen jeweils aus deren Vorgängern [1].
- Speziell zur Erzeugung autokorrelierter Werte mit beliebiger, aber invertierbarer Randverteilung bieten sich der sog. *ARTA-Ansatz* (AutoRegressive-To-Anything) an [2].

Weil der ARTA-Ansatz Vorzüge gegenüber den vorgenannten bietet [14], sei dessen Prinzip nachfolgend kurz erläutert.

Grundlage bildet ein AR-Basisprozess Z_t mit den partiellen Autokorrelationskoeffizienten α_h und dem Störterm ε_t :

$$Z_t = \sum_{h=1}^p \alpha_h Z_{t-h} + \varepsilon_t \quad \text{mit} \quad t \in \mathbb{Z} \quad (3)$$

Um einen ARTA-Prozess Y_t , also Zufallszahlen mit einer gewünschten Randverteilung und Autokorrelationsstruktur, zu erhalten, ist die Transformation $F_Y^{-1}[\Phi(Z_t)]$ anzuwenden (Φ entspricht der Verteilungsfunktion der Standardnormalverteilung) und sicherzustellen, dass Z_t einer Standardnormalverteilung $\mathcal{N}(0, 1)$ genügt. Dies wird über den Störterm ε_t erreicht. Dieser sei normalverteilt $\mathcal{N}(0, \sigma^2)$ und die Gleichung

$$\sigma^2 = 1 - \sum_{h=1}^p \alpha_h r_h \quad (4)$$

erfüllt. Hierbei gibt r_h den Autokorrelationskoeffizienten zum Lag h , $h = 1, 2, \dots, p$ des AR-Prozesses an und ist mit $r_h = \text{Corr}[Z_t, Z_{t+h}]$ beschrieben. Bis auf den Fall einer Gleichverteilung [8] besteht keine geschlossene Lösung zur Bestimmung von r_h . Ein automatisches, numerisches Suchverfahren wird aber von [2] gegeben. Die verbleibende Bestimmung der α_h erfolgt bspw. über ein Yule-Walker-Gleichungssystem [1].

4 Autokorrelierte Zufallszahlen in der ereignisdiskreten Simulation

Im vorangegangenen Abschnitt 3 werden verschiedene Verfahren zur Erzeugung autokorrelierter Zufallszahlen erwähnt. Im Kontext der Abbildung autokorrelierter Prozesse wird der Einsatz des ARTA-Ansatzes empfohlen. Dies begründet sich vor allem wegen seiner Flexibilität zur Erzeugung von Zufallszahlen mit beliebiger (invertierbarer) Randverteilung und Autokorrelationsfunktion sowie der einfachen Anwendung. Demgegenüber erweisen sich die restlichen Verfahren meist als aufwendig und kompliziert in der Parametrierung bzw. im Falle von Markovschen Ankunftsprozessen eher geeignet zur analytischen Betrachtung von Bedienmodellen – eine ausführliche Diskussion kann in [14] nachvollzogen werden.

Konkret wird zur Einbindung in eine Simulationsumgebung der Einsatz des Zufallszahlengenerators *JARTA* empfohlen. Dieser wurde von den Autoren dieses Beitrages mitentwickelt, ist in JAVA geschrieben und setzt den erwähnten ARTA-Ansatz um [13]. *JARTA* kann direkt als JAVA-Bibliothek in einen Simulator integriert (bspw. in AnyLogic) bzw. über Schnittstellen angesprochen werden. Alternativ lässt es sich als Kommandozeilenprogramm ausführen, womit sich ein Datenaustausch über Text-Files anbietet.

Die Vorteilhaftigkeit des ARTA-Ansatzes bzw. von *JARTA* wurde in einer Vergleichsstudie mit dem Ziel der möglichst einfachen und exakten Modellierung autokorrelierter, empirischer Daten bzw. vorgegebener Verteilungsparameter hergeleitet. Beispielfähig zeigt Abbildung 3 das Histogramm sowie Autokorrelogramm des in Tabelle 2 bereits erwähnten Realdatensatzes II. Abbildung 4 zeigt beide Grafiken für eine Zeitreihe generiert mit *JARTA* wobei die statistischen Eigenschaften des Datensatzes II als Grundlage dienen. Die unterschiedliche Form der Autokorrelogramme bei Koeffizienten hoher Lags ist aus praktischer Sicht zu vernachlässigen, denn den größten Einfluss auf die Ergebnisgenauigkeit von Simulationsstudien hat die exakte Modellierung niedriger Lags [11]. Diese stimmen für beide Autokorrelogramme nahezu exakt überein.

Die Genauigkeit der Simulationsergebnisse wird mit Verwendung der autokorrelierten Zufallszahlen zur Beschreibung des Ankunftsprozesses immens erhöht. Wurde im einleitenden Beispiel die Warteschlange vor der Bedienstation noch um 1950 % unterschätzt (vgl. Tabelle 2), weil fälschlicherweise unkorrelierte Zwischenankunftszeiten angenommen werden, weichen die Ergebnisse im angepassten Modell nur noch um 5 % ab.

5 Zusammenfassung, Ausblick

Zusammenfassend muss konstatiert werden, dass die Parametrierung von Simulationsmodellen mit unabhängigen Ereignissen mit hoher Wahrscheinlichkeit zu falschen Modellen führt. Grundlage dieser Einschätzung bildet zum einen die zitierte Studie zur Untersuchung des Auftretens von Autokorrelation in intralogistischen Systemen und die dabei festgestell-

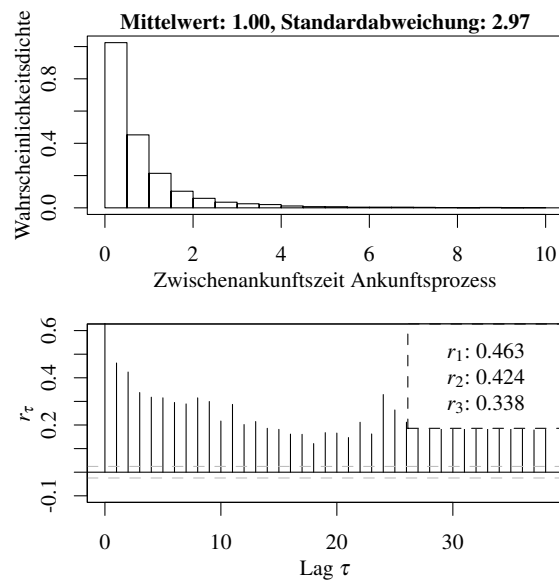


Abbildung 3: Kennzahlen empirischer Daten – oben: Histogramm, unten: Autokorrelogramm

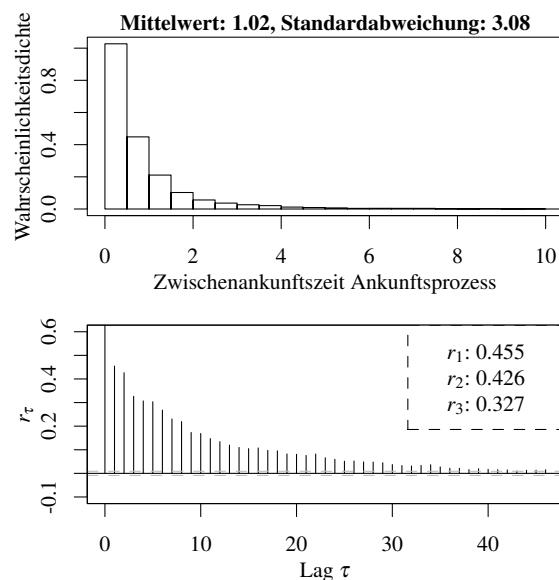


Abbildung 4: Kennzahlen zu Zufallszahlen erzeugt mit *JARTA* – oben: Histogramm, unten: Autokorrelogramm

ten Autokorrelationen in nahezu allen Datensätzen. Zum anderen ist dabei im Vergleich zu unkorrelierten Modellen mit Abweichungen zu rechnen, die nicht als (vernachlässigbare) Modellungenauigkeiten abgetan werden können.

In Konsequenz sollten bei der Simulation und Planung von Materialflusssystemen Zufallszahlengeneratoren zum Einsatz kommen, welche die Modellierung autokorrelierter Ereignisse zulassen. Auf wissenschaftlicher Ebene werden eine Reihe von Ansätzen zur Erzeugung abhängiger Zufallszahlen diskutiert. Im vorliegenden Fall wird die Verwendung des sog. AutoRegressive-To-Anything-Ansatzes vorgeschlagen. Dieser hat sich als flexibel, anwenderfreundlich und performant erwiesen. Eine JAVA Implementierung, sog. „JARTA“, zur Modellierung und Anpassung von ARTA-Prozessen wird als Download zur freien Verfügung gestellt [12]. Mit JARTA und den vorgestellten Ergebnissen ist nunmehr eine Basis gegeben, zukünftige Simulationsstudien mit einer höheren Genauigkeit durchzuführen, indem der Effekt von autokorrelierten Ereignissen berücksichtigt wird. Abweichungen bei der Anlagenplanung können damit nachhaltig minimiert werden.

Literatur

- [1] BOX, G. E. P. ; JENKINS, G. M. ; REINSEL, G. C.: *Time series analysis: forecasting and control*. Fourth Edition. Hoboken : Wiley, 2008
- [2] CARIO, M. C. ; NELSON, B. L.: Autoregressive to anything: Time-series input processes for simulation. In: *Operations Research Letters* 19 (1996), Nr. 2, S. 51–58
- [3] KRIEGE, J.: *Fitting simulation input models for correlated traffic data*. Dortmund, Technische Universität Dortmund, Dissertation, 2012
- [4] LAW, A. M. ; KELTON, D. W.: *Simulation Modeling and Analysis*. Third Edition. Singapore : McGraw-Hill Professional, 2000
- [5] L'ECUYER, P.: Uniform random number generation. In: *Annals of Operations Research* 53 (1994), Nr. 1
- [6] LEHMER, D. H.: Mathematical Methods in Large-Scale Computing Units. In: *The Annals of Computation Laboratory of Harvard University* 26 (1951), S. 141–146
- [7] LEWIS, P. A. W. ; MCKENZIE, E.: Minification processes and their transformations. In: *Journal of applied probability* 28 (1991), Nr. 1, S. 45–57
- [8] LI, S. T. ; HAMMOND, J. L.: Generation of Pseudorandom Numbers with Specified Univariate Distributions and Correlation Coefficients. In: *IEEE Transactions on Systems, Man and Cybernetics* 5 (1975), Nr. 5, S. 557–561
- [9] MELAMED, B.: TES: A Class of Methods for Generating Autocorrelated Uniform Variates. In: *ORSA Journal on Computing* 3 (1991), Nr. 4, S. 317–329
- [10] NELSEN, R. B.: *An Introduction to Copulas*. Second Edition. New York : Springer, 2006
- [11] SCHMIDT, T. ; ROSE, O. ; RANK, S. ; UHLIG, T.: *Beherrschung stark korrelierter Logistik- und Produktions-Prozesse*. Dresden, München, Technische Universität Dresden, Universität der Bundeswehr München, Schlussbericht zu dem IGF-Vorhaben 17344BR, 2013
- [12] UHLIG, T. ; RANK, S.: *A Java library to model and fit ARTA processes*. <http://sourceforge.net/projects/jarta/>. Version: 2013
- [13] UHLIG, T. ; RANK, S. ; ROSE, O.: JARTA - a java library to model and fit autoregressive-to-anything processes. In: PASUPATHY, R. (Hrsg.) ; KIM, S.-H. (Hrsg.) ; TOLK, A. (Hrsg.) ; HILL, R. (Hrsg.) ; KUHL, M. E. (Hrsg.): *Proceedings of the 2013 Winter Simulation Conference*. Washington DC : Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc., 2013, S. 1203–1211
- [14] UHLIG, T. ; RANK, S. ; ROSE, O.: Evaluation Of Modeling Tools For Autocorrelated Input Processes. In: ROEDER, T. M. K. (Hrsg.) ; FRAZIER, P. I. (Hrsg.) ; SZECHTMAN, R. (Hrsg.) ; ZHOU, E. (Hrsg.) ; HUSCHKA, T. (Hrsg.) ; CHICK, S. E. (Hrsg.): *Proceedings of the 2016 Winter Simulation Conference*. Washington DC. Im Druck : Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc., 2016

Simulation sozioökonomischer Prozesse mit zellulären Automaten - räumliche Analyse des Kundenverhaltens konkurrierender Apotheken

Marcel Stasik¹, Prof. Dr. rer. nat. Anja Voß-Böhme², Prof. Dr.-Ing. Thomas Wiedemann³,

Institution: Hochschule für Technik und Wirtschaft Dresden

¹ s68494@htw-dresden.de

² anja.voss-boehme@htw-dresden.de

³ wiedem@informatik.htw-dresden.de

Mathematische Modellierung und Simulation stellen zunehmend wichtige Hilfsmittel bei der Steuerung komplexer sozioökonomischer Prozesse dar. Beispielsweise ist die Zahl der Kunden eines Unternehmens abhängig von der Qualität der Produkte und der Preisgestaltung im Vergleich zu Konkurrenten, aber auch vom Aufwand des Kunden, an das Produkt des Unternehmens zu gelangen. Im Falle von Apotheken, die in dieser Arbeit betrachtet werden, sind Unterschiede in der Qualität der Produkte zu vernachlässigen. Dafür sind topografische Einflüsse relevant, da die Lage der Apotheke im Stadtgebiet und die Laufentfernung zur Apotheke darüber entscheiden, wie schwierig es für den Kunden ist, bei der Apotheke einzukaufen. Soziale Netzwerke, beispielsweise räumliche Nachbarschaftsbeziehungen, spielen ebenfalls eine Rolle für das Kaufverhalten, indem sich ein potentieller Kunde am Kaufverhalten seiner Nachbarn orientiert. In der vorliegenden Arbeit wird untersucht, inwiefern der Einfluss dieser beiden topografischen Aspekte, sowie der Effekt zusätzlicher unternehmerischen Maßnahmen der einzelnen Apotheken, auf das Kundenverhalten mit Hilfe zellulärer Automaten beschrieben und analysiert werden können. Dabei stehen besonders Untersuchungen zur Umsetzbarkeit und Modellierungseffizienz der Aufgabenstellung im Vordergrund.

1 Einleitung

-Ein Baum, der nicht mehr wächst, ist tot. - [unbekannt]

Grundlage für langfristig erfolgreiches unternehmerisches Handeln ist Wachstum. Die Steuerung eines Unternehmens so, dass der Kundenstamm auch in Konkurrenzsituationen nachhaltig wächst, ist eine komplexe und ressourcenverschlingende Aufgabe. Mit Hilfe realitätsnaher Simulationsmodelle können verschiedene Szenarien auf dem Rechner analysiert und getestet werden. Dies erlaubt es, den Einfluss verschiedener Einflussgrößen wesentlich systematischer abzuschätzen, als es ohne mathematisches Modellierung möglich wäre [1,2].

In dieser Arbeit soll die Entwicklung der Kundenzahlen von verschiedenen, konkurrierenden Apotheken untersucht werden. Ziel des Modells ist es, die zeitliche Dynamik der räumlichen Einzugsgebiete von Apotheken zu beschreiben, so dass die Auswirkungen folgender Faktoren analysiert werden können:

- topographische Einflüsse und natürliche Hindernisse, die über die Laufentfernung zu den Apotheken die Wahl der Apotheke beeinflussen;

- Einfluss von Marketing-Maßnahmen, wie Rabatt-Aktionen und Treue-Boni-Programme, sowie der Gestaltung der Öffnungszeiten in den einzelnen Apotheken auf die Kundenzahl;
- Rückkopplung der Kundenzahl auf die Preisgestaltung, indem größere Kundenzahlen effektiveres Wirtschaften und damit geringere Preise erlauben;
- Einfluss der Organisationsform - Einzelfiliale oder Mitglied einer Apothekenkette - über die Preisgestaltung auf die Kundenzahl.

Zelluläre Automaten sind performante Systeme, die es erlauben, viele gleichartige, miteinander wechselwirkende und räumlich angeordnete Komponenten zu beschreiben und das entstehende Gesamtverhalten zu analysieren [3]. Daher soll hier am Beispiel der Kundenzahlentwicklung bei konkurrierenden Apotheken untersucht werden, inwiefern sozioökonomische Wachstumsprozesse mit zellulären Automaten effektiv und zielführend modelliert, simuliert und analysiert werden können.

Kontinuierliche Systeme, beispielsweise Multi-Agentensysteme oder partielle Differentialgleichungssysteme sind für die betrachtete Fragestellung weniger geeignet. Bei Agentensystemen wird für jede aktiv zu betrachtende Komponente des Problems (Kunden, Apotheken, usw.) ein eigener Agent eingerichtet, dessen Zustand dynamisch aktualisiert wird, in der Regel über die Berechnung einer Differentialgleichung. Das entstehende Differentialgleichungssystem führt bei realistischen Kundenzahlen schnell zu einem sehr hohen Rechenaufwand. Zwar erlauben Agentensysteme die Einbeziehung individueller Kundenparameter, jedoch sind diese Daten in der Praxis oft gar nicht verfügbar, so dass stark vereinfachte Systeme bereits ausreichen, um erste Aussagen zu treffen. Partielle Differentialgleichungssysteme würden für die numerische Simulation ebenfalls wieder diskretisiert werden müssen, so dass es günstiger erscheint, gleich mit einer diskreten Modellierung, den zellulären Automaten, zu beginnen.

Etablierte Softwaresysteme zur Simulation ökonomischer Prozesse, wie zum Beispiel Taylor Enterprise Dynamics, sind auf die Analyse und Optimierung von Produktions-, Transport- und Logistikketten ausgelegt und betrachten keine räumlichen Aspekte. Die räumliche Vernetzung der Kunden ist jedoch für das betrachtete Problem essentiell.

Unter Beachtung des Rechenaufwands, des Konfigurationsaufwands und auch des Beherrschens aller notwendigen Aspekte erscheinen zelluläre Automaten daher erfolgsversprechend. Im Zuge der Analyse wird ebenfalls erprobt, wie zugänglich aktuell frei verfügbare Simulationssysteme sind und ob sich diese für die Simulation von sozioökonomischen Prozessen eignen. Zur Nutzung kommt die Simulationssoftware Morpheus der Technischen Universität Dresden, die auf einem speziellen Typ zellulärer Automaten, dem „Cellular Potts Model“ aufbaut, und bisher hauptsächlich für die Simulation von biologischem Zellverhalten genutzt wird (vgl. [7] und [8]).

2 Das mathematische Modell

Es wird zunächst ein stark vereinfachtes Modell entwickelt, bei dem eine homogene Kundengruppe betrachtet wird. Alle simulierten Kunden sind gleich alt und haben die gleiche Lebenssituation. Zusätzlich wird ein einheitlicher Bedarf an Gütern der Apotheken zugrunde gelegt. Ebenso wird keine Apotheke bevorzugt oder benachteiligt, alle starten mit den gleichen Bedingungen.

2.1 Zustände

Die räumlichen Beziehungen in einem zellulären Automaten werden innerhalb eines zweidimensionalen Spielfeldes S beschrieben, welches durch Diskretisierung in eine reguläre Gitterstruktur mit gleichartigen Zellen $x = (x_1, x_2)$ zerlegt ist. Jeder Zelle wird ein Zustand $\eta(x)$ aus einer endlichen Zustandsmenge Q zugeordnet, welcher entweder das Kaufverhalten eines an diesem Ort wohnenden Kunden oder eine topografische Besonderheit repräsentiert. Zudem sind an jedem Ort die Entfernungen zu den einzelnen konkurrierenden Apotheken als skalare Felder hinterlegt. Sie bilden einen Einflussfaktor für die Kaufentscheidung des Einzelkunden.

Zustandsgrößen des Modells: $Q = \{1, \dots, 7\}$, wobei

- 1 Apotheke 1
- 2 Apotheke 2
- 3 Apotheke 3
- 4 Kunde, kauft bei Apotheke 1
- 5 Kunde, kauft bei Apotheke 2
- 6 Kunde, kauft bei Apotheke 3
- 7 Sonst (keine Apotheke und kein Kunde)

und

$n=7 \dots$ Gesamtanzahl an möglichen Zuständen.

Die Spielfläche wird durch ein rechtshändiges kartesisches Koordinatensystem, sprich dem 1. Quadranten eines kartesischen Koordinatensystems, beschrieben.

n_{ZXS} = Anzahl Zellen in x-Richtung
 n_{ZXY} = Anzahl Zellen in y-Richtung
 $F_{GS} = n_{ZXS} n_{ZXY}$ Gesamtfläche des Spielfeldes

Der aktuelle Zustand des zellulären Automaten zum Zeitpunkt t wird beschrieben durch die Matrix, oder Konfiguration,

$$\eta_t = (\eta_t(x_1, x_2))_{(x_1, x_2) \in S},$$

wobei $\eta_t(x_1, x_2) \in Q$ den Zustand der Zelle $x = (x_1, x_2) \in S$ zum Zeitpunkt t beschreibt.

Um die Entfernungen zwischen den Apotheken und den Kunden abzubilden, wird jeder Zelle auf dem Spielfeld ein Vektor der Entfernungswerte zu den einzelnen Apotheken zugeordnet.

Grundlage für die Entfernungsberechnung ist die gängige Formel zur Ermittlung von Abständen in zweidimensionalen Ebenen:

d = Entfernung zwischen zwei Punkten in einer zweidimensionalen Ebene

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Diese Werte werden zu Beginn der Simulation berechnet und während der Simulation nicht mehr verändert.

2.2 Zeitliche Dynamik

Die Berechnung der Zustandsänderungen pro Zeitschritt erfolgt nach den Regeln eines zellulären Pottsmodells („Cellular Potts Model“ (CPM), vgl. z.B. [5]). Dabei wird für eine zufällig gewählte Zelle die Wahrscheinlichkeit für eine mögliche Zustandsänderung mit Hilfe einer Hamiltonfunktion H berechnet. Diese Hamiltonfunktion beschreibt die „Energie“ einer Konfiguration. Sie kann verstanden werden als Maß dafür, inwiefern sich die einzelnen Zellzustände in Konflikt miteinander und mit den globalen Regeln des Systems befinden. Die Hamilton-Funktion besteht aus unterschiedlichen Termen, die genutzt werden um einzelne Aspekte des Verhaltens einer Zelle individuell einzubeziehen:

$$H = H_I + H_V + H_1 + \dots + H_m.$$

Dabei sind der Interaktionsterm H_I und der Volumen-Kontrollterm H_V in der Regel immer vorhanden und die Terme H_1, \dots, H_m werden problemspezifisch hinzugefügt (siehe unten). Generell ist die Wahrscheinlichkeit für eine Zustandsänderung abhängig vom Energiegewinn ΔH . Dieser ergibt und berechnet sich nach Festlegung zweier Hilfsparameter $T > 0$, der „Temperatur“ und $h \neq 0$, der Transitionsschwelle, über folgenden Algorithmus (modifizierten Metropolis Algorithmus):

1. Starte mit der Ausgangskonfiguration η
2. Wähle eine Zelle $x \in S$ zufällig, mit gleichmäßiger Verteilung
3. Wähle einen zufälligen Nachbarn y , mit gleichmäßiger Verteilung, aus allen Gitternachbarn von Zelle x und bilde die neue Konfiguration η_x^y , bei der die Zelle x den Zustand von Zelle y übernimmt.
4. Berechne den Energiegewinn des Kopierversuches mit Hilfe der Hamilton-Funktion

$$\Delta H = \Delta H_x^y := H(\eta_x^y) - H(\eta)$$

5. Ist der Energiegewinn kleiner als die Schwelle ($\Delta H_x^y < h$), wird der Kopierversuch angenommen, die Versuchskonfiguration als neue Ausgangskonfiguration übernommen ($\eta := \eta_x^y$) und mit Schritt 1 erneut begonnen. Ist der Energiegewinn größer als die Schwelle ($\Delta H_x^y > h$) wird die Versuchskonfiguration mit Wahrscheinlichkeit

$$P(\eta \rightarrow \eta_x^y) = e^{-\frac{(\Delta H_x^y - h)}{T}}$$

übernommen. Je nach Ergebnis der Berechnung wird die Versuchskonfiguration übernommen oder verworfen und mit Schritt 1 erneut begonnen.

In Schritt 3. des Algorithmus kommt die Interaktion der Kunden untereinander zum Ausdruck. Ein Kunde wählt einen seiner Nachbarn aus und übernimmt dessen Kaufverhalten mit einer gewissen Wahrscheinlichkeit, die von seinem „Gewinn“, modelliert in der Hamiltonfunktion H (siehe unten), abhängt. Im zellulären Automaten wird für die Beschreibung der Nachbarschaftsbeziehungen die Moore-Nachbarschaft [3] verwendet. Haben alle Nachbarn das gleiche Kaufverhalten, so ändert sich das Kaufverhalten bei einer versuchten Zustandsänderung nicht. Tatsächliche Interaktion findet deshalb nur an den Rändern der Einzugsbereiche/Zelltypen statt, siehe Abb. 2-1. Weitere Interaktionen zwischen den Kunden werden nicht betrachtet, d.h. der Interaktionsterm H_I wird hier gleich Null gesetzt.

Das Bestreben der Apotheken immer neue Kunden zu gewinnen wird innerhalb des CPM über den Volumen-Kontrollterm definiert. Jede Apotheke ist bestrebt, alle potentiellen Kunden des Spielfeldes für sich zu gewinnen, das heißt die Zielgröße für das Volumen des Einzugsgebietes jeder Apotheke ist gleich F_{GS} , der Größe des Spielfeldes. Für jedes Einzugsgebiet wird das aktuelle Volumen berechnet,

$$V_w(\eta) = \sum_{x \in S} \delta(w, \eta(x)),$$

wobei $w=1,2,3$ und δ das Kronecker-Symbol ist, definiert durch

$$\delta(u, v) = \begin{cases} 1, & \text{für } u = v \\ 0, & \text{sonst.} \end{cases}$$

Dieses aktuelle Volumen wird verglichen mit dem Zielvolumen und die quadratische Differenz gewichtet durch einen Kontroll-Parameter λ_V . Dies ergibt

$$H_V(\eta) := \sum_{w=1,2,3} \lambda_V (F_{GS} - V_w(\eta))^2$$

Daneben sollen die Entfernungen zu den Apotheken in der Hamilton-Funktion berücksichtigt werden. Dazu wird die Entfernung der aktuellen Konfiguration, d_x , mit der Entfernung des Updateversuches, d_y , verglichen. Um das etwas zu veranschaulichen wird dies an einem Beispiel etwas näher erklärt.

Eine Zelle x sei aktuell der Apotheke 1 zugeordnet, also $\eta(x) = 1$. Nun soll in einem Updateversuch geprüft werden ob diese Zelle nicht besser der Apotheke 2 zugeordnet werden soll. Somit wird die Entfernung zur Apotheke 1 als d_x und die Entfernung zur Apotheke 2 als d_y gespeichert. Als nächstes werden die beiden Werte verglichen:

$d_y - d_x > 0$ ist eine Verkürzung des Weges (vorteilhaft),

$d_y - d_x < 0$ ist eine Vergrößerung des Weges (nachteilig).

Dieser Vergleich wird nun an die bestehende Hamilton-Funktion als neuer, zusätzlicher Teiloperator angefügt. Zur Energiedifferenz ΔH_V wird also ein Entfernungsterm hinzugefügt:

$$\Delta H_x^y := \Delta H_V - \lambda(d_y - d_x)$$

wobei λ ein Parameter ist, der die Sensitivität der Kunden bezüglich Wegveränderung beschreibt.

In einem Monte-Carlo-Schritt (MCS) werden unabhängig voneinander so viele Zustandsänderungen nach obigen Algorithmus versucht, wie es Zellen im Spielfeld S gibt. Die gesamte Simulation wird über 500 MCS betrachtet, damit nach einer Anlaufphase ein eingeschwungener und aussagefähiger Zustand erreicht wird.

Abbildung 2-1 stellt die Interaktionen in einem CPM dar. Interaktionen finden nur an den Grenzbereichen der Zelltypen statt. Ein Zelltyp ist definiert über einen eindeutigen Index und somit ergeben sich nur Wechselwirkungen zwischen Zellen mit einem unterschiedlichen Index. Die Reaktionsstärke wird über J ausgedrückt und die Fußnoten repräsentieren welche Zelltypen in Reaktion stehen.

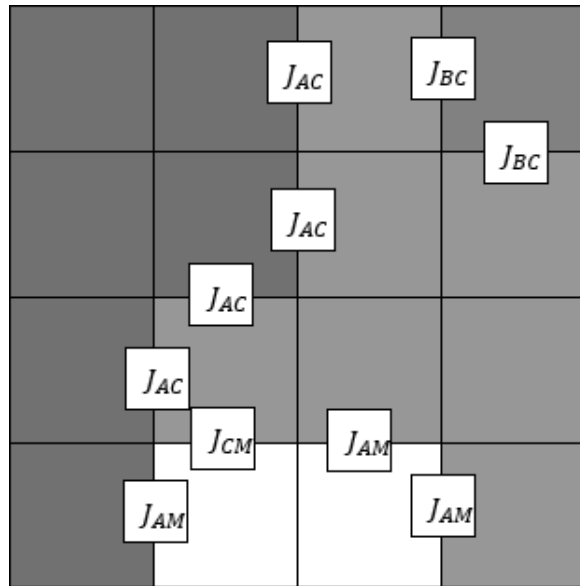


Abbildung 2-1 Interaktion zwischen Zelltypen

Somit wird die Reaktion zwischen Zelltyp A und C über J_{AC} ausgedrückt, zwischen Zelltyp B und C über J_{BC} , zwischen Zelltyp A und B über J_{AB} . Zusätzlich kann auch die Reaktionsstärke der Zelltypen mit dem Medium beeinflusst werden, die Notation bleibt identisch. J_{AM} definiert die Stärke der Reaktion zwischen Zelltyp A und dem Medium M.

2.3 Anfangs- und Randbedingungen

Die Simulation wird mit einem leeren Spielfeld begonnen auf dem drei Apotheken fest platziert werden. Zusätzlich werden die Randbedingungen auf abgeschlossen gesetzt, d.h. Wege sind nur durch das Spielfeld möglich und nicht, wie bei periodischem Rand, von einem Rand zum anderen. In der Mitte des Spielfeldes wird ein Hindernis platziert. Dieses Hindernis soll einen Fluss repräsentieren, der nur in der Mitte über eine Brücke überquert werden kann. Somit wird die Entfernung der Kunden zu den Apotheken, je nach Standort der Apotheke und des Kunden, zusätzlich vergrößert.

3 Modellierung mit Morpheus

In dem nachfolgenden Abschnitt wird erklärt, wie das Modell in Morpheus umzusetzen ist. Wie aus dem Handbuch der Software zu entnehmen ist, basiert die Software auf einem einfachen Prinzip. Es existieren neun Hauptbestandteile, sogenannte Container in die wiederum Funktions-Container (Plug-Ins) eingefügt werden können [6].

3.1 Das Basismodell

Durch den Container CPM werden dem Modell die Eigenschaften eines Cellular Potts Models verliehen. Das Plug-In **MCSDuration** skaliert die Simulationszeit mit Monte-Carlo-Schritten.

Abbildung 3-1 visualisiert die drei skalaren Felder welche die Entfernungen der Kunden zu den Apotheken beinhalten. Unter a) ist die Entfernung zur Apotheke 1 visualisiert unter b) zur Apotheke 2 und unter c) zur Apotheke 3.

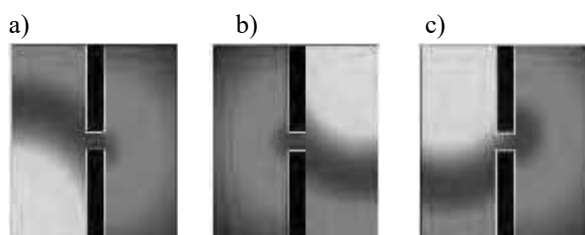


Abbildung 3-1 Visualisierung der skalaren Felder welche die Entfernung der Kunden zu den Apotheken repräsentieren

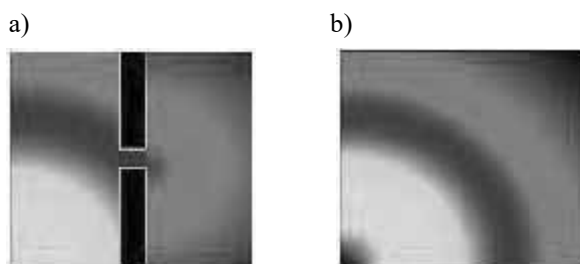


Abbildung 3-2 Vergleich der Entfernungen von Apotheke 1 zu den Kunden mit Hindernis und ohne Hindernis (Luftlinie)

Abbildung 3-2 vergleicht die Entfernungen der Kunden zu Apotheke 1, einmal mit einem natürlichen Hindernis a) und einmal ohne natürliches Hindernis b). Dabei ist deutlich zu erkennen wie das Hindernis die Entfernungen verlängert.

Das Plug-In Logger wertet während der Simulation Daten aus und schreibt diese in eine Textdatei, oder erzeugt mit den Daten ein Diagramm. Dieses Plug-In ermöglicht die Auswertung der Kundenanzahl welche bei einer Apotheke kaufen.

Die genaue Konfiguration der graphische Visualisierung erfolgt mit dem Plug-In Plots, was in das Plug-In Logger eingefügt wird. Dieses ermöglicht es, sehr anschauliche Diagramme in einstellbaren Zeitabständen über die gesamte Simulationszeit zu erstellen (vgl. Abb. 3-1, 3-2, 4-1 bis 4-4).

3.2 Das erweiterte Modell GIS-Daten

In einer zukünftigen Entwicklung des Simulationsmodells soll eine reale Stadtkarte importiert werden und die Entwicklung der Kundenanzahl auf dieser Stadtkarte simuliert werden.



Abbildung 3-3 Beispielstadtkarte für die Weiterentwicklung des Simulationsmodells

Aktuell unterstützt Morpheus den Import einer Dual-Ton TIFF-Datei. Dies bedeutet, die zu importierende Datei darf nur zwei Farbtöne aufweisen, schwarz und weiß. Ist diese Datei importiert, dient diese als neue Spielfläche. Weiße Zellen dürfen erobert werden, schwarze nicht. Schwarze Zellen zählen weiterhin als Hindernis.

4 Simulationsergebnisse

Im Zuge der vorliegenden Arbeit wurden zwei verschiedene Modelle simuliert und berechnet. Es wurde zuerst ein grundlegendes mathematisches Modell entwickelt und dieses dann in der Simulationssoftware Morpheus umgesetzt. Dieses Grundmodell wurde dann verändert um die Möglichkeit zu schaffen einen realen Stadtplan zu implementieren.

4.1 Ergebnisse des Grundmodells

Das Grundmodell beinhaltet drei Apotheken die auf einem Spielfeld platziert werden und sich dann Kunden erobern. Für eine Zuordnung zu einer Apotheke sind mehrere Faktoren verantwortlich. Zum einen wird untersucht, wie viele der angrenzenden Kunden bereits bei einer Apotheke einkaufen und zum anderen wird die Entfernung des Kunden zu einer Apotheke berücksichtigt. Die Simulationsergebnisse werden direkt durch die Software Morpheus zur Verfügung gestellt. Dabei wird das Spielfeld graphisch ausgewertet. Die Zuordnung der Zellen, also Kunden entsprechend ihres Kaufverhaltens zu einer Apotheke, erfolgt durch eine farbliche Markierung. Rot entspricht Apotheke 1, blau Apotheke 2 und grün Apotheke 3. Zusätzlich wird die Anzahl der Kunden, die einer Apotheke zugeordnet sind, als Diagramm und in

einer Tabelle ausgegeben, vgl. Abb.4-2 und 4-4 sowie Tabelle 1.

Abbildung 4.1 zeigt den Simulationsverlauf nach einer entsprechenden Anzahl von Schritten.

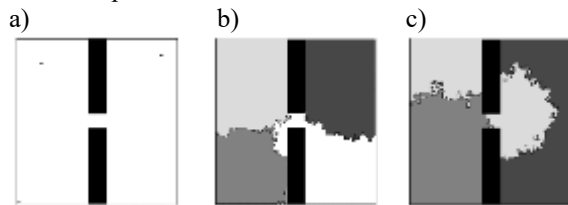


Abbildung 4-1 Ablauf der Simulation des Grundmodells nach 0, 100 und 250 Simulationsschritten

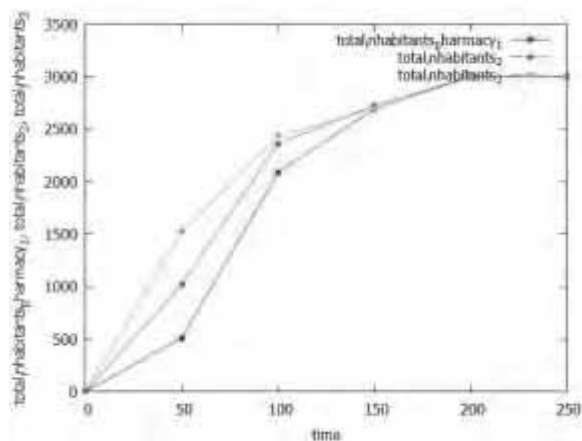


Abbildung 4-2 Entwicklung der Kundenanzahl für die drei Apotheken, nach 250 Simulationsschritten

Die nachfolgende Tabelle 1 zeigt die Entwicklung der Kundenanzahl während der Simulation. Sowohl die Grafik wie auch die Zahlenwerte zeigen deutlich, dass sich über den Zeitverlauf ein konkurrierendes Verhalten mit temporären Gewinn und Verlust ausbildet.

Simulationszeitschritt	Anzahl Kunden Apotheke 1	Anzahl Kunden Apotheke 2	Anzahl Kunden Apotheke 3
0	1	1	1
50	512	1023	1529
100	2086	2358	2434
150	2691	2724	2694
200	3000	3000	2999
250	2999	3001	2999
300	3000	3000	2999
350	3000	3000	2999
400	3000	3000	2999
450	2999	3000	3000
500	3000	3000	2999

Tabelle 1 Entwicklung der Kundenanzahl im Grundmodell über den Simulationszeitraum hinweg

4.2 Ergebnisse der Erweiterung des Grundmodells

Das Grundmodell wurde so erweitert und verändert, dass eine reale Straßenkarte als Spielfeld importiert und genutzt werden kann (vgl. 3.2). Die zugrundeliegende Überföhrungsfunktion unterscheidet sich dabei nicht vom Grundmodell, jedoch vom mathematischen Modell. Die Zuordnung der Kunden zu den Apotheken ist identisch mit der Zuordnung im Grundmodell. Entscheidend ist, dass durch die Gegebenheiten der Software Morpheus kein Algorithmus zur Berechnung der Entfernung eingesetzt werden kann, sondern nur die direkte, kürzeste Entfernung. Dies entspricht der Luftlinie. Die reale Entfernung, die durch die Straßenführung, Brücken oder besondere topografische Gegebenheiten zustande kommt, kann nicht berücksichtigt werden.

Die Auswertung der Simulationsergebnisse erfolgt identisch wie im Grundmodell um eine bessere Vergleichbarkeit zu gewährleisten.



4-3 Ablauf der Simulation auf dem Spielfeld mit der importierten Karte, nach 0, 300 und 700 Simulationszeitschritten

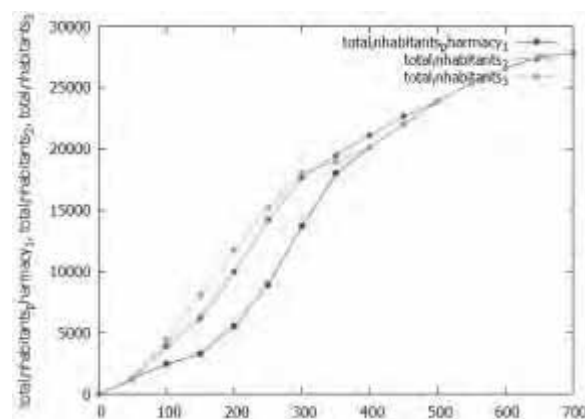


Abbildung 4-4 Entwicklung der Kundenanzahl für die drei Apotheken, nach 700 Simulationsschritten auf dem Spielfeld mit der importierten Karte

Abbildung 4-4 verdeutlicht die Entwicklung der Kundenanzahl auf dem Spielfeld mit der importierten Karte als Grundlage. Da das Spielfeld deutlich größer ist, als das Spielfeld des Grundmodells, sind auch deutlich mehr Kunden vorhanden. Die Entwicklung der Kundenanzahl ist vergleichbar mit der Entwicklung der Kundenanzahl im Grundmodell.

5 Fazit und Ausblick

Eine der Haupteigenschaften von zellulären Automaten, komplexe Sachverhalte mit einfachen Regeln abzubilden, ist auch eine der Hauptcharakteristiken von CPMs. Die Schwierigkeit liegt dabei, die einfachen Regeln aus der Realität abzuleiten. Wie bereits aufgezeigt sind sozioökonomische Prozesse Wachstumsprozesse mit sozialen Aspekten. Diese sind prädestiniert für eine Simulation mittels eines CPM. Durch den Einfluss der Nachbarschaft auf den zukünftigen Status kann der soziale Aspekt sehr einfach abgebildet werden. Zusätzlich kann die Größe der Nachbarschaft einfach verändert und an die jeweiligen Modellanforderungen angepasst werden. In Zeiten der sozialen Netzwerke, in denen etablierte Produkttester oder Produkt-Blogger mehr Vertrauen genießen als der Hersteller selbst, wird es immer wichtiger die Reputation eines Produktes oder einer Dienstleistung mit in die Betrachtung einzubeziehen.

Der Motor für ein CPM ist die Hamilton-Funktion, bestehend aus den Teiloperatoren. Somit kann das Wachstum über definierte Teiloperatoren abgebildet und umgesetzt werden. Die Teiloperatoren können einfach an das zu simulierende Problem angepasst und verändert werden. Zusätzliche Steuer- oder Hilfsgrößen können schnell und simpel an ein funktionierendes und valides Grundmodell hinzugefügt werden.

5.1 Eignung der Software Morpheus

Morpheus ist eine Simulationssoftware, die entwickelt wurde, um biologische Zellen zu simulieren. Dies schlägt sich deutlich in der Anwendung der Software nieder. Die einzelnen Plug-Ins sind sehr stark auf biologische Zellen spezialisiert. Dies erschwert die Zugänglichkeit zu dieser Software, wenn der Anwender nicht aus der biologischen, sondern aus der wirtschaftlichen Richtung ein Problem analysiert. Jede Modellvariable aus dem mathematischen Modell muss mit dem entsprechenden biologischen Gegenstück in der Software ausgestattet werden.

Ein Vorteil der Software, biologische Simulationsmodelle ohne Programmierkenntnisse zu erstellen, wurde im Laufe der Modellentwicklung zu einem Problem. Da die Software auf biologische Zellen ausgelegt ist werden zusätzliche Berechnungen oder Konfigurationen für die Nutzung in anderen Fachbereichen notwendig. Dafür muss jedoch ein stärkerer Eingriff in den Programmablauf erfolgen. Als Beispiel soll die Abweichung von mathematischen Modell zum Grundmodell dienen. Laut mathematischem Modell soll bei einem Updateversuch zuerst der Volumen-Kontrollterm ausgewertet werden und danach die Entfernung zu den Apotheken. Bei der Entfernungsauswertung soll überprüft werden ob die neue

Entfernung kürzer oder länger ist, als die bestehende. Dies ist mit der aktuellen Version der Software Morpheus nicht möglich. Hier erfolgt mit dem Plug-In Chemotaxis nur die Auswertung der Entfernung innerhalb eines Zelltyps, nicht wie benötigt über mehrere Zelltypen hinweg.

Für Testzwecke wurde versucht das Modell mit nur einem Zelltyp darzustellen. Hier konnte jedoch das Plug-In Chemotaxis gar nicht genutzt werden. Es stehen drei skalare Felder zur Verfügung die miteinander verglichen werden sollen. Ein if-Konstrukt, um zu entscheiden welche skalaren Felder miteinander verglichen werden müssen, konnte nicht entwickelt werden. Da in dem entsprechenden Eigenschaftsfeld nur eine Feldreferenz eingetragen werden kann. Zusätzlich kann keine Auswertung einer logischen Operation in dem Eigenschaftsfeld durchgeführt werden.

Vorteilhaft ist die automatische Erstellung des Dependency Graph in der Software Morpheus. Hier wird, während der Programmnutzung, ein Übersichtsgraph erstellt der alle Modellvariablen beinhaltet. Zusätzlich wird über diesen Graphen visualisiert, wie die einzelnen Modellvariablen untereinander in Beziehung stehen.

Zu Beginn ist das Konzept der Software, ein Simulationsmodell ohne Programmierkenntnisse zu erstellen, sehr vorteilhaft um schnell erste Erfolge zu erzielen. Wird jedoch das Fachgebiet, für den die Software entwickelt wurde, verlassen, zeigt eben dieses Verhalten schnell die Grenzen der Software auf. Die Programminterne Dokumentation erklärt die wichtigsten Funktionen und enthält die mathematischen Berechnungsformeln. Dies unterstützt den Anwender während der Modellentwicklung aber auch während der Validierung. Ein weiterer großer Vorteil der Software ist die umfassende Debugging-Option. Mit deren Hilfe kann effektiv analysiert werden, wo das aktuelle Modell einen Fehler aufweist.

Dennoch kann gesagt werden, dass grundsätzlich CPM ein effektives Verfahren darstellen um sozioökonomische Prozesse abzubilden. Die Simulationssoftware Morpheus stellt, in dem aktuellen Entwicklungsstand Version 1.9.1 Revision 06.04.2016, noch keine leicht zugängliche Simulationssoftware dar. Morpheus kann jedoch bei der Einarbeitung in das Thema Simulation und bei der Entwicklung erster grundlegender Modelle einen erheblichen Beitrag leisten. Mit Hilfe der Erfahrung, die während der Erstellung von Modellen gesammelt wird, kann Morpheus entsprechend weiterentwickelt werden, und so auch für andere Fachbereiche eine ansprechende und leistungsgerechte Alternative darstellen.

6 Zukünftige Entwicklung und Ausblick

In der hier vorliegenden Arbeit wurde ein erstes Grundmodell entwickelt, mit dem die Kundenanzahl für drei verschiedene Apotheken untersucht werden kann. Zusätzlich kann auch eine reale Stadtkarte implementiert werden. Als nächster Entwicklungsschritt muss ein Algorithmus entwickelt werden um die Entfernungen effizient zu ermitteln. In dem aktuellen Modell geschieht das nur über die Formel zur Ermittlung von Abständen in zweidimensionalen Ebenen.

Es müssen weitere Faktoren berücksichtigt werden, die beschreiben, unter welchen Umständen ein Kunde eine Apotheke wechselt. Also wenn ein Kunde bereits einer Apotheke zugeordnet wird, muss genauer definiert werden, wann der Kunde einer anderen Apotheke zugeordnet wird.

Ebenso fehlen noch weitere Rückkopplungen. So zum Beispiel sollte sich die geringere Anzahl von Kunden der Apotheke 1, während der ersten Simulationsschritte, negativ auswirken, da eine Apotheke mit weniger Kunden weniger Umsatz erzeugt und dementsprechend weniger Liquidität zur Verfügung hat, um in zusätzliche Kampagnen zu investieren.

Ein weiterer interessanter Aspekt ist, dass in einem zukünftigen Projekt erforscht werden kann, ob bereits bestehende Modelle, die andere Bestandteile ökonomischer Wachstumsprozesse und deren Auswirkungen simulieren, in das hier beschriebene Modell integriert werden können.

Ebenfalls könnte das Modell dahingegen erweitert werden, dass auch die Verteilung der Arztpraxen berücksichtigt werden und wie sich diese Entfernungen auf die Apotheken auswirken. Im Zuge dieser Erweiterung muss auch die Auswirkung der Öffnungszeiten berücksichtigt werden.

Auch denkbar ist eine Anpassung und Erweiterung der verwendete Simulationssoftware Morpheus um komplexere und vor allem fachfremde Simulationsprojekte einfacher umzusetzen. Eine denkbare Strategie, die überprüft werden muss, ist eine Trennung in zwei Anwendergruppen. So kann eine Anwendergruppe, zum Beispiel Anfänger, mit der bestehenden Software Modelle entwickeln und umsetzen. Die Anwendergruppe Fortgeschrittener kann dagegen in die Formeln und den Simulationsablauf eingreifen. Eine Umstellung zwischen den beiden Anwendergruppen kann per Menüeintrag erfolgen. Auch hier ist, wie bereits erwähnt, eine genauere Prüfung über den Umfang und die Auswirkungen des Eingriffes notwendig.

All dies sind weitere Projekte die, die hier vorgestellte Arbeit sinnvoll erweitern können.

7 References

- [1] Modellbildung und Simulation, Eine anwendungsorientierte Einführung Herausgeber Prof. Dr. Hans-Joachim Bungartz, Dr. Stefan Zimmer, Dipl.-Inf. Martin Buchholz, Dipl.-Inf. Dirk Pflüger, 2009 Springer-Verlag Berlin Heidelberg
- [2] Modellierung komplexer Prozesse durch naturanaloge Verfahren, Soft Computing und verwandte Techniken, Herausgeber: PD Dr. phil. Christina Klüver, Prof. Dr. phil. Jürgen Klüver, Dr. rer. nat. Jörn Schmidt, Springer Fachmedien Wiesbaden 2009, 2012
- [3] Pixelspiele, Modellieren und Simulieren mit zellulären Automaten, Herausgeber: Daniel Scholz, Springer-Verlag Berlin Heidelberg 2014
- [4] Simulation diskreter Prozesse, Methoden und Anwendungen, Herausgeber: Ulrich Hedtstück, Springer-Verlag Berlin Heidelberg 2013
- [5] Multi-Scale Modeling in Morphogenesis: A Critical Analysis of the Cellular Potts Model Herausgeber Professor Anja Voß-Böhme, Artikel auf der Webseite PLOS One unter: <http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0042852> abgerufen am: 20.07.2016
- [6] Morpheus-Webseite unter: <https://imc.zih.tu-dresden.de/wiki/morpheus/doku.php?id=start>, abgerufen am 24.03.2016
- [7] Cellular Potts modeling of tumor growth, tumor invasion, and tumor evolution, Herausgeber András Szabó und Roeland M. H. Merks, Artikel auf der Webseite unter: <http://journal.frontiersin.org/article/10.3389/fonc.2013.00087/full> abgerufen am 20.07.2016
- [8] Integrating Intracellular Dynamics Using CompuCell3D and Bionetsolver: Applications to Multiscale Modelling of Cancer Cell Growth and Invasion, herausgeber: Vivi Andasari, Ryan T. Roper, Maciej H. Swat und Mark A. J. Chaplain, Artikel auf der Webseite PLOS One unter: <http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0033726> abgerufen am: 20.07.2016

An Explicit Approach for Asynchronous Step Size Control in Co-simulation

Wolfgang Müller¹, Felix Breiteneker²

¹Austrian Institute of Technology, ²Vienna University of Technology
wolfgang.mueller.fl@ait.ac.at

Co-simulation systems can benefit from adaptive communication step size. Unfortunately, many simulation tools don't provide the information necessary for error estimates. Recent research has provided various solutions to this problem. Different properties of the parts of a co-simulation setting also limit the numerical methods that can be used. Asynchronous algorithms are a topic of current research. In this paper, the use of an explicit error estimator for the co-simulation subsystems in combination with an asynchronous communication algorithm is examined. A numerical example is used to compare the presented approach to classic approaches and demonstrate its potential advantages. The applicability of the approach in current co-simulation settings with their current limitations is discussed, in particular with respect to the Functional Mock-up Interface.

1 Introduction

There are currently two common approaches to the simulation of multi-disciplinary systems. The implementation in a general purpose modeling language and the use of co-simulation. Although general purpose languages evolved a lot since their inception, the use of co-simulation is often necessary. This can be because of the use of different tools that are a standard in their field, whose reimplementing or verification is too expensive, or that are closed source. Also, the use of co-simulation in the area of distributed computing is of interest. The use of different pieces of simulation software and the exchange of data between them is not a new concept, but has been increasingly used over the last years.

An effort to standardize co-simulation is the Functional Mock-up Interface (FMI), an effort to specify how data is exchanged between simulation tools [1]. This interface specification is open source and a growing number of tools supports it. A closer look at the FMI specification reveals one of the problems in co-simulation, especially when using commercial tools: the limited access to the involved submodels. It is not necessarily known which algorithm are used, the local

error estimates remain hidden and it might be impossible to redo a time step, e.g. with smaller step size.

The most basic approach in co-simulation is to choose a sufficiently small time grid, let the subsystems calculate their results and exchange data at this fixed communication points. Since fixed step size can be very inefficient, communication step size control is desirable, if all the involved tools support this. Approaches for macro step size control include the use of Richardson extrapolation [2] and an explicit predictor/corrector estimator [3]. In this paper, the latter is used to calculate the next macro step for each subsystem and at these steps the interpolated values of the other subsystems are used.

Co-simulation can be described as a system of $r \geq 2$ coupled subsystems, which are described by Ordinary Differential Equations (ODEs) or Differential Algebraic Equations (DAEs):

$$\left. \begin{aligned} \dot{x}_i &= f_i(x_i, u_i, t), \\ y_i &= g_i(x_i, u_i, t), \\ u_i &= c_i(y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n) \end{aligned} \right\} i = 1, \dots, r, \quad (1)$$

where the x_i are the state vectors, u_i the inputs, y_i the outputs of the respective subsystems, and the c_i define the coupling of the subsystems [4].

In classic approaches there is a grid of communication points $T_{\text{start}} = T_0 < \dots < T_N = T_{\text{end}}$, where data is exchanged. Each macro step has the step size H_n with $T_{n+1} = T_n + H_n$, and inside these steps the subsystems are solved by their respective solvers, with one or several micro steps for each macro step. For the micro steps the inputs can be interpolated, depending on the used software, in the worst case with polynomials of degree 0, using the input value at the macro step.

2 Error Estimation

For macro step size control, the algorithm described in [3] is used. The local error of each subsystem i can be analyzed using the local truncation error at the step $T_n \rightarrow T_{n+1} = T_n + H_n$, using the calculated result $y_{i,n+1}$ and the actual result $y_i(T_{n+1})$:

$$\epsilon_{i,n+1} = \|y_{i,n+1} - y_i(T_{n+1})\|.$$

Because the repetition of macro steps might not be possible, the error of subsystem i is estimated explicitly using Lagrange extrapolation polynomials of degree q :

$$\tilde{y}_{i,n+1}^{(q)}(t) = \sum_{j=0}^q y_{i,n-r-1+j} \cdot L_j^{(q)}(t),$$

where $y_{i,n-q+j}$ are the output vectors at the $q+1$ previous macro steps and $L_j^{(q)}(t)$ are the according Lagrange basis polynomials. This prediction polynomial relates to the actual value by

$$\tilde{y}_{i,n+1}^{(q)}(T_{n+1}) = y_i(T_{n+1}) + C_{i,n+1}^{(q)} \cdot H_n^{q+1} + \mathcal{O}(H_n^{p+2}),$$

with $C_{i,n+1}^{(q)}$ being the leading error constant. Further, the result of the next step can be written as

$$y_{i,n+1}(T_{n+1}) = y_i(T_{n+1}) + C_{i,n+1} \cdot H_n^p + \mathcal{O}(H_n^{p+1}),$$

where p is the order of the polynomial used for interpolating the inputs, $C_{i,n+1}$ again being the leading error constant. The estimators for the local errors are

$$\tilde{\epsilon}_{i,n+1} = \|y_{i,n+1} - \tilde{y}_{i,n+1}^{(p+1)}(T_{n+1})\|, \quad i = 1, \dots, r,$$

they are utilized for a macro step size controller. According to [5], a PI-controller can be used for determining the step size:

$$H_{n+1} = H_n \cdot \min \left(\frac{H_{\max}}{H_n}, q_{\max}, \max \left(\frac{H_{\min}}{H_n}, q_{\min}, q \cdot \left(\frac{TOL}{\max_i \{\tilde{\epsilon}_{i,n+1}\}} \right)^\alpha \cdot \left(\frac{\max_i \{\tilde{\epsilon}_{i,n}\}}{TOL} \right)^\beta \right) \right). \quad (2)$$

H_{\min} and H_{\max} are allowed minimum and maximum values for the step size, q_{\min}/q_{\max} minimum/maximum step size ratios, q a safety factor, $\alpha = \frac{0.7}{p+1}$, $\beta = \frac{0.4}{p+1}$, TOL the macro tolerance.

It is important to mention that when having a large step size

$$H_n > \sqrt[q-p]{\frac{\|C_{i,n+1}\|}{\|C_{i,n+1}^{(q)}\|}},$$

the error estimator is not working as an estimator for the local error anymore.

3 Variable Step Size Algorithm

The standard ways of doing co-simulation use communication points to exchange data and interpolation in between these steps [4]. Different time constants of the involved subsystems lead to the idea of getting rid of communication points and using interpolated values during the whole integration. This idea is explored e.g. in [6, 7]. The algorithm used here is similar, although the notion of macro steps is kept, since the internal behavior of the subsystems cannot necessarily be controlled. In particular, this is of importance if the subsystems are unable to internally use interpolation of the inputs for their solvers.

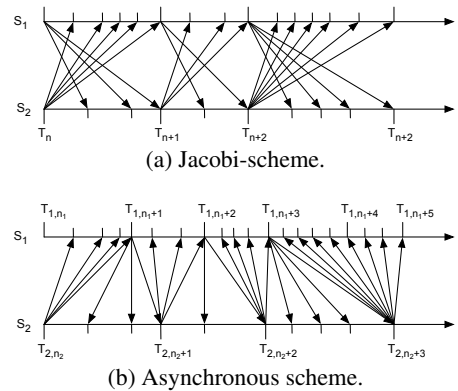


Figure 1: Data exchange at different times for the algorithms used in this paper.

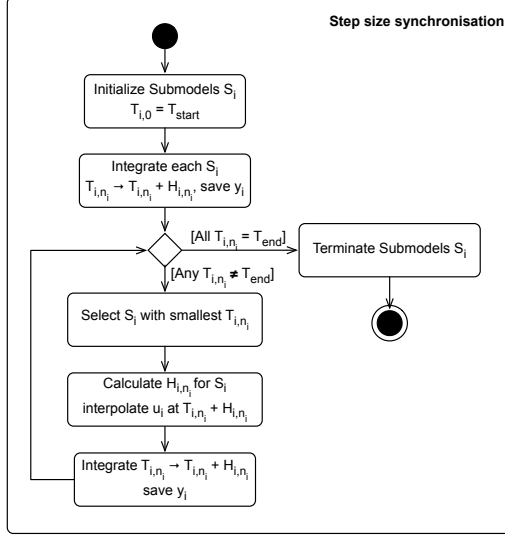


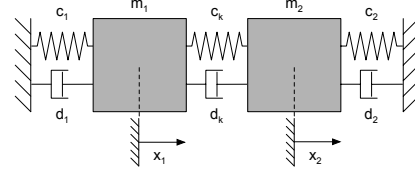
Figure 2: Flowchart for the asynchronous algorithm.

Commonly, adaptive macro step size is used to calculate the size of the next macro step and advance all subsystems to this point, exchange data, and iterate this step until the end of the simulation. This method is also referred to as Jacobi-scheme, see Fig. 1a. In the algorithm presented here, the same external estimates are made, but for each individual subsystem. After this, the subsystem that is least advanced gets the interpolated inputs of the other systems, and advances its next macro step, see Fig. 1b. This is repeated until each subsystem has reached the end of the simulation, the flowchart for the algorithm is given in Fig. 2.

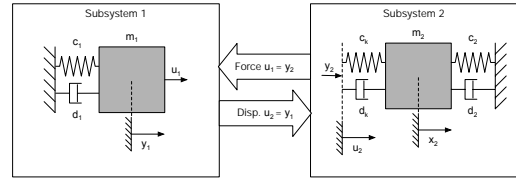
4 Test Model

As a test function for co-simulation, linear two mass oscillators (Fig.3) are commonly used [2, 3]. The oscillator can be decomposed into two subsystems, resulting in the following equations, cf. eq. (1):

$$\begin{aligned} \ddot{x}_1 &= -\frac{d_1}{m_1}\dot{x}_1 - \frac{c_1}{m_1}x_1 + \frac{1}{m_1}f_k, \quad y_1 = \begin{pmatrix} x_1 \\ \dot{x}_1 \end{pmatrix}, \\ \ddot{x}_2 &= -\frac{d_2+d_k}{m_2}\dot{x}_2 - \frac{c_2+c_k}{m_2}x_2 + \frac{d_k}{m_2}\dot{x}_1 + \frac{d_k}{m_2}x_1, \\ y_2 &= \begin{pmatrix} 0 \\ f_k \end{pmatrix}, \quad f_k = d_k\dot{x}_2 + c_kx_2 - d_k\dot{x}_1 - c_kx_1. \end{aligned}$$



(a) Two-mass oscillator.



(b) Separated two-mass oscillator.

Figure 3: Used test model: A two-mass oscillator decomposed into two single-mass oscillators by force/displacement coupling.

The parameters used in the test equation are the following, except c_k which in one example is modified to provide a stiff equation:

$$\begin{aligned} m_1 &= 1, \quad c_1 = 1e4, \quad d_1 = 2D\sqrt{c_1m_1} & D &= 0.01, \\ m_2 &= 1, \quad c_2 = 1e5, \quad d_2 = 2D\sqrt{c_2m_2} & D_k &= 0.01, \\ c_k &= 1e3, \quad d_k = 2D_k\sqrt{c_k\frac{m_1m_2}{m_1+m_2}}. \end{aligned}$$

The modification of c_k will be that way, that it is 1e3 on the time intervals $[0, 0.2]$ and $[0.5, 1]$, 1e6 on the interval $[0.3, 0.4]$ and continuously connected by third order splines between these times.

The algorithms performing the co-simulation are the ones outlined in the last section. For better comparability, the MATLAB solver ODE45 is used for the integration of the subsystems, with different tolerances TOL_{ODE45} .

Normally, the degree q of the extrapolation polynomial for the error estimator is chosen to be $p + 1$, i.e. the degree of the internal interpolation polynomial of the subsystem plus one. Since interpolation is not necessarily a feature provided by a Functional Mock-up Unit (FMU, a dynamic library or C-source code, combined with an XML description, implementing the FMI), in some examples different values for q are chosen while $p = 0$.

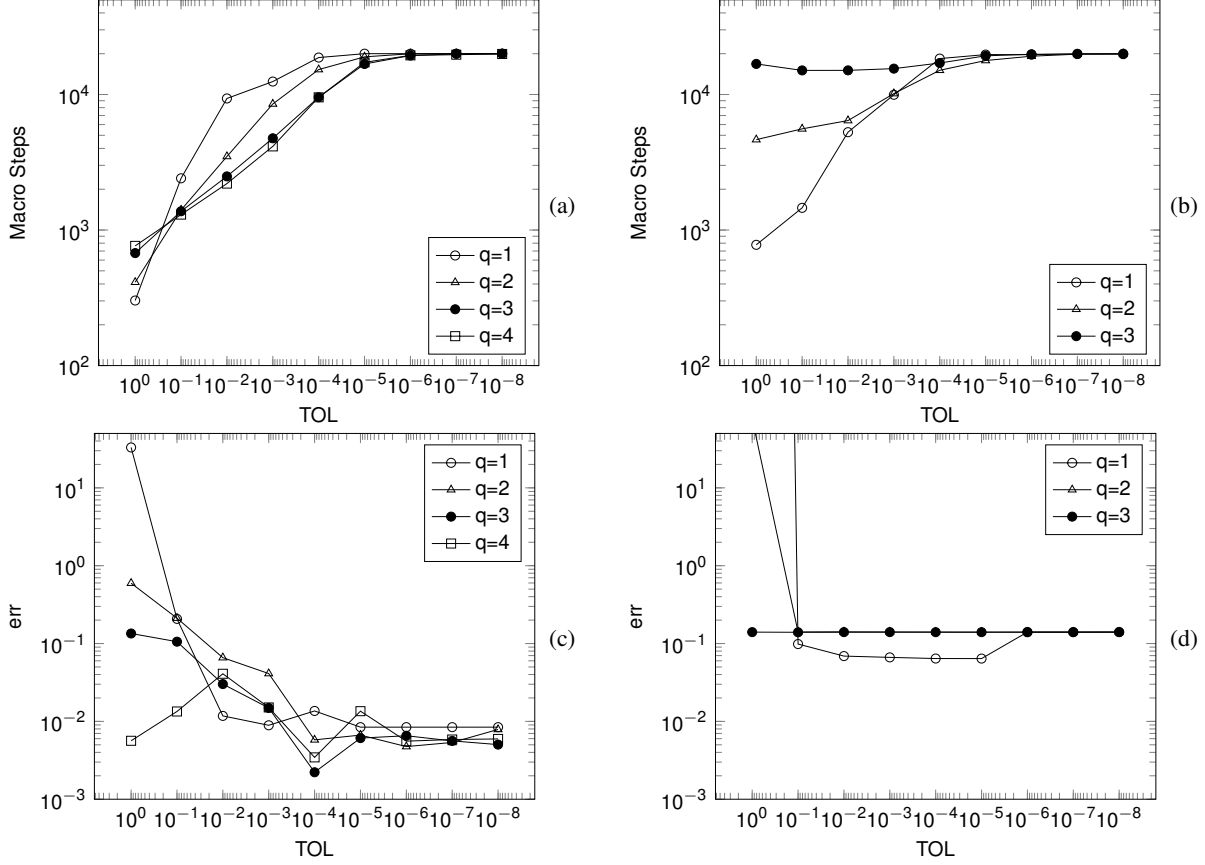


Figure 4: Number of macro steps and global truncation error using the subsystem solvers with $TOL_{ODE45} = 1E - 12$. $H_{\min} = 1E - 4$, no internal interpolation. (a) and (c) show the results using the Jacobi scheme, (b) and (d) the results using the asynchronous scheme.

5 Results

In Fig. 4 the number of macro steps and the global errors using the Jacobi-scheme and the asynchronous scheme are compared. The macro step size is limited by $H_{\min} = 1e-4$. For the subsystems a tolerance $TOL_{ODE45} = 1e-12$ is used. Since interpolation is only an optional feature of the FMI standard, no internal interpolation is used, but the estimated input for the subsystems is extrapolated with polynomials of degree p before starting the calculation.

The maximum of macro steps is reached at a certain point, using a lower tolerance only makes sense when lowering the minimal macro step size. The global error reaches its expected minimum due to the given H_{\min} . Since the test model is relatively smooth, no

problems are encountered, but with e.g. stiff problems a smaller step size or interpolation of the inputs would be necessary.

Regarding the numeric effort, Fig. 5b shows the sum of the macro steps of the two subsystems, to compare this with the classic approach, the number of steps of the latter is multiplied by two, since for each step both subsystems continue their calculations.

It can immediately be seen that the use of higher order estimates is not useful for the asynchronous algorithm in this setting. First order estimates are comparable for certain larger values of TOL .

Fig. 5 is the same example as before, only this time with interpolation of the input variables in the subsystems. It shows that the asynchronous algorithm is

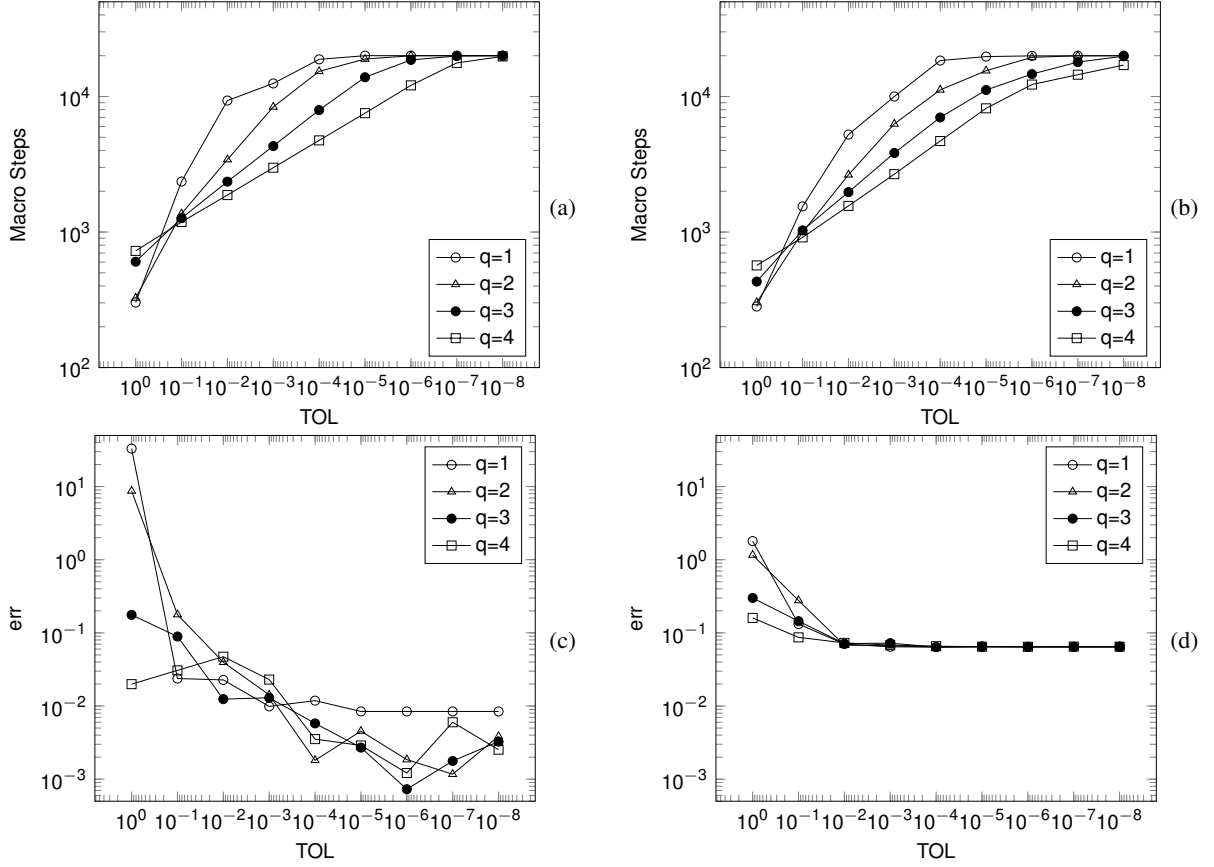


Figure 5: Number of macro steps and global truncation error using the subsystem solvers with $TOL_{ODE45} = 1e-12$. $H_{\min} = 1e-4$, with internal interpolation. (a) and (c) show the results using the Jacobi scheme, (b) and (d) the results using the asynchronous scheme.

quite effective regarding the reduced number of macro steps. The algorithm also converges faster than the classic approach.

Fig. 6 shows the results of the algorithm applied to a stiff problem, with the modifications to the test model described in the last section. The settings are $TOL = 1e-6$ and $TOL_{ODE45} = 1e-6$. Here, without interpolation, a reduced $H_{\min} = 1e-8$ is necessary to provide convergence at all, while with interpolation $H_{\min} = 1e-4$ is still sufficient.

Table 1 compares the number of macro steps and the global error when using the Jacobi- and the asynchronous scheme. In this situation, the asynchronous algorithm is more efficient, especially since with the classic algorithm, both subsystems have to be solved for each given step.

p	Jacobi		Asynchronous		
	Macro Steps	err	Steps 1	Steps 2	err
0	46023501	5.387e-4	319003	1656380	2.245e-3
1	344246	1.636e-3	168671	308671	2.991e-3
2	201300	2.669e-3	189382	71094	7.315e-3
3	249839	2.084e-3	222678	105896	4.330e-3

Table 1: Comparison of macro steps between classic and asynchronous approach for different polynomial degrees, with $H_{\min} = 1e-8$, $TOL = 1e-6$, and $TOL_{ODE45} = 1e-6$, no internal interpolation.

6 Conclusion and Outlook

The results show that the presented methods can be utilized to improve the efficiency of co-simulation set-

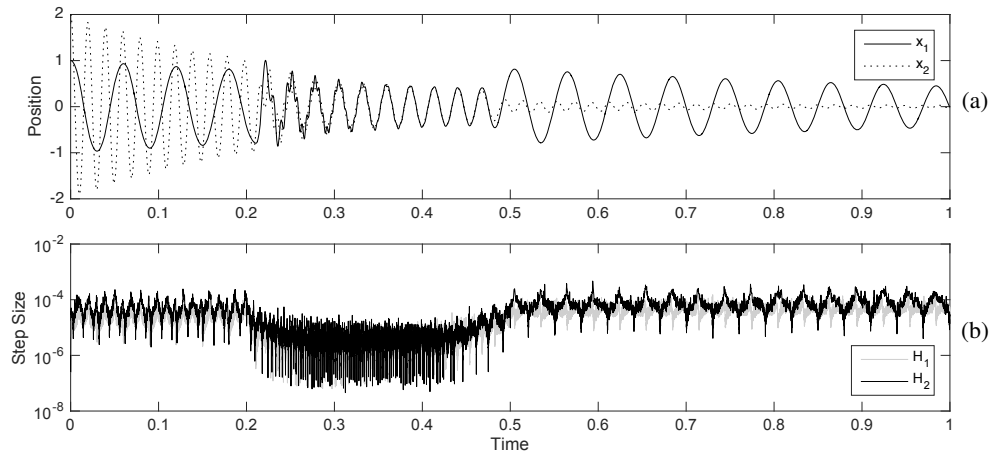


Figure 6: Result and step sizes for a stiff system, $H_{\min} = 1e-8$, $TOL = 1e-6$, $TOL_{ODE45} = 1e-6$, no internal interpolation.

tings that are limited by their subsystems. In general, the complete access to the internals of the involved subsystems would be preferable, but is unrealistic in the foreseeable future. Thus, methods to bypass certain limitations are needed. The methods in this paper require a minimum of features of the involved subsystems. No rewinding of the involved subsystems is necessary and the whole work is done within the master algorithm

While the asynchronous algorithm without internal interpolation might be more effective when applied to certain stiff equations, further investigations are necessary to see if it is possible to decide in which situation its application is warranted. When used with internal interpolation, on the other hand, it seems be able to improve the effectiveness of co-simulation.

Comparing the results without internal interpolation of the input variables, it is clear that the FMI can profit from interpolation. In the current 2.0 version, the ability to interpolate inputs is optional and implemented via the passing derivatives of the inputs. The possibility to more effectively handle input polynomials could be an alternative. Still, it is possible to implement the interpolation completely internally, using more than the last received value. In this case it would be interesting to know about the ability of FMUs to do so, to make sure to use this capability more efficiently and avoid additional potential sources of numerical errors.

References

- [1] Functional mock-up interface. www.fmi-standard.org.
- [2] T. Schierz, M. Arnold, and C. Clauß. Co-simulation with communication step size control in an fmi compatible master algorithm. In *9th International Modelica Conference*, 2012.
- [3] M. Busch and B. Schweizer. An explicit approach for controlling the macro-step size of co-simulation methods. *ENOC 2011*, 2011.
- [4] R. Kübler and W. Schiehlen. Two methods of simulator coupling. *Mathematical and Computer Modelling of Dynamical Systems*, 6(2):93–113, 2000.
- [5] K. Gustafsson, M. Lundh, and G. Söderlind. A pi stepsize control for the numerical solution of ordinary differential equations. *BIT Numerical Mathematics*, 28(2):270–287, 1988.
- [6] K. Petridis, A. Klein, and M. Beitel Schmidt. Asynchronous method for the coupled simulation of mechatronic systems. *PAMM*, 8(1):10521–10522, 2008.
- [7] S. Liang, H. Zhang, and H. Wang. Combinative algorithms for the multidisciplinary collaborative simulation of complex mechatronic products based on major step and convergent integration step. *Chinese Journal of Mechanical Engineering*, 24(3):355–363, 2011.

Automatic Layout of Scilab/Xcos Diagrams

Chenfeng Zhu¹, Umut Durak², Sven Hartmann¹, Clément David³

¹Clausthal University of Technology, Department of Informatics

²German Aerospace Center (DLR), Institute of Flight Systems

³Scilab Enterprises

chenfeng.zhu@tu-clausthal.de

Scilab/Xcos is a graphical modeling and simulation environment for hybrid dynamic systems. It provides a graphical editor which allows representing models with block diagrams. While each block represents a computational function, links specify the data and event flow. However, as the number of the blocks and the links increases, the Xcos schema can quickly become messy and difficult to read. In this paper, we present an approach for automatically updating the layout of an Xcos schema by manipulating the links and the split blocks, so that the diagrams can be kept well-presented and readable. In this approach, we update the link styles with an optimal route and then, rearrange the positions of blocks. The proposed approach is exemplified with sample Xcos models. In addition to providing the automatic layout capability to the Scilab/Xcos user, an application programming interface is also specified for the Scilab/Xcos developer who want to further enhance the provided feature set.

1 Introduction

Scilab is a free and open source software about numerical computation for engineering and scientific applications [1]. Xcos is the graphical modeling environment of Scilab for modeling and simulation of hybrid dynamical systems [2].

When using Xcos to create models, we often create blocks to implement computational functions and use links to connect them for data and event flow. They are all well-organized at the beginning as we start building up the model. However, as the model becomes more and more complex, and the number of blocks increases, we require layout rearrangements more frequently; we start moving the blocks or links every now and then. But manual layouting is hard, labor intensive and error prone. Such an effort usually ends up with a model in a messed up layout, which makes the diagram ugly and difficult for modification. Thus the readability and the maintainability of the model is decreased. Figure 1.1 is an example of a disordered Xcos diagram.

A constant manual work for relocating the blocks and rearranging the links between blocks is required in order to maintain the model readability. While block positioning is relatively straight forward and

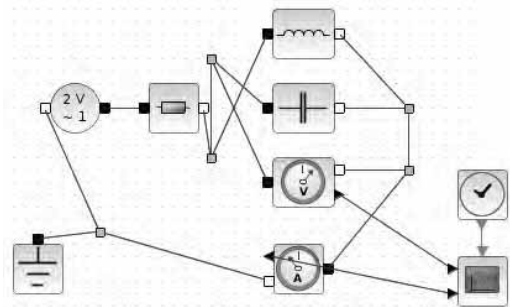


Figure 1.1: A Sample Xcos Model

even may be more efficient manually, the link rearrangements are hard and cumbersome. This effort is about developing the capability for automatically improving the model layout by manipulating the links and split blocks that connect the links to each other, thereby keeping the model readable. The Optimal Link Style (OLS) that is introduced in the Section 4 proposes an optimal route for a link which could make the link clear in the diagram. In the Section 5, the Split Block Automatic Position (SBAP) that rearranges split blocks in better positions is presented. Lastly in Section 6, the conclusion is presented and we discuss future work for a better automatic Xcos layouting.

2 Related Work

2.1 Graph Theory

For finding the optimal link between two blocks, one can apply a graph search based approach. The approaches from the graph theory to find the shortest route with the minimum cost are already quite mature. The Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which was conceived by computer scientist Edsger W. Dijkstra in 1956 and published three years later [3, 4]. The Bellman-Ford algorithm is an algorithm that computes shortest paths from a single source vertex to all of the other vertices in a weighted digraph [5]. The Floyd-Warshall algorithm is an algorithm for finding shortest paths in a weighted graph with positive or negative edge weights (but with no negative cycles) [6, 7]. These theories are practical and useful for solving the shortest path problem which could also be extended to solve the minimum-cost problem. However, in modeling graphical model, there are always more than one alternative for the connection which look good. And sometimes, some links which look good and readable are not the shortest ones.

As to the problem about the positions of split blocks, we could use some basic graph drawing theories to re-order blocks. For instance, pseudo hierarchical tree and vertical or horizontal aligned layout would be helpful to make diagrams easy to read and clear to maintain. But for both cases, our approach to the problem was to develop heuristics that capture user insight for readability.

2.2 Layouting in MATLAB/Simulink

Other graphical modeling environments such as MATLAB/Simulink also provide capabilities for formatting the layout of their own diagrams. Simulink is also a graphical editor for Model-Based Design which provides customizable block libraries, and solvers for modeling and simulating dynamic systems [8]. Compared to the automatic layout-ing of other graphical modeling environments, Simulink achieves quite an outstanding work about this. It not only provides the beautiful layout, but also could implement the dynamic features. Simulink can automatically find the "optimal path" so that the new signal line is as short as possible, has minimal 90 degree turns, and does not overlap other blocks and text. Moreover, as you draw

the signal line, Simulink lets you know exactly what the path is going to look like before you release the mouse button [9]. Additionally, Simulink provides one-click to beautify the model diagrams and auto-arrangement the blocks and lines when building new functionalities [10].

2.3 Layouting in Scilab/Xcos

In fact, even in Scilab/Xcos, it is possible to improve the general look of a diagram in using the blocks alignment options and the links style [11]. Besides Straight style and the free style with control points, there are only other 2 types of link styles provided for auto layout: Vertical and Horizontal, as Figure 2.1 and Figure 2.2 show respectively.

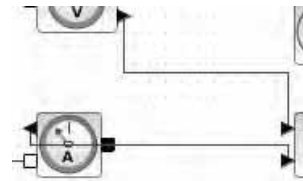


Figure 2.1: Vertical Link Style

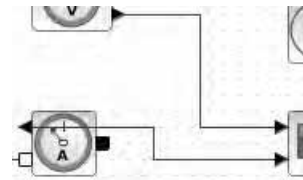


Figure 2.2: Horizontal Link Style

When the diagram becomes too complicate, the results are obviously unsatisfying. So, the effort presented in this paper targets at rearranging the blocks by putting the blocks in some new reasonable positions and to find optimal routes for the connection. After this automated process, the layout of a diagram should be enhanced and beautified for readability and maintainability.

3 Technical Solution

3.1 Overview on Scilab/Xcos

Scilab/Xcos palette provides varieties of predefined blocks such as signal processing, mathematical operations and discrete and continuous system blocks while

it is also possible to develop user-defined blocks. Despite of different types of blocks, when it comes to representation, they all belong to *BasicBlock*. We can abstract an Xcos diagram as shown in Figure 3.1.

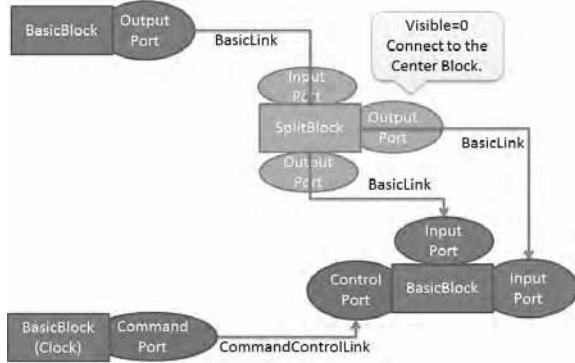


Figure 3.1: Basic Structure of Xcos Diagram

Normally, every block from palette is a *BasicBlock*. Every block owns its port(s) of input or output as its children (*BasicPort*) which belongs to four types: *InputPort* (subclasses: *ExplicitInputPort* or *ImplicitInputPort*) or *OutputPort* (subclass: *ExplicitOutputPort* or *ImplicitOutputPort*), *ControlPort* or *CommandPort*. Ports can be connected with links (*BasicLink*). *ExplicitLink*/*ImplicitLink* can be used to connect *InputPort* and *OutputPort*, and *CommandControlLink* can only be used to connect *CommandPort* and *ControlPort*. And there are also other classes implementing other functionality such as graph, palette and utilities.

All the Java codes of the Xcos program are in the "org.scilab.modules.xcos" package. The structure is shown in Figure 3.2.

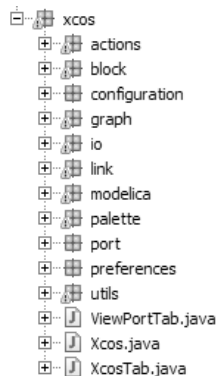


Figure 3.2: Structure of Xcos Java program

Scilab/Xcos code base consist of various types of files, such as C codes, Java codes, xml files for help docu-

ments, image resources and files for locales. The user interface is generated by using Java. So, in order to implement an automatic layouting for Xcos, we need to conduct the implementation in Java.

3.2 JGraphX

JGraphX [12] is the underlying graphics framework of Scilab/Xcos. Aligned with that, the implementation of the technical solution encompasses utilization of JGraphX for the development of automatic layouting features. JGraphX is a Java Swing diagramming (graph visualization) library licensed under the BSD license. The library is strong and easy to extend and inherit. The documentation and Application Programming Interface (API) is quite mature. JGraphX not only provides functionality for visualization and interaction with node-edge graphs, but also includes functionality like XML support which would help save the current layout of the diagram avoiding that the layout needs to be recalculated every time it is opened. Besides the features about graph interaction and graph layouts which are being used in Xcos, JGraphX provides an analysis package which includes a range of analysis functions which provided us with a number efficient building blocks for the automatic layouting.

The core architecture of JGraphX includes the JGraphX model, the transactional model and *mxCell*. The JGraphX model (*mxGraph*) is the core model that describes the structure of the graph. The class called *mxGraphModel* is the underlying object that stores the data structure of the graph [12]. The graph class *ScilabGraph* in Scilab extends *mxGraph*. The transactional model is a transaction of models update which contains a series of actions. Transaction starts with *beginUpdate* and ends with *endUpdate*. With the help of transactional model, a set of events for the compound changes could be fired together after transaction. The *mxCell* is the cell object for both vertices and edges [12]. The three key attributes for an *mxCell* is its value, its style and its geometry. The *ScilabGraphUniqueObject* extends *mxCell* and it is also the ancestor of *BasicBlock* and *BasicLink*.

We use the geometry to change the position of blocks and use the style to change the routes of links in Xcos. And we need to save all the states of Xcos diagram including the positions of blocks and the styles of links so that we do not need to re-calculate the layout every time re-opening an Xcos file.

4 Optimal Link Style

Optimal Link Style (OLS) is to find a route with more blank padding and with less turning and to use it as the style of a link.

4.1 The Functional Flow

The design of the functional flow can be briefed as below:

1. Change the style of the selected links one by one in a loop in one transaction.
2. Check whether the two points of the ports are aligned and make sure that there are no blocks between them. If so, make the link with straight style, e.g. connect them directly.
3. Otherwise, use two new points each of which is a distance away from its corresponding port (if it was SplitBlock, use its center directly instead of its port).
4. Then start with these two new points, try to find the new route with one single turning point or 2 turning points. Otherwise, get another new point away from the starting point and use this to find a route with the same method. This could be recursed in several times.
5. Remove the unnecessary points and get the final optimal route for the link.

We modify the source files and create the classes listed in Table 4.1 to implement the functionality:

Class	Description
StyleOptimalAction	Action events
XcosRoute	Compute route
XcosRouteUtils	Common utilities

Table 4.1: Classes Created

4.2 The Methods

Here, we would like to introduce the methods and underlying mathematical model for OLS. The method signatures are declared at the beginning of each section. Thus, the reader is informed about the application programming interface for that particular method.

4.2.1 Get the position of a cell

Method:

mxPoint getCenterPoint(mxICell cell, XcosDiagram graph)

This method is used to get the position of a cell where a link will connect to. There are three situations according to this cell:

- (a) If it is a Port and its parent is a SplitBlock, use the center point of its parent.
- (b) If it is a Port and its parent is not a SplitBlock, use the state of this cell (*graph.getView().getState(cell)*) to get port's *mxCellState* to get the point.
- (c) If it is a BasicBlock, use the center point according to its geometry attribute.

4.2.2 Check if a point is in a line segment

Method:

boolean pointInLineSegment(double x1, double y1, double x2, double y2, double x3, double y3)

This method is used to check P_1 in Segment (P_2, P_3) which is shown in Figure 4.1.

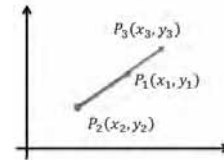


Figure 4.1: Point in Line Segment

If it is, then P_2P_1 and P_2P_3 must have the same direction and P_1 is between P_2 and P_3 .

For the same direction:

$$\angle P_2P_1 = \angle P_2P_3 \quad (4.1)$$

$$\frac{y_1 - y_2}{x_1 - x_2} = \frac{y_3 - y_2}{x_3 - x_2} \quad (4.2)$$

$$(y_1 - y_2)(x_3 - x_2) = (y_3 - y_2)(x_1 - x_2) \quad (4.3)$$

For P_1 between two points:

$$\min(x_2, x_3) \leq x_1 \leq \max(x_2, x_3) \quad (4.4)$$

$$\min(y_2, y_3) \leq y_1 \leq \max(y_2, y_3) \quad (4.5)$$

4.2.3 Check superimposition

Method:

boolean linesCoincide(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4)

These two methods are used to check whether two lines coincide or not. The second one is to check strict superimposition of two line segments. In the first one, the lines would move parallel and then check all of them. This could avoid that two line segments are too close.

Segment A (P₁, P₂) and Segment B (P₃, P₄) will coincide in these situations:

- (a) Segment A is inside Segment B, e.g. both P₁, and P₂ are in Segment B;
- (b) Segment B is inside Segment A, e.g. both P₃, and P₄ are in Segment B;
- (c) Segment A and Segment B are parallel and one of the endpoints of one segment is in the other segment.

If lines are parallel,

$$(x_1 - x_2)(y_3 - y_4) = (x_3 - x_4)(y_1 - y_2) \quad (4.6)$$

4.2.4 Check obstacles

Method:

boolean checkObstacle(double x1, double y1, double x2, double y2, Object[] allCells)

This method is used to detect whether there are obstacles between two points.

The definition of obstacles is: All top blocks and links and all the ports of blocks. EXCEPT: itself (link), its Source and Target (i.e. port) and SplitBlock.

If any of the below situations happens, it means that there is an obstacle between two points:

- (a) If it is a Link,
 - Check lines superimposition.
 - Check whether points are in the link.
- (b) If it is a Block,
 - Use `mxRectangle.intersectLine` to get an intersection if it exists according to its geometry.

4.2.5 Get orientation of ports

Method:

Orientation getPortRelativeOrientation (BasicPort

port, XcosDiagram graph) *Orientation getNewOrientation(mxICell cell, double cx, double cy, mxICell otherCell, double ox, double oy, XcosDiagram graph)*

These two methods are used to get the current orientation of a port according to its relative position to parent block.

We also consider the ports of different blocks, because the ports of a SplitBlock are not visible or the target point has no parent blocks.

- (a) If its parent is a normal Block, calculate orientation according to the relative position of the port to its parent block as Figure 4.2 shows. For instance, the orientation of the port will be EAST if this port is in the EAST zone of its parent block.
- (b) If its parent is a SplitBlock, get the orientation of the InputPort of the SplitBlock according to the relative position of link's source (it is the same mathematical model in the first case); get the orientation of one OutputPort of the SplitBlock according to the orientation of the InputPort and the positions of both OutputPorts. For instance, when one of the OUT target is on NORTHEAST to IN source, its orientation will be NORTH if the other out target is on its right side and the orientation of IN is not north; its orientation will be EAST if the other out target is on its left side.

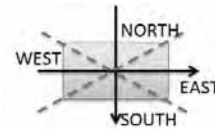


Figure 4.2: Orientation in Zones

4.2.6 Get a point away from port

Method:

mxPoint getPointAwayPort(mxICell port, double portX, double portY, Orientation orien, Object[] allCells, XcosDiagram graph)

This method is used to get a new point away from a port according to the orientation of this Port as Figure 4.3 shows. If there are obstacles between the Port and the new point, reduce the distance and try another new point. Then use the new Point as the start/end point to compute the route.



Figure 4.3: Get a New Point away from Port

4.2.7 Choose an optimal line

Method:

```
double choosePoint(List<Double> list, double p1, double p2)
```

This method is used to choose a better line (which is the average number in the widest range in a certain density) from the discrete numbers as Figure 4.4 shows. Consider the points between $p1$ and $p2$ as a priority as Figure 4.5 shows.

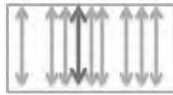


Figure 4.4: Choose an Optimal Line I

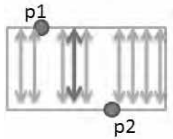


Figure 4.5: Choose an Optimal Line II

4.3 Implementation in Detail

Firstly, we get the position of the source and the target cell. If two points are aligned (boolean isStrictlyAligned(double, double, double, double)) and there are no obstacles between them (see 4.2.4), then we connect two points directly and we do not need to do the steps further.

Then, we create a point away from the port according to the orientation of each port (see 4.2.6). Using these two new points as the new starting point and the ending point, find a simple route with 2 turning points. If the source is EAST/WEST orientation, we try the point($x2$, $y1$) as the turning point and check the obstacles among the new source point, this point and the new target point. In this case, the away point for the source is unnecessary. Otherwise, we try point($x1$, $y2$) and check the obstacles among the new source point, this point and the new target point. In this case, the away point for the source is necessary. The away point

for the target is similar. This is shown in Figure 4.6. If the source is SOUTH/NORTH orientation, we try the point($x1$, $y2$) and check the obstacles among them. In this case, the away point for the source is necessary. Try the point($x2$, $y1$) and check the obstacles among them. In this case, the away point for the source is unnecessary. The away point for the target is also similar. This is shown in Figure 4.7. If we could get a route, we do not need to do the steps further.

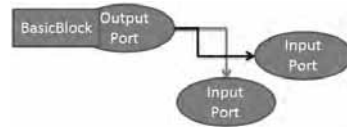


Figure 4.6: Single Turning I

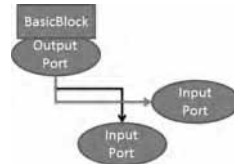


Figure 4.7: Single Turning II

At the third step, we check all the possible horizontal or vertical connections of the points whose y or x is between the two points as Figure 4.8 shows ("possible" means no obstacles between points). In case that it is full of obstacles between two blocks, we extend the range of the detection. If the orientation is horizontal, we check x firstly (the left one in the figure). Otherwise, check y firstly (the right one in the figure). If more than one optimal route is found, choose the optimal one (see 4.2.7).

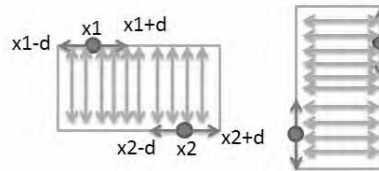


Figure 4.8: Simple Routes

Finally, if there is no optimal solution in simple mode which is introduced above, we get new away points of the start point in 3 directions and use the new points to find a simple route. Otherwise, we try to find a complex route in a recursion.

4.4 A Sample Application

In this example case, Figure 4.9 is the original diagram in a mess. Figure 4.10 is the diagram which we used OLS to format the links.

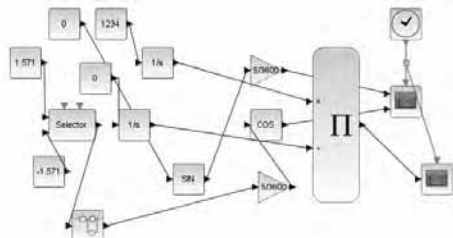


Figure 4.9: The Diagram in Original Version

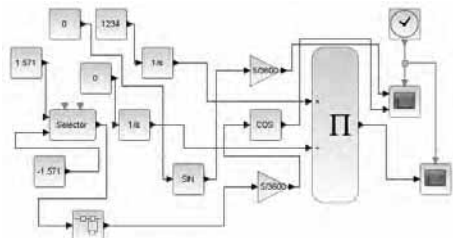


Figure 4.10: The Diagram after Using OLS

It looks better than previous layout and the links were clear for users to read. But it does not work well when there are split blocks. So we need to do some optimizations to make it more beautiful.

5 Split Block Auto Position

Split Block Auto Position (SBAP) is to find a position for split block where the links which connect to it look clearer in the optimal routes.

5.1 The Functional Flow

The design of functional flow can be introduced as below:

1. In the whole connection where the split block is, get one of the normal blocks as the source and all other blocks as the targets.
2. Compute their optimal routes separately.
3. Choose the conjunct point of the routes to be the new position of every split block.

4. Update the orientations of ports in the split block according to the routes.
5. After getting new position(s) and new orientations, update the links.

We modified the source files and created the classes listed in Table 5.1 to implement this functionality:

Class	Description
AutoPositionSplitBlockAction	Action events
BlockAutoPositionUtils	Calculate position

Table 5.1: Classes Created

5.2 The Methods

We will explain the methodology which is used for Split Block Auto Position. In this section, the methods and underlying mathematical model for SBAP will be presented. As it was for the OSL, the method signatures are declared at the beginning of each subsection in order to reveal the application programming interface for that particular method.

5.2.1 Get the root split block

Method:

SplitBlock getRootSplitBlock(SplitBlock splitblock)

This method is used to get the root split block when there are multiple split blocks in the whole connection.

1. Check if the block which connects to the IN port of this split block is a normal block.
2. If it is a normal block, this split block is the root split block.
3. If it is a split block, then check as step 1 again and start this loop until find the normal block. Then the split block is the root one.

5.2.2 Adjust routes

Method:

void adjustRoutes(List<List<mxPoint>> listRoutes, Object[] allObstacles, List<mxICell> listPorts)

This method is used to adjust routes after getting the optimal routes (using OLS). As shown in Figure 5.1, some segments in two links might be parallel (similar to 4.2.3). We move segments to make them superimposed if there are no obstacles. Then there will

be more superimpositions between routes and the last conjunct point of routes will be more meaningful.

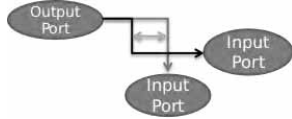


Figure 5.1: Adjusting Routes

5.2.3 Get the last conjunct point

Method:

mxPoint getSplitPoint(List<List<mxPoint>> listRoutes)

This methods is used to get the last conjunct point of all routes. After the routes are adjusted, there are different last conjunct points between every 2 routes. We choose the one which is in all routes.

5.2.4 Update orientation of port

Method:

void updatePortOrientation(SplitBlock split, List<List<mxPoint>> listRoutes, XcosDiagram graph, BasicPort input)

Orientation getInputOrientation(List<List<mxPoint>> list, mxPoint startPoint, mxPoint splitPoint)

Orientation getPortOrientation(List<mxPoint> list, mxPoint splitPoint)

These methods are used to get the orientation of ports in a split block according to the relative routes and update them. There are routes passing the split block. As shown in Figure 5.2,

1. For the IN port, from the split block point to the previous turning point is the orientation.
2. For the OUT port, there are 2 cases. If the split block is in the turning point, then from the turning point to next turning point is the orientation. Otherwise, from the previous turning point to the next turning point is the orientation.

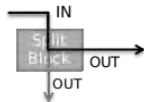


Figure 5.2: Get and Update Orientation

5.3 Implementation in Detail

When a link is split, there will be one split block generated. When the link is split several times, there will

be several split blocks. Then one split block must have one IN port and two OUT ports.

1. Calculate the number of split blocks in this whole part of linking.
2. If there is only one split block,
 - (a) Get the port which connects to the IN port of the split block as a source. And get the ports which connect to the two OUT ports of the split block as targets.
 - (b) Find the optimal routes for the source to each target.
 - (c) Use the last conjunct point in both routes as the new position of split block as shown in Figure 5.3.

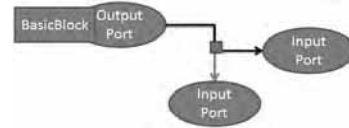


Figure 5.3: New Position of SplitBlock

- (d) Find the orientations for each ports in the split block according to optimal routes.
3. If there are more than one split blocks,
 - (a) Get the root split block. Get the port which connects to the IN port of this root split block as a source.
 - (b) And get the ports of all basic blocks which connect to all other children split block as targets.
 - (c) Find the optimal routes for the source to each target.
 - (d) Use the last conjunct point in different routes as the new position of split blocks.
 - (e) Find the orientations for each ports in the split blocks according to optimal routes.
 4. After getting new position(s) and new orientations, update the links.

5.4 A Sample Application

In the case study, Figure 1.1 is the original diagram. Figure 5.4 is the diagram which we use SBAP and OLS to format.

Now, the layout is much better. The diagram is easy to read and maintain.

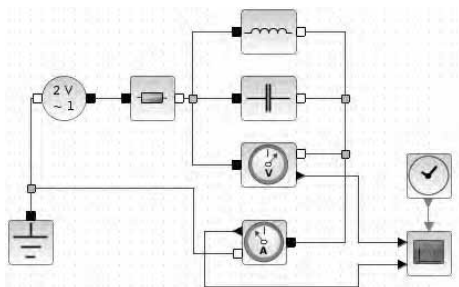


Figure 5.4: The Diagram after Using SBAP and OLS

6 Conclusion and Future Work

The paper presents Optimal Link Style and Split Block Auto Position for automatic layouting Scilab/Xcos diagrams. Optimal Link Style focuses on the link styles which format the link, while Split Block Auto Position concentrates mainly on position changing of split blocks. According to his needs, user can decide in modeling time which one to apply.

Every project or every team has its own standards or criteria about the format of the diagrams. It is not easy to decide which layout is really perfect or fulfills the needs of users. What we have tried to achieve was not only providing an automatic layout, but also giving some options for users to make their own decisions about the final layout which they would like to maintain. On the other hand, user therefore needs to select the blocks or the links which he wants to change and click some buttons to make them in an optimal layout. That means that our automatic layout is still static instead of dynamic. And we could not get a preview of the automatic layout while we are drawing our diagrams or get a direct result about this automatic feature. Sometimes, users would like to draw a link with the optimal route when creating the connection between blocks. They would also like to format the layout of the links after they move some blocks without clicking some buttons.

Based on the previous paragraph, the future work includes improving the user experience. While the application programming interface for the layouting methods provides a baseline, it is necessary to experiment various user interaction scenarios.

References

- [1] S.L. Champbell, J.P. Chancelier, and R. Nikoukhah. *Modeling and Simulation in Scilab/Scicos*. Springer Science and Business Media, Inc., 2006.
- [2] Scilab Enterprise, *Xcos Features*. Retrieved July 30, 2016 from www.scilab.org/scilab/features/xcos.
- [3] P. Frana. *An Interview with Edsger W. Dijkstra*. Communications of the ACM 53 (8): 41–47, 2010.
- [4] E. W. Dijkstra. *A note on two problems in connexion with graphs*. Numerische Mathematik 1: 269–271, 1959.
- [5] J. Bang-Jensen, and G. Gutin. *The Bellman-Ford-Moore algorithm*. Digraphs: Theory, Algorithms and Applications, Springer Science and Business Media, 2000.
- [6] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms (1st ed.)*. MIT Press and McGraw-Hill, 1990.
- [7] K.H. Rosen. *Discrete Mathematics and Its Applications, 5th Edition*. Addison Wesley, 2003.
- [8] MathWorks, *Simulation and Model-Based Design*. Retrieved July 30, 2016 from <http://www.mathworks.com/products/simulink/>.
- [9] G. Roulaeu. *Smart Signal Routing*. Retrieved July 30, 2016 from <http://blogs.mathworks.com/simulink/2012/10/11/smart-signal-routing/>.
- [10] L.K. Klauske, and C. Dziobek. *Improving Modeling Usability: Automatic Layouting for Simulink*. Retrieved July 30, 2016 from <http://www.mathworks.com/videos/improving-modeling-usability-automatic-layouting-for-simulink-93139.html>.
- [11] Scilab Enterprise. *Xcos for Very Beginners*. Retrieved July 30, 2016 from <http://www.scilab.org/community/news/20130830>.
- [12] JGraph Ltd. *GraphX (JGraph 6) User Manual*. Retrieved July 30, 2016 from <https://jgraph.github.io/>.

A new approach for integrating discrete element method into component-oriented system simulations

Christian Richter¹

¹Technische Universität Dresden, Chair of Construction Machinery
christian.richter1@tu-dresden.de

The working process of construction and conveying machines is characterized by the interaction with granular and bulk materials. In order to allow prospective analysis of machine behaviour under real operating conditions, coupled simulations are increasingly used. While modelling the equipment happens within the scope of component-oriented system modelling, reproducing particle-mechanical behaviour is done with discrete element method. The work presented here introduces a new integrated approach which allows a closed modelling and simulation of system models and discrete element method. The creation and calculation of coupled simulations is thus facilitated by a multiple.

1 Introduction

In recent years the usage and importance of simulations has significantly increased in the field of construction and conveying machines. Both sectors have in common that some kind of machine is handling with some kind of granular material (e.g. sand, gravel, pellets). Typical processes are the digging of a hole with an excavator or the transport of material with a conveyor belt. Knowing the forces arising from these processes is very important. On the one side they cause strains on single parts, on the other side they can affect the entire machine behaviour. Let's consider an excavator for example. The forces coming from the digging process are acting on the bucket and causing stress and wear, but they will also lead to an increasing hydraulic pressure in the cylinders. This in turn will affect the pump- and engine-activity. In order to make prospective statements about machine behaviour under real operating conditions it's necessary to simulate the machine as well as the process in common.

For simulating the equipments behaviour, component-oriented system models are often used. These kind of models are describing the machine as a network of components and subsystems which can be part of different domains like hydraulics, electrical or control engineering. One option for simulating bulk materi-

als is using the discrete element method (DEM). With help of this method it's possible to reproduce the motion of granular materials as well as the strains they cause on mechanical parts.

Bringing both simulation techniques together isn't as easy as it seems. In addition to various methods of calculation, the modelling paradigms are also completely different. Arranging coupled simulations is for that very difficult and time consuming. This paper presents a new integrative approach for closed modelling and simulation control.

2 Basics

2.1 Discrete Element Method

The discrete element method (DEM) was presented first in 1973 by Cundall and Strack [1]. It's a numerical method for simulating the behaviour and motion of large numbers of discrete, interacting objects. In most cases, as done here, these objects are referred as particles. Basis of the method is the calculation of forces acting between the particles or between a particle and an adjacent surface. The basic calculation cycle should be explained briefly below.

After insertion every particle has an initial position and velocity. The simulation loop starts with *collision detection*. In this phase all particle-particle and particle-wall contacts are determined. After that the forces and torques acting on every particle must be calculated. These forces result on the one hand from field forces like gravity and on the other hand from the particle deformation as a consequence of collision. For that different *force-deformation laws* are used. By summing up all single forces and torques, the translational and angular acceleration of each particle can be obtained. The last step is *solving the equations of motion*. For that the new positions and velocities are resolved by integrating translational and angular acceleration two times (equations 1 and 2). Figure 1 shows this loop. It is repeated until a predetermined number of iterations is reached.

$$\vec{F}_i = m_i \cdot \ddot{\vec{x}}_i \rightarrow \vec{x}_i = \iint \ddot{\vec{x}}_i dt \quad i = 1, 2, \dots, N \quad (1)$$

$$\vec{M}_i = J_i \cdot \ddot{\vec{\phi}}_i \rightarrow \vec{\phi}_i = \iint \ddot{\vec{\phi}}_i dt \quad i = 1, 2, \dots, N \quad (2)$$

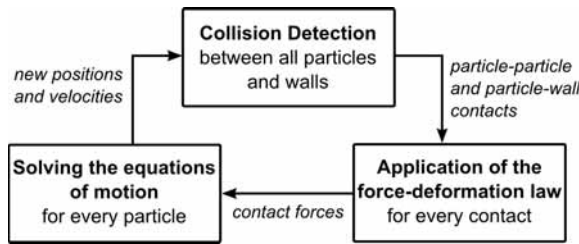


Figure 1: DEM Calculation Loop

Most software for modelling and simulating discrete element models works command-oriented. The user tells the software what to do by typing single instructions into a command line tool or by loading an input script. Graphical user interfaces are very rare. After typing the commands they are processed sequentially. Typical representatives are open source solutions like LIGGGHTS® and Yade® or commercial tools like PFC3D®. This kind of modelling and user-software-interaction has historical reasons and is not very user friendly. The biggest problem is that the user has to have good knowledge about the commands and syntax. Furthermore its very complicated to adapt existing models onto new problems. Figure 2 shows an excerpt of a typical input script for LIGGGHTS®.

Most of these applications doesn't have any integrated post-processing tool. This is a disadvantage because

```

atom_style      granular
atom_modify     map array
boundary        mm mm mm
newton          off
echo            both
communicate     single vel yes

units           si
neighbor        0.200000 bin
neigh_modify    delay 0
region          reg block -2.750 0.000 -2.200 2.200 -0.100 4.070 units box
create_box      2 reg

# Material Parameters
fix m0 all property/global youngsModulus peratomtype 2e9 2e9
fix m1 all property/global poissonsRatio peratomtype 0.3 0.3
fix m2 all property/global coefficientFriction peratomtypepair 2 0.5 0.4 0.4 0.3
fix m3 all property/global coefficientRestitution peratomtypepair 2 0.5 0.4 0.4 0.3
fix m4 all property/global coefficientRollingFriction peratomtypepair 2 0.5 0.4 0.4 0.3
fix m5 all property/global cohesionEnergyDensity peratomtypepair 2 0 5000 5000 10000

pair_style      gran model hertz tangential history cohesion sjkr rolling_friction epsd2
pair_coeff       * *

variable dt equal 0.00001

fix integrator all nve/sphere

```

Figure 2: Input Script for LIGGGHTS®

visualization of results and particle data is very important. For that it's not seldom that users have to use third-party applications like ParaView®.

2.2 Networkbased System Simulations

Simulating complex machines - as already mentioned - is often done by using equation-based component-oriented system models. For the description of such models Modelica [2] has established as a kind of standard language in many areas. Meanwhile, there is a plurality of applications using and supporting Modelica out there (e.g. SimulationX® or Dymola®). Modelica allows the model description on several levels. One of the most common types of building up simulation models is the network-based or component-oriented modelling technique. Therefore configurable components and subsystems (network elements), which have clearly defined interfaces, are connected together in a network structure. Apart from classic advantages of object-oriented approaches, such as encapsulation and re-usability through modularity and inheritance, this method also has other benefits. One is that building up and modifying models can be done very easy and rapidly. A further is the high degree of clarity, as the real and virtual structure often correspond with one another. Exemplary the top-level structure of an engine model is shown in figure 3.

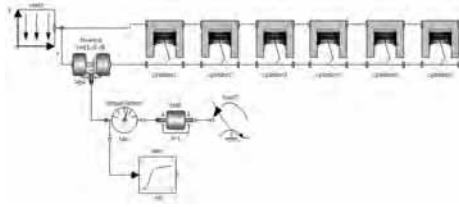


Figure 3: Networkmodel of an engine with 6 cylinders

2.3 Coupled Simulations

As coupled simulation generally the calculation of coupled systems is meant. Coupled-systems consisting of two or more models from different domains, which have to exchange information with each other at simulation runtime. Furthermore the number of integrators and/or modelling tools must be greater than one. According to [3] there are different kinds of coupled simulations with different names as shown in figure 4.

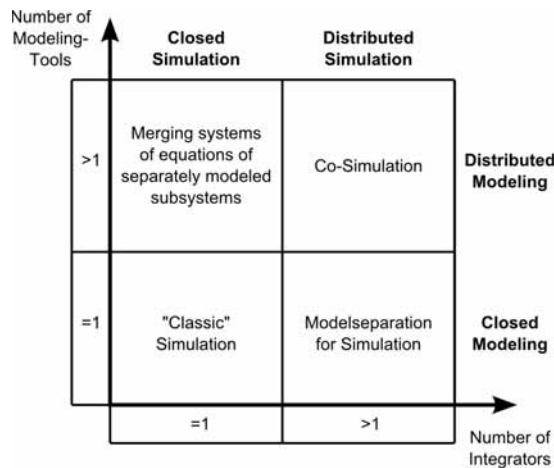


Figure 4: Classification of coupled simulations

Assuming to this most recent approaches of coupling system simulations with discrete element method must be classified as co-simulation. One example for this is the approach presented in [4]. This approach starts with building up a machine model and exporting it as a functional mock-up unit (FMU). Afterwards a discrete element model is build up and connected to the FMU. The calculation is performed by two integrators with information exchange between both models at discrete time events. Building up and connecting the models causes a lot of work. This amount of work

can be reduced by implementing a new functionality which allows closed modelling.

Generally, there are two possibilities for implementing such a functionality. Either it's attempted to integrate the DEM into component-oriented systems modelling or the other way around. From the points and facts mentioned above the first variant seems to be the better way for getting a high degree of usability.

3 Integrating DEM into component-oriented system models

3.1 Component Library

For transforming the command-oriented modelling paradigm of classic DEM-tools into a component-oriented modelling technique several things have to be done. First of all, all relevant modelling functions must be identified. After this new abstracted components and parameters must be designed representing and implementing these functions. This components have to be very self-explaining and easy to understand for the user. Table 1 shows a selection of components and their corresponding function.

Component	Functions
SimulationBox	get total count and mass of all particles
ParticleSource	generate particles
ParticleSink	remove particles
ParticleFlowSensor	measure the number and mass of particles passing a surface
ParticleRegionSensor	measure the number and mass of particles in a volumetric region
ParticleSet	loading existing data
RigidBody	interaction of a rigid body with the particles

Tabelle 1: Components and corresponding functions

As last step a translator must be implemented, which is capable to translate these components into commando sequences.

3.2 System structure

A component library as described before can be used in any Modelica-Tool. It allows the closed modelling of machine and process models. In order to perform a distributed simulation, models have to be subsequently separated. For a better understanding how this is done figure 5 shows the basic structure of all simulation components. This structure is divided into two main areas - a front-end and a back-end.

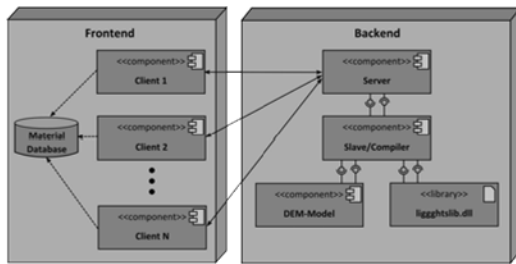


Figure 5: System structure

The front-end consists essentially of the component library. Each of the components forms a client, which is capable to connect and communicate via TCP/IP to a server. Due to the fact that the presented solution was developed as part of a research project, the component library has some SimulationX[®]-specific features. One of them is the material selection via a database interface.

The server the components can connect to forms the root node of the back-end-structure. It receives the messages from the components and forwards them to a specific compiler. It's possible that components belonging to different simulation models can connect to the server. For that reason the server can handle more than one compiler at the same time. The compiler uses the information coming from the server to build up a copy of the current component structure. With help of this model commando-sequences are generated which can be executed by a specific DEM-tool. The software used here was LIGGGHTS[®] compiled as a shared library to make information exchange easier. Holding back the whole component structure at the back-end

is necessary for the translation process because some commands can only be created with information coming from two or more components.

The division into front-end and back-end communicating via TCP with each other seems at first sight a little bit complicated but brings some advantages. One of them is the fact that distributed computations are made possible. That means solving the system model can be done on a normal computer while discrete element simulation runs on a workstation or computer cluster. Especially for time consuming and expensive DEM simulations that's a plus.

3.3 Communication

At this point, communication between front-end and back-end is briefly explained. At the beginning of every simulation - that means during initialization phase of the system model - all components respectively clients connect to the server. Required connection settings, such as server address and port, are coming from the so called *SimulationBox*-component. The following code snippet shows this.

```
1 outer SimulationBox simbox „simBox“;
2 Boolean isConnected(start=false);
3 ExternalObject client=TcpClient();
4 equation
5   when initial() then
6     isConnected=connectToServer(
7       client,
8       simbox.address,
9       simbox.port);
10  end when;
```

The SimulationBox is a kind of superordinate parent or world object. Besides basic simulation settings it contains information about the spatial domain for DEM and gravity forces acting on all particles. All other components can access this information.

After a successful connection, the communication between front-end and back-end starts. It's divided into three phases:

1. During initial phase all front-end components send their parametric information to the server. These are, for example, initial positions and orientation, geometrical data or material values. All

this information are collected at the back-end and used to generate executable commando sequences.

2. At simulation runtime information are exchanged in regular intervals. Here mainly new calculated positions of rigid bodies are transmitted from front- to back-end and forces as well as torques are returned.
3. At the end of simulation an information is sent to the server, which tells this that calculation is over. This will reset the back-end simulation.

Simplified code, containing the complete communication algorithm, is shown below.

```

1 parameter Real tc=0.0001;
2 Boolean commTrigger=sample(0, tc);
3 algorithm
4   if isConnected then
5     when initial() then
6       ...send initial data...
7     elseif commTrigger then
8       ...send and receive data...
9     elseif terminal() then
10      ...send final data...
11   end when;
11 end if;
```

4 Bucket Elevator

4.1 Analytical considerations

For functional testing and evaluating the new solution a bucket elevator was modelled and simulated. A bucket elevator is a mechanism for hauling flow able bulk materials (e.g. grain or sand) vertically. For that it is often used in hoppers.

Of decisive importance in analysing and constructing bucket elevators is material deflation. Basically there are two ways of emptying the bucket at the upper turning point. Either the material is thrown out by centrifugal forces or the material falls out in reaction to gravity forces. The second variant should be avoided because there's a high risk that particles fall back into the elevator housing.

To figure out which kind of material deflation is dominating point P is constructed like shown in figure 6.

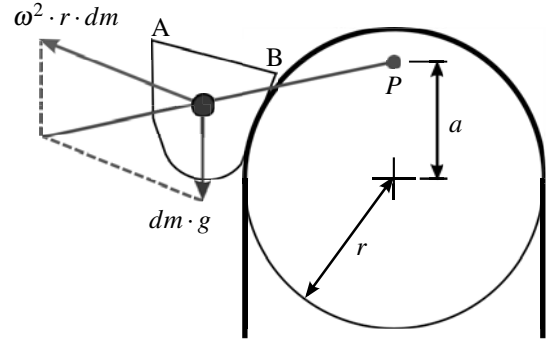


Figure 6: Construction of point P and distance a

Thus particles are thrown out over edge A there must be $a < r$ [5]. For achieving this the minimal rotational velocity can be calculated like in equation 3.

$$\frac{g}{\omega^2} = a; a < r \rightarrow \omega > \sqrt{\frac{g}{r}} \quad (3)$$

What is completely ignored in this calculation is the internal friction of the material and the friction between material and bucket wall. So there's no possibility to get reliable prospective statements if the bucket is emptied the right way. Furthermore it's very difficult to investigate the bucket filling process or dynamic forces acting on the drive chain. That's where the discrete element method can help.

4.2 Modelling

First step to simulate the bucket elevator was to create a simple machine model, including buckets, housing and a simplified drive train. After this was done it was extended by adding DEM specific components. For continuous filling a particle source was inserted at the lower housing aperture. This particle source has an output rate of 35000 particles every second. At the upper outlet a particle sensor was added for measuring the mass and particle flow leaving the bucket elevator. Additionally another particle sensor was added for measuring the number of particles falling back into the housing. The complete model is shown in figure 7

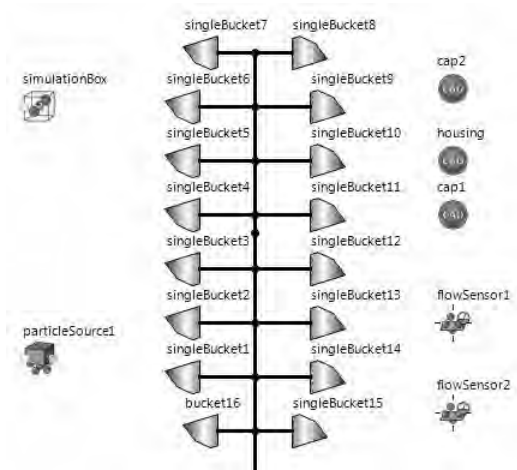


Figure 7: Structure view of the bucket elevator model in Simulation X®

4.3 Results

For evaluation the bucket elevator was simulated for 4.0 seconds. After about 3.0 seconds, a steady state is achieved, at which the number of particles inserted is approximately equal to the number of particles leaving the bucket elevator. At this point of time there are about 65000 particles in the system. Figure 8 shows the 3d-simulation-view at this point of time.



Figure 8: 3d-simulation-view of steady state

It has been found that at a peripheral speed of 4 m/s nearly no particles fall back into the housing. Furthermore the forces acting on all buckets can be measured

and used for optimizing the bucket design and drive chain. The temporal course of forces acting on a single bucket is exemplary shown in figure 9.

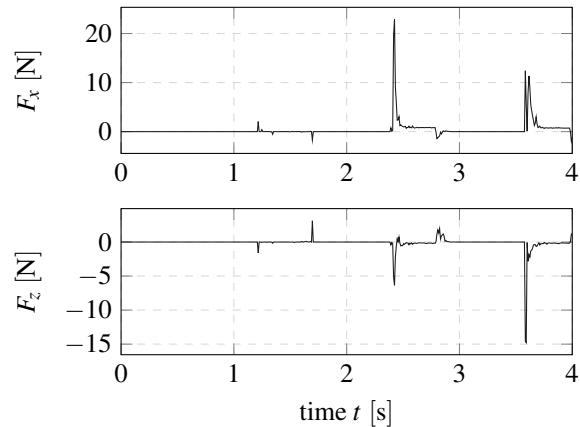


Figure 9: Forces on single bucket

5 Conclusion

In this work, a new concept was presented allowing the closed modelling of machine models and discrete element systems in one simulation tool. For that the command-oriented modelling technique many DEM-applications work with was transferred into an object-oriented approach. Especially for new users which are not familiar with discrete element method but also for old experienced users creating models and coupled simulations is getting very easy.

References

- [1] P. Cundall and O.D.L. Strack. *A discrete numerical model for granular assemblies*
- [2] M. Otter. *Modelica-A Unified Object-Oriented Language for Physical Systems Modeling-Language Specification*, 2000
- [3] M. Geimer, T. Krüger and P. Linsel. *Co-Simulation, gekoppelte Simulation oder Simultorkopplung? Ein Versuch der Begriffsverein-*

heitlichung, O+P Ölhydraulik und Pneumatik,
pp. 572-576, 2006

- [4] G. Kunze, A. Katterfeld, C. Richter, H. Otto and C. Schubert. *Plattform- und softwareunabhängige Simulation der Erdstoff-Maschine Interaktion* In: *5. Fachtagung Baumaschinentechnik*, Dresden, 2012
- [5] W. Noack. *Der Entleerungsvorgang bei Becherwerken*, In: *Agrartechnik*, 1955

Komplexität beim Modellieren und Simulieren: Eine Analyse und ein Plädoyer für schlanke Modelle

Jochen Wittmann¹

¹Hochschule für Technik und Wirtschaft, Studiengang Umweltinformatik, Wilhelminenhofstraße 75A,
D – 12459 Berlin,
wittmann@htw-berlin.de

Der Beitrag geht von der Beobachtung aus, dass die Anforderungen bei der Durchführung von Simulationsstudien bezüglich der Komplexität der behandelten Systeme und Modelle steigt und die Beherrschung der Komplexität sowohl durch softwaretechnische Massnahmen als auch durch eine Verstärkung der Hardware aufwändig und teuer ist. Von diesem Ausgangspunkt wird eine für die Praxis taugliche Definition des Komplexitätsbegriffes versucht. Auf dieser Grundlage werden die Phasen einer Simulationsstudie (Formulierung der Zielstellung, Systemabgrenzung, Modellbildung, Modellexperiment, Modelldatenauswertung) dahingehend analysiert, wo und wodurch Komplexität entsteht und wie sie behandelt wird. Die Analyse zeigt, dass Komplexität häufig dadurch entsteht, dass ein Modellierungsansatz verwendet wird, der nicht passgenau auf die Fragestellung der Modelluntersuchung abgestimmt ist und damit wesentlich aufwändigere Untersuchungen impliziert als notwendig wären. Gerade die Verwendung von agentenbasierten Modellen, deren Anwendung durch die objektorientierten Programmiersprachen vereinfacht ist, werden kritisch bewertet, weil sie eine Vielzahl zu bestimmender Parameterwerte und ein komplexes Experiment-Design für eine präzise Ergebnisinterpretation nach sich ziehen. Fazit ist ein Plädoyer für passgenau an der Fragestellung orientierte Modelle und damit auch für frühzeitig spezifizierte Ziel- und Zweckbestimmung der Simulationsuntersuchung als Ganzes. Auf diese Weise kann Aufwand und Komplexität sozusagen „an der Wurzel“ reduziert und eine effektive und effiziente Durchführung der Simulationsstudie gewährleistet werden.

1 Problemstellung und Definitionsansätze für „Komplexität“

1.1 Komplexe Modelle und Simulationsaufgaben

Das Paradigma der Objektorientierung hat für die Modellierung und Simulationstechnik einen starken Trend hin zur objektorientierten Modellierung dynamischer Systeme und zur agentenbasierten Simulation ausgelöst. Einerseits bieten diese Konzepte eine einfache Übertragung der zu modellierenden realen Systemwelt an, andererseits werden durch entsprechende Modellierungssprachen und Simulationsumgebungen mächtige Werkzeuge zur Verfügung gestellt, die es erlauben, große Zahlen von gleichzeitig agierenden und kommunizierenden bzw. interagierenden Objekten zu beschreiben und zu simulieren.

Die Beschränkungen durch die Hardware greifen erst bei erheblichen Objektzahlen, so dass auch von dieser Seite der Entwicklung wenig entgegensteht.

Allerdings zeigt sich, dass derartige Modelle erhebliche Probleme bei der Parametrisierung, der Validie-

rung und bei der Interpretation der Ergebnisse aufweisen. Die Anwender sprechen von einer Komplexität der Modelle, die nur noch schwer zu durchschauen und zu beherrschen ist und rufen nach Unterstützung, diese Komplexität durch geeignete Massnahmen, sei es auf Ebene der Software, sei es durch die automatische Analyse und Kennzahlenbildung zu reduzieren und somit eine einfachere Schnittstelle für den vielzitierten „Entscheidungssträger“ anzubieten.

Dieses Paper versucht eine Analyse dieses Problemfeldes „Komplexität“ und möchte sehr pragmatische Vorschläge zu deren Reduktion bzw. Beherrschung machen.

1.2 Komplexität in der Informatik

Am Anfang steht eine Übersicht über den Begriff „Komplexität“ selbst, der in den unterschiedlichen Disziplinen vollkommen unterschiedlich gebraucht wird.

In der Informatik beschreibt die Komplexität grob gesagt den Ressourcenverbrauch eines Algorithmus bezogen auf Speicherbedarf und Rechenzeit. Die Kom-

plexität dient dann einer Klassifikation von Algorithmen in Bezug auf ihre Lösbarkeit. Diese Klassifikation wird formalisiert und drückt sich in der sogenannten O-Notation bzw. der NP-Vollständigkeit aus (nach [Wegener 2006]).

Beide Klassifikationen sind für den Modellierer und Simulationsexperten wenig aussagekräftig, beziehen sie sich doch auf die Abarbeitung von Modellen, die durch das Simulationssystem gewährleistet ist. Sicher beeinträchtigen lange Rechenzeiten und enormer Speicherverbrauch die Handhabbarkeit von Modellen, letztlich stellen sie jedoch „nur“ Widerstände auf der Implementierungsebene dar und spiegeln nicht das konzeptionelle Komplexitätsproblem, das in 1.1 problematisiert ist.

1.3 Komplexität in der Spieltheorie

Ein Blick auf die Spieltheorie zeigt eine etwas andere Komplexitätsdefinition: Hier wird im Wesentlichen (z.B. nach [Gibbons 1992]) unterschieden zwischen

- a) der Zustandsraum-Komplexität und
- b) der Entscheidungs-Komplexität

Wobei a) im die Größe und Struktur der möglichen Systemzustände abbildet und damit dem Komplexitätsbegriff der Systemmodellierung intuitive schon recht ähnlich ist, und b) sich auf die Beschreibung der alternativen Entscheidungen und deren Bewertung bezieht.

1.4 Komplexität in der Systemtheorie

Punkt 1.3 a) versucht die Systemtheorie (z.B. bei [Rapoport1988]) zu quantifizieren, indem sie die Komplexität eines Systems als Funktion der Anzahl der System-Komponenten und der Anzahl der Verknüpfungen zwischen diesen Komponenten definiert. Systeme mit vielen, stark vernetzten Komponenten sind also „komplexer“ als Systeme mit wenigen Komponenten und wenigen Verknüpfungen.

Bei genauer Betrachtungsweise kann dieses Komplexitätsmass jedoch auch nicht vollständig überzeugen, da ein entscheidender Punkt quantitativ nicht bewertet wird: die Art der Verknüpfung bzw. deren „Schwierigkeit“. Tatsächlich spricht die Systemtheorie von der „Unüberschaubarkeit“ der Verknüpfungen ([Milling1981]) und meint damit, in welcher Form die Verknüpfung wirksam wird: als Rückkopplung, Linearität oder Nicht-Linearität, Stochastik, usw. So kommt die quantitative Berechnung einer Masszahl für Komplexität an dieser Stelle ebenfalls an ihre Grenzen, indem

sie eine schwierige und dann doch nur wieder qualitative Klassifikation der Verknüpfungsqualität erfordern würde.

1.5 Komplexität in der Entscheidungstheorie nach Nutt

Einen gänzlich anderen, auf den ersten Blick auch für die Simulationstechnik nutzbaren Komplexitätsbegriff findet sich in der Entscheidungstheorie. Nutt ([Nutt1998]) erweitert die rein technische Dimension des Problems um eine politische Ebene, indem für jede Entscheidung einer „procedural difficulty“ und eine „political difficulty“ einführt.

In der Dimension 1, der „procedural difficulty“ stützt er sich auf die vorgenannten Definitionen und berechnet die Komplexität als Funktion der Parameter

- Anzahl Komponenten
- Anzahl Alternativen
- und der Existenz eines geeigneten Bewertungsmechanismus.

Bis hierher fasst er also im Wesentlichen die Bestimmungstücke aus Systemtheorie und Spieltheorie zusammen, ohne allerdings auf das Problem der Quantifizierbarkeit einzugehen, das selbstverständlich weiterbesteht und zwar nicht nur für die „Unüberschaubarkeit“ der Beziehungen, sondern jetzt auch noch für die Quantifizierung der Eignung des Bewertungsmechanismus.

Wesentlich neuer Aspekt ist aber seine 2. Dimension, in der er die „political difficulty“ als Widerstand gegen eine Entscheidung einführt. Obwohl auch diese Dimension nur qualitativ bewertbar ist, lassen sich durch eine Kombination von Dimension 1 und 2 vier Typen von Entscheidungen unterscheiden, die auf die Situation im Bereich der Simulationstechnik gut übertragbar erscheinen:

1. Routine-Entscheidungen sind prozedural einfach und politisch akzeptiert
2. komplizierte Entscheidungen sind prozedural kompliziert und politisch akzeptiert
3. entmutigende Entscheidungen sind prozedural einfach und politisch nicht akzeptiert
4. blockierte Entscheidungen sind prozedural kompliziert und politisch nicht akzeptiert

2 Prozessschritte bei der Entscheidungsfindung mit Modellen

Mit diesen Definitionen im Hinterkopf wollen wir nun die Entstehung von Komplexität beim Prozess der Modellbildung und Simulation näher untersuchen. Wir beziehen uns dabei auf die Prozessübersicht, wie sie z.B. bei Schmidt [Schmidt1985] dargestellt ist.

Am Anfang steht eine Fragestellung oder Zielstellung für die gesamte Simulationsuntersuchung. Auf deren Basis wird ein zu untersuchendes System abgegrenzt und im Schritt der Modellbildung ein analoges Ersatzsystem, das Modell, gebildet. Experimente an diesem Modell führen zu Simulationsdaten, die analysiert werden und die Grundlage für eine Entscheidung in Beantwortung der Fragestellung bilden.

3 Wodurch entsteht Komplexität?

Die in Abschnitt 2 genannten Arbeitsschritte sollen nun einzeln auf ihren Einfluss auf die Komplexität des gesamten Simulationsvorhabens analysiert werden. Dabei werden erstens typische Entwurfsentscheidungen herausgearbeitet und dahingehend untersucht, wie sie auf die Komplexität des Gesamtprojektes wirken, eine Vorgehensweise, die zweitens Anhaltspunkte liefern kann, wie umgekehrt die Komplexität verhindert oder zumindest reduziert werden kann, wenn in jedem Schritt der Prozesskette Komplexitätsüberlegungen in das Design einfließen.

3.1 Komplexität durch komplexe Fragestellungen

Am wenigsten diskutiert und in Frage gestellt wird häufig die erste Entscheidung im Rahmen einer Simulationsstudie, nämlich die Frage nach der exakten Zielstellung und den erwarteten Ergebnissen. Für eine spätere Validierung unerlässlich muss hier das zu untersuchende System eingegrenzt und Angaben über Genauigkeit und zeitliches Intervall für die Ergebnisse spezifiziert werden.

Es ist eine Selbstverständlichkeit, dass die Komplexität einer Modelluntersuchung steigt, wenn:

- der Allgemeinheitsgrad des Modells hoch sein soll,
- eine hohe Genauigkeit für die Werte der Modellgrößen verlangt wird,
- ein großer zeitlicher Horizont verlangt wird

- zusätzlich zur Systemdynamik eine Bewertung und/oder Optimierung eingeschlossen ist.

Andererseits ist die Spezifikation der Fragestellung die einfachste Art, Komplexität zu reduzieren, wenn man sich zur Lösung eines Problems von Anfang an mit weniger zufrieden geben kann. Daher sollte unbedingt mit dem Anwender geklärt werden, welche Aussagen aus der Modelluntersuchung notwendigerweise abgeleitet werden müssen, um ein passgenaues Modell zu erstellen. Umgekehrt sollte der Anwender für die Korrelation zwischen seinen Anforderungen und der Komplexität (und letztlich dem Preis) der Studie sensibilisiert werden.

Grundsätzlich sollte aber am Ende eine Vereinbarung stehen, die sich ganz konkret auf Werte von Modellgrößen mit exakter Angabe von Scale und Validitätsintervall bezieht.

3.2 Komplexität durch komplexes Datenmaterial

Während die Komplexität der Fragestellung wesentlich durch den Auftraggeber einer Simulationsstudie beeinflusst ist, ist man bei der Modellbildung häufig auf bereits erfasstes Datenmaterial angewiesen, dessen Art und Umfang technischen wie organisatorischen Nebenbedingungen unterliegt.

Gerade im Bereich der Modelle mit Geobezug arbeitet man häufig mit Satellitendaten, die Raum- und Zeitbezug aufweisen und meist in hoher Auflösung vorliegen. Hier ist die für das Modell verlangte Auflösung häufig sogar geringer als die Auflösung der Messdaten. Eine frühzeitige Vorverarbeitung und Datenreduktion kann für diesen Fall eine erhebliche Komplexitätsreduktion für die Modellstudie bewirken.

Umgekehrt verhält es sich, wenn die gewünschte Genauigkeit nur durch komplexe Versuchsanordnungen und hochempfindliche Messinstrumente zu erreichen ist. Hier kann nur hinterfragt werden, ob tatsächlich mit der spezifizierten Genauigkeit gemessen werden muss, oder ob zu Gunsten des Kosten-Nutzen-Verhältnisses auf hohe Genauigkeitsanforderungen verzichtet werden sollte.

Im dritten Fall ist die zu erreichende Genauigkeit durch Messeinrichtungen vorgegeben. In diesem Fall ist eine Untersuchung des Gesamtmodells notwendig, damit nicht an anderen Stellen mit wesentlich größerer Genauigkeit gemessen, modelliert und simuliert wird, die Genauigkeit der Endergebnisse jedoch durch die

geringste Genauigkeit in einem Modul dominiert bleibt. In diesem Fall lohnt die Überlegung, auch das Gesamtmodell mit der geringeren Genauigkeit zu konzipieren und zu simulieren, und dadurch die Gesamtkomplexität zu reduzieren, ohne die Aussagekraft zu beeinträchtigen.

3.3 Komplexität durch komplexe Modelle

Dies ist der Punkt, an den man wohl zuerst denken wird, wenn es um Komplexität in der Modellierung geht. Hier steckt die Hauptschwierigkeit nach Meinung des Autors darin, sich nicht durch eine Vielzahl von Features und Design-Möglichkeiten moderner Modellierungs- und Simulationssysteme vom ursprünglichen Ziel der Studie abbringen zu lassen.

So verleiten gerade moderne Umgebungen durch modular-hierarchischen Modellaufbau (z.B. basierend auf den grundlegenden Ideen von [Zeigler1990]) sehr einfach und schnell ein komplexes Modell zu erstellen, das inhaltlich wesentlich mehr umfasst, als es bezüglich der Fragestellung bedarf. Eine Erweiterung sowohl in die (gemessen an der Zielstellung) nicht notwendige Tiefe und Breite ist durch das Baukastensystem spielerisch einfach und ein detaillierteres Modell verspricht ja auch mehr Genauigkeit. Allerdings immer zu Lasten der gestiegenen Komplexität und des damit verbundenen Mehraufwandes, zusätzliche Komponenten, Modellgrößen und Parameter bedienen zu müssen.

Verstärkt wird dieser Effekt noch durch das Paradigma der objektorientiert aufgebauten, bzw. der agentenbasierten Modelle (zu Definition und Unterscheidung z.B. [Ortmann1999]). Bei der Verwendung dieser Ansätze ist es sehr leicht möglich, sehr große Mengen von Objekten zu generieren, die dann jedoch einzeln parametrisiert und in ihrem Zeitverhalten beobachtet und analysiert werden müssen, was methodisch und bezogen auf das Datenvolumen einen erheblichen Mehraufwand zum Beispiel gegenüber den klassischen Differentialgleichungsmodellen darstellt.

In beiden Fällen ist die Modellierung durch moderne Werkzeuge derart vereinfacht worden, dass ohne großen Aufwand komplexe Modelle erstellt werden können, die sich dann aber in ihrer Behandlung zur Laufzeit und bei der Experimentauswertung als äußerst zeitintensiv und methodisch anspruchsvoll erweisen. Man denke diesbezüglich etwa an eine Parallelisierung des Laufzeitsystems, die Daten-Aggregation aus den ein-

zelnen Objekten/Agenten zu aussagekräftigen Kennzahlen, die notwendige statistische Interpretation der Einzelergebnisse, usw.

Hier wird durch die Simulationssoftware ein Komfortniveau bei der Modellierung geboten, das in den folgenden Phasen der Simulationsstudie nicht gehalten werden kann.

3.4 Komplexität durch komplexe Experimente

Jeder Simulationslauf gibt immer nur Information über eine einzige Trajektorie ausgehend vom gegebenen Anfangszustand wieder. Folglich müssen immer eine Vielzahl von Simulationsläufen mit veränderten Anfangsbelegungen durchgeführt werden, um eine statistische Aussage über die Ergebnisse treffen zu können.

Je größer der Zustandsraum des Modells und je größer der Parameterraum des Modells ist, umso größer wird die Zahl der notwendigen Simulationsläufe ausfallen. Berücksichtigt man darüber hinaus, dass die Parameter- und Anfangsbelegungen für die einzelnen Größen nicht unabhängig voneinander in Bezug auf die zu untersuchende Systemdynamik sind, so ergeben sich bereits bei nur wenigen Parametern komplexe Experiment-Designs, die lange Rechenzeiten und eine komplexe Auswertung und Interpretation nach sich ziehen. Die Komplexität der Experimente steigert sich selbstverständlich noch um ein Vielfaches, wenn es sich nicht ausschließlich um deterministische Größen, sondern um stochastische handelt.

Berücksichtigt man darüber hinaus noch, dass Simulationsexperimente in der Regel nicht einen einzelnen Standardlauf zum Ziel haben, sondern meist der Sensitivitätsanalyse oder der Systemoptimierung dienen, so wird klar, dass sauber belegte Experimentergebnisse ohne systematische Hilfsmittel zum Experimentdesign und zur Datenanalyse kaum zu ermitteln sind.

3.5 Komplexität durch komplexe Datenanalyse

Hinzu kommt, dass die reinen Simulationsergebnisse meist noch einer Transformation bedürfen, um sinnvoll interpretiert werden zu können. Gerade bei agentenbasierten Modellen müssen zum Beispiel Aggregationen über die Menge der Agenten vorgenommen werden, um statistische Aussagen über die Dynamik des Gesamtsystems abzuleiten. Ähnlich ist die Vorgehensweise, wenn komplexes Systemverhalten auf eine oder mehrere Kennzahlen oder Indikatoren abgebildet werden soll. Das vereinfacht die Interpretation, allerdings steigt die Komplexität, weil natürlich die Funk-

tion zur Kennwertbildung wiederum Teil des Gesamtmodells wird, dieses vergrößert und selbstverständlich erneut zumindest eine Sensitivitätsanalyse für den Kennwert erfordert.

Die hier beschriebene Tatsache, dass mit der Größe des Parameterraumes und mit dem Grad der Vernetzung der Modellgrößen auch die Interpretation und Wertung der Simulationsergebnisse schwieriger wird und die Komplexität der Datenanalyse wesentlich erhöht, ist an dieser Stelle des Prozessmodells eine Selbstverständlichkeit. Diese Tatsache sollte jedoch bereits auch bei den vorausgehenden Arbeitsschritten bedacht werden, und insbesondere sollte der Arbeitsaufwand, der bei der Datenanalyse notwendig wird schon früh in Relation zum Gewinn durch die detailliertere, aber (wegen der Modularität) einfachere Modellbildung gesetzt werden.

Bleibt die Aufgabe, die Ergebnisse am Ende zu bewerten. Parametrisierung, Optimierung und Bewertung sollten im Idealfall sauber modularisierte Schritte der Simulationsstudie darstellen. In der Praxis gehen diese Aufgaben jedoch ineinander über, gerade wenn bei komplexen Aufgabenstellungen eine effiziente Implementierung und Experimentdurchführung im Vordergrund steht. Wenn es jedoch gelingt, diese Arbeitsschritte weitestgehend zu entkoppeln, kann man die Anzahl der Interdependenzen und damit die Komplexität des Experimental Designs erheblich reduzieren. Besondere Rückwirkung hat eine Reduktion der Verknüpfungen in dieser Phase einer Studie, weil ja jede Änderung in der Bewertung in der Regel eine neue Optimierung mit einer Vielzahl von statistisch erneut abzusichernden Simulationsläufen mit sich bringt. Eine Änderung in dieser Phase potenziert in der Regel den Aufwand durch ihre Rückwirkung auf die vorausgehenden Phasen und ist bezüglich der Komplexitätsbetrachtung damit besonders relevant.

4 Beobachtungen und Bemerkungen zu den Quellen der Komplexität

Nach der Analyse des Modellbildungs- und Simulationsprozesses sollen einige Beobachtungen stehen, die sich kritisch und auch bewusst provokativ mit der Entstehung der Komplexität von Modellen auseinandersetzen.

4.1 Komplexität der Realität impliziert Komplexität des Modells?

Modelle werden als vereinfachte Analogsysteme der Wirklichkeit gemacht. Modelle haben eine Zweckbestimmung. Muss man tatsächlich die Komplexität des Systems immer vollständig verstehen und abbilden, um den Modellzweck zu erreichen? Oder andersherum: Kann ein Modell mit einer geschickt eingeschränkten Zweckspezifikation seinen Zweck nicht auch erreichen, ohne die Komplexität des Realsystems nachbilden zu müssen?

4.2 Isomorphe Modelle per se besser als homomorphe?

Eine typische Entscheidung ist dabei, ob man ein strukturgleiches (isomorphes) Modell einem strukturähnlichen (homomorphem) Modell in jedem Fall vorziehen sollte. Selbstverständlich bietet das isomorphe Modell einen höheren Grad an Detaillierung und Anschauung, allerdings in der Regel auch eine größere Anzahl Modellkomponenten und Verknüpfungen, folglich eine erhöhte Komplexität. Das homomorphe Modell ist in der Regel weniger komplex, dafür aggregierter und/oder abstrakter.

Konkret auf den aktuellen Trend hin zu agentenbasierten Modellen bedeutet dies: Mit Sicherheit bildet die Verwendung des agentenbasierten Paradigmas die Realität strukturgleich ab und macht damit die Diskussion mit den Fachexperten und die Modellspezifikation anschaulicher. Im Gegensatz dazu erhöht die Nutzung dieses Paradigmas allerdings die Komplexität bezüglich der Modellstruktur (Anzahl Modellkomponenten und Beziehungen), der Parametrisierung, des Laufzeitverhaltens sowie der Interpretation und Analyse der Simulationsergebnisse.

Eine Abwägung ist in jedem Einzelfall zu treffen, die Aussage, dass ein Modell umso "besser" ist, je strukturähnlicher es der Realität kommt, erscheint mir in dieser Allgemeinheit nicht gegeben.

4.3 Daten in der höchstmöglichen Auflösung notwendig?

Die analoge Diskussion kann man auf der Ebene der zeitlichen Auflösung und der Skalen für die Modellgrößen führen: Selbstverständlich ist es gut, aktuelles Wissen und aktuelles Datenmaterial bei der Modellierung zu berücksichtigen. Es kann aber der Fall vorliegen, dass der aktuelle State of the Art in einem Forschungsbereich einen solchen Präzisionsgrad erreicht hat, dass dieser den Arbeitsschritt der Modellierung

durch zu hohe Präzision und die enormen damit verbundenen Datenmengen eher behindert als befördert, und -wie im Abschnitt 3.2 ausgeführt- eine nicht notwendige Komplexität des Modells generiert.

4.4 Sind massgeschneiderte Modelle weniger komplex als modulare Modelle mit wiederverwendbaren Komponenten?

Eine weitere Überlegung betrifft den Trend hin zu Modellbanken und komponentenorientierten Modellbaukästen: Dort werden parametrisierbare Baiskomponenten gesammelt, die modular-hierarchisch je nach Gegebenheit zu einem Gesamtmodell zusammengefügt werden können. Vorteil ist die Wiederverwendbarkeit der Komponenten und der transparente, strukturierte Modellaufbau.

Aus Sicht der Komplexitätsreduktion erzeugt diese Vorgehensweise jedoch ein Mehr an Parametern (durch die Parametrisierbarkeit und Anpassbarkeit) und Beziehungen (wegen der flexiblen Modellstruktur) und damit ein Mehr an Komplexität. Im Gegensatz dazu stünde eine flache Modellhierarchie ohne interne Komponentenstruktur und ausschliesslich mit den Modellgrößen, die für die jeweilige Zielstellung nötig wären.

Besteht tatsächlich ein solcher Zielkonflikt zwischen der Wiederverwendbarkeit von Modellkomponenten und der Zielgenauigkeit der Modellierung? Hier sollten tiefergehende Untersuchungen für eine präzise Diskussion dieser Hypothese angestossen werden.

4.5 Design-Entscheidungen in frühen Phasen des Prozessmodells potenzieren den Aufwand in späteren Phasen

Betrachtet man den Aufwand einer Designentscheidung im Verlauf des dargestellten Prozessmodells, so kann man leicht einsehen, dass eine einzige zu Beginn des Prozesses zusätzlich eingeführte Messgröße/Parameter im weiteren Verlauf den Aufwand nichtlinear ansteigen lässt: Müssen doch für diese Größe Sensitivitätsuntersuchungen, statistische Untersuchungen, und letztlich eine Untersuchung auf Interdependenzen zu sämtlichen anderen Größen des Modells analysiert werden, was eine Vielzahl von Simulationsläufen und ein sehr ausgefeiltes Experimental Design nach sich zieht und damit die Komplexität der Gesamtaufgabe drastisch erhöht.

Auch eine nur geringe Ausweitung der Fragestellung kann eine derartige Komplexitätskaskade in den darauffolgenden Prozessschritten auslösen: Man denke

zum Beispiel an die einfache Frage, ob ein Grenz-Wert erreicht wird, oder die allgemeinere und komplexere Frage nach der statistischen Verteilung des Wertes.

5 Mögliche Folgerungen für die Praxis des Modellierens und Simulierens

Wesentliche Zielrichtung dieser Überlegungen ist die Frage, ob nicht ein erheblicher Teil der Komplexität beim Modellieren und Simulieren durch mehr oder weniger gedankenloses „fast forward“-Denken erzeugt wird, das sich darauf verlässt, immer mächtigere Informatikwerkzeuge an die Hand zu bekommen, mit denen die komfortabel generierte Komplexität auch komfortabel beherrscht werden kann. Allein die Werkzeuge geraten an ihre Grenzen und der Aufwand bei der Bearbeitung hochkomplexer Modelle steigt derart, dass auch die Kosten und für eine Simulationsstudie an die Grenze des für die Anwender erträglichen stoßen und somit die Methode Simulation insgesamt in einer Kosten-Nutzen-Abwägung schlecht abschneiden wird.

Als Konsequenz all dieser Überlegungen möchte der Autor die folgenden Thesen bzw. Fragen aufstellen, um die Diskussion strukturieren und weiterzuführen:

1. Besteht tatsächlich ein Widerspruch zwischen der Zielgenauigkeit der Modellierung und der Wiederverwendbarkeit von Modellkomponenten aus baukastenartigen Modellbanken?
2. Trägt eine zielorientierte Dokumentation der Modelluntersuchung zur Reduzierung von Komplexität bei?
3. Können ausgehend von präzisen Experimentzielspezifikationen in Umkehrung des Prozessmodells schlanke und effective Modelle und Experimente entwickelt werden?
4. Wie könnte man sich eine Software vorstellen, die eine derartige Vorgehensweise unterstützt? Können Konzepte der Datenbankanfragesprachen und/oder der Künstlichen Intelligenz sinnvoll auf Simulationsumgebungen übertragen werden?

6 Fazit

Der vorliegende Artikel erhebt nicht den Anspruch, einer mathematisch exakten Definition von Komplexität bei Simulationsuntersuchungen gerecht zu werden. Er versucht allerdings, die Gründe für die Entstehung der Komplexität bei Simulationsstudien an Hand einer Analyse der einzelnen Prozessschritte herauszuarbei-

ten und damit den Weg einzuschlagen, der Komplexität nicht durch noch komplexere Softwaresysteme beherrschbar machen möchte, sondern der vielmehr bei der Entstehung der Komplexität hinterfragt, ob diese genau an dieser Stelle zur Erreichung des Simulationsstudienziels unabdingbar ist.

7 References

- [Gibbons 1992] Robert Gibbons: A primer in game theory. Harvester Wheatsheaf, New York 1992
- [Milling1981] Milling, P.: Systemtheoretische Grundlagen zur Planung der Unternehmenspolitik. Berlin: Duncker & Humblot, 1981
- [Nutt1998] Nutt, P.C., Evaluating Complex Strategic Choices, Management Science, 44 (8), 1998, 1148-1166
- [Ortm99] Ortmann, Jörg: Ein allgemeiner individuenorientierter Ansatz zur Modellierung von Populationsdynamiken in Ökosystemen unter Einbeziehung der Mikro- und Makroebene; Dissertation am Fachbereich Informatik, Universität Rostock, 1999
- [Rapoport1988] Rapoport, A.: Allgemeine Systemtheorie. Wesentliche Begriffe und Anwendungen. Verlag Darmstädter Blätter, 1988
- [Schmidt1985] Schmidt, B.: Systemanalyse und Modellaufbau, Berlin, 1985
- [Wegener 2006] Wegener, Ingo: Komplexität; in: Rechenberg, Peter, Pomberger, Gustav: Informatik Handbuch, Hanser Verlag, München, Wien, 2006, S.119 ff
- [Zeigler1990] Zeigler, B.P.: Object-Oriented Simulation with Hierarchical, Modular Models, Boston, San Diego, 1990

Eine Machbarkeitsstudie zur modellbasierten Beurteilung stadtklimatischer Indikatoren auf der Basis von Bebauungsplänen im GIS

Marie Francoise Aimée Kokam Tchabop, Regine Grafe, Jochen Wittmann

¹Hochschule für Technik und Wirtschaft, Studiengang Umweltinformatik, Wilhelminenhofstraße 75A,
D – 12459 Berlin,

wittmann@htw-berlin.de; grafe@htw-berlin.de

Für eine Erstbewertung von Klimatopveränderungen in Folge von Bebauung kann die Darstellung mit Hilfe von ArcMap verwendet werden. Unter Einbindung vorhandener Fachinformationssysteme zur Bodenversiegelung, des bodennahen Luftaustausches und der bioklimatischen Verhältnisse im Umfeld kann der Einfluss städtebaulicher Planungsvorhaben auf das lokale Klima visualisiert und prognostiziert werden. Da im Planzustand keine konkreten Bebauungsstrukturen festgelegt, sondern u.a. lediglich der Versiegelungsgrad und die maximale Höhe der Bebauung vorgegeben sind. Unter Zuhilfenahme von ArcMap aus der Familie ArcGIS kann eine Visualisierung von ausgewählten Prüfkriterien, wie z.B. Tag-Nacht-Temperaturen und lokale Windströmung im Zusammenhang mit der Anordnung und der Höhe der geplanten Gebäude erfolgen. Mit der von ArcMap vorgegebenen Farbkodierung werden Baukörper als Polygone nach Höhe und Anordnung im Windfeld selektiert. Eine Überlagerung von Luftaustausch- und Bioklimakarten auf die Grundkarte des Geländes ermöglicht die Ableitung und Darstellung einer optimierten Geometrie der Bebauungsstruktur im Hinblick auf deren Wirkung auf das Klimatop. Damit kann die Realisierung eines Planvorhabens in neuer Qualität in die Diskussion mit Investoren und mit der Bürgerschaft geführt werden. Diese Art der Erstbewertung ersetzt jedoch nicht die detaillierten Prognosen, die mit Simulationsmodellen erreicht werden können. Klimagerechtes Bauen erfordert zukünftig neben leistungsfähigen Fachinformationssystemen auch maßgeschneiderte Simulationstools für die planenden Behörden.

1 Einleitung und Problemstellung

Zunehmende Diskussionen zum anthropogenen Klimawandel umfassen auch die städtebaulichen Planungsvorhaben im Zusammenhang mit Bebauungsstrukturen und deren stadtklimatische Wirkungen. Da Umweltbelastungen von erheblicher Gesundheitsrelevanz sind, wird aktuell das Bauen unter dem Gesichtspunkt human-biometeorologische bzw. als bioklimatische Beeinflussung diskutiert. Dieser Ansatz implementiert auch die Notwendigkeit, den Fokus auf human-biometeorologische Aspekte in der Bebauungsplanung zu legen. Gängige Verfahren, die für städtebauliche Planungen verpflichtend sind, sind die Bauleitplanung mit dem Flächennutzungsplan und darauf basierend die Bebauungspläne der Kommunen. Die besondere Bedeutung der umweltrelevanten Aspekte in der Bauleitplanung werden im Baugesetzbuch (BauGB) deutlich, da hier die Erstellung einer Umweltprüfung erforderlich ist und die Ergebnisse in einem sog. Umweltbericht als gesonderter Teil der Begründung zum Bebauungsplan zu erstellen ist. Das geplante Vorhaben ist im Hinblick auf die umweltrelevanten Rechtsgrundlagen, wie die Naturschutz-, Immissionsschutz-, Bodenschutz- und Wasserhaus-

haltungsgesetzgebung etc. zu prüfen. Mit der integrativen Nutzung von Fachinformationssystemen (FIS) im Rahmen eines Bebauungsplanes, die in den Kommunen in unterschiedlicher Form, Qualität und Handhabung vorgehalten werden, besteht auch die Möglichkeit, städtische Klimatope und deren human-biometeorologische Bedeutung stärker in ein klimagerechtes Bauen einzubeziehen. Die Dringlichkeit, Fachinformationssysteme als effiziente Planungsinstrumentarien mit aktuellen Datensätzen vorzuhalten und nutzbar zu machen, zeigt sich u.a. in den aktuellen Anstrengungen, die von Bund und Ländern unternommen werden (Vgl. Bimüller, 2009) [1]. Simulationsmodelle wie FITNAH und KALM sind derzeit die am häufigsten eingesetzten Werkzeuge für Bewertungen der human-biometeorologischen Wirkungen von Bebauungsstrukturen. Allerdings werden diese häufig aus Kostengründen nicht verwendet. Für eine Erstbewertung von Klimatopveränderungen durch Bebauung kann jedoch auch ArcGIS verwendet werden.

1.1 Der Bebauungsplan als komplexes Basisdokument für städtebauliche Planungen

unterscheiden sich die Handlungsfelder grundsätzlich zwischen bestehendem Planungs- und Baurecht sowie der Schaffung neuen Planungs- und Baurechtes durch Bebauungspläne. Während der gestalterische Spielraum der Kommune im bestehenden Recht eher im Bereich beantragter Abweichungen und Befreiungen liegt, lässt der Gesetzgeber den Gemeinden bei der Schaffung neuen Planungsrechts viele Möglichkeiten der Einflussnahme im Sinne eines übergeordneten Interessenausgleichs zum Wohle der Allgemeinheit zu. Auf gesamtstädtischer Ebene finden sich diese Betrachtungen in den sektoralen Stadtentwicklungsplänen der Kommunen wieder, die ebenfalls Abwägungsgrundlage in den Bebauungsplänen sind. Hier erfolgt auch neben anderen Schutzgütern eine Bewertung des Schutzgutes Gesundheit der Bevölkerung. Besonderes Augenmerk liegt dabei auf den Themen Lärm- oder Licht-, Wärme- sowie Schadstoffimmissionen, die die human-biometeorologische Bedingungen sowie die Erholungsfunktionen behandeln. Die aus der Prüfung und Bewertung resultierenden Regelungen werden durch Festsetzungen der maßgeblichen Anforderungen auf der Basis von verschiedenen umweltrechtlichen Grundlagen gesichert. In der Abbildung 1 sind die maßgeblich beteiligten Fachressorts und deren Gesetzesauftrag ersichtlich, deren Prüfergebnisse in den vom BauGB geforderten Umweltbericht einfließen.



Abb.1 Vereinfachte Darstellung - Gebäudebeziehung

1.2 Auswahlkriterien für stadtklimatische Wirkungen

Die Auswahl der Prüfkriterien für einen aufzustellenden Bebauungsplan ist von außerordentlicher Bedeutung, weil das Verfahren selbst von einer Vielzahl von

Rahmenbedingungen und Parametern abhängig ist.. Jeder Bebauungsplan erfordert den Abgleich aller Kriterien. Ist ein Bebauungsplan aus der Sicht möglicher stadt-klimatischer Beeinflussung zu bewerten, sind sowohl die Schutzgüter Luft und Klima, Boden, Mensch nach Folge nicht nach Alphabet! sowie die stadtökologische Prägungen zu beachten. Wie sehr eine Kommune oder ein bestimmtes Stadtgebiet in Folge der Veränderung von Baustrukturen durch ein Planvorhaben stadtklimatisch betroffen ist, ist von einem komplexen System von mehreren verknüpften Faktoren unterschiedlicher kommunaler Zuständigkeiten abhängig. Die jeweilige Betroffenheit kann mit Hilfe einer Gefährdungspotenzial-analyse ermittelt werden (Vgl. Stock, 2005)[2], die die Wirksamkeit der lokalen Klimaänderung im Hinblick auf Fehlfunktionen im Luftaustausch, die bioklimatische Belastung der Bevölkerung und damit mögliche Gefährdungen der Gesundheit aufzeigt. Ist ein Bebauungsplan aus der Sicht möglicher stadtklimatischer Beeinflussung zu bewerten, sind sowohl die Schutzgüter Luft und Klima, Boden, Mensch sowie die stadtökologische Prägungen zu beachten. Wie sehr eine Kommune oder ein bestimmtes Stadtgebiet in Folge der Veränderung Veränderung von Baustrukturen durch ein Planvorhaben stadtklimatisch betroffen ist, ist von einem komplexen System von mehreren verknüpften Faktoren unterschiedlicher kommunaler Zuständigkeiten abhängig. Die jeweilige Betroffenheit kann mit Hilfe einer Gefährdungspotenzial-analyse ermittelt werden (Vgl. Stock, 2005) [2], die die Wirksamkeit der lokalen Klimaänderung im Hinblick auf Fehlfunktionen im Luftaustausch, die bioklimatische Belastung der Bevölkerung und damit mögliche Gefährdungen der Gesundheit aufzeigt. Ist ein Bebauungsplan aus der Sicht möglicher stadtklimatischer Beeinflussung zu bewerten, sind sowohl die Schutzgüter Luft und Klima, Boden, Mensch sowie die stadtökologische Prägungen zu beachten. Wie sehr eine Kommune oder ein bestimmtes Stadtgebiet in Folge der Veränderung von Baustrukturen durch ein Planvorhaben stadtklimatisch betroffen ist, ist von einem komplexen System von mehreren verknüpften Faktoren unterschiedlicher kommunaler Zuständigkeiten abhängig. Die jeweilige Betroffenheit kann mit Hilfe einer Gefährdungspotenzial-analyse ermittelt werden (Vgl. Stock, 2005)[2], die die Wirksamkeit der lokalen Klimaänderung im Hinblick auf Fehlfunktionen im Luftaustausch, die bioklimatische Belastung der Bevölkerung und damit mögliche Gefährdungen der Gesundheit aufzeigt. In aller Regel

angeordnet, kommt es zu einer Windbeschleunigung (Vgl. Stützel, 2007) [6]. Da im vorliegenden Fallbeispiel keine detaillierte Bebauungsstruktur zum Zeitpunkt der Planung vorliegt und auch nicht vorliegen kann, werden verschiedene Varianten gewählt, um eine Beurteilung der Beeinflussung des Luftaustausches durch die geplante Änderung der Nutzungs- und Bebauungsstruktur ableiten zu können. Die für die Diskussion über den Luftaustausch notwendigen Bebauungsstrukturen sind exemplarisch ausgewählt worden. Das betrifft sowohl die ausgewählten Gebäudehöhen als auch deren Ausrichtung an der Hauptwindrichtung des Windfeldes und die Breite der Abstandsflächen zwischen den Gebäuden. In einem ersten Schritt wurden die für eine Bebauung vorgesehenen Flächen mit großräumigen Gebäudegrundflächen, nicht flächenmäßig angegebenen Durchwegungen und unterschiedlichen Gebäudehöhen eingetragen. Die gewählten Höhenvarianten ergeben sich aus den im Bebauungsplan festgesetzten Bebauungshöhen, den zur Verfügung gestellten Überbauungsflächen und den im Plangebiet bekannten Hauptwindrichtungen. Bei den gewählten Varianten, sind die Hauptwindrichtung West-Ost sowie die Windeinfallschneise Nord-Süd und Gebäudehöhen von kleiner 15m oder zwischen 15m und 20m in unterschiedlicher Ausrichtung zu den Hauptwindeinfallschneisen ausgewählt. Mit Hilfe von selektierten Höhen und verschiedenen, selbst ausgewählten Anordnungen von Gebäudekomplexen im Verhältnis zu den hauptsächlich vorherrschenden Windrichtungen ist deren Einfluss auf die Luftaustauschprozesse des Planzustandes visualisiert worden. Werden hohe Gebäude, hier Höhen mit >15m in der Hauptwindrichtung angelegt, kommt es zwangsläufig zu einer Windkanalisierung. Da es derzeit weder im BauGB noch im Bebauungsplan verbindlich festgelegte Maße für Abstandsflächen für Gebäude zueinander gibt, die mögliche Düseneffekte verhindern, wurden für eine Variantendiskussion zum Einfluss von Abstandsflächen auf den bodennahen Luftaustausch keine Abstandsflächen quantifiziert. Um den bodennahen Luftaustausch im Plangebiet entsprechend der Datenlage im FIS Broker bewerten zu können, wurde die Luftaustauschkarte aus dem FIS Broker, deren Daten auf den Ergebnissen von Messstellen des Deutschen Wetterdienstes beruhen, verwendet. Durch Überlagerung der Grundkarte des Plangebietes mit dem Luftaustausch-Feature aus dem FIS Broker ist es möglich, Zonen für das Auftreten von Düseneffekten darzustellen. Die im Feature enthaltene Pfeilsignatur ermöglicht es, den

Einfluss der Hauptwindrichtung und zufließende Kaltluft aus den umliegenden aktiven Kaltluftschneisen verbal zu bewerten. Um die Auswirkung auf die bioklimatische Verhältnisse des östlich vom Plangebiet liegenden Wohngebietes mit einer hohen Bevölkerungsdichte einschätzen zu können, wird die globale Bioklimakarte aus dem FIS Broker zur Beurteilung herangezogen. Für die human-bioklimatische Bewertung des Einflusses des Planvorhabens ist der Bewertungsindex Predicted Mean Vote (PMV) als Grundlage verwendet worden (Vgl. GEO-NET, 2014) [7]. Der PMV dient der Einschätzung von Veränderungen in städtischen Klimatopen. Er ist prinzipiell in stadtklimatisch relevante Planungsprozesse einzubeziehen (Vgl. PMV-Richtlinien 3787) [8]. Durch Überlagerung der Bioklimakarte aus dem FIS Broker auf das Planfeld kann anhand des Farbverlaufes, hier von Grün bis Rot, der PMV zugeordnet werden. Die aus dem FIS Broker verwendete Bioklimakarte basiert auf Messungen in 2m Höhe über Grundboden zum Zeitpunkt 04:00 Uhr MEZ während einer austauscharmen sommerlichen Strahlungsnacht (Vgl. FIS Broker, 2005). Für das untersuchte Umfeld des Plangebietes können so PMV Werte zwischen -1,5 bis 2 abgeleitet werden, was bedeutet, dass eine mittlere negative Beeinflussung stattfinden kann. In der Abbildung 7 wird eine optimierte Bebauungsstruktur unter Berücksichtigung von Bebauungshöhen, Kaltluftzustrom, nicht quantifizierten Abstandsflächen zur Vermeidung von Düseneffekten und von Wärmebelastungen entsprechend PMV des östlich gelegenen Wohngebietes dargestellt.

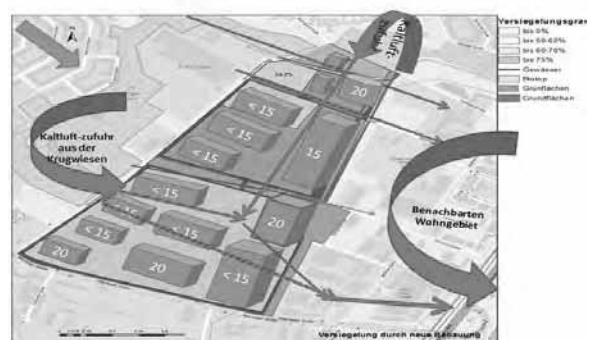


Abb. 4 Optimierte Bebauungsstruktur auf dem Planfeld [4]

2 Fazit und Forderungen

Identifikation und Qualifizierung der Wirkung eines Bebauungsplanes stellen ein komplexes Beziehungsgeflecht von unterschiedlichen, teils konkurrierenden Einzelsachverhalten dar. Das wird besonders deutlich bei der präventiven Beurteilung von stadtklimatischen Wirkungen, die in Folge von Bebauungsplänen im kommunalen Bereich auftreten können. Im vorliegenden Fallbeispiel wird mit Hilfe softwaregestützter Anwendung von Fachinformationssystemen (FIS Broker Berlin) dargestellt, wie über eine Variantendiskussion die Wirkung eines Bebauungsplanes auf stadtklimatische Verhältnisse prognostiziert werden kann. Da der zur Verfügung stehende Datenbestand zum Teil stark veraltet war (Stand 2008), kann in Auswertung der mit ArcMap erstellten Karten trotzdem dargestellt werden, dass diese Art der Visualisierung der Wirkung von Bebauungsstrukturen durchaus für eine verbale Ableitung ihrer Wirkung geeignet ist. Obwohl sie für eine gesicherte Prognose nicht ausreichend sind, kann diese Form der Darstellung von Wirkungsszenarien sowohl für die öffentliche als auch für die politische Diskussion zu Planvorhaben zielführend sein. Unter Aufzeigen und Visualisierung des Ist- und des Planzustandes eines Vorhabens kann zwar auf die Problemlagen möglicher Wirkungen hingewiesen und eine verbale Diskussion abgeleitet und geführt werden, entbehrt aber einer situativen Prognosesicherheit. Für eine qualifizierte Beurteilung potenzieller Wirkungen ist sie für städtebauliche Vorhaben jedoch nicht ausreichend. Da diese derzeit auch nicht nach BauGB gefordert ist und das Bebauungsplanverfahren als Rechtsinstrument kommunaler Entwicklungsplanung simulativ ermittelte Prognoseverfahren für einen Realzustand, d.h. nach der Bebauung, nicht vorsieht, scheint es geboten, den politischen Blick stärker in Richtung präventives klimagerechtes Bauen zu richten. Von einigen Kommunen wird das klimarechte Bauen bereits in ihre sektorale Entwicklungsplanung aufgenommen. Im Fokus eines zukunftsfähigen Bauens steht jedoch das stadtklimatische Bauen als Herausforderung für alle Akteure. Das stadtklimagerechte Bauen bietet ein neues Entwicklungsfeld, erfordert aber gleichermaßen auch neue technische Lösungen auf verschiedenen Ebenen.

Auf der Ebene des technischen Bedarfes steht eine leistungsfähige Informationstechnik, die arbeitsplatzspezifische Lösungen für Planer, Gutachter und Politik anbietet und flexibel auf Änderungen reagieren kann. Dazu gehören auch Entwicklungen von nutzer-

spezifischer leistungsfähiger Simulationssoftware und von Modellierungsinstrumenten. Gleichermäßen ist es erforderlich Ausbildungsprofile für Informatiker und Stadt- und Regionalplaner für diese Herausforderungen zu schaffen.

3 Referenzen

- [1] Bimüller, Benner (2009): Modellprojekt "XPlanung" – Mitteilung 4/2009, Abschlussbericht Seite 14-18, veröffentlicht von www.it-planungsrat.de
- [2] Stock(2005): Handlungskataloge und Maßnahmen zur Anpassung an den Klimawandel, in: Handbuch Stadtklima: Maßnahmen und Handlungskonzepte für Städte und Ballungsräume zur Anpassung an den Klimawandel, S. 21
- [3] FIS Broker FIS Broker (2005): Bioklima- Wärmebelastung in der Nacht, Bewertungsindex PMV (Predicted Mean Vote) in 2 m über Grund um 04:00 Uhr während einer austauscharmen sommerlichen Strahlungsnacht (2005), http://fbinter.stadt-berlin.de/fb/index.jsp?Szenario=fbinter_jsc [Stand: 28.10.15, 17:47]
- [4] Kokam, Aimée (2015): Bachelorthesis Software gestützte Ableitung von wertsetzenden Indikatoren für ein stadtklimagerechtes Bauen an einem konkreten Bauvorhaben, Diskussion der Ergebnisse, Optimierte Bebauungsstruktur auf dem Planfeld, S. 48
- [5] ArcMap Tool aus ArcGIS 10
- [6] CleanTechPark Cleantech Business Park Berlin Marzahn (2010): Bebauungsplan 10 -56, in: Schema-tische Darstellung der Baufläche, S.13, 3.03.2010
- [7] Stützel Veronika (2007): Stadt, Luft, Fluss; in: Teil A, Klimawirksames Planen und Bauen, Luftaustausch in der Stadt, S.2 [PDF <https://www.google.com/search?q=Veronika%2C+St%C3%BCtzel%3A+Stadt%2C+Luft%2C+Fluss&ie=f-8&oe=utf-8>] [Stand 30.10.2015, 19:07]
- [8] GEONET GEO-NET Umweltconsulting GmbH (2014): Prof. Dr. G. Groß, Klimafunktionskarte und Planungshinweiskarte Klima/Luft für die Universitätsstadt Gießen, Analyse der klima- und immissionsökologischen Funktionen im Stadtgebiet von Gießen Modellgestützte Analyse 2014, , 28. Juli 2014, S. 22, [PDF]

Vergleich der Wirksamkeit von Steuerungsregeln zur Distribution von Flüssigaluminium in Nicht-Eisen-Schmelz- und Druckgussbetrieben durch eine hybride Fertigungssimulation

Dominik Jeckle¹, Wolfgang Schlüter¹, Ringleb Ansgar¹

¹Hochschule Ansbach

d.jeckle@hs-ansbach.de

Diese Arbeit leistet einen wichtigen Beitrag zur Entwicklung eines simulationsbasierten Prozessmanagementsystems, dass der Optimierung der energetischen und logistischen Prozesse in der Nicht-Eisen-Schmelz- und Druckgussindustrie, dient. Durch eine Simulationsstudie wird gezeigt, dass durch die Verwendung eines intelligenten Algorithmus zur Befüllung von Druckgussanlagen die Anzahl der angefahrenen Druckgussanlagen in bestimmten Betriebsfällen um bis zu 30 % reduziert wird. Dadurch ist im Vergleich zu einem in der Schmelz- und Druckgussindustrie weit verbreiteten heuristischen Ansatz eine Energieeinsparung bei den Druckgussanlagen zu erwarten. Der energetische Vorteil resultiert daraus, dass sich mit jeder Befüllung der elektrisch betriebenen Druckgussanlage das flüssige Aluminium abkühlt. Die zugrundeliegende gekoppelte Material- und Energieflusssimulation ist durchgehend in Matlab/Simulink/Stateflow implementiert. Für die Steuerung des Materialflusses ist ein Steuerungsmodul verantwortlich, welches ebenfalls in Matlab entwickelt wurde. Dieses beinhaltet unter anderem die beiden Algorithmen zur Befüllung der Druckgussanlagen, auf welche im Verlauf dieser Arbeit genauer eingegangen wird. Komplettiert wird die nach dem gängigen Prinzip der objektorientierten Programmierung entwickelte Software Suite durch ein Modul zur Visualisierung und Auswertung.

1 Einleitung

Die Steigerung der Energieeffizienz in Schmelz- und Druckgussbetrieben der Metallindustrie ist seit vielen Jahren ein wichtiges Thema der Gießerei-Branche. Bereits durch einfache Maßnahmen, wie z.B. der Anzeige des nächsten Produktionsauftrages sowie der aktuellen Anlagenzustände, lassen sich bedingt durch die verbesserte Informationslage der Mitarbeiter, beachtliche Energieeinsparungen erzielen [1]. Umfassendere, den Prozessablauf betreffende Maßnahmen werden, aufgrund der Komplexität der Prozesse und der nicht vorhersehbaren Auswirkungen, nicht betrachtet. Hier kann mittels Simulation Transparenz geschaffen werden [2]. Dieser Ansatz wird im Teilprojekt *Smart Melting* des Forschungsverbundes Green Factory Bavaria [3] aufgegriffen. Mittels eines simulationsbasierten Prozessmanagementsystems (PMS) soll der Energieverbrauch der Schmelzanlagen (ca. 50 % des Gesamtenergieverbrauches) in der Nicht-Eisen (NE) Schmelz- und Druckgussindustrie optimiert werden. Eine wichtige Randbedingung ist, dass die Maßnahmen keine negativen Auswirkungen auf die Gesamtproduktivität des Betriebes haben dürfen. Mit Hilfe der entwickelten Simulation werden z.B. die Auswirkungen von logistischen Verteilungs-

regeln (Staplersteuerung) getestet, die entweder eine Erhöhung der Produktivität oder eine Verbesserung der Energieeffizienz fokussieren. Algorithmen mit hohem Verbesserungspotential sollen anschließend in dem geplanten PMS implementiert werden. Zusätzlich soll die Simulation selbst Teil des geplanten PMS sein, da so der zukünftige Betriebszustand berechnet werden kann. Daraus können notwendige Maßnahmen zur Prozesssteuerung in Extremsituationen, wie der Ausfall eines oder mehrerer Schmelzöfen, abgeleitet werden. Die Steuerungsanweisungen können zukünftig in Form eines Assistenzsystems über ein Display an den Werker erfolgen oder direkt an die Maschinensteuerungen übergeben werden. Zur Übersicht und Auswertung der Anlagenzustände eignet sich die Anzeige über einen entsprechenden Leitstand.

1.1 Beschreibung der Software Suite

Die durchgehend in Matlab umgesetzte Software Suite besteht aus den drei Modulen *Simulation*, *Steuerung* und *Visualisierung & Auswertung* (siehe Abbildung 1).

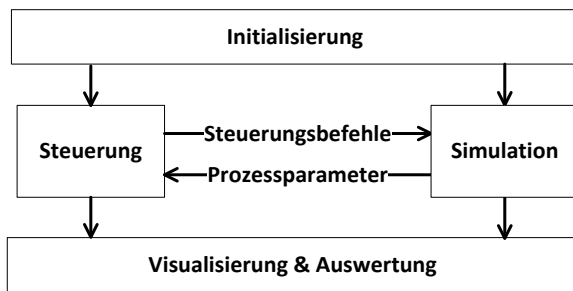


Abbildung 1. Schema der Software Suite

Die Parametrierung des Simulations- bzw. des Steuerungsmoduls wird in Excel vorgenommen und in Matlab importiert. Hier können sowohl verschiedene Betriebsstrukturen als auch Steuerungsmodi konfiguriert werden. Die Steuerung (Matlab) und Simulation (Simulink/Stateflow) kommuniziert bidirektional über Interpreted Matlab Functions. Dabei werden sowohl Prozessparameter als auch Steuerungsbefehle ausgetauscht. Der interne Aufbau der Module berücksichtigt das gängige Prinzip der objektorientierten Programmierung. Die Datenhaltung der einzelnen Komponenten (z.B. Stapler, Schmelzöfen, Schmelzbrenner, Druckgussanlage) erfolgt somit in Objekten[4]. Mehrere Objekte einer Klasse werden in Strukturen gebündelt.

Das Simulationsmodul besteht aus einer gekoppelten Material- und Energieflusssimulation, welche in Simulink bzw. Stateflow abgebildet sind. Sie umfasst den kompletten Anlagenbetrieb eines beliebigen NE-Schmelz- und Druckgussbetriebes (siehe Abbildung 2).

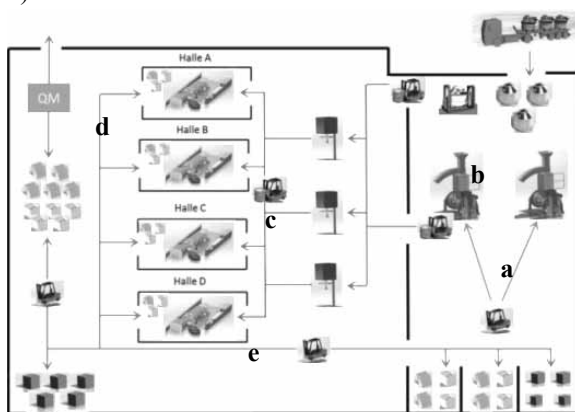


Abbildung 2. Schema der Prozesssimulation eines Schmelz- und Druckgussbetriebes

Die simulierten und gesteuerten Prozesse im Einzelnen:

- Beschickung der gasbetriebenen Schachtschmelzöfen über Stapler mit Masseln, Rücklauf- oder Ausschussmaterial (Abbildung 2, a)
- Erwärmungs- und Schmelzvorgänge im Schmelzofen unter Berücksichtigung von Beschickungs- und Entnahmevorgängen mit Berechnung des Gasverbrauchs (Abbildung 2, b)
- Verteilung des flüssigen Aluminiums mit Staplern auf die Dosieröfen der Gussmaschinen (Abbildung 2, c)
- Produktion von Gussteilen unter Berücksichtigung von Taktzeit, Schussgewicht und Ausschussrate (Abbildung 2, d)
- Berücksichtigung geplanter Stillstände aufgrund von Rüstvorgängen (Abbildung 2, d)
- Transport von vollen/leeren Materialbehältern aus dem Druckgussbetrieb oder von Masselpaketen aus dem Lager zum Schmelzbetrieb (Abbildung 2, e)

Die separate Entwicklung der Material- und Energieflusssimulation macht eine Kopplung notwendig, ermöglicht aber auch eine getrennte Weiterentwicklung beider Systeme. Die abgebildeten Anlagen werden in Stateflow beschrieben. Änderungen des Zustandes einer Komponenten bzw. Aktionen, die aus den verschiedenen Zuständen der Anlagen resultieren, werden in Interpreted Matlab Functions verarbeitet. Hier erfolgt auch die Aktualisierung der Prozessparameter des entsprechenden Objektes. Für die Entwicklung des Energiemodells, welches die thermodynamischen Vorgänge des Schmelzofens abbildet, wurden eine dynamische Prozesssimulation in Verbindung mit dem Computational Fluid Dynamics Simulationsverfahren eingesetzt [5]. Die Differentialgleichungen für die Berechnung des Material- und Energieflusses innerhalb des Schmelzofens sind in Simulink abgebildet [6]. Die benötigten Schnittstellen zur Kopplung der beiden Simulationen (siehe Abbildung 3) sind durch Objekte und entsprechende Methoden für den Austausch der Informationen realisiert.

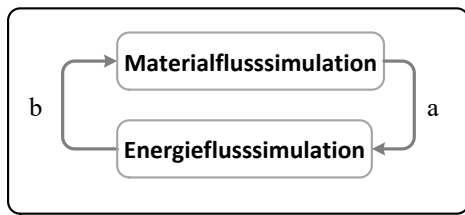


Abbildung 3: gekoppelte Material- und Energieflusssimulation

Aus dem Materialfluss werden dem Energiemodell des Schmelzofens Informationen über die Beschickung und Entnahme zur Verfügung gestellt (siehe Abbildung 3, a). Das Energiemodell übergibt dem Materialfluss (siehe Abbildung 3, b) die neu berechneten Füllstände der Komponenten des Schmelzofens.

Das zweite Modul *Steuerung* übernimmt die Aufgabe der Prozesssteuerung. Darunter fallen hauptsächlich die Steuerung der Stapler (Abbildung 2, a, c, e) aber auch Vorgänge, wie das Freischmelzen und Reinigen der Schmelzöfen. Durch die Parametrierung in Excel ist es möglich den Materialfluss über verschiedene logistische Regeln zu steuern. Das Modul *Visualisierung & Auswertung* kann den Betriebszustand der einzelnen Anlagen zu verschiedenen Zeitpunkten der Simulation visualisieren. Des Weiteren werden logistische Kennzahlen wie z.B. die Auslastung der Stapler oder der Druckgussanlagen für den gesamten Simulationszeitraum berechnet. Die energetischen Auswirkungen werden mittels des spezifischen Energieverbrauchs (Energieverbrauch pro Tonne geschmolzenes Aluminium) sowie des Gasverbrauchs der Schmelzöfen bewertet. Eine detailliertere Auswertung wird nach einer Funktionserweiterung dieses Modules in kürze möglich sein. Die Funktionserweiterung erlaubt es Anlagenzustände und Kennzahlen von einem oder mehreren Simulationsdurchläufen gleichzeitig in verschiedenen Simulationszeiträumen zu betrachten bzw. in Beziehung zu einander zu setzen.

1.2 Steuerung der Verteilung des Flüssigaluminiums auf die Gussanlagen

Für den Produktionsbetrieb ist eine gesicherte Versorgung der Druckgussmaschinen mit Aluminium entscheidend. Daher kommt der Steuerung der Stapler zur Verteilung des Flüssigaluminiums vom Schmelzofen über eine Impellerstation (Imp) zu bis zu vier Druckgussmaschinen (DGM) besondere Bedeutung

zu. Die Steuerungsbefehle werden über Aufträge an die Simulation übergeben.

Quelle	Impellerstation	Senke	Materialmenge
Ofen1	Imp 2	DGM 10	950 kg

Tabelle 1. Beispiel für einen Auftrag zur Verteilung von flüssigem Aluminium

Bevor der Algorithmus die Auftragsdaten (siehe Tabelle 1) bestimmen kann, müssen vorab bestimmte Bedingungen (Constraints) erfüllt sein. Im Falle der zu ermittelnden Druckgussanlagen dürfen höchstens vier Anlagen mit einer Mindestauftragsmenge von je 300 kg angefahren werden.

Der eigentliche Algorithmus zur Ermittlung der Auftragsdaten für die Verteilung des flüssigen Aluminiums auf die Dosieröfen der Gussmaschinen besteht aus drei Teilen:

1. Ermittlung der Materialquelle,
2. Ermittlung der Impellerstation zum Reinigen der Schmelze,
3. Ermittlung der zu befüllenden Druckgussanlagen inklusive der zu befüllenden Mengen.

Im Folgenden wird Teil 3, die Ermittlung der zu befüllenden Druckgussanlagen, genauer erläutert.

Die Befüllung kann nach verschiedenen Kriterien erfolgen, wobei hier das in der Industrie häufig verwendete Ampelverfahren (Variante A) und alternativ ein auf der Analyse der verbleibenden Restlaufzeiten der Gussmaschinen basierendes Verfahren (Variante B) behandelt werden.

Das Ampelverfahren stellt eine Schrankenregelung dar, bei der ein Füllstand innerhalb eines bestimmten Grenzwertes eine diesem Bereich zugeordnete Signalfarbe der Ampel auslöst. Diese Signalfarbe dient im Sinne einer Steuerung als Information zur Priorisierung der Befüllung. Im Rahmen der vorliegenden Untersuchung werden dazu drei Füllstandbereiche gekennzeichnet und durch drei Signalfarben, unterschieden: Ein Füllstand zwischen 100 % und 80 % gilt als unkritisch und entspricht der Signalfarbe „grün“, weshalb keine Befüllung ausgelöst wird. Ein Füllstand zwischen 80 % und 60 % entspricht der Signalfarbe „gelb“, was als unkritisch übersetzt wird, der jedoch eine Befüllung erlaubt. Ein Füllstand klei-

ner 60 % wird als kritischer Füllstand übersetzt und entspricht der Signalfarbe „rot“. Das bedeutet, dass diese Druckgussanlage priorisiert zu befüllen ist.

Statt dem Ampelverfahren kann die Befüllung nach der kürzesten Restlaufzeit der Gussmaschinen (Steuerungsvariante B) erfolgen. Dabei wird derjenige Dosierofen zuerst befüllt, der nach aktuellem Füllstand ($m_{\text{Füllstand}}$) und gegebener Taktzeit (t_{Takt}), Schussgewicht (m_{Schuss}) die geringste Restlaufzeit (t_{Rest}) bis zu einem Stillstand aufgrund von Materialmangel aufweist.

$$t_{\text{Rest}} = \frac{m_{\text{Füllstand}}}{m_{\text{Schuss}}} \times t_{\text{Takt}} \quad (1)$$

Die verarbeitete Information über den Füllstand bei Steuerungsvariante A ist ungenau, da nur zwischen Füllstandbereichen unterschieden werden kann und damit ein Füllstand von 10 % gleich behandelt wird wie ein Füllstand mit 59 %. Der Staplerfahrer wählt anhand der Farben (rot/gelb/grün) eine oder mehrere Anlagen aus. Bei der Steuerungsvariante B ist der Informationsgehalt erheblich größer, da neben dem genauen aktuellen Füllstand noch weitere Produktionsdaten wie Schussgewicht und Taktzeit mit eingehen. Dies lässt eine verbesserte Versorgung der Druckgussmaschinen mit Flüssigaluminium erwarten.

1.3 Auswahl der Druckgussanlage

Zunächst werden von allen relevanten Druckgussanlagen, die die Constraints erfüllen (zu befüllende Menge ≥ 300 kg), das Kriterium (nach gewählter Variante relativer Füllstand oder Restlaufzeit) ermittelt.

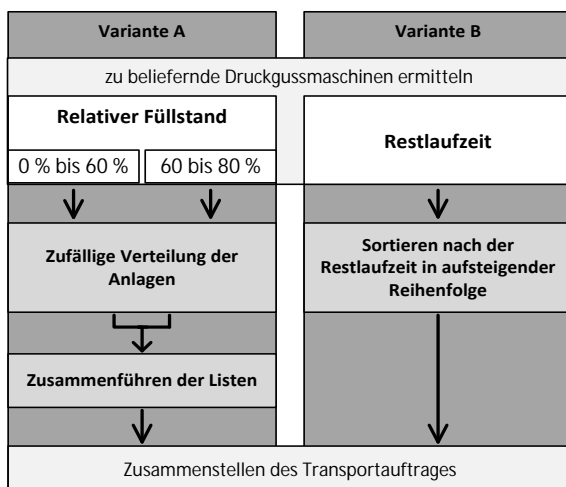


Abbildung 4. Algorithmus zur Auswahl der Druckgussmaschinen

Die ermittelten Anlagen werden in temporären Listen gespeichert. Um bei der in der industriellen Praxis üblichen Steuerungsvariante A die zufällige Auswahl von Druckgussmaschinen, deren Füllstände sich in einem Bereich (rot, gelb) befinden, nachzubilden, werden die Druckgussanlagen zufällig in zwei Listen angeordnet. Bei Konfiguration B werden die Druckgussanlagen anhand der Restlaufzeit in einer Liste in aufsteigender Reihenfolge sortiert. Danach werden bei Konfiguration A die zwei Listen zusammengeführt, wobei Aufträge mit rotem Signallicht höher als Aufträge mit gelbem Signallicht priorisiert werden.

Nun muss ein Auftrag aus den ausgewählten Anlagen, deren zu befüllende Menge große Schwankungen aufweisen kann (300 kg – 2450 kg), zusammengestellt werden. Dazu wird der erste Wert aus der Liste entnommen und geprüft, ob die zu befüllende Menge ($m_{\text{zu befüllen}}$) größer ist als Kapazität des Transporttiegels ($m_{\text{max}} = 1200$ kg). Ist dies der Fall wird die ausgewählte Druckgussanlage mit der Gesamtauftragsmenge (m_{Auftrag}) von 1200 kg beliefert. Ist die zu befüllende Menge kleiner als 1200 kg, werden der Reihe nach so viele Druckgussanlagen ausgewählt, bis eine Gesamtauftragsmenge von 1200 kg erreicht wurde. Dies gilt nur, wenn mehrere Druckgussanlagen Material benötigen. Ansonsten kann auch ein Auftrag mit einer Auftragsmenge kleiner 1200 kg erteilt werden.

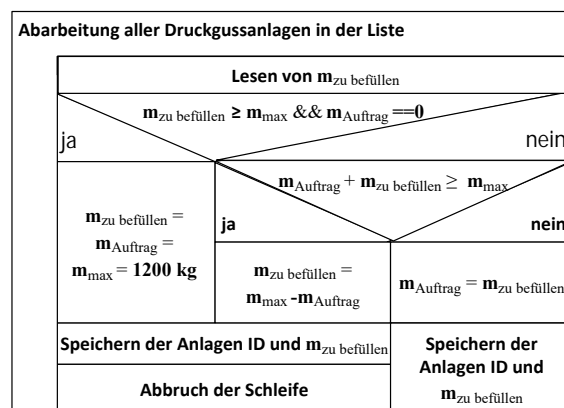


Abbildung 5. Finale Auswahl der Druckgussanlagen

Danach werden die bestimmten Druckgussanlagen und der im Teil 1 bestimmte Schmelzofen und die bestimmte Impellerstation (Teil 2) an den Staplerfahrer weitergegeben.

2 Simulationsstudie

Die abgebildete Anlagenstruktur des Schmelzbereiches umfasst 4 Schmelzöfen mit einer Gesamtschmelzleistungskapazität von $P_{\text{Schmelzbetrieb}} = 11,3$ to./h. Der Gussbereich umfasst 31 Gussmaschinen, wobei 11 Gussmaschinen (große Anlagen) ein hohes Schussgewicht von 25 kg und eine lange Taktzeit von 120 s aufweisen. Die übrigen 20 Gussmaschinen (kleine Anlagen) werden mit einem kleineren Schussgewicht von 5 kg und einer kürzeren Taktzeit von 80 s betrieben. Damit verfügen die modellierten Gussmaschinen über eine Nachfragekapazität (ohne Stillstandszeiten) von $P_{\text{Gussbetrieb}} = 12,6$ to./h. Zudem werden 3 Stapler eingesetzt, um das flüssige Aluminium auf die verschiedenen Druckgussanlagen zu verteilen. Simuliert wird eine Kalenderwoche, was 17 Schichten á 8 Stunden pro Schicht entspricht.

Zur Untersuchung der Auswirkung der Distributionsvarianten A und B auf den Gesamtbetrieb werden mehrere Betriebsfälle einer Kalenderwoche betrachtet. Diese sind durch unterschiedliche Stillstandszeiten der Druckgussmaschinen (z.B. zu Reparaturzwecken und Formwechseln) charakterisiert. Damit werden Betriebsfälle bei unterschiedlichen Auftragslagen nachgestellt, die die Auslastung der produzierenden Druckgussanlagen bestimmen. Die Stillstandszeiten, welche 25 % der Planlaufzeit betragen (Datenaufnahme Referenzbetrieb) werden schrittweise reduziert, sodass eine Planlaufzeit von 100 % erreicht wird. Die daraus resultierende Nachfrage der Druckgussmaschinen ($P_{\text{Gussbetrieb}}$) berechnet sich aus der Betriebszeit ($t_{\text{Gussbetrieb}}$), den geplanten Stillstandszeiten ($t_{\text{Stillstand geplant}}$), der Taktzeit (t_{Takt}) und der benötigten Masse pro Schuss (m_{Schuss}).

$$P_{\text{Gussbetrieb}} = \text{floor}\left(\frac{t_{\text{Gussbetrieb}} - t_{\text{Stillstand geplant}}}{t_{\text{Takt}}}\right) \times m_{\text{Schuss}} \quad (2)$$

Tabelle 2 beinhaltet eine Übersicht der durchgeführten Simulationen.

Fall	1	2	3	4	5	6
t Stillstand geplant [%]	25	20	15	10	5	0
$P_{\text{Gussbetrieb}}$ [to./h.]	9,0	9,8	10,5	11,2	11,9	12,7
$P_{\text{Schmelzkap}}$ [to./h.]	11,3	11,3	11,3	11,3	11,3	11,3

Tabelle 2. Darstellung des Angebot-und-Nachfrage-Verhältnis im Schmelz- und Druckgussbetrieb

Die simulierten Betriebsfälle werden anhand von drei Kennzahlen bewertet. Gemäß (3) stellt die Auslastung der Stapler (L_{Stapler}) die Reaktionsbereitschaft der Stapler in kritischen Betriebsphasen dar. Die Auslastung berechnet sich aus dem Verhältnis aus Betriebszeit der Stapler ($t_{\text{Betrieb Stapler}}$) und der verfügbaren Arbeitszeit ($t_{\text{Arbeitszeit}}$).

$$L_{\text{Stapler}} = \frac{t_{\text{Betrieb Stapler}}}{t_{\text{Arbeitszeit}}} \quad (3)$$

Als zweite Kennzahl dient die Stillstandszeit der Druckgussmaschinen aufgrund von Aluminiummangel ($t_{\text{Stillstand Alu}}$).

Die letzte Kennzahl ist die Masse der hergestellten Gutteile (m_{Gutteile}) zur Bewertung der Gesamtproduktivität. Die Stillstandszeit der Druckgussanlagen als zweite Kennzahl ist dazu nur bedingt aussagekräftig, da auf den Druckgussanlagen verschiedene Gussteile mit unterschiedlichen Schussgewichten und Taktzeiten hergestellt werden.

3 Ergebnisse

Die Ergebnisse der Simulationsstudie zeigen, dass es zwischen den beiden getesteten Steuerungskonfigurationen A und B erst ab Fall 4 (Stillstandszeit $t_{\text{Stillstand geplant}} \leq 10\%$) Unterschiede gibt. Entgegen der Erwartung konnte die Strategie B, welche mehr Informationen verarbeitet, keine Reduktion der Ausfallzeiten der Druckgussanlagen bewirken. So sind die Stillstandszeiten bei Variante A um ca. 30 % bis 40 % niedriger als bei Variante B (siehe Tabelle 3).

Fall	1	2	3	4	5	6
A: $t_{\text{Stillstand Alu}}$ [h]	0	0	3	132	280	435
B: $t_{\text{Stillstand Alu}}$ [h]	0	0	0	189	443	688

Tabelle 3. Vergleich der Stillstandszeiten der Druckgussanlagen aufgrund von Aluminiummangel

Da bei den Simulationsstudien die Schmelzkapazität der Öfen nicht verändert wurde (siehe Tabelle 2) und somit in beiden Varianten die gleiche Menge an flüssigem Aluminium zur Verfügung stand deutet dies darauf hin, dass durch die zufallsbedingte Steuerung (Variante A) mehr kleine Anlagen (Taktzeit 80 s) angefahren werden. Diese verbrauchen pro Stunde weniger Aluminium und weisen bei einem vollen Speicher im Mittel eine höhere Restlaufzeit auf als

große Anlagen (Taktzeit 120 s). Dies könnte die kürzeren Ausfallzeiten bei Variante A bewirkt haben.

In Tabelle 4 wird die Masse der hergestellten Gutteile zur Bewertung der Gesamtproduktivität dargestellt. Es zeigt sich, dass bei beiden Strategien sowohl die Masse der hergestellten Gutteile bis Fall 3 um 8 % bzw. 7 % je Fall ansteigt. Danach (Fall 4 bis 6) liegt die Steigerungsrate bei ca. 1 %. Dies zeigt, dass dem Gesamtsystem zu wenig Aluminium zur Verfügung steht und die kürzeren geplanten Stillstandszeiten (siehe Tabelle 2) keine signifikanten Produktivitätssteigerungen zur Folge haben.

Fall	1	2	3	4	5	6
A: m Gutteile [t]	689	744	793	801	802	803
B: m Gutteile [t]	689	743	796	806	807	811

Tabelle 4. Vergleich Masse der hergestellten Gutteile

Zudem fällt auf, dass durch die Anwendung von Strategie B, die höhere Ausfallzeiten zur Folge hat (siehe Tabelle 3), die verbrauchte Masse (Fall 4 bis 6) an Aluminium höher ist, wobei auch hier der Unterschied nicht relevant ist. Dies deutet ebenfalls darauf hin, dass durch die zufallsbedingte Auswahl der Druckgussanlagen eher bevorzugt kleine Anlagen ausgewählt werden und bezogen auf die Ausfallzeiten die Strategie B keine Verschlechterung darstellt.

In Tabelle 5 wird die Auslastung der Stapler dargestellt. Bei beiden Strategien ist die Auslastung in den Fällen 1 bis 3 nahezu identisch. In den Fällen 4 bis 6 ist eine Abweichung zwischen 2 % und 4% festzustellen.

Fall	1	2	3	4	5	6
A: L_{Stapler} [%]	92	94	92	78	75	73
B: L_{Stapler} [%]	92	95	93	76	71	69

Tabelle 5. Vergleich der Auslastung der Stapler

Bei beiden Strategien zeigt sich ab Fall 4 eine deutlich niedrigere Auslastung. Dies ist mit den bisher vorgestellten Ergebnissen nicht zu erklären. Deshalb werden für die weitere Analyse der Staplerauslastung die Anzahl der Stapleraufträge ($n_{\text{Staplerauftrag}}$) und die Anzahl der angefahrenen Druckgussanlagen (n_{mittel}

n_{DGM}) herangezogen. Des Weiteren werden die mittlere Auftragsmenge je Staplerauftrag in kg (m_{Stapler}) und die mittlere Auftragsmenge je Druckgussanlage in kg ($m_{\text{mittel DGM}}$) für die Bewertung der Auslastung der Stapler verwendet. Dies erfolgt anhand der Betrachtung der Fälle 3 und 4 der Strategie A in Tabelle 6.

Es ist zu konstatieren, dass von Fall 3 zu Fall 4 die Anzahl der Stapleraufträge um ca. 10 % und die Anzahl der angefahrenen Druckgussanlagen um ca. 27 % zurückgeht. Gleichzeitig erhöhen sich die mittlere Auftragsgröße der Stapler und die Menge an flüssigem Aluminium die der Druckgussanlage je Auftrag zugeführt wird

Fall	$n_{\text{Staplerauftrag}}$	n_{DGM}	$m_{\text{mittel Stapler}}$ [kg]	$m_{\text{mittel DGM}}$ [kg]
3	1420	3702	1041	410
4	1278	2687	1167	587

Tabelle 6. Analyse der Staplerauslastung für Strategie A

Folglich werden mit weniger Staplerfahrten, weniger Druckgussanlagen angefahren. Diesen wird aber mehr Material zugeführt und die Transportkapazität des Staplers von 1200 kg wird besser ausgenutzt. Somit reduziert sich die Auslastung der Stapler. Betrachtet man noch die Gesamtmenge des von den Staplern ausgelieferten Aluminiums, so lässt sich feststellen, dass diese in beiden Fällen nahezu gleich ist. Bei einem Vergleich von Fall 1 und 6 zeigt sich das oben beschriebene Verhalten deutlicher. Hier liegt für Fall 1 die mittlere Auftragsmenge je Staplerauftrag bei 915 kg und in Fall 6 bei 1200 kg. Zudem wurde die ausgelieferte Menge um 15 % gesteigert.

In Tabelle 7 werden für beide Strategien, die in Tabelle 6 verwendeten Kennzahlen, bezogen auf Fall 6, gegenüber gestellt. Es zeigt sich dass die Anzahl der Stapleraufträge und die mittlere Auftragsmenge bei beiden Strategien gleich sind. Ein Unterschied ergibt sich erst bei einer Betrachtung der Anzahl der angefahrenen Druckgussanlagen (n_{DGM}) und der ausgelieferten Masse pro DGM ($m_{\text{mittel DGM}}$). So werden durch die Anwendung der Strategie B, bei gleichbleibender transportierter Gesamtmasse, rund 30 % weniger Anlagen angefahren, diese aber mit rund 30 % mehr Material pro Auftrag versorgt.

	n Staplerauftrag	n DGM	m _{mittel} Stapler [kg]	m _{mittel} DGM [kg]
A	1239	2218	1200	689
B	1239	1693	1200	952

Tabelle 7. Vergleich der Staplerauslastung der Strategien A und B für Fall 6

Der Grund weshalb sich dieser deutliche Unterschied in der Auslastung der Stapler mit nur 4 % widerspiegelt ist, dass die Transportzeiten zwischen den Druckgussanlagen in der Simulation relativ kurz im Vergleich zur Gesamtdauer eines Auftrages gewählt sind. Übertragen auf den realen Schmelz- und Druckgussbetrieb kann durch die Reduzierung der angefahrenen Druckgussanlagen im Allgemeinen eine Verbesserung des logistischen Prozesses erwartet werden. Bezüglich der nachzuweisenden Energieeffizienz kann davon ausgegangen werden, dass sich durch Strategie B eine Reduktion des Energieverbrauchs der elektrisch betriebenen Druckgussanlagen einstellt, da bei jeder Befüllung der Druckgussanlage der Speicherbehälter geöffnet werden muss und deshalb eine Abkühlung des Materials stattfindet.

4 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurden zwei Steuerungs-algorithmen (A/B) zur Steuerung der Stapler für die Belieferung von Druckgussanlagen mit flüssigem Aluminium und deren Simulation in unterschiedlichen Betriebsfällen vorgestellt. Anhand einer Simulationsstudie wurde gezeigt, dass der mehr Informationen verarbeitende Algorithmus B im Vergleich zur Anwendung des Algorithmus A, der die in der Industrie übliche Strategie nachbildet, keine signifikanten Verbesserungen oder Verschlechterungen bezogen auf die Ausfallzeiten der Druckgussanlagen, die produzierte Gesamtmasse und die Auslastung der Stapler liefert. Allerdings bewirkt die Anwendung der Strategie B eine Reduktion der angefahrenen Druckgussanlagen für die jeweiligen Betriebsfälle, bei denen das Verhältnis von Angebot und Nachfrage kleiner gleich 1 ist. Hier kann auch angesichts der Vereinfachungen, die bei der Modellierung der Material- und Energiefluss-simulation gemacht wurden, eine Anwendung von Variante B empfohlen werden. Für die Betriebsfälle, bei denen das Angebot-Nachfrage Verhältnis größer 1 ist (Fall 1 bis 3), konnten im Verlauf der Untersuchung keine Vorteile der Strategie B festgestellt werden. Hier sind für eine tiefergehende Analyse weitere

technische Hilfsmittel notwendig. Diese stehen in Form des Analyse-Tools *Melting Analytics* in Kürze zur Verfügung. Des Weiteren soll der Algorithmus B um die Kenntnis der in dem Enterprise Resource Planning System festgelegten geplanten Stillständen (z.B. Rüsten) ergänzt und als zusätzliche Variante untersucht werden. Die Kenntnis von Stillstandszeiten wird aller Voraussicht nach zu einem noch besseren Prozessverhalten führen.

5 References

- [1] A. Effenberger. *Aus einem Guss. Neues MES-System für die Eisengießerei Baumgarte*. Gieserei-Erfahrungsaustausch, Nr. 5+6, S. 6–9, 2016.
- [2] E. Fuss, S. Drvendzija, A. Krewald. Effenberger. *Transparenz als Schlüssel zur energieeffizienten Produktion: Ablaufplanung in Schmelzerei und Gießerei mittels einer planungsbeleitenden Energie- und Materialflusssimulation*. Productivity Management, Nr. 3, S. 35–38, 2013.
- [3] <http://www.greenfactorybavaria.net>, 22.07.2016.
- [4] D. Jeckle, W. Schlüter und A. Ringleb. *Objektorientierte Entwicklung einer hybriden Materialflusssimulation eines NE-Schmelz- und Druckgussbetriebes*. in Workshop der ASIM/GI Fachgruppen STS 2016: Simulation technischer Systeme, in Stralsund, S. 209–217, 2015.
- [5] J. Schmidt, W. Schlüter und A. Ringleb. *Kopplung von Prozess- und CFD-Simulation zur Untersuchung des Wärmeverlusts beim Flüssigaluminiumtransport*. in Workshop der ASIM/GI-Fachgruppen: Simulation technischer Systeme, in Stralsund, S. 209–217, 2015.
- [6] J. Schmidt, W. Schlüter. *Ein dynamisches Prozesssimulationsmodell für die energetische Betrachtung von Aluminium-Schmelzöfen in einer betriebsumfassenden Materialflusssimulation* in Workshop der ASIM/GI Fachgruppen STS und GMMS 2016, Hamm: ARGESIM Verlag Wien, 2016, pp. 29–37.

OptPlanEnergie - AN OPTIMIZATION AND SCHEDULING PLATFORM FOR THE ENERGY-EFFICIENT PRODUCTION OF TEMPERED GLASS

Frank Baumann¹, Heike Wilson¹, Stephan Seidel², Matthias Franke², Ulrike Gromnitza³

¹DUALIS GmbH IT Solution, Dresden

²Fraunhofer Institut für Integrierte Schaltungen, Institutsteil Entwicklung Adaptiver Systeme, Dresden

³Bischoff Glastechnik GmbH, Bretten

FBaumann@dualis-it.de, stephan.seidel@eas.iis.fraunhofer.de, ulrike.gromnitza@bgt-bretten.de

The aim of the project OptPlanEnergie is to develop an integrated optimization and scheduling platform that specifically targets the reduction of energy consumption by optimizing the production orders in the glass tempering industry. The platform will use multi-criteria optimization techniques to determine energy-efficient loading sequences, utilizing thermo-dynamic black box models trained on data measured by energy meters at the tempering process of safety glass.

1 INTRODUCTION

One of the main cost factors in many industrial processes is energy consumption. Minimizing energetic losses is therefore a key challenge faced by energy-intensive industries, such as glass and steel production or chemical industries.

Energetic losses often result from two main factors, insufficiently calibrated production processes and the generation of waste heat.

In the project OptPlanEnergie, production processes for the manufacturing of tempered glass are analyzed and optimized. The aim of the project is to develop an integrated optimization and scheduling platform that applies rough thermo-dynamic models of the tempering process to calculate the expected energy consumption and optimization algorithms to determine optimal production sequences. This will allow the minimization of energy-usage per produced unit while maintaining high quality as well as high production output. The optimization and scheduling platform will aid manufacturers in the operative planning of existing processes as well as in the strategic planning of production facilities.

In this paper we concentrate on the tempering process and the analysis of the available measurement data, which we describe in the subsequent section. The main focus is on optimizing production sequences

and batch sizes. Section 3 gives a concise overview of the main components that will form the basis of the energy-aware scheduling platform. Finally, a conclusion is given in Section 4. Changes to the hardware of the tempering process such as the installation of heat recovery systems however was not part of the project.

2 DATA ANALYSIS

2.1 Tempering Process and Data Sources

Tempered or toughened glass is produced from standard glass in an oven where it is heated above the transition temperature to around 640°C. Afterwards

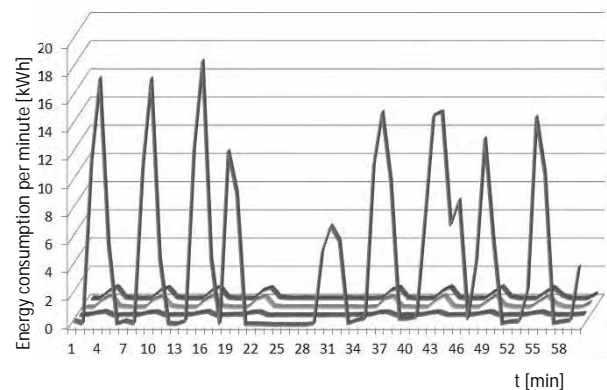


Figure 1: Power consumption of the tempering oven when the glass is rapidly cooled by blowing cool air onto the surface. This tempering process requires a lot of

heat which is usually generated from electrical energy. The heated air however is not recovered in heat exchangers or other processes. The oven which was used to gather the data has a power consumption of up to 2 MW for heating and up to 1 MW for three large fans that are used to cool the glass. Roughly 80% of the energy is required for the heating of the glass and the remaining 20% are used for cooling. The heating unit consists of 27 heating coils which are independently controlled (on/off).

In order to develop a black-box model of the oven and the tempering process a detailed data analysis and a large data pool is required. The only available data source at the start of the project was the ERP (enterprise resource planning) system, which stores job data for every sheet of tempered glass. An EMS (energy monitoring system) was not available and had to be installed before data collection could begin. This EMS records the energy consumption of the oven with its main components heating, fan 1, 2, and 3, storing one data point per minute (see Figure 1). A much more frequent data recording rate would have been very beneficial but the EMS did not provide that. The main purpose of the EMS is to obtain data used for the ISO 50001 certification. Data from both the ERP and the EMS are combined and imported into Matlab. Matlab is then employed for an in-depth data analysis which executes multiple steps to obtain characteristic indicators which can be used to train black-box models that are able to predict the total energy-consumption of a scheduling sequence based solely on the ERP data.

2.2 Data Preparation and Processing

The quality of the production data (ERP) is relatively poor w.r.t time stamps for oven batch data sets. Time stamps are generated whenever an operator scans a barcode that accompanies every glass sheet. The scanning procedure takes place before a glass sheet enters the oven and again after it leaves the cooling chamber. However, the accuracy of time stamps depends on the operator and what type of glass is being tempered. The tempering time is typically between 5 and 10 minutes but the time difference between the scan in and scan out time stamps varies between 5 and 90 minutes due to breaks, scanning of barcodes of multiple sheets in a row and other reasons.

For each batch (one or more glass sheets that are tempered together) there is a typical peak in the energy trend (see Figure 2). Due to the variations in time

stamps, it is not trivial to find the energy peak belonging to a single batch of glass. Within the time between the scan in and scan out time stamps the data from the EMS might show more than one peak. A Matlab script is then used to process energy and production data and to identify energy peaks (see Figure 2) and their corresponding production data set. A manual identification of energy peaks and production data would be feasible for data of up to a few hours production time. However, in order to analyze data from weeks or even months an automatic analysis is inevitable. The black-box approach we chose to model the oven behaviour requires statistical data analysis and therefore a large amount of sample data.

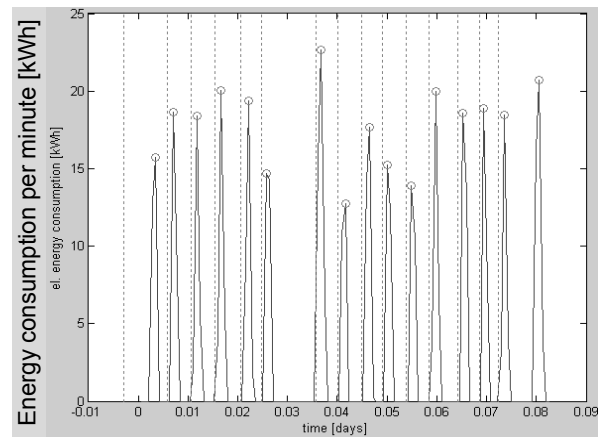


Figure 2 Identified peaks in el .power consumption

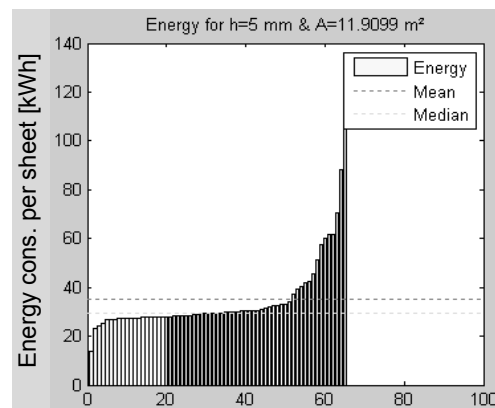


Figure 3 Cleaned data for glass type & dimension (idle load removed)

The key indicators required for characterizing the tempering process and creating a detailed model are as follows:

- Stand-by electrical power consumption of the oven (heating and fans) which depends on the standby temperature.

- Power consumption per glass volume. This depends on glass thickness and type, surface coatings and prints etc.
- Trend and typical durations for a transition between two temperatures for cooling and heating and for a change of the operating mode (off, standby and ready).
- Typical heat loss and effectiveness of the oven at different temperature levels.
- Typical tempering times for different glass types and dimensions. These can be obtained from the parameter set.

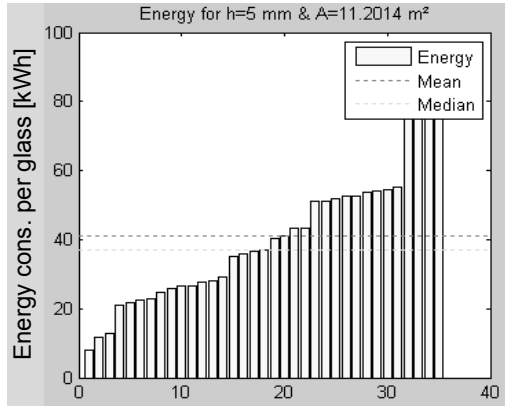


Figure 4 Insufficient and unreliable data

The most influential indicator is the required heating power consumption per volume. This is closely related to the specific heat capacity of the glass:

$$\Delta Q = c \cdot m \cdot \Delta T$$

$$\Delta Q = c \cdot \rho \cdot A \cdot h \cdot \Delta T$$

Because ΔT (temperature difference), c (specific heat capacity), ρ (density), A (glass area) and h (glass height) are known, the theoretically required heat for tempering the glass can be determined. However this value does not take heat and other losses into account and is only a fraction of the required electrical power.

A quick plausibility check reveals the efficiency of the oven. The theoretical heat required for heating up a sheet of glass ($c=0.84$ kJ/kg*K, $\rho=2.500$ kg/m³, $A=11.909$ m², $h=5$ mm) from 25°C to 650°C is 21.5kWh. As shown in Figure 3, the mean value of 65 production data sets is 35.1 kWh and the median is somewhat lower at 29.6kWh. These values do not

include the oven's base load. The efficiency factor of the oven for that glass type is therefore around 70%.

The aforementioned parameters are going to be established automatically for a wide range of glass types and dimensions and are used later on for training of the oven model.

Data analysis is employed to determine the aforementioned key indicators. First the energy data is analyzed and the idle load (energy required for keeping the oven temperature) of the oven is determined. This is done by identifying and eliminating all energy peaks due to glass being tempered and looking at the remaining ground load. The remaining consumption values are all in the range of 80kW which is the base load. This value is subtracted from all energy values so only the energy peaks due to production are remaining.

Afterwards the time distance between production data sets is analyzed. Data sets that are too close to each other due to human interaction are eliminated from the data pool. Only data sets that can be clearly assigned to an energy peak are taken into the data pool for training and parameterizing the model. As mentioned before the data points of the EMS have a relatively low resolution of 1/minute. In order to enhance the data set for training of the models interpolation is used to obtain energy trends with a resolution of 20 seconds.

3 THE MAIN COMPONENTS OF THE OPTIMIZATION AND SCHEDULING PLATFORM

The optimization and scheduling platform developed in OptPlanEnergie will consist of three main components that can be integrated into existing APS-systems.

3.1 MODELING OF THERMO-DYNAMIC PROCESSES

The energy-critical steps in the manufacturing of tempered glass are the uniform heating of the panes in a furnace to a specified temperature and subsequent the rapid cooling with forced air drafts using blowers. To produce high quality glass it is crucial to maintain the necessary temperature inside the furnace within a small tolerance throughout the whole processing time.

By modeling the characteristic heat absorption properties of preprocessed glass panes of various types

and dimensions we will be able to determine the optimal starting temperature of the furnace and the total energy consumption of any given sequence of production orders. As the optimizer used in the scheduling module needs to evaluate a large number of possible sequences, we rely on system identification techniques and black-box models trained on real data to quickly compute the expected energy demand with the necessary accuracy.

The oven models are linear FIR (finite impulse response) filters which are trained based on the preprocessed production and consumption data sets (see Section 2.2). These filters are suited for this purpose because of their finite response. Trained with measured data sets, cf. [1],[2], their response is very similar to the measured data as shown in Fig. 6.

Currently the oven model is designed in Modelica and exported as Functional Mock-Up Unit (FMU). The optimizer will feature a Functional Mock-Up Interface (FMI) with which the FMU is processed.

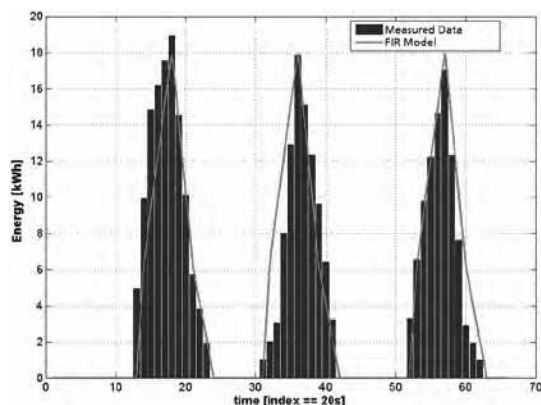


Figure 5 FIR filter response vs. measured data

3.2 OPTIMIZATION OF PRODUCTION SEQUENCES AND BATCH SIZES

Using the black-box models described above, the scheduling module will determine energy-efficient loading sequences for the furnace from a given set of production orders. Here it is necessary to take into account a wide area of practical constraints, such as the technical limitations of the furnace, the work schedules, and the number of available personnel.

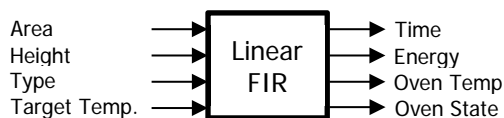


Figure 6 Model structure

The two main points that influence the energetic quality of a loading sequence are setup costs and idle times. Setup costs occur when the furnace has to be heated up or cooled down to the mandated starting temperature between two batches of glass. Avoidable idle times arise when loading the next batch takes longer than processing the current one.

The scheduling module will adapt multi-criteria search techniques from discrete optimization to utilize the energy data provided by the models. Its main goal is to compute feasible loading sequences and batch sizes that minimize setup and idle times to reduce total energy consumption and peak energy demands while respecting the constraints of the manufacturing process and potentially conflicting objectives like high output or compliance of delivery dates.

3.3 VISUALIZATION OF ENERGY CONSUMPTION AND SAVINGS POTENTIALS

The scheduling module will compute a set of pareto-optimal solutions that are presented to the user of the platform graphically. The Gantt-charts commonly used in planning applications to display schedules will be extended to include the energy-consumption over time of the different solutions.

This will enable planners to choose from the set of pareto-optimal solutions the one which balances the energy-efficiency of the schedule with possible other objectives.

4 References

- [1] R. Isermann und M. Münchhof. *Identification of Dynamic Systems*. Springer, Berlin Heidelberg, 2011.
- [2] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, London, 2012.

5 Acknowledgement

This research project is funded by



Bundesministerium
für Wirtschaft
und Energie

**Forschungsvorhaben:
03ET1162C**

Web-basierte Simulation zur Optimierung von Logistikstrukturen

Mathias BöS

SDZ GmbH, Dortmund

mathias.boes@sdz.de

Der Beitrag beschreibt eine web-basierte Anwendung für die Gestaltung von Supply Netzwerken und setzt sich dabei auch mit den Anforderungen an die Entwicklung web-basierter Anwendungen auseinander. Die Anwendung selber ermöglicht die Ermittlung einer optimalen logistischen Netzwerkstruktur, bezogen auf die Anzahl und Lage der logistischen Standorte. Die Anwendung ist nahezu vollständig automatisiert und bietet den Anwendern so einen maximalen Nutzen.

1 Motivation

Die Internationalisierung der Märkte sowie die weiter fortschreitende Ausweitung der Vertriebskanäle bestimmen die Anforderungen an die Wettbewerbsfähigkeit der Unternehmen. Gleichzeitig bieten diese Entwicklungen den Unternehmen die einmalige Chance, gezielt ihre Kundenmärkte und Marktanteile zu erweitern. Dies erfordert von den Unternehmen jedoch eine permanente Weiterentwicklung ihrer Prozesse sowie der Unternehmensorganisation. Innerhalb der dynamischen Zielmärkte werden immer höhere Anforderungen von Kunden an den Lieferservice gestellt. Neue Methoden der IKT führen zudem zu einer höheren Transparenz innerhalb der Märkte und steigern weiter den Kostendruck für die Unternehmen. Die Unternehmen begegnen diesen Entwicklungen durch die Gestaltung immer komplexerer und eng verketteter Wertschöpfungsketten.

Dieses Spannungsfeld führt dazu, dass immer stärker die gesamten Supply Chain Prozesse in den Fokus der Unternehmen rücken, um existierende Potenziale zur Steigerung des Lieferservice bei gleichzeitig einhergehender Kostensenkung identifizieren zu können. Die erste Herausforderung für die Unternehmen stellt sich bereits bei der Identifikation dieser Potenziale. Noch schwieriger ist für die Unternehmen allerdings die Bewertung von entwickelten Maßnahmen zur Ausschöpfung der erkannten Potenziale. Hier stehen die Verantwortlichen unter dem Druck, auf Anhieb die richtigen Maßnahmen umzusetzen und die erwarteten Verbesserungen zu realisieren.

Doch welche Potenziale bestehen tatsächlich? Und welche Maßnahmen garantieren die Erreichung der erwarteten Verbesserungen? Aufgrund der immanenten Komplexität der Supply Chain Prozesse sind diese Fragen für jedes Unternehmen nur schwer zu

beantworten. Denn das Umfeld ist hoch dynamisch und volatil. So müssen täglich unterschiedliche Sendungsmengen zu unterschiedlichen Belieferungsstandorten transportiert werden. Regelmäßig kommen neue Kundenstandorte hinzu während existierende wegfallen. Das Sortiment ändert sich ebenso wie die Belieferungsfrequenzen.

Um überhaupt existierende Potenziale identifizieren zu können sowie zur Bewertung der Wirksamkeit von entwickelten Maßnahmen bietet sich die Methode der Simulation an. Diese wird für Fragestellungen im Bereich des Supply Chain Designs bisher nur vereinzelt von Großunternehmen und Konzernen eingesetzt. Entsprechende Systeme sind zwar verfügbar, sind jedoch mit einem hohen Invest für die Unternehmen verbunden und müssen zudem umfangreich auf die jeweilige Fragestellung angepasst werden. Diese Systeme erfordern eine hohe Qualifikation der Anwender, wozu ein grundlegender Know-how-Aufbau im Unternehmen notwendig ist. Andererseits werden derartige Analysen nur unregelmäßig durchgeführt, müssen dann aber in kürzester Zeit zu einem Ergebnis führen (z.B. im Rahmen von Transportausschreibungen oder zur Analyse neuer Business-Cases).

An dieser Stelle setzt die in diesem Beitrag vorgestellte Methode an. Sie adressiert die Thematik der Analyse der Logistikstruktur und richtet sich an kleinere Unternehmen (KMU), sowohl an Logistikdienstleister als auch an Versender. Bei der Anwendung handelt es sich um eine Center of gravity-Analyse von Logistikstrukturen. Das neuartige an dieser Anwendung ist deren Umsetzung als web-basierte Anwendung. So entfällt die Notwendigkeit für den Kauf und die Installation einer komplexen Software für die Anwender. Die Anwendung kann mit allen gängigen Browser genutzt werden.

Sie soll Unternehmen bei der Analyse von Distributionsstrukturen unterstützen. Innerhalb von Supply Netzwerken ist die Lager von logistischen Standorten ein entscheidender Kostentreiber, an denen Waren für einen (oder mehrere) Kunden gelagert und kommissioniert werden und von wo sie zu Ablieferstellen distribuiert werden. Eine optimale Lage dieser Standorte hat für Unternehmen ökonomische, ökologische und Wettbewerbsvorteile. Je günstiger (in Hinsicht auf die resultierende Transportgesamtleistung) dieser Standort ausgewählt wird umso konkurrenzfähiger kann sich ein Unternehmen um derartige Geschäfte bewerben.

Die Anwendung ersetzt dazu Analysen, die bis heute manuell durchgeführt werden (entweder durch das Zeichnen von Kreisen auf Landkarten oder durch langwierige, manuelle Ermittlungen von Transportdistanzen über Tabellenwerke). Um zukünftig einen optimalen Standort zu finden oder die Lage ihres eigenen Standortes zu bewerten können Unternehmen die standardisierte Web-Anwendung nutzen. Auf diese Weise ist es möglich, Kosten- und Serviceaspekte bei der Suche nach dem optimalen Standort miteinander zu vergleichen.

2 Web-basierte Anwendungen

Bedingt durch die Entwicklungen im IKT-Bereich, insbesondere der Nutzung web-basierter Anwendungen mittels mobiler Devices, gibt es einen Wandel in der Nutzung von Anwendungen. Im privaten Bereich nutzt jeder inzwischen selbstverständlich sein Smartphone für web-basierte Kommunikationsanwendungen wie auch für Recherchen. Dieses Nutzerverhalten wirkt sich auch auf den geschäftlichen Bereich aus. Die Nutzer erwarten für ihren Aufgabenbereich geeignete Anwendungen, die ihre Tätigkeiten gezielt unterstützen und ihnen die Arbeit erleichtern. Vermehrt wird dabei auch eine umfassende „Eigenintelligenz“ der Systeme erwartet, die optimale und den Nutzern bis dato noch nicht bekannte Lösungsvorschläge machen. Für diese Nutzer ist es selbstverständlich, dass es sich dabei um web-basierte Anwendungen handelt. Diese Anwendungen verdrängen daher zunehmend die klassischen Desktop-Anwendungen.

Für die Anwender bieten sich durch die Nutzung web-basierter Anwendungen zahlreiche Vorteile. So stehen diese Anwendungen jederzeit und an jedem Ort zur Verfügung und können ohne Installationsaufwand vom Anwender genutzt werden. Es ist lediglich

ein aktueller Browser und der Zugang zum Internet erforderlich. Dabei wird vorausgesetzt, dass diese Anwendungen – wo notwendig – über weitere Dienste (Services) auf verfügbare Daten und Informationen zugreifen können.

Hindernisse für eine intensivere Nutzung web-basierter Anwendungen durch Unternehmen stellen sich hingegen noch durch den Aspekt der Datensicherheit. Dieser Aspekt soll in diesem Beitrag jedoch nicht betrachtet werden, da es sich hierbei um ein grundlegendes Thema im IKT-Bereich handelt.

Betrachtet werden müssen dagegen die Anforderungen von Anwendern an die Usability von Anwendungen, insbesondere in Hinsicht auf Anwendungen zur Simulation. Denn hier stellen sich ganz neue Anforderungen gegenüber den klassischen (stationären) Anwendungen der Simulation. Erwartet werden Anwendungen, die wenige Eingaben durch den Nutzer erfordern und die schnell Ergebnisse liefern. Auch die – gegenüber stationären Systemen – begrenzten Eingabemöglichkeiten mobiler Geräte führen zu Restriktionen, die zu Anforderungen an die Gestaltung web-basierter Anwendungen führen.

Eingabemöglichkeiten web-basierter Anwendungen basieren in der Regel auf vorgegeben Auswahlfeldern oder definierten Eingabedialogen. Diese grenzen die Auswahlmöglichkeiten einer Anwendung ein und limitieren diese auf einen definierten Anwendungsumfang. Als Folge davon sind web-basierte Anwendungen immer auf eine spezifische Aufgabe ausgerichtet und bieten daher nicht die Flexibilität, die stationäre Engineering-Systeme charakterisieren.

Bei der Entwicklung web-basierter Anwendungen müssen zudem auch technische Aspekte berücksichtigt werden. Im Gegensatz zu konventionellen (Desktop) Anwendungen gliedern sich web-basierte Anwendungen in eine Anwendungsoberfläche (Frontend) und die Anwendungslogik (Backend, inklusive Datenbank). Über das Frontend wird die Anwendung bedient und Ergebnisse dargestellt. Über das Backend erfolgen die Verarbeitung von Eingaben und Daten und die Ermittlung der Ergebnisse. Beide sind über das Internet miteinander gekoppelt, so dass auch die Übertragungsgeschwindigkeit zwischen beiden Komponenten den Umfang der Anwendungen beschränkt. So begrenzen beispielsweise häufige Interaktionen beider Komponenten die Performance der Anwendung ebenso wie komplexe Animationen oder Visualisierungen.

Diese Anforderungen erschweren die Konzeption von web-basierten Anwendungen für Logistik-Simulationen. Denn einerseits sollen sie schnell, ohne viel Eingabeaufwand zu Ergebnissen führen. Andererseits müssen diese Anwendungen auch die Individualität jedes Unternehmens berücksichtigen. Dies erfordert wissensbasierte Anwendungen, die auf spezifischen, konfigurierbaren Simulationsmodellen basieren.

Mit diesem Ansatz können ganz neue Nutzergruppen für die Simulation gewonnen werden, insbesondere bei jüngeren Generationen. Diese Nutzer haben eine ganz andere Erwartungshaltung an diese Anwendungen. Sie erwarten fundierte Lösungsvorschläge und möchten diese – ganz im Sinne einer Gamifizierung – ‚spielerisch‘ analysieren oder weiterentwickeln. Dabei gibt der Nutzer Variationen des Simulationsmodells vor und die Anwendung bewertet die Wirksamkeit dieser Variationen.

Um diesen Anforderungen gerecht werden zu können sind diese Anwendungen auf jeweils eine konkrete Aufgabenstellung ausgerichtet. Sie erfordern ein standardisiertes Abbildungsmodell mit definierten Variationsmöglichkeiten. Die Bewertung von Variationen muss automatisch erfolgen und dem Anwender direkt die Wirksamkeit einer Variation aufzeigen. Diese konkrete Ausrichtung auf eine Aufgabe begrenzt allerdings die Nutzungsmöglichkeiten einer Anwendung. Als Folge davon erfordern Abwandlungen einer Aufgabe die Entwicklung neuer, abgewandelter Anwendungen. Dies bedeutet eine Abkehr von den doch eher generisch gestalteten stationären Simulationssystemen, die dem Anwender eine größere Flexibilität bei den Anwendungen erlauben. Diese Flexibilität wurde in der Vergangenheit jedoch durch höhere Aufwände bei der Modellierung erkauft.

3 Anwendungsbeispiel

Die SDZ GmbH hat im Rahmen verschiedener Verbundforschungsprojekte neue web-basierte Planungs- und Simulationsanwendungen entwickelt und diese auf einer Plattform zusammengefasst. Für die Anwendung, die diesem Beitrag zugrunde liegt, wurde ein bestehender Lösungsansatz von der SDZ GmbH für ein web-basierte Planungs- und Simulationssystem konzeptionell weiterentwickelt und umgesetzt.

Diese Anwendung (Median.Web) umfasst einen standardisierten Anwendungsfall für die Analyse von Standortstrukturen im Rahmen der Gestaltung von logistischen Netzwerken. Dabei handelt es sich um

Distributionsnetzwerke mit regelmäßigen Transporten zu unterschiedlichen Belieferungspunkten. Die Anwendung ist darauf ausgerichtet, die Suche nach einem oder mehreren service-optimalen Logistikstandorten für das Supply Netzwerk zu unterstützen. Die Bewertung kann dabei nach betriebswirtschaftlichen oder logistischen Aspekten erfolgen.

Die Anlieferstellen können in ganz Deutschland verteilt sein. Für die Standortsimulation werden als Basisdaten (Inputdaten) lediglich die Adresse jeder Anlieferstelle sowie die jährliche Transportleistung benötigt. Lagerstandorte können vorgegeben werden, um deren Eignung als Netzknoten bewerten zu können. Möglich ist aber auch die Suche nach optimalen Lagerstandorten, wozu das Suchraster durch den Anwender definiert werden kann.

Weitere Vorgaben für Inputdaten sind nicht erforderlich, lediglich die Such- und Optimierungskriterien müssen vom Anwender noch eingestellt werden. Dafür können Optimierungskriterien (zum Beispiel die Gewichtung von Einflussfaktoren wie der Entfernung) durch den Anwender definiert werden.

Abbildung 1: Dialog zur Spezifikation der Simulation

Um Entfernungen ermitteln zu können erfolgt zunächst eine automatische Geo-Kodierung der übergebenen Adressen durch die Anwendung. Die Adressen können dazu im MS-Excel-Format für die Simulation bereitgestellt werden. Auf Basis der Geo-Daten aller

Standorte (Lager- und Belieferungsstandorte) können Entfernungen (Straßendistanz) zwischen Standorten ermittelt werden.

Bei dem gewählten Verfahren zur Optimierung der Standortstruktur handelt es sich um eine Schwerpunktanalyse (Center-of-Gravity). Diese basiert auf den jährlichen Transportmengen je Belieferungsstandort und ermittelt für jeden Belieferungsstandort die resultierende Transportleistung für alle vorgesehenen Lagerstandorte. Für jede Standortvariante wird so eine Gesamt-Transportleistung ermittelt, die nach unterschiedlichen Kriterien bewertet werden kann. Dazu zählen die reinen Transportdistanzen (Kilometer), die Transportkosten (€) oder die resultierenden Emissionen (Kilogramm CO₂).

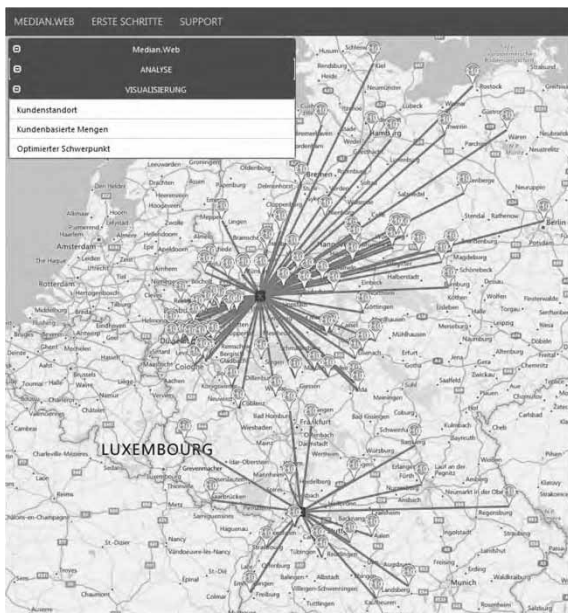


Abbildung 2: Ergebnisdarstellung der Logistik- und Belieferungsstandorte

Der Anwender kann für jeden Simulationslauf zudem vorgeben, wie viele Standorte die Simulation in Betracht ziehen soll. In diesem Fall erfolgt eine Zuordnung jedes Belieferungsstandortes zu einem Standort, so dass pro Standort die resultierende Transportleistung ermittelt werden kann. Um eine optimale Standortstruktur für mehrere Standorte zu ermitteln erfolgt die Analyse unterschiedlicher Kombinationen möglicher Standorte je Simulationslauf. Dazu kommt ein mathematisches Verfahren (Teitz-Bart-Heuristik) zur Anwendung, um in kurzer Zeit möglichst optimale Lösungen innerhalb des gesamten Lösungsraums zu identifizieren. Auf diese Weise kann auch ein Ver-

gleich von zentraler und dezentraler Standortstruktur erfolgen.

Die Ergebnisse werden direkt nach Durchführung eines Simulationslaufes grafisch auf digitalen Karten dargestellt. Der Anwender erkennt so die ermittelten Lagerstandorte sowie die Zuordnung von Belieferungspunkten zu den Lagerstandorten.

Für einen Lagerstandort können zudem noch alternative Standorte dargestellt werden, die je nach Eignung klassifiziert dargestellt werden (grün-gelb-rot). Der Anwender kann so die Eignung verfügbare Standorte prüfen, die ihm angeboten werden (z.B. von Kommunen oder Wirtschaftsförderungsgesellschaften).

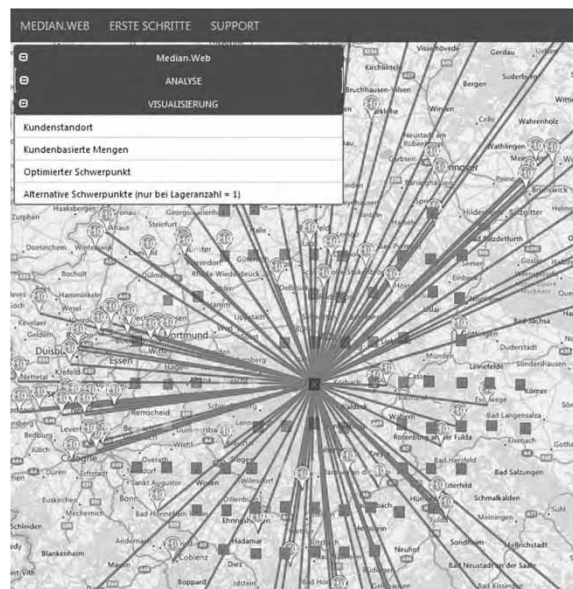


Abbildung 3: Darstellung alternativer Lagerstandorte, klassifiziert nach Eignung

Die Anwendung stellt noch weitere Ergebnisse zu der ermittelten Transportleistung in Form von Kennzahlen dar, um die Simulationsläufe auswerten und vergleichen zu können. Ausgewiesen werden die ermittelten Entfernungen aller Relationen (Summe Distanzen aller Belieferungspunkte zu zugeordnetem Lagerstandorten), die Transportkosten (bezogen auf ein Jahr) und die ermittelten Emissionen (Kilogramm CO₂). Für die ermittelten Lagerstandorte werden die Geo-Koordinaten ausgewiesen.

Ergebnis		
Summe Entfernung (km)	16.145,00	
Summe Transportkosten (€)	17.017.125,00	
Summe Emissionen (kg)	22,20	
Lagerstandort	Breitengrad	Längengrad
Warehouse-017-007	51,72	7,97
Warehouse-005-010	49,09	8,74

Abbildung 4: Ausgewiesene Ergebnisse je Simulationslauf

Für eine weitere Transparenz der Leistungsanforderungen der Transporte können die Belieferungsstandorte auch mit den gewichteten Transportleistungen je Belieferungsstandort dargestellt werden. Der Anwender kann durch diese Darstellung auch visuell die Ergebnisse der Simulation nachvollziehen. Mögliche Schwerpunkte innerhalb des Belieferungsnetzwerkes können durch die gewichtete Darstellung je Belieferungsstandort und evtl. Überlagerungen dieser Darstellungen sehr gut nachvollzogen werden.

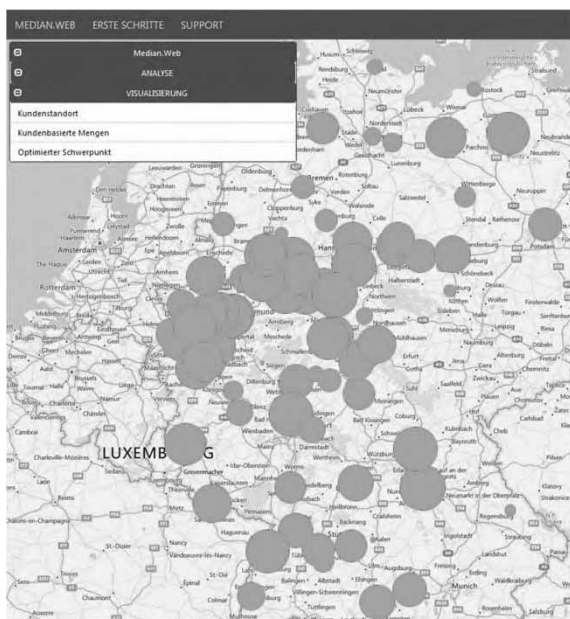


Abbildung 5: Gewichtete Darstellung aller Belieferungsstandorte

4 FAZIT

Der Einsatz web-basierter Anwendungen für Logistik-Simulationen wird zunehmend von Nutzern in Industrie und Handel gefordert. Diese Anwendungen müssen auf eine schnelle und einfache Nutzung ausgelegt werden. Erwartet werden Anwendungen, die logistisches Wissen enthalten und auf dieser Grundla-

ge optimale Lösungsvorschläge ermitteln, sowohl in ökonomischer, zunehmend jedoch auch in ökologischer Hinsicht.

Web-basierten Anwendungen, die insbesondere für die Nutzung über mobile Endgeräte konzipiert sind, müssen dazu auf spezifische Aufgaben ausgerichtet sein, um den Anforderungen der Nutzer in Punkto Usability gerecht zu werden. Gefordert wird eine möglichst intuitive Bedienung mit möglichst wenig Eingabeaufwand.

Als Folge davon handelt es sich bei den Anwendungen eher um konfigurierbare, direkt nutzbare Simulationsmodelle als um Simulationsprogramme mit der Notwendigkeit zur Erstellung von Simulationsmodellen. Diese Anwendungen sind daher auf einen spezifischen Use-Case ausgerichtet. Variationen sind nur innerhalb des vorgegeben Modells möglich. Erweiterte Anwendungsoptionen erfordern eine Modifikation bzw. Erweiterung der Anwendung durch den Entwickler.

Für den Entwickler bzw. Anbieter derartiger Anwendungen ergeben sich allerdings noch weitere Herausforderungen. So können im Internet nicht die klassischen Geschäftsmodelle genutzt werden, die nutzerabhängige Lizenzmodelle beinhalten. So existiert im Internet eine Kultur der kostenlosen Nutzung von Services und Dienstleistungen. Zudem beschränkt die begrenzte Funktionalität der Anwendungen deren breiteren Einsatz in Unternehmen, was allerdings bei Erhebung von Lizenzgebühren von den Unternehmen gefordert würde.

Nötig sind daher neue Geschäftsmodelle für web-basierte Anwendungen. Diese könnten derart gestaltet werden, dass Entwickler ergänzenden Support für diese Anwendungen kostenpflichtig anbieten oder ein individuelles Customizing der Anwendungen.

Auch das Marketing und der Vertrieb dieser Anwendungen muss ganz neu gedacht werden und wird ausschließlich im Internet erfolgen. Um eine möglichst hohe Zahl an Nutzern zu gewinnen bietet sich eine Verbreitung über Partner an, beispielsweise über Betreiber branchenspezifischer Portale. Die Anwendungen könnten in diese Portale integriert werden, was einerseits die Attraktivität dieser Portale erhöht und andererseits die Bekanntheit der Anwendungen steigert.

Ein Beispiel dafür beinhaltet das Internetportal ‚Lagerfläche.de‘ für die Vermarktung von freien Lagerflächen.

LAGER[flaeche]

Serviceplattform für Lagerlogistik

Die Anwendung ist vollständig in das Portal integriert und steht allen Besuchern des Portals zur Nutzung zur Verfügung.

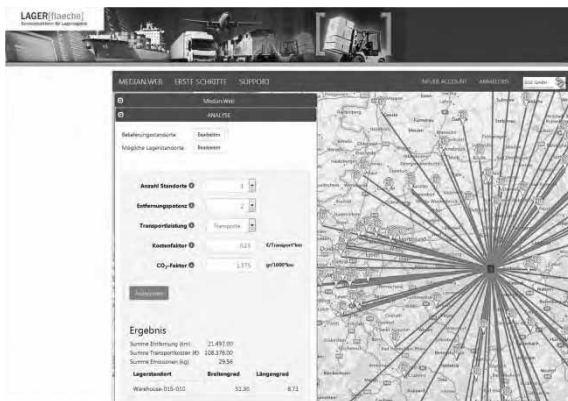


Abbildung 6: Integration von Anwendungen in Internetportale

Möglich ist dazu auch eine spezifische Modifikation von Anwendungen für derartige Integrationen, z.B. die direkte Übernahme verfügbarer Flächen für Lagerstandorte, die über das Portal vermarktet werden.

5 Literaturverzeichnis

- [1] Kühn, W. (2006): Digitale Fabrik, Fabriksimulation für Produktionsplaner, Carl Hanser, München, Wien
- [2] März, L. (2011): Simulation und Optimierung in Produktion und Logistik. Praxisorientierter Leitfaden mit Fallbeispielen. Berlin: Springer.
- [3] VDI (2000): Richtlinie 3633 Blatt 5: Simulation von Logistik-, Materialfluß- und Produktionssystemen, Integration der Simulation in die betrieblichen Abläufe. Beuth, Berlin
- [4] VDI (2008): Richtlinie 4499 Blatt 1: Digitale Fabrik - Grundlagen. Beuth, Berlin
- [5] VDI (2010): Richtlinie 3633 Blatt 1: Simulation von Logistik-, Materialfluß- und Produktionssystemen - Begriffsdefinitionen. Beuth, Berlin
- [6] Wenzel, S., Weiss, M., Collisi-Böhmer, S., Pitsch, H., Rose, O. (2008): Qualitätskriterien für die Simulation in Produktion und Logistik: Planung und Durchführung von Simulationsstudien, Springer Verlag, Berlin, Heidelberg
- [7] Wenzel, S., Bernhard, J. (2008): Definition und Modellierung von Systemlasten für die Simulation logistischer Systeme, in: Nyhuis, P. (Hrsg.): Beiträge zu einer Theorie der Logistik, Springer Verlag, Berlin, Heidelberg, S. 487-509
- [8] Wenzel, S. (2010): VDI-Richtlinien zur Modellbildung und Simulation. In: Zülch, G.; Stock, P. (Hrsg.): Integrationsaspekte der Simulation: Technik, Organisation und Personal, KIT Scientific Publishing, Karlsruhe, S. 543-550.

A Framework for the Metamodeling of Multi-variant Systems and Reactive Simulation Model Generation and Execution

Thorsten Pawletta¹, Artur Schmidt¹, Umut Durak², Bernard P. Zeigler³

¹Hochschule Wismar – University of Applied Sciences, RG CEA, Wismar, Germany

²DLR Institute of Flight Systems, Braunschweig, Germany

³RTSync Corp. and ACIMS Tucson/AZ, USA

thorsten.pawletta@hs-wismar.de

The simulation-based study of Cyber-Physical Systems or complex production systems leads often to a vast number of system variants. Each system variant is characterized by a particular model structure and parameter settings, although system variants may also share common parts. There are two main approaches for modeling such a set of system variants. On the one hand, all variants are mapped in a big model with variation points and on the other hand variants are specified on a higher level of abstraction using a metamodel that is processed with appropriate transformation methods. This paper proposes an approach for modeling system variants using the System Entity Structure (SES) Ontology. It introduces new concepts and advances the SES by a procedural knowledge specification. Moreover, it proposes a software infrastructure for the automated and reactive generation and execution of simulation models based on a SES in combination with a model base. Finally, it refers to a prototype implementation within MATLAB/Simulink and forward-looking within Python.

1 Introduction

The study of multifaceted end user requirements of Cyber-Physical Systems or of multi-variant production systems leads to a vast number of system variants. Both problem types can be considered as a variability problem. Variability has been defined as the ability of a system or an artefact to be configured, customized or extended for employment in a particular context [1]. In software engineering Software Product Lines (SPL) are widely employed for developing systems that are characterized by a high degree of variability. SPL define variation points where different variants of products can be derived for varying requirements [2]. Variability management has also been introduced as a challenge to be tackled for model-based testing architectures [3], for model-based concept development tools [4] and for studying multi-variant production systems [5] or reactive robot controls [6]. In this context the problem of reactivity is also discussed [5, 6, 7]. We will consider reactivity as the generation and execution of a new system variant depending on current and previous results.

Variability mechanisms shall be defined at particular levels of abstraction, ranging from metamodeling to implementation of the source code. Using a metamodel for variability modeling requires appropriate model transformation methods for mapping to the

execution level. This is a particular challenge, because such methods are not supported by the established modeling and simulation environments used in the engineering or production system domain. Another approach is the combination with software tools for requirement or variant management [8] or with domain oriented tools [4]. However, in this case often different kinds of models have to be maintained and kept consistent.

For these reasons, in engineering and production system applications variability is still often encoded within the executable system models. However, these models are often hard to manage. Therefore, specific modularization and configuration methods have been developed to tackle the complexity. From the simulation theory the approach of dynamic structure or variable structure systems [9, 10] is known. On the other side, rather pragmatic solutions have been developed, such as for the MATLAB/Simulink environment in [11, 12].

For the modeling and simulation of modular, hierarchical systems, Zeigler introduced the *System Entity Structure* (SES) for specifying a set of system configurations, called a family of systems. The SES approach has evolved steadily to an ontology for model and data engineering [13, 14]. In combination with a model base (MB), organizing a set of configurable

basic models, the SES approach has been advanced to a modeling and simulation framework (SES/MB) [15]. In this paper, a reworked version of [16], we extend the SES ontology by adding new features. In addition, we advance the SES/MB framework to an infrastructure for reactive model generation and simulation execution and we refer to a prototype implementation. Using an exemplified multi-variant engineering problem, a concrete SES model, which is a metamodel, is developed. Based on the example, fundamental elements and axioms of the baseline SES ontology are briefly summarized. Next, some extensions to the SES ontology are discussed. The main new features are SESVariables and SESFunctions, which expand the SES ontology by procedural knowledge elements. After that, the selection of a concrete system variant from an SES metamodel is considered and the whole procedure for generating an executable simulation model is depicted under the aspect of reactivity.

2 Multi-variant Engineering Example

The example is an extension of an application that has been introduced by The MathWorks in [12] to demonstrate features for variant modeling within MATLAB/Simulink. We use that example to make our approach comparable with The MathWorks solution for experienced users. The substantial problem statement is illustrated in Figure 1a. Different controller (ctrl) designs, based on a linear (lc) or a nonlinear (nc) control structure, should be investigated using different signal sources from a signal generator (sg). In addition to the control structure, the signal types {sine | ramp | step} and the number of signal sources {1...3} may vary. Figure 1a shows the two control approaches (lc_ctrl | nc_ctrl) as alternative submodels of model ctrl. Due to the varying number of possible input signals, both approaches lead to three different internal model structures. The minimal internal structure of a ctrl model with one input signal is illustrated with full lines. The extension for two or three input signals is pictured with dashed lines. In the same manner, the internal structure of the system generator (sg) depends on the number and type of included signal sources. Overall, the exemplary problem comprises $(3^1 + 3^2 + 3^3) * 2$ various system structures. All possible system structures can be aggregated using 7 basic systems. In this case, the basic systems are blocks from the Simulink blockset, which represent a *model base* (MB), as shown in Figure 1b.

Figure 1c illustrates as an example the model structure of a specific system variant, which we call a *model under study* (MUS). In this case, the MUS consists of an nc_ctrl model, which is influenced by a sg model with three signal sources. Two sources are of type *sine* and one of type *step*. For simplification purposes, a separation between MUS and experimental frame (EF) according to [15] is not considered.

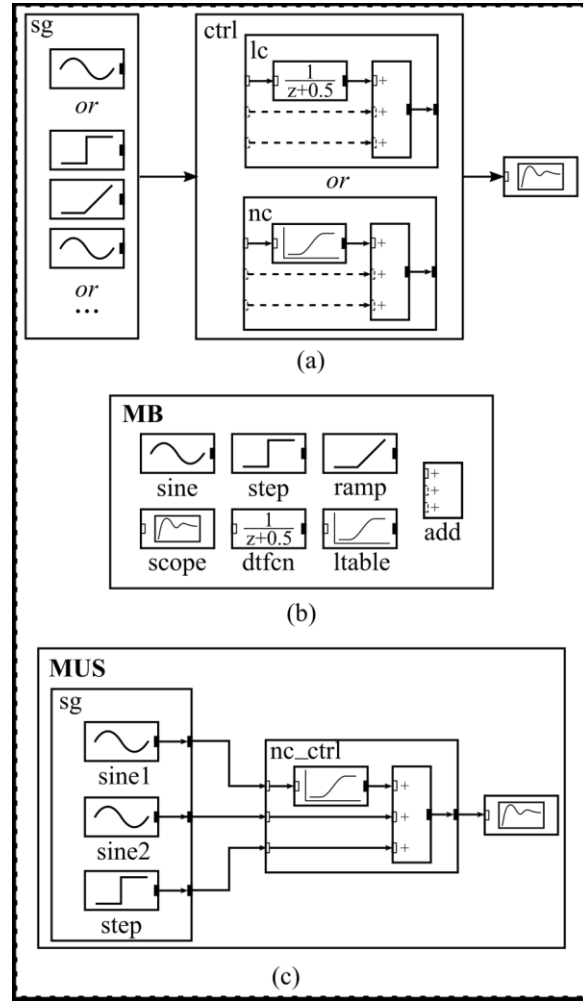


Figure 1. (a) Overall engineering problem with a set of system variants; (b) Blocks from the Simulink blockset representing the MB; (c) Model structure of a concrete system variant.

3 Metamodel-based Variant Modeling

This section describes the specification of the exemplary problem to demonstrate multi-variant modeling using an SES. The specification is based on the baseline SES definitions in [13], but it uses some modifications based on former works in [17] and introduces some new concepts, such as the *SESFunctions*.

3.1 SES fundamentals and SESVariables

The SES ontology supports the description of a family of systems regarding their elements and the relations between them. It is axiomatically defined and can be represented as a directed labeled tree. Nodes are divided into two types, *entity* and *descriptive* nodes, which can define specific *attributes*. Entity nodes describe system elements and the system itself (root node). The leaf nodes are always entity nodes, whose attributes define a link to a basic model in the MB (attribute *mb*) and possible parameter settings for the referenced basic model. Descriptive nodes express relationships between entities and are subdivided into: *aspect*, *specialization* and *multi-aspect* nodes. The SES axioms will be considered subsequently, as necessary for the example. Figure 2 illustrates an SES tree that maps the problem described in Section 2. In the tree descriptive nodes are marked with name suffixes: (i) DEC for aspect, (ii) SPEC for specialization and (iii) MULT for multi-aspect. At this point the SES axiom *alternating mode* for entity and descriptive nodes should be noted.

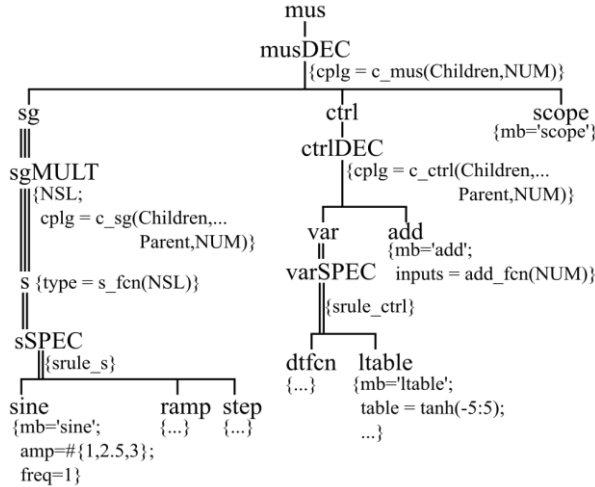


Figure 2. SES tree for the example in Figure 1a

Before describing the SES tree in detail, the new concept of *SESVariables* as the input interface of an SES is explained. This new feature was introduced to support the integration of an SES metamodel, referring to the metamodel definition in [18], in the later suggested infrastructure. In the infrastructure the selection of a particular system variant depends on the current settings of *SESVariables*. The selection procedure itself is described in the next section. *SESVariables* have a global scope and are written in uppercase letters in the tree. Two *SESVariables* in the tree in Figure 2 are defined as input arguments and a

third one as an auxiliary variable. They have the following definitions:

```

SESVariables={SPEC_CTRL,NSL} with
SPEC_CTRL ∈ {'lc','nc'}
NSL ∈ {(i),(i,j),(i,j,k)}
i ∈ {'sine[x]','ramp','step'}^
j ∈ {'sine[x]','ramp','step'}^
k ∈ {'sine[x]','ramp','step'}^
x ∈ {1,2,3}
auxiliarySESVariable={NUM} with
NUM=numel(NSL)

```

According to the exemplary problem (see Fig. 1), the variable *SPEC_CTRL* encodes the desired control structure and the variable *NSL* specifies a list with the signal sources to be selected. The index value *x* allows the encoding of different parameter selections for a *sine* signal. The auxiliary variable *NUM* calculates the current number of elements (*numel*) in *NSL*. An example for an admissible value assignment to *SESVariables* is given as follows.

```

NSL ={'sine[1]','sine[2]','step'}
SPEC_CTRL={'nc'}
→ NUM =3

```

3.2 Decomposition of systems with variable coupling relations

The system itself (*mus*) is represented in the SES tree with the root node. The subsequent aspect *musDEC* and vertical lines define a decomposition of *mus* (parent) in the entities *sg*, *ctrl* and *scope* (children). The aspect attribute *{cplg=...}* defines the coupling relations of *mus*. Model couplings can be divided into internal couplings (IC) between children, and external input as well as external output couplings (EIC, EOC) between the parent and its children. However, a coupling relationship always has the following structure:

```
{'SrcEntity','FromPrt','SinkEntity','ToPrt'}
```

In the example some ICs of entity *mus* depend on the number of signal sources defined by *sg* (see Fig. 1). To express such dynamics with minimal effort and to keep a lean SES tree, the concept of *SESFunctions* has been introduced. The *SESFunctions* are like ordinary functions. They extend the declarative specification defined by the baseline SES by procedural knowledge descriptions. *SESFunctions* are calculated during the processing of an SES, called *pruning*, and are described in the next section. This means that the terms:

```

cplg=c_mus (Children,NUM)
cplg=c_ctrl(Children,Parent,NUM)

```

represent ordinary function calls that return the coupling relations, which depend on the current settings of the input arguments. The variables *Children* and *Parent* are implicit attributes of each tree node, which save the names of the successor (left-to-right) and predecessor nodes. Hence, the set of variable couplings of entity *mus*, derived from the overall problem illustrated in Figure 1, can be defined using the following SESFunction (in MATLAB syntax):

```
function cplg=c_mus(children,num)
%create empty data structure for couplings
cplg=cell(num+1,4);
%set variable ICs btwn sg & ctrl
for i=1:num %for 1 to num
    cplg(i,1:4)={children{1},num2str(i),...
        children{2},num2str(i)};
end
%set fixed IC btwn ctrl & scope
cplg(num+1,1:4)={children{2},'1',
    children{3},'1' };
end
```

The children *sg* and *ctrl* of *mus* are composed entities, while *scope* is an atomic entity. Leaf node *scope* maps a basic system in the SES and defines with its attribute *mb*=*'scope'* a corresponding link to the MB. The decomposition of entity *ctrl* in the entities *var* and *add* is specified by its successor node *ctrlDEC*. In both control approaches, the linear and nonlinear (see Fig. 1), the coupling relations of *ctrl* depend on the number of external inputs, which again depend on the current number of signal sources. Thus, the coupling relations at *ctrlDEC* are specified by an SESFunction analogous to node *musDEC*.

3.3 Variable system attributes and the specialization of systems

Leaf node *add* represents a basic model, such as node *scope*. In contrast to *scope*, it defines a variable attribute for parameter settings, using the SESFunction call *inputs=add_fcn(NUM)*. As illustrated in Figure 1, the configuration of *add* depends on the number of inputs. This problem is specified with the following SESFunction (MATLAB syntax):

```
function inputs=add_fcn(num)
inputs(1)='|'; inputs(2:num+1)='+';
end
```

The characteristic of entity node *var* is specified by the succeeding specialization node *varSPEC*, marked with double-line edges. A specialization describes an *is-a-relation* concerning the succeeding nodes; in this case, entity *var* can be *dtfcn* or *ltable*. While processing an SES, the selection is controlled by evaluat-

ing selection rules that are specified as node attribute. In this case the following rule is defined.

```
srule_ctrl={SPEC_CTRL=='lc' → dtfcn |
    SPEC_CTRL=='nc' → ltable}
```

For specializations the specific SES axiom *inheritance* is defined. Its effects will be explained in the next subsection. The leaf nodes *dtfcn* and *ltable* represent once again basic models. The node *ltable* shows a further example for a variable attribute definition.

3.4 Variable decomposition of systems

According to the problem description in Section 2, the node *sg*, following the aspect *musDEC*, represents a system entity composed of a variable number of signal sources of various types. Referring to the baseline SES definition, such selection and composition has to be specified using a combination of aspect or multi-aspect and specialization nodes, possibly supplemented by selection constraints. However, this approach quickly leads to a confusing SES tree. In the following, an approach for keeping a lean SES tree will be described.

In former work [17] regarding concepts of SES, a first idea for solving this specific problem was discussed under the constraint of relaxing the SES *strict hierarchy* axiom. Based on this idea, we will suggest a complete solution without the violation of the strict hierarchy axiom. In Figure 2 the entity *sg* is characterized by the succeeding multi-aspect *sgMULT* with triple-line edges. According to the baseline SES definition, a multi-aspect is a special case of an aspect in which the succeeding entities are homogeneous in nature. Thus, it has only one succeeding entity node and defines the number of replications of this node as an attribute. Accordingly, the node *sgMULT* has one succeeding entity node named *s*. However, the node attribute definitions of the multi-aspect *sgMULT* and the succeeding entity *s* are more complex referring to the baseline SES definition. Node *sgMULT* specifies in the SESVariable *NSL* a list of types for replication. The number of replications is implicitly specified by the number of list elements. Remember the example

```
NSL={'sine[1]','sine[2]','step'}
```

stated at the end of Subsection 3.1. Furthermore, *sgMULT* defines variable coupling relations using the SESFunction call *cplg=c_sg(...)*, analogous to the aspect nodes *musDEC* and *ctrlDEC*.

The entity s specifies an attribute $type$. The concrete value of this attribute is determined by calling the SESFunction $s_fcn(NSL)$ when processing the SES. The SESFunction s_fcn defines a simple iterator.

```
function [type]=s_fcn(NSL)
persistent idx %static variable
if isempty(idx), idx=1; end %init iterator
type=NSL(idx); idx=idx+1;
if numel(NSL)==idx, idx=1; end %reinit iter
end
```

Thus, for each replication of entity s an individual value assignment is made, such as in our supposed case $type='sine[1]'$, $type='sine[2]'$ and $type='step'$, when processing the SES. Based on the setting of attribute $type$, replications of entity s can be specialized using a succeeding specialization node. This is specified in the SES tree with the node $sSPEC$, which defines the various signal sources as succeeding entities and the following selection rule as its attribute.

```
srule_s={Parent.type=='sine[x]' → sine[x] |
Parent.type=='ramp' → ramp |
Parent.type=='step' → step}
```

This means that the selection at $sSPEC$ depends on the value assignment to attribute $type$ at the parent node of $sSPEC$. Details of this subject will be discussed in the next section (see Fig. 4).

The leaf node entities $sine$, $ramp$ and $step$ once again represent basic systems, which specify a link to the MB and parameter configurations. The attribute $amp=\{1,2,5,3\}$ of entity $sine$ defines an ordered multiset for different parameter configurations. Therefore, specifications referring to a $sine$ signal source are extended by the index x to choose an element from the multiset amp .

4 Selecting a Distinct System Variant

An SES, such as in Figure 2, codes a set of system variants and is a metamodel referring to the definition in [18]. For simulation studies a single or several distinct system variants must be derived from the SES metamodel. The selection of a particular model structure, including parameter settings for basic models, depends on the current settings of SESVariables and the selection itself is performed by graph pruning. The result of pruning is a decision-free tree, called *Pruned Entity Structure* (PES), which contains all of the necessary knowledge for building a distinct simulation model using basic models from the MB. Figure

3 shows one PES derived from the SES in Figure 2 using the described value assignments to the SESVariables.

```
NSL={'sine[1]','sine[2]','step'}
SPEC_CTRL={'nc'}
→ NUM=3
```

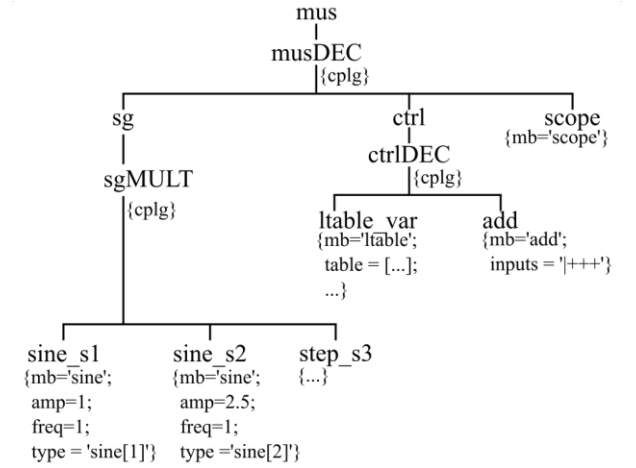


Figure 3. PES derived from SES in Figure 2

This PES codes a system structure analogous to the MUS in Figure 1c. Subsequently, we will describe the pruning operation in detail.

Starting at the root node of the SES, the first decision operation occurs at aspect $musDEC$. The SESFunction called $cplg=c_mus(Children,3)$ is executed to determine the coupling relations. The result is:

```
musDEC.cplg={'sg' , '1' , 'ctrl' , '1';
'sg' , '2' , 'ctrl' , '2';
'sg' , '3' , 'ctrl' , '3';
'ctrl' , '1' , 'scope' , '1' }
```

The next decision point is at multi-aspect $sgMULT$. According to the number of elements in SESVariable NSL , the entity s , including its following sub-tree, has to be replicated three times. During this operation replicas of s are renamed to comply with the *valid brothers* axiom. Moreover, any replica is assigned an exact value to its attribute $type$ by executing the iterator SESFunction s_fcn with the input argument $NSL=\{'sine[1]','sine[2]','step'\}$. The results of this operation are the replicated and renamed entities $s1$, $s2$, $s3$ with their identical sub-trees but an individual value assignment to their attribute $type$, as illustrated in Figure 4. Now, for each entity s_i the replicated sub-tree is evaluated. This means that the selection rule $srule_s$ is evaluated for each node $sSPEC$. In our case, it delivers the following selection $sine[1]$, $si-$

ne[2] and *step*. Remember, the indices of *sine* denote the parameter selection for the multiset of attribute *amp*.

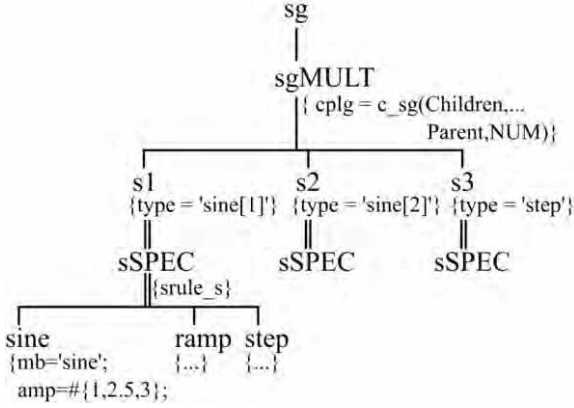


Figure 4. Part of 'intermediate PES' (sub-tree of *sg*) during pruning of multi-aspect *sgMULT*

Next, the parent and child entity of each specialization relation is combined according to the *inheritance* axiom. In this case, only the entity names and attributes have to be combined, e.g. *sine_s1*{*mb*='sine';...;*type*='sine[1]'}'. Finally, the current coupling relations, specified at node *sgMULT*, are determined by executing the SESFunction call *cplg=c_sg(Children, Parent, NUM)*. The result is:

```
sgMULT.cplg={ 'sine_s1', '1', 'sg', '1';
  'sine_s2', '1', 'sg', '2';
  'step', '1', 'sg', '3' }
```

The sub-tree of entity *ctrl* in Figure 2 is resolved in a similar manner during pruning. The resulting coupling relations for entity *ctrlDEC* are the following:

```
ctrlDEC.cplg={ 'ctrl', '1', 'ltable_var', '1';
  'ltable_var', '1', 'add', '1';
  'add', '1', 'ctrl', '1';
  'ctrl', '2', 'add', '2';
  'ctrl', '3', 'add', '3' }
```

As mentioned in the beginning, the PES contains all of the necessary knowledge for building a simulation model using basic models from the MB. Sometimes, the PES contains unnecessary attributes due to the pruning operation, such as *type* in the entities *sine_s1* and *sine_s2*, which can be neglected when building the simulation model. Moreover, referring to [15], the PES can be *flattened* by restructuring. Then, in our case the inner nodes *sg*, *sgMULT*, *ctrl*, *ctrlDEC* are resolved and all coupling relations are restructured in the *cplg* attribute of aspect *musDEC*.

5 Software Infrastructure and Prototype Implementation

Figure 5 shows the proposed infrastructure for multi-variant modeling and reactive model generation and execution. The core element is the SES/MB framework according to [15], which is extended by an input and output interface using the introduced SESVariables. This part of the infrastructure maps the functionality as described in the previous sections: (i) basic models are organized in an MB; (ii) the set of system variants is specified in an SES; (iii) the selection of a particular system variant depends on the current settings of SESVariables, it is performed by the pruning operation and its result is a decision-free tree structure, called PES.

Then, an executable simulation model (EM) can be generated based on the PES and basic models from the MB using an appropriate *translation* method. The composition of EM as tuple (MUS_i, EF_j) means that it consists of a Model Under Study (MUS) and a corresponding Experimental Frame (EF), according to the theory in [15, 19]. The indices *i* and *j* are markers for a certain system configuration. An EM is transmitted to the *Execution Unit* (EU). The EU performs three major tasks: (i) linking an EM with a simulation engine; (ii) executing a simulation run; and (iii) collecting the results.

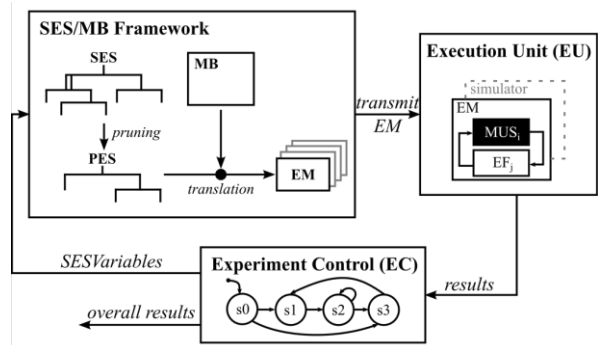


Figure 5. Infrastructure for multi-variant modeling and reactive model generation and execution

Once the execution phase is complete, the results are sent to the *Experiment Control* (EC).

The EC manages the order of EM generations and executions. Based on an application-dependent algorithm, the EC computes new settings for SESVariables as current input for the SES/MB framework to start the next cycle. Additionally, the EC collects all intermediate results from the EU. By means of the feedback loop, structural changes of a variable struc-

ture system or experiments with several system configurations can be executed in a reactive manner. Finally, the EC provides the overall results to the user or another software component.

For the investigation of multiple system configurations, such as in our introduced engineering example, it can be useful to generate an EM suite, as illustrated in Figure 5, and to execute it in a sequential or distributed manner by the EU.

The proposed infrastructure has been implemented in the MATLAB/Simulink environment. Thus, a parallel or distributed execution of EMs by the EU is directly supported. Basic implementation aspects of the infrastructure within MATLAB/Simulink and its usage for solving a specific class of multi-variant problems are discussed in [20]. A core element of the infrastructure is the SES toolbox for MATLAB/Simulink, which has been developed by the Research Group CEA [21, 22]. The toolbox provides a graphical SES editor and several methods, such as: (i) *merging* to synthesize various SESs; (ii) *pruning* for deriving a PES; (iii) *flattening* for the hierarchy reduction of a PES; (iv) *validity checking* of an SES and PES; and (v) *translation* scripts or templates to build EM for Simulink or MATLAB/DEVS [23]. Advanced engineering applications for deploying the SES toolbox for MATLAB/Simulink in the field of model-based testing can be found in [20]. Moreover, a new prototype of the SES toolbox, implemented with Python and supporting an XML interface, is in development to open the way for investigating the approach in conjunction with other simulation environments.

6 Conclusion

Multi-variant modeling and reactive model generation and execution is an important requirement in systems and production engineering. This paper presented a metamodel-based approach using the SES ontology and introduced an appropriate infrastructure to solve this requirement. In addition to the baseline SES definition, the approach uses some new extensions which have been explained step by step based on an engineering example. The introduced concept of SESFunctions advances the declarative knowledge representation through a procedural knowledge specification. Particularly for the modeling of systems with a high degree of variability, the SESFunctions support maintaining a lean SES even for complex problems. In a next step, this assumption has to be proven by applying the approach to more complex examples.

The discussed infrastructure, implemented within MATLAB/Simulink, provides a basis for solving more complex engineering problems. Currently, it is used for developing the reactive and structure variable controls of interacting industrial robots and in the field of objective fidelity evaluation of flight and research simulators. Moreover, a new prototype of the SES toolbox, implemented with Python and supporting an XML interface, is in development to open the way for investigating the approach in conjunction with other simulation environments.

ACKNOWLEDGMENTS

The authors, T. Pawletta and A. Schmidt, gratefully acknowledge the grant from the German Science Foundation DFG (PA 631/2).

References

- [1] Capilla R., Bosch J., Kang K.-C. *Systems and Software Variability Management*. Springer Pub., 2013.
- [2] Polzer A. et al. *Managing complexity and variability of a model-based embedded software product line*. Innovations System Software Engineering. Vol. 8, 35-49, 2012.
- [3] Sargardui G., Etxeberria L., Agirre J.A. *Variability management in testing architectures for embedded control systems*. In: The 4th Int. Conf. on Advances in System Testing and Validation Lifecycle (VALID 2012). IARIA Press, 73-78, 2012.
- [4] Krausz M., Zimmer M., Reuss H.C. *OverNight Testing – The fully automated simulation environment for evaluation of car concepts*. SNE – Simulation Notes Europe, Vol. 24(2), 87-94, 2014.
- [5] Schmidt A., Pawletta T.: *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, 481-490, 2013.
- [6] Schwatinski T., Pawletta T., Pawletta S. *Flexible Task Oriented Robot Controls Using the System Entity Structure and Model Base Approach*. SNE – Simulation Notes Europe, Vol. 22(2), 107-114, 2012.

- [7] Zander J. *Model-based Testing of Real-Time Embedded Systems in the Automotive Domain*. PhD Thesis, TU Berlin, 2008.
- [8] Beuche D. *Modeling and building software product lines with pure::variants*. In: Proc. 12th Int. Software Product Line Conf., IEEE Comp. Soc., 358-358, 2008.
- [9] Barros, F.J. *The dynamic structure discrete event system specification formalism*. Transactions of the SCS International, No.1, 35-46, 1996.
- [10] Pawletta T., Lampe B., Pawletta S., Drewelow, W. *A DEVS-based approach for modeling and simulation of hybrid variable structure systems*. In: Modeling, Analysis, and Design of Hybrid Systems, Lecture Notes in Control and Information Sciences 279, Springer Pub. , 107-129, 2002.
- [11] Haber A. et al. *First-class variability modeling in MATLAB/Simulink*. In: Proc. of the 7th Int. Workshop on Variability Modelling of Software-intensive Systems. ACM Press, 11-18, 2013.
- [12] The MathWorks. *Variant management*. <http://de.mathworks.com/help/simulink/ug/variant-management.html>, Date of reading 12-Nov. 2015.
- [13] Zeigler B.P., Hammonds P.E. *Modeling and Simulation-Based Data Engineering*. Academic Press, 2007.
- [14] Zeigler B.P., Sarjoughian H.S., *Guide to Modeling and Simulation of Systems of Systems*. Springer Pub., 2012.
- [15] Zeigler B.P., Prähofer H., Kim T.G., *Theory of Modeling and Simulation*. Academic Press, 2000.
- [16] Pawletta T., Schmidt A., Zeigler B.P, Durak U. *Extended Variability Modeling Using System Entity Structure Ontology within MATLAB/ Simulink*. In: Proc. SprigSim-ANSS 2016, Pasadena/CA, USA, 62-69, 2016.
- [17] Rozenblit, J.W., Zeigler, B.P. *Representing and constructing system specifications using the system entity structure concepts*. In: Proc. of the 1993 Winter Simulation Conf., 604-611, 1993.
- [18] Pidcock W., Uschold M. *What are the differences between a vocabulary, a taxonomy, a thesaurus, an ontology, and a meta-model?* In: The Web Graph Database, <http://infogrid.org/trac/wiki/Reference/PidcockArticle>, Date of reading 7-July 2016.
- [19] Traoré M., Muzy A. *Capturing the dual relationship between simulation models and their context*. Simulation Modelling Practice and Theory, No. 2, 126-142, 2006.
- [20] Schmidt A., Durak U., Pawletta T. *Model-based testing methodology using system entity structures for MATLAB/Simulink models*. Simulation: Transactions of SCS Int., Sage Pub., DOI: 10.1177/0037549716656791, 18 pp., 2016.
- [21] Pawletta T., Pascheka D., Schmidt A., Pawletta S. *Ontology-assisted system modeling and simulation within MATLAB/Simulink*. SNE – Simulation Notes Europe, Vol. 24(2), 59-68, 2014.
- [22] Pawletta T., Schmidt A., Pascheka D. *SES Toolbox for MATLAB/Simulink*. https://www.mb.hs-wismar.de/cea/SES_Tbx/, Date of reading 12-Nov. 2015.
- [23] Deatcu C., Schwatinski T., Pawletta T. *DEVS Toolbox for MATLAB*. https://www.mb.hs-wismar.de/cea/SES_Tbx/, Date of reading 12-Nov. 2015.

A Framework for Parametric Aircraft, Engine and Sub-Systems Models

Michael Sielemann¹, Olga Silantjeva², Majed Sammak²

¹Modelon Deutschland GmbH, Germany

²Modelon AB, Sweden

michael.sielemann@modelon.com

Multi-disciplinary design optimization of aircraft sub-systems and functions typically requires connecting a wide array of tools and technologies. While this approach is feasible, it is fragile (many tools, many versions) and slow. The objective of this paper is to propose a unified framework for model-based design of aircraft sub-systems while properly representing their interactions with the airframe and the propulsion system. The implemented framework is based on the modeling language Modelica. It was found that the framework is feasible and produces meaningful results from a unified model in two engineering applications, aircraft level sizing and synthesis, and six degrees of freedom dynamic simulation.

1 Introduction

Aeronautical systems such as aircraft and their sub-systems are becoming more and more integrated. The integration takes place along a number of trends. We mention two of these. The first one is the electrification of secondary power on-board aircraft. This trend is also called the “More Electric Aircraft” and has shaped the industry road maps since more than two decades. Secondary power is provided by the jet engines, and is used to drive the consumer sub-systems such as flight control, air conditioning, or cabin lighting. Historically, three different types of secondary power were equal, namely, electric power, hydraulic power, and pneumatic power. With the “More Electric Aircraft” this is changing in favor of electric power. The main reason lies in the anticipated development potential of power electronics, which is all but exhausted (like that of pneumatic and hydraulic power). The second trend is more recent and is the electrification of primary power. Primary power is also generated by the jet engines on today’s aircraft and propels the aircraft. Thrust is a synonym for primary power. Aircraft with partially or fully electrified primary power systems are called hybrid or fully electric aircraft.

These large trends emerge as the benefit of incremental and local improvements to aircraft and their sub-systems are exhausted. Global optimization of the entire aircraft package is the only remaining way to achieve substantial improvements. Conventional engineering methods work poorly however for such unconventional solutions, as they rely on statistical data of legacy products and established problem breakdowns. Design of unconventional aircraft and sub-system solutions has to become more deterministic and efficient to realize the potential of the above mentioned concepts.

A number of contributions were made towards the model-based design of aircraft and their sub-systems. On the level of overall aircraft and sub-system architecture Kaslusky et al. [8] were first to define processes to define architecture alternatives and to interface different discipline codes with each other. However, their study was limited to loosely coupled disciplinary modules. Liscouët-Hanke et al. [15] describe a simulation framework, in which different codes are interfaced to each other, and aircraft and sub-system architectures are analyzed in a semi-automatic fashion. Such heterogeneous frameworks are feasible but result in inefficiencies due to the interfacing effort. Bals and co-authors [2, 21] in turn present an integrated or unified simulation framework, in which all compo-

nents of the model are implemented consistently in the modeling language Modelica [18]. The scope of the effort does not include sizing optimization however. De Tenorio et al. [5] present an approach in which simulation and sizing optimization are coupled with decomposition of functional requirements. Lammering [13] interfaced sub-system analysis modules with an aircraft sizing and synthesis code as well as a gas turbine performance code.

Additionally, numerous authors contributed to the state of the art on the level of sub-system sizing optimization. We mention work on electric power generation and distribution systems by Schallert [20] and by Seresinhe [22]. Work on the environmental control system has been published by Sielemann et al. [25]. Additionally, Lüdders et al. [17] conduct model-based design of a multi-functional fuel cell system.

Other relevant aspects such as jet engine performance [28, 7, 10] and flight dynamics [26, 29, 16] were studied in details by the respective scientific communities.

However, up to now nobody has proposed a plausible unified framework for modeling and simulation of aircraft and their sub-systems for a broad range of applications ranging from aircraft and sub-system conceptual design to detailed design of the components they consist of. For time and resource efficient implementation of model-based design processes reuse of model assets is pivotal. Additionally, this improves consistency of results.

The objectives of this paper are

1. To suggest a unified framework for modeling and simulation of aircraft and its sub-systems for a broad range of applications ranging from aircraft and sub-system conceptual design to detailed design of the components they consist of. We restrict the scope however to only cover system simulation, i.e., all processes governed by ordinary differential equations (ODE) or differential-algebraic equations (DAE). Processes governed by partial differential equations are beyond the scope of the framework, unless their partial derivatives have been suitably discretized to match the formal framework of an ODE or DAE (e.g., one-dimensional discretization of the balance equations of thermo-fluid dy-

namics).

2. To substantiate why the unified framework can plausibly be applied to industrial-scale problems involving “complex” models and “sophisticated” analyses
3. To apply the framework to two engineering problems, which previously had to be solved in different engineering environments, namely, the sizing and synthesis of a fixed-wing aircraft, and the six degrees of freedom dynamic simulation of its flight dynamics.

2 Framework and implementation

First, we select the Modelica object-oriented and acausal modeling language to implement our framework. This is one of the main factors through which we aim to fulfill objective 2. We expect that models of the sub-systems will be the largest part of the models developed for the applications discussed herein. For this reason, and the high proliferation of this modeling language in the respective industry, we believe that it is a natural choice to integrate the three aspects of aircraft, engine, and sub-systems in this modeling language. The literature substantiates how industrial scale problems can be solved when implemented in the Modelica modeling language, such as optimal control [1], nonlinear-model predictive control [14] and many more. We also believe in the advantage of leveraging off-the-shelf software for the framework. This is rich in the eco system around the open Modelica language with many open source and commercial compiler, library, and tool options.

One of the critical ways to exploit framework flexibility is through mixing handbook and physics-based models. We define handbook methods or models through their reliance on statistical correlation of key design aspects based on databases of conventional designs. An important example of handbook methods is the classic literature on aircraft design [27, 19, 24, 9]. Physics-based methods in turn rely on first principles such as conservation equations of mass, energy, and momentum. Obviously, the latter are suitable for studying unconventional designs, while the former are

not. Like this, the framework allows mixing physics-based models for unconventional system aspects with handbook methods for everything around them.

2.1 Aircraft models

2.1.1 Sizing models

Aircraft sizing and synthesis is concerned with estimating key performance aspects such as aerodynamic characteristics and masses. As such, this discipline relies on involved field simulation such as Computational Fluid Dynamics of aerodynamics or structural analysis via the Finite Element Method. These are beyond the scope of this framework. Instead, we rely on handbook methods for sizing on aircraft level.

The aircraft is broken down in a hierarchical fashion. It consists of the airframe, the primary and secondary power system, and the consumer and cabin systems. For a conventional aircraft, the break-down is

- Airframe
 - Wing
 - Fuselage
 - Tails
 - Landing gears
 - Primary and secondary flight control
- Primary and secondary power system
 - Turbo fan engines including generation of secondary power
 - Auxiliary Power Unit including generation of secondary power
 - Emergency power
 - Fuel system
 - Electric power distribution
 - Hydraulic power distribution
 - Pneumatic power distribution
- Consumer and cabin systems
 - Environmental control and cooling systems
 - Cabin and furnishings
 - Wing and engine ice protection
 - Avionics

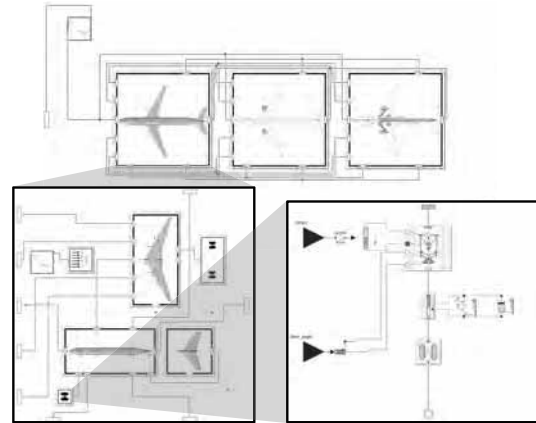


Figure 1: Top-level model breakdown shown on the top, airframe break-down on the lower left, front landing gear break-down on the lower right

The breakdown is illustrated graphically in figure 1. This figure shows the actual view presented to the user in the graphical user interface of a Modelica IDE.

Each type of component in the break-down above is represented through a class hierarchy of interfaces, templates, and implementations. Some of these classes have attributes describing the geometry of the component they represent. Examples are components like wings, tails, and classic transport aircraft fuselages. These can be represented in a first order approximation by a number of scalars following usual aircraft sizing practice [27, 19]. Other components of the break-down are not represented with their geometry, both because this is not customary in aircraft sizing, and because they cannot be represented as a single component (an electric power distribution system is a heterogeneous sub-system of the aircraft).

Figure 2 shows the parameter dialog of a wing component in a Modelica Integrated Development Environment (IDE). The first order representation of the component is given through 42 scalars in the framework. The set of variables consists of typical non-dimensional variables such as taper or aspect ratio, dimensional variables like reference area or span as well as coordinates of important reference points of the component. The component implementation contains equations relating these scalars to each other. The user can pick a sub-set of these scalar that shall be used to define the geometry (note the check boxes with the text “Prescribe?” on the right of each scalar



Figure 2: Parameter dialog of a wing component

in figure 2). Following the established principles of acausal modeling languages [18] a Modelica compiler will symbolically reorder and solve the equations relating the geometry parameters to each other and compute the unknown variables from the given ones. This is successful if the equation system is non-singular for the variables selected as inputs by the user.

All dimensions and coordinates in the sizing context are expressed in the so-called “weight-and-balance frame”¹. Figure 3 illustrates this coordinate system (in black). It is right-handed with the origin near the fuselage nose tip, its x-axis points along the center line of the fuselage, its y-axis points to the starboard wing tip, and its z-axis points up. The geometry view of the sized aircraft is updated automatically to match the given sizing parameters of the components.

The basic interfaces defined here carry information about position, orientation, mass, and rotational inertia. This is sufficient for the model use discussed in this section 2.1. Further physical interfaces are added together with the implementation of physics-based models later in section 2.3.

Based on the interfaces described above we implemented handbook methods for all parts of aircraft. These models are largely implement following Shev-

¹This coordinate system is also called an aerodynamic coordinate system in the aerodynamics literature, see, e.g., [3]. However, to avoid confusion with the wind coordinate system introduced later in section 2.1.2, we avoid that name and use the one mentioned here.



Figure 3: Aircraft coordinate systems: Weight and balance system in black, body-fixed system in green, stability system in red, and wind system in blue

ell [24] and Kroo [9]. These are the baseline models in our framework and can be mixed and matched with physics-based ones. The models estimate the mass and center of gravity of each component the aircraft. As illustrative example we give the wing mass correlation due to Shevell [24]

$$W_{wing} = 4.22S_{wg} + 1.64210^{-6} \cdot \frac{N_{ult}b^3\sqrt{MTOW} \cdot MZFW(1+2\lambda)}{(t/c)_{avg} \cos^2\Lambda_{ea}S_{wg}(1+\lambda)} \quad (1)$$

Here, W_{wing} , $MTOW$, $MZFW$ are the wing, maximum take-off, and zero fuel weights in pounds mass. S_{wg} is the gross wing area in square feet, N_{ult} is the ultimate load factor, b is the span, λ is the taper, $(t/c)_{avg}$ is the average thickness over chord, and Λ_{ea} is the sweep of the structural axis.

An important capability in the sizing models is being able to capture so-called snowball effects due to subsystem design alternatives. These are often neglected in aircraft design. In a nutshell, higher (or lower) subsystem weight and secondary power off-takes lead to higher (or lower) mission block fuel. This amount of fuel has obviously to be carried by the aircraft, and feeds back into the component masses. The present framework captures such snowball effects. On one

hand, equation (1) exemplifies how total masses influence component structural design and mass. This is typical for several component mass correlations. Additionally, there are more complex cases, for instance the fuselage mass correlation per Shevell [24]. Here, the fuselage mass is driven by a bending moment load index and a pressurization load index. If heavier equipment is installed on the wing (such as the propulsion system or accessories) then relief on the fuselage bending moment load index is provided. Therefore, higher sub-system mass can also reduce a component mass (if the relevant load index is the dominating one).

Additionally, section 2.2.2 later introduces the means in this framework to relate secondary power off-take to the mission block fuel.

2.1.2 Dynamic models

In addition to the “weight and balance” coordinate system introduced in section 2.1.1 we now introduce three additional coordinate systems. The origin of all three of them is at the center of gravity. We introduce the body-fixed coordinate system, the stability coordinate system, and the wind coordinate system following Stevens and Lewis [26]. All three are right-handed coordinate systems and were already illustrated in figure 3. On the body-fixed coordinate system the x-axis points forward in longitudinal direction, and the z-axis points downward. In the stability axes the x-axis is rotated around the y-axis downward according to the angle of attack, α . The wind axes finally are rotated around the body-fixed z-axis by the slide slip angle β .

The masses and inertia of the aircraft are either estimated through the sizing part of the aircraft model by using a corresponding airframe, power system, and consumer system models as described in section 2.1.1. Alternatively, they can be entered manually on a “lumped model” (i.e., lumped parameter mass and rotational inertia scalars or matrices).

In order to dynamically model flight and ground maneuvers, models of forces and moments have to be added. First and foremost, these are the aerodynamic forces. These are implemented in a container class that aggregates three force sub-models (lift force, drag force, side force) and three moment sub-models (rolling moment, pitching moment, yawing moment).

Typically, the aerodynamics forces are implemented globally and act on the wind or stability coordinate systems attached to the center of gravity. Moments in turn are often expressed in body axes. The framework provides a number of convenient choices such as the possibility to specify force vector rotation separately from moment vector rotation, or together with the moments. Thrust is always applied to the aircraft model locally at the engine installation points. This is required for meaningful simulation of asymmetrical thrust conditions such as engine failure cases.

While several classic aerodynamics models build on a single set of aerodynamic forces and moments applied globally to the center of gravity of the aircraft, the present framework is flexible in that it can also accommodate arbitrary local forces. These can be used for truly local forces such as the external aircraft drag due to ram air inlets, or to mimic distributed forces of a higher fidelity aerodynamic model where the distributed forces and moments replace the global aerodynamic force model.

The main application of the dynamic models is the simulation of the aircraft with six degrees of freedom. These models can be initialized arbitrarily. Typically, position, speed and orientation angles are prescribed. Alternatively, other quantities such as rotational velocities can be provided by the user if this is meaningful for a given application.

Alternatively to simulating the free motion of the aircraft through the atmosphere, test rig models can be used to prescribe motion and compute forces and moments due to aerodynamics, thrust, and inertia over time. This is often useful for debugging.

In this effort we have implemented two sets of force and moment models. The first one is based on the handbook-type method of Shevell [24, 9]. For lift, the method first estimates the outer wing panel maximum lift coefficient $C_{l,max}$ and wing maximum lift coefficient, $C_{L,max}$. Based on this, the effects of different kinds of slats and flaps are approximated, and a Mach Number correction is applied. The drag in turn contains components due to parasite drag, induced or lift-dependent drag, and wave drag. Details are described in the given references.

The second set of force and moment models is based on the Research Civil Aircraft Model (RCAM) [12]. The RCAM forces are presented in wind frame and

moments are implemented in body-fixed coordinate system. Once they are implemented the forces and moments in the aircraft model can easily be redeclared to such higher fidelity models.

2.2 Engine models

2.2.1 Handbook methods

Handbook methods for jet engines typically rely on scaling laws to “rubberize” the engine characteristics. We again implemented the scaling laws of Shevell [24]. These depend on the sea level static thrust as main scaling factor, as well as different empirical factors and exponents based on engine technology (high-bypass, prop-fan etc.). Based on these scaling laws, geometry, maximum available thrust, and thrust specific fuel consumption are estimated. However, no statements are made about the detailed thermodynamic state in the engine locations at various operating conditions, or the cost of secondary power off-takes in terms of shaft power or bleed extraction.

2.2.2 Physics-based models

Our physics-based models of jet engine propulsion largely rely on Walsh and Fletcher [28]. Particular emphasis was put on the important topic [28, 11] of fully rigorous thermodynamic properties. Here, the fact that the isentropic exponent is not constant along the theoretical isentropic deceleration required to relate total and static quantities to each other is respected carefully. For this, the entropy function [28] is leveraged as suggested by Sethi [23].

Our compressor model employs the map formats for compressors and turbines suggested by Walsh and Fletcher [28]. Additionally, we apply the scaling procedure for based described by Jonas [6]. The component models support the modeling of secondary air systems. For instance, bleed can be extracted from a compressor model, routed through an arbitrary network, and be supplied for turbine film cooling or customer purposes. Our models support both on-design and off-design simulation.

Based on these component models, different cycles can be built up using the interface and template model structure introduced in section 2.1 on the level of the

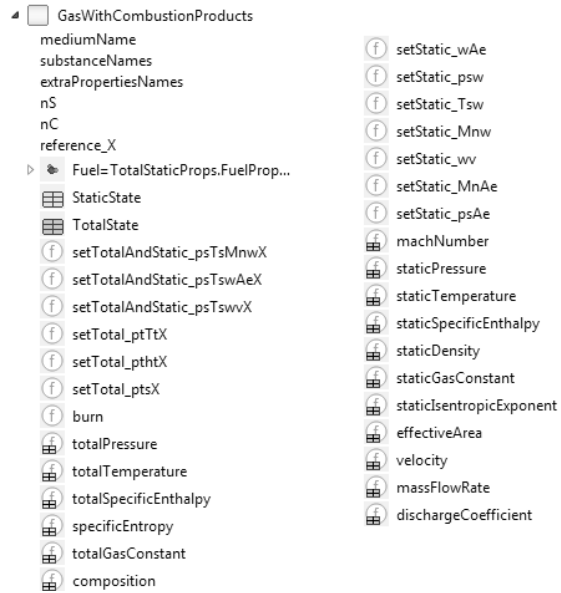


Figure 4: Thermodynamic property functions: Static and total state records plus functions acting on total state records to the left and function acting on static state records to the right

jet engine, be it a turbo fan, turbo jet, geared turbo fan or anything else. An example breakdown of a turbo fan engine is illustrated graphically in figure 5. This figure shows the actual view presented to the user in the graphical user interface of a Modelica IDE.

2.3 Sub-system models

2.3.1 Handbook methods

Classic handbook methods for sub-systems such as the ones implemented herein are very simple. The mass of a sub-system is usually a relatively simple function of a few parameters such as number of seats or passengers, wing reference area and so on. The presentation given in section 2.1.1 is representative.

2.3.2 Physics-based models

When implementing physics-based models of the sub-systems the sub-system interfaces are extended with the respective additional physical interfaces. The object-oriented model templates are refined with re-

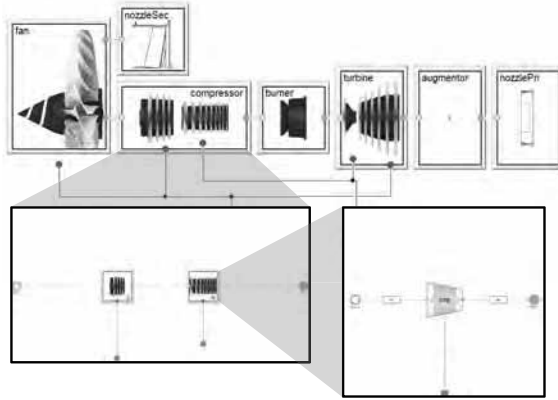


Figure 5: Top-level turbofan model breakdown shown on the top, compressor break-down on the lower left, high pressure compression section break-down on the lower right

defined constraining classes. These models can typically be implemented with an off-the-shelf Modelica library. The details of such models are beyond the scope of this paper and already discussed elsewhere (see the references given in section 1 for instance).

3 Results

3.1 Aircraft sizing

Following the objective to apply the framework to the engineering problem of sizing and synthesis of a fixed-wing aircraft, we select the Douglas DC-9-30 as reference design. Table 1 contains data on some of the sizing and mission parameters according to Kroo [9].

In order to give an example of the scaling laws resulting from the framework implementation we first freeze the geometry of the aircraft to the data given in table 1, and scale it up in terms of take-off gross weight, and thus design fuel weight. Figure 6 shows the results of this exercise. On the upper right we see the cruise range of the aircraft over the maximum take-off gross weight. Note that this is the maximum cruise range; a normal payload-range diagram [27, 19, 24, 9] can be constructed for each point along the line. Each point then defines a candidate rightmost point in the payload-range diagram. Note how the plot starts at zero cruise range for a take-off

Variable	Value
Pax in single class	115
Seat layout	2 seats, aisle, 3 seats
Seat pitch	32 in
Wing reference area	1000.7 ft ²
Wing aspect ratio	8.701
Wing sweep	24.5°
Flap span to wing span	0.627
Flap chord to wing chord	0.36
Horiz. tail to wing ref. area	0.26
Vert. tail to wing ref. area	0.161
Static sea level thrust per engine	14500 lbf
Cruise altitude	31000 ft
Cruise Mach Number	0.78
Maximum altitude	43000 ft
Cruise range	1596.4 nm

Table 1: Exemplary DC-9-30 sizing and design mission parameters

gross weight of approximately 85,000 lbm. The circle indicates the sizing of the baseline aircraft per table 1.

Further details are provided in the upper left of figure 6. Here, we see different weights as function of the above mentioned variation on the maximum take-off gross weight. We have the take-off weight for the design mission (right-most point on the payload-range diagram), the initial cruise weight (after burning fuel for the climb segment), final cruise weight (after burning fuel for the climb and cruise segments), as well as zero fuel weight. The difference between the latter and the landing weight are the fuel reserves. The zero fuel and operating empty weights are particularly interesting. Their value only increases slightly as we double the maximum take-off gross weight. The equality between initial and final cruise weights to the far left matches the results seen in the upper right of the same figure. As all available fuel is burnt for climbing and landing no fuel for the cruise segment remains. Thus, the nominal cruise range is 0 nm (the climb and landing distance is often neglected in classic approaches).

While the plots seemingly indicate that we can scale up the baseline aircraft to double take-off gross weight and more than double cruise range, this is obviously not the case. Freezing the aircraft geometry is not appropriate; common sense tells us that the wing size for instance has to be increased to actually enable this.

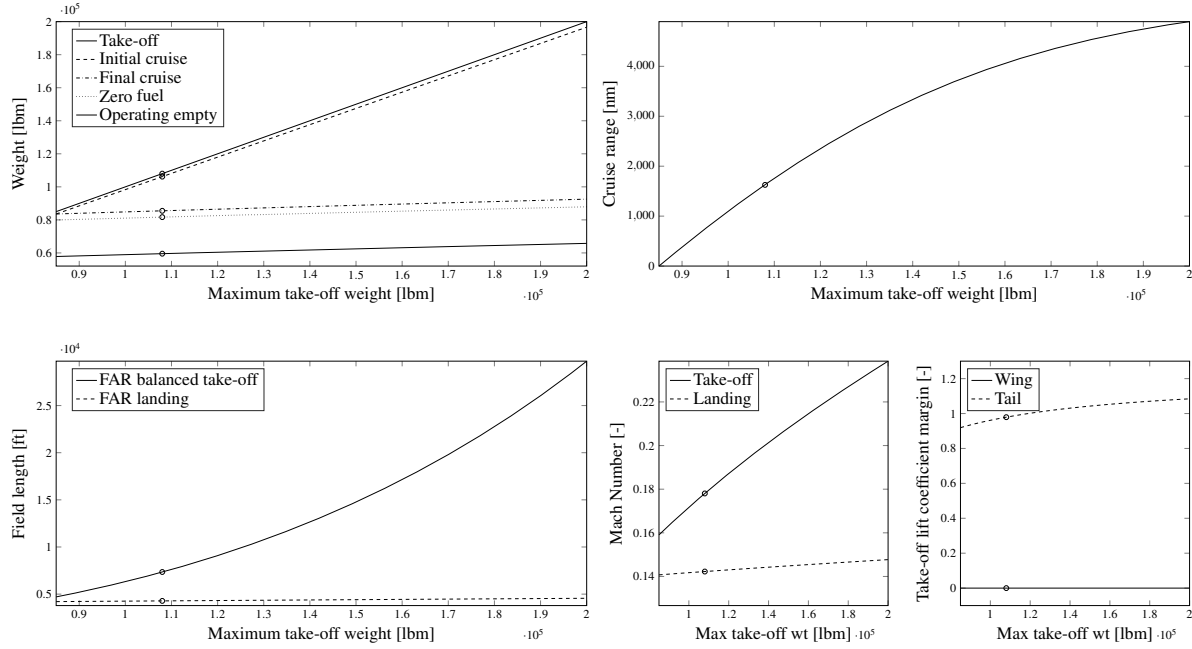


Figure 6: Variation of various quantities with maximum take-off gross weight: Weights, cruise range, field length, Mach Number, lift coefficient margin (from upper left to lower right)

This can be read off a number of a number of quantities. Figure 6 contains some of these in the left and middle plots in the lower row. In the first one, we see the FAR balanced take-off field length and the landing field length. While the latter remains acceptable, the take-off field length largely grows beyond reasonable values as we scale up the aircraft. In a real design case an inequality constraint would be used to modify the aircraft geometry appropriately. In this simple example none of this happens, and we see on the middle plot in the lower row that also the take-off Mach Number grows to high values. Finally, the lower right plot in figure 6 shows the resulting lift coefficient margin on take-off. As we base the take-off Mach Number in the middle plot on take-off wing zero lift margin [24], the corresponding value is constantly zero. The plot also shows the resulting tail lift margin to trim the aircraft. This is non-zero across the maximum take-off gross weight variation.

3.2 Impact of sub-system sizing

In this example we touch upon the analysis of the impact of sub-system sizing on aircraft level metrics. We

select the air conditioning or environmental control and anti-ice sub-system of the above mentioned DC-9-30 aircraft. This is described in more detail by Cleaves et al. [4]. Kroo [9] reports a total weight of 1584 lbm for it. Shevell's correlation [24] predicts a weight of 1725 lbm, Torenbeek [27] predicts 1311 lbm. We again freeze the geometric design and scale it up and down, this time to maintain the cruise range given in table 1 while varying the sub-system weight. We scale the nominal air conditioning and anti-ice weight of 1725 lbm by factors ranging from 0 to 400 %.

3.3 Aircraft dynamic simulation

In our last example we define and size an aircraft representing the Airbus A300-600R. We then compare the six degrees of freedom dynamic simulation trajectories generated with the forces and moments following Shevell's methodology [24] on one hand and the previously mentioned well-known RCAM forces and moments [12] on the other. The reference aircraft used in the latter reference is similar to the given model, and we thus expect this comparison to yield somewhat meaningful results. Looking at the details

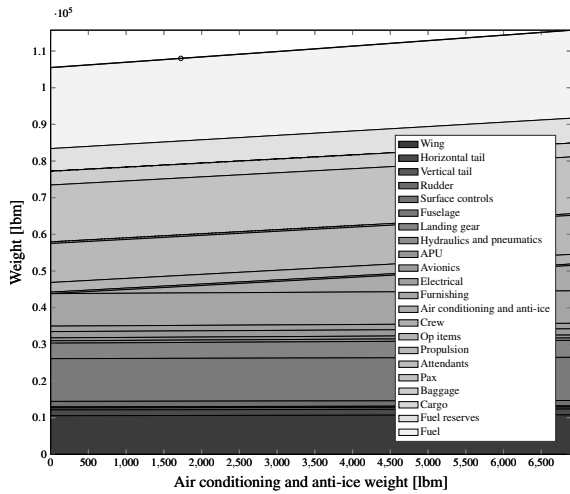


Figure 7: Variation of air conditioning and anti-ice sub-system weight. Dashed line indicates operating empty weight, dash-dotted zero fuel weight, and solid line maximum take-off gross weight

of the two sets of models, a number of differences become apparent, such as the difference in drag coefficient (RCAM assumes a deployed high lift configuration and correspondingly higher drag coefficient, which had to be mimicked by a manual drag increment in the sizing methodology, where the high lift deployment only affects the maximum lift coefficient and the pitching moment). Both trajectories shown in figure 8 we generated based on the same control law (and coefficients). At time $t = 0$ the commanded vertical speed is changed to a non-zero value as illustrated in the top sub-plot. The same sub-plot also shows the actual values. Due to differences in pitching moment the elevator deflection in the middle sub-plot has a different offset. The controller response is shown here. The bottom sub-plot finally shows the resulting altitude in both scenarios.

4 Conclusions

A modeling and simulation framework for aircraft, engine and sub-system models and a broad range of applications was described. As model assets are expensive, particular emphasis was put on enabling reuse through a general interface and template structure. The results show how early conceptual design aircraft

level studies can be flown virtually through interactive dynamic simulation.

Parametric models based on handbook methods were implemented in the framework due to their convenience. They enable an agile way of working, and result in reduced setup time as estimation logic and scaling laws are imbedded in models. Their fidelity is often sufficient to capture first order effects. The framework provides flexibility however to enable mixing and matching physics-based models for unconventional system aspects with these handbook methods for everything else.

The framework and models were applied to small exemplary analyses capturing scaling laws on aircraft level, and simulating the six degrees of freedom aircraft dynamics.

5 Acknowledgments

The authors acknowledge the enlightening discussions with Johan Andreasson, Hubertus Tummescheit, and Mathias Strandberg. Without their support and contributions this framework would not have become reality.

References

- [1] J. Åkesson, K.-E. Årzén, M. Gäfvert, T. Bergdahl, and H. Tummescheit. Modeling and optimization with Optimica and JModelica.org—languages and tools for solving large-scale dynamic optimization problems. *Computers & Chemical Engineering*, 34(11):1737–1749, 2010.
- [2] J. Bals, G. Hofer, A. Pfeiffer, and C. Schallert. Object-oriented inverse modelling of multi-domain aircraft equipment systems with modelica. In P. Fritzson, editor, *Proceedings of the Third International Modelica Conference*, Linköping, Sweden, 2003.
- [3] R. T. Biedron, J.-R. Carlson, J. M. Derlaga, P. A. Gnoffo, D. P. Hammond, W. T. Jones, B. Kleb, E. M. Lee-Rausch, E. J. Nielsen, M. A. Park,

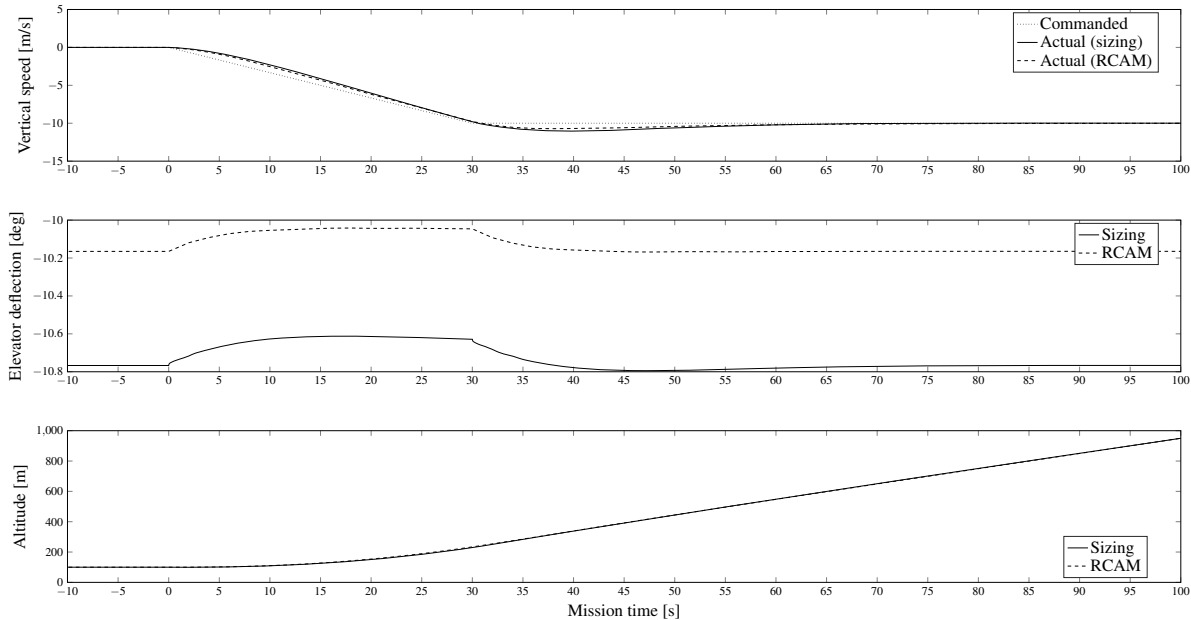


Figure 8: Dynamic simulation of a change in commanded vertical speed at $t = 0$

- et al. FUN3D manual: 12.8. Technical Report TM-2015-218807, NASA, October 2015.
- [4] V. F. Cleaves, H. Hauger, and J. Perlee. DC-9 environmental control design and first year's service experiences. *Journal of Aircraft*, 5(1):64–72, 1968.
 - [5] D. M. Cyril de Tenorio, Michael Armstrong. Architecture subsystem sizing and coordinated optimization methods. In *47th AIAA Aerospace Sciences Meeting and Exhibit*, Orlando, Florida, January 2010.
 - [6] S. M. Jones. An introduction to thermodynamic performance analysis of aircraft gas turbine engine cycles using the numerical propulsion system simulation code. Technical Report TM—2007-214690, NASA, March 2007.
 - [7] S. M. Jones. Steady-state modeling of gas turbine engines using the numerical propulsion system simulation code. In *ASME Turbo Expo 2010: Power for Land, Sea, and Air*, pages 89–116. American Society of Mechanical Engineers, June 2010.
 - [8] S. Kaslusky, D. Sabatino, and L. Zeidner. ITAPS: A process and toolset to support aircraft level system integration studies. In *45th AIAA Aerospace Science Meeting and Exhibit, Proceedings of the*, Reno, NV, USA, January 2007.
 - [9] I. Kroo and R. Shevell. *Aircraft design: Synthesis and analysis*. Desktop Aeronautics Inc., 2001.
 - [10] J. Kurzke. Advanced user-friendly gas turbine performance calculations on a personal computer. In *ASME 1995 International Gas Turbine and Aeroengine Congress and Exposition*. American Society of Mechanical Engineers, June 1995.
 - [11] J. Kurzke. About simplifications in gas turbine performance calculations. In *Proceedings of the ASME Turbo Expo*, volume 3, pages 14–17, May 2007.
 - [12] P. F. Lambrechts, S. Bennani, G. Looye, and A. Helmersson. Robust flight control design challenge problem formulation and manual: the research civil aircraft model (RCAM). Technical Report TP-088-3, Group for Aeronautical Research and Technology in Europe GARTEUR, 1997.

- [13] T. Lammering. *Integration of Aircraft Systems into Conceptual Design Synthesis*. PhD thesis, RWTH Aachen, 2014.
- [14] P.-O. Larsson, F. Casella, F. Magnusson, J. Andersson, M. Diehl, and J. Åkesson. A framework for nonlinear model-predictive control using object-oriented modeling with a case study in power plant start-up. In *Computer Aided Control System Design (CACSD), 2013 IEEE Conference on*, pages 346–351. IEEE, 2013.
- [15] S. Liscouët-Hanke, S. Pufe, and J. Maré. A simulation framework for aircraft power management. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 222(6):749–756, 2008.
- [16] G. Looye. The new DLR flight dynamics library. In *Proceedings of the 6th International Modelica Conference*, pages 193–202, March 2008.
- [17] H. P. Lüdders, H. Strummel, and F. Thielecke. Model-based development of multi-functional fuel cell systems for more-electric-aircraft. *CEAS Aeronautical Journal*, 4(2):151–174, June 2013.
- [18] S. Mattsson, H. Elmqvist, and M. Otter. Physical system modeling with Modelica. *Control Engineering Practice*, 6(4):501–510, 1998.
- [19] J. Roskam. *Airplane Design: Component Weight Estimation*. Roskam Aviation and Engineering Corp., 1985.
- [20] C. Schallert. A novel tool for the conceptual design of aircraft on-board power systems. In *AeroTech Congress and Exhibition, Proceedings of the*, Los Angeles, CA, USA, September 2007.
- [21] C. Schallert, A. Pfeiffer, and J. Bals. Generator power optimisation for a more-electric aircraft by use of a virtual iron bird. In *25th International Congress of the Aeronautical sciences (ICAS), Proceedings of the*, Hamburg, Germany, September 2006.
- [22] R. Seresinhe and C. Lawson. Electrical load-sizing methodology to aid conceptual and preliminary design of large commercial aircraft. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 229(3):445–466, March 2015.
- [23] V. Sethi. *Advanced performance simulation of gas turbine components and fluid thermodynamic properties*. PhD thesis, Cranfield University, April 2008.
- [24] R. S. Shevell. *Fundamentals of flight*, volume 2. Prentice Hall Englewood Cliffs, New Jersey, 1989.
- [25] M. Sielemann, T. Giese, B. Oehler, and M. Gräber. Optimization of an unconventional environmental control system architecture. *SAE International Journal of Aerospace*, 4(2):1263–1275, 2011.
- [26] B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation*. John Wiley & Sons, 2004.
- [27] E. Torenbeek. *Synthesis of subsonic airplane design*. Delft University Press, 1976.
- [28] P. P. Walsh and P. Fletcher. *Gas turbine performance*. John Wiley & Sons, 2004.
- [29] P. Zipfel. A C++ architecture for unmanned aerial vehicle simulations. In *AIAA Infotech*, May 2007.

Behaviour of Rotor Blade with Double-Sweep in a Whirl Tower

M. Rohin Kumar, Sriram Palika

Rotavio Labs Pvt Ltd, India

kumar.rohin@gmail.com

Modelling and Simulation in Aviation

1 Introduction

Advanced blade shapes have been used in helicopter rotors for performance enhancement and noise alleviation. One such technology is the Blue Edge™ concept [1, 2] developed by Airbus Helicopters in association with ONERA. The Blue Edge™ blade, featuring double-sweep and tip-anhedral, is a passive BVI-noise reduction concept based on the ERATO project [3, 4] which was carried out within the framework of cooperation between ONERA, DLR and Airbus Helicopters. Apart from this fact, not much information has been published in open literature with regards to the precise geometric details of the blade. While it has been claimed that the advanced shape of the blade reduces helicopter noise by 3-4 dB, its effects on structural dynamics and vibratory loads are largely unknown.

The prediction of loads and response of the helicopter rotor requires a comprehensive analysis program. Since helicopter rotor loads and response evaluation is an aeroelastic problem, it requires structural, inertia and aerodynamic models. Based on the models described in [5] and [6, 7], two programs – RLDyn and RL-CAR were developed at Rotavio Labs for rotorcraft comprehensive analysis. Validation of the two programs has been presented in the two references.

RLDyn is a finite element program developed for structural dynamic analysis of generic rotor blades. The rotor blade is modelled as rotating, long, slender beam-type finite elements. Equations of motion of the blade were derived using Hamilton's principle. The strain displacement relationships were considered as moderate, with couplings in flap, lag, torsional and axial deformations. The blade is assumed to be made of isotropic material but can have non-uniform properties along the span. This program can study the effects of various advanced geometric shapes, concentrated masses along span-wise / chord-wise directions and the effects of linear lead-lag

damper on natural frequencies and mode shapes of the blade.

RL-CAR is a comprehensive analysis program which uses modal vectors given by RLDyn to transform the rotor blade aeroelastic loads problem from the finite-element domain to the modal domain. Formulations for rotor blade includes geometry features like pre-cone, pre-sweep, pre-droop, pre-twist, torque offset, hinge offset, swept and anhedral tips. The rotor aerodynamics module has several hierarchical models, of which the Generalized Dynamic Wake theory (Peters-He) [8] for inflow and a modified ONERA stall theory [9] for airloads calculations are the most prominent ones.

The objective of this paper is to study the effects of giving a double-sweep along with a tip-anhedral to an untwisted rotor blade of known geometric properties. The effects of the shape-modification on structural dynamics and oscillatory loads of the rotor in a whirl tower analysis are presented. It is to be noted that for the purpose of this paper, the term double-sweep includes the tip-anhedral as well.

2 Analysis

For the current analysis, the behavior of a baseline straight blade was compared with those of blades given the double-sweep shape along with a tip-anhedral. A straight untwisted blade whose properties are given in [5] is taken as the baseline blade (Fig 1(a)). Two blades with the same properties were given the double-sweep shape by sweeping forward from 50%-80% span of the blade and sweeping backward at a slightly larger angle from 80% span to tip. In addition, an anhedral angle was given from 90% span to the tip. One blade was given sweep forward of 10 deg, sweep backward of 15 deg and anhedral angle of 5 deg (Fig 1(b)). The other blade was given sweep forward of 15 deg, sweep backward of 20 deg and anhedral angle of 7.5 deg (Fig 1(c)). The blade mass distribution was taken as 8.45 kg/m and non-

dimensional chord (c/R) was taken as 0.0757. The blades are clamped at the root. The lengths of the elastic axes of the three blades are kept the same. It must be noted that this leads to a reduction of the rotor radius in the case of the blades with double-sweep.

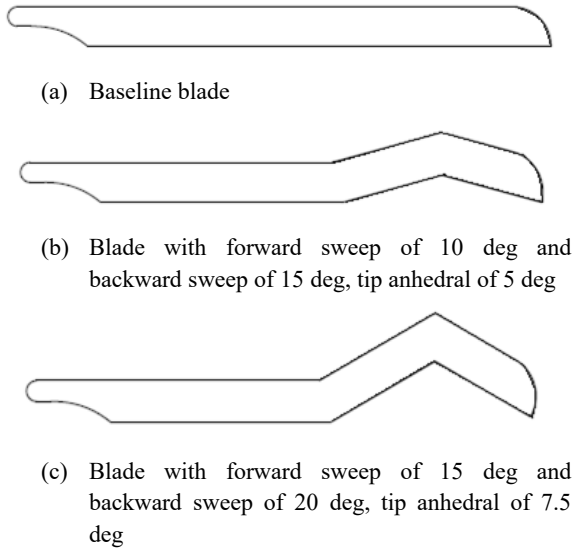


Figure 1. Representative figures of the investigated rotor blades

For obtaining the structural dynamics characteristics, the linear, undamped model is solved in the finite element domain to obtain the natural frequencies and the corresponding mode shapes of the rotating blade. The governing equation for the i^{th} finite element is

$$[M_i]\{\ddot{q}_i\} + [K_i]\{q_i\} = 0 \quad (7)$$

where $[M_i]$ represents the mass matrix of the i^{th} element and $[K_i]$ the stiffness matrix.

For cases with sweep/anhedral, the governing equation for the corresponding element is given by

$$[A]^T[M_t][A]\{\ddot{q}_t\} + [A]^T[K_t][A]\{q_t\} = 0 \quad (8)$$

where $[A]$ is the local-to-global transformation matrix for elements with sweep and anhedral [7].

The global aeroelastic equation for the whole blade is obtained by assembling the elemental matrices from the kinetic and strain energy contributions,

$$[M]\{\ddot{q}\} + [C]\{\dot{q}\} + [K]\{q\} = \{F_{AD}\} \quad (9)$$

where $[C]$ is the damping matrix and $\{F_{AD}\}$ is the aerodynamic force contribution.

The derivation and validation of the structural and aerodynamic models and the procedures for complete aeroelastic analysis have been detailed in [5-7]. The current work is concerned with only the whirl tower analysis (isolated rotor). The flowchart in Fig. 2 summarizes the procedure for whirl tower analysis. The loads, inflow and response equations are all given as differential equations and solved iteratively till a converged solution is obtained over several revolutions.

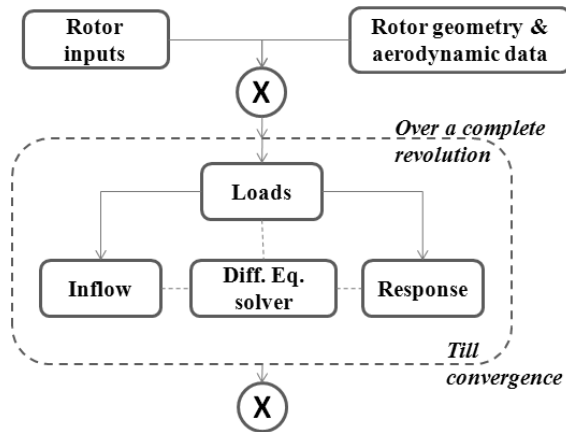


Figure 2. Flowchart for whirl tower analysis

3 Results

3.1 Structural Dynamics

The natural frequencies of the three blades at a rotational speed of 32.8 rad/s are given in the Table 1 for comparison. The modes have been identified only for the straight blade and not for the other two blades because of the bending-torsion and the bending-axial coupling due to sweep and anhedral. In the case of rotor blades with sweep/anhedral, apart from the mass and stiffness distribution, additional factors affect the natural frequencies. In the current case, the forward and backward sweep of the blade results in shifting of the c.g. of the blade which results in centrifugal stiffening and twisting of the blade. Apart from this, the reduction of net rotor radius in the case of double-swept blade also affects the frequencies. From the table, it is seen that lag frequencies are slightly increasing with increase in double-sweep while the lower flap frequencies are slightly decreasing. The higher flap frequencies and torsion are the most

Table 1. Natural frequencies @ 32.8 rad/s

Blade a	Blade b	Blade c
0.8647(1L)	0.8780	0.8842
6.9053(2L)	6.9975	7.2025
1.1764(1F)	1.1733	1.1700
3.1272(2F)	3.1192	3.1130
5.8738(3F)	6.7519	6.8842
9.54(4F)	10.133	10.768
5.8025(1T)	5.3384	5.2381

Note: L-Lag, F-Flap, T-Torsion

affected by the double-sweep. The higher flap frequencies increase with increase in double-sweep while the torsion frequencies have a decreasing trend.

The mode shapes of the three blades are studied next. As pointed out, the double-sweep feature introduces bending-torsion-axial coupling in the blade. Figure 3 shows the first lag mode along with the corresponding coupled torsion, and flap components. It can be seen that the secondary torsion and flap components increase with increase in the sweep/anhedral angles. Similarly, in Fig 4, it is seen that the secondary torsion and lag components of the flap mode increase with increase in the sweep/anhedral angles.

3.2 Whirl Tower Loads

The geometry of the rotor used for the whirl tower analysis is given in Table 2. The rotor is first given only collective angle inputs. In the next case, a combination of collective and cyclic angles is given as input.

Collective input

Figure 5 shows the effects of the double-sweep feature on thrust and power coefficients. With increase in the sweep and anhedral angles, there is only a slight increase in thrust; however, the power required increases 3 times. This is because of the increase in drag. The junctions of geometry changes in the rotor blade are important sections from design considerations. Figure 7 shows the moments at 0.5R which is the junction between the straight and swept portions of the blade. The flap moment curve (Fig 6(a)) decreases slightly and then increases with increase in the double-sweep. This is because of the

Table 2. Rotor data of blade (a)

Parameters		Value
Number of blades	N_b	4
Air density at sea level	ρ (kg/m ³)	1.224
Blade mass distribution	m_0 (kg/m)	8.45
Blade chord	c (m)	0.5
Rotor radius	R (m)	6.6
Solidity ratio	σ	0.09646
Weight coefficient	C_w	0.00734
Lift-curve slope	$C_{l\alpha}$	5.73
Profile drag coefficient	C_{d0}	0.01
Lock number	γ	6.4
Rotational speed	Ω (rad/s)	32.8

increase in the offset of the c.g of the blade from the original radial line. The lag bending moment at 0.5R (Fig 6(b)) which is close to zero in the straight blade increases drastically with increase in the double-sweep. This implies that to introduce a double-sweep in the geometry, the junction needs to be sufficiently stiffened. The torsional moment (Fig 6(c)) is seen to increase with increase in the double-sweep.

Figure 7 shows the effects on the root bending moments. In general, flap, lag and torsional bending moments at the root are seen to increase with increase in the sweep/anhedral angles.

Collective input together with Cyclic inputs

Cyclic inputs produce oscillatory loads in the whirl tower blades. In this section, the effects of double-sweep on the blade oscillatory loads due to 5° collective, applied along with 2° lateral and 2° longitudinal cyclic, are studied.

Figure 8 shows the tip response of the rotor blade due to application of collective and cyclic inputs. It is seen that the flap, lag and torsion responses increase with increase in sweep angle. The increases in lag and torsion responses are drastic with increase in double-sweep. The increase in lag is because of the increase in drag caused by the double-swept portion of the blade and the increase in torsion is caused by the offset of the c.g of the double-swept part of the blade from the original radial line.

Figure 9 shows the moments at $0.5R$ due to the application of collective and cyclic inputs. Flap moments (Fig 9(a)) at the junction slightly decrease with double-sweep; however, lag (Fig 9(b)) and torsion (Fig 9(c)) moments increase significantly with increase in sweep angle. Figure 10 shows the root blade moments due to application of collective and cyclic inputs. The flap, lag and torsion moments increase with increase in the sweep angle.

4 Concluding Remarks

In this study, the dynamic characteristics of rotor blades with double-sweep and tip-anhedral were compared with those of a straight blade. Two blades were given the double-sweep features, one with mild sweep/anhedral angles and another with large sweep/anhedral angles. The following observations can be made based on this study:

1. The double-sweep geometry introduces bending-torsion-axial coupling in the rotor blade.
2. The natural frequencies of the rotor blade are dependent on the magnitude of the sweep/anhedral angles. The double-sweep causes a shift in the c.g of that portion of the blade which contributes to the centrifugal stiffening. Double-sweep also reduces the net radius of the rotor disc in this case-study. These factors affect the natural frequencies of the blade. Torsional and higher flapping frequencies were seen to be the most affected.
3. In the whirl tower simulation with only collective input, the double-sweep geometry resulted in drastic increase in power requirement because of the increase in drag. The lag and torsion moments at junction of straight and swept portions ($0.5R$) increased significantly and are hence, a major design factor. The torsion moment increase is because of the offset of the c.g of the swept portion of the blade from the original radial line. These increases in the moments at the junction are also carried over to the root moments.
4. In the whirl tower simulation with collective and cyclic inputs, the double-sweep geometry resulted in increase of tip response. The bending moments at the junction between straight and swept portion increased substantially and these increases were carried over to the root moments.

5 References

- [1] M. Gervais, V. Gareton. Analysis of Main Rotor Noise Reduction Due to Novel Planform Design – The Blue Edge™ Blade. 37th European Rotorcraft Forum, Italy, 2011.
- [2] P. Rauch et al. Blue Edge™: The Design, Development and Testing of a New Blade Concept. 67th Annual Forum of the AHS, USA, May 2011
- [3] V. K. Truong. Dynamics studies of the ERATO blade, based on Finite Element Analysis. 31st European Rotorcraft Forum, Italy, 2005.
- [4] B.G. van der Wall et al. From ERATO Basic Research to the Blue Edge™ Rotor Blade. 72nd Annual Forum of the AHS, USA, May 2016
- [5] S. Palika. Structural Dynamic Analysis of Articulated Bearingless Hingeless Rotor Systems. Masters Dissertation, I.I.T. Kanpur, India, 2012
- [6] M. Rohin Kumar and C. Venkatesan. Rotorcraft Aeroelastic Analysis using Dynamic Wake / Dynamic Stall Models and its Validation. Journal of Aeroelasticity and Structural Dynamics, Vol 3, No 1, 2013
- [7] M. Rohin Kumar and C. Venkatesan. Effects of blade configuration parameters on helicopter rotor structural dynamics and whirl tower loads. The Aeronautical Journal 120, No. 1224, 2016
- [8] D.A. Peters and C.J. He. Correlation of Measured Induced Velocities with a Finite-State Wake Model. Journal of the American Helicopter Society, 36, No.3, 1991.
- [9] D. Petot. Differential Equation Modeling of Dynamic Stall. La Recherche Aerospatiale (English Edition), 5, 1989.

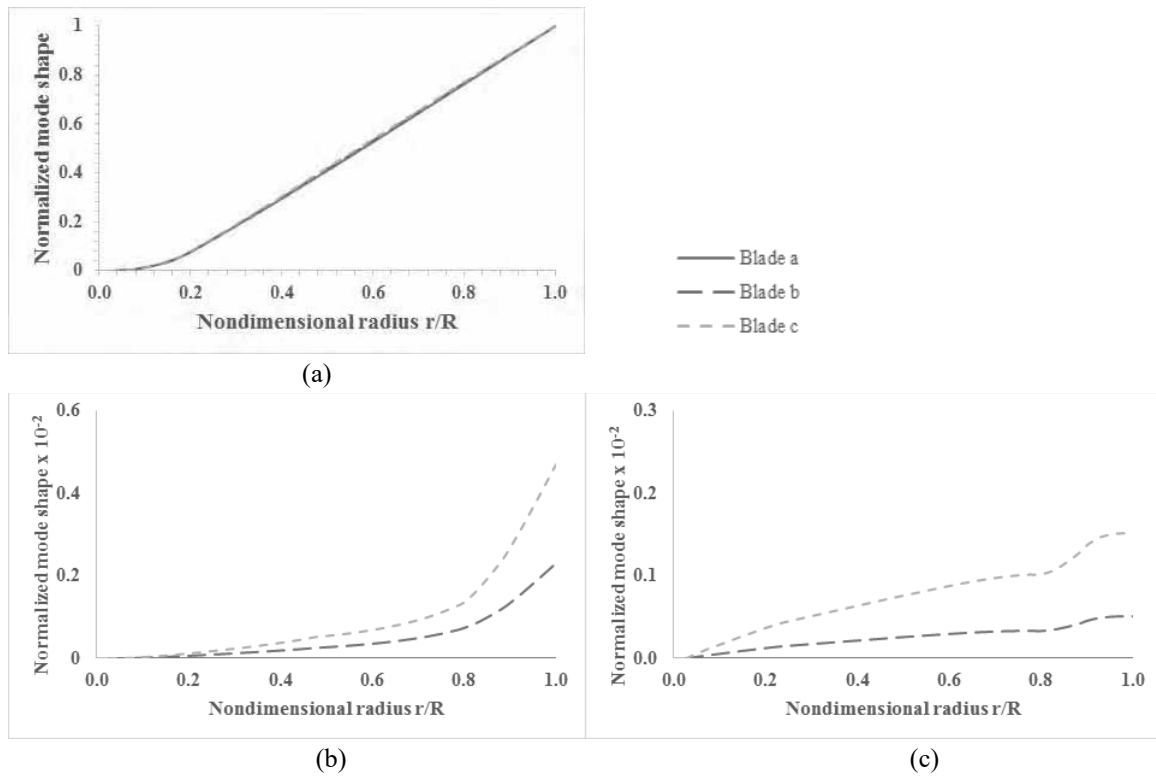


Figure 3. (a) 1st Lag mode, (b) Flap participation in this mode, and (c) Torsion participation in this mode

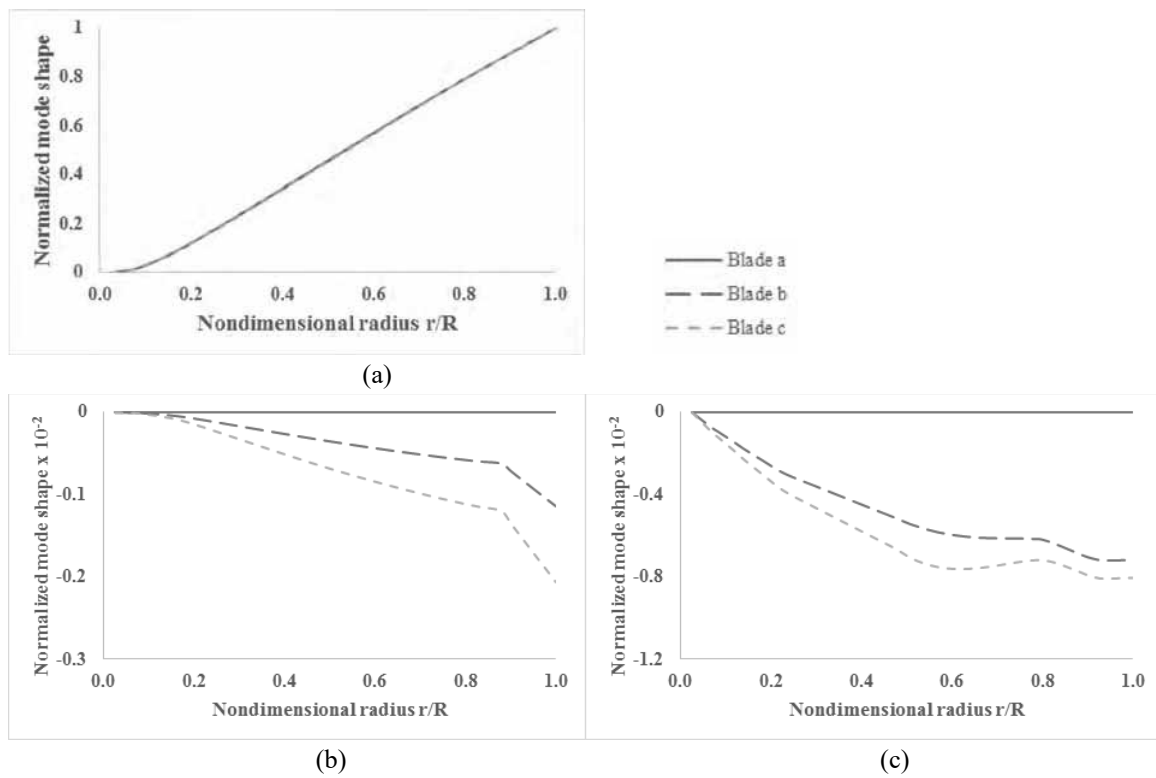


Figure 4. (a) 1st Flap mode, (b) Lag participation in this mode, and (c) Torsion participation in this mode

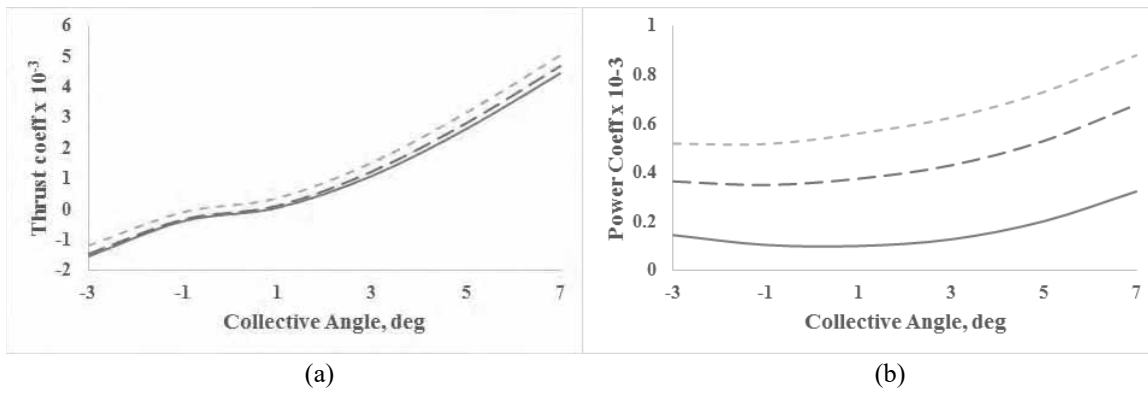


Figure 5. (a) Thrust and (b) Power variation with Collective input

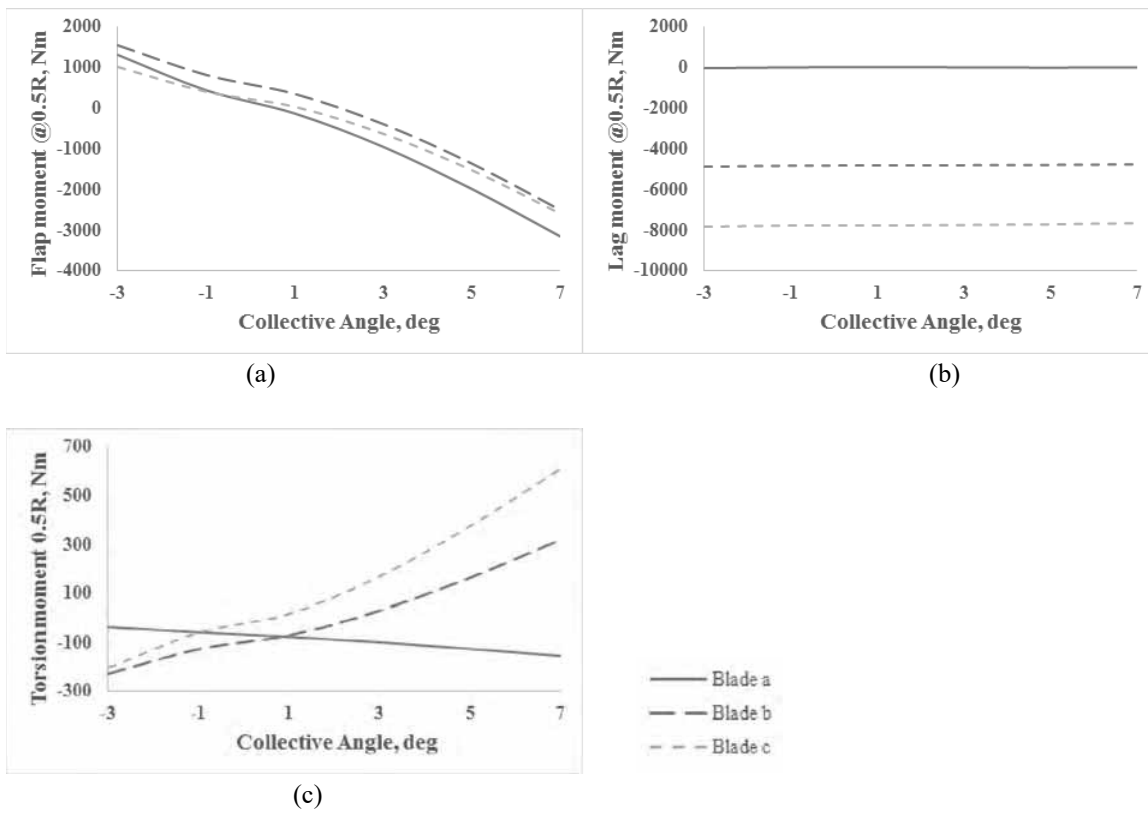


Figure 6. Variation of Moments @0.5R due to Collective input

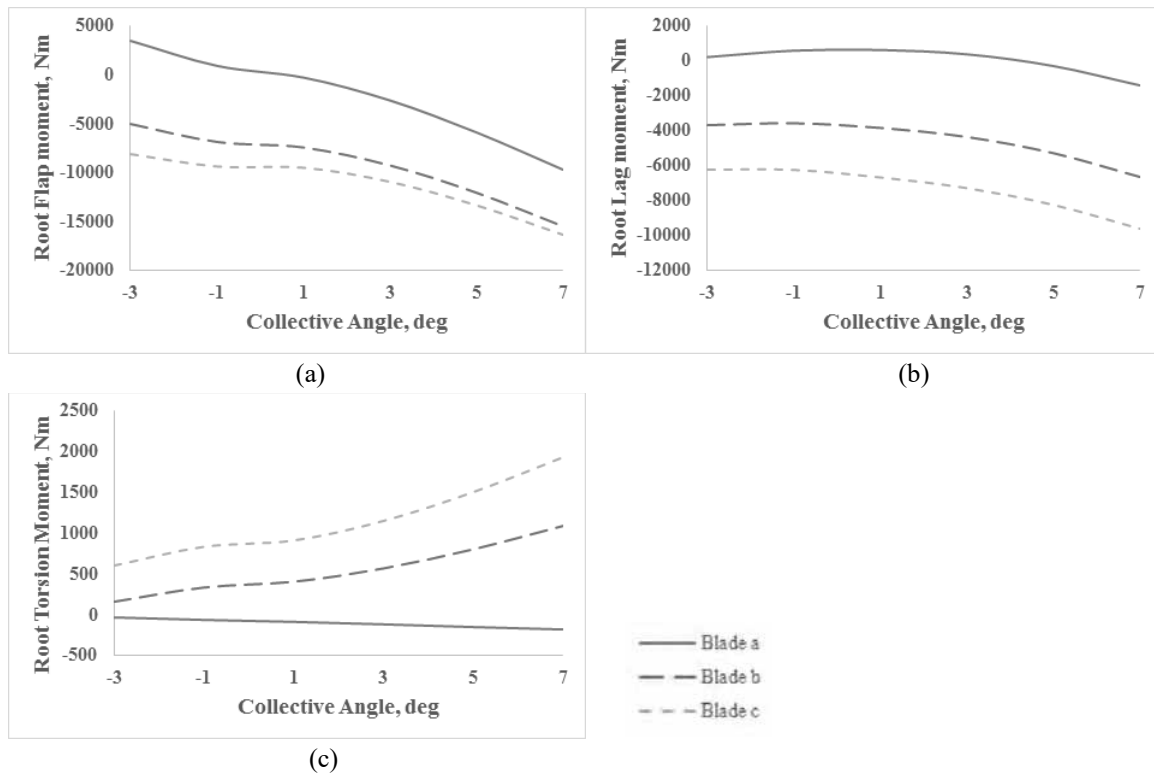


Figure 7. Variation of Root Moments with only Collective input

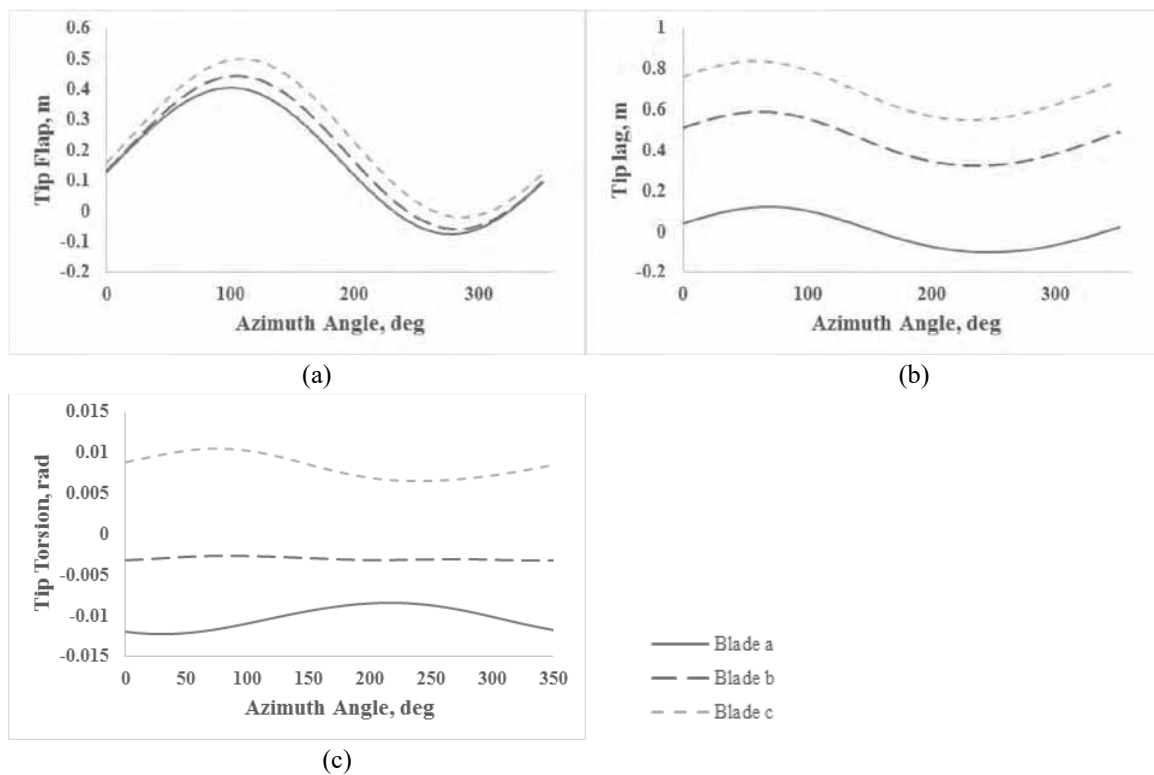


Figure 8. Tip Response due to 5° collective, 2° lateral and 2° longitudinal input

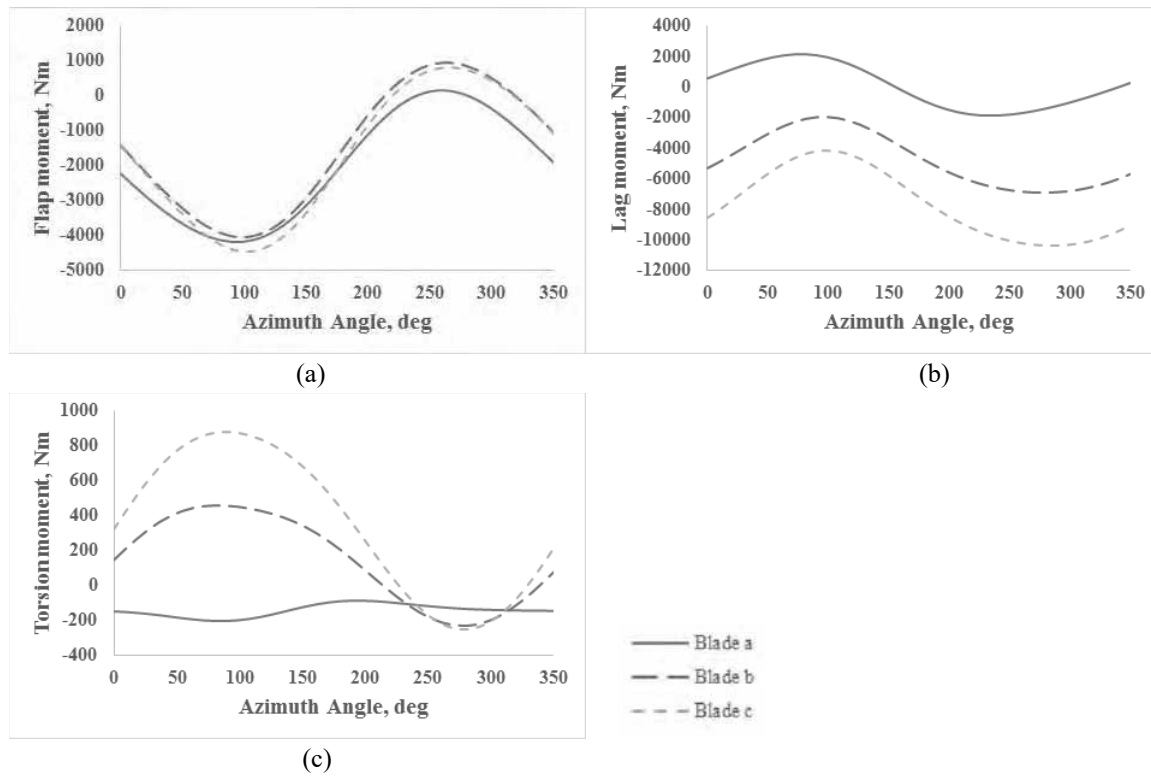


Figure 9. Moments @ 0.5R due to 5° collective, 2° lateral and 2° longitudinal input

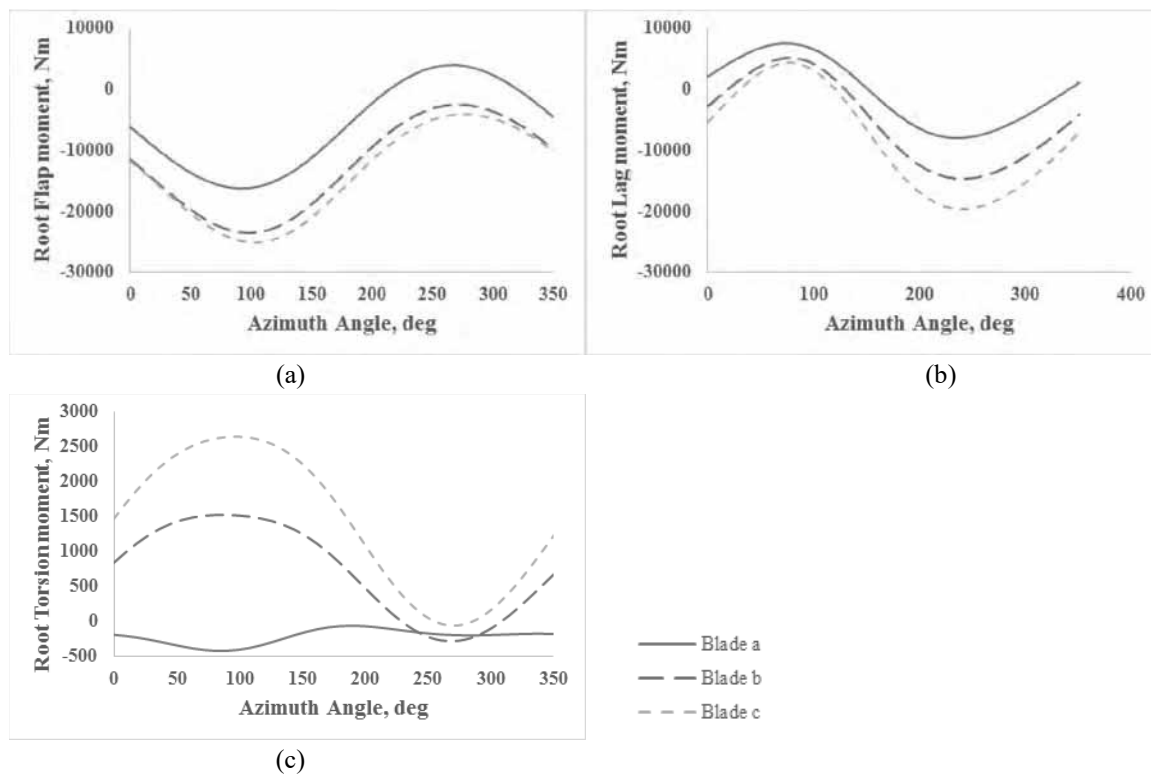


Figure 10. Root Moments due to 5° collective, 2° lateral and 2° longitudinal input

Model-based Development of Enhanced Ground Proximity Warning System for Heterogeneous Multi-Core Architectures

Umut Durak¹, David Müller¹, Jürgen Becker², Nikolaos S. Voros³, Panayiotis Alefragis³, Timo Stripf², Pierre-Aimé Agnel⁴, Gerard Rauwerda⁵, Kim Sunesen⁵

¹DLR Institute of Flight Systems

²Karlsruhe Institute of Technology

³Technological Educational Institute of Western Greece

⁴Scilab Enterprises

⁵Recore Systems

umut.durak@dlr.de

The aerospace domain, very much similar to other cyber-physical systems domains such as automotive or automation, is demanding new methodologies and approaches for increasing performance and reducing cost, while maintaining safety levels and programmability. While the heterogeneous multi-core architectures seem promising, apart from certification issues, there is a solid necessity for complex toolchains and programming processes for exploiting their full potential. The ARGO (WCET-Aware PaRallelization of Model-Based Applications for HeteroGeneOus Parallel Systems) project is addressing this challenge by providing an integrated toolchain that realizes an innovative holistic approach for programming heterogeneous multi-core systems in a model-based workflow. Model-based design elevates systems modeling and promotes simulation with the executing these models for verification and validation of the design decisions. As a case study, the ARGO toolchain and workflow will be applied to a model-based Enhanced Ground Proximity Warning System (EGPWS) development. EGPWS is a readily available system in current aircraft which provides alerts and warnings for obstacles and terrain along the flight path utilizing high resolution terrain databases, Global Positioning System and other sensors-. After a gentle introduction to the model-based development approach of the ARGO project for the heterogeneous multi-core architectures, the EGPWS and the EGPWS systems modelling will be presented.

1 Introduction

The trend in avionics architectures is shifting towards more central computing platforms which are categorized as Integrated Modular Avionics (IMA) [1]. Rather than decentralized and dedicated computing cards, in IMA, multiple applications utilize the same computing card [2]. The operating system allows the operation of independent application software in partitions in order to address safety requirements. Partitions are defined as isolated execution environments with separate sets of resources that guarantee resource availability and timing. Furthermore, there are some recent efforts that target parallelization and utilization of multi-core architectures in IMA.

In 2012, Nowatsch and Paulitsch from EADS Innovation Works examined the utilization of multi-core systems in partitioned environments like IMA for running applications of different safety-criticality [3].

In 2013, Karray and Paulitsch from EADS Innovation Works with Koppenhöfer and Geiger from CASSIDIAN presented the non-functional requirements for the application of multi-core architectures for a degraded vision landing system for a helicopter [4]. In 2015, Koppenhöfer and Geiger presented a Helicopter Terrain Awareness and Warning System (HTAWS) as a sample application of their demonstrator [5]. They aim at providing a comprehensive, map based overview of a helicopter's surroundings to prevent avoidable collision with ground or obstacles.

In parallel with these efforts, Agrou and colleagues from THALES presented design principles of predictable and efficient multi-core systems to meet embedded computer requirements in avionics [6]. In 2014, Löfwenmark from Saab Aeronautics and Nadjm-Tehrani from Linköping University presented challenges and described research directions to ad-

dress guaranteeing determinism for avionic applications running on multiple cores and interacting through shared memory [7].

While these efforts reported initial results of parallelization in flight systems development using multi-core architectures, they do concentrate on the applicability regarding the safety constraints of the avionics domain. Nevertheless, there is no reported effort that attacks the development methodology for avionics application using multi-core architectures. The aerospace domain is thus demanding complex toolchains and programming processes for exploiting the full potential of these next generation heterogeneous parallel platforms.

The rise of model-based approaches has been phenomenal. System architecture is defined as the structure of system components, relationships and rules governing their design and evolution over time [8]. In model-based approaches the models of system architectures, namely system models, are placed in the center of the development process. Simulation is utilized with executing system models as the native mechanisms to address measures of performance and

measures of effectiveness throughout conceptual design, development and later life cycle phases [9]. The productivity is boosted with generation of systems development artefacts including software code through transformations and stepwise refinement of system models [10].

The ARGO (WCET-Aware PaRallelization of Model-Based Applications for HeteroGeneOus Parallel Systems) project is addressing the development of heterogeneous multi-core systems by providing an integrated toolchain that realizes a model-based workflow.

The ARGO toolchain and workflow will be validated with a model-based Enhanced Ground Proximity Warning System (EGPWS) development case study. EGPWS is selected due to its feature set that is suitable for parallelization. It can benefit a lot from multi-core architectures for performance and feature enhancement. In the following sections, the model-based development approach of the ARGO project will gently be introduced. Then the EGPWS and the EGPWS systems modelling will be presented.

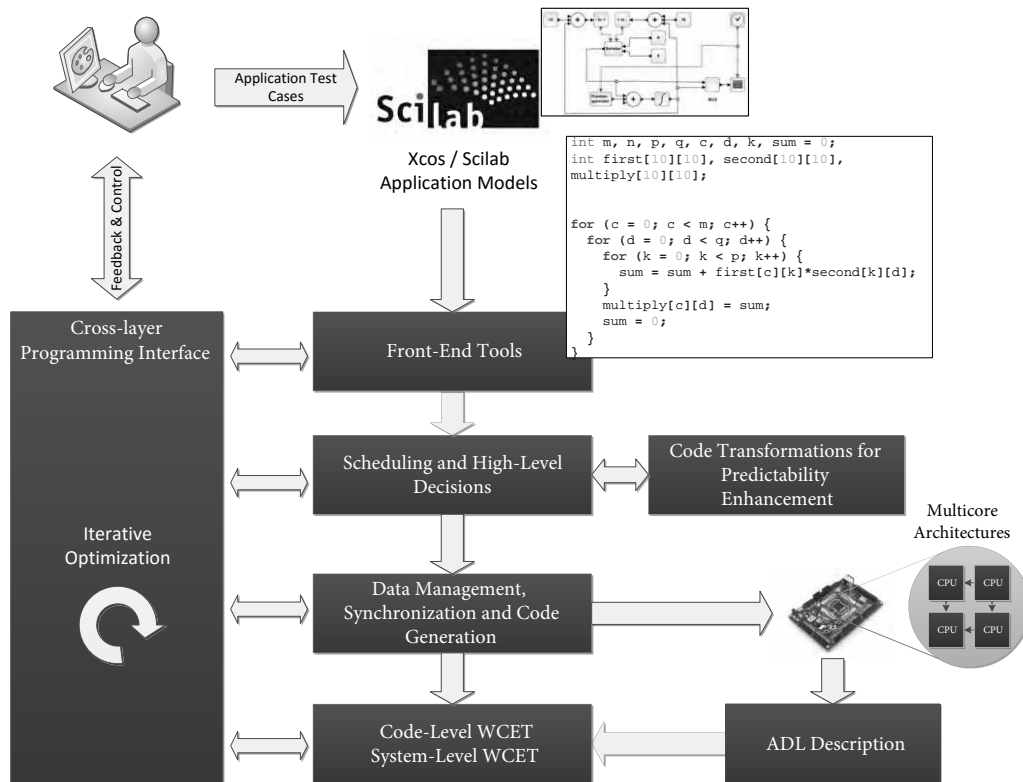


Figure 1: ARGO Workflow and Toolchain

2 Model-based Development Approach for Heterogeneous Multi-Core Architectures

Model-based approaches propose the development of models and generation of executable software entities through successive model-to-model and model-to-text transformations [11]. The model-based development is a model-based approach that is characterized by the seamless use of executable and graphical data flow oriented block diagram models and state machines for system specification, design and implementation, employing modeling and simulation tools such as Scilab/Xcos or MATLAB/Simulink [12].

The objective of the model-based development approach (Figure 1) of the ARGO project is to design, implement and deploy hard real-time applications on multi-core targets through parallel code generation with top-notch Worst Case Execution Time (WCET) analysis in a programming environment that will guarantee efficiency and productivity. The approach extends previous work to cover real-time applications [13].

The model-based development environment allows engineers to design a system from a high-level point of view. Design models specify executable system architecture. Model-in-the-Loop (MIL) simulations are used for the early validation of the systems design. Code generation and code transformations are performed with a strong objective of keeping the code base predictable or warning the user as early as possible of possible problems in WCET estimation in the current design. The targeted architecture, defined with an Architecture Description Language (ADL), and specific low level transformations ensure parallelization with WCET constraints as tight as possible. Targets include any hardware platform with a parallel programming model that can express time-predictable computation and communication. Software-in-the-Loop (SIL) simulations that also exploit target specifications are used to advance the validation of the design. In the ARGO project the approach will be evaluated on the multi-core platform of Recore Systems, a specialist in flexible multi-core platforms and subsystems IP [14]. Hardware-in-the-Loop (HIL) simulations will be used to validate the performance of the system.

Constant feedback is provided to the user at each step. The possibility to select the transformations and perform them in an interactive manner results in a

semi-automatic, guided process. The models are enriched with the results of the code generation, the real time constraints analysis and x-in-the-loop simulations, thus tracing and controlling the results of an iteration of the process for early verification and validation.

3 Enhanced Ground Proximity Warning System

EGPWS is a name that is used for current Terrain Awareness and Warning Systems (TAWS) which aim to prevent controlled flight into the terrain. There are various TAWS options available in the market for various platforms in various configurations. Examples may include EGPWS from Honeywell [15], T2CAS from ACSS [16], LANDMARK™ from L3 [17] and TAWS from Universal Avionics [18]. A brief comparison of these systems and more can be found in [19].

The core feature set of EGPWS is to create visual and aural warnings in order to avoid controlled flight into the terrain. These warnings are categorized in 5 modes:

Mode 1: Excessive Descent Rate provides alerts for excessive descent rates for all phases of flight.

Mode 2: Excessive Terrain Closure Rate provides alerts to protect the aircraft from impacting the ground when terrain is rising rapidly with respect to the aircraft.

Mode 3: Altitude Loss After Take-off provides alerts when a significant altitude loss is detected after take-off or during a low altitude go around.

Mode 4: Unsafe Terrain Clearance provides alerts when there is no sufficient terrain clearance regarding the phase of the flight, aircraft configuration and speed.

Mode 5: Excessive Deviation Below Glideslope provides alerts when the aircraft descends below the glideslope.

The modes 1 to 5 are regarded as suitable for coarse grain parallelization.

Additionally, an EGPWS provides some enhanced functions based on a terrain database. These functions are:

Terrain Awareness Display (TAD) provides an image of the surrounding terrain represented in various

colors on the Navigation Display as well as the warnings and cautions regarding the terrain interactions.

Terrain Clearance Floor (TCF) provides a low terrain warning during landing and thus enhances the basic functions with alerts for the descent below a

predefined “Terrain Clearance Floor” disregarding the aircraft configuration.

The terrain processing and particularly collision detection algorithms that are required for TAD and TCF are regarded as candidates for fine grain parallelization.

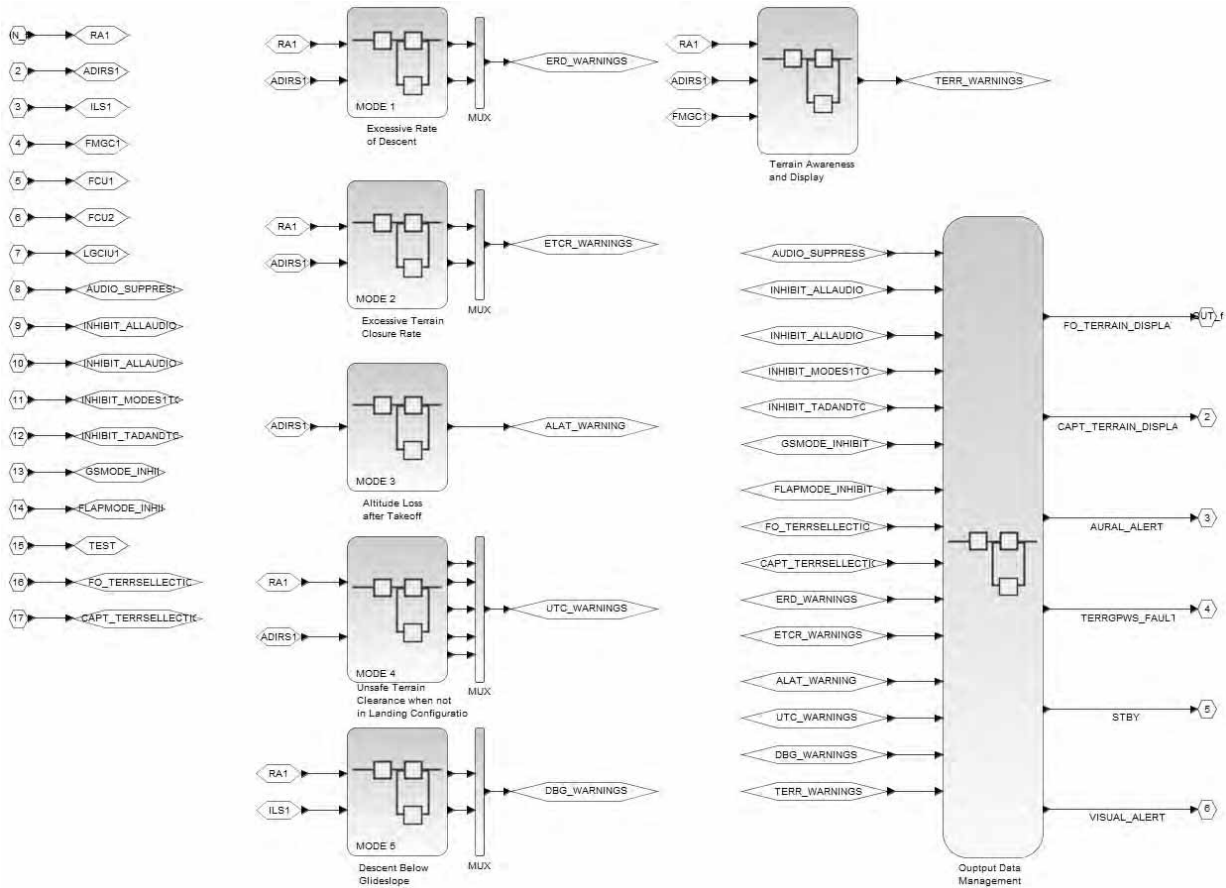


Figure 2: Top Level EGPWS Model

4 EGPWS Systems Modeling

Figure 2 shows the top level of the ARGO EGPWS prototype model. The model is being developed using the graphical modeling environment Scilab/Xcos [20].

The ARGO EGPWS will be designed based on a commercial system as it is deployed in DLR’s Advanced Technology Research Aircraft (ATRA). Therefore, the development refers to the EGPWS description in the A320’s Flight Crew Operating Manual (FCOM; section 1.34.70 in [21]). ATRA’s EGPWS is supplied by Honeywell.

4.1 EGPWS Modes 1 to 5

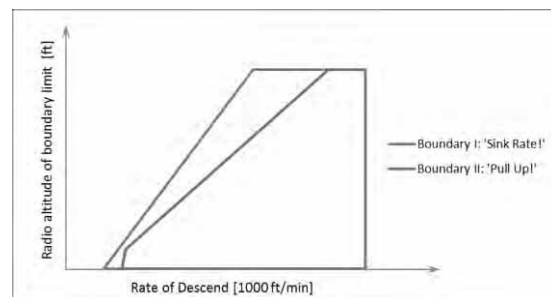


Figure 3: EGPWS Mode 1 Boundaries

In the FCOM, the functionalities of the modes 1 to 5 are described using graphs (Figure 3) that show the limit altitudes (the reference being the radio altitude)

associated with each mode as functions of other parameters like airspeed or rate of descent.

By using Xcos' "Interpolation" blocks, those graphs are modeled for the ARGO EGPWS. For an example see Figure 4. An Interpolation block needs to be provided with two vectors for parametrization, one containing a selection of input and the other a corresponding number of output data points of the function that has to be modeled. It is between these points that the output matching a given input can then be interpolated.

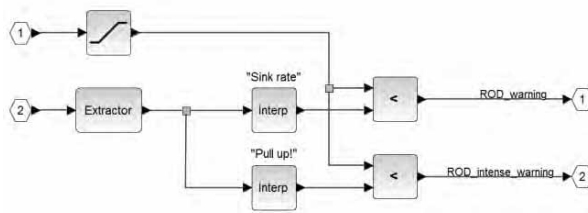


Figure 4: Xcos Model of EGPWS Mode 1

Figure 4 shows the implementation of Mode 1 as a model in Xcos. Mode 1 focuses on the aircraft's Rate Of Descend (ROD) within a medium proximity to the terrain. For every ROD value, there is a limit altitude associated to it. In this way, two different boundaries are formed for Mode 1, each triggering a vocal alert and illumination of a designated GPWS warning lamp in the cockpit. When penetrated by sinking below the limit altitude, the first boundary causes a repetitive "SINK RATE!", while the second one triggers a more demanding "PULL UP!". This alert is also repeated until the aircraft climbs above the limit altitude or reduces its ROD.

Each of the two boundaries is modeled using an Interpolation block, as can be seen in Figure 4. The input, namely the aircraft's ROD, is taken from a signal vector (which simulates a data bus called ADIRS1 - Air Data Inertial Reference System - in the real A320) by utilizing an "Extractor" block. This block allows extracting a single signal out of a bus or multiplex signal.

The altitude limit obtained through the interpolation is then compared to the aircraft's actual radio altitude. This is the signal from input port 1 in Figure 4. If its value is lower than the computed limit, the signals "ROD_warning" or "ROD_intense_warning" are set to the value 1, which acts as a trigger to the associated vocal alert and the warning lamp.

The radio altitude signal runs through a "Saturation" block which imposes limit values on a signal. It is used here to make sure that Mode 1 does not give out warnings when the aircraft is on ground. This is done by limiting the signal value to 10 ft above ground level and above.

4.2 Terrain Awareness Display and Terrain Clearance Floor

The Terrain Awareness Display and Terrain Clearance Floor features of an EGPWS need a terrain database from which they can gather information about the terrain surrounding the aircraft's current position during flight.

The 3D representation of the terrain is referred to as Digital Elevation Model (DEM) [22]. It is available as elevation data organized in the form of a matrix. Regarding the increasing demand for DEMs with global coverage, the Shuttle Radar Topography Mission (SRTM) provided global high quality DEMs at resolution levels of 1 arc second (~30 m) or 3 arc second (~90 m)[23]. The ARGO EGPWS terrain databases are created using SRTM 3 arc second data.

Two-phase processing, namely broad phase and narrow phase, is a common approach in collision detection algorithms [24]. While the broad phase is used to identify the particular terrain database segments to be used, narrow phase uses these segments for calculating colors and their densities in the TAD as well as the TAD and TCF warnings and cautions.

In broad phase, spatial partitioning techniques are utilized for identifying the segments of the terrain database to be processed. Uniform grids are used to divide the terrain into equally sized regions that are associated to a database segment. This way, an easy and fast terrain data access mechanism is developed for the given coordinates of the airplane. While the initial grid size is selected as 1 degree, it will be further tuned for optimizing the overall performance.

The TAD terrain picture and TCF are straight-forward computation of the narrow phase in which the elevation of terrain data points is compared to the aircraft as a point, either for collision as in TCF or for color mapping as in the TAD terrain picture. However, the warnings and cautions from the TAD algorithm require a relatively complex collision detection processing: the vertical and horizontal terrain caution and warning envelopes define two polygons. The intersection of these polygons and the terrain is used

to trigger the related caution and warning messages. The narrow phase is responsible for the collision detection between caution and warning envelopes and the terrain. A comprehensive survey of collision detection algorithms can be found in [25]. Image-based algorithms have been employed for making use of the processing power of graphics cards [26]. The interference test is conducted based on a depth map and is maintained in an image buffer which is generated by projecting the object on a plane. In the ARGO EGPWS vertical ray casting is employed in points of the terrain database and the depth map of the terrain caution and warning envelopes is then compared to the elevation data of the particular point to identify the collision (Figure 5).

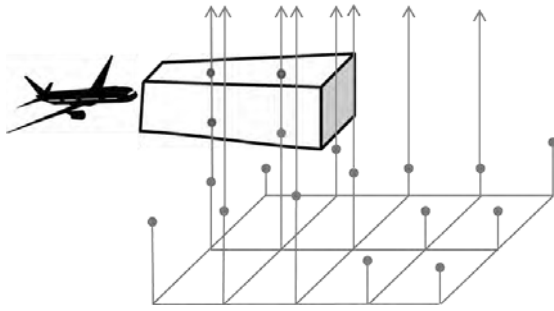


Figure 5: Collision Detection Approach

As opposed to the model elements that represent the ARGO EGPWS modes, which are purely Xcos based, the TAD and TCF algorithms are being developed

using Scilab scripts and are integrated to the Xcos model as a user defined block.

4.3 Output Data Management

According to the current system architecture, the modes 1 to 5 as well as the terrain functions TAD and TCF reside in separate Xcos blocks. In addition, there is a block containing the Output Data Management, which evaluates its inputs in order to create triggers for the appropriate visual and aural warnings.

The core of the Output Data Management is an algorithm that applies a priority list to the trigger signals. This is done to avoid several alerts being active at the same time in the case of more than one trigger signal having the value 1.

The entire trigger signals being used in the ARGO EGPWS and the modes to which they belong are listed in Table 1. Although the modes are designed to detect different critical situations, the pilot's task is always the same: avoid impacting the terrain, either by a change of course or, especially after urgent warnings, by pulling up and gaining altitude. For this reason, some of the vocal cues are the same and can thus share the same level of priority, making the algorithm less complex.

Table 1: Names of trigger signals in the ARGO EGPWS and their respective vocal cues

Nr.	Mode	name of trigger signal	vocal cue	situation	priority
1.	1	ERD_warning	Sink rate!	always	2
2.		ERD_intense_warning	Pull up!	always	1
3.	2	ETCRa_warning	Terrain!	always	6
4.		ETCRb_warning	Terrain!	always	6
5.		ETCR_intense_warning	Pull up!	always	1
6.	3	ALAT_warning	Don't sink!	take-off	7
7.	4	UTCa_warning	Too low, terrain!	cruise/approach	5
8.		UTCa_gear_warning	Too low, gear!	cruise/approach	4
9.		UTCb_warning	Too low, terrain!	cruise/approach	5
10.		UTCb_flaps_warning	Too low, flaps!	cruise/approach	4
11.		UTCc_warning	Too low, terrain!	take-off	3
12.	5	DBG_warning	Glideslope!	approach	8
13.		DBG_intense_warning	GLIDESLOPE!	approach	7
14.	TAD	TAD_caution	Terrain ahead!	always	5
15.		TAD_warning	Terrain ahead, pull up!	always	1
16.	TCF	TCF_warning	Too low, terrain!	cruise/approach	5

Table 2: Priority rating of signals in Table 1 and assignment to the main phases of flight

situation: priority:	take-off / missed approach	cruise	approach / landing
1	ETCR_intense_warning ERD_intense_warning TAD_warning	ETCR_intense_warning ERD_intense_warning TAD_warning	ETCR_intense_warning ERD_intense_warning TAD_warning
2	ERD_warning	ERD_warning	ERD_warning
3	UTCc_warning		
4	UTCb_flaps_warning UTCa_gear_warning	UTCb_flaps_warning UTCa_gear_warning	UTCb_flaps_warning UTCa_gear_warning
5	UTCb_warning UTCa_warning TCF_warning TAD_caution	UTCb_warning UTCa_warning TCF_warning TAD_caution	UTCb_warning UTCa_warning TCF_warning TAD_caution
6	ETCRa_warning ETCRb_warning	ETCRa_warning ETCRb_warning	ETCRa_warning ETCRb_warning
7			DBG_intense_warning
8	ALAT_warning		DBG_warning

Table 3: Abbreviations in signal names in Table 1 and Table 2

Abbreviation	Explanation	Abbreviation	Explanation
ALAT	Altitude Loss After Take-Off	TAD	Terrain Awareness Display
DBG	Deviation Below Glideslope	TCF	Terrain Clearance Floor
ERD	Excessive Rate of Descent	UTC	Unsafe Terrain Clearance
ETCR	Excessive Terrain Closure Rate		

Table 2 lists the trigger signals again, organized by their level of priority and assigned to the phases of flight in which they are relevant. This serves to point out that the warnings of Mode 3 (Altitude Loss After Take-off) and Mode 5 (Excessive Deviation Below Glideslope), which are designed specifically for take-off and approach, respectively, are considered less urgent in the ARGO EGPWS than the warnings designed for the whole flight envelope. Furthermore, the highest priorities are given to the warnings that directly demand the pilot to pull up.

Table 3 presents the explanations for the abbreviations used in the signal names in Table 1 and Table 2.

The algorithm will also handle additional influences on the triggering of alerts, such as pushbuttons in the cockpit that allow the pilot to alter the EGPWS settings to his needs. For example, there are two buttons in the overhead panel which are labeled “SYS – OFF” and “G/S MODE – OFF”. Their purpose is to disable all of the EGPWS Modes or just Mode 5,

respectively. Other buttons may inhibit the use of aural alerts, leaving only the optical cues to catch the pilot’s attention.

The logic that is represented in the tables will be modeled using state machines which are implemented in Scilab/Xcos as Automata (finite-state machine) block [27].

Conclusion

After introducing the recent advance on heterogeneous multi-core architectures in avionics, the paper gently presents the model-based development approach of the ARGO project. This approach is being exercised in the development of ARGO Enhanced Ground Proximity Warning System due the suitability of its feature set for parallelization.

In the modeling, modes and Output Data Management are developed using Xcos, while Scilab scripting is used for the Terrain Awareness Display and Terrain Clearance Floor calculations. Thereby we aim

at evaluating diverse model-based parallel application development capabilities of the ARGO approach.

As the initial prototype of the system model has been constructed, the future work will include x-in-the-loop testing. The first step will be from model-in-the-loop testing which will be eventually followed by software-in-the loop and hardware-in-the loop testing with the utilization of the ARGO toolchain for code generation.

5 Acknowledgement

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No 688131 — ARGO.Action. www.argo-project.eu

6 References

- [1] P. J. Prisaznuk. *Integrated modular avionics*. Proceedings of the IEEE 1992 National Aerospace and Electronics Conference, pp.39-45, 1992.
- [2] C. B. Watkins, and Randy Walter. *Transitioning from federated avionics architectures to integrated modular avionics*. IEEE/AIAA 26th Digital Avionics Systems Conference, pp. 2-A, 2007.
- [3] J. Nowotsch and M. Paulitsch. *Leveraging multi-core computing architectures in avionics*. 9th European Dependable Computing Conference (EDCC), pp. 132-143, 2012.
- [4] H. Karray, M. Paulitsch, B. Koppenhöfer, and D. Geiger. *Design and implementation of a degraded vision landing aid application on a multicore processor architecture for safety-critical application*. 16th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), pp. 1-8, 2013
- [5] B. Koppenhöfer, and D. Geiger. *EMC2 Use Case:Hybrid Avionics Integrated Architecture Demonstrator*. HiPEAC, Workshop EMC², 2016.
- [6] H. Agrou , P. Sainrat, M. Gatti, and P. Toillon. *Mastering the behavior of multi-core systems to match avionics requirements*. AIAA 31st Digital Avionics Systems Conference (DASC), pp.6E5-1, 2012.
- [7] A. Löfwenmark, and S. Nadjm-Tehrani. *Challenges in future avionic systems on multi-core platforms*. International Symposium on Software Reliability Engineering Workshops (ISSREW).pp.115-119, 2014.
- [8] A. Tolk, and T.K. Hughes. *Systems engineering, architecture, and simulation*. In: Modeling and Simulation-based Systems Engineering Handbook, Editors: D. Gianni, A. D'Ambrogio, and A. Tolk. CRC Press, pp. 11-41, 2014.
- [9] INCOSE, *Vision 2020*. No. INCOSE-TP-2004-004-02, INCOSE, 2007.
- [10] C. Atkinson and T. Kühne. *Model-driven development: a metamodeling foundation*. IEEE Software, Vol. 20, No. 5, pp. 36-41, 2003.
- [11] D. Gasevic, D. Djuric, and V. Devedic. *Model driven engineering and ontology development*. Springer Science & Business Media, 2009.
- [12] I. Stürmer, M. Conrad, I Fey and H Dörr. *Experiences with model and autocode reviews in model-based software development*. Proceedings of the 2006 International Workshop on Software Engineering for Automotive Systems, pp. 45-52, 2006.
- [13] T. Stripf, O. Oey, T Bruckschloegl, J. Becker, G. Rauwerda, K. Sunesen, G. Goulas, P. Alefragis, N.S. Voros, S. Derrien, and O. Sentieys. *Compiling Scilab to high performance embedded multicore systems*. Microprocessors and Microsystems, Vol.37, No. 8, pp.1033-1049, 2013.
- [14] L. Berrojo, R. Moreno, R. Regada, E. Garcia, R. Trautner, G. Rauwerda, K. Sunesen, Y. He, S. Redant, G. Thys, and J. Andersson. *Scalable sensor data processor: a multi-core payload data processor ASIC*. The international Space System Engineering Conference DASIA, in Barcelona, Spain, 2015.
- [15] Honeywell. *Terrain and Traffic Awareness*. Retrieved March 16, 2016 from <https://aerospace.honeywell.com/en/product-listing/terrain-and-traffic-awareness>
- [16] ACSS. *T2CAS*. Retrieved March 16, 2016 from www.acss.com/products/t2cas/
- [17] L3. *LANDMARK™ Terrain Awareness & Warning Systems*. Retrieved March 16, 2016

- from www.l-3avionics.com/products/landmark/
- [18] Universal Avionics. *TAWS | Terrain Awareness and Warning System*. Retrieved March 16, 2016 from www.uasc.com/home/shop/avionics/taws
 - [19] D. Smith. *Traffic Alert Collision Avoidance Systems—TCAS Buyer's Guide*. Pilot's Guide to Avionics 2005, pp. 34-41, 2005.
 - [20] S.L. Campbell, J.P. Chancelier, and R. Nikoukhah. *Modeling and Simulation in Scilab/Scicos*. Springer New York, 2006.
 - [21] Airbus Industries S.A.S., *A320 Flight Crew Operating Manual, System Description 1; REV 24*, AI Toulouse.
 - [22] Z. Li, C. Zhu, and C. Gold. *Digital terrain modeling: principles and methodology*. CRC Press, 2004.
 - [23] B. Rabus, M. Eineder, A. Roth, and R. Bamler. *The shuttle radar topography mission—a new class of digital elevation models acquired by spaceborne radar*. ISPRS Journal of Photogrammetry and Remote Sensing, Vol.57, No.4, pp.241-262, 2003.
 - [24] C. Ericson. *Real-time collision detection*. CRC Press, 2004.
 - [25] M. Lin, and S. Gottschalk. *Collision detection between geometric models: A survey*. Proceedings of IMA Conference on Mathematics of Surfaces Vol. 1, pp. 602-608, 1998.
 - [26] G. Baciu, W.S.K. Wong. *Image-based Collision Detection*, In: Integrated Image and Graphics Technologies, Springer, 75-94, 2006.
 - [27] M. Najafi, and R. Nikoukhah. *Implementation of hybrid automata in Scicos*. 2007 IEEE International Conference on Control Applications, pp. 819-824, 2007.

Numerische Lösung eines mathematischen Modells für eine optimale Krebskombinationstherapie aus Anti-Angiogenese und Strahlentherapie

Kurt Chudej¹, Dominik Huebner¹, Hans Josef Pesch¹
¹Lehrstuhl für Ingenieurmathematik, Universität Bayreuth
kurt.chudej@uni-bayreuth.de

Es gibt eine Reihe von verschiedenen mathematischen Krebsmodellen. Wir betrachten hier speziell das Modell AR6 von Ledzewicz/Schättler [5, 9] für das zwar einige theoretische Teilresultate vorhanden sind jedoch eine numerische Lösung bzw. eine Synthese der optimalen Lösung bisher fehlen. Die präsentierte numerische Lösung erweist sich als anspruchsvoll.

1 Krebsmodell ohne Therapie

Ausgangspunkt dieser Arbeit ist ein Krebsmodell von Hahnfeldt et al. [4] in der Formulierung als Optimales Steuerungsproblem nach Ledzewicz/Schättler [5, 9]. Das Modell beschreibt die Zelldynamik zweier Komponenten, zum einen des variablen Tumorumfangs p und zum anderen der ebenfalls variablen vaskulären Trägerkapazität q . Der Grund für diese zweite Komponente ist die starke Abhängigkeit des Tumors von Blutgefäßen, die ihn mit Nährstoffen und Sauerstoff versorgen. Hauptsächlich ist hierfür das Volumen der Endothelzellen entscheidend, weswegen q auch oft als endothelialer Träger bezeichnet wird.

Zunächst wird das Tumorumfang mit der Gompertz-schen Wachstumsfunktion modelliert:

$$\dot{p} = -\xi p \ln(p/q). \quad (1)$$

Die vaskuläre Trägerkapazität ist nicht konstant, da durch äußere Einflüsse angeregt fortlaufend Blutgefäße gebildet oder abgebaut werden können. Dafür ist insbesondere das Verhältnis von Stimulatoren und Inhibitoren entscheidend, die sowohl vom Körper selbst als auch von jeder Krebszelle produziert werden:

$$\dot{q} = -\mu q + S(p, q) - I(p, q). \quad (2)$$

Der erste Term $-\mu q$ steht für die Abnahme der vaskulären Trägerkapazität aufgrund natürlicher Ursachen,

wie beispielsweise durch Absterben. Da der Parameter μ relativ klein ist, ist der Einfluss dieses Effektes im Vergleich zu den anderen weitestgehend vernachlässigbar für die Dynamik und wird deshalb in manchen Modellen zur Vereinfachung sogar weggelassen wie zum Beispiel in [2]. Der zweite Term $S(p, q)$ beschreibt den durch ausgeschüttete Stimulatoren hervorgerufenen Zuwachs des endothelialen Trägers. Die von den Inhibitoren ausgelöste Abnahme wird durch den Term $I(p, q)$ modelliert. Hierbei werden in der Modellierung zur Vereinfachung nur die Stimulatoren und Inhibitoren berücksichtigt, die vom Tumor selbst ausgeschüttet werden [6].

Die verwendeten Ausdrücke $I(p, q)$ und $S(p, q)$ sind keinesfalls trivial; nach Prehn [7] unterscheidet sich die Wirkungsreichweite der von den Tumoren produzierten Stimulatoren und Inhibitoren: Während die von den Tumoren produzierten Inhibitoren im ganzen Körper zirkulieren, sind die wachstumsfördernden Stimulatoren nur lokal beim erzeugenden Tumor wirksam. Befinden sich zwei Tumore in einem Organismus, so ist die Gesamtanzahl aller Inhibitoren beim kleineren Tumor größer als die Anzahl der dort anzutreffenden Stimulatoren und demzufolge werden keine neuen Blutgefäße gebildet, die für eine weitere Entwicklung notwendig wären. Wegen seines zusätzlichen Volumens schüttet der größere Tumor mehr Stimulatoren aus als der kleinere. Infolgedessen sind in seinem Umfeld mehr stimulierende als hemmende

Faktoren vorhanden, und er kann durch die Neubildung von Blutgefäßen weiter wachsen.

Unter der Annahme, dass der Tumor kugelförmig ist, sowie weiterer Annahmen, gelingt es Hahnfeldt Formeln für $I(p, q)$ und $S(p, q)$ zu gewinnen:

$$I(p, q) = bp^{\frac{2}{3}}q, \quad S(p, q) = bp. \quad (3)$$

2 Modell der Anti-Angiogenese

Als indirekte Therapieform bekämpft die Anti-Angiogenese nicht die Krebszellen selbst, sondern verhindert ihr weiteres Wachstum, indem sie auf die für den Tumor notwendigen Blutgefäße abzielt. Genauer gesagt unterbrechen die Angiogenese-Hemmer das Wachstum der Endothelzellen, die für die Neubildung von Blutgefäßen unabdingbar sind. Für die Modellierung der Wirkungsweise der Anti-Angiogenese wird eine Kontrolle u eingeführt, die für die Dosis der Angiogenese-Hemmer steht. Aus mathematischer Sicht wählt man für u eine Lebesgue-messbare Funktion, deren Werte im Intervall $[0, u_{\max}]$ liegen, wobei u_{\max} die Maximaldosis der Angiogenese-Hemmer beschreibt. Außerdem wird für die Modellierung die sogenannte *log-cell-kill Hypothese* angewendet, die besagt, dass nach jedem Behandlungszyklus mit gleicher Dosis der gleiche Anteil an Zellen abgetötet wird, jedoch nicht die gleiche Anzahl [2]. Hieraus folgt für die Anti-Angiogenese, dass die Anzahl der abgetöteten Zellen bei der Therapie sowohl von der Dosis der Angiogenese-Hemmer u , als auch von der Größe des vaskulären Trägers q abhängt. Diese Überlegungen führen zu dem Term $-\gamma qu$, wobei γ die Effektivität der Therapie beschreibt. Wird die allgemeine Dynamik der vaskulären Trägerkapazität (2) um die Anti-Angiogenese erweitert, ergibt sich die Dynamik

$$\dot{q} = -\mu q + S(p, q) - I(p, q) - \gamma qu. \quad (4)$$

Verknüpft man diese Dynamik mit der Differentialgleichung für das Volumen der Krebszellen (1), erhält man das allgemeine Modell einer Krebsdynamik mit Anti-Angiogenese als Monotherapie.

Da die Angiogenese-Hemmer die Krebszellen nur indirekt abtöten, liefern sie auch nur einen temporären Effekt und der Tumor kann nach Einstellung der Behandlung sofort wieder wachsen. Deshalb soll

die Anti-Angiogenese nun zur Verbesserung der Behandlung mit der direkten Strahlentherapie kombiniert werden [5].

3 Modell der Radiotherapie

Bei der Strahlentherapie wird der Tumor lokal mit ionisierter Strahlung beschossen, die dann freie Radikale in den betroffenen Zellen aus Sauerstoffatomen bildet. Diese freien Radikale schädigen die DNA. Ist ein bestimmter Schaden angerichtet, kann sich die Zelle nicht mehr regenerieren und wird vom Körper abgebaut. Während ein Bruch eines einzelnen DNA-Strangs noch gut von der Zelle behoben werden kann, ist die Selbstheilung bei einem Bruch in beiden DNA-Strängen deutlich schwieriger. Liegen die Brüche zudem räumlich sehr nahe zusammen, kann die DNA irreparabel geschädigt werden und die Zelle stirbt infolgedessen ab. Für die Modellierung hat sich das sogenannte *linear quadratische Modell* etabliert. Es beschreibt die Wahrscheinlichkeit, dass eine Zelle bei der Bestrahlung mit der Gesamtdosis D überlebt, mit dem Ausdruck $\exp(-\alpha D - \beta D^2)$, wobei α und β die radiosensitive Parameter sind [8]. Diese hängen einerseits von der Wahrscheinlichkeit irreparabler Doppelbrüche an der DNA und andererseits von der Reparaturrate für die Strahlenschäden seitens der Zellen ab. Zur Vereinfachung wird angenommen, dass der Schaden an der DNA, der durch die freien Radikale entsteht, aus einem linearen und einem quadratischen Anteil besteht. Der lineare Anteil entspricht den irreparablen Brüchen in beiden Strängen, die durch ein einziges ionisiertes Partikel hervorgerufen wurden. Solch ein Partikel wird dann als letal oder todbringend in dem Sinne bezeichnet, dass die betroffenen Zellen danach keine weitere Zellteilung mehr durchführen können. Im Gegensatz dazu ist ein Bruch in einem einzelnen Strang noch nicht fatal, sondern kann von bestimmten Reparaturmechanismen der Zelle wieder hergestellt werden. Kommt es jedoch zu einem weiteren Bruch des anderen Stranges in der unmittelbaren Nähe des Ersten, bevor dieser vollständig repariert wurde, kann auch das zum Zelltod führen. Der quadratische Anteil beschreibt genau solche irreparablen Schäden, die durch die zweite Variante entstehen. Unter der zusätzlichen Annahme, dass die Anzahl der Einzel- und Doppelstrangbrüche einer Poisson-Verteilung folgen, ergibt sich dann eben genau die-

ser exponentielle Ausdruck für die Überlebenswahrscheinlichkeit der Zellen von $\exp(-\alpha D - \beta D^2)$. Der Nachteil dieser Modellierung ist, dass er nur für niedrige Strahlungsdosen anwendbar ist und für höhere Dosen weiter verallgemeinert werden muss.

In dieser Arbeit soll eine spezielle Variante des linear quadratischen Modells betrachtet werden, die auf Wein et al. [11] zurückgeht. Bei dieser Variante wurde zusätzlich berücksichtigt, dass bei den Schäden, die durch den quadratischen Ausdruck modelliert sind, zwischen den beiden Brüchen eine gewisse Zeitspanne liegen kann. Zwar sinkt dadurch die Wahrscheinlichkeit für eine letale Schädigung der Tumorzelle, dennoch ist sie möglich. Für die Modellierung wird die weitere Kontrolle w eingeführt, durch die man die Strahlendosis regulieren kann. Mathematisch gesehen ist w wieder eine Lebesgue-messbare Funktion, deren Werte im Intervall $[0, w_{\max}]$ liegen. Der Schaden, der durch die Bestrahlung am Tumor verursacht wird, wird durch

$$-p(t) \left(\alpha + \beta \int_0^t w(s) \exp(-\rho(t-s)) ds \right) w(t) \quad (5)$$

beschrieben, wobei die Parameter α, β und ρ positiv sind. Die lineare Komponente $-\alpha pw$ ist gleichbedeutend mit einem log-cell-kill Term und beschreibt den letalen Schadensanteil, der durch Doppelstrangbrüche von einzelnen Partikeln mit der zugehörigen Wahrscheinlichkeit α erzeugt wird. Mit der Konstante β wird die Wahrscheinlichkeit für den hier komplexeren quadratischen Anteil beschrieben, dass zwei Einzelbrüche an einer DNA-Doppelhelix eine letale Schädigung bewirken. Der zusätzliche Parameter ρ steht für die Reparaturrate der Krebszellen an ihrer DNA. Die Wahrscheinlichkeit, dass ein Bruch zur Zeit s und ein weiterer zur Zeit t für die Zelle todbringend sind, nimmt exponentiell mit dem zeitlichen Abstand $t-s$ und der Reparaturrate ρ ab, die die Effizienz dieses Vorgangs widerspiegelt. Je größer der zeitliche Abstand zwischen beiden Brüchen ist, umso länger hat die Zelle Zeit, ihre DNA zu reparieren. Wurde der erste Bruch bereits behoben, bevor der zweite stattfindet, kann die Krebszelle mit einem Einfachbruch nicht todbringend geschädigt werden. Die hier wieder auftretenden Parameter α und β werden in der medizinischen Literatur auch als *LQ Parameter* bezeichnet. Der komplexe Ausdruck für den Strahlenschaden

kann zusätzlich vereinfacht werden, da

$$r = \int_0^t w(s) \exp(-\rho(t-s)) ds \quad (6)$$

genau die Lösung der linearen Differentialgleichung erster Ordnung

$$\dot{r} = -\rho r + w, \quad r(0) = 0, \quad (7)$$

ist. Mit der neuen Variable r lässt sich der Strahlenschaden am Tumor folgendermaßen formulieren:

$$-p(t)(\alpha + \beta r(t))w(t). \quad (8)$$

Trotzdem gibt es bei dieser Beschreibung immer noch keinen ersichtlichen quadratischen Anteil, sondern nur $\beta p(t)r(t)w(t)$. Wird jedoch eine konstante Strahlendosis \bar{w} angenommen, so hat die Differentialgleichung (7) ein Equilibrium bei $\bar{r} = \frac{\bar{w}}{\rho}$. Setzt man dieses Gleichgewicht in (8) ein, dann entsteht der allgemeine linear quadratische Ausdruck

$$-p \left(\alpha \bar{w} + \frac{\beta}{\rho} \bar{w}^2 \right). \quad (9)$$

Eine kleine Reparaturrate des Tumors verstärkt offensichtlich den Einfluss des Integralterms bei der Abnahme des Tumolvolumens p und erhöht somit die Effektivität der Behandlung. Im Gegensatz dazu ist der Einfluss bei großen Reparaturraten deutlich geringer, und das Integral kann durch seinen Gleichgewichtswert ersetzt werden, wie in (9) [5].

Bei einer Bestrahlung werden leider nicht nur die Krebszellen getroffen, sondern auch die umliegenden Blutgefäße und gesunden Zellen. Deshalb muss bei der Modellierung zudem der Strahlenschaden an der vaskulären Trägerkapazität berücksichtigt werden. Dies geschieht analog zum Strahlenschaden des Tumorgewebes durch

$$-q(t)(\eta + \delta r(t))w(t), \quad (10)$$

wobei die Parameter η und δ die gleichen Bedeutungen haben wie α und β bei der Schädigung des Tumors, sich aber in ihren Werten deutlich voneinander unterscheiden können. Weil sich die Reparaturrate von Krebszellen, Endothelzellen und gesunden Gewebe aufgrund ihrer speziellen Eigenschaften im Allgemeinen unterscheiden, muss zusätzlich die Variable r für jede der drei Arten getrennt voneinander in der zugehörigen Differentialgleichung

$$\dot{r}_i = -\rho_i r_i + w, \quad r_i(0) = 0, \quad (11)$$

beschrieben werden, wobei der Index p für die Tumorzellen, q für den vaskulären Träger, und z für das gesunde Gewebe steht. Außerdem beschreiben die Parameter ρ_p, ρ_q und ρ_z die gewebespezifischen Reparaturraten des Tumors, des Gefäßsystems und des gesunden Gewebes. Bringt man jetzt alle bisherigen Ergebnisse zum Tumorwachstum und zu den Therapieformen Anti-Angiogenese und Strahlenbehandlung zusammen, ergibt sich ein Modell für die Kombinationstherapie aus eben diesen beiden Therapieformen.

Das allgemeine Anti-Angiogenese-Radiotherapie-Modell ist gegeben durch

$$\begin{aligned}\dot{p} &= -\xi p \ln\left(\frac{p}{q}\right) - (\alpha + \beta r_p) p w, \\ \dot{q} &= -\mu q + S(p, q) - I(p, q) - \gamma q u - (\eta + \delta r_q) q w, \\ \dot{r}_p &= -\rho_p r_p + w, \quad \dot{r}_q = -\rho_q r_q + w, \quad \dot{r}_z = -\rho_z r_z + w\end{aligned}\quad (12)$$

und $p(0) = p_0, q(0) = q_0, r_p(0) = r_q(0) = r_z(0) = 0$, wobei die positiven Anfangswerte p_0, q_0 und die positiven Parameter $\alpha, \beta, \gamma, \delta, \eta, \rho_p, \rho_q, \rho_z$ vorgegeben sind. Die Funktionen $S(p, q)$ und $I(p, q)$ können, wie bereits beschrieben, gewählt werden. Die Steuerungen u und w sind Lebesgue-messbar und nehmen Werte in $[0, u_{\max}]$ bzw. in $[0, w_{\max}]$ an.

Auf den ersten Blick kann die Differentialgleichung für r_z unnötig erscheinen, da sie nicht in die Dynamiken des Tumorumfanges und des vaskulären Trägers eingeht. Jedoch wird sie wichtig, wenn das Modell zu einem optimalen Steuerprozess erweitert wird, wofür die Schädigung des umliegenden gesunden Gewebes mit einer oberen Schranke versehen werden muss. Diese darf auf keinen Fall überschritten werden, um die Nebenwirkungen möglichst gering zu halten und nicht mehr zusätzlichen Schaden anzurichten, als zu beheben [5].

4 Kombinationsmodell

Wir untersuchen jetzt eine Spezialisierung dieses allgemeinen Modells von Hahnfeldt (12) nach Ledzewicz/Schättler [5, 9], indem $\rho_p = \rho_q \stackrel{\text{def}}{=} \rho$ gewählt wird. In diesem Modell gilt dann auch $r_p = r_q \stackrel{\text{def}}{=} r$. Wir limitieren nun die kummulierte Gesamtdosis des Antiangiogenese-Medikaments $\int_0^T u(t) dt \leq y_{\max}$ und die kummulierte Schädigung des gesunden Gewebes $\int_0^T (1 + \theta r_z) w dt \leq z_{\max}$. Diese Nicht-Standard-

Nebenbedingungen lassen sich wie folgt in ein standardmäßiges optimales Steuerungsproblem umformen:

Für eine freie Endzeit T , minimiere

$$p(T)$$

unter Berücksichtigung der Dynamik des Prozesses

$$\begin{aligned}\dot{p} &= -\xi p \ln\left(\frac{p}{q}\right) - (\alpha + \beta r) p w, \\ \dot{q} &= b p - \left(\mu + d p^{\frac{2}{3}}\right) q - \gamma q u - (\eta + \delta r) q w, \\ \dot{r} &= -\rho r + w, \\ \dot{r}_z &= -\rho_z r_z + w, \\ \dot{y} &= u, \\ \dot{z} &= (1 + \theta r_z) w,\end{aligned}\quad (13)$$

den Anfangsbedingungen

$$p(0) = p_0, q(0) = q_0, r(0) = r_z(0) = y(0) = z(0) = 0,$$

den Endbedingungen

$$y(T) \leq y_{\max}, \quad z(T) \leq z_{\max}$$

und des Steuerbereiches

$$u(t) \in [0, u_{\max}], \quad w(t) \in [0, w_{\max}].$$

Die benutzten Parameter sind [9] entnommen, mit Ausnahme von ξ , dessen Wert [5] entnommen ist. Weitere detailliertere Informationen zu diesem Modell sowie vielen weiteren Modellen und ein sehr ausführliches Literaturverzeichnis finden sich in [9].

Mithilfe des Minimumprinzips nach Pontryagin lassen sich erste notwendige Bedingungen an eine optimale Lösung des Problems formulieren, siehe [5, 9]. Jedoch ist eine Synthese, d.h. insbesondere die optimale Lösung noch unbekannt.

5 Optimale Lösung

Das Hauptaugenmerk dieser Arbeit gilt der numerischen Bestimmung von optimalen Therapieverläufen.

Dazu wird als erstes ein Zeitgitter mit der Gitterfeinheit N , hier $N = 1000$, eingeführt. Über diesem Zeitgitter werden die beiden Steuerungen als Treppenfunktionen approximiert. Zudem werden die Differentialgleichungen über dem Zeitgitter mit dem impliziten Eulerverfahren diskretisiert. Mit dieser Transkription entsteht ein nichtlineares Optimierungsproblem

[1]. Mittels der Modellierungssprache AMPL [3] wird das nichtlineare Optimierungsproblem an einen Optimierungsalgorithmus, z.B. IPOPT [10], übergeben. Der Vorteil der Benutzung von AMPL ist, das AMPL erste und zweite Ableitungen der Nebenbedingungen und der Zielfunktion automatisch an den Optimierungsalgorithmus übergibt.

Die Ausgangslage bildet das Tumervolumen $p(0) = 8000$ und das Volumen des vaskulären Trägers $q(0) = 10000$.

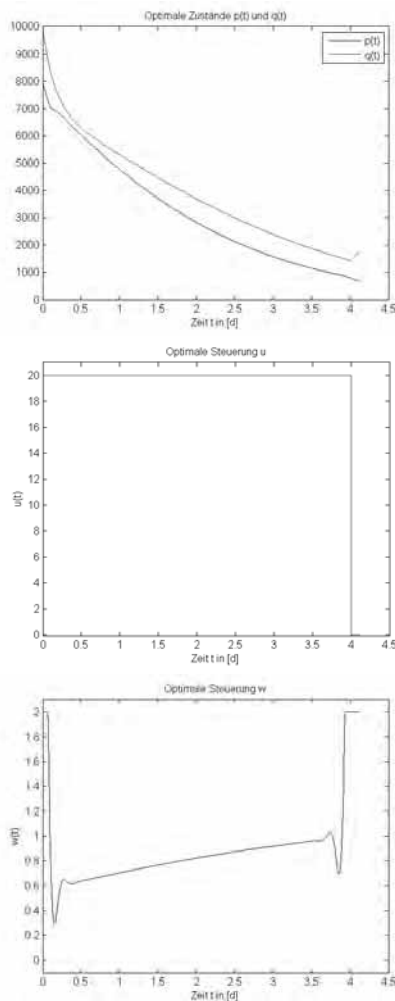


Abbildung 1: Optimale Lösung mit den Obergrenzen für die Anti-Angiogenese $u_{\max} = 20$, $y_{\max} = 80$ und für die Bestrahlung $w_{\max} = 2$, $z_{\max} = 4$.

Das erste Zahlenbeispiel soll die gerade beschriebe-

ne Ausgangslage mit den festgelegten Obergrenzen $u_{\max} = 20$ für die Dosis der Angiogenese-Hemmer und $y_{\max} = 80$ für die maximale Gesamtmenge an Hemmern seitens der Anti-Angiogenese verbessern. Für die zweite Therapieform der Bestrahlung ist die Maximaldosis mit $w_{\max} = 2$ festgelegt und als Schaden am gesunden Gewebe wird maximal $z_{\max} = 4$ geduldet.

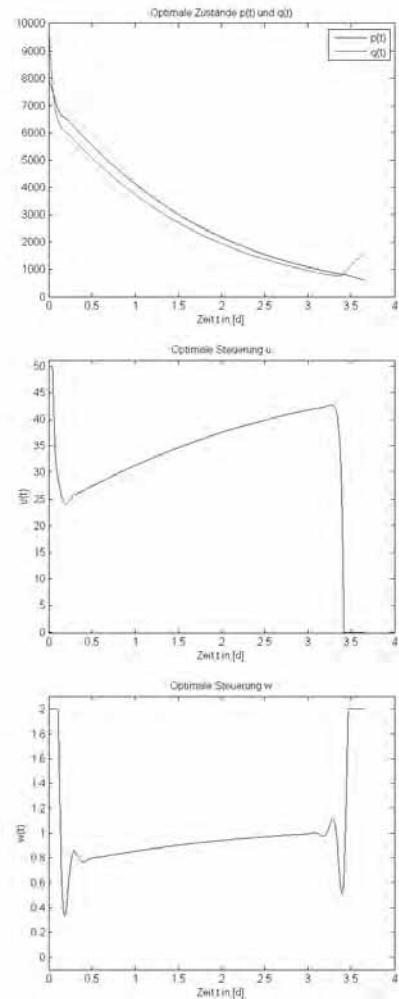


Abbildung 2: Optimale Lösung mit den Obergrenzen für die Anti-Angiogenese $u_{\max} = 50$, $y_{\max} = 120$ und für die Bestrahlung $w_{\max} = 2$, $z_{\max} = 4$.

Die von IPOPT berechnete Lösung (siehe Abbildung 1) erreicht zur Endzeit $T = 4.12$ und unter Einsatz aller vorhandenen Angiogenese-Hemmer und ex-

akter Einhaltung der Obergrenze für den Strahlenschaden am gesunden Gewebe das minimale Tumorumfang von $p(T) = 680.039$. Das Volumen des vaskulären Trägers zur Endzeit ist mit $q(T) = 1745.55$ deutlich höher als das Tumorumfang, weshalb der Tumor nach Beendigung der Therapie sofort wieder wächst. Die optimale Steuerung der Angiogenese-Hemmer sieht eine Verabreichung der Maximaldosis $u_{\max} = 20$ über vier Tage vor. Danach ist die vorhandene Gesamtmenge aufgebraucht, und es können demzufolge keine weiteren Hemmer bis zum Therapieende gegeben werden. Die Steuerung der Bestrahlung sieht nach einer sehr kurzen anfänglichen Maximaldosis bis $t_1 = 0.07$ eine singuläre Phase vor, die bis $t_2 = 3.94$ andauert. In dieser steigt die Dosis kontinuierlich von $w = 0.62$ auf $w = 1.04$ an¹. Abschließend wird noch einmal die Maximaldosis verwendet. Bei der Betrachtung der p - q -Dynamik fällt auf, dass das Volumen des vaskulären Trägers über die ganze Therapie hinweg größer ist als das Tumorumfang. Folglich bewirkt die Anti-Angiogenese für die hier gewählten Obergrenzen und Ausgangswerte nicht direkt eine Reduktion des Tumorumfangs. Trotzdem ist die Anti-Angiogenese nicht nutzlos, da durch sie der vaskuläre Träger reduziert wird und folglich das Eigenwachstum des Tumors gemäß der ersten Differentialgleichung in (13) relativ klein gehalten wird. Dadurch kann die Bestrahlung insgesamt ein besseres Ergebnis mit gleichzeitiger Anti-Angiogenese erzielen als ohne.

Wegen der nur indirekt wirkenden Anti-Angiogenese sollen für das zweite Zahlenbeispiel die zugehörigen Obergrenzen auf die Werte $u_{\max} = 50$ und $y_{\max} = 120$ angehoben werden, während die Beschränkungen für die Strahlentherapie unverändert bleiben (siehe Abbildung 2).

Durch die Erhöhung der Obergrenzen wurde die optimale Therapiedauer auf $T = 3.646$ reduziert. Zu diesem Zeitpunkt betragen das minimale Tumorumfang $p(T) = 615.161$ und das vaskuläre Trägervolumen $q(T) = 1535.8$. Im Vergleich zum ersten Beispiel bedeutet dies trotz deutlicher Anhebung der Obergrenzen für die Anti-Angiogenese keine wesentliche Verbesserung des Resultats. Jedoch ist nun in der p - q -Dynamik zu erkennen, dass das

vaskuläre Trägervolumen über einen längeren Zeitraum kleiner ist als das Tumorumfang und die Anti-Angiogenese somit direkt an der Volumenreduktion des Tumors beteiligt ist. Außerdem zeigt sich, dass der vaskuläre Träger anfänglich am stärksten abnimmt, bis zum Zeitpunkt $t_1 = 0.05$, an dem die Steuerung $u(t)$ singulär wird. Dies hängt mit dem stark differential-algebraischen Charakter der Differentialgleichung für die q -Dynamik (13) zusammen, weshalb die q -Dynamik schnellstmöglich ihre Gleichgewichtslösung anstrebt. Während die optimale Steuerung der Bestrahlung w kaum einen Unterschied zum vorherigen Beispiel aufweist, ist er für die optimale Steuerung u deutlich. Nach einer anfänglichen Verabreichung der vollen Dosis über den sehr kurzen Zeitraum bis $t_1 = 0.05$, folgt eine lange singuläre Dosierung bis zum Zeitpunkt $t_2 = 3.29$, an dem die festgelegte Gesamtmenge aufgebraucht ist.

Trotz der Anhebung der Obergrenzen für die Angiogenese-Hemmer und Inkaufnahme stärkerer Nebenwirkungen war es im zweiten Beispiel nicht möglich das Tumorumfang noch einmal deutlich zu reduzieren. Deshalb soll nun ausgehend vom ersten Beispiel die Obergrenze für die Anti-Angiogenese unverändert bleiben und durch eine Erhöhung der Obergrenze für den Strahlenschaden am gesunden Gewebe auf $z_{\max} = 6$ ein besseres Resultat erzielt werden. Die von IPOPT bestimmte optimale Therapie (siehe Abbildung 3) erreicht nach einer Dauer von $T = 4.426$ das minimale Tumorumfang $p(T) = 226.866$, während das Volumen des vaskulären Trägers $q(T) = 939.551$ beträgt. Folglich konnte durch die Anhebung der Obergrenze z_{\max} der Tumor im Vergleich zum ersten Beispiel um weitere 64% reduziert werden. Das legt die Vermutung nahe, dass die Bestrahlung bei der hier vorliegenden Modellierung deutlich mehr zur Reduktion des Tumors beiträgt als die Anti-Angiogenese, zumindest bei den hier gewählten Obergrenzen und Anfangswerten. Genau wie im ersten Beispiel lässt sich in der p - q -Dynamik ablesen, dass $q(t)$ immer größer als $p(t)$ ist und somit die Anti-Angiogenese lediglich indirekt zur Reduktion beiträgt. Für die optimale Steuerung der Bestrahlung $w(t)$ bewirkt die Anhebung von z_{\max} keine Veränderung an der Struktur selbst, sondern nur an den Werten $w(t)$, die hier in der singulären Phase größer sind als beim ersten Zahlenbeispiel. Die optimale Steuerung subfigure der Angiogenese-Hemmer $u(t)$ beginnt mit einer singulären Phase bis $t_1 = 0.72$, auf die an-

¹Die Überschwingungen der Steuerungen w am Anfang und am Ende der singulären Phase wurden hierbei nicht beachtet, da sie durch die Diskretisierung des Problems entstehen.

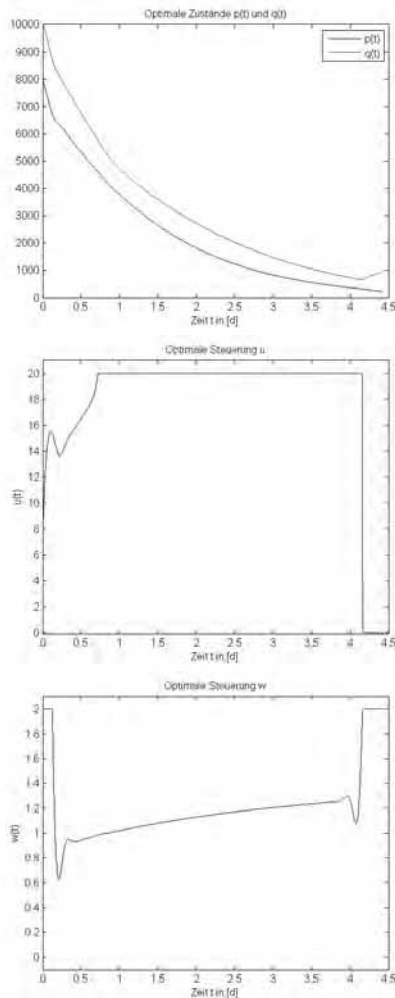


Abbildung 3: Optimale Lösung mit den Obergrenzen für die Anti-Angiogenese $u_{\max} = 20$, $y_{\max} = 80$ und für die Bestrahlung $w_{\max} = 2$, $z_{\max} = 6$.

schließlich die volle Dosierung bis zum vollständigen Verbrauch der Medikamente folgt, was ungefähr bei $t_2 = 4.16$ geschieht. Danach wird der Tumor bis zum Therapieende nur noch durch die Bestrahlung reduziert.

6 Suboptimale Lösungen

Die optimalen Therapieverläufe sind aufgrund ihrer Struktur, insbesondere wegen den singulären Phasen,

für die praktische medizinische Umsetzung nicht geeignet. Deshalb soll in diesem Abschnitt für jedes der drei Zahlenbeispiele eine suboptimale Therapie mit stückweise konstanten Steuerungen bestimmt werden. Dafür wird ab Therapiebeginn, also $t = 0$, für jede der beiden Steuerungen solange eine konstante Dosis verabreicht, bis die zugehörige Obergrenze y_{\max} beziehungsweise z_{\max} erreicht wurde. Diese konstanten Dosen gilt es neben der freien Endzeit so zu bestimmen, dass das Tumorumfang zum Ende der Therapie minimal wird. Folglich muss ein dreidimensionales Optimierungsproblem gelöst werden, was mithilfe von MATLAB geschah.

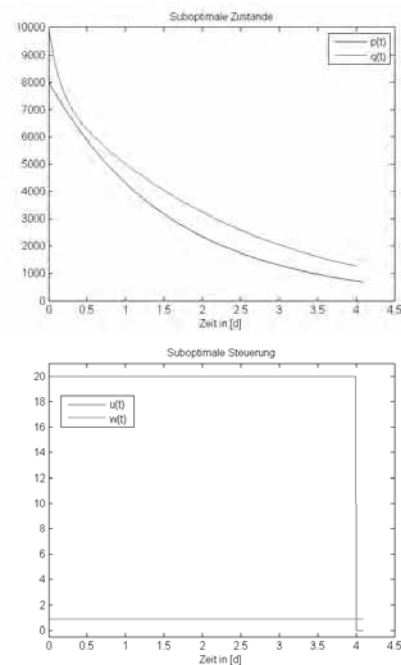


Abbildung 4: Suboptimale Lösung für das erste Beispiel.

Für die gleiche Ausgangslage wie beim ersten Beispiel aus dem vorherigen Abschnitt lässt sich durch dieses Vorgehen das minimale Tumorumfang $p(T) = 689.882$ nach einer Therapiedauer von $T = 4.094$ erreichen (siehe Abbildung 4). Für dieses Resultat muss über vier Tage hinweg die volle Dosis $u_{\max} = 20$ an Angiogenese-Hemmern verabreicht und während der gesamten Therapie mit der konstanten Dosis $w = 0.8889$ bestrahlt werden. Das entspricht dem glei-

chen Vorgehen für die Anti-Angiogenese wie beim zuvor bestimmten optimalen Therapieverlauf. Verglichen mit den $p^*(T) = 680.039$ für die durch IPOPT bestimmte optimale Lösung liegt die Abweichung des suboptimalen Resultats lediglich bei 1.5%, was für die getroffenen Einschränkungen an die Steuerungen ein sehr gutes Ergebnis darstellt.

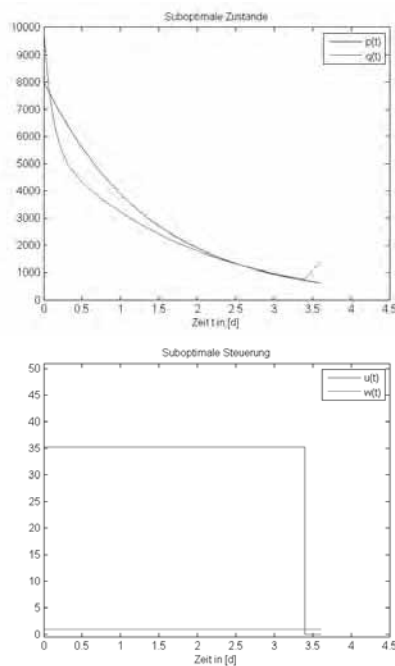


Abbildung 5: Suboptimale Lösung für das zweite Beispiel.

Für das zweite Zahlenbeispiel kann der Tumor durch optimales Vorgehen nach $T = 3.6036$ auf $p(T) = 622.854$ reduziert werden. Die Abweichung zum optimalen Resultat $p^*(T) = 615.161$ ist mit knapp über einem Prozent wieder gut vertretbar. Um dieses Ergebnis zu erhalten muss über 3.4 Tage die Dosis $u = 35.29$ an Angiogenese-Hemmern verabreicht und über die ganze Therapiedauer mit $w = 1$ bestrahlt werden. Auch bei der hier bestimmten suboptimalen Lösung fällt im Vergleich zum ersten Beispiel in der p - q -Dynamik das Verhältnis $p(t) < q(t)$ auf, das über einen gewissen Zeitraum hinweg besteht. Bereits bei den optimalen Lösungen trat diese Relation nur nach Erhöhung der Obergrenzen für die Anti-Angiogenese auf.

Literatur

- [1] J.T. Betts: *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM, Philadelphia, 2001.
- [2] A. Ergun, K. Camphausen, L. Wein: *Optimal Scheduling of Radiotherapy and Angiogenic Inhibitors*. Bulletin of Mathematical Biology, 65:407-424, 2003.
- [3] R. Fourer, D.M. Gay, B.W. Kernighan: *AMPL: A Modelling Language for Mathematical Programming*. Duxbury, 2002².
- [4] P. Hahnfeldt, D. Panigrahy, J. Folkman, L. Hlatky: *Tumor Development under Angiogenic Signaling: A Dynamical Theory of Tumor Growth, Treatment Response, and Postvascular Dormancy*. Cancer Research, 59:4770-4775, 1999.
- [5] U. Ledzewicz, H. Schättler: *Multi-Input Optimal Control Problems for Combined Tumor Anti-Angiogenic and Radiotherapy Treatments*. Journal of Optimization Theory and Applications, 153:195-224, 2012.
- [6] U. Ledzewicz, H. Schättler: *Anti-angiogenic therapy in cancer treatment as an optimal control problem*. SIAM J. on Control and Optimization, 46(3):1052-1079, 2007.
- [7] R. Prehn: *Two competing influences that may explain concomitant tumor resistance*. Cancer Research, 53:3266-3269, July 1993.
- [8] M. Reiser, F. Kuhn, J. Debus: *Duale Reihe Radiologie*. Thieme Verlag, 2011³.
- [9] H. Schättler, U. Ledzewicz: *Optimal Control for Mathematical Models of Cancer Therapies: An Application of Geometric Methods*. Springer, New York, 2015.
- [10] A. Wächter: *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. PhD Thesis, Carnegie Mellon University Pittsburgh, Pennsylvania, 2002.
- [11] L. Wein, J. Cohen, J. Wu: *Dynamic optimization of a linear-quadratic model with incomplete repair and volume-dependent sensitivity and repopulation*. Int. J. Radiation Oncology Biol. Phys., 47(4):1073-1083, 2000.

Simulation von Stützprozessen und Logistik in der Klinikplanung

Dr.-Ing. Carsten Matysczok¹, Gime Batija¹, Markus Kühn¹, Moritz Schele¹

¹UNITY AG

carsten.matysczok@unity.de

Bei der Prozess- und Gebäudeplanung in Krankenhäusern spielen die Logistik- und Stützprozesse eine immer größere Rolle. Kostendruck, Fachkräftemangel und medizinische sowie technische Entwicklungen stellen die bestehenden Strukturen vor Herausforderungen, denen nur mit modernem Planungsvorgehen adäquat begegnet werden kann. Mit der digitalen Simulation von Logistik- und Stützprozessen weitet die UNITY AG den erfolgreichen Einsatz dieser Technologie im Klinikumfeld aus und legt einen weiteren Grundstein zur vollständig digitalen Planung in dieser Branche. Innovative Abläufe, quantifizierte Raum- und Zeitbedarfe sowie abgesicherte Entwicklungsmöglichkeiten stellen einen Mehrwert für die Klinikbetreiber dar und belegen den Nutzen dieser Technologie.

Einleitung

Seit mehreren Jahren befindet sich das deutsche Gesundheitssystem in der öffentlichen Kritik. Die Fortschritte in der Medizin und -technik führen zu immer höheren Kosten. Parallel dazu stellen die Alterung der Gesellschaft und der Rückgang der Bevölkerungszahlen die Finanzierung der sozialen Sicherungssysteme vor große Herausforderungen. Die Kliniken sind nun gefordert, ihre immer aufwändigeren Leistungen bei gleicher Qualität wirtschaftlich zu erbringen. Der Druck zum Handeln ist hoch: die steigenden Kosten und immer komplexeren Organisationsabläufe stellen viele Kliniken vor große Herausforderungen [1].

Etablierte Methoden und der zielorientierte Einsatz der Ablaufsimulation, die seit vielen Jahren erfolgreich in der Fertigungsindustrie eingesetzt werden, können dazu beitragen, diese Herausforderungen zu meistern [2, 3, 4]. Durch sie ist eine vorausschauende und durchgängige Ablaufplanung innerhalb der Klinik sowie in den vor- und nachgelagerten Prozessketten möglich. Ziel der Simulation der Stützprozesse (Prozesse zur Unterstützung der klinischen Kernprozesse) und der Logistik in der Klinik ist die Beantwortung unterschiedlicher Fragestellungen: Wie wird eine effiziente Logistik innerhalb eines Kliniknetzwerkes abgebildet? Wie ist eine Wäscherei, eine Sterilgutaufbereitung, ein Zentrallager oder eine Zentralküche zu dimensionieren, um den Klinikverbund optimal versorgen zu können? Reichen Fahrstuhl- und Intralogistikkapazitäten aus, um bei steigender Fallzahl eine Versorgung der Klinikbereiche mit Medikamenten, Essen, sauberer Wäsche und Operationsinstrumenten sicherzustellen?

In diesem Beitrag sind die Herausforderungen der Stützprozesse und Logistikabläufe im Krankenhaus dargestellt, Lösungsansätze der Simulationstechnik aufgeführt und deren Einsatz anhand von Praxisbeispielen dargestellt.

Simulation Stützprozesse und Logistik in der Klinik

Herausforderungen bei der Planung von Stützprozessen und Logistikabläufen in Krankenhäusern

Das klassische Planungsvorgehen in Krankenhäusern ist geprägt von einer Outside-In Herangehensweise. Da Krankenhausgebäude häufig in der öffentlichen Wahrnehmung stehen, spielt bei den Entscheidungsprozessen die Anmutung häufig eine wichtigere Rolle, als die Anforderungserfüllung zum reibungsfreien Betrieb. Insbesondere bei den logistischen Abläufen wirkt sich dies aber negativ auf die Effizienz aus und schafft neben Mehraufwänden auch Unzufriedenheit bei Personal und letztlich auch Patienten [5]. Der Druck zum Handeln ist groß, schließlich stellen die Personalkosten den größten Anteil während der Betriebsphase dar [6] und effiziente Prozesse bilden die Basis für einen wirtschaftlichen Krankenhausbetrieb. Die durch die Logistik verursachten Kosten tragen dabei maßgeblich zur Wirtschaftlichkeit eines Krankenhauses bei und weisen großes Einsparpotenzial auf [7].

Bezogen auf die Logistikabläufe und Stützprozesse zeichnen sich zudem mehrere Trends und Entwicklungen ab, die den Betrieb von Krankenhäusern in Zukunft beeinflussen werden. Eine dieser Entwicklungen hat ihren Ursprung im Fachkräftemangel für

Pflegeberufe. Immer mehr Häuser haben bereits heute Schwierigkeiten dabei, die offenen Stellen adäquat zu besetzen. Ein wirksames Gegenmittel ist die Fokussierung auf die Kernkompetenzen „Pflegen und Heilen“, was im Falle der Stützprozesse eine zunehmende Zentralisierung und intensivere Kooperation mit Dienstleistern bedeutet. Sind solche Bereiche bereits extern betrieben (regelmäßig bei Küche und Wäscherei der Fall), werden die Kooperationen nun häufig auch bis zur Speiseversorgung auf Station oder die Aufbereitung der Betten ausgeweitet. Solche Modelle sind häufig neu und den Krankenhäusern fehlen Erfahrungswerte zur Planung der Prozesse und Gebäude.

Zudem zeichnet sich in vielen Bereichen eine stärkere Kooperationsbereitschaft mit anderen Häusern ab, beispielsweise bei der Sterilgutaufbereitung oder der zentralen Material- und Logistikverwaltung. Speziell hier sind enorm hohe Ansprüche an Qualität, Zuverlässigkeit und Hygiene gestellt, was nur durch optimale Prozesse und in unterstützenden Strukturen mit ausreichend Logistikfläche und entsprechenden Notfallkonzepten erfüllt werden kann.

Dazu kommt, dass IT- und medizintechnische Entwicklungen den Krankenhausbetrieb verändern werden. Die Durchgängigkeit und Verfügbarkeit von Informationen eröffnet neue Möglichkeiten, beispielsweise bei der Materialbereitstellung für den OP (Stichwort Fallwagen). Zudem wird, ebenfalls im OP-Bereich besonders relevant, zunehmend Einwegmaterial eingesetzt und dieses häufig OP-spezifisch vorkommissioniert. Dies hat einen erheblichen Einfluss auf die Materialflüsse zwischen Zentrallager und OP-Bereich und stellt bestehende Strukturen und Lagergrößen zunehmend vor Herausforderungen. Kommen dann noch Aspekte zum Tragen, wie immer strengere gesetzliche Vorgaben zur Rückverfolgbarkeit der Materialien, sind etablierte Prozesse und Strukturen häufig überfordert.

All das führt dazu, dass die Stützprozesse und Logistikabläufe in Krankenhäusern unter neuen Prämissen neu geplant werden müssen. Bei dieser neuen Denkweise bietet sich die Simulationstechnik als wirksames Mittel an, um unbekannte und schwer zu greifende Anforderungen zu bewerten und zu quantifizieren - was zur erfolgreichen Bau- und Betriebsplanung unumgänglich ist.

Simulation als Werkzeug zur Begegnung der Herausforderungen

Die Simulationstechnik hat sich zu einem wirksamen Beratungswerkzeug für die digitale Klinikplanung bei der UNITY entwickelt. Zu Beginn wurde der OP-Bereich mit Hilfe der OP-Simulation geplant und optimiert. Wegen des großen Erfolgs wurden zunehmend auch weitere Klinikbereiche, ihre Stützprozesse und die Logistik mit Hilfe der Simulationstechnik bewertet und festgelegt. So lässt sich ein Zukunftsmodell der Klinik erstellen, um die zuvor skizzierten Fragestellungen in unterschiedlichen Szenarien zu bewerten und frühzeitig die richtigen Entscheidungen zu treffen. Sie leistet dabei einen entscheidenden Beitrag, die Planungszeiten von Klinikneubauten oder Erweiterungen zu verkürzen und die bestmögliche Alternative für die Realisierung auszuwählen. Dieses belegt eine Vielzahl von erfolgreich durchgeführten Projekten, wovon eine Auswahl im Kapitel „Projektbeispiele“ angeführt ist.

Bereits auf der ASIM Konferenz im März 2016 in Lippstadt wurden in einem Beitrag zur „End-to-End Kliniksimulation“ die speziellen Herausforderungen und die eingesetzte Simulationstechnik dargestellt [8]. An dieser Stelle wird daher nur verkürzt und in Auszügen auf den Aufbau der Simulationsmodelle eingegangen.

Für die Simulation wird die Software Plant Simulation von Siemens PLM genutzt und durch eine Reihe von speziell entwickelten Klassenbibliotheken erweitert. Diese Klassenbibliotheken werden dabei stetig an die individuellen Anforderungen aus den Planungsprojekten angepasst und weiterentwickelt. Heute kann der Programmierer bereits auf eine Fülle an Standardelementen zurückgreifen, die übergeordnete Funktionsbausteine darstellen und durch ihre Kombination schließlich ganze Funktionsbereiche bilden können. Beispielsweise werden Standardelemente für die Abbildung von Einleitungsräumen, OP-Sälen und Aufwachräume genutzt, die in ihrer Verbindung einen ganzen OP-Bereich repräsentieren können.

Im Aufbau lässt sich das Simulationsmodell in fünf Ebenen unterscheiden (siehe Abbildung 1). Jede Ebene stellt dabei eine Abstraktions- oder Detailebene dar: Die oberste Ebene bildet die makroskopische Sichtweise auf ein Netzwerk aus mehreren Kliniken (Ebene 1) und kann somit beliebig viele Klinikstandorte enthalten. Um die wechselseitigen Beziehungen zwischen den Standorten darzustellen, werden auf

Ebene 1 meist auch die logistischen Prozesse abgebildet.

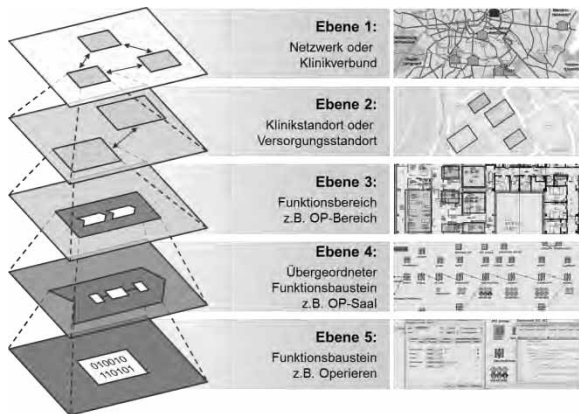


Abbildung 1. Die fünf Ebenen der End-to-End-Kliniksimulation [8]

Auf der Ebene 2 werden einzelne Klinik- oder Versorgungsstandorte mit verschiedenen Gebäuden und Einrichtungen simuliert. Zusätzlich werden wechselseitige Beziehungen abgebildet. Es können beispielsweise Patienten aus der Operationsspanne in die Bettenstation transportiert werden, welche dann auch in der Simulation das Gebäude wechseln.

Ein Standort setzt sich dazu aus mehreren Funktionsbereichen wie der Notaufnahme, den Dienstbereichen, einem oder mehreren OP-Bereichen, der Elektivaufnahme, der Bettenstation und falls am Standort enthalten auch Sterilgutaufbereitung, Wäscherei und Küche zusammen (Ebene 3). Auf der dritten Ebene werden dazu die einzelnen Funktionsbereiche modelliert und simuliert. Oft wird den Modellen dazu auch ein Grundriss (in Form eines 2D-CAD-Modells) hinterlegt, um eine realistische räumliche Abbildung des Bereiches mit den dazugehörigen Strecken (Transport und Patientenfluss) zu simulieren und das Raumkonzept zu validieren.

Auf Ebene 4 befinden sich die übergeordneten Komponenten, aus denen die Funktionsbereiche aufgebaut sind. Diese können den eigens entwickelten Klassenbibliotheken von UNITY entnommen oder individuell für die jeweilige Fragestellung entwickelt worden sein. Werden wesentliche Bausteine neu erstellt, finden diese wiederum Eingang in die Klassenbibliotheken. Somit wächst der ad-hoc verfügbare Funktionsumfang der Funktionsbereiche mit jedem Projekt.

Auf der untersten Ebene 5 sind schließlich die Funktionsbausteine, ihre Steuerungsmethoden, Tabellen und weitere Elemente der Simulationssoftware zu

finden. Diese Bausteine bilden einzelne Detail-Funktionen des Modells ab und halten die Prozessparameter zur Steuerung der Abläufe.

Zusätzlich können die Eigenschaften der Standardbausteine aufgrund der konsequenten Vererbung der Klasseneigenschaften und der über Excel-Tabellen steuerbaren variablen Parametern auf den speziellen Anwendungsfall eingestellt werden. Diese generisch aufgebauten Modelle lassen sich dadurch prinzipiell auf jede Klinik- oder Netzwerkgröße skalieren und um jede relevante Funktion im Kliniknetzwerk erweitern. Nach der erstmaligen Modellbildung können die Modelle dann flexibel auf sich ändernde Planungsprämissen eingestellt und verschiedene Simulationsexperimente durchgeführt werden.

Bezogen auf die Stützprozesse und Logistikabläufe sind alle fünf Ebenen relevant. Während Stützprozesse selbst durch Komponenten der Ebenen 3-5 abgebildet sind und beispielsweise die Bettenaufbereitung oder die Sterilgutaufbereitung darstellen, sind die Materialfluss- und Logistikabläufe innerhalb eines Standorts auf den Ebenen 2 und 3 abgebildet und die Transporte zwischen den Standorten auf Ebene 1. Die Ebenen 1-3 stellen dabei auch die Verbindung zwischen Stütz- und Hauptprozessen dar. Wird beispielsweise Material aus der Sterilgutaufbereitung zur Operation benötigt, so wird dieses auf Ebene 3 des Simulationsmodells zur Verfügung gestellt, nachdem der Transport gegebenenfalls auf den Ebenen 1 und 2 abgebildet wurde.

Projektbeispiele

Simulation von Stützprozessen und der zugehörigen Versorgungslogistik

Ein Beispiel für die Simulation von Stützprozessen und die Lieferung der Hilfsgüter stellt die Planung und Auslegung von zwei zentralen Standorten zur Sterilgutaufbereitung für zehn Klinikstandorte im urbanen Großraum dar.

Um Kosten zu reduzieren, werden die in den Operationen genutzten Instrumente gesammelt, verpackt, zentral an zwei Standorten aufbereitet und in den Kreislauf rückgeführt. Ziel des Planungsprojektes war die Auslegung beider Aufbereitungsstandorte sowie der Logistikkapazitäten für den Transport von verunreinigten und aufbereiteten Sterilgütern mit maximaler Versorgungssicherheit der Kliniken.

Das Simulationsmodell wurde dazu in den beschriebenen Ebenen aufgebaut. Auf der obersten Ebene (Ebene 1) werden die verschiedenen Klinik- und Aufbereitungsstandorte dargestellt (siehe Abbildung 2).

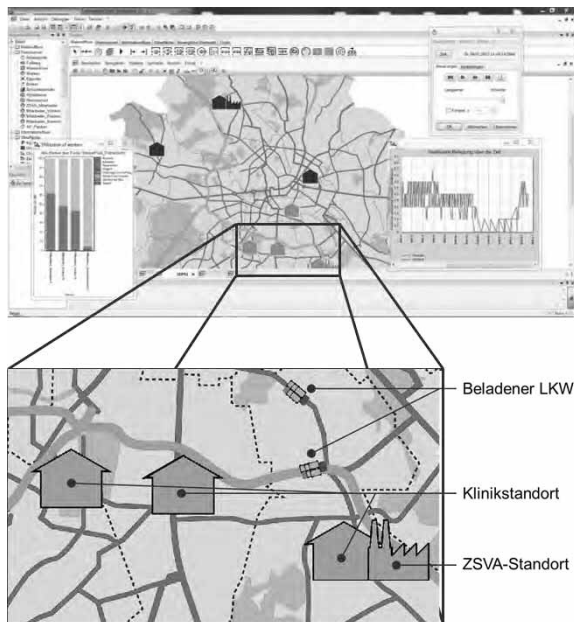


Abbildung 2. Simulationsmodell Ebene 1

Ebenfalls auf Ebene 1 sind die Transporte zwischen den Standorten abgebildet. Für die Simulation der Wegzeiten wurden dabei sowohl Pannenwahrscheinlichkeiten, Stauwahrscheinlichkeiten und eine Verzögerung in der Rushhour berücksichtigt.

Klinik- und Aufbereitungsstandorte stellen die nächste Ebene dar. Da hier an den einzelnen Standorten nicht zwischen Gebäuden (Ebene 2), Funktionsbereichen (Ebene 3) und übergeordneten Funktionsbausteinen (Ebene 4) unterschieden wird, erfolgt die Simulation der eigentlichen Stützprozesse auf Ebene 5. Die Übergabe der beweglichen Elemente zwischen den Ebenen erfolgt durch die Weitergabe der beladenen LKWs.

In den Klinikstandorten wird auf der Grundlage der Datenanalyse ein realistisches Nutzungsverhalten der Operationsinstrumente simuliert. Die Erzeugung von unreinen Instrumenten erfolgt daher charakteristisch zu ca. 90% über die Operationskernzeit und nur zu ca. 10% während Spätsaalzeiten oder Nachts (außerplanmäßige Operationen und Notfälle).

Ebenfalls auf der Datenanalyse basierend werden die Instrumentensets nach Operationsart, Fachbereich, Größe und Sondereigenschaften spezifiziert. So wird

sichergestellt, dass sich ein realistischer Instrumentenmix für die Aufbereitung ergibt und eine Auslegung möglichst realitätsnah erfolgen kann.

Die Aufbereitungsstandorte sind deutlich detaillierter modelliert als die Klinikstandorte. Abgebildet sind Lagerflächen für die Anlieferung von unreinen und für den Abtransport von reinen Instrumenten, eine Vorreinigungszone, die Reinigungs- und Desinfektionsgeräte, die manuellen Packplätze und die Sterilisatoren (Abbildung 3).

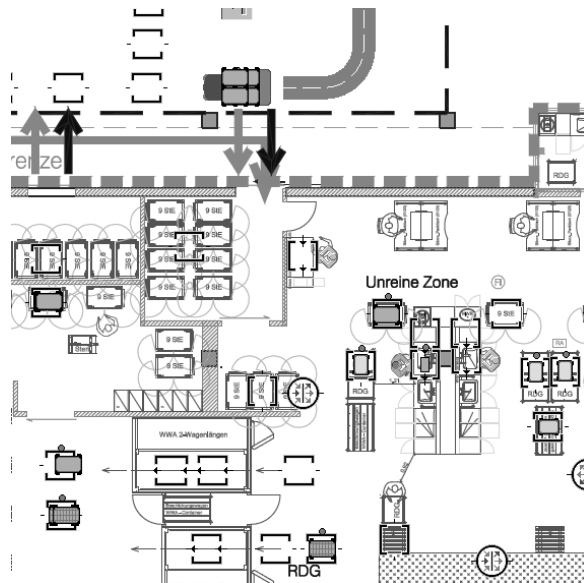


Abbildung 3. Ausschnitt aus dem Simulationsmodell eines Aufbereitungsstandortes

Während für die manuellen Arbeitsschritte Prozesszeiten aus der Datenanalyse zu Grunde gelegt werden, sind die Zykluszeiten für die Reinigungsgeräte den Spezifikationen der zukünftig verwendeten Anlagen entnommen. Aufgrund der Verwendung von sowohl DIN- als auch ISO-genormten Behältern und ihrer unterschiedlichen Abmessungen, ergibt sich ein zu berücksichtigender Ladefaktor der Anlage. Das maximale Fassungsvermögen der Geräte kann nur erreicht werden, wenn eine sortenreine Beladung erfolgt.

Zusätzlich zum Kernprozess zur Reinigung und Desinfektion der Operationsinstrumente werden in einem Nebenprozess die Transportbehälter und -wagen gereinigt. In der Simulation wurde dies ebenfalls berücksichtigt wie die anschließende Konsolidierung der beiden Prozesse.

Der durch die Simulation erzielte Nutzen für die Planung der Sterilgutaufbereitung ist hoch:

- Die Anzahl der einzusetzenden Reinigungsgeräte, der notwendigen Packtische und die Größe der Lagerfläche für reine und unreine Instrumente konnten bestimmt werden.
- Notfallstrategien für den Ausfall einzelner Maschinen und eines ganzen Standortes wurden durch die Simulation validiert.
- Ein optimales und robustes Transportkonzept zur Sicherstellung der Versorgung der Klinikstandorte inklusive Tourenplan konnte entwickelt und validiert werden.
- Ein Betriebskonzept inklusive Schichtmodell und Minimal- und Maximalbesetzung wurde ermittelt.
- Der Nutzen der Reorganisation der Sterilgutaufbereitung konnte dargestellt und an alle Klinikstandorte vermittelt werden.

Erstellung eines zukunftssicheren Logistikkonzepts für einen Krankenhausneubau

In einem weiteren Anwendungsfall wurde das komplette Logistikkonzept eines Klinikneubaus mit Hilfe der Simulationstechnik erstellt. Bei der Planung der Versorgungsstrukturen und -beziehungen gibt es eine Vielzahl von Kombinationsmöglichkeiten, die den Lösungsraum sehr schnell unüberschaubar groß werden lassen. Im vorliegenden Fall wurde eine Vorauswahl der möglichen Kombinationen getroffen, diese mithilfe der Simulationstechnik analysiert und zur weiteren Planung ausgewählt. Dabei stellten sich primär Fragestellungen zur Bereichsanordnung im Neubau und zur Auslegung und Organisation der Transporte. Das zugehörige Simulationsmodell bildet also im Wesentlichen Ebene 2 ab.

Eine konkrete Fragestellung galt der Organisation und Auslegung der Anlieferzone. Die Lokalisierung des Neubaus inmitten eines dicht besiedelten Wohngebiets ließ nur an einer Stelle eine sinnvolle Positionierung der Anlieferung zu, welche zudem räumlich eingeschränkt war. Als Datengrundlage fanden aktuelle Lieferpläne und Materialmengen Eingang, welche entsprechend der zu erwartenden Entwicklungen skaliert wurden (geplante Fallentwicklung des Krankenhauses sowie technische Entwicklungen an Medizinprodukten). Im Simulationsmodell wurde die Anzahl der benötigten LKW-Docks ermittelt und die Zukunftsfähigkeit durch Anpassung der Anlieferfrequenz bestätigt (Abbildung 4). Zudem wurden die benötigten Flächen zur Entladung, Qualitätskontrolle und Warenvereinnehmung dimensioniert. Ein beson-

derer Fokus musste hierbei auf die Anlieferung von Frischwaren gelegt werden, die nur für kurze Zeit ungekühlt bleiben dürfen.

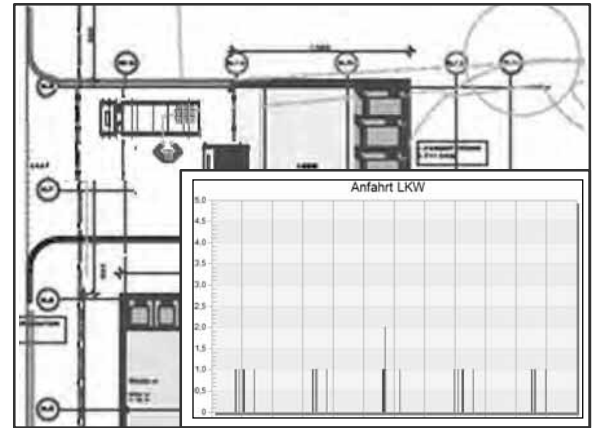


Abbildung 4. Detail des Simulationsmodells (LKW-Dock)

Da die Kubatur des Gebäudes durch die umliegenden Gebäude, Straßen und Grundstücke beschränkt war, stellte auch die Anordnung der Lagerflächen eine zu prüfende Fragestellung dar. Es wurden verschiedene Lokalisierungen geprüft sowie eine räumliche Trennung der Lagerflächen nach Produktkategorie. Durch die Simulation konnte bestätigt werden, dass eine Zentralisierung der Lagerinhalte an einem Ort die wirtschaftlichste Alternative darstellt, auch wenn dieses Lager von den Hauptverbrauchsstellen etwas weiter entfernt lag.

Die Positionierung der Lagerflächen und die dadurch erforderlichen längeren Wege stellten besonders hohe Anforderungen an die Zuverlässigkeit der Transporte zwischen den Bereichen. Eine neuralgische Verbindung bei stockwerksübergreifenden Transporten stellen die Aufzugverbindungen dar, deren Zuverlässigkeit im Simulationsmodell überprüft wurde. Die Dimensionierung wurde daher insbesondere unter Berücksichtigung der Wachstumsfähigkeit überprüft und Alternativkonzepte wie Rutschenverbindungen für Abwärtstransporte (beispielsweise für Wäsche, Abfälle, etc.) wurden in das Konzept einbezogen.

So konnten mit Hilfe der Simulation unbekannte Flächenbedarfe ermittelt werden, die Raum- und Funktionsanordnung optimiert und bewertet werden und neuartige Konzepte zur Versorgung der Bereiche berücksichtigt werden. Die Planungsqualität wurde gesteigert und die Zukunftsfähigkeit der Gebäudestrukturen konnte sichergestellt werden.

Personen- und Bettenaufzugsimulation

Neben logistischen Transportprozessen zwischen Standorten oder innerhalb einer Etage spielen Aufzüge für den Vertikaltransport in einem Krankenhaus eine wesentliche Rolle. Die Verfügbarkeit von Aufzügen und somit die Wartezeit stellt einerseits einen Komfortfaktor für Besucher dar, gleichzeitig kann aber auch die Effizienz von Logistikprozessen beeinträchtigt werden, wenn Aufzüge überlastet sind. Im Krankenhaus kommt der Funktionalität von priorisierten Fahrten bei Notfällen zusätzlich eine besondere Rolle zu. Neben den Personentransporten für Mitarbeiter, Patienten und Besucher stellt die Beförderung von Betten und anderem medizinischen Gerät eine zusätzliche Beanspruchung derselben Anlagen dar. In kaum einer anderen Branche werden die Aufzuganlagen gleichermaßen zur Durchführung logistisch relevanter Transporte wie auch zur Beförderung von Personen ohne Prozessrelevanz genutzt.

Zur Dimensionierung der Aufzugkapazitäten für einen Krankenhausneubau wurde die Simulation genutzt, um diese vielfältige Belastung der Aufzuganlagen abzubilden. Das Beförderungsaufkommen wurde hierbei teilweise aus verfügbaren Daten entnommen (die interne Leistungsverrechnung stellte sich als eine ergiebige Quelle heraus), teilweise mussten Annahmen getroffen werden (Besucherkzahlen).

Die Aufzugsimulation bestand aus zwei Aufzuggruppen mit jeweils zwei Aufzügen und verband dabei insgesamt sieben Etagen. Es erfolgte eine Zuordnung von Transportarten zur wahrscheinlichen Aufzuggruppe: Besucher-Transporte wurden aufgrund der räumlichen Nähe zum Besuchereingang über die Aufzuggruppe A und OP-Transporte vorwiegend über die Aufzuggruppe B durchgeführt. Weiterhin wurden die Versorgung des Hubschrauberlandeplatzes und die Materialversorgung in der Simulation berücksichtigt (Abbildung 5).

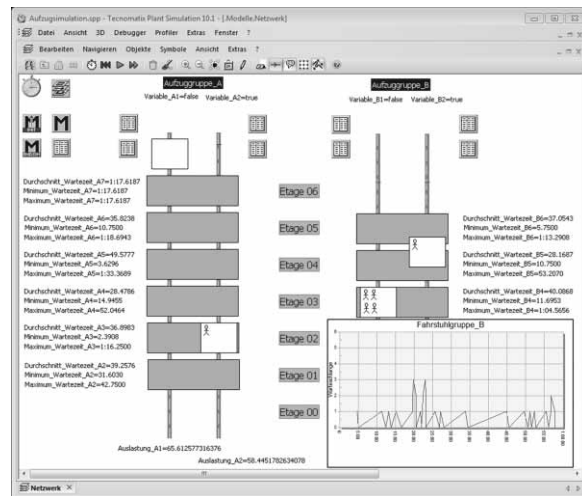


Abbildung 5. Aufzugsimulation im Krankenhaus

Folgende aufzugsrelevante Transportdaten wurden für das Simulationsmodell berücksichtigt:

- Anzahl Patiententransporte
- Anzahl Besucher
- OP- und NFA-Transporte
- Mitarbeiter-Transporte
- Transporte zur Ver- und Entsorgung
- Speiseversorgung

Zur Bewertung der Bauplanung wurden zwei Fahrstuhltypen simuliert: der Standardfahrstuhl mit einer Fördergeschwindigkeit von 0,6 m/s und eine leistungsstärkere Variante mit einer Fördergeschwindigkeit 1 m/s.

Insgesamt wurden drei Lastfälle von je einer Stunde simuliert, die im Tagesverlauf unterschiedliche Anforderungen an die Aufzüge stellen:

- 7-8 Uhr: Hohes Transportaufkommen durch Mitarbeiterverkehr und Ver- und Entsorgung (Frühstück)
- 10-11 Uhr: Hohes Transportaufkommen durch den Transportdienst und Ver-/ Entsorgung (Mittagessen + Warenverkehr)
- 15-16 Uhr: Hohes Transportaufkommen durch Besucherverkehr und Ver-/ Entsorgung (Abendessen)

Auf Basis dieser Eingangsdaten konnte die Bewertung der Aufzugkapazität hinsichtlich Leistungsfähigkeit bei Spitzenbelastung und Wirtschaftlichkeit im Normalbetrieb sichergestellt werden.

Simulation der Materialversorgung und Logistik

Um zukünftig optimale Prozesse zur Sterilgutversorgung sicherstellen zu können, wurde in einem weiteren Planungsprojekt ein geschlossenes Logistikkonzept der Sterilgut-Logistik erarbeitet und mittels Simulation abgesichert. Dabei mussten alle Prozessschritte - von den Abläufen in der zentralen Sterilisation (Ebenen 3-5), über die Anlieferung ins Krankenhaus bis zur Feinverteilung zu den OP-Zonen, den Ambulatorien und den Stationen (Ebenen 1 und 2) - ausgearbeitet werden. Insbesondere hausinterne Restriktionen und Engpässe mussten hierbei berücksichtigt werden (z.B. Aufzugskapazitäten, Stellflächen, etc.). Zudem sollten die benötigten Personalressourcen aus dem definierten Prozess abgeleitet werden.

Zur Gewährleistung einer ganzheitlichen Planung und Absicherung des entwickelten Soll-Konzeptes wurde der geplante Materialfluss in einem Simulationsmodell abgebildet. Merkmale wie Layouts, Gebäudestrukturen, definierte logistische Abläufe sowie definierte Prozess- und Nebenzeiten fanden ebenso Berücksichtigung. Neben zeitlichen und räumlichen Restriktionen wurden auch organisatorische Rahmenbedingungen innerhalb des Logistikprozesses in der Sterilgutversorgung hinterlegt. Die Ergebnisse aus den Simulationsstudien wurden anschließend hinsichtlich eines Nutzwerts untereinander verglichen.

Im Rahmen von Simulationsexperimenten konnten Engpässe identifiziert und beseitigt werden. Weiterhin wurde ein optimaler Lieferplan für die LKW-Transporte entwickelt, welcher iterativ optimiert wurde (Abbildung 6). Die geplanten Anfahrzeitpunkte wurden werktags und am Wochenende um jeweils eine Fahrt reduziert. Ebenso konnten die Standzeiten zum Be- und Entladen um ca. 20% reduziert werden.

Tag	Geplante Zeiten		Reduzierte Zeiten	
	Geplante Zeiten	Ø Standzeit [min]	Reduzierte Zeiten	Ø Standzeit [min]
Montag bis Freitag	06:15 Uhr	Mo: 15 Min Di-Fr: 45 Min	06:15 Uhr	Mo: 15 Min Di-Fr: 33 Min
	13:00 Uhr	23 Min	—	—
	16:30 Uhr	23 Min	14:30 Uhr	28 Min
	22:15 Uhr	30 Min	22:15 Uhr	41 Min
Wochenende	06:15 Uhr	Sa: 43 Min So: 18 Min	—	—
	16:30 Uhr	9 Min	09:00 Uhr	Sa: 28 Min So: 18 Min
	22:15 Uhr	4 Min	21:00 Uhr	14 Min
	—	—	—	—

Abbildung 6. Geplante und reduzierte Anfahrzeitpunkte

Die optimierte Tourenplanung belegt, dass durch Simulation Zukunftskonzepte überprüft und prozess-

sicher ausgelegt werden können. Reduzierte Standzeiten, optimierte Ladefaktoren und der Wegfall von täglich einer Tour stellt eine relevante Kosteneinsparung für das Krankenhaus dar.

Zusammenfassung und Fazit

Immer mehr Kliniken holen sich externe Unterstützung von professionellen Unternehmensberatungen. Kostendruck, sinkende Umsätze, fehlende Flexibilität die Liste aktueller Herausforderungen in den Kliniken ist lang und komplex. Zusätzlich spielen bei der Prozess- und Gebäudeplanung die Logistik- und Stützprozesse eine immer wichtigere Rolle: Wie wird eine effiziente Logistik innerhalb eines Kliniknetzwerkes abgebildet? Wie ist eine Wäscherei, eine Sterilgutaufbereitung, ein Zentrallager oder eine Zentralküche zu dimensionieren, um den Klinikverbund optimal versorgen zu können? Reichen Fahrstuhl- und Intralogistikkapazitäten aus, um bei steigender Fallzahl eine Versorgung der Klinikbereiche mit Medikamenten, Essen, sauberer Wäsche und Operationsinstrumenten sicherzustellen?

Die UNITY setzt zur Beantwortung dieser Fragestellungen innerhalb der klinischen Prozesskette die Simulation ein. Damit wurde der erfolgreiche Einsatz der Simulationstechnik im Klinikumfeld auf die Planung und Optimierung der Logistik- und Stützprozesse ausgeweitet und ein weiterer Grundstein zur vollständig digitalen Planung in dieser Branche gelegt.

References

- [1] J. Debatin, A. Ekkernkamp und B. Schulte, Krankenhausmanagement: Strategien, Konzepte, Methoden, Berlin: MWV Medizinisch Wissenschaftliche Verlagsgesellschaft, 2010.
- [2] C. Matysczok, „Erfolgreiche Etablierung von Best Practices der Fabrikplanung im Klinikumfeld,“ in *Arbeitszeitgestaltung im Krankenhaus*, Karlsruhe, 2012.
- [3] J. Gausemeier, C. Plass und C. Wenzelmann, Zukunftsorientierte Unternehmensgestaltung: Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen, München: Carl Hanser Verlag GmbH & Co. KG, 2009.
- [4] C. Matysczok, M. Schele und H. Wilson, „Digitale Klinikplanung - Reibungsloser Betrieb durch Simulation,“ in *ASIM SPL - Entscheidungsunterstützung von der Planung bis zur Steuerung*, Paderborn, 2013.
- [5] T. Pfänder und J. Fischlein, Industrielles Klinikmanagement, München: FinanzBuch

Verlag, 2008.

- [6] J. Klauber, M. Geraedts und J. Friedrich,
Krankenhausreport 2010, Stuttgart: Schattauer
GmbH, 2010.
- [7] K. Eisenring und D. Saladin, „Spitallogistik -
Einsparungen sind möglich,“ *Management und
Qualität*, 2010.
- [8] C. Matysczok, M. Kühn, M. Schele und Batija,
Gime, „End-to-End Kliniksimulation –
Absicherung der Klinikplanung durch Simulation
vollständiger Prozessketten,“ in *ASIM 2015*,
Lippstadt, 2015.

Automatische Klassifikation von Herzschlägen im Elektrokardiogramm mit Hilfe von Support Vector Machines und Self Organizing Maps

Natalie Gschaider¹, Martin Bachler², Christopher Mayer², Siegfried Wassertheurer²

¹Technische Universität Wien, Institut für Analysis and Scientific Computing,
Wiedner Hauptstraße 8, 1040 Wien, Austria

²AIT Austrian Institute of Technology GmbH, Health & Environment Department,
Business Unit Biomedical Systems, Donau City Straße 1, 1220 Wien, Austria
*Natalie.Gschaider@gmx.at, Martin.Bachler@ait.ac.at, Christopher.Mayer@ait.ac.at,
Siegfried.Wassertheurer@ait.ac.at*

Herz-Kreislauf-Erkrankungen sind in Österreich die häufigste Ursache für einen frühzeitigen Tod, was ForscherInnen und MedizinerInnen veranlasst, Früherkennungssysteme zu konzipieren. Das Ziel dieser Arbeit ist, basierend auf von MedizinerInnen annotierten EKG-Daten einen Algorithmus zu entwickeln, der Herzrhythmusstörungen in EKG-Aufzeichnungen von verschiedenen Herzpatienten analysiert und klassifiziert. Es werden dafür Self Organizing Maps (SOM) und Support Vector Machines (SVM) angewendet. Zunächst werden die entsprechenden Daten vorverarbeitet und für die Klassifikation aufbereitet. Dafür werden Statistiken höherer Ordnung (HOS) und Hermiteische Basisfunktionen (HER) zur Merkmalsextraktion angewendet. Die Evaluierung der Resultate erfolgt durch einen Vergleich mit den Annotationen der Daten. Die Klassifikation der annotierten Daten ergab eine Erkennungsrate von 74.7% für die SOM. Die SVM wies unter Verwendung der beiden vorverarbeitenden Methoden HOS und HER eine Erkennungsrate von 88.4% für HOS und 87% für HER auf. Diese werden durch ein Verfahren zur Kombination (gewichtetes Voting Prinzip) mit einem durchschnittlichem Wert von 92.8% deutlich verbessert. Abschließend kann diese Kombination damit als Basis für eine automatische EKG-Annotation dienen.

1 Einleitung

Da Herzrhythmusstörungen oft ein Indiz für Herzkrankheiten sind, stellen sie einen guten Ausgangspunkt dar, eine genauere Untersuchung des Patienten einzuleiten. Mittels EKG aufgezeichnete Daten können dafür so ausgewertet werden, dass diese Rhythmusstörungen klassifiziert werden können. Störungen sind durch Abweichungen der regelmäßigen Herzschlagfolge oder der üblichen Form des EKG-Signals der einzelnen Herzschläge gekennzeichnet. Dabei gilt es, Arten von Herzschlägen zu bestimmen und in weiterer Folge Herzrhythmusstörungen von gesunden Herzrhythmen zu unterscheiden. Das Ziel dieser Arbeit ist also die automatische Klassifikation von be-

reits detektierten Herzschlägen, aber nicht deren Detektion.

2 Daten

Die verwendeten Daten stammen aus der MIT-BIH Arrhythmia Database, die vom Massachusetts Institute of Technology sowie vom Beth Israel Hospital 1980 erstellt wurde [1]. Diese wird von PhysioNet online frei zur Verfügung gestellt und enthält eine große Sammlung von aufgezeichneten physiologischen Signalen [2]. PhysioNet garantiert die Anonymisierung der Daten, daher können sie ohne weitere Zustimmung einer Ethik-Kommission verwen-

det werden. Die vom BIH gesammelten Daten bilden insgesamt eine Menge von 4000 24-stündigen EKG-Aufzeichnungen von stationär behandelten Patienten (60%) und ambulant behandelten Patienten (40%). Die Abtastrate der Aufzeichnungen ist 360 Hz. Neben dem normalen Herzrhythmus werden 14 Arten von Arrhythmien, die im Zuge dieser Arbeit mit verschiedenen Methoden klassifiziert (siehe Tabelle 1). Die Klassifikation und die Bestimmung ihrer Qualität erfordert eine Aufteilung in Trainings- und Testdaten. Hier wird das Verhältnis von 80 zu 20 in Trainings- und Testdaten für höher besetzte Typen gewählt sowie ein Verhältnis von 90 zu 10 für kleinere Datengruppen.

Tabelle 1: Auflistung der verschiedenen Typen von Herzschlägen oder Herzrhythmen in der MIT-BIH Datenbank [1, 2].

	Herzschläge	Abk.
1.	normal beat	N
2.	left bundle branch block beat	L
3.	right bundle branch block beat	R
4.	atrial premature beat	A
5.	aberrated atrial premature beat	a
6.	nodal (junctional) premature beat	J
7.	ventricular premature beat	V
8.	fusion of ventricular and normal beat	F
9.	ventricular flutter wave	I
10.	atrial escape beat	e
11.	nodal (junctional) escape beat	j
12.	ventricular escape beat	E
13.	paced beat	P
14.	fusion of paced and normal beat	f
15.	unclassifiable beat	Q

3 Merkmalsextraktion

Zur Verarbeitung der EKG-Daten müssen Merkmale aus den Signalabschnitten in der Umgebung der R-Zacken extrahiert werden. Diese Phase ist eine sehr entscheidende, da ohne diese keine aussagekräftige Klassifikation vorgenommen werden kann.

Zuerst wird eine Region der Länge d um die annotierten R-Zacken (n Herzschläge) ausgewählt. Daraus ergibt sich eine $n \times d$ - Matrix. Jeder i -te Herzschlag

der Länge d wird dann zu einem Merkmalstupel y der Länge M (Feature vector) umgewandelt,

$$\begin{aligned} x^{(i)} &= (x_1^{(i)}, x_2^{(i)}, \dots, x_d^{(i)}) \Rightarrow \\ y^{(i)} &= (y_1^{(i)}, y_2^{(i)}, \dots, y_M^{(i)}) \end{aligned} \quad (1)$$

indem er durch seine Merkmale (Features) beschrieben wird. Diese stellen eine Annäherung an die Form der EKG-Daten, genauer der QRS-Verläufe der einzelnen Herzschläge, dar. Dieser Merkmalstupel entspricht somit den am meisten relevanten Aspekten eines jeden Herzschlages. Für die Klassifikation reichen schon wenige Merkmale zur Unterscheidung aus, daher fällt der Merkmalstupel kürzer als die Länge eines Herzschlages aus.

3.1 Higher Order Statistics

Statistiken höherer Ordnung (Higher Order Statistics, HOS), wie Momente und Kumulanten, werden in der Signalverarbeitung zur Analyse von Signalen verwendet. Die Statistiken enthalten Informationen über Amplituden und Phasen eines Signalprozesses [3].

Im Allgemeinen gilt, dass die Verwendung von Kumulanten im Falle von stochastischen Signalen bestimmte Vorteile bringt, während es für deterministische Signale besser ist, Momente zu gebrauchen [3]. Bei den EKG-Daten handelt es sich um stochastische Signale, also sind im Weiteren nur die Kumulanten von Interesse. Diese werden über die Momente hergeleitet [4].

Das EKG-Signal $X(t)$ wird als stochastischer Prozess gesehen. Die Momente einer Verteilungsdichtefunktion einer Zufallsvariablen X werden durch $m_{nx} := E[x^n] = \int_{-\infty}^{\infty} x^n f_X(x) dx$ als n -tes Moment definiert. Die kumulantenenerzeugende Funktion K_X entsteht durch Logarithmierung der momentenerzeugenden Funktion:

$$K_X(\omega) = \ln(E[e^{i\omega x}]). \quad (2)$$

Die n -te Kumulante oder Kumulante der Ordnung n der Verteilung von X ist definiert als

$$k_{nx} := (-i)^n \left. \frac{d^n K_X(\omega)}{d\omega^n} \right|_{\omega=0}. \quad (3)$$

Kumulanten der 2., 3. und 4. Ordnung entsprechen den Koeffizienten der Taylor-Entwicklung der kumulantenenerzeugenden Funktion. Für einen stochastischen

Prozess $\{X(t), t \in \mathbb{Z}\}$ sind die Kumulanten der 2. und 3. Ordnung gleich der entsprechenden Momente. Kumulanten der 4. Ordnung brauchen Informationen über die Momente der 2. und 4. Ordnung [5, 3].

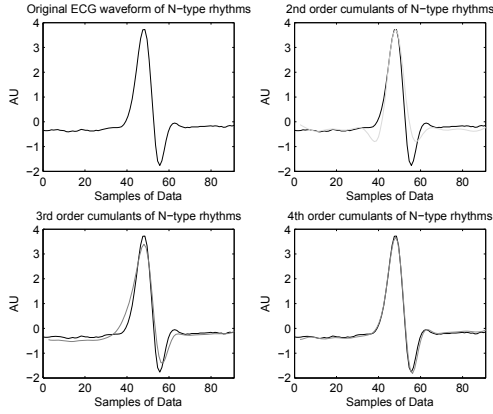


Abbildung 1: Normaler Herzschlag und dessen entsprechenden Repräsentationen durch Kumulanten der 2., 3. und 4. Ordnung

Die Verwendung der Higher-Order Spectral Analysis (HOSA) Toolbox in MATLAB® brachte gute Erfolge [6]. Mit dieser Methode wurden für einen 91-elementigen Vektor, d.h. $d = 91$ aus Gleichung (1), der den QRS-Komplex eines Herzschlags repräsentiert, 91 Kumulanten berechnet. Nach S. Osowski et al. [5] wurden insgesamt 15 Kumulanten, 5 pro Ordnung, jeweils zu den Zeitverschiebungen 15, 30, 45, 60 und 75, also gleichverteilt über den QRS-Komplex, als features gewählt. Je höher die Ordnung, desto genauer ist die Annäherung (siehe Abbildung 1).

Wenn man die HOS Charakteristiken für verschiedene Herzrhythmen vergleicht, bemerkt man, dass die Unterschiede zwischen ihnen deutlicher werden und somit die Herzrhythmen besser erkennbar und leichter zu unterscheiden sind [5].

3.2 Hermiteische Basisfunktionen

Eine andere mögliche Repräsentation der Herzsignale können die Koeffizienten der Hermiteischen Basisfunktionen (HER) leisten [5, 7].

Zuerst wird das EKG-Signal aus einem um den Bezugspunkt (R-Zacke) zentrierten $t_0 = 250$ ms Zeitfenster extrahiert. Jeder QRS-Komplex wird dann

durch einen $2t_0$ -dimensionalen Vektor $\vec{x} = x(t)$ repräsentiert - 250 ms QRS-Komplex mit einem 125 ms Signal von Nullen an jeder Seite. Dadurch wird sichergestellt, dass die Signalwerte außerhalb des QRS-Komplexes nahe genug an Null sind. Genauer besteht das QRS-Segment aus 91 Datenpunkten um die R-Zacke mit 45 Punkten vor und nach R, die Nullen sind zusätzliche 45 Datenpunkte links und rechts vom QRS-Komplex (also insgesamt 181 Datenpunkte).

Die EKG-Daten werden mithilfe der sogenannten Hermiteischen Basisfunktionen dargestellt. Das EKG-Signal $x(t)$ ist dann gleich der Entwicklung in Hermiteischen Basisfunktionen der n -ten Ordnung ϕ_n mit Koeffizienten c_n für $n = 0, \dots, N-1$:

$$x(t) = \sum_{n=0}^{N-1} c_n(\sigma) \phi_n(t, \sigma), \quad (4)$$

wobei σ einen Parameter für die Weite darstellt, der proportional zur Breite des QRS-Komplexes steht.

Die Hermiteischen Basisfunktionen sind folgendermaßen definiert:

$$\phi_n(t, \sigma) = \frac{1}{\sqrt{\sigma 2^n n! \sqrt{\pi}}} e^{-\frac{t^2}{2\sigma^2}} H_n\left(\frac{t}{\sigma}\right), \quad (5)$$

wobei H_n die Hermiteischen Polynome bedeuten und rekursiv geschrieben werden als

$$H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x) \quad \text{für } n > 1 \in \mathbb{N} \quad (6)$$

mit $H_0(x) = 1$ und $H_1(x) = 2x$.

Als Merkmale der EKG-Kurven werden die Koeffizienten c_n der Hermiteischen Basisfunktionen gewählt, da durch diese die Annäherung an die originale Kurve erfolgt. Man erhält diese durch Minimierung der Abweichungen zwischen dem EKG-Signal und dessen Approximation.

Hermiteische Basisfunktionen werden somit mit ziemlich genauen Ergebnissen an die originalen EKG-Daten approximiert (siehe Abbildung 2). Je höher die Ordnung der Hermiteischen Basisfunktionen, desto genauer wird die Approximation. Ab der 9. Ordnung ist die Approximation sehr nahe am Kurvenverlauf.

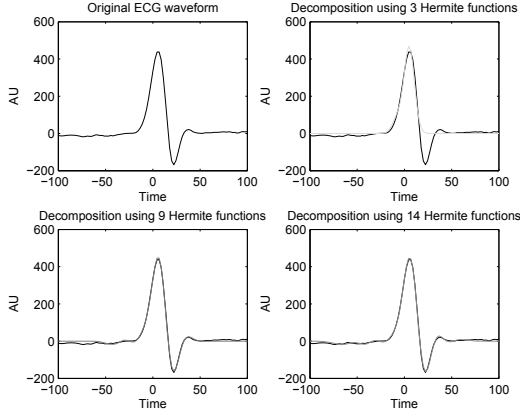


Abbildung 2: Normaler Herzschlag und dessen Hermitesche Repräsentationen der 3., 9. und 14. Ordnung

4 Methoden zur Klassifizierung

Für die Klassifikation gibt es überwachte und unüberwachte Lernverfahren, wobei überwacht die Zuordnung von Datenpunkten zu einer vorgegebenen Klasse und unüberwacht die automatische Einteilung der Datenpunkte in Cluster meint.

4.1 Support Vektor Maschine

Die Grundidee einer Support Vektor Maschine (SVM) ist die Konstruktion einer optimalen Hyperebene, der sogenannten Entscheidungsebene, die alle Daten einer Klasse von denen einer anderen mit dem größten Abstand voneinander trennt [8].

Die Wahl der optimalen Hyperebene unter den möglichen Hyperebenen schließt also mit ein, dass der Generalisierungsfehler des Klassifikators klein gehalten bleibt. Dieser maximale Abstand ist die Summe der Abstände von der Hyperebene zu den nächsten Datenpunkten von jeder der beiden Klassen. Diese nächst gelegenen Datenpunkte werden Support-Vektoren (SVs) genannt, denn sie beschreiben („stützen“) die Hyperebene [8], siehe Abbildung 3.

Angenommen, die Merkmalsextraktion der Rohdaten wurde schon durchgeführt. Dann seien die annotierten Trainingsdaten gegeben durch $\mathbf{y}^{(i)} \in \mathbb{R}^M$ mit den Kennzeichnungen oder Labels $l_i \in \{-1, +1\}$ für $i = 1, \dots, n$ Herzschläge, wobei -1 für Klasse 1 und $+1$

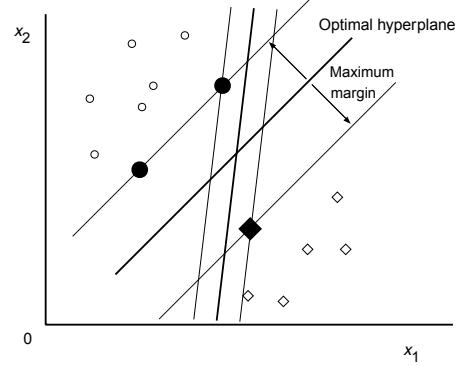


Abbildung 3: Support Vektor Maschine für linear separierbare Daten in einem 2-dimensionalen Raum. Optimale separierende Hyperebene mit maximalem Abstand zu den nächsten Datenpunkten jeder Klasse [9].

für Klasse 2 steht.

Sind die Daten $\in \mathbb{R}^M$ zweier Klassen linear trennbar, dann sei die M -dimensionale separierende Hyperebene oder Entscheidungsfunktion $D(\mathbf{y})$ gegeben durch

$$D(\mathbf{y}) = \mathbf{w} \cdot \mathbf{y} + b, \quad (7)$$

wobei \mathbf{w} ein M -dimensionaler Vektor und b ein Verschiebeterm ist und für $i = 1, \dots, n$ gilt

$$\mathbf{w} \cdot \mathbf{y}^{(i)} + b = \begin{cases} > 0 & \text{für } l_i = 1, \\ < 0 & \text{für } l_i = -1. \end{cases} \quad (8)$$

Die optimale, separierende Hyperebene kann dann durch folgendes Minimierungsproblem für \mathbf{w} und b gelöst werden:

$$\text{Minimiere } Q(\mathbf{w}, b) = \frac{\|\mathbf{w}\|^2}{2} \quad (9)$$

unter der Nebenbedingung (u.d.N.)

$$l_i(\mathbf{w} \cdot \mathbf{y}^{(i)} + b) \geq 1, \quad \text{für } i = 1, \dots, n.$$

Dieses Optimierungsproblem wird zu einem Problem der Quadratischen Programmierung [9]. Das QP-Problem wird zur Berechnung in ein duales Problem unter Verwendung der Methode der Lagrange Multiplikatoren umgewandelt. Die Lösung erfüllt dann die sogenannten Karush-Kuhn-Tucker (KKT) Bedingungen.

Da die EKG-Daten keine linear separierbaren Daten sind und in diesem Fall die Entscheidungsfunktion

keine lineare Funktion der Daten ist, werden die Daten aus dem Eingangsraum in einen höher-dimensionalen Merkmalsraum durch eine nichtlineare Transformation abgebildet. In diesem Merkmalsraum höherer Dimension wird die verallgemeinerte optimale Trennhyperebene konstruiert. Die Daten können im Merkmalsraum linear getrennt werden, obwohl sie im Eingangsraum nicht linear trennbar sind [8].

4.2 Selbst Organisierte Karte

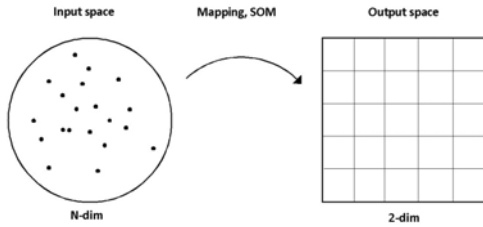


Abbildung 4: Die SOM Abbildung von einem höher dimensional Raum zu einem diskreten 2-D-Raum [7].

Die Idee der Selbst Organisierten Karte (SOM) ist die Abbildung einer Menge von M -dimensionalen Inputvektoren $\mathbf{y}^{(i)}$ für $i = 1, \dots, n$ auf einen diskreten 2-dimensionalen Raum mit K Positionen durch ein neuronales Netzwerk (siehe Abbildung 4). Jede Position im Ausgangsraum wird durch einen Ausgangsknoten h_j für $j = 1, \dots, K$ repräsentiert. Jedem h_j wird eine Position ω_j im Eingangsraum zugeordnet. Der Abstand zwischen h_j und einem Eingangsvektor $\mathbf{y}^{(i)}$ ist gegeben durch

$$d_j = \sum_{k=1}^M \left(\omega_{jk} - y_k^{(i)} \right)^2. \quad (10)$$

Für jedes Merkmalstupel $\mathbf{y}^{(i)}$ wird das h_j gewählt, zu dem der Abstand am kleinsten ist. Für das gewinnende Neuron h_{i^*} gilt dann

$$d_{i^*} = \min_j d_j. \quad (11)$$

Die Eingangsvektoren mit h_{i^*} als Gewinner werden dem Cluster (Ausgangsknoten) i^* zugeordnet. Die Positionen ω_j von h_j im Eingangsraum können als Gewichte im Zuordnungsprozess gesehen werden. Die Gewichte werden nach jedem Schritt aktualisiert, sodass h_j auf die vorliegenden Muster eingehen kann.

Das erfolgt durch

$$\omega_{jk} \longrightarrow \omega_{jk} + \Delta \omega_{jk} \quad (12)$$

$$\Delta \omega_{jk} = \begin{cases} \eta (y_k^{(i)} - \omega_{jk}) & \text{wenn } j = i^* \\ 0 & \text{wenn } j \neq i^* \end{cases} \quad (13)$$

wobei η die Schrittgröße in Richtung des Merkmaltupels darstellt, die sich im Laufe des Trainings ändert.

Es ist oft wünschenswert, eine topologische Ordnung unter den Ausgabeeinheiten zu haben, sodass benachbarte Einheiten im Ausgangsraum auf benachbarte Punkte im Eingangsraum antworten. Eine solche topologische Karte zu erhalten entspricht der Technik, die Einheiten in der Nähe vom Gewinner-Knoten in der gleichen Weise zu aktualisieren wie den Gewinner-Knoten selbst. Umgesetzt wird dies mit einem Schritt, der zusätzlich zu η auch von der Distanz r zum Gewinner Neuron im Ausgangsraum abhängt. Auf diese Weise enden benachbarte Einheiten mit ähnlichen Gewichtsvektoren. Dies wird mit sogenannten Potts Neuronen vollzogen, unter diesen der Gewinner-Knoten seinen Schritt mit seinen Nachbarn „teilt“. Gleichung (13) ändert sich zu

$$\Delta \omega_{jk} = \eta (y_k^{(i)} - \omega_{jk}) v_{i^*m} \quad \text{für alle } m, \quad (14)$$

mit

$$v_{i^*m} = \frac{e^{-r_{i^*m}^2/T}}{\sum_k e^{-r_{i^*k}^2/T}}, \quad (15)$$

wobei r_{ij} die Distanz zwischen den Ausgabeeinheiten in der Ausgabeschicht und somit die Topologie des Netzwerkes definiert. Die Weite T verringert sich während der Trainingsphase. Für $T \rightarrow 0$ wird Gleichung (14) zu (13) [7].

5 Kombination der Klassifizierer

Einzelne Klassifikatoren zu einem großen ganzen Gesamtklassifikator zu vereinen (siehe Abbildung 5), erfolgt durch das gewichtete Votieren (*weighted voting principle*). Das bedeutet, es wird für einen Herzschlag das Ergebnis mit dem größten Gewicht als Resultat der Gesamtklassifikation gewählt. Mit anderen Worten, das Ergebnis mit dem größten Gewicht entspricht der am häufigst vorkommenden Klasse unter den Einzelklassifizierern [5]. Es können entweder verschiedene Netzwerke von Klassifikatoren oder nur verschiedene Methoden zur Vorverarbeitung der Daten

in einem Klassifikator zusammen geschaltet werden [5]. Die Abbildung 5 zeigt das Schema eines solchen Klassifikationssystems. In dieser Arbeit war einerseits die Kombination mehrerer SVMs erforderlich, da die SVM einen binären Klassifikator darstellt, der immer nur Daten zweier Klassen voneinander trennt. Das bedeutet für 15 Klassen gibt es 105 Einzelklassifikatoren, deren Einzelresultate zu einem Gesamtergebn gewichtet wird. Dieses wurde für jede verwendete Methode zur Merkmalsextraktion gebildet. Andererseits wurden diese Gesamtergebnisse wiederum zu einem neuen Resultat vereint.

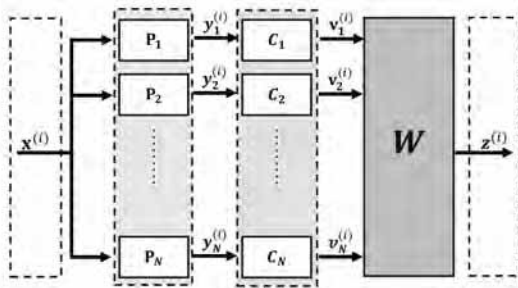


Abbildung 5: Schema einer Gesamtklassifikation.

K ... Anzahl der Klassen

N ... Anzahl der Klassifikatoren

P_1, \dots, P_N = Aufbereitungsblöcke

C_1, \dots, C_N = neuronale Klassifikatoren (classifier)

W ... Integrationsmatrix, Kombination der Resultate der Einzelklassifikatoren

$z^{(i)} = (z_1^{(i)}, z_2^{(i)}, \dots, z_N^{(i)})$... klassifizierter Herzschlag nach der Gesamtklassifikation

6 Resultate und Diskussion

Wie man aus den Tabellen 2 und 3 erkennen kann, zeigen sich bei der Klassifikation durch die Support Vektor Maschine und vor allem bei der Kombination von den Vorverarbeitungsmethoden HOS und HER die besten Ergebnisse. Schlechter schnitt die Klassifikation durch die Selbst Organisierte Karte ab.

Optimale Resultate ergaben sich für V mit 1.99% relativer Fehlerrate bei SVMs mit kombinierter Auswertung von HOS und HER Merkmalen, für N mit 1.25% bei SVMs mit HER Merkmalen, für V mit 2.99% bei SVMs mit HOS Merkmalen und für N mit 3.19% bei SOM mit HER. Die Kombination der SVMs unter

Verwendung der HOS und HER Merkmale erzielt sogar eine perfekte Klassifikation in den Gruppen P und E mit jeweils 0.00% relativem Testfehler. Die Gruppen e und Q wurden durch jede implementierte Methode zu 100% falsch klassifiziert.

Aus Tabelle 2 ist klar ersichtlich, dass durch die SVM große Gruppen wie N oder auch V, für die viele Datensätze aus der Datenbank zur Verfügung stehen, besser klassifiziert werden. Die Ursache dafür ist, dass durch eine hohe Anzahl an Daten die SVM für ein breiteres Spektrum einer Gruppe besser trainiert ist. Eine ähnliche Tendenz zeigte sich auch bei Osowski et al. [5]. Umgekehrt werden e und Q falsch klassifiziert. Das ist auf die geringe Anzahl an Daten, die für diese Gruppen zur Verfügung stehen, zurückzuführen.

Weitere gängige Parameter zur Evaluierung der Erkennungsleistung eines Klassifikators sind der positive Vorhersagewert (Precision) und die Sensitivität (Recall) eines Klassifikators. Die Hälfte der Precision- und Recall-Werte in der Tabelle 3 ergaben mit 0.9 bis 1.0 ein sehr gutes Ergebnis. Die Ausreißer e und Q spiegeln sich auch in diesen Werten mit 0 und undefiniertem Ergebnis für alle Methoden wieder. Optimale Werte ergeben sich für die Gruppe E mit Precision-Werten 1 für alle Methoden und Recall-Wert 1 für die Kombination der Methoden HOS und HER. Auch diese Ergebnisse werden somit durch Kombination deutlich verbessert. Abschließend geht eindeutig hervor, dass die Kombination der beiden Methoden HOS und HER für die SVM die besten Ergebnisse erzielte (Tabelle 2), analog zu [5]. Das Resultat war 6.73% für die relative Gesamttestfehlerrate sowie die besten Ergebnisse bei Precision und Recall bei der Kombination beider Methoden. Dieser Erfolg ist insbesondere auf das einfach zu implementierende Modell des gewichteten Votings zurückzuführen.

Im Gegensatz zur Publikation von Osowski et al. [5] werden in dieser Arbeit 15 Typen von Herzschlägen verwendet, um eine gleichzeitige Bearbeitung und einen weitest möglichen Vergleich von dieser und der Veröffentlichung von Lagerholm et al. [7] zu erreichen. Einige tendenzielle Übereinstimmungen mit der Publikation von Osowski et al. [5] finden sich bei der besseren oder schlechteren Klassifikation von den einzelnen Typen. Zum Beispiel wurden mit der Methode HOS die Gruppen a und j mit einer Fehlerrate von 50% und 48.65% sehr schlecht klassifiziert (siehe Tabelle 2). Im Vergleich dazu wurden diese beiden bei Osowski et al. sogar am schlechtesten bewertet, wo-

bei die Klassen e und Q aber ausgeschlossen sind. Außerdem geht aus der Tabelle 2 noch hervor, dass J als drittkleinste Klasse mit der Methode HER nach e und Q mit einem Wert von 71.43% am schlechtesten abschneidet, analog zu Osowski et al. [5]. Die Gesamtfehlerrate pro Methode hängt also sehr stark von der Anzahl und der Wahl der Daten in den Gruppen ab.

Die SOM ist in der Klassifikation nur mit einer Erkennungsrate um 75% erfolgreich. Nur in höherbesetzten Klassen wie N, L, R, V und P wurden überhaupt verwertbare Ergebnisse erzielt, wie z.B. der Bestwert für die Klasse N mit einer Fehlerrate von 3.19%. Im Gegensatz dazu zeigten die niedrigbesetzten Klassen eine Fehlerrate von 100%. Alle Gruppen mit weniger als 2546 Datensätzen sind zu klein oder zu inhomogen, um erkannt zu werden. Es konnten jedoch 84557 von insgesamt 109813 gewählten Datensätzen von der SOM richtig klassifiziert werden. Den Großteil macht jedoch die Gruppe der normalen Herzschläge N aus, die ca. 70% der ganzen Datenmenge einnehmen. Es wird gezeigt, dass die meisten Cluster als N klassifiziert wurden und daher in die Gruppe N eingehen. Für die Precision- und Recall-Werte dieser SOM liegen nur aussagekräftige Werte für die höherbesetzten Klassen vor (siehe Tabelle 3).

Ein Vergleich der Erkennungsraten (accuracy rates) der einzelnen Methoden ergibt 88.4% für die SVMs unter Verwendung von HOS Merkmalen, 87% für die SVMs mit HER Merkmalen, 92.8% für die SVMs mit der Kombination von HOS und HER Merkmalen sowie 74.7% für die SOM unter Verwendung der HER Merkmale.

Das Ergebnis dieser Arbeit unterliegt mehreren Einschränkungen. Zuerst wurden die Klassifikatoren nur auf die Daten aus der MIT-BIH Arrhythmien Datenbank angewendet und in vorgegebene annotierte Klassen aufgeteilt. Das Ungleichgewicht zwischen pathologischen und nicht pathologischen Daten verursachte eine Tendenz in den Resultaten. Außerdem hatten einige Klassen offenbar zu wenig Daten, um ein sinnvolles Training durchzuführen.

7 Fazit

Im Zuge dieser Arbeit wurden mehrere Algorithmen zur automatischen Klassifikation von Herzschlägen in

EKG-Daten entwickelt und miteinander verglichen. Dabei wurde gezeigt, dass für reale EKG Messungen, welche naturgemäß Störungen, Artefakte und Outlier enthalten, eindeutig die SVM mit einer Kombination aus HOS und HER zur Featureextraktion die geeignetere Wahl ist. Ein nächstes Vorhaben ist, die Erkennungsrate von 92.8% weiter zu verbessern, da sie für klinische Zwecke noch zu wenig ausreichend ist.

Literatur

- [1] G. Moody und R. Mark. *The impact of the MIT-BIH Arrhythmia Database*. IEEE Eng in Med and Biol, Vol. 20(3), S. 45-50, Mai-Juni 2001.
- [2] A. L. Goldberger et al. *PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals*. Circulation, Vol. 101(23), S. 215-220, Juni 2000.
- [3] C. L. Nikias und J. M. Mendel. *Signal processing with higher-order spectra*. IEEE Signal processing magazine, Vol. 10(3), S. 10-37, 1993.
- [4] B.-U. Köhler. *Konzepte der statistischen Signalverarbeitung*. Springer-Verlag, 2005.
- [5] S. Osowski, L.T. Hoai und T. Markiewicz. *Support vector machine-based expert system for reliable heartbeat recognition*. IEEE Transactions on Biomedical Engineering, Vol. 51(4), S. 582-589, 2004.
- [6] A. Swami, J. M. Mendel und Chrysostomos L. (Max) Nikias. *Higher-Order Spectral Analysis Toolbox: For Use with MATLAB®*, MathWorks Incorporated, 1998.
- [7] M. Lagerholm et al. *Clustering ECG complexes using Hermite functions and Self-organizing maps*. IEEE Transactions on Biomedical Engineering, Vol. 47(7), S. 838-848, Juli 2000.
- [8] S.-W. Lee und A. Verri. *Pattern Recognition with Support Vector Machines: First International Workshop, SVM 2002, Niagara Falls, Canada, August 10, 2002. Proceedings*, Vol. 2388, Springer-Verlag, 2003.
- [9] S. Abe. *Support vector machines for pattern classification*, Springer-Verlag, Vol. 2, 2005.

Tabelle 2: Die Ergebnisse der Mustererkennung von verschiedenen Typen von Herzschlägen durch die SVMs mit HOS als Merkmale, die SVMs mit HER als Merkmale, die Kombination dieser beiden Resultate sowie durch die SOM mit Verwendung von HER Merkmalen. Genauer, der **Vergleich der relativen Testfehler in Prozent (%)** dieser Methoden anhand von Testdaten jedes Typs für die SVMs und anhand aller Daten für die SOM.

Typen der Herzschläge	Anzahl der Testdaten	SVMs mit HOS	SVMs mit HER	SVMs mit HOS & HER	Anzahl der Daten	SOM mit HER
N	800	6.50	1.25	4.75	74948	3.19
L	240	11.67	8.75	5.26	8066	82.13
R	200	7.00	9.50	4.50	7244	40.38
A	180	14.44	22.22	12.22	2544	100.00
a	24	50.00	62.50	20.83	150	100.00
J	7	28.57	71.43	28.57	83	100.00
V	502	2.99	10.56	1.99	7122	33.14
F	128	25.78	18.75	18.75	802	100.00
I	76	34.21	44.74	18.42	472	100.00
e	1	100.00	100.00	100.00	16	100.00
j	37	48.65	67.57	29.73	229	100.00
E	17	17.65	11.77	0.00	106	100.00
P	200	6.00	9.00	0.00	7016	78.96
f	80	45.00	31.25	21.25	982	100.00
Q	3	100.00	100.00	100.00	33	100.00

Tabelle 3: **Vergleich der positiven Vorhersage Werte (Precision) und Vergleich der Sensitivität (Recall)** der SVMs unter Verwendung der HOS und HER Merkmale, deren Kombination zu einem Gesamtklassifizierungskonzept sowie der SOM unter Verwendung der HER Merkmale für jeden Typ von Herzschlag.

	Precision				Recall			
	SVMs mit HOS	SVMs mit HER	SVMs mit HOS & HER	SOM mit HER	SVMs mit HOS	SVMs mit HER	SVMs mit HOS & HER	SOM mit HER
N	0.9122	0.7611	0.9214	0.7654	0.9350	0.9875	0.9525	0.9407
L	0.9907	0.9910	0.9913	0.8688	0.8833	0.9125	0.9500	0.1749
R	0.9688	0.9945	0.9845	0.8037	0.9300	0.9050	0.9550	0.6578
A	0.9747	0.9722	0.9753	NaN	0.8556	0.7778	0.8778	0.0000
a	0.9231	0.9000	0.9048	NaN	0.5000	0.3750	0.7917	0.0000
J	0.8333	0.6667	0.8333	NaN	0.7143	0.2857	0.7143	0.0000
V	0.7550	0.9433	0.8693	0.6051	0.9701	0.8944	0.9801	0.6345
F	0.8120	0.9369	0.9369	NaN	0.7422	0.8125	0.8125	0.0000
I	0.9434	0.9767	0.9688	NaN	0.6579	0.5526	0.8158	0.0000
e	0.0000	0.0000	NaN	NaN	0.0000	0.0000	0.0000	0.0000
j	0.9048	0.9231	0.8966	NaN	0.5135	0.3243	0.7027	0.0000
E	1.0000	1.0000	1.0000	NaN	0.8235	0.8824	1.0000	0.0000
P	0.9741	0.9945	0.9852	0.5486	0.9400	0.9100	1.0000	0.2091
f	0.9362	1.0000	0.9692	NaN	0.5500	0.6875	0.7875	0.0000
Q	NaN	NaN	NaN	NaN	0.0000	0.0000	0.0000	0.0000

Design, Simulation and Operation of Task-oriented Multi-Robot Applications with MATLAB/Stateflow

Birger Freymann, Sven Pawletta, Artur Schmidt, Thorsten Pawletta

Hochschule Wismar - University of Applied Sciences, Research Group CEA, Wismar, Germany

{birger.freymann, sven.pawletta, artur.schmidt, thorsten.pawletta}@hs-wismar.de

Robot programming software is mostly proprietary and cannot be used for other manufacturers' robots. Nevertheless, there is a desire to allow interactions between robots being developed by different manufacturers in order to set up a *Multi-Robot System* (MRS). An MRS refers to a team consisting of interacting industrial robots which share skills to increase performance. The mapping of classical control development methods for *Single-Robot Systems* (SRSs) to MRSs is difficult. Based on a classification of interactions within MRSs, a *Task-Oriented Control* (TOC) approach is suggested and examined. Furthermore, with *Simulation Based Control* (SBC) an approach for continuous development of event-driven multi-robot controls according to the *Rapid Control Prototyping* (RCP) is presented. SBC supports TOC and the mapping of interactions via tasks. Based on SBC and MATLAB/Stateflow a prototypical, task-oriented model library for interacting robots is developed and tested.

1 Introduction

New fields for applied robotics are made up continuously, something which increases the need for research. For example, current research focus on mobile robotics, service robotics and human-robot interaction is referred to as *Collaborative Robots* (Cobots) [1]. In manufacturing, industrial robots are often used as flexible handling and production units.

In the context of Industry 4.0, there is a desire for intelligent and networking factories [2]. Among other things, these are characterized by a flexible and versatile production of individual parts up to mass production. A *Single-Robot System* (SRS) mostly does not have sufficient skills to meet these requirements. Therefore, *Multi-Robot Systems* (MRSs) are proposed as a key technology. In an MRS, the robot systems which are involved form a team by sharing skills in order to enhance performance. In this work, an MRS consists of commercially available SRSs with a shared central control computer.

In contrast to an SRS the involved robot systems for an MRS always influence each other, which is called interaction. One motivation for the use of an MRS is to increase the efficiency of the robot team, thus reducing the production time and costs. Nevertheless, the use of an MRS presents new challenges, because classical industrial robots and their software tools are usually not designed to be used in a team.

For an SRS, different design methods for control development are known [3,4] but the transfer to an MRS is difficult [5]. In particular, the use of robots from different manufactures represents a major challenge. Industrial robots are mostly programmed in proprietary development environments, which do not support third party robots. Some of these environments are already supporting a control development for MRSs, but usually only for small team sizes, a limited number of predefined interactions and exclusively for robots of the particular manufacture. Also the integration of sensors and actuators is usually restricted to vendor specific products.

Proprietary development environments are based on vendor-specific robot programming languages. Attempts to standardize robotic programming languages, such as the *Industrial Robot Language* (IRL) and its successor, the *Programming Language for Robots* (PLR), are usually ignored by robot manufactures. However, the ongoing spread of the *Robot Operating System* (ROS) [6], an open source project to provide cross-vendor robot programming and process visualization, shows that there is an unbroken interest in unified development methods and environments.

In the context of control development, different demands exist [7]. In this work the following issues will be considered in more detail: (i) flexible adaptation to new problems and quick commissioning (ii) testability, maintainability (iii) cross-vendor interactions between

robots and (iv) support of vendor-independent hardware and software.

In order to meet these demands, the software and conceptual foundation are presented. At first a non-vendor-specific control of industrial robots is shown, using the *Robotic Control & Visualization* (RCV) Toolbox for MATLAB as middleware with associated interface. Then the concept of Task-Oriented Control (TOC) development [4,8] is described for an SRS. For MRS control development, the *Simulation Based Control* (SBC) procedure model is proposed which follows the *Rapid Control Prototyping* (RCP) approach by Abel [9]. SBC enables continuous control development from the early planning phase to the operational control use and defines a framework for practical implementation. In addition, the investigations in [10–12] show that the SBC approach supports TOC. Furthermore a classification of interactions between industrial robots is introduced based on [13]. Based on a case study, the mapping of interactions to tasks is examined. The implementation of the SBC approach is realized in the MATLAB/Stateflow environment.

2 Fundamentals

In this section, the necessary basics for the development of a *Multi-Robot System* (MRS) control are explained. The methods described and the tools are already used for *Single-Robot Systems* (SRSs) and will be applied to MRSs.

2.1 Robotic Control & Visualization Toolbox for MATLAB/Simulink

As mentioned before, the programming of robotic systems is affected by vendor dependent languages and proprietary development environments. Existing standards are mostly ignored by manufacturers, which makes control development for multi-robot controls difficult or even impossible. In engineering, control development is often associated with Scientific and Technical Computing Environments (SCEs), such as MATLAB. To close the gap between vendor-specific robotic environments and common SCEs, the *Robotic Control & Visualization* (RCV) Toolbox for MATLAB/Simulink was developed [14,15]. The RCV Toolbox consists of three modules: (i) a set of generalized robot-oriented MATLAB functions, (ii) a set of MATLAB functions for process visualization and (iii) robot-specific command interpreters. The basic structure of an RCV based control is shown in Figure 1. The Control PC runs a MATLAB instance which processes

the control program on the basis of functions of the module (i). The Visualization PC runs another instance of MATLAB and visualizes the process based on functions of the module (ii). The communication with real robots is performed by the robot controllers and specific command interpreters according to the module (iii). A detailed description of the RCV Toolbox can be found in [16].

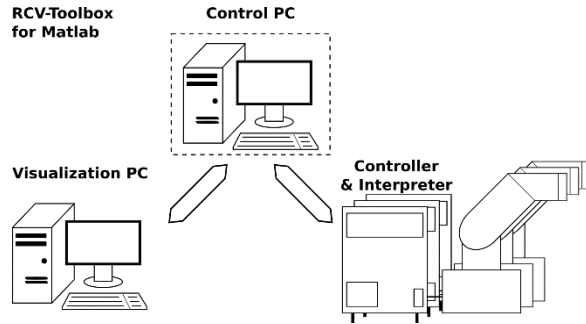


Figure 1: Principle structure of RCV based multi-robot application

It should be noted that the use of an SCE, such as MATLAB, simplifies the integration of additional sensors and actuators to the control.

2.2 Task-oriented Control Design

Task-Oriented Control (TOC) design [4,8] is already known for SRSs [17,18]. The principle is to divide the complex control problems into a set of tasks and their couplings. Tasks are logical, mostly independent, abstract operations. Once identified, tasks are coupled together to map the control problem. Couplings can be carried out sequentially, conditionally or in a loop. Some problem descriptions mean it is necessary to use tasks multiple times or to work them off in parallel. The general procedure corresponds to the human way of thinking in solving complex problems.

Figure 2 shows the TOC approach on the example of a transportation problem. The problem can be divided (decomposition) into the two tasks *PickPart* and *PlacePart*.

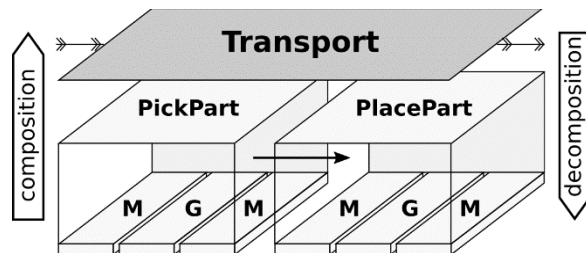


Figure 2: Tasks and subtasks of transport problem

The task *PickPart* specifies the grasping of a component at a defined position and the task *PlacePart* corresponds to placement at a new location. The task *PickPart* has to be executed before the task *PlacePart*, which corresponds to a serial execution. Furthermore, the figure shows that a task can be composed of other (sub-) tasks. For example, the task *PickPart* can be described by the serial linking of the tasks *Move* (M), *GripperAction* (G) and *Move* (M). According to [10], TOCs can be realized using *Top Down* (decomposition) or *Bottom Up* (composition) principles.

A TOC specification is not directly executable, because tasks are an abstract description of operations. A task describes ‘what’ but not ‘how’ something has to be solved. To perform tasks, a transformation method is required, as shown schematically in Figure 3. All tasks have to be transformed into control commands in order to execute them. To perform the transformation it is either possible to use a world model or a process model.

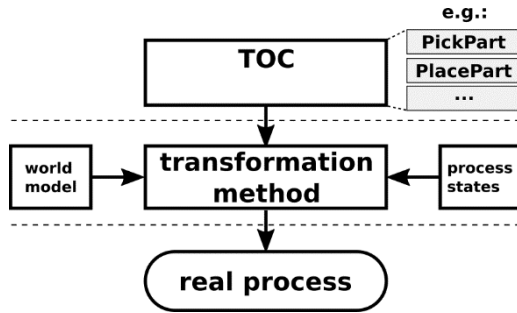


Figure 3: Processing of a TOC according to [8]

Early TOCs used a compilation approach which produces an executable control program in advance. The transformation was carried out before the commissioning by using a world model. More recent works [8,17] understand the transformation step as a function of the operational control. This has the advantage that current process states can be used instead of the world model. Thus, reactive controls can be realized, which can flexibly adapt to current process states and, for example, respond to faults by alternative processing of individual tasks. Here, the task transformation has to keep pace with the process dynamics.

Another extension of classical TOC is parameterized tasks. These allow the grouping of similar tasks into a common task. Thus, the reusability in repositories increases which also reduces development times, costs and errors.

2.3 Simulation Based Control Approach

The *Simulation Based Control* (SBC) approach [11] is a methodology for control development and defines a framework for practical implementation. The main goal of SBC is to provide a systematic way for *Rapid Control Prototyping* (RCP) [9].

Accordingly, a gradual and consistent model-based control development from the early planning phase to real operational use is supported (Figure 4). The control designs can be successively tested by simulation and supplemented by additional requirements. To enable this form of step-wise control development, an appropriate software environment or tool chain is required.

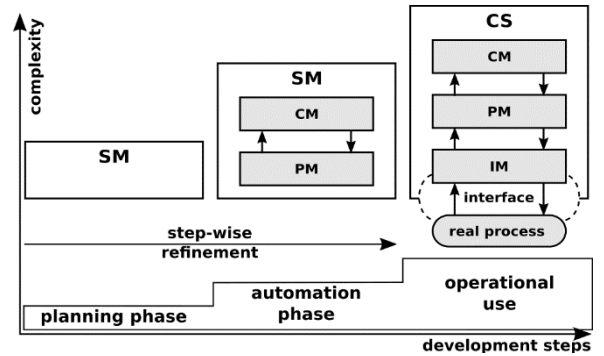


Figure 4: The SBC approach

The consequent usage of simulation in all phases of the development process is key for detecting design errors as early as possible. The *Simulation Models* (SMs) are step-wise refined. During the transition from planning to the automation phase, a separation of the SM into a *Control Model* (CM) and a *Process Model* (PM) is introduced. Within the automation phase the SM is completed by an *Interface Model* (IM). In this stage the SM can be used for system simulation (in conjunction with an appropriate physical process model), or for *Software-in-the-Loop* (SiL) simulation.

During the transition from the automation to the operational phase, the SM becomes real *Control Software* (CS). This transition is usually known as Code Generation. Depending on the real-time requirements of the control application, the SBC approach distinguishes between explicit and implicit code generation. The first type is the classical method for high real-time requirements in conjunction with mostly embedded controller hardware. In this case, the explicit code generation is done by an appropriate compiler. For applications with rather slow timing, implicit code generation

is suitable. That means the last SM of the design process is used as CS without modification. This is possible due to the IM. It provides a process interface enabling SiL but also operational use.

Following the SBC approach the CS always includes a PM. Thus, observer concepts can easily be realized. In the case of poorly or not measurable process states, the PM can provide at least estimated values.

In [11,17] the SBC-Framework has been successfully applied for SRS control development for different manufacturers. Furthermore, it is shown that SBC supports a task-oriented control specification (Figure 5).

In this case, the CM contains the task-oriented control specification. The PM still has a component-oriented structure according to the elements of the real process. With a task-oriented CM on top, the PM is also the place of task transformation as described in Section 2.2. The tasks are mapped onto the generalized commands of the RCV Toolbox (Section 2.1). Thus, the PM remains independent according to concrete robots. The vendor-specific mapping is subject of the IM.

Below, the mapping of interactions in MRS to a task-based control specification using the SBC framework is shown after, the term interaction is discussed.

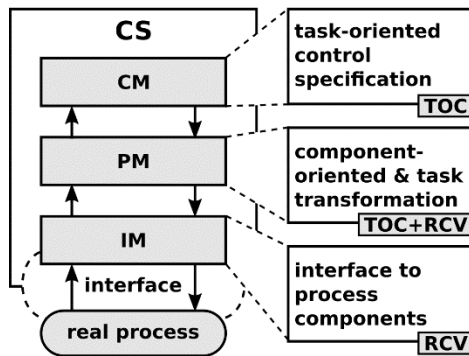


Figure 5: Use of TOC and RCV Toolbox within the SBC Framework

3 Interactions in MRS

In the previous sections, methods and tools for control design of *Single-Robot Systems* (SRSs) were introduced, which should be applied to *Multi-Robot Systems* (MRSs). A fundamental element of an MRS is interactions. This section treats characteristics and classification of interactions in MRSs.

3.1 Discussion

In MRSs, the robots usually support each other in order to improve the team performance. The term interaction

describes the mutual influence of the robots. An interaction often requires the sharing of a limited resource, such as a robot's workspace (Figure 6). For example, if two robots need to pass a workpiece, their workspaces must overlap. This implies that the robots have to coordinate the handover to avoid a collision. One possibility for realizing the coordination is to exchange information between the robot systems.

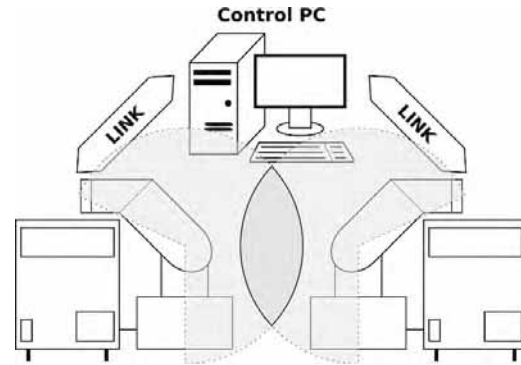


Figure 6: Example of an MRS with two industrial robots

3.2 Classification

Based on the general classification in [13], interactions of industrial robots can be divided into six classes. Figure 7 shows the example of a transportation problem for better illustration. Parts have to be transported from an *Input Buffer* to an *Output Buffer*. *Class 0* starts with an SRS and hence involves no interaction. *Class 1* to *Class 6* solve the same problem with an MRS. The complexity of the transportation problem can increase from one class to another class. For example, new part types are introduced with special requirements. To solve the problem, the necessary level of interaction will increase too.

Class 0: One type of part has to be transported by an SRS. The robot's tool is adapted to the part. The task is completed when all parts have been transported from the *Input Buffer* to the *Output Buffer*. There is no interaction.

Class 1: An MRS consisting of two robots (R_1, R_2) with separate workspaces. Both robots have identical tools. No exchange of information between the robots is required. The interaction refers to the collective solution of a problem by two or more robots.

Class 2: As for *Class 1*, supplemented by a new type of part, which requires the exchange of one robot tool. Regarding the interaction there is no change compared to *Class 1*. The robots still solve one problem and have

separate workspaces. Therefore, *Class 1* and *Class 2* have the same degree of interaction.

Class 3: As for *Class 2*, but the robots share a common workspace. To avoid collisions the robots have to coordinate their motions. Therefore, the degree of interaction is increased compared with *Classes 1 & 2*.

Class 4: As for *Class 3*, but another type of part leads to a stronger interaction. The new type can only be moved by the two robots together. Thus, the degree of interaction is increased compared to *Class 3*.

Class 5: One robot supports the other one, even if the tool is not ideal for this purpose. This form of interaction is used to compensate for the overload of one robot by irregular arrival of parts.

Class 6: A new type of part which cannot be handled by the robot team requires the replacement of a robot or the tool. Interaction refers to the modification of team members.

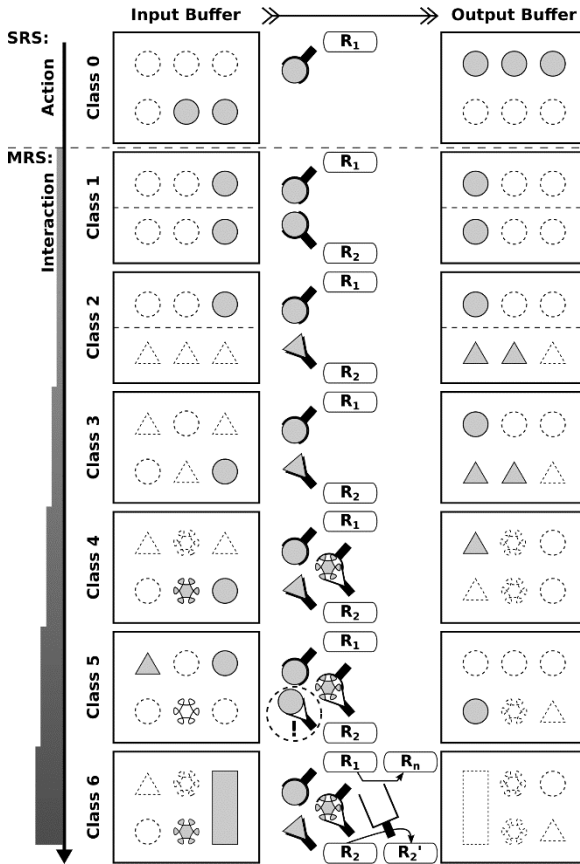


Figure 7: Classification of interacting robots based on [13] – with level of interaction

4 Case Study

In the following, the control development for a *Multi-Robot System* (MRS) is examined by a handling and assembly problem. Different interactions are implemented using a task-oriented control design. Based on the above described methods and tools, the control will be developed in the MATLAB/Stateflow environment.

4.1 Layout and Workflow

The experimental setup is illustrated in Figure 8. It consists of two robots (R_1 , R_2) and two cameras (C_1 , C_2). Each robot has a separate *Input* (I_1 , I_2) and *Output Buffer* (O_1 , O_2) in its workspace. Furthermore, both robots share a portion of the workspace by a *Common Output Buffer* O_C . The two *Input Buffers* are externally stocked up with parts of type A or B.

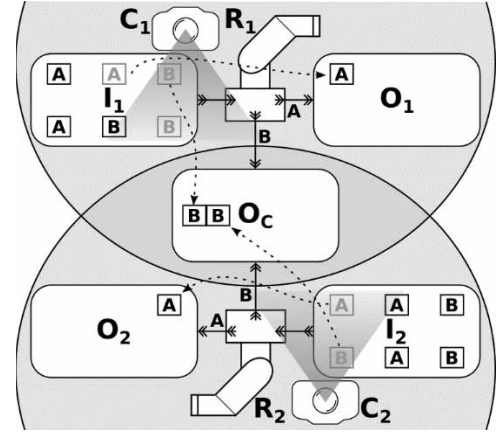


Figure 8: Experimental setup of the case study

Each robot randomly picks up a part of its respective *Input Buffer*. The position of the parts is assumed to be known. The gripped part can either be of type A or B which makes identification by a corresponding camera necessary. The result of the identification dictates how to proceed with the part. If the gripped part is of type A, it can be directly placed in the *Output Buffer* of the robot. Then the robot picks up another part from the *Input Buffer* and the procedure is repeated until a part of type B is identified. Two parts of type B always have to be assembled with each other by both robots before they are stored in O_C .

If a robot picks up a part of type B, he moves to a mounting position above O_C and is blocked then (field a) until the second robot also picks up a part of type B and reaches its mounting position (field b). Both robots proceed with a synchronous motion (field c) in order to place both parts in O_C (click mounting, field d). Subsequently, a new cycle takes place.

Figure 9 shows the subtasks which have to be done to solve the task *CommonPlacePart* (CPP). If a robot picks up a part of type B, he moves to a mounting position above O_c and is blocked then (field a) until the second robot also picks up a part of type B and reaches its mounting position (field b). Both robots proceed with a synchronous motion (field c) in order to place both parts in O_c (click mounting, field d). Subsequently, a new cycle takes place.

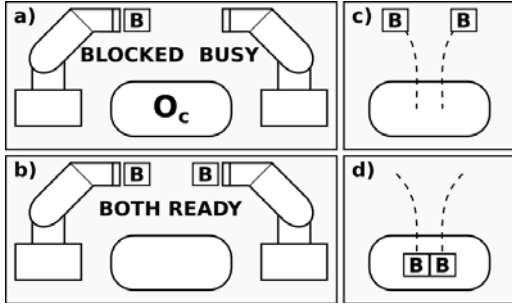


Figure 9: Workflow of two robots doing task *CommonPlacePart* (CPP)

4.2 Task-oriented Analysis

A task-oriented description of the workflow is shown in Figure 10. The notation is based on Harel statecharts. The figure shows the first step of the task-oriented control design. The problem description is done using the tasks *PickPart*, *IdentPart*, *PlacePart*, *MoveToPos* and *CommonPlacePart* (CPP). The control logic is identical for both robots and can be mapped by two identical parallel states (AND states), drawn as dashed lines.

As long as no part of type B is identified, both robots work independently and in parallel. The task sequence is: (i) grip a part (*PickPart*), (ii) identify the gripped part with the camera (*IdentPart*) and (iii) place the part in the *Output Buffer* (*PlacePart*).

If a part of type B is identified, the task sequence changes. Instead of *PlacePart*, the task *MoveToPos* will be executed. In consequence, the robot moves to its mounting position above O_c . The subsequent task *CPP* blocks the robot (Figure 9a) until the second robot also executes the task *CPP* (Figure 9b). Both robots coordinate the motions until the task *CPP* ends (Figure 9c, d).

The independent handling of type A parts by a sequence of tasks corresponds to a *Class 1* interaction. The interaction can be modeled by parallel execution of task sequences. The mapping of the assembly process of type B parts corresponds to a *Class 4* interaction because it necessitates the coordination of both robots. The necessary time synchronization and coordination is mapped in the form of a separate task (CPP).

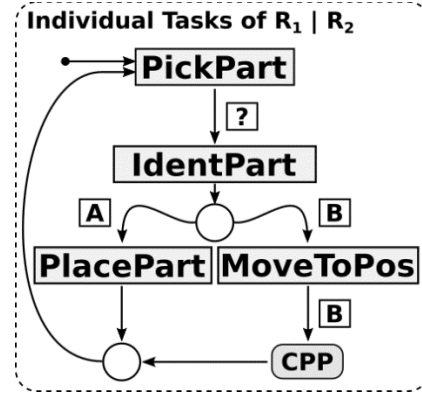


Figure 10: Identical Task Composition for both Robots

4.3 Implementation Aspects

The implementation of the control is based on the *Simulation Based Control* (SBC) approach and realized in the MATLAB/Stateflow environment. In accordance with SBC, the control consists of a *Control Model*

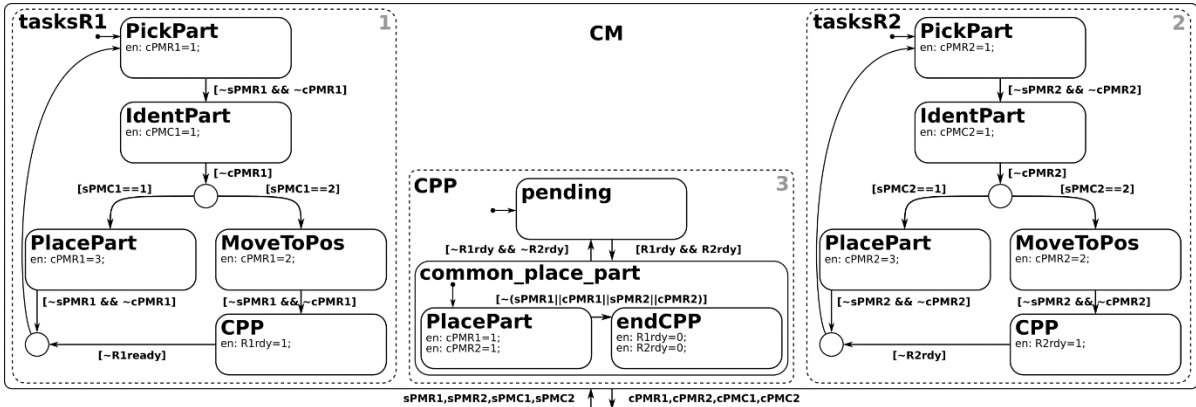


Figure 11: Statechart of CM based on SBC Framework

(CM), *Process Model* (PM) and *Interface Model* (IM) (Figure 4, 5). The CM shown in Figure 11 implements the *Task-Oriented Control* (TOC) logic according to Figure 10. The task sequences of both robots are implemented as parallel states. The task *CPP* in both parallel states (*tasksR1*, *tasksR2*) is signaling a standby for assembling. The implementation of the necessary interaction Figure 9c, d) was modeled as a separate task *CPP* in another parallel state. Thus, the interaction principle is mapped to a common, reusable task.

The PM is modeled analogously to fulfill the requirements of the SBC and the TOC and is illustrated in Figure 12. Each process component is represented as a parallel state and with two substates, *idle* and *execTask*. The state *idle* indicates that the component is on standby. The state *execTask* implements the process depending on task transformation. Because the task transformation is more clearly implemented procedurally it has been transferred to a MATLAB function call. The input parameter is a numeric value and represents the decoded task.

The IM, shown in Figure 13, is realized analogously to the PM. It defines a parallel state with two sub-states for each process component, which needs to be controlled. Communication with the real process components is implemented, in each case, as a MATLAB function. To control the robot, the function *execTask()* is used which is based on the *Robotic Control & Visualization* (RCV) Toolbox (Section 2.1). For the control of other peripherals, such as the cameras, MATLABs *Instrumental Control Toolbox* [19] can be used.

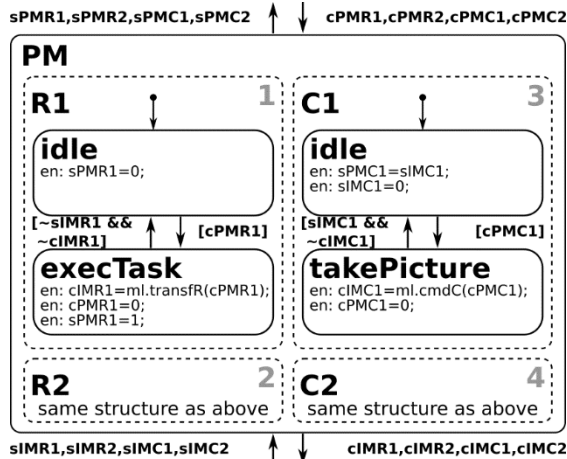


Figure 12: Statechart of PM based on SBC Framework

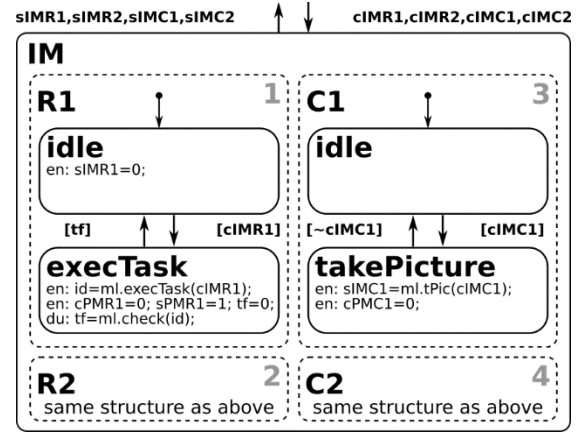


Figure 13: Statechart of IM based on SBC Framework

The SBC based control development in the MATLAB/Stateflow environment supports a gradual, simulation-assisted development of control software.

5 Summary and Outlook

Based on methods and tools, which are established in the control development for *Single-Robot Systems* (SRSs), an approach for *Multi-Robot Systems* (MRSs) was developed. In contrast to SRSs, interactions must be considered in MRSs. For this purpose, a characterization and classification of interactions in MRSs was introduced. An example demonstrating two interaction classes showed how interactions can be modeled as a general, reusable task, according to the task-oriented robot control paradigm.

The prototypical implementation was based on the *Simulation Based Control* (SBC) approach in the MATLAB/Stateflow environment and used the *Robotic Control & Visualization* (RCV) Toolbox for MATLAB. It was demonstrated that the control development of an MRS can be realized vendor independently, analog to an SRS and with simulative tests throughout the control development cycle.

At the beginning, defined requirements are largely fulfilled by prototypical implementation. The next step will be to examine if other identified interaction classes can be mapped to reusable tasks. Additionally, a proof of concept for a more advanced application will be provided in the future.

6 References

- [1] M. Bélanger-Barrette. *Robotiq: Collaborative Robot EBook*, available at <http://blog.robotiq.com/collaborative-robot-ebook> (accessed on April 19, 2016).
- [2] M. Hermann, T. Pentek, B. Otto. *Design Principles for Industrie 4.0 Scenarios: A Literature Review*, available at http://www.snom.mb.tu-dortmund.de/cms/de/forschung/Arbeitsbereiche/Design-Principles-for-Industrie-4_0-Scenarios.pdf (accessed on April 14, 2016).
- [3] W. Jacak. *Intelligent robotic systems: Design, planning and control*, Kluwer Academic, New York, London, 1999.
- [4] B. Siciliano. *Springer handbook of robotics*, Springer, Berlin u.a., 2008.
- [5] S. Makris, G. Michalos, A. Eytan, G. Chryssolouris. *Cooperating Robots for Reconfigurable Assembly Operations: Review and Challenges*, Procedia CIRP 3 (2012) 346–351.
- [6] Open Source Robotics Foundation. *Robot Operating System*, available at <http://www.ros.org/> (accessed on April 14, 2016).
- [7] H. Meier, *Verteilte kooperative Steuerung maschinennaher Abläufe*, Dissertation. TU München, Munic, 2001.
- [8] W. Weber, *Industrieroboter: Methoden der Steuerung und Regelung*, Carl-Hanser, München, 2009.
- [9] D. Abel, A. Bollig, *Rapid Control Prototyping, Methoden und Anwendung*, Springer, 2006.
- [10] G. Maletzki, *Rapid Control Prototyping komplexer und flexibler Robotersteuerungen auf Basis des SBC-Ansatzes*. Dissertation, Rostock, 2014.
- [11] T. Pawletta, S. Pawletta, G. Maletzki, *Integrated Modeling, Simulation and Operation of High Flexible Discrete Event Controls*, MATHMOD 09 6th Vienna Conference on Mathematical Modelling (2009).
- [12] T. Schwatinski, T. Pawletta, S. Pawletta, C. Kaiser, *Simulation-based development and operation of controls on the basis of the DEVS formalism*, Prague, Czech, 2010.
- [13] T. Lüth, T. Längle, *Multi-Agenten-Systeme in der Robotik und Artificial-Life*, in: GMD Workshop, Sankt Augustin, 1995.
- [14] M. Christern, A. Schmidt, T. Schwatinski, T. Pawletta, *KUKA-KAWASAKI-Robotic Toolbox for Matlab*, 2011, available at http://www.mb.hs-wismar.de/cea/KK_Robotic_Tbx/KK_Robotic_Tbx.html (accessed on May 15, 2015).
- [15] J. Otto, T. Schwatinski, T. Pawletta, *KUKA-KAWASAKI-Visualization Toolbox for Matlab*, 2011, available at http://www.mb.hs-wismar.de/cea/MatlabKK_Robotic-and-Visualization_Tbx/MatlabKK_Robotic_Visualization_Tbx.html (accessed on May 15, 2015).
- [16] C. Deatcu, B. Freymann, A. Schmidt, T. Pawletta, *MATLAB/Simulink Based Rapid Control Prototyping for Multivendor Robot Applications*, 25th ed., SNE - Simulation Notes Europe, 2015.
- [17] T. Schwatinski, T. Pawletta, S. Pawletta, *Flexible Task Oriented Robot Controls Using the System Entity Structure and Model Base Approach*, SNE - Simulation Notes Europe, ARGESIM/ASIM Pub. TU Vienna 22 (2012) 107–114.
- [18] T. Pawletta, S. Pawletta, G. Maletzki, *Integrated Modeling, Simulation and Operation of High Flexible Discrete Event Controls*, Argesim Report, Vienna, Austria, 2010.
- [19] The MathWorks, *Instrument Control Toolbox*, available at <http://de.mathworks.com/products/instrument/> (accessed on April 22, 2016).

Simulation von Strahlströmungen mittels des SST-Turbulenzmodells

Ansgar Ringleb¹, Oliver Sommer², Wolfgang Schlüter¹, Günter Wozniak²

¹ Fakultät Ingenieurwissenschaften, Hochschule Ansbach, Germany

² Institut für Mechanik und Thermodynamik, Technische Universität Chemnitz, Germany

ansgar.ringleb@hs-ansbach.de

Strahlströmungen stellen die Grundlage vieler technischer Strömungen dar, bspw. der Strahl eines Gasbrenners in Industrieöfen oder die Strahlen eines chemischen Reaktors. Forschungen zur Auslegung und Effizienz ebenso wie zur Verbesserung der Berechnungsmethoden befinden sich stets im Fokus von Industrie und Wissenschaft. Die vorliegende Untersuchung widmet sich der numerischen Simulation solcher Strahlströmungen. Als grundlegende Strömung wird der runde Freistrah und darauf aufbauend der Einfluss einer begrenzenden Wand untersucht. Als komplexere Strahlanordnung wird die Strömung eines hexagonalen Strahlbündels in einem Rohr simuliert. Dazu wird die Anwendung der Reynolds-averaged Navier-Stokes Gleichungen (RANS) und des Shear-Stress-Transport Turbulenzmodells (SST) untersucht. Es wird gezeigt, dass sich die Qualität der Simulationsergebnisse durch Anpassung der Modellkoeffizienten signifikant erhöhen lässt. Abschließend wird eine Diskussion zur Eignung und den Grenzen der angewendeten Modellierung geführt.

1 Einleitung

Die numerische Simulation von Strahlströmungen, z.B. unter Einfluss einer begrenzenden Wand und unter Einsatz der Reynolds-averaged Navier-Stokes Gleichungen (RANS) ist eine häufig anzutreffende Aufgabenstellung. Die Verbesserung der dabei verwendeten Turbulenzmodellierung ist Gegenstand verschiedener wissenschaftlicher Untersuchungen, z.B. [1], [2] und [3]. Häufig interessieren dabei zeitgemittelte Strömungsgrößen, bspw. die Entwicklung der Strahlgeschwindigkeit entlang der Strahlachse $U_0(x)$ (engl.: center line velocity decay, CLVD). Zur tiefergehenden Untersuchung turbulenter Wirbelstrukturen, von Interesse z.B. bei Mischungsvorgängen, ist u.a. der Einsatz der Large-Eddy-Simulation (LES) Stand der Technik [4]. Im Vergleich zu RANS-Simulationen bietet diese Methode i.d.R. Vorteile bei der Berechnung instationärer Strömungen oder Strömungen mit anisotroper Turbulenz, allerdings zu Lasten eines erheblich höheren Rechenaufwands, welcher bis zu einem Faktor von 10^6 betragen kann [5]. Aus diesem Grund stellt die Weiterentwicklung RANS-basierender Turbulenzmodelle für die Simulation von Strahlströmungen ein großes Potential sowohl für die wissenschaftliche Forschung als auch für die technische Anwendung dar.

In der vorliegenden Untersuchung werden drei Strahlströmungen betrachtet: der Freistrah, der be-

grenzte Strahl und das hexagonale Strahlbündel (Abbildung 1).

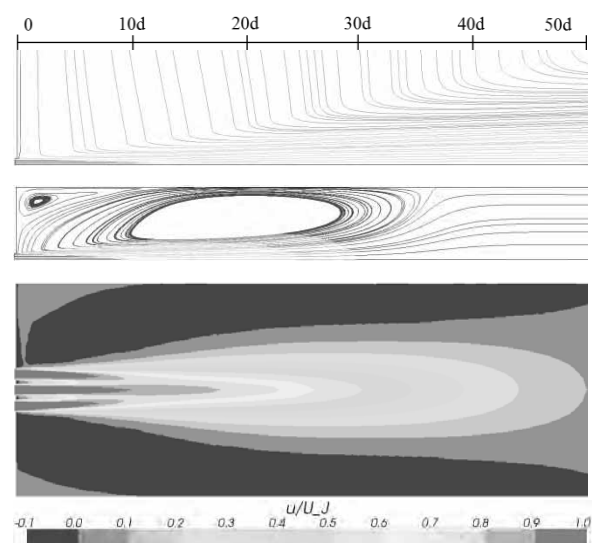


Abbildung 1: Stromlinien-Darstellung zum Freistrah (oben), begrenzten Strahl (Mitte) und Geschwindigkeitsfeld des Strahlbündels (unten)

Im ersten Fall strömt das Fluid aus einer Düse mit kreisrunden Querschnitt in den freien Strömungsraum. Aufgrund der Düse wird hierbei ein sog. Block-Profil der Axialgeschwindigkeit mit sehr dünner Scherschicht im Randbereich angenommen. Dazu wird die Strömungsgeschwindigkeit im Düsenquerschnitt U_j so gewählt, dass sich eine auf den Düsen-

durchmesser d bezogene Reynoldszahl von $Re_d = 11.000$ ergibt. (Die ebenfalls auf den Düsenquerschnitt bezogene Mach-Zahl beträgt $Ma_d = 0,18$. Somit kann die Strömung als inkompressibel behandelt werden.) Im zweiten Fall wird um den Strahl rotationssymmetrisch eine begrenzende Wand (Rohr) mit dem Durchmesser D angeordnet, weshalb die Bezeichnung des begrenzten Strahls eingeführt werden soll. Diese Strömung entspricht bei kleinen Durchmesserverhältnissen D/d der axialsymmetrischen rückspringenden Stufenströmung (engl.: round sudden expansion). Für diesen Fall beträgt die Reynoldszahl $Re_d = 10^4$ und die Mach-Zahl $Ma_d = 0,02$. Im dritten Fall, dem hexagonalen Strahlbündel, werden um einen zentralen Strahl achssymmetrisch sechs Strahlen hexagonal angeordnet. Das so gebildete Strahlbündel mündet ebenfalls in einen rotationssymmetrisch begrenzten Strömungsraum (Rohr). Aufgrund der hexagonalen Anordnung ergibt sich ein einheitlicher Abstand A zwischen den Strahlen. Die Reynoldszahl beträgt ebenfalls $Re_d = 10^4$ ($Ma_d = 0,02$). Aus versuchstechnischen Gründen wird in den Düsenmündungen ein turbulentes Rohrprofil angenommen.

2 Modellierung und Simulation

Der runde Freistrah und begrenzte Strahl weisen geometrisch und für zeitgemittelte Strömungsgrößen eine Symmetrie bzgl. der Strahlachse auf, weshalb ein rotationssymmetrisches 2D-Modell angewendet werden kann (Abbildung 2 oben). Das Strahlbündel stellt hingegen eine 3-dimensionale Strömung dar, welche ein 3D-Modell erfordert (Abbildung 2 unten).

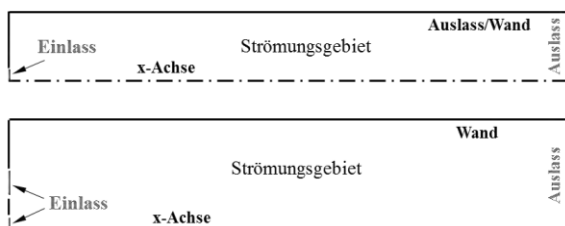


Abbildung 2: Schnittansicht des Strömungsgebietes für den Freistrah und begrenzten Strahl (oben) sowie das Strahlbündel (unten)

2.1 Rechengebiet und Vernetzung

Zur räumlichen Diskretisierung des Rechengebietes wird ein hybrides kartesisches Gitter mit hexaedrischer Vernetzung in Wandnähe und uniformer kartesischer Vernetzung im Strömungsraum verwendet. Zur Beurteilung der Netzsensitivität der Simulation wur-

den je Strömung 3 Gitter herangezogen, wobei sich die Gitterweite jeweils halbiert. Die nachfolgend dargestellten Ergebnisse beziehen sich auf die Lösung des mittleren Gitters, wobei die Scherschicht des Mündungsbereiches der Düse mit einer Gitterweite von $\Delta x = d/16$ aufgelöst ist (Abbildung 3). Weiter stromab ist der Strahlbereich in bis zu 3 Zonen mit lokaler Verfeinerung aufgelöst, wobei die Gitterweite sich schrittweise verdoppelt. Das Strömungsgebiet im Rohr ist somit mit einer Gitterweite von $\Delta x = d$ aufgelöst.

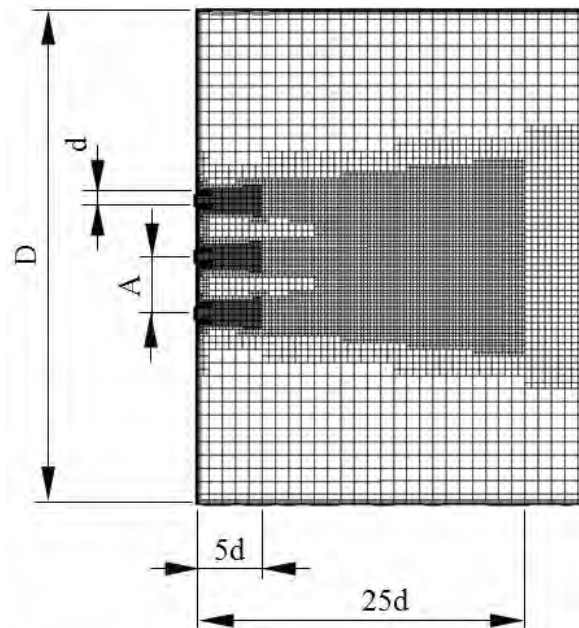


Abbildung 3: Schnittansicht des Strahlbündel-Gitters

2.2 Solver

Zur Approximation der RANS-Gleichungen wird die Finite-Volumen-Methode (FVM) genutzt. Die Konvektionsterme der Transportgleichungen für u , v , w , k und ω werden nach 2. Ordnung, die Diffusionsterme mit Hilfe der 2. Gradienten (engl.: cross diffusion) diskretisiert. Auf das resultierende gekoppelte Gleichungssystem wird eine sequentielle Lösungsmethode (engl.: segregated solver) angewendet. Weiter werden ein Unter-Relaxationsverfahren sowie eine algebraische Multigrid-Methode zur Steigerung der Lösungsstabilität eingesetzt. Das so linearisierte Gleichungssystem wird durch Anwendung des Gauss-Seidel-Verfahrens gelöst.

Die Strömungen des Freistrahls und des begrenzten Strahls können stationär gelöst werden. Für das Strahlbündel ergeben sich instationäre Lösungen. Zur Berechnung der instationären Terme wird eine Drei-

Ebenen-Methode 2. Ordnung angewendet, die Berechnung der Zeitwerte der inneren Iterationen erfolgt durch quadratische Rückwärtsapproximation, wozu die implizite Euler-Methode herangezogen wird.

2.3 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 zwei gekoppelte partielle Differentialgleichungen zur Schließung der Impulsgleichungen bereitstellt. Dies sind Transportgleichungen für die turbulente kinetische Energie k (1) und die turbulente Wirbelfrequenz ω (2).

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

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

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 der Dissipationsterme β und β^* sowie des Diffusionsterms σ_{k2} angepasst. Der Koeffizient des Produktionsterms γ ergibt sich mit $\kappa = 0,41$ gemäß (5) aus der Anpassung der genannten Koeffizienten. Die Werte des Standard-Modells können Tabelle 1 entnommen werden.

$$\gamma_2 = \frac{\beta_2}{\beta^*} - \sigma_{\omega 2} \frac{\kappa^2}{\sqrt{\beta^*}} \quad (5)$$

3 Ergebnisse

Wie bereits erwähnt, lässt sich die Ergebnisqualität der CFD-Simulationen signifikant durch eine entsprechende Anpassung der Modellkoeffizienten verbessern. Um dies zu bewerten werden die Simulationsergebnisse mit Messdaten aus eigenen Experimenten verglichen. Dazu wurde die Axialgeschwindigkeit u sowie deren Schwankungswert u' mittels Laser-Doppler-Anemometrie (LDA) gemessen. Die ange-

wendete Messmethode wurde anhand des Freistrahls und des begrenzten Strahls ($D/d = 2,0$) mit den Messdaten von [7] und [8] validiert, wobei die Messabweichung i.d.R. kleiner 5% beträgt. Die maximale Messabweichung beträgt bzgl. des Mittelwertes \bar{u}/U_j kleiner 20 % und bzgl. der Varianz u'^2/U_j^2 kleiner 30 %. Zum Vergleich werden je Strömungsfall die Simulationsergebnisse mit den Standard-Koeffizienten (SST) und den optimierten Koeffizienten (SST-opt 1 bzw. SST-opt 2) mit den Messdaten verglichen.

Wie bereits in vorangegangenen Untersuchungen [9] und [10] gezeigt wurde, besitzt der Koeffizient des Dissipationsterms β^* einen starken Einfluss. Im Strahlkern wird eine zu geringe turbulente kinetische Energie berechnet, was die Entwicklung der Strahlgeschwindigkeit weiter stromabwärts beeinflusst. Durch Anpassung von β^* kann dieser Mangel beim Freistrah behoben werden, so dass sich eine sehr gute Übereinstimmung zwischen Simulation und Experiment ergibt (Abbildung 4).

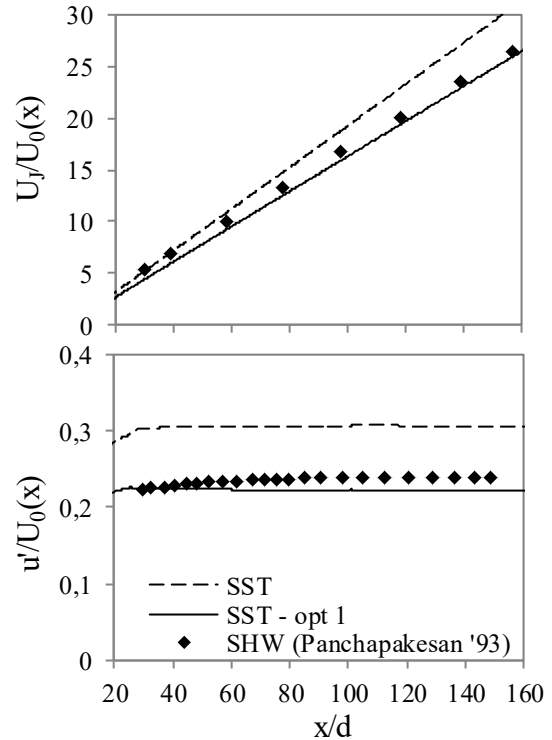


Abbildung 4: Einfluss der Modell-Koeffizienten auf die Axialgeschwindigkeit entlang der Strahlachse des Freistrahls ($Re_d = 11.000$): Axialgeschwindigkeit (oben) und Längsfluktuationen (unten)

Für den Fall des begrenzten Strahls führt die Anpassung von β^* ebenfalls zu einer deutlichen Verbesse-

rung. Die zusätzliche Anpassung des Produktionskoeffizienten β_2 und des Diffusionskoeffizienten σ_{k2} bewirkt eine weitere Ergebnisverbesserung. So ergibt sich hinsichtlich des Mittelwertes der Axialgeschwindigkeit ein nahezu deckungsgleicher Verlauf zwischen Simulation und Experiment (Abbildung 5 oben). Anhand der Standardabweichung (Abbildung 5 unten) ist gut zu erkennen, dass durch die Koeffizientenanpassung der gewissermaßen „systematische“ Fehler der Simulation – die Berechnung einer zu geringen turbulenten kinetischen Energie im Strahlkern – nahezu vollständig kompensiert wird. So zeigt sich auch hinsichtlich der Längsfluktuationen eine sehr gute Übereinstimmung zwischen Simulation und Experiment.

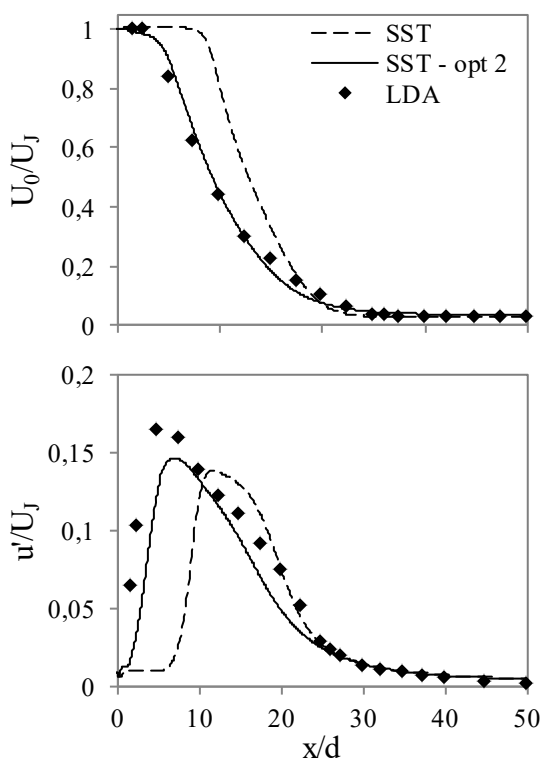


Abbildung 5: Einfluss der Modell-Koeffizienten auf die Axialgeschwindigkeit entlang der Strahlachse des begrenzten Strahls ($D/d = 6,2$; $Re_d = 10^4$): Axialgeschwindigkeit (oben) und Längsfluktuationen (unten)

Die Strömung des Strahlbündels ist wesentlich komplexer als die zuvor untersuchten Strömungen des Freistrahls und des begrenzten Strahls. So wirkt nicht nur die Scherschicht im mündungsnahen Bereich oder die turbulente Strömung im Fernfeld des Freistrahls sowie der Einfluss der Rezirkulation und Wandgrenschicht des begrenzten Strahls. Zusätzlich dazu wirkt

zwischen den Strahlen eine Anziehungskraft, wenn der Strahlabstand A klein genug ist, so dass es zur Ausbildung eines lokalen Unterdruckgebietes kommt. Dabei werden die Strahlen voneinander angezogen, so dass sich ein stabiles Strahlbündel bildet. Zusätzlich wirkt der Einfluss der umgebenden Wand auf die einzelnen Strahlen, kommt es zur Ausbildung eines lokalen Unterdruckgebietes zwischen Strahl und Wand, sind die äußeren Strahlen bestrebt sich an die Wand anzulegen. Die betrachtete Strömung weist einen Strahlabstand von $A/d = 1,4$ und einem Durchmesser Verhältnis von $D/d = 19,0$ auf. Dabei überwiegt der Einfluss der Strahlen den der Wand, so dass sich ein stabiles Strahlbündel ausbildet, dessen Strömung der des begrenzten Strahls ähnelt.

Wendet man zu dessen Simulation die für den begrenzten Strahl optimierten Koeffizienten an (SST - opt 2), so zeigt sich ein signifikanter Einfluss auf das Simulationsergebnis. Im vorderen Strahlbereich ($x/d < 40$), wo sich die Strahlen zu einem Strahlbündel zusammenschließen, werden die lokalen Maxima und Minima des Turbulenzgrades (Standardabweichung der instationären Schwankungen $\sigma(u)$ und turbulenten Schwankungen u'^2) durch die optimierten Koeffizienten gut wiedergegeben (Abbildung 6 unten). Auch weiter stromabwärts ($x/d > 40$) werden die Auflösung des Strahlbündels, der Abklingprozess der turbulenten Schwankungen und der Übergang zur Rohrströmung durch die Simulation gut abgebildet. Der zeitliche Mittelwert stellt sich im Experiment jedoch anders dar als in der Simulation (Abbildung 6 oben). Mit Standard-Koeffizienten wird der vordere Strahlbereich (Bildung des Strahlbündels) nur ungenau wiedergegeben, wobei sich die Geschwindigkeitsabnahme geringer darstellt. Das Ende des Strahlbündels wird durch einen stufenförmigen Verlauf bei etwa $x/d = 30$ markiert. Der darauf folgende Übergang zur Rohrströmung wird durch die Simulation mit einer größeren Steigung berechnet als im Experiment zu beobachten ist. Die Lauflänge bis zur Ausbildung der Rohrströmung bei $x/d = 70$ wird wiederum in guter Näherung zum Experiment berechnet.

Mit optimierten Koeffizienten wird ein ähnlicher Verlauf der Axialgeschwindigkeit berechnet. Im vorderen Strahlbereich ergibt sich eine bessere Übereinstimmung zum Experiment, da nun die Abnahme der Geschwindigkeit erheblich schneller erfolgt. Jedoch wird dadurch auch der hintere Strahlbereich beeinflusst, sodass sich ein deutlich früherer Übergang zur Rohrströmung bei etwa $x/d = 40$ ergibt.

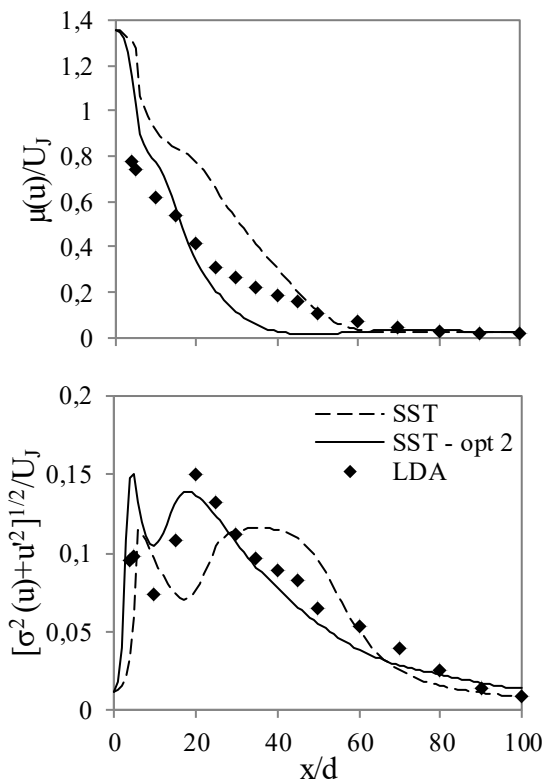


Abbildung 6: Vergleich von Simulation und Experiment anhand der Strömung entlang der Strahlachse für das hexagonale Strahlbündel mit $A/d = 1,4$ und $D/d = 19$ bei $Re_d = 10^4$: Axialgeschwindigkeit (oben) und Längsfluktuationen (unten)

4 Diskussion

Hier sollen zwei Aspekte diskutiert werden: Zum einen die Zuverlässigkeit und Übertragbarkeit der Koeffizientenanpassung des SST-Turbulenzmodells. Zum anderen die Ursache für die zuvor dargestellten Probleme bei der Berechnung des Strahlbündels mittels SST-Turbulenzmodell.

Wie im vorangegangenen Abschnitt gezeigt wurde, können die wesentlichen Aspekte einfacher Strahlströmungen durch Anwendung des RANS-Ansatzes und des SST-Turbulenzmodells abgebildet werden. Durch Anpassung der Modellkoeffizienten (vgl. Tabelle 1) lässt sich die Ergebnisqualität so steigern, dass sich für den Freistrah und begrenzten Strahl eine sehr gute Übereinstimmung zum Experiment ergibt. Gleiches gilt auch für die komplexere Strömung des hexagonalen Strahlbündels, jedoch zeigt sich Potential für weitere Verbesserungen.

Für große Durchmesserhältnisse weist der begrenzte Strahl im düsen nahen Bereich Ähnlichkeiten zum Freistrah auf. Trotzdem sind unterschiedliche

Koeffizienten notwendig, um eine gute Ergebnisqualität zu erreichen. Ebenso weist die Strömung des Strahlbündels im Bereich des Rezirkulationsgebietes Ähnlichkeiten zum begrenzten Strahl auf. Dennoch wird auch für das Strahlbündel eine spezielle Anpassung der Modellkoeffizienten notwendig, um dieselbe Ergebnisqualität wie beim begrenzten Strahl erreichen zu können. Es kann an dieser Stelle nur vermutet werden, dass auch für das Strahlbündel ein entsprechender Koeffizientensatz existiert, der allerdings noch gefunden werden muss.

Tabelle 1: Zusammenhang zwischen der Strömungstopologie und den SST-Modellkoeffizienten

Bezeichnung	β^*	β_2	σ_{k2}	γ_2
SST (Standard)	0,09	0,0828	1,0	0,440
SST - opt 1	0,13	0,0828	1,0	5,107
SST - opt 2	0,11	0,0700	3,0	0,203

Die untersuchten Strömungen stellen Beispiele für Strömungen von Strahlen bzw. (freien) Scherschichten dar. Anhand dessen wird ersichtlich, dass aus ähnlichen Strömungsgemeotrien bzw. auf Basis sich ähnelnder Strömungsfelder nicht auf gleiche Koeffizienten des SST-Turbulenzmodells geschlossen werden kann. Vielmehr wird ein spezifisches Set von Modellkoeffizienten notwendig, welches sich für die entsprechende Strömung eignet. Damit entspricht die angewendete Methode eher einer spezifischen Modellanpassung.

Dies zeigt aber auch, dass sich die angewendete Modellierung (SST, Wirbelviskosität, RANS) nicht universell auf gleichartige Strömungen anwenden lässt, was auf eine gewisse phänomenologische Schwäche der Modellierung hinweist. Dies wird deutlicher, wenn man die Strömung des Strahlbündels betrachtet, die als 3-dimensional anzusehen ist. (Die Geometrie kann aufgrund des hexagonalen Musters als achssymmetrisch betrachtet werden. Für die Strömung hingegen, sind lediglich in Mündungsnähe achssymmetrische Eigenschaften denkbar, da dort die Strahlen noch nahezu unbeeinflusst sein können. Weiter stromabwärts sind aufgrund der Strahlinteraktion bzw. des Wandeinflusses keine Symmetrieeigenschaften zu erwarten.) Im Unterschied dazu sind der Freistrah und der begrenzte Strahl rotationssymmetrisch, was bedeutet, dass deren Transversalfuktuationen v'

und w' gleich sind. Dies wiederum führt dazu, dass die Normalspannungen v'^2 und w'^2 sowie Schubspannungen $u'v'$ und $u'w'$ gleich sind. Damit kann der Spannungstensor τ_{ij} auf drei Größen reduziert werden, wobei die Intensität von u'^2 und damit auch von $u'v'$ größer ist als die von v'^2 .

Aufgrund der 3-dimensionalen Eigenschaften des Strahlbündels ist anzunehmen, dass der Spannungstensor ebenso keine symmetriebedingten Eigenschaften aufweist. Vielmehr ist zu erwarten, dass sich auch die Transversalfluktuations v' und w' voneinander unterscheiden, wodurch sich ebenso unterschiedliche Schubspannungen $u'v'$, $u'w'$ und $v'w'$ ergeben. Dies würde eher dem Charakter von Drall-behafteten Strömungen entsprechen, wofür die Gruppe der Zwei-Gleichungsmodelle bekanntermaßen weniger geeignet ist. Dies würde die Schwierigkeiten des SST-Modells bei der Berechnung des Strahlbündels erklären. Fraglich bleibt daher, in wie weit das Modell mit Hilfe der Koeffizientenanpassung verbessert werden kann und wie übertragbar die Ergebnisse auf andere Strahlbündel sind (Variation von A und D).

5 Zusammenfassung und Ausblick

Gegenstand der vorliegenden Untersuchungen war die Optimierung der Koeffizienten des SST-Turbulenzmodells für Strahlströmungen. Dazu wurde ausgehend vom Freistrahls die Strömung des begrenzten Strahls und des hexagonalen Strahlbündels untersucht. Dabei konnte jeweils eine signifikante Verbesserung der Ergebnisqualität erreicht werden, wozu eine spezifische Anpassung der Modellkoeffizienten an den jeweiligen Strömungsfall notwendig ist. Innerhalb des betrachteten Strömungsfalls ist der gefundene Koeffizientensatz auch für andere Geometrien (Variation von d und D) und turbulente Strömungen (Variation von Re_d) gültig. Der komplexere Fall des Strahlbündels lässt sich durch den Koeffizientensatz des begrenzten Strahls nicht mit gleicher Qualität abbilden, weshalb ein spezieller Koeffizientensatz zu suchen wäre. Fraglich bleibt, ob diese Methode gerechtfertigt ist, da sich das Strahlbündel hinsichtlich des Spannungstensors grundsätzlich vom Freistrahls oder begrenzten Strahl unterscheidet.

6 Quellen

- [1] Y. Reinhardt, L. Kleiser: *Simulation of round jets with nozzle-dependent inflow conditions using the $k-\epsilon$ turbulence model*, Proceedings

in Applied Mathematics and Mechanics (PAMM), Vol. 14, Issue 1, 2014

- [2] B. Guo, T.A.G. Langrish, D.F. Fletcher: *CFD simulation of precession in sudden pipe expansion flows with low inlet swirl*, Applied Mathematical Modelling, Vol. 26, S. 1-15, 2002
- [3] Y. Bae, Y. Kim, K.K. Kim, J. Y: *Comparative study of turbulence model performance for axisymmetric sudden expansion flow*, Transactions of the Korean Nuclear Society Autumn Meeting, Gyeongju (Korea), 24.-25. Okt. 2013
- [4] 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
- [5] 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
- [6] F.R. Menter: *Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications*, AIAA Journal, Vol. 32, No. 8, August 1994
- [7] N.R. Panchapakesan, J.L. Lumley: *Turbulence measurements in axisymmetric jets of air and helium. Part 1. Air jet*, J. Fluid Mechanics, Vol. 246, S. 197-223, 1993
- [8] R.D. Gould, W.H. Stevenson, H.D. Thompson: *Investigation of Turbulent Transport in an Axisymmetric Sudden Expansion*, AIAA Journal, Vol 28, No. 2, S. 276-283, Feb. 1990
- [9] A. Ringleb, W. Schlüter, G. Wozniak: *Numerische Simulation des runden Freistrahls unter Anwendung des SST-Turbulenzmodells*, Tagungsband ASIM 2014, 22. Symposium Simulationstechnik, in Berlin, S. 17-22, 2014
- [10] A. Ringleb, W. Schlüter, G. Wozniak: *Ein Beitrag zur numerischen Simulation der Strömung des runden Freistrahls unter Anwendung des SST-Turbulenzmodells und dessen Anpassung für den selbstähnlichen Bereich*, Strömungstechnische Tagung 2014, in Dresden, Schriftenreihe aus dem Institut für Strömungsmechanik, Band 10, J. Fröhlich, S. Odenthal, K. Vogeler (Hrsg.), S. 239-249, 2014

Towards Safe Robotics – Modellbasierte Entwicklung von High Integrity Robotern

Dipl.-Ing. Maximilian Apfelbeck 1, Dipl.-Ing. Manuel Fédou 1, Dr.-Ing. Stephan Myschik

¹The MathWorks GmbH, Ismaning, Germany

manuel.fedou@mathworks.de

In Zukunft wird der Anteil an Robotern, die mit dem Menschen interagieren, rapide ansteigen. Diese sind mit einer großen Anzahl von Sensoren zum Erfassen der dynamischen Umgebung ausgestattet. Die Sensordaten werden bewertet und fließen in Algorithmen ein, die für eine sichere Zusammenarbeit und Reaktion des Roboters sorgen. Die verwendeten Algorithmen müssen auch gemäß Sicherheitsstandards wie der IEC 61508-3 [1] entwickelt werden. Eine Möglichkeit diesen Standards gerecht zu werden ist die Modellbasierte Entwicklung. Diese wird unter anderem schon in Automobilindustrie sehr erfolgreich eingesetzt. In diesem Vortrag wird eine Vorgehensweise zur Verifikation und Validierung von IEC 61508-3 konformen Softwarekomponenten kollaborierender Roboter anhand eines Beispiels diskutiert.

1 Einleitung

„Ein Roboter darf kein menschliches Wesen verletzen ...“ ist das Erste der Asimov’schen Gesetze [2]. Damit dieses Gesetz nicht verletzt wird, ist heutzutage ein sehr großer Prozentsatz der eingesetzten Roboter unterschiedlichster Größe durch Zäune oder Lichtschranken vom Menschen getrennt. Diese Trennung muss jedoch durch andere Sicherheitsmechanismen ersetzt werden, so dass Robotern neue Applikationen und Einsatzgebiete, vor allem in der Mensch-Roboter-Interaktion, realisierbar werden. Beispiele für neue Anwendungen sind unter anderem der Einsatz von Robotern als Produktionsassistent [3] oder in der Pflege von bedürftigen Personen.

Es gibt unterschiedliche technische Lösungen um Sicherheit in der Mensch-Roboter-Interaktion zu gewährleisten. Dies kann zum Beispiel durch das mechanische Design realisiert werden. Ein Ansatz hierfür ist die Begrenzung des dynamischen Bereiches des Roboters, so dass keine Verletzung des Menschen [4] bei Kontakt möglich ist. Andere Optionen sind das Einbauen von mechanischen Klemmen, die bei einer Überbelastung durchrutschen oder der Einsatz von seriellen-elastischen Aktuatoren [5]. Bei mechanischen Klemmen verhindert ein Durchrutschen der Klemme einen potentiell gefährlichen Mensch-Roboter Kontakt entstanden. Jedoch ist diese Lösung kaum anwendbar, da eine Neukalibrierung des kompletten Systems nötig wäre. Ein weiterer Lösungsansatz ist der Einsatz von Software und das Überwachen von Sensoren. Dies lässt sich mit wenig Aufwand erfüllen. Dazu wird der

Roboter mit einer intelligenten, sensiblen Haut ausgestattet, die bei unerwarteten oder hochenergetischen Kontakt den Roboter stoppt [6]. Für den Fall dass ein Werkzeug, wie zum Beispiel ein Schraubenzieher, potentiell gefährdend für den Menschen ist, wird mehr Intelligenz in den Algorithmen benötigt. Ein möglicher Lösungsansatz wird in [7] vorgestellt. Ein Hauptargument für den Einsatz von derartiger Überwachungs- und Sicherheitssoftware ist der Stückkostenpreis. Dieser nimmt mit der Anzahl der Endprodukte ab. Bevor diese allerdings in einem fertigen Produkt verwendet werden darf, müssen Fehlerfreiheit und eine Entwicklung in Übereinstimmung mit Sicherheitsnormen nachgewiesen werden.

Modellbasierte Entwicklung kann bei der Entwicklung von Regelungsalgorithmen oder Überwachungslogik für diese sicherheitsrelevanten Applikationen unterstützen. Verifikations- und Validierungskonzepte zum Nachweis der korrekt implementierten Funktionalität im Hinblick auf Anforderungen oder Normen sind Bestandteile der Modellbasierten Entwicklungsmethodik.

Im ersten Teil dieser Veröffentlichung wird der Modellbasierte Entwicklungsansatz am Beispiel eines Roboters gezeigt. Im zweiten Teil, werden Verifikations- und Validierungsschritte diskutiert, so dass die Konformität der erzeugten Softwarekomponente nach dem IEC 61508-3 Standard nachweisbar ist. Abschließend werden Anwendungsbeispiele aus der Luftfahrt und der Medizintechnik für den erfolgreichen Einsatz des aufgezeigten Entwicklungsprozesses präsentiert.

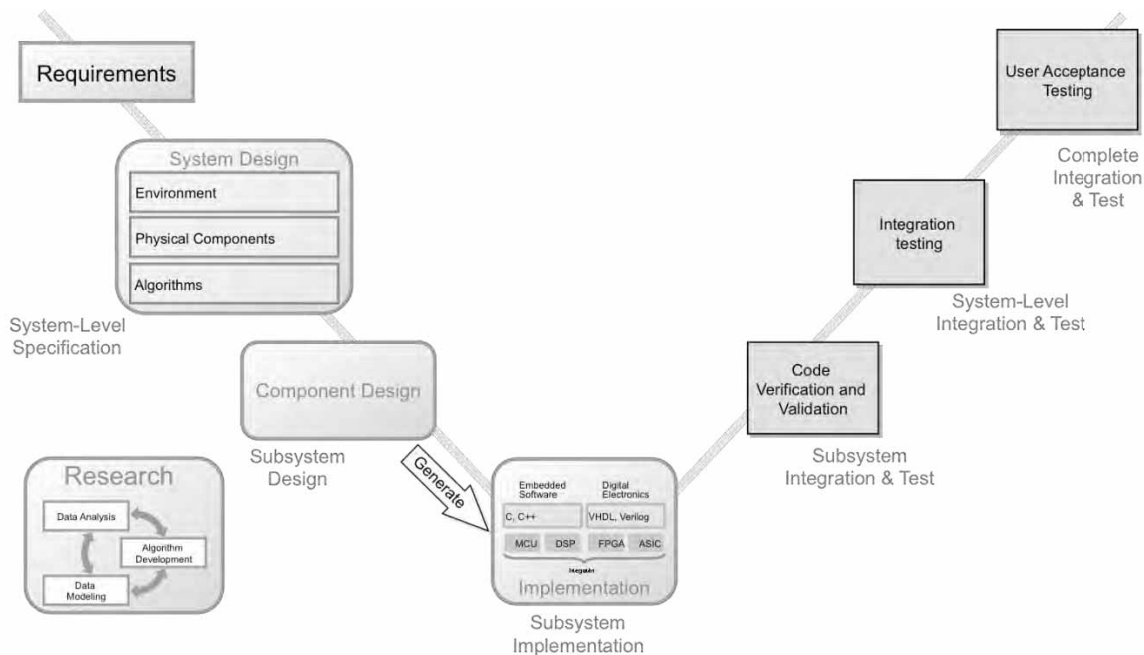


Abbildung 1: Das V-Modell als Vorgehensmodell der Softwareentwicklung

2 Modellbasierte Entwicklung entlang des V-Modells

Im Folgenden wird anhand des V-Modells [8], siehe Abbildung 1, die Entwicklung sicherheitsrelevanter Algorithmen für Roboter erklärt. Ein Entwicklungsprozess nach dem V-Modell ist weit verbreitet. Der verwendete Roboter ist in Abbildung 2 als CAD-Modell dargestellt. Das Ziel ist es, „Pick and Place“ Applikationen für den Roboter zu entwickeln. Dabei soll erlaubt sein, dass sich ein Mensch im Arbeitsraum des Roboters befindet. Neben der übergeordneten Aufgabe „Pick and Place“ Operationen durchzuführen werden technische Anforderungen, wie maximale Leistung,

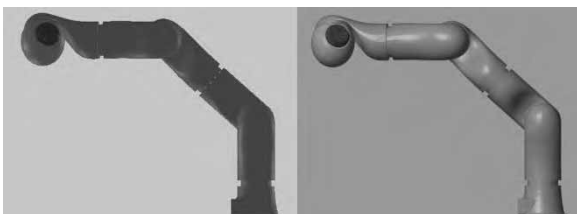


Abbildung 2: Screenshot des CAD-Systems (links) und des Mehrkörpersystems in Simscape Multibody™ (rechts)

Bauraum oder Wiederholbarkeit, und sicherheitsrelevante Anforderungen im Anforderungsdokument definiert. Die Anforderungen sind vermehrt in einem An-

forderungsmanagementsystem hinterlegt. Ein mögliches Requirement zur Gewährleistung einer sicheren Mensch-Roboter Kollaboration kann folgendes sein:

- REQ X.Y: Der Roboter soll innerhalb von x, x Sekunden seine Bewegung in jeder Konfiguration in einer potentiell gefährlichen Situation stoppen. Eine Gefährdung ist dann erreicht, wenn das Gelenkdrehmoment in jeder Achse $y\%$ oder z Nm über das erforderliche Moment übersteigt.

Diese Situation kann zum Beispiel in dem Fall eines Kontaktes mit Objekten oder Menschen während des Betriebes auftreten. Auf Basis des Anforderungsdokuments und des zu entwickelnden Systems wird in der Modellbasierten Entwicklung als erster Schritt die Systemarchitektur definiert. Hier wird eine Segmentierung in unterschiedliche Einheiten durchgeführt und es werden die Schnittstellen zwischen den Komponenten definiert. Ein Roboter kann in Software und Hardware aufgeteilt werden. Die Software ist in Überwachungslogik, koordinierenden Regler und Gelenkregler zerlegbar; die Hardware in Mechanik und Elektronik. Dieser Ansatz in Abbildung 3 dargestellt und in Simulink® umgesetzt worden.

Die Anforderungen werden für eine optimale Nachverfolgung mit den jeweiligen Komponenten bidirektio-

nal verknüpft. Im weiteren Entwicklungsverlauf werden nun die einzelnen Komponenten funktional entwickelt und nach der Interpretation der Anforderungen modelliert. Ein Vorteil von Simulink® ist, dass die Komponenten und somit die Requirements ausführbar sind. Somit erhält der Ingenieur sofortige Rückmeldung über die Qualität seines Designs.

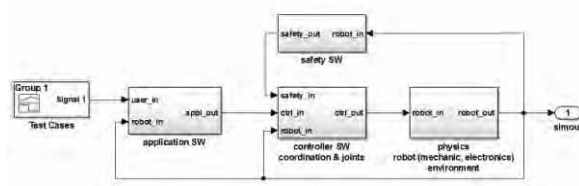


Abbildung 3: System Modell in Simulink®

Er kann frühzeitig im Entwicklungsprozess Fehler erkennen und somit das Projektrisiko erheblich minimieren. Die Erstellung der einzelnen Modellkomponente wird durch speziell angepasste Werkzeuge sehr effizient unterstützt. Die Mechanik kann aus CAD-Baugruppen in ein Mehrkörpersystem, bestehend aus Gelenken und Einzelkörpern, abgeleitet werden. Der CAD-Import in SimMechanics™ stellt somit sicher, dass die korrekten Bauteilparameter, wie Masse, Schwerpunkt oder Schwerpunktslage in der Simulation verwendet werden. Darüber hinaus wird die grafische Repräsentation der einzelnen Komponenten mitübernommen. Dieser Ansatz kann die Modellierung des mechanischen Systems erheblich beschleunigen [9]. Abbildung 2 zeigt die grafische Darstellung der importierten Baugruppe. Das importierte System kann auch als Startpunkt für Kinematikoptimierung oder Aktuatorauslegung verwendet werden. Neben den Motoren können weitere physikalische Komponenten (Getriebe) oder für die Regelung relevante Effekte (Reibung, Spiel, Steifigkeit) sehr einfach mit der Mehrkörpermechanik verknüpft werden. Dies wird in Abbildung 4 gezeigt. Die Parametrierung dieser Komponenten kann oft aus Datenblättern entnommen werden.

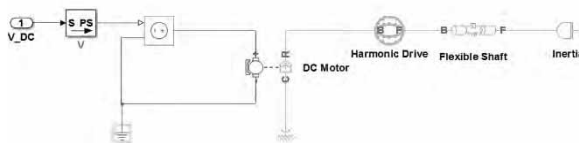


Abbildung 4: Modellierung einer Gelenkantriebseinheit

Zur Modellierung der Regler eignet sich am besten Simulink®; für die Überwachungslogik Stateflow®, siehe Abbildung 5. Sobald die physikalische Strecke und erste Softwarekomponente modelliert sind, können erste virtuelle Tests des Gesamtsystems durchgeführt werden. In kleinen Iterationsschritten wird nun immer mehr Intelligenz und Funktionalität zur Software hinzugefügt und durchgehend getestet. Mit jedem Simulationslauf wird das aktuelle Modell gegenüber den Anforderungen validiert und verifiziert. Zusätzlich lässt sich eine optimale Parametrierung der Softwarekomponente in der Simulation ermitteln. Abbildung 5 zeigt das Ergebnis der Sprungantwort eines optimierten PID-Reglers für ein Robotergelenk. Vorteile der Modellbasierten Entwicklungsmethodik sind

- Die Repräsentation eines Gesamtsystems mit Modellen erhöht die Verständlichkeit der erstellten Systeme und verbessert die Kommunikation zwischen den einzelnen Ingenieursdisziplinen in einem Unternehmen.
- Die Verwendung einer Entwicklungsumgebung reduziert den Aufwand in der Konvertierung von Daten und Informationen zwischen unterschiedlichen Softwaretools.
- Die Simulation des Gesamtsystems in frühen Phasen hilft erheblich in sehr frühen Phasen Fehler zu finden und zu beheben.

In folgender NASA Studie [10] wird die Wichtigkeit Fehler in frühen Entwicklungsstadien zu finden bestätigt. Je später ein Fehler innerhalb eines Projektes gefunden wird, umso teurer wird es diesen zu beheben.

Nachdem das Softwaredesign komplett auf Modellebene entwickelt wurde und die Einhaltung der Anforderungen nachgewiesen ist, kann Produktionscode aus dem Modell mit dem Embedded Coder erzeugt werden. Dieser ist vom TÜV Süd für die Software Entwicklung für IEC 61508-3 kompatiblen Code vorqualifiziert [11].

3 Verifikation und Validierung von Modell und generiertem Code

Folgende Fragen müssen in der Software Verifikation und Validierung bei der Modellbasierten Entwicklung positiv beantwortet werden können [12]:

- Sind im Modell die textuellen Anforderungen korrekt implementiert?

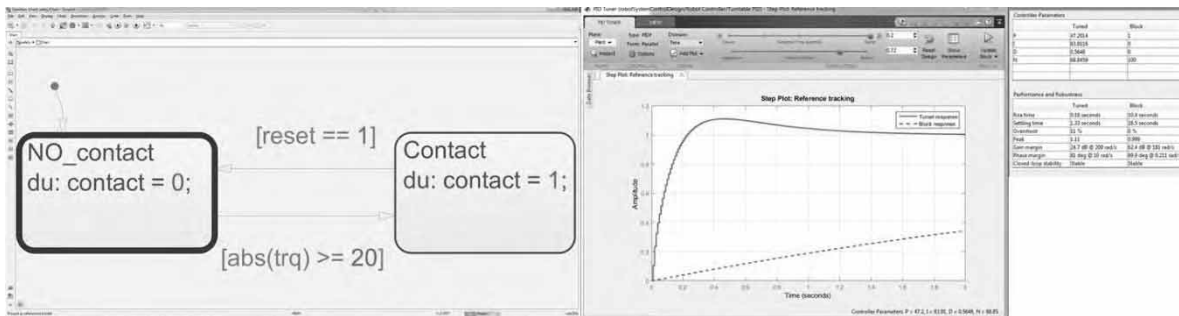


Abbildung 5: Modellierung der Überwachungslogik in Stateflow (links). Der aktuelle Zustand wird durch den blauen Rahmen hervorgehoben. Sprungantwort des optimierten Gelenkreglers (rechts). Das ursprüngliche Design ist blau gestrichelt dargestellt.

- Spiegelt der Objekt Code, der später am Roboter verwendet wird, das Verhalten des Modells korrekt wieder?

MathWorks hat einen Referenzworkflow [13] entwickelt der bei der Beantwortung der beiden Fragen hilft. Dazu wird die Entwicklung in die Design Verifikations- und Code Verifikationsphase unterteilt.

3.1 Design Verifikation

Das Ziel dieser Phase ist es zu beweisen, dass das Modell die Anforderungen erfüllt und keine unerwünschte Funktionalität beinhaltet. Dies wird sowohl durch statische Analysen als auch mittels funktionalen Tests gezeigt. Bei der statischen Analyse wird das Modell auf Konstrukte untersucht, die z.B. nicht-optimal für die Codegenerierung sind. Darüber hinaus können vordefinierte oder eigene Modellierungsrichtlinien überprüft werden. Modelle, die in Produktionscode überführt werden, sind auch im Hinblick auf Konformität mit Sicherheitsstandard überprüfbar. Abbildung 6 zeigt das Ergebnis für die Überprüfung eines Modellteils nach „Modeling Standards for IEC 61508“. Dieser Bericht unterstützt bei der Identifikation von Modellierungsfehlern, Berichtigung und Dokumentation eines erfolgreichen Tests. Über eine Modell Checksumme kann der Bericht eindeutig einem System und einer Modellversion zugeordnet werden. Sobald die Modellierungsrichtlinien erfolgreich überprüft sind, wird eine Abdeckungsanalyse des Modells durchgeführt. Ziel ist es unerwünschte Funktionalität oder fehlende Anforderungen aufzudecken. Ein mögliches Beispiel für den im Abschnitt 2gezeigten Roboter, ist das Berechnen eines richtigen Ergebnisses der Inverskinematik. Die berechneten Gelenkwinkel müssen

in jedem Fall innerhalb der Gelenkbegrenzungen liegen. Dies kann z.B. durch das in Abbildung 7 gezeigte Konstrukt bewiesen werden.

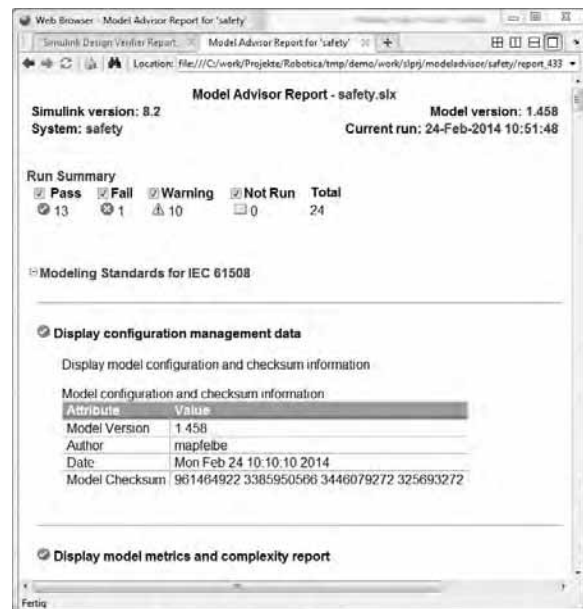


Abbildung 6: Ergebnis der Überprüfung der Modellierungsrichtlinien nach IEC61508 für das safety Modell

Damit die funktionale Korrektheit der Inverskinematik nachgewiesen wird, müssen die Ergebnischecks erfüllt werden. Abbildung 8 zeigt eine Verletzung dieser Checks auf Grund einer falschen Implementierung.

Außerdem muss auch eine komplette Modellabdeckung der Inverskinematik vorliegen. Das Ergebnis der Analyse eines spezifischen Testfalls, in Abbildung 9, zeigt, dass bestimmte Bereiche der Inverskinematik nicht ausgeführt werden. Mit diesem Test ist somit

keine Aussage über die funktionale Korrektheit bestimmter Modellteile zu beweisen. Alle Tests können automatisch durchgeführt und dokumentiert werden.

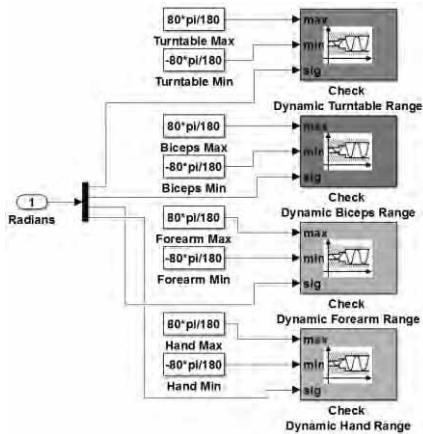


Abbildung 7: Überprüfen der Gelenkwinkelgrenzen auf Modellebene

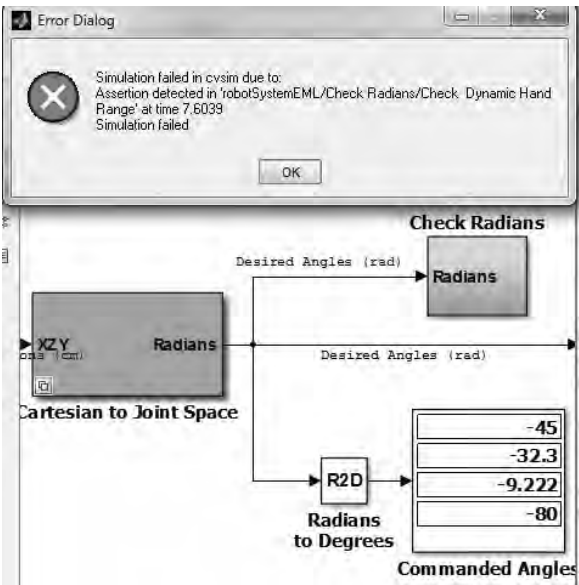


Abbildung 8: Nicht-Bestehen eines virtuellen Tests aufgrund von Überschreitung der Gelenkgrenzen

#71: if handSearchAngle < 90 * pi/180 && forearmSearchAngle < 90 * pi/180

Decisions analyzed:		
if handSearchAngle < 90 * pi/180 && forearmSearchAngle < 90 * pi/180		
		50%
false		0/101
true		101/101

Conditions analyzed:		
Description:	True	False
handSearchAngle < 90 * pi/180	101	0
forearmSearchAngle < 90 * pi/180	101	0

Abbildung 9: Ergebnis der Abdeckungsanalyse für einen spezifischen Modellteil

Neben diesen funktionalen Tests ist es auch möglich Testvektoren automatisch abzuleiten. Nachdem alle statischen und funktionalen Tests erfolgreich durchgeführt wurden und eine komplette Abdeckung der Anforderungen durch das Modell bewiesen wurden, wird automatisch Code aus dem Modell generiert. Abbildung 10 zeigt schematisch die Vorgehensweise bei der Code-Generierung. Der aus dem validierten Modell abgeleitete Code kann nun auf die Zielplattform übertragen werden. Die Code Generierung erzeugt einen Bericht, der bei der Nachverfolgbarkeit von den Anforderungen in das Modell und weiter in den Code unterstützt. Der bisher diskutierte Entwicklungsprozess geht davon aus, dass ein Aufspielen von eigener Software auf die Robotersteuerung möglich ist. Hat der Anwender jedoch nur die Freiheit zwischen Betriebsmodi und deren Parametrierung auszuwählen, ist die Arbeit am Modell nach dem erfolgreichen Ermitteln der Parametrierung, z.B. durch Optimierung, größtenteils abgeschlossen. Für diese Fälle wird oftmals das Modell zum Ermitteln von optimalen Trajektorien weiterverwendet.

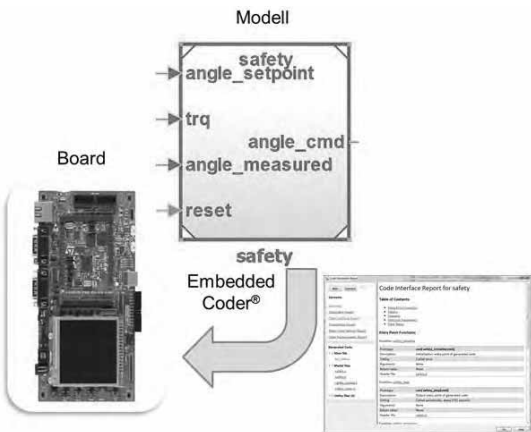


Abbildung 10: Automatisches Erzeugen von Produktionscode mit dem Embedded Coder® [14]

3.2 Code Verifikation

Zur kompletten Validierung des Algorithmus muss bewiesen werden, dass kein funktionaler Unterschied zwischen Modell und Objekt Code vorliegt. Dazu werden Modell und Objekt Code mit identischen Test Vektoren angeregt. Der Objekt Code wird direkt auf der Zielplattform ausgeführt. Embedded Coder bietet dazu „Prozessor-in-the-Loop“ Simulation (PIL) an. Dies wird in Abbildung 11 gezeigt. Die Ergebnisse der Simulation des verifizierten Modells werden neben den Testvektoren als Erwartungswerte abgespeichert. Bei der PIL-Simulation wird der Objekt Code auf die Zielplattform über eine Debug-Schnittstelle, zum Beispiel

JTAG, übertragen. Das auf dem Prozessor übertragene Simulink Modell wird im PIL-Modus ausgeführt. Die Kommunikation zwischen dem in Simulink ausgeführten Robotermodell und dem Regler auf der Zielplattform wird über die serielle Schnittstelle oder TCP/IP hergestellt.



Abbildung 11: Konfiguration bei der PIL-Simulation

Korrekte Funktionalität des Objekt Codes ist dann bewiesen, wenn ausreichend gute Übereinstimmung zwischen den beiden Simulationen (verifiziertes Modell und PIL) vorhanden ist. Einflüsse unterschiedlicher Compiler oder Fließkommazahl-Einheiten können Unterschiede zwischen Simulation und Objekt Code verursachen. Neben dem Nachweis der funktionalen Äquivalenz muss auch bewiesen werden, dass bei der Code-Generierung keine unerwünschte Funktionalität eingefügt wurde. Dies kann durch den Vergleich der Abdeckung auf Modell und Code-Ebene sowie durch das Überprüfen der Nachverfolgbarkeit, welche der Embedded Coder® durch eine automatische Dokumentation unterstützt, gezeigt werden.

Ein Nachverfolgbarkeitsbericht wird in Abbildung 12 gezeigt. Dieser Bericht enthält das komplette Mapping des Models auf den Code und eine Liste von Modellkomponenten, die wegen Codeoptimierung im erzeugten Code fehlen. Eine bidirektionale Verlinkung von Model zum Code sowie Links zu den Anforderungen sind ebenso dokumentiert, siehe Abbildung 13.

4 Anwendungen in der Industrie

In den beiden vorausgehenden Kapiteln wird eine mögliche Vorgehensweise zur Erzeugung von zertifizierbarer Software mit Modellbasierter Entwicklung erläutert. Dieser Prozess wird ganz oder in Teilen von vielen Unternehmen unterschiedlichster Industrien genutzt. Weinmann [15] verwendet Teile des beschriebenen Entwicklungsprozesses um Software für neue Transportbeatmungsgeräte zu entwickeln. Bei diesem

Block Name	Comment
Root/volatile_setpoint	Output
Root/volatile_measured	Output
Root/volatile_reset	Output
Root/volatile_setpoint	Output
Root/volatile_measured	Output
Root/volatile_reset	Output
Root/volatile_setpoint	Output
Root/volatile_measured	Output
Root/volatile_reset	Output

Abbildung 12: Bericht zur Nachverfolgung von Modellkomponenten im Code

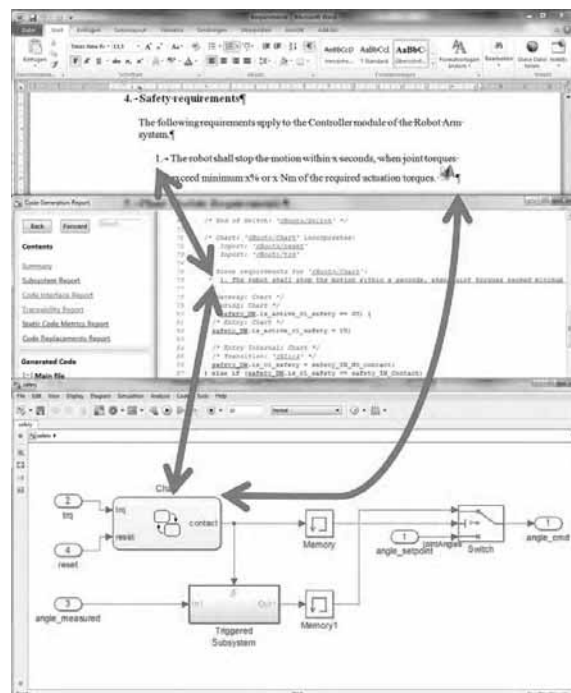


Abbildung 13: Bidirektionale Verlinkung zwischen Anforderungen, generiertem Code und Modell

Produkt ist die Komplexität der Algorithmen um vielfaches höher als in der Vergangenheit. Neben den Algorithmen wurde auch die Funktionalität der menschlichen Lunge und des Ventilators modelliert. Damit war es möglich das Gesamtsystem, bestehend aus Software, Gerät und Menschen, vorab zu berechnen und eine Vielzahl unterschiedlicher Designalternativen zu evaluieren und zu testen. Dies war mit deren herkömmlicher Entwicklungsmethodik nicht möglich. Bevor Produktionscode erzeugt wurde, mussten die Algorithmen auch Abdeckungstests auf Modellebene

bestehen. Der Wechsel zur Modellbasierten Entwicklung beschleunigte den Entwicklungsprozess bei Weinmann, so dass die Zertifizierung für sicherheitskritische Systeme viel schneller möglich ist. Die modellierten Systeme werden in zukünftigen Projekte zudem vielfach wiederverwendet.

Durch den Einsatz von Modellbasierter Entwicklung beschleunigt Eurocopter die Entwicklung von DO-178B zertifizierter Software [16]. Deren Hauptherausforderung bestand darin, dass Design Fehler durch eine falsche Interpretation der Anforderungen oder falsche Implementierung der Systeme, die sich richtig aber nicht wie beabsichtigt verhielten, eingeführt wurde. Eurocopter schätzt, dass ungefähr 90% der Probleme, die spät im Projekt gefunden wurden, durch Fehler in der Spezifikations- oder Designphase eingeführt wurden. Die Anzahl der Fehler wurde durch das Einführen der Modellbasierten Entwicklung auf Grund des frühzeitigen Testens und der kontinuierlichen Verifikation und Validierung erheblich reduziert. Die erstellten Modelle werden mit Modell-Standard Checks und Modell Abdeckung auf Einhaltung von Modellierungsrichtlinien zur Erfüllung der DO-Norm analysiert. Der automatisch generierte Code wird nach erfolgreicher Überprüfung in Objektcode kompiliert. Dieser wird gegen die schon vorhandenen Testvektoren getestet. Der beschriebene Prozess unterstützt Eurocopter, um automatisch Code zu generieren, der dann für die DO-178B Norm der EASA zertifiziert wird. Mit dieser Vorgehensweise wurde die Zeit zum Testen der Software um 2/3 reduziert und die Requirements konnten viel früher eingefroren werden. Im Vergleich zu ähnlichen Projekte werden diese nun ungefähr 1 Jahr früher endgültig fixiert.

5 Zusammenfassung

Es wurde eine Vorgehensweise zum Erstellen von zertifizierten Code nach IEC61508-3 für robotische Systeme diskutiert. Die Anwendung dieses Software Entwicklungsprozesses führt zu einer höheren Software Qualität, verminderten Entwicklungszeiten und zertifizierte Software mit weniger Aufwand. Darüber hinaus unterstützt die Modellierung der Algorithmen und Systeme und deren grafische Darstellung technische Kommunikation sowie das Systemverständnis in Projektteams. Modellbasierte Entwicklung hat das Potential ein wichtiger Baustein für zukünftige Entwicklungen von kollaborierenden Robotern zu werden.

6 References

- [1] International Electrotechnical Commission: *Functional safety of electrical/electronic/programmable electronic safety related systems – Part 3: Software requirements*, IEC 61508-3 ed.2, Geneva, 2010.
- [2] I. Asmiov: *Runaround*, 1942.
- [3] Robert Bosch GmbH, http://www.bosch-apas.com/en/apas/start/bosch_apas.html.
- [4] ABB AG, <http://new.abb.com/products/robotics/de/yumi>.
- [5] Rethink Robotics, <http://www.rethinkrobotics.com/baxter/>.
- [6] MRK Systeme GmbH, http://www.mrk-systeme.de/produkte_interaction.html.
- [7] S. Haddadin: *Towards Safe Robots*, Springer Tracts in Advanced Robotics, Vol. 90, 2013.
- [8] J. Friedrich; M. Kuhrmann; M. Sihling; U. Hammerschall: *Das V-Modell XT Für Projektleiter und QS-Verantwortliche kompakt und übersichtlich*, Springer-Verlag Berlin Heidelberg 2009.
- [9] Saneon GmbH, <http://www.saneon.de/cms/index.php/en/component/content/article/1-aktuelle-nachrichten/123-simulation-saneon.html>.
- [10] NASA, *Return on Investment for Independent Verification & Validation*, 2004.
- [11] TÜV SÜD Certificate, Z10 11 12 67052 014.
- [12] M. Conrad; G. Sandmann: *A Verification and Validation Workflow for IEC 61508 Applications*, SAE Technical Paper 2009-01-0271, 2009.
- [13] The MathWorks, Inc. <http://www.mathworks.com/products/iec-61508/>.
- [14] The MathWorks, Inc. <http://www.mathworks.de/products/embedded-coder/>.
- [15] The MathWorks, Inc. http://www.mathworks.com/tagteam/76574_91946v01_Weinmann_UserStory_final.pdf.
- [16] The MathWorks, Inc. http://www.mathworks.com/tagteam/77159_92118v00_Eurocopter_UserStory_final.pdf.

Verteilte Simulation mehrachsiger magnetischer Positionssensorsysteme

Jörg Bretschneider¹, Xuehao Wang², Willi Neudeck¹

¹Fraunhofer Institut für Integrierte Schaltungen IIS, Institutsteil EAS Dresden

²Technische Universität Dresden, Fakultät Elektrotechnik

joerg.bretschneider@eas.iis.fraunhofer.de

Zusammenfassung: Die Simulation mehrachsiger, magnetischer Positionssensorsysteme im Rahmen des modellbasierten Systementwurfs stellt an die Hardware erhebliche technische und an den Entwurfsingenieur erhebliche didaktische Anforderungen. In dem Beitrag wird ein Ansatz erläutert, wie die Komplexität dieses Entwurfs didaktisch reduziert und die Leistungsanforderungen durch ein angepasstes und verteiltes Simulationskonzept deutlich reduziert werden können.

1 Modellbasierter Entwurf mehrachsiger Magnetpositionssensorik

Integrierte Sensoren, die magnetische Felder vektoriell punktgenau erfassen können, sind bereits seit mehreren Jahren in der industriellen Anwendung angekommen, sowohl als anwendungsspezifische integrierte Schaltkreise etwa in Waschmaschinen oder Joysticks, aber auch als Standard-ICs, die der klassischen Weg- oder Winkelmessung durch zusätzliche Messgrößen höhere Genauigkeit oder Robustheit verleihen [1],[2]. Beispiele dafür sind die Sensoren der AS54xx Serie von ams [3] oder die HAL3xx Serien von Micronas [4]. Damit werden allerdings die Potentiale dieser Sensortechnologie nur zu einem geringen Teil ausgeschöpft, denn theoretisch können mit ausreichend vielen 3D-Magnetfeldsensoren auf einem IC alle Bewegungen im Raum, also je drei Verschiebungen und Verdrehungen von Objekten, sowie zusätzliche Magnetfeldgrößen erfasst werden, was der magnetischen Positionsmessung neue Anwendungsfelder im Bereich der intelligenten und sicherheitskritischen Sensorik eröffnet [11].

Dank technischer Weiterentwicklungen in der Hallsensorik, getrieben vor allem durch das Fraunhofer Institut für integrierte Schaltungen IIS in Erlangen und Dresden (vgl. [5],[6],[7],[8],[9]), sind magnetische Positionssensorsysteme, die mehrere Bewegungsachsen gleichzeitig erfassen können, inzwischen auch praktisch unter der Fraunhofer-IIS-Marke HallinMotion[®] als kompakte, eingebettete Systeme realisierbar [12],[13].

Solche Systeme bieten sowohl im Bereich der Industrieautomation als auch im Fahrzeugbau und in der

Medizintechnik eine Reihe von Vorteilen, die über die klassische Verschleißfreiheit der berührungslosen Messung deutlich hinausgehen.

Neben der Möglichkeit, in Dreh- und Weggebern neben der eigentlichen Messgröße auch die unvermeidlichen Sekundärbewegungen zu messen und so deren negative Auswirkungen auf die Genauigkeit der Messung auszugleichen [13], ermöglichen diese Systemlösungen auch die Erfassung von räumlichen Bewegungen, also Verschiebungen und/oder Verdrehungen, die auf mehr als einer Achse gleichzeitig stattfinden.

Herkömmlich werden zur Detektion aller Freiheitsgrade mehrere Hall-Sensoren und oft auch mehrere Gebermagnete benötigt, die zudem als räumlich separierte Systeme mit allen Kosten- und Bauraumnachteilen ausgeführt werden müssen, was einen Einsatz in vielen Fällen unwirtschaftlich macht. HallinMotion[®]-Systeme bestehen hingegen aus nur einem Standard-Magneten und einem HallinOne[®]-Sensor-IC, der mit einem industrietüblichen Mikrokontroller verbunden ist.

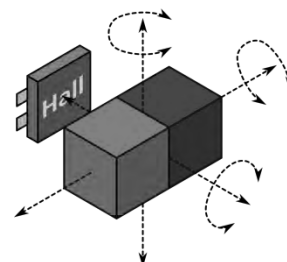


Bild 1 Prinzipieller Aufbau eines magnetbasierten Positionssensorsystems mit mehreren mechanischen Freiheitsgraden.

Die Messaufgabe gelingt also bei HallinMotion® mit einem einzigen kompakten Modul und einem einzigen Gebermagneten, der zudem normalerweise keine Spezialanfertigung sein muss, sondern mit üblichen Materialien und Standardformen auskommt [13] (Bild 2). Dies reduziert nicht nur die Stückkosten, sondern auch die Systemkosten gegenüber Systemen mit mehreren diskreten Sensormodulen erheblich.

Gegenüber den weitverbreiteten Weg- und Drehgebern sind jedoch solche Systeme aufgrund der räumlichen Dimension und der Nichtlinearität räumlicher Magnetfelder erheblich komplexer und der Entwurf und die Entwicklung entsprechend aufwendiger. Die Magnetfeldmesswerte lassen sich nun nicht mehr durch rein konstruktive Maßnahmen als lineare Abbildung der Position darstellen und durch einfache Linearisierungstabellen in die gesuchte Weg- oder Winkelgröße umrechnen, sondern können innerhalb des mehrdimensionalen Bewegungsraumes viele Werte annehmen, die bei ungünstiger Konstruktion nicht mehr eindeutig auf die Position zurückführbar sind oder über dem Bewegungsgebiet ein sehr ungleich verteiltes Signal-Rauschverhalten aufweisen, was eine einheitliche Genauigkeit der Positionsbeurteilung beeinträchtigt. Auch kommt in einem räumlichen Magnetfeld und räumlichen Bewegungsraum schnell die menschliche Vorstellungskraft an ihre Grenzen, was eine Plausibilisierung von Messwerten und Signalverläufen beträchtlich erschwert, oft sogar – ohne geeignete Hilfsmittel zur Reduktion – unmöglich macht.

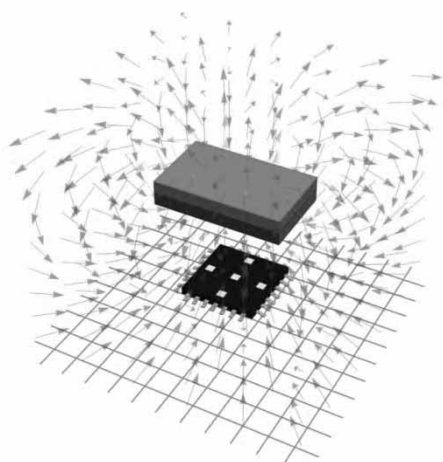


Bild 2 HallinMotion®: Fraunhofer HallinOne® Sensor-IC mit 5 3D-Hall-Sensoren im Magnetfeld eines quaderförmigen Gebermagneten für einen einzelnen Positionswert. Simulation im Rahmen einer 6D-Applikationsentwicklung

Durch einen simulationsgestützten, modellbasierten Systementwurf wird diese Komplexität strukturierbar, reduzierbar und damit für den Entwurfsingenieur fassbar. Hierfür sind applikationsspezifische Modelle der Systemkomponenten Magnet, Bewegungstrajektorie und Sensorsystem notwendig, deren Variation einen Vergleich und die Bewertung unterschiedlicher Systementwürfe bereits in sehr frühen Phasen der Anwendungsentwicklung ermöglicht. Durch schrittweise Steigerung der Modellgranularität kann die Belastbarkeit der Ergebnisse entsprechend der Anforderungen der jeweiligen Entwurfsphase gesteigert werden [14].

Mittels Corner-Simulation der Modellparameter lässt sich prinzipiell das Systemverhalten bis hin zum Serienprodukt abklären, das dann durch Prototypentests lediglich verifiziert werden muss. So wird der Entwurfsprozess schneller, flexibler und kostengünstiger. Dadurch sinkt die Markteinführungszeit und die Innovationskraft der Entwickler steigt.

Für räumliche magnetische Positionssensorik setzt dieser modellbasierte Entwurfsprozess jedoch voraus, dass die Systemmodelle simulierbar bleiben, was angesichts der mit jedem mechanischen oder anderen Freiheitsgrad exponentiell wachsender Datenmenge sowohl Rechenzeit als auch Speicherplatz schnell an technische Grenzen selbst leistungsfähiger PC-Hardware bringt.

Im Folgenden werden Ansätze dargestellt, mit denen sich das Datenvolumen reduzieren lässt, ohne auf die vollständige Überdeckung des Systemraumes zu verzichten, und wie der modellbasierte Entwurfsprozess durch geeignete Projektions- und Visualisierungsmethoden wirksam unterstützt werden kann.

2 Simulierbarkeit durch adaptives, verteiltes Datenmodell

Die simulationsgestützte Entwicklung eines magnetischen Positionssensorsystems entspricht der Berechnung eines Vektorfeldes und in der Folge weiterer abgeleiteter Vektorfelder über einem Trajektorienraum mit mehreren mechanischen Freiheitsgraden $nDoF$, der hierzu hinreichend fein diskretisiert werden muss. Ergeben sich bei $nDoF = 3$ Freiheitsgraden und üblichen 100 diskreten Stützpunkten pro Achse noch 10^6 Gitterpunkte, für die pro Vektorfeld wenigstens drei float-Werte á 4Byte zu speichern sind, also 24MB, so erfordert bereits ein vierter Freiheitsgrad 2.4GB, sechs Freiheitsgrade hingegen 24TB Speicher

für ein Vektorfeld, allein um das Magnetfeld am Ort eines 3D-Hallsensors über dem Trajektorienraum zu speichern. HallinOne[®]-Sensoren haben üblicherweise drei bis fünf 3D-Sensoren auf einem IC, und die Abbildung reicht von der Position über das zu berechnende Magnetfeld und die interne Umwandlung in eine Hallspannung sowie eine Digitalisierung des Messwerts auf das Ausgangssignal des Sensors bis zurück auf die aus diesen Signalen durch eine Auswertungseinheit berechnete „gemessene“ Position, welche die eigentliche Ziel- und Bewertungsgröße in ihrem Vergleich zur tatsächlichen Position darstellt. Um die in den Anwendungen erforderlichen Genauigkeiten und Auflösungen nachzuweisen, sind ab einem gewissen Entwurfsfortschritt double-Werte sinnvoll.

Es ist daher kaum vorstellbar, ein solches System mit üblicher PC-Hardware und Vorgehensweise vollständig zu simulieren, schon weil der Speicherplatz hierfür nicht ausreicht. Zudem benötigt die Magnetfeldberechnung je nach Magnetform und zugehörigem Magnetfeldmodell selbst auf hochperformanter Hardware eine erhebliche Rechenzeit, so dass eine 6D-Simulation mit 10^{12} Gitterpunkten etwa auf einem handelsüblichen Quadcore-PC mehrere Tage in Anspruch nehmen würde.

Daher müssen für die Simulation Wege gefunden werden, um die Anzahl der Datenpunkte drastisch zu reduzieren und dabei dennoch globale und lokale Aussagen über Magnetfeldverläufe im Trajektorienraum und die Güte der Positionsrechnung sowie Modellierungsfehler treffen zu können.

Eine Möglichkeit hierzu besteht darin, die Stützstellen nicht als regelmäßiges Gitter, sondern ungleich verteilt anzulegen, um etwa Teilbereiche von größerem Interesse fein aufgelöst zu simulieren, andere Bereiche hingegen gröber zu diskretisieren, etwa Bereiche, in die das Objekt in der Applikation eher selten bewegt wird. Es ist jedoch schwierig, solche Bereiche bereits im Entwurf zu identifizieren und vor allem, sie sinnvoll voneinander abzugrenzen.

Eine andere Möglichkeit besteht in einer schrittweise verfeinerten Vorgehensweise, bei der zunächst eine recht grobe Diskretisierung vorgenommen wird, die vergleichsweise schnell simulierbar ist. Diese Diskretisierung wird für die Untersuchung der generellen Machbarkeit und Übereinstimmung der Modellierung mit der Spezifikation verwendet und liefert grobe, aber belastbare Anhaltspunkte zur Bewertung des Ge-

samtentwurfs. Für genauere Untersuchungen werden dann lediglich vom Entwerfer direkt definierte oder ausgewählte Hyperebenen oder Teilbereiche feiner diskretisiert, zum Beispiel ein innerer oder Randbereich gleicher Dimension oder auch Schnittlinien oder -ebenen durch den Gesamt-Trajektorienraum, etwa entlang eines oder zweier Freiheitsgrade.

Eine Alternative stellt eine Diskretisierung mit über dem Trajektorienraum verteilten Zufallspunkten dar. Auch hier lassen sich grobe Anhaltspunkte gewinnen, allerdings können bei der grafischen Darstellung Interpolations-Artefakte deutlich störend wirken, die durch die ungleiche Verteilung der Punkte entstehen. Dies lässt sich durch geeignete Wahl der Verteilung beeinflussen, bei Verwendung etwa von Normalverteilungen liegen die meisten Punkte um das Zentrum des Trajektorienraums. Generell sind bei Zufallsverteilungen aber die Ränder kaum oder gar nicht abgedeckt, die jedoch für die im Entwurfsprozess notwendigen Grenzaussagen erheblich sind. Dies kann aber durch eine Kombination sichergestellt werden, die Randpunkte zusätzlich zu den Zufallspunkten einfügt.

Schließlich lassen sich auch sogenannte Lissajous-Kurven hinreichend hoher Ordnung und geeigneter Parametrierung verwenden, um den Trajektorienraum überdeckend mit Stützstellen zu diskretisieren. Hierbei handelt es sich um Linearkombinationen harmonischer Schwingungen, die z.B. von Oszillographen bekannt sind.

Eine n-dimensionale, den Raum hinreichend überdeckende diskrete Lissajous-Figur entsteht durch Kombination mehrerer Lissajous-Kurven mit Amplituden c_j und geeigneten Verhältnissen der Frequenzen ω_j und Phasen φ_j als

$$x_n(k) := \sum_{j=0}^n c_j \cdot \sin(\omega_j \cdot t_k + \varphi_j) \cdot \mathbf{e}_j, \quad (1)$$

wobei \mathbf{e}_j den j-ten Einheitsvektor in \mathbb{R}^n darstellt. Ein Vorteil hierbei ist, dass man die Stützstellen auf jeder Lissajouskurve durch äquidistant verteilte „Zeitpunkte“ $t_k = k \cdot \Delta t$ definieren und so das gewohnte Raster-schema verwenden kann.

Es ist zu beachten, dass sich die Stützstellen bei ungünstiger Wahl der Lissajous-Parameter sehr ungleichmäßig im Raum verteilen und dann auch die Ränder des Trajektorienraumes deutlich unterdurchschnittlich belegt sind. Dies wird bereits anhand der abhängig von der Parametrierung deutlich variierende Form dieser Kurven offensichtlich (vgl. Bild 3).

Um sicherzustellen, dass die Stützstellen t_k auf der diskretisierten Lissajouskurve x_n hinreichend dicht liegen, ist der Parameter Δt nach der Vorschrift

$$\Delta t \leq \frac{dx}{\sqrt{\sum_j c_j^2 \omega_j^2}} \quad (2)$$

zu wählen, wenn der Abstand zweier benachbarter Punkte auf der Kurve dx nicht überschreiten soll. Die Wahl der geeigneten Lissajous-Parametrierung für eine gewünschte Überdeckung des Trajektorienraumes von mehr als vier Freiheitsgraden ist allerdings nicht trivial.

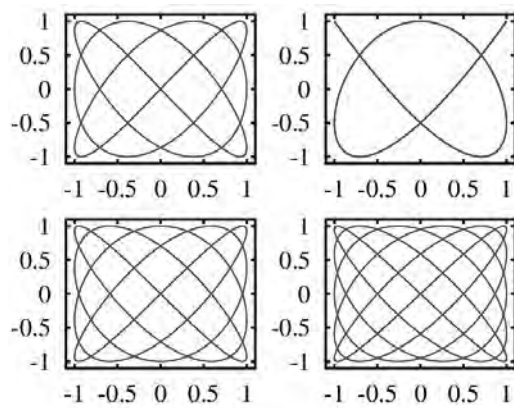


Bild 3 Lissajous-Figuren für 2 Freiheitsgrade mit verschiedenen Frequenzverhältnissen und Phasenverschiebungen. Die Parameterkombination rechts oben führt überdeckt den 2D-Raum nur unzureichend.

In jedem Fall ergeben sich mit zunehmendem Fortschritt der Anwendungsentwicklung und entsprechend steigenden Anforderungen an die Auflösung bei der Simulation immer größere Datenräume. Diese nehmen schnell Speicherplatz in einem Umfang in Anspruch, der mit herkömmlichen, dateibasierten Datenstrukturen nicht mehr sinnvoll bearbeitbar ist, weil diese zu langsam sind und Arbeitsspeicher samt Auslagerungsdatei um Größenordnungen überfordern.

Diese komplexen und großen Datenmengen benötigen zur Simulation und schnellen Visualisierung eine Datenstruktur, die in der Lage ist, dedizierte Teildatensätze schnell und separat zu berechnen und auf physikalisch verteilten Speichern abzulegen, um sie danach für eine grafische Anzeige oder Weiterverarbeitung ebenso schnell zu extrahieren.

Heute bieten verteilte Simulationen auf Rechnerclustern und die Nutzung von Clouddiensten hierfür Möglichkeiten in bisher nicht gekanntem Ausmaß und Flexibilität, doch mit herkömmlichen Datenstrukturen können diese nicht genutzt werden.

Voraussetzung für eine Erfüllung der Anforderungen ist ein hierarchisches Datenmodell, das für verteilte Berechnungen geeignet ist. Solche Datenmodelle wurden in den vergangenen Jahren für verschiedenste wissenschaftliche Anwendungen – etwa in der Klimaforschung – entwickelt, wobei CDF und das HDF5-Format weite Verbreitung gefunden haben.

Diese Datenformate erlauben auch bei sehr großen Modellen einen schnellen Zugriff auf Teilmengen (Slicing), wie es nicht zuletzt für die grafische Darstellung und Bewertung durch den Entwurfsingenieur unabdingbar ist. Sie gestatten aber auch die Organisation mehrerer oder sogar sehr vieler Modellvarianten in einem Modell, was Variantensimulationen und Modellvergleiche erleichtert.

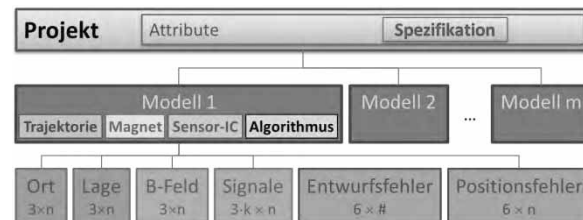


Bild 4 HDF5 Datenmodell für den Entwurf eines Magnet-Positionssensorsystems [15]. Datenelemente auf der untersten Ebene können bei klassischer Rasterdiskretisierung mehrere TB groß werden.

Um die Fehlerfreiheit eines durch ein parametrisiertes Modell abgebildeten Gesamtsystems nachzuweisen, sind Variationen der verschiedenen Systemparameter über den in der Spezifikation angegebenen Streubereichen zu untersuchen. Um dies leisten zu können, muss die Varianten-Simulation auf sehr wenige Stützstellen pro Freiheitsgrad und auf Minimal-, Nominal- und Maximalwert pro Streuparameter beschränkt werden. Mit steigender Parameteranzahl ergibt ihre Kombination eine immense Anzahl von Varianten, die mit einem vollständigen Systemmodell nicht in vertretbarer Zeit simulierbar sind.

3 Komplexitätsreduktion in der grafischen Darstellung

Selbst Personen mit großem Abstraktionsvermögen können sich Kurven, Flächen und Objekte in mehr als drei Dimensionen nicht vorstellen, erst recht nicht Abbildungen zwischen diesen. Auch trainierte Entwurfsingenieure benötigen Ausschnitte, Transformationen und Projektionen, um eine maximal drei, besser zwei- oder eindimensionale Sicht zu gewinnen, innerhalb derer sich Trends, Streuungen und Grenz-

verläufe/-überschreitungen schnell und sicher erfassen, verstehen und im Rahmen des Systemmodells interpretieren lassen. Dies gilt natürlich auch für Magnetfeldverläufe aus Sicht eines vektoriell messenden Sensorarrays auf einem integrierten Sensor-IC, wenn sich der Magnet auf einem relativ zum Sensor im Raum verschieblichen und ggf. sogar verdrehbaren Objekt befindet.

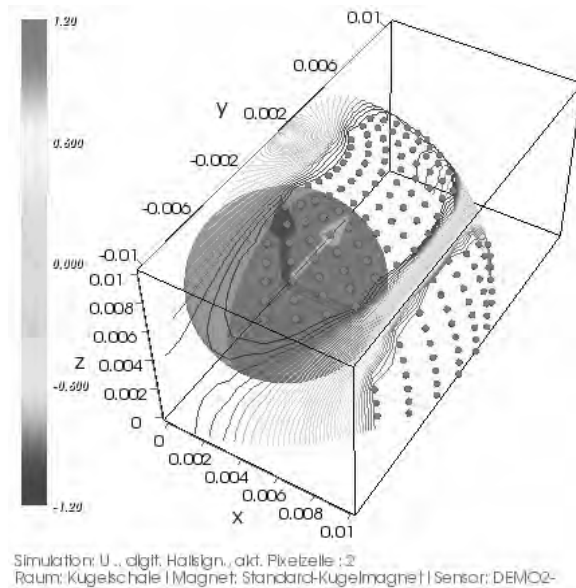


Bild 5 Fehlerindikator auf einer 2D-Trajektorie mit Überlagerung einer Signalkomponente, der Magnet ist symbolisch als Kugel dargestellt.

Um Magnetfeldwerte und Sensorsignalverläufe solcher Systeme in ihrem Bewegungsraum sichtbar zu machen, müssen spezielle Darstellungen gewählt werden, in denen dieser Raum diskretisiert und je nach Aufgabe stark reduziert wird. Diese Anforderung ergibt sich also sowohl aus entwurfsdidaktischen als auch aus rechentechnischen Anforderungen und Randbedingungen.

Beim Systementwurf für räumliche Positionssensorik kommen hierfür zunächst Projektions- und Hyper-ebenenverfahren in Betracht. Durch Fixierung bestimmter Bewegungsachsen kann die Plausibilität der Entwurfsidee einzeln auf jeder Bewegungsachse des Systems, aber auch summarisch für alle Achsen und Signale überprüft werden, äußerlich genauso wie es der Ingenieur vom klassischen Dreh- oder Weggeberentwurf gewöhnt ist.

Wird diese Projektion zudem vom Datenmodell nahtlos unterstützt, wird auch die Extraktion des entsprechenden Teildatensatzes aus dem in der Regel sehr

großen Gesamtdatenblock schnell und effizient sein und im Zusammenspiel mit einer leistungsfähigen Grafik ein schnelles Rendern ermöglichen. So kann der Entwerfer schnell über verschiedene Schnitte „blättern“ und einen Überblick über den gesamten Entwurfsraum gewinnen.

Ein ganz wesentlicher Aspekt beim modellbasierten Entwurf besteht darin, frühzeitig im Rahmen des Safety Engineering (siehe z.B. [17]) Entwurfsfehler festzustellen, also Verletzungen der in der Spezifikation gegebenen Anforderungen in einem oder mehreren Betriebszuständen. Um in einem komplexen Modell Fehler festzustellen, bedarf es geeigneter grafischer Hilfsmittel, um diese zunächst zu signalisieren, im weiteren aber auch genauer zu identifizieren und im speziellen Fall eines Positionsmesssystems auch zu lokalisieren und zu quantifizieren, um ggf. Hinweise für die entwurfstechnische Behebung des Fehlers zu erhalten.

Bild 5 zeigt einen Fehlermarker auf der Projektion einer dem Magnetfeld entsprechenden Signalkomponente auf die (virtuelle) Sensortrajektorie eines Scheinwerfer-Kipp- und Schwenkmoduls. Anhand Vorhandensein, Farbe und Ort der Fehlermarker kann der Entwerfer unmittelbar die Gültigkeit des Entwurfs beurteilen und erhält Korrektur-Hinweise. Bei mehr als zwei Freiheitsgraden ist diese Visualisierung allerdings nicht mehr leicht verständlich.

Für diese Aufgabe haben sich Projektionen, (gestapelte) Schnitte und sogenannte Parallelkoordinatendiagramme als geeignet erwiesen. Projektionen ermöglichen die Darstellung räumlicher Features in einer Ebene, wobei topologische Informationen erhalten bleiben, die einen Rückschluss z.B. auf Fehlerursachen ermöglichen (z.B. Magnet zu groß, zu nah, schief magnetisiert, fehlpositioniert, Einstellungsfehler des Sensor-ICs usw.).

Projektionen erlauben theoretisch eine Darstellung des Gesamtsystems ohne Informationsverluste, allerdings wird das grafische Ergebnis in der Praxis möglicherweise unübersichtlich und schwer zu interpretieren sein und durch Renderfehler und Überdeckungen beeinträchtigt. Daher ist es oft sinnvoll, den Entwurf nicht insgesamt, sondern auf Schnitten durch den Trajektorienraum zu analysieren, die auch in einer Serie (stacked slices) angeordnet sein können. Sinnvollerweise werden diese Schnitte so gelegt, dass bestimmte Freiheitsgrade auf einen oder mehrere diskrete Werte fixiert sind.

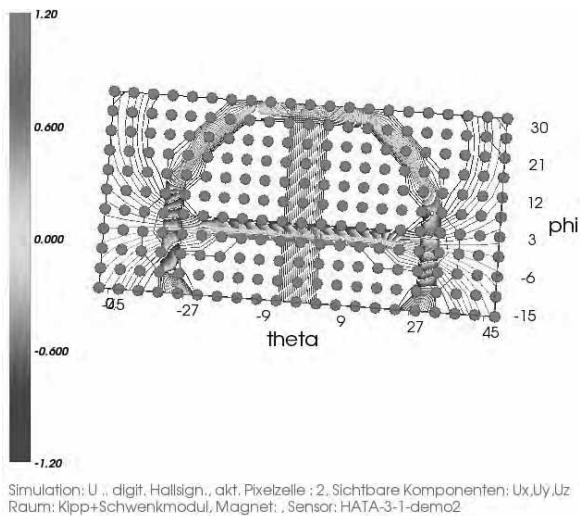


Bild 6 Fehlerindikator auf der Projektionsebene der 2D-Trajektorie einer Scheinwerferpositioniervorrichtung für alle drei Magnetfeldkomponenten eines 3D-Hallsensors auf einem Fraunhofer HallinOne®-IC, der Magnet selbst ist nicht dargestellt.

Bild 6 zeigt eine Projektion aller drei Magnetfeldkomponenten mit Fehlermarkern auf die Ebene zweier Freiheitsgrade. Trotz Verdichtung der Informationen bleibt die Darstellung übersichtlich, der 3D-Hall-Sensor übersteuert in mindestens einer Komponente über der gesamten Trajektorie – ein Entwurfsfehler der durch Anpassung der Verstärkung oder des Abstandes bereits im Vorentwurf behoben werden kann.

Eine andere Möglichkeit besteht in sogenannten Parallelkoordinatendiagrammen, die eine Rückverfolgung von Fehlerursachen über mehrere Freiheitsgrade hinweg erlauben (Bild 7). Etwaige Fehler in den Zielgrößen können auf Ausgangsgrößen, hier Freiheitsgrade, und darin wiederum auf Werte-Bereiche zurückgeführt werden.

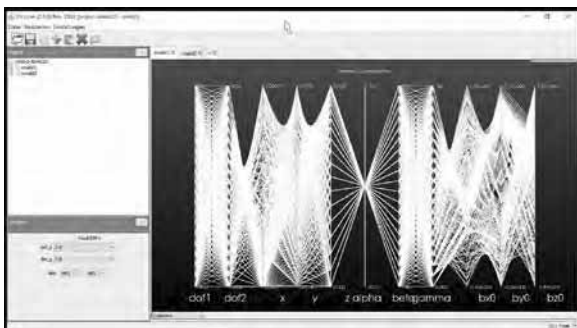


Bild 7 Parallelkoordinatendarstellung für die Berechnung des vektoriellen Magnetfelds $B:=(b_x,b_y,b_z)^T$ über acht Freiheitsgraden. Quelle: [15]

4 Anwendung

Die gezeigten Ansätze wurden für eine Entwurfssoftware für magnetbasierte räumliche Positionssensorensysteme evaluiert, die parallel zu entsprechenden Sensortechnologien und -systemen am Fraunhofer Institut für Integrierte Schaltungen entwickelt wurde [14]. Solche Positionssensorensysteme werden zum Beispiel in kompakten, verschleißfreien und robusten Bedienelementen im Industrie- und Fahrzeugbereich eingesetzt (Bild 9), eignen sich aber auch bei der Überwachung von beweglichen Verbindungen aller Art in diesen und anderen Anwendungsfeldern, etwa in der Medizintechnik.

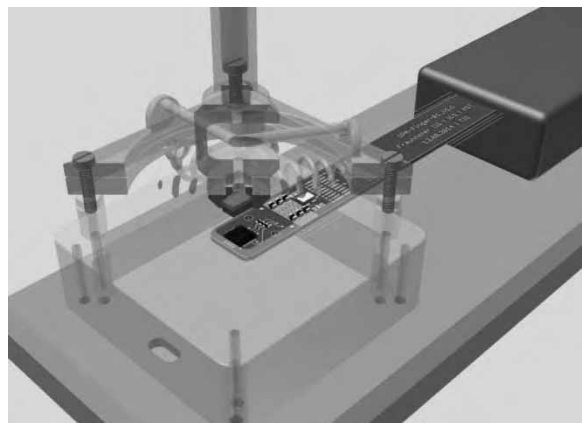


Bild 8 6D-Joystick. CAD-Visualisierung eines HallinMotion®-Demonstrators für ein Bedienelement mit 6 Freiheitsgraden

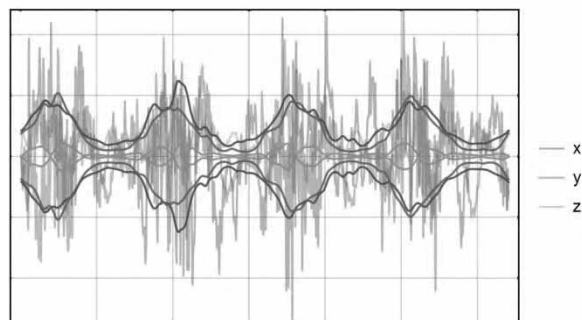


Bild 9 Analyse des Positionsfehlers eines Positionssensorsystems mit 6 Freiheitsgraden, Projektion auf die drei verschiebungsfreiheitsgrade x,y,z. Diskretisierung auf Lissajouskurven.

In Bild 9 wird der simulierte Positionsfehler der Ortskomponente eines solchen Systems anhand Projektionen und klassischer Entwurfstechniken untersucht. Die Lösung wurde als HallinMotion®-Demonstrator mittels 3D-Druck in Hardware umgesetzt (Bild 10), wobei ein handelsüblicher ARM4-Mikrokontroller

und ein angepasster HallinOne®-ASIC des Fraunhofer IIS eingesetzt wurden, der bereits in Produkten eines führenden Herstellers für robuste Bedienelemente auf dem Markt erhältlich ist [12].

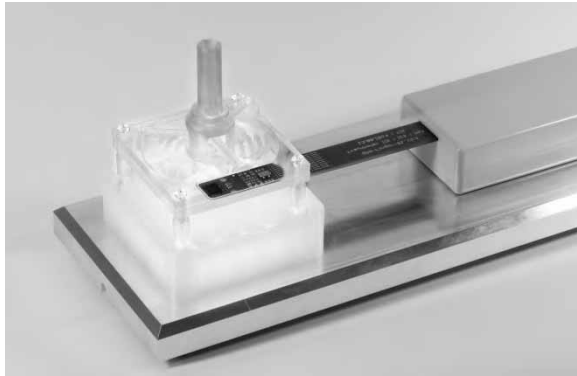


Bild 10 HallinMotion®-Demonstrator 6D-Joystick.

5 Zusammenfassung und Ausblick

Mehrachsig messende magnetische Sensoren, wie sie vom Fraunhofer IIS entwickelt und inzwischen in Größenordnungen über ASIC-Lösungen und lizenzierte Standard-ICs in den Markt eingeführt wurden, bieten großes Potential für leistungsfähigere und dabei preiswerte Lösungen in der berührungslosen Positionssensorik. Der Entwurf solcher Systeme ist jedoch mit herkömmlichen Mitteln extrem aufwändig und nur durch modellbasierte Entwicklung zu beherrschen. Auch dieses Entwurfsvorgehen setzt signifikante Ressourcen bei der Simulationshardware, verteilt simulierbare Datenmodelle und geeignete Visualisierungen voraus, die im vorliegenden Beitrag diskutiert und beispielhaft dargelegt werden. Insbesondere sind für eine tragfähige Produktentwicklung im Rahmen der Anwendung von FMEA- und FMECA-Methoden Variantensimulationen durchzuführen, die ohne cloudbasierte hochperformante Ressourcen nicht realisiert werden können. Die hier geschilderte Methodik gibt eine Richtung vor, um das Potential der neuartigen Sensorik für innovative Produkte nutzen zu können.

Die gezeigten Ansätze sind Teil einer Simulationsumgebung zum modellbasierten Entwurf räumlicher magnetischer Positionssensorik, die am Fraunhofer Institut für Integrierte Schaltungen prototypisch entwickelt wurde, wobei Ansätze aus der Simulation und der Big-Data-Analyse mit mathematischen und entwerfsdidaktischen Lösungen verbunden wurden.

6 References

- [1] M. Hackner, H. Hohe, J. Sauerer, R. Janke, M. Besemann. *Multi 3D Hall Sensor for Gradient Measurement*. Proc. European Magnetic Sensors & Actuators Conf. (EMSA), Caen, 2008
- [2] H. Hohe et.al. *Robust position measurement systems based on integrated 3-D magnetic field sensors*, in Heuberger et al., *Microelectronic systems. Circuits, systems and applications*, Springer, Berlin, 2011, pp.41-48
- [3] AS5410 3D-Hall Absolute Linear Position Sensor, <http://ams.com/eng/Products/Magnetic-Position-Sensors/Linear-Position/AS5410>
- [4] HAL37xy 3D HAL Sensorfamilie der zweiten Generation, <https://www.micronas.com/de/produkte/direktwinkel-sensoren/hal-37xy>
- [5] R. Popovic: *The Vertical Hall-Effect Device*, IEEE Electronic Device Letters 5 (1984), p. 357
- [6] M. Hackner, R. Ernst, H. Hohe, M. Stahl-Offergeld, U. Schlag. *Vertikaler Hall-Sensor*. Deutsches Patent DE102006017910A1, 25.10.2007
- [7] A. Wilde; G. Lantzsch, T. Obenaus, P. Schneider, H. Hohe, J. Sauerer. *Design of Position Sensor Systems based on 3D Hall probe*. 13th SENSOR Conf., Nürnberg, 2007.
- [8] H.Hohe, J. Sauerer. *Robuste Positionsmessung in Hydraulik- und Pneumatikzylindern*. Mechatronik 11-12, 2008, pp. 60-63.
- [9] M. Hackner, H. Hohe, J. Sauerer. *xD-Hallsensorik zur mehrdimensionalen Positionsmessung*. In Gesch, H.: Landshuter Symposium für Mikrosystemtechnik 2008, p. 93.
- [10] J. Bretschneider; A. Wilde. *Entwurfsumgebung für hallbasierte 3D-Positionssensorsysteme*. In Deatcu, C., Hrsg.: Modellierung, Regelung und Simulation in Automotive und Prozessautomation, 4. ASIM STS Workshop, Wismar, 29.-30. Mai 2008, ARGESIM Report, Vienna, 2008, pp. 235-240.
- [11] M. Hackner, H. Hohe, M. Stahl-Offergeld. *An integrated nine-dimensional Hall-Gradient-Sensor*. Proc. Int. Conf. on Sensors, Technologies, Electronics and Applications (SENSOR), 2009, pp.23-28

- [12] W. Neudeck, J. Bretschneider, D. Cichon, H. Hohe. *Performanz mehrachsiger magnetischer Positionsmessung als eingebettetes Smart System*, 18. GMA/ITG Fachtagung Sensoren und Messsysteme, Nürnberg, 2016
- [13] M. Hackner. *Mehr Freiheitsgrade, höhere Störsicherheit*, Elektor Business Magazin Mai/Juni 2016, S.4-8
- [14] J. Bretschneider, A. Wilde, P. Schneider, H. Hohe, U. Koehler. *Design of multi-dimensional magnetic position sensor systems based on HallinOne® technology*. In: Industrial Electronics (ISIE), IEEE International Symposium, Bari, 2010, S. 422–427
- [15] X. Wang. *Prototypische Entwicklung statistischer und visueller Methoden zur Simulation und Bewertung von Parameterschwankungen auf den Signalfluss von magnetbasierten Positionssensorsystemen*, Diplomarbeit, TU Dresden, 2015
- [16] *FMEA - Fehlermöglichkeits- und Einflussanalyse*. Deutsche Gesellschaft für Qualität. DGQ-Band 13-11, Beuth Berlin, 5. Auflage 2012
- [17] Methoden des Safety Engineering – Design-FMEA. <http://www.systema-gmbh.de/methoden/fmea/design-fmea.html>

Parallel multi-agent smart grid simulation

Christian Kuschel, Ulrich Rude
Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl für Systemsimulation (Informatik 10)
Cauerstr. 11, 91058 Erlangen
christian.kuschel@fau.de

The approach in this paper renders it possible to simulate large-scale smart grids by efficient parallel computations. This permits a detailed analysis of the consumption behaviours, efficiency and impact of green energies, and self-sustainability of a smart grid. The smart grid is modelled as a multi-agent system. Each agent represents a building which is optimally controlled. That is, an agent meets its prescribed energy demand by trading energy or applying devices, e.g. solar panels and fuel cells, minimising its costs. A cooperative bargaining game is devised in which the agents participate to obtain a global optimal solution. In this paper, this inherently serial bargaining game is parallelised. The parallelisation is necessary to be able to deal with the large amount of data and computations which need to be performed. In the experiments the validity of the presented approach is shown and as a proof of concept a large smart grid of over 40 million agents is simulated.

1 Introduction

With technologies for integrating energy generation and storage in residential buildings, the notion of smart grid was derived [1, 2]. Smart grids have been in the focus of research because of various aspects. Among others, the energy generation and storage devices themselves, e.g. combined heat and power generation and energy storages, and the coordination of them pose many challenges [3, 4, 5, 6, 7]. The focus of this paper, however, is on the interactions of participants of the smart grid.

The smart grid is modelled by a multi-agent system. Each agent optimally controls a residential building and is able to generate and store energy with the appropriate devices [8, 9]. Based on the models of [10, 11], the agents are able to communicate with each other to participate in a cooperative bargaining game which possesses a unique Nash equilibrium. In addition, the pricing scheme from [14] is applied to obtain fair energy prices during bargaining.

The efforts of this paper go beyond the previously mentioned approaches of simulating smart grids: The

smart grid model used in this paper extends the model [10] by incorporating a more detailed agent model which was published in [11]. In both publications a serial, weak coupling approach for solving the bargaining game is realised. The goal of this paper is enable simulations of realistically sized, large smart grids. Therefore the serial approach is parallelised for a distributed memory architecture, because a single computer can no longer deal with the intended complex and memory consuming simulation.

With this parallelisation, detailed information about the optimal, efficient usage of energy can be obtained. In particular, the efficient usage of green energies and the self-sustainability of the smart grid can be examined. To that end, a smart grid of more than 40 million agents, which is of equivalent size of Germany [13], is simulated in parallel. To the knowledge of the authors, this is the largest smart grid simulation performed until today.

In section 2, a brief overview about the applied model and the serial algorithmic approach is given. The parallelisation of the cooperative bargaining game is presented in section 3. Finally, the experimental validation and performance results are presented in section

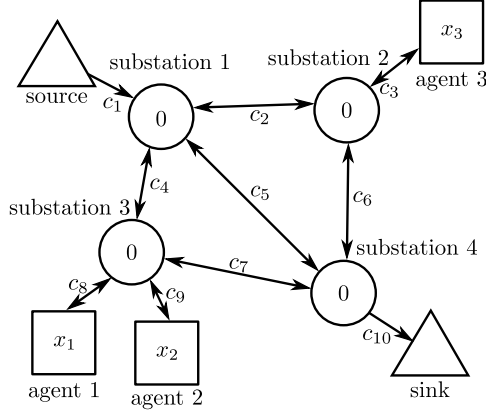


Figure 1: Schematic illustration of a smart grid with line capacities. The differently shaped nodes indicate different roles. The values within nodes specify the required net load.

4. The test results validate the approach and show the parallel performance in strong scaling experiments.

2 Smart grid model

The smart grid model and the solution approach are essentially based on [10, 11, 14]. In the following a brief summary of the method published in [11] is given. Consider a smart grid as depicted in figure 1. In this paper the line capacities of figure 1 c_i are assumed to be infinite, so that the grid does not pose any restrictions and can be neglected.

A smart grid consists of agents and a grid operator which are capable of communicating with each other. The grid operator uses power plants to generate energy and substations to distribute the energy. Since the line capacities are infinite, the substations can be neglected.

Buildings are optimally controlled by agents by a solving optimisation problem. The optimisation problem takes devices, like e.g. central heating, fuel cells, refrigerators, into account, and enforces that a prescribed energy demand is met. This is incorporated in constraints of the optimisation problem. The constraints define a non-empty, convex feasible set Ω_n for each agent n . The details about the feasible set are not relevant for the considerations below, but can be found in [11]. It is sufficient to assume that the set is non-empty (to ensure feasibility) and that the choice

of the optimal solution is restricted in one way or the other (excluding a trivial solution).

The agents' optimisation problems are solved to minimise their respective costs. Since the goal is to simulate millions of agents, it is assumed that the agents form a market which determines the price with respect to demand and supply. Therefore, the agents can adapt their demand to the prices and influence the prices by changing the demand. This adaption is modelled by a cooperative bargaining game to minimise the agents' costs which is presented below.

The total incurring costs \mathcal{C} of the budget balanced grid operator to supply the agents is defined by

$$\mathcal{C} := \left(\sum_{n=1}^N x_n \right)^2,$$

where N is the number of agents and x_n is the net load of agent n . The net load subsumes an agent's demand and supply in one variable. If $x_n > 0$ it represents a demand, if $x_n < 0$ it represents a supply. The cost function \mathcal{C} can be understood as the squared deviation of a self-sustaining smart grid. If $\mathcal{C} \neq 0$ the grid operator must intervene and costs incur which must be covered by the agents.

Based on the fair pricing scheme suggested by [14], the total costs are split among the agents proportionally to their contribution to the total costs. This yields the individual incurring costs \mathcal{C}_n of agent n with respect to all other agents' loads x_{-n} :

$$\mathcal{C}_n(x_n; x_{-n}) = \overbrace{(x_n + x_{-n})}^{\text{price}} \cdot \overbrace{x_n}^{\text{net load}}, \quad (1)$$

with $x_{-n} := \sum_{\substack{j=1 \\ j \neq n}}^N x_j$.

So each agent minimises (1), the local objective function, in an optimisation problem. In particular, $\mathcal{C} = \sum_{n=1}^N \mathcal{C}_n(x_n; x_{-n})$.

In order to obtain a global optimum, a bargaining game is devised:

- **Players:** All agents in the smart grid.
- **Strategies:** Player n computes its best response $x_n = \arg \min \mathcal{C}_n(x_n, x_{-n})$ s.t. $x_n \in \Omega_n$.
- **Costs:** \mathcal{C}_n for agent n .

For the considered game it was shown in [10] that a unique Nash equilibrium exists. Therefore, the locally computed optima by the agents lead to a global optimum. This Nash equilibrium is characterised by each player obtaining its optimum. Moreover, if one player deviates from the optimum solution, the costs increase for that player.

Since the costs of one agent are dependent on the net loads of all other agents, as indicated in (1), the game is played in rounds. In each round, all agents adjust their played net load to their respective current best responses after another. This is done until no adjustments are carried out by all agents. This procedure of the game is summarised in algorithm 1. In the subsequent section, this algorithm is parallelised.

Algorithm 1 Serial bargaining algorithm.

```

1:  $k \leftarrow 0$ 
2: Set  $x^k$  to initial total grid load
3: repeat
4:    $\delta \leftarrow 0$ 
5:    $y^k \leftarrow x^k$ 
6:   for each agent  $n$  in the smart grid do
7:      $x_{-n}^k \leftarrow y^k - x_n^k$ 
8:      $x_n^{k+1} \leftarrow \arg \min \mathcal{C}_n(x_n^{k+1}; x_{-n}^k)$  s.t.  $x_n^{k+1} \in \Omega_n$ 
9:      $y^k \leftarrow x_n^{k+1} + x_{-n}^k$ 
10:     $\delta \leftarrow \delta + |x_n^k - x_n^{k+1}|/N$ 
11:   end for
12:    $x^{k+1} \leftarrow y^k$ 
13:    $k \leftarrow k + 1$ 
14: until  $\delta < \varepsilon$ 

```

3 Parallel bargaining game

In order to parallelise algorithm 1, the best response computation is distributed on processes which run in parallel. In general, more agents than processes are used, therefore one process is assigned multiple agents for computation. Thus each process computes the best responses of its agents serially as in algorithm 1. This can be interpreted as computing a single best response for each process, independent of the actual number of agents associated with the process. So instead of referring to the agents' best responses, below the processes' best responses are considered.

An essential property of the game in algorithm 1 is the successive best response computation. Evidently, when processes compute the best responses in parallel as described above, this property is violated. Consequently, for the parallel approach a synchronisation scheme needs to be devised which is applied after the processes' best response computation to compensate the lack of the successive best response computations.

Consider again algorithm 1. Since the best response computation is the solution to the agents' optimisation problems, this entails that the process' costs never increase. This non-increase property is expressed as

$$\mathcal{C}_n(x_n^{k+1}; x_{-n}^k) \leq \mathcal{C}_n(x_n^k; x_{-n}^k) \quad (2)$$

and must hold for every iteration k . This also implies that $|x_n^{k+1}| \leq |x_n^k|$ for all $n = 1, \dots, N$. In fact, the objective value monotonically decreases until a global optimum has been found [10, 15].

The monotonic decrease in (2) holds for each process individually. However it must also hold for the total incurring costs in the smart grid (as in the serial case), therefore

$$\mathcal{C}^{k+1} \leq \mathcal{C}^k := \sum_{n=1}^N \mathcal{C}_n(x_n^k; x_{-n}^k). \quad (3)$$

In the parallel approach, a monotonic decrease in the total costs is obtained by computing the best response with respect to an auxiliary term \bar{x}^k , which represents the total net load in the grid. Let

$$\bar{x}_{-n}^k := \bar{x}^k - x_n^k.$$

So instead of minimising $\mathcal{C}_n(x_n^{k+1}; x_{-n}^k)$, $\mathcal{C}_n(x_n^{k+1}; \bar{x}_{-n}^k)$ is minimised.

The auxiliary term is defined as

$$\bar{x}^{k+1} = \frac{N-1}{N^2} \cdot \sum_{n=1}^N (x_n^{k+1} + \bar{x}_{-n}^k) + \frac{1}{N} \cdot \sum_{n=1}^N x_n^{k+1}. \quad (4)$$

It can be shown that (3) holds for this choice of \bar{x}^k . To show that $(\bar{x}^k)^2 \geq (\bar{x}^{k+1})^2$, the reduction γ is introduced to yield $(\bar{x}^k - \gamma)^2 = (\bar{x}^{k+1})^2$. Applying the definition of \bar{x}^{k+1} from (4) reads

$$\begin{aligned} (\bar{x}^k - \gamma)^2 &= \left(\bar{x}^k - \frac{1}{N} \bar{x}^k \right. \\ &\quad \left. + \frac{N-1}{N^2} \cdot \sum_{n=1}^N \left(\frac{2 \cdot N - 1}{N-1} \cdot x_n^{k+1} - x_n^k \right) \right)^2. \end{aligned}$$

Obviously,

$$\gamma = \frac{1}{N} \bar{x}^k - \frac{N-1}{N^2} \cdot \sum_{n=1}^N \left(\frac{2 \cdot N - 1}{N-1} \cdot x_n^{k+1} - x_n^k \right) \quad (5)$$

must be within $0 \leq \gamma \leq 2 \cdot \bar{x}^k$ for $\bar{x}^k \geq 0$ or $0 \leq -\gamma \leq 2 \cdot \bar{x}^k$ for $\bar{x}^k \leq 0$ to yield a decrease in the objective value.

In the following the case of $\bar{x}^k \geq 0$ is elaborated. The case $\bar{x}^k \leq 0$ can be derived analogously. To show that $\gamma \geq 0$, (5) is rewritten as

$$\frac{1}{N} \cdot \left(\bar{x}^k - \sum_{n=1}^N x_n^{k+1} \right) - \frac{N-1}{N^2} \cdot \sum_{n=1}^N \left(x_n^{k+1} - x_n^k \right).$$

Both terms in parenthesis represent the change in the objective value from iteration k to $k+1$. The term in the first parenthesis is the change with respect to the auxiliary term \bar{x}^k whereas the term in the second is the change of the best responses $\sum_{n=1}^N x_n^k$. Both expressions in parenthesis are negative, but due to the averaging of \bar{x}^k the term in the first parenthesis is more inertial to change than the second term. Thus the absolute value of the term in second parenthesis is larger so that $\gamma \geq 0$ holds.

(5) is inserted to $\gamma \leq 2 \cdot \bar{x}^k$ to show this inequality also holds:

$$(2 \cdot N - 1) \cdot \bar{x}^k \geq \frac{2 \cdot N - 1}{N} \sum_{n=1}^N x_n^{k+1} - \frac{N-1}{N} \sum_{n=1}^N x_n^k.$$

Recall that $|x_n^{k+1}| \leq |x_n^k|$ from (2) is implied, so

$$(2 \cdot N - 1) \cdot \bar{x}^k \geq \left(\frac{2 \cdot N - 1}{N} - \frac{N-1}{N} \right) \sum_{n=1}^N x_n^k = \sum_{n=1}^N x_n^k$$

is obtained. In the course of iteration, \bar{x}^k approximates $\sum_{n=1}^N x_n^k$. Consequently, both terms are within the same order and the estimate

$$\bar{x}^k \leq \sum_{n=1}^N x_n^k \leq 2 \bar{x}^k < (2N-1) \cdot \bar{x}^k \Rightarrow \left| \bar{x}^k - \sum_{n=1}^N x_n^k \right| \leq |\bar{x}^k|$$

holds. $|\bar{x}^k - \sum_{n=1}^N x_n^k|$ is generally small and decreases with increasing number of iterations k , since the x_n^k are

computed as solution to optimisation problems and used to construct \bar{x}^k . Therefore (4) yields a monotonic decrease.

With the monotonic decrease shown in each iteration, it yet needs to be validated that the sequence of \bar{x}^k converges to the optimum. This is done by showing convergence of $\bar{x}^k \rightarrow \sum_{n=1}^N x_n^k$ for $k \rightarrow \infty$. By expressing \bar{x}^k with respect to the best responses x_n^{k+1} and \bar{x}^k the following recurrence equation for a given initial value $\bar{x}^0 = \sum_{n=1}^N x_n^0$ is obtained:

$$\begin{aligned} \bar{x}^{k+1} &= \left(\frac{N-1}{N} \right)^{k+1} \cdot \bar{x}^0 + \frac{2 \cdot N - 1}{N^2} \cdot \sum_{n=1}^N x_n^{k+1} \\ &\quad + \frac{N-1}{N^2} \cdot \sum_{m=1}^k \left(\frac{N-1}{N} \right)^{k-m+1} \cdot \sum_{n=1}^N x_n^m. \quad (6) \end{aligned}$$

Since the sequence \bar{x}^k is monotonically decreasing, i.e. converging to a finite limit, the subsequence x_n^m is also convergent. So the sum in (6) containing x_n^m can be bound by a geometric series (since $|1/N| < 1$) by replacing x_n^m with x_n^k . The geometric series can be rewritten in a closed form. Using $x^* = \sum_{n=1}^N x_n^k$ for $k \rightarrow \infty$, from (6)

$$\lim_{k \rightarrow \infty} \bar{x}^{k+1} = 0 + \left(\left(\frac{N-1}{N} \right)^2 + \frac{N-1}{N^2} + \frac{1}{N} \right) \cdot x^* = x^*$$

is obtained. Consequently, using an auxiliary term as defined in (4) implies convergence to the optimum. Thus, the serial bargaining game can be parallelised by this method.

Concluding this section, the parallel bargaining is summarised in algorithm 2. The new variable P in line 15 denotes the total number of agents in the entire smart grid. This number is required to check the termination criterion in line 17.

Each process performs the computations for its assigned agents independent of other processes on a distributed memory architecture. The “allreduce”-operation is the only operation which requires inter-process communication. It acts as barrier and all processes wait for each other at this point. When it is called, each process collects the passed value from the other processes and sums up the values, respectively. The return value of this function is the same for all processes.

Although in algorithm 2 it is described that three communication steps in lines 14 to 16 are required, all re-

Algorithm 2 Parallel bargaining algorithm.

```

1: Set  $X$  to the initial net load
2:  $X \leftarrow \text{allReduce}(X, +)$ 
3:  $k \leftarrow 0$ 
4: repeat
5:    $Y, \delta \leftarrow 0$ 
6:   for each agent  $j$  of process  $n$  do
7:      $X \leftarrow X - x_j^k$ 
8:      $x_j^{k+1} \leftarrow \arg \min \mathcal{C}_j(x_j^{k+1}; X)$  s.t.  $x_j^{k+1} \in \Omega_j$ 
9:      $X \leftarrow X + x_j^{k+1}$ 
10:     $Y \leftarrow Y + x_j^{k+1}$ 
11:     $\delta \leftarrow \delta + |x_j^k - x_j^{k+1}|$ 
12:   end for
13:    $X \leftarrow \text{allReduce}(X, +)$ 
14:    $Y \leftarrow \text{allReduce}(Y, +)$ 
15:    $\delta \leftarrow \text{allReduce}(\delta, +)/P$ 
16:    $X \leftarrow (N-1) \cdot X/N^2 + Y/N$ 
17:    $k \leftarrow k+1$ 
18: until  $\delta < \varepsilon$  and  $|X - Y|/N < \varepsilon$ 

```

ductions can be fused in only one reduction by writing all three variables in one buffer and perform one reduction on the buffer instead of one per variable.

4 Simulation results

Experiments were conveyed to validate the approach of section 2 and also analyse the parallel performance of the presented bargaining algorithm. The algorithms were implemented in C++ using the GNU linear programming kit (glpk) [16] as central solver for optimisation problems and Open MPI [17] for the parallelisation and interprocess communication.

In section 4.1, the validity of the approach from section 3 is shown. The parallel performance is analysed with respect to strong scaling in section 4.2.

4.1 Validation of the approach

The validation was carried out for a reference smart grid with 10,000 agents. The smart grid is simulated for 100 time steps, which represents roughly one day in 15 minutes intervals. 200 bargaining iterations were performed in each case.

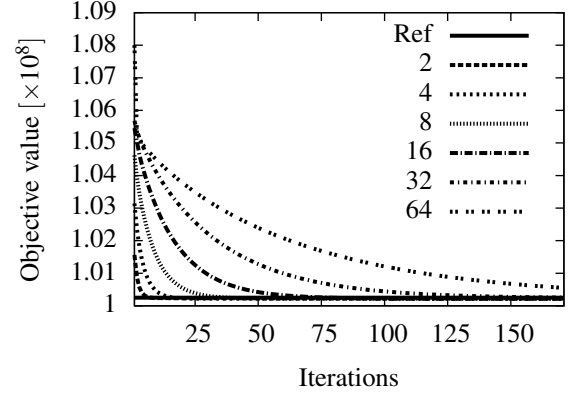


Figure 2: Convergence of parallel bargaining for 2 to 64 processes, and the reference solution.

As shown in figure 2, all plots converge to the reference result, which was computed serially by the approach of [11]. So the approach as described in algorithm 2 allows to compute the optimum in parallel. It can be observed that, although convergent, the more processes are used, the slower the convergence. This can be traced back to the coefficients dependent on N in (6) which tend to 1 for large N . Therefore the more processes are used, the slower the convergence rate.

Although the plot shows that the number of required iterations to reach the optimum increases with increasing number of processes, the total runtime does not increase. These details of the performance are presented in the next section.

4.2 Parallel performance

The efficiency of the implementation is considered in the following. Therefore the strong scaling behaviour of the parallel bargaining approach is presented. The termination criterion was set to a maximal average change of net load per agent per time step of 10^{-2} kW h. Furthermore, the agents are uniformly distributed on the processes in a round robin fashion.

The strong scaling test scenario is the same as in section 4.1. It was computed on the local compute cluster of the department of computer science 10 of Friedrich-Alexander-Universität Erlangen-Nürnberg [18]. It has eight compute nodes each of which consists of four Intel Xeon E7-4830 (eight cores each) at 2.4 GHz and

# Processes	1	2	4	8	16	32
# iterations	3	4	5	5	6	6
it. time [s]	798	349	181	81	37	21
Speed-up	1	1.7	2.8	6.0	10.9	20.1

Table 1: Strong scaling results for 10,000 agents, showing the number of iterations, the averaged iteration time and the speed-up with respect to the serial computation.

256 GB RAM per node. The serially computed scenario took 42.5 minutes to finish.

As already mentioned, in figure 2 it can be observed that the more processes are used the more iterations are required to attain the optimum. This, however, does not entail a longer total runtime, as can be seen in table 1. In fact, since the optimisation problems of the agents are uniformly distributed, the time per iteration approximately halves when using twice as much processors. The number of iterations hardly increases with increasing number of processes. Consequently, the parallelisation has a greater impact than the number of increased iterations, therefore the total runtime decreases.

In table 1, it is stated that there is close to perfect speed-up up to eight processes. Beyond that the speed-up of the computations increases less. The reason for that are the increased communication in addition to the fluctuating times to solution of an agent's optimisation problem. For the considered scenario the latter times are between 2 s and 0.05 s with a median of 0.07 s. Obviously, most of the agents' optimisation problems are solved quickly and few slowly. If the optimisation problems are distributed among the processes in a way that the average time to solve all optimisation problems per process is approximately equal, a good speed-up can be expected.

For the considered scenario more than 1,250 agents per process yielded excellent scaling. Fewer agents caused some processes to wait for other, slower processes reducing the effects of parallelisation. This effect is expected to be alleviated by using dynamic instead of the static load balancing used in the experiments. Nevertheless the parallel efficiency does not drop below 60 % in the experiments, which indicates good scalability.

In addition to the strong scaling computations, one large simulation as proof of concept for the realisa-

tion of a nation-sized smart grid was carried out. Algorithm 2 was executed on the Emmy cluster of Regionales Rechenzentrum Erlangen [19] simulating 40 million agents on 1280 processes. This was the largest simulation with respect to the physical limitations of the compute nodes and it completed within 16 hours. This scenario corresponds to a smart grid in the order of Germany, which is the country with the most private households in the EU [20]. To the knowledge of the authors, this is the largest simulated smart grid until today.

This carried out experiment merely hints at the potential of this approach. Being able to simulate smart grids representing countries, analysis of efficiency and sustainability can be performed. Especially if investments pay off can be analysed. From the simulation aspect, the weak scaling has yet to be examined. This is necessary to determine the efficiency of this approach with increasing number of agents on an increasing number of processes. This is analysed in the current work in progress.

5 Conclusion and future work

In this paper, a cooperative bargaining game was successfully parallelised. Since the serial algorithm relies on successive best response computation, it is necessary to introduce an auxiliary term to ensure convergence of the approach.

Almost perfect speed-up is reached in the strong scaling scenarios, in which 10,000 agents are considered, if the average time to solution per process is almost equal. This was commonly the case for more than 1,250 agents per process in the experiments.

In a next step the parallel efficiency needs to be evaluated for weak scaling scenarios. When simulating large smart grids, the presented algorithm is required to scale well with the problem size. In conclusion, the largest run optimised more than 40 million agents, a problem size equivalent to a country. Being able to simulate such a large grid, estimates on a realistic scale about self-sufficiency, demand and supply and efficiency of green energies can be fleshed out.

References

- [1] Md M. Biswas et al. "Towards Implementation of Smart Grid: An Updated Review on Electrical Energy Storage Systems". In: *Smart Grid and Renewable Energy* 4.4 (2013), pp. 122–132.
- [2] A. Monti and F. Ponci. "Power Grids of the Future: Why Smart Means Complex". In: *Complexity in Engineering, 2010. COMPENG '10*. 2010, pp. 7–11.
- [3] Michiel Houwing, Rudy R. Negenborn, and Bart De Schutter. "Demand response with micro-CHP systems". In: *Proceedings of the IEEE* 99.1 (Jan. 2011), pp. 200–213.
- [4] Mohammad T. Arif, Amanullah M.T. Oo, and A.B.M. Shawkat Ali. "Role of Energy Storage on Distribution Transformer Loading in Low Voltage Distribution Network". In: *Smart Grid and Renewable Energy* 4.4 (2013), pp. 237–251.
- [5] Italo Atzeni et al. "Demand-Side Management via Distributed Energy Generation and Storage Optimization". In: *Smart Grid, IEEE Transactions on* 4.2 (2013), pp. 866–876.
- [6] Marco Giuntoli and Davide Poli. "Optimized Thermal and Electrical Scheduling of a Large Scale Virtual Power Plant in the Presence of Energy Storages". In: *IEEE Transactions on Smart Grid* 4.2 (2013), pp. 942–955.
- [7] Peter M. van de Ven et al. "Optimal Control of End-User Energy Storage". In: *IEEE Transactions on Smart Grid* 4.2 (2013), pp. 789–797.
- [8] Jürgen Karl. *Dezentrale Energiesysteme: neue Technologien im liberalisierten Energiemarkt*. Oldenbourg, 2006.
- [9] J. Ekanayake et al. *Smart Grid: Technology and Applications*. Wiley, 2012.
- [10] A.-H. Mohsenian-Rad et al. "Autonomous Demand-Side Management Based on Game-Theoretic Energy Consumption Scheduling for the Future Smart Grid". In: *IEEE Transactions on Smart Grid* 1.3 (2010), pp. 320–331.
- [11] Christian Kuschel, Harald Köstler, and Ulrich Rüde. "Multi-Energy Simulation of a Smart Grid with Optimal Local Demand and Supply Management". In: *Smart Grid and Renewable Energy* 6 (2015), pp. 303–315.
- [12] Chun-Hao Lo and Nirwan Ansari. "Decentralized Controls and Communications for Autonomous Distribution Networks in Smart Grid". In: *IEEE Transactions on Smart Grid* 4.1 (2013), pp. 66–77.
- [13] Bundesamt für Statistik. *Staat & Gesellschaft – Haushalte & Familien – Statistisches Bundesamt (Destatis)*. www.destatis.de/DE/ZahlenFakten/GesellschaftStaat/Bevoelkerung/HaushalteFamilien/HaushalteFamilien.html. Feb. 2016.
- [14] Zahra Baharlouei et al. "Achieving Optimality and Fairness in Autonomous Demand Response: Benchmarks and Billing Mechanisms". In: *IEEE Transactions on Smart Grid* 4.2 (2013), pp. 968–975.
- [15] Siegfried K. Berninghaus, Karl-Martin Ehrhart, and Werner Güth. *Strategische Spiele*. Springer-Lehrbuch. Springer-Verlag Berlin Heidelberg, 2006.
- [16] GLPK. *GNU Linear Programming Kit*. www.gnu.org/software/glpk. July 2015.
- [17] The Open MPI Project. *Open MPI: Open Source High Performance Computing*. www.open-mpi.de. Mar. 2016.
- [18] Computer Science 10, Friedrich-Alexander-University Erlangen-Nuremberg. *HPC Cluster*. www10.cs.fau.de/en/research/hpc-cluster. Feb. 2016.
- [19] Regionales Rechenzentrum Erlangen. *RRZE – Emmy Cluster*. www.rrze.fau.de/dienste/arbeiten-rechnen/hpc/systeme/emmy-cluster.shtml. Feb. 2016.
- [20] Bundesamt für Statistik. *EU – Privathaushalte in den Mitgliedsstaaten 2014*. de.statista.com/statistik/daten/studie/74720/umfrage/anzahl-der-privathaushalte-in-den-laendern-der-eu. Mar. 2016.

VOMAS for Validation of Agent-based Models – Requirements and Application

Julian Ruths¹, Niki Popper¹, Florian Miksch²

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

²dwh simulation services

julian.ruths@hotmail.com

Agent-based models are difficult to validate. This is mostly due to their complex structure and emergent behavior. Therefore, new validation methods targeted at agent-based models are developed. One of those methods uses a virtual overlay multi-agent system (VOMAS), which is able to monitor the simulation and its agents, check for violations of model assumptions and if necessary can log information for later use. In this paper we present a detailed concept of VOMAS components with a platform-independent design, which offers easy and flexible usability on part of the user. To test the VOMAS design, it is applied to an existing agent-based model. The developed VOMAS proves to be a very flexible tool for validating agent-based models, allowing the user to choose which simulations agents should be observed and what attributes need to be monitored and logged. Applied to the test case model, the VOMAS is able to gather new information giving a detailed insight into the agent behavior that can be used to further validate the model at hand.

1 Introduction

Agent-based models are becoming increasingly popular in the field of modeling and simulation. They are able to simulate very complex behavior and thanks to the ongoing increase of computing power, they can be applied to real world problems. Like all other models, agent-based models need to be validated before being included in the decision making process. Validation is tasked with determining if a model is able to answer given research questions in a satisfactory manner. In other words validation tries to answer if the *right* model is being developed? The process of validation is never done but is an ongoing crucial part in the modelling process. A model that has not been properly validated can not be included in the decision making process and can therefore for all its intends and purposes be deemed useless.

In spite of their intuitiveness, agent-based models are very difficult to validate. Some general validation methods can be applied, but methods specifically targeted at agent-based models need to be developed, tested and their usage properly documented. A new approach for validating agent-based models uses a virtual overlay multi-agent system (VOMAS), which is the focus of this work.

This paper is structured as follows: First we give a brief overview of other validation methods for agent-based models and introduce the reader to the

VOMAS concept, outlining its intended usage. In section 3 we outline the main objectives we set out to accomplish throughout this work. In section 4 we (1) discuss the problems of validating agent-based models, (2) list general validation methods that can still be applied to agent-based models and (3) guide the reader through the process of the development of a possible VOMAS design. Our results are presented in section 5 followed by a case study. The paper closes with a short discussion.

2 Related Research

Validation of agent-based models is faced with a number of challenges, which are described further on in this paper. There have however been some developments in the field of validation of agent-based models, of which we would like to mention two in particular:

A new validation method developed by Klügl in [1] combines several general validation methods into its own procedure. Starting with a run-able model, face validation, sensitivity analysis, calibration, plausibility checks and statistical validation are applied in a step-by-step approach, providing the modeling expert with a deployable model in the end.

Another approach developed by Klügl and Louloudi presented in [2], uses 3D visualisation to provide the user with a detailed insight into the simulation. This

method requires a 3D engine to be incorporated in the simulation model. The user can then either solely observe the simulation through the “eyes” of a simulation agent, or if need be also control the agent’s behavior.

2.1 The VOMAS approach

The concept of a VOMAS was first introduced in 2009 by Niazi et al. in [3] and subsequently applied to a forest fire model in [4]. It can comprise ‘*various types of agents, which form an overlay on top of the agent-based simulation model that needs to be validated*’ [3]. Figure 1 outlines the basic idea of the VOMAS concept. The arrows indicate the flow of information. It is vital that information can only flow out of the simulation layer. Agents within the actual simulation can never know of the existence of the agents in the VOMAS layer.

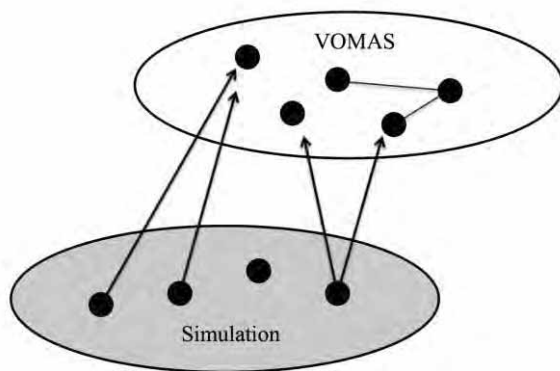


Figure 1. Interaction between the VOMAS and the simulation layer

In short, a VOMAS can observe the simulation and its agents, gather information on its/their behavior and log information for later analysis. This is achieved by several different types of VOMAS agents. Muazi et. al distinguish between 8 different components that make up a VOMAS: (1) VO-Agents have access to the simulation agents and represent the link between the VOMAS and the actual simulation, (2) Watch-Agents can monitor the agents’ behavior and attributes, (3) Constraint-Agents can check for violations of model assumptions, (4) a Console-Agent can provide the user with information during runtime, (5) a Logger-Agent saves information for later analysis, (6) a VO-Manager acts as the central communication unit within the VOMAS and can also gather information about the simulation, (7) the Sim-Agents represent the actual simulation agents, (8) Log

Entries act as standardized units of information that are sent to the Logger-Agent who then saves the information contained within those log entries.

The authors also stress the importance of an early involvement of a Subject Matter Expert (SME) in the development of a VOMAS. A SME is a person, who possesses special knowledge of the area or domain the model problem is set in. Also a VOMAS should be developed side-by-side with the model from the very beginning of the modeling process.

Though the original paper outlines the properties of a possible VOMAS design, it fails to give a more detailed insight into the process of developing a VOMAS and its possible usages.

3 Objectives

Since its publication in 2009, the VOMAS method has been frequently cited in other papers about validation and verification of agent-based models. Despite this fact there hasn’t been a thoroughly documented case where a VOMAS was used in the validation process of a model. The main objective of this paper is therefore to develop a structure within the basic framework idea of a VOMAS, which enables the user to apply several validation methods for agent-based models. This can be divided into 4 sub-goals:

1. Identify validation methods, which can be applied to agent-based models.
2. Design a structure within the basic concept of a VOMAS described in [3] that offers easy usability for users not familiar with the details of the VOMAS (preferable the Subject Matter Experts).
3. Implement the developed VOMAS design within an existing or newly developed agent-based model to test its usage.
4. Provide a detailed documentation of the steps taken above to guarantee reproducibility.

4 Theory and Methods

The following is based on [7].

4.1 Validation of agent-based models

Validation of agent-based models is difficult due to a number of reasons, as listed in [1]:

1. Availability of data. This does not only cover the physical availability of data because it might be too costly to collect it or because of privacy issues, but also the fact that the data necessary for agent-based models is sometimes very difficult to measure and apply statistical methods to.
2. Overparameterisation. Introducing enough different parameters, model results can always be fitted to a given set of data points. That does not necessarily imply that the model is valid. Agent-based models, by their nature, usually must use many different parameters to describe the agents' behavior and are therefore very vulnerable for overparameterisation.
3. Emergent behavior. The very principle, on which the agent-based modeling paradigm is built upon, might also be its greatest weakness. Agent-based models are strongly non-linear and minor changes in an agent's behavior can result in a completely different model behavior. Due to a high number of non-deterministic processes in agent-based simulations those changes are very difficult to track throughout a simulation run and examine their influence.

There already exist a variety of general validation methods ranging from formal (based on mathematical principles) to informal methods. Balci in [5] and Sargeant in [6] list and describe many of them in great detail. Taking the problems of validating agent-based models into consideration, we were able to isolate general validation methods that can still be applied to agent-based models (Figure 2).

Informal	Dynamic	Symbolic
Audit	Black-Box Testing	Cause-Effect Graphing
Desk Checking	Bottom-Up Testing	
Face Validation	Predictive Validation	Constraint
Inspections	Sensitivity Analysis	
Reviews	Statistical Analysis	Assertion Checking
Turing Test	Stress Testing	
Walkthroughs	Submodel Testing	Static
	Visualization	
	Animation	
	Model Alignment	Consistency Checking
	Event Validity	Data Flow Analysis
	Traces	

Figure 2. Validation methods for agent-based models

4.2 Requirements of a VOMAS

Since the basic tasks of the VOMAS agents are already outlined in [3] we are now tasked with figuring out the detailed workflow and internal mechanisms of an exemplary VOMAS design. Although a VOMAS should be implemented from scratch with every new model, we design an abstract framework within the basic requirements given in [3] that can be applied to a variety of different models, but still only needs a few alterations when faced with an actual model. Much focus should also be put into enabling an easy change of (1) the target set the VOMAS agents need to observe, (2) what attributes they need to monitor more closely and (3) the way gathered information should be stored. Issues of performance still play a deciding role in agent-based modeling. Therefore, the developed VOMAS design should be as little time consuming as possible.

Agent-based modeling usually involves a large amount of programming. Object-oriented languages lend themselves by their nature for developing agent-based models. Therefore, it is only logical to design the VOMAS framework with an object-oriented approach in mind. This also allows us to use tools of software design and software documentation such as UML (Unified Modeling Language).

To examine whether the developed design can be used to help in validating a model, we examine if it is able to determine all or parts of the following types of validity proposed in [1]: empirical validity, structural validity, face validity and behavioral validity.

4.3 Developing the VOMAS design

Taking all information about a VOMAS we have gathered so far into consideration, we can assume a hierarchical structure with the VO-Manager at the top delegating tasks to its subordinate agents.

The Console-Agent is very self explanatory and only needs to implement a method that displays messages at run-time.

The Logger-Agent's sole task is to save information that is sent to it in the form of log entries. It might be the case that the user would like different information to be stored in different formats. We can realize that by introducing different kinds of log entries and add a method that allows the Logger-Agent to distinguish between those entries, storing each in their own way. Since the Logger-Agent also acts completely independent from the rest of the VOMAS agents, it can always run parallel to the rest of the simulation.

The VO-Agents are the only connection between the VOMAS and the simulation agents (Sim-Agents). Therefore, interfaces need to be developed to make the gathering of information about the Sim-Agent possible. For the sake of flexibility a VO-Agent should be able to have more the one Sim-Agent as a target.

Agent attributes and changes in behavior that might be of interest for the validation should be established throughout the modeling cycle in cooperation with the modeling expert and the Subject Matter Expert. A VOMAS should be able to monitor those attributes and changes. This can be done by introducing a Watch-Agent for each of those, so-called *watch-values*. With each watch-value we have to define certain *key attributes*, that – with the help of an evaluation method – determine whether something of interest has happened to a Sim-Agent and if further information should be gathered and sent to the Logger-Agent. Watch-Agents regularly check those key attributes from a list of Sim-Agents given to it by its superior VO-Agent. To increase the performance the Watch-Agents should be able to mark a target as “not-interesting” if said target is no longer of interest for the user. It might also be the case that the user only wants to observe a certain type of Sim-Agent, ignoring others completely. This can be done by adding a method that checks if a Sim-Agent has some, or all of the so-called *selective attributes* (for example age or gender). Those selective attributes can be saved in a parameter sheet and altered before each simulation run if necessary. This can be very useful if the monitoring of *all* simulation agents might be too time consuming.

The Constraint-Agent has to check the agents' behavior for violations of model assumptions and unwanted

behavior. What those might be varies from model to model and has to be established by the modeling expert and the Subject Matter Expert. The Constraint-Agent can work very much the same as the Watch-Agent except for two key aspects:

- Selective attributes should not be introduced with the Constraint-Agents. The difference being that a violation of model assumptions should if possible never be overlooked.
- The Constraint-Agent should – upon detecting a violation of model assumptions – be able to terminate the simulation run. This might prove to be useful for verification and might save a lot of time during the parameterisation of the model.

The VO-Manager should have access to global attributes of the simulation and must be able to notify its subordinate VO-Agents to start the process of gathering data and monitoring simulation agents.

5 Results

The developed VOMAS design is platform-independent and can be implemented in most object-oriented software environments. The structure (Figure 3) includes all the principal ideas of a VOMAS mentioned in [3] and implements features and concepts, which can be of further value to the user.

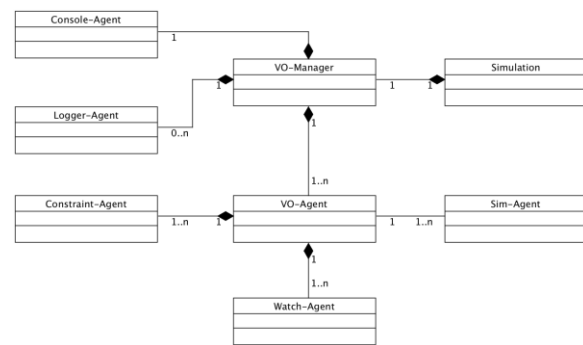


Figure 3. Class diagram of the developed VOMAS design

While the design has a core structure, which should be kept without regard to the application, other parts are designed to be interchangeable, depending on the model at hand and the preferences of the user.

The design is furthermore documented in detail in [7] using sequence diagrams, class diagrams (Figure 4),

method descriptions and internal method procedures (Figure 5).

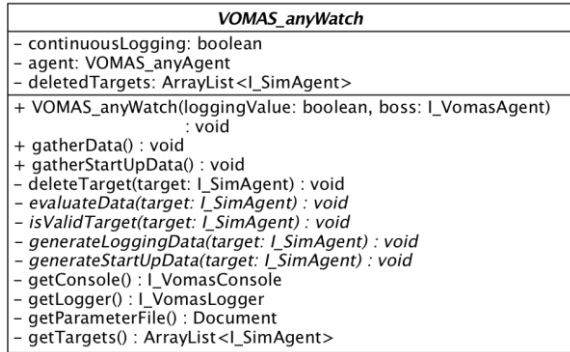


Figure 4. Exemplary class diagram of a VOMAS component



Figure 5. Exemplary documentation of internal method procedures

A VOMAS developed for an agent-based model can help apply the following validation methods:

- Turing Test, Black Box Testing, Predictive Validation, Statistical Techniques, Traces, Model Alignment: provided that there is enough real world data, those methods can be applied with the use of the Watch- and Logger-Agents.
- Animation, Visualisation, Immersive Face Validation: those methods rely heavily on how much information can be gathered about the simulation agents and their behavior and might only be applicable to certain agent-based models.

- Sensitivity Analysis, Event Validity: changes in attributes or the occurrence of certain events can be monitored by Watch-Agents.
- Assertion Checking, Extreme Condition Tests: those methods try to detect faulty or wrong model behavior. Constraint-Agents are designed for exactly this purpose.
- Bottom Up Testing: a VOMAS can be developed only for submodels or certain parts of a model.

A VOMAS can therefore be used to assess face validity, empirical validity and behavioral validity.

6 VOMAS applied to a test case model

To test the usage of the developed VOMAS design, we implement it for an existing agent-based model that tries to simulate the spread of the influenza virus within the population of Austria. The influenza model consists of people distinguishable by their gender, age and occupation. They each are assigned to a household and depending on their occupation a workplace or a school. With every iteration – representing the passing of one day – each person visits their assigned locations and has contact with other people also assigned to that location. Having contact with an infectious person can in turn lead to another infection. An infected person can develop mild or severe symptoms, which might also result in the person having to stay at home for the duration of the illness. After recovering from the disease, a person is declared immune and cannot be infected again. A more detailed description of the model is given in [8].

The spread of the influenza disease is the core subject of the model study. Therefore the infections are the key aspects when it comes to validating this model and we can define the first watch-value *infection*. The key attribute for this watch-value is the model variable *isInfected* of type Boolean. Thus the evaluation method is obviously a simple if-else statement. If a person has been infected throughout the course of a day, a watch-agent will gather more information about the infection and sent it to the Logger-Agent in the form of a log entry. Information that is of interest for us is the day the infection happened, the person that caused it and the location where the infection took place. By definition a person cannot be infected twice, which allows us to mark a person that has been infected once as “not interesting”, meaning this per-

son will no longer be checked for this particular watch-value.

To make sure that no illegal state changes occur, Constraint-Agents constantly check for such a violation. This can be very useful in early stages of the programming process, where logical errors might still occur.

Another watch-value should be the state changes of a person that has been infected. Those state changes are (1) start becoming infectious, (2) developing mild or severe symptoms, (3) becoming naturally immune and (4) stop being infectious. The key attributes are again Boolean variables and the additional information that should be gathered is the day of the state change, the type of the state change and – if the person has developed symptoms – whether the person stays at home. Both the infection watch-value and the state change watch-value use the same selective attributes, which are the age, gender and occupation of a person.

To introduce another level of flexibility, the user can choose which watch-values should be considered before each simulation run. The information about which watch-values to apply as well as the selective attributes is stored in a parameter sheet.

In addition to information that is gathered at run time, the VOMAS agents should also gather information about the simulation agents that will not change throughout a simulation run (e.g. information about each person and its assigned locations).

Because of the vast amount of information that will accumulate when running the simulation with over 8 million agents – representing the Austrian population – we have to use a database to store the information and a Logger-Agent needs to be designed accordingly.

Because of the fact that in this particular case the VOMAS is *not* developed side-by-side with the model, there have to be some minor changes in the original model. Those are mostly changes that make the necessary information accessible to the VOMAS agents. The original model has to be verified one more time after the alterations are done.

The influenza model was implemented in the object-oriented language Java and it is convenient to implement the VOMAS in Java as well. Every other object-oriented language would work just as well, the only difference being that the interfaces between the

VOMAS and the simulation might be a bit more complex.

Using the full potential of the implemented VOMAS we can now extract vital information from the data provided to us by the VOMAS (Figure 6 and 7).

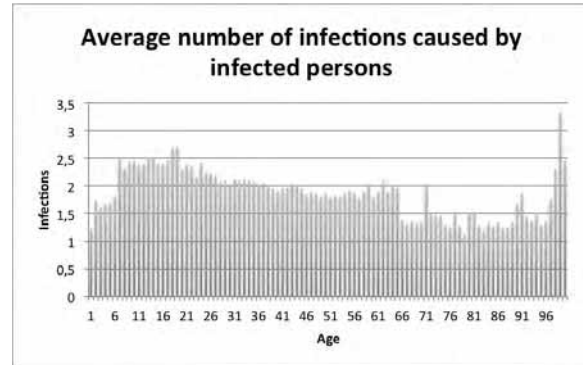


Figure 6. Average number of infections caused by infected persons that infected at least one other person

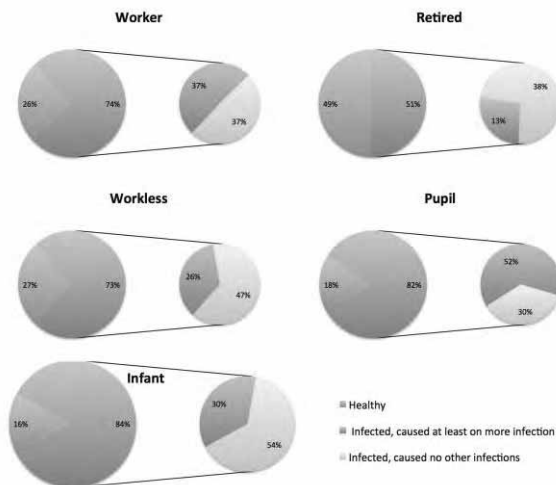


Figure 7. Percentage of persons being infected

While Figure 6 needs to be assessed by a Subject Matter Expert in terms of plausibility, Figure 7 immediately points towards a fault in the model behavior. More than 80% of the infants are infected with the influenza virus, which is simply not credible. The reason for this could be a wrong choice of parameters, or even a mistake in the conceptual model.

7 Discussion

The developed design was able to cater to all the desired actions taken by the user. In case of the influenza simulation the VOMAS was implemented afterwards. To maximize the potential and flexibility of

a VOMAS, it should be developed side-by-side with the actual model it should later help to validate. To further improve the developed VOMAS design, it needs to be applied to other agent-based models as well. Since a VOMAS is able to detect unwanted or if need be unrealistic agent behavior, it would be interesting to apply it to a model composed of intelligent agents which can form new rules of behavior during a simulation run. The idea being that the user can specify counterintuitive behavior, the VOMAS then observes the simulation agents and isolates faulty behavior, making it possible to reconfigure the internal logic by which rules of behavior are adapted.

To allow for an easier use of a VOMAS, a graphic user interface should be included in future implementations.

8 References

- [1] F. Klügl. *A validation methodology for agent-based simulations*. Proceedings of the 2008 ACM symposium on Applied computing, pages 39-43, 2008.
- [2] A. Louloudi and F. Klügl. *Immersive Face Validation: A new validation technique for agent-based simulation*. Computer Science and Information Systems (FedCSIS), pages 1255-1260. IEEE, 2012.
- [3] Muaz A. Niazi, Amir Hussain and Mario Kolberg. *Verification and validation of agent based simulations using the vomas (virtual overlay multi-agent system) approach*. Computing Science and Mathematics Conference Papers and Proceedings, 2009.
- [4] Muaz A. Niazi, Qasim Siddique, Amir Hussain and Mario Kolberg. *Verification & validation of an agent-based forest fire simulation model*. Proceedings of the 2010 Spring Simulation Multiconference, 2010.
- [5] O. Balci. *Validation, verification, and testing techniques throughout the life cycle of a simulation study*. Annals of Operations Research vol. 53, pages 121-173, 1994.
- [6] R.G. Sargent. *Verification and validation of simulation models*. Proceedings of the 2010 Winter Simulation Conference (WSC), pages 166-183, 2010.
- [7] J. Ruths. *Analyse der Struktur und Funktionalität eines Virtual Overlay Multi-agent System bei einem Influenza-Modell*. Master's Thesis, University of Technology Vienna, 2016.
- [8] P. Pichler. *Simulation und Validierung von agentenbasierten Modellen für die Ausbreitung von Epidemien*. Master's Thesis, University of Technology, Vienna, 2013.

Calibration Strategies for Agent-based Simulation Models with Variability

Claire Rippinger¹, Martin Bicher¹, Florian Miksch²

¹Institute for Analysis and Scientific Computing, TU Wien, Vienna Austria

²DWH Simulation Services, Vienna Austria

0826387@student.tuwien.ac.at

During the development of an agent-based simulation model, the model often has to be calibrated, which means adjusting the parameters such that a reference system can be reproduced. A major problem in calibrating an agent-based simulation model is the variability of the results, due to random choices made by the agents. To reduce the variability, the numbers of agents has to be increased, which in return increases the computation time of the simulation. An attempted solution to this problem consists of increasing the numbers of agents gradually. This approach is tested with two different calibration algorithm: simulated annealing and evolutionary algorithm. Different updating schedules are applied on a test model and examined in terms of their running time and their performance. It is shown that a evolutionary algorithm with an increasing agent count manages to produce similar results as a standard calibration using only half the computation time. To conclude, the best performing calibration process is used to calibrate an existing agent-based model simulating a well known past influenza epidemic.

1 Introduction and Objectives

Agent-based simulation is a relatively new modelling technique [1]. It has experienced increasing application in several fields since it offers many benefits over other modelling methods [2]. According to Bonabeau, the main advantages of agent-based models are their flexibility, their natural way to describe a system, and their ability to produce an emergent behavior [3]. Contrary to other modelling techniques, it does not try to dictate the general behaviour of the system. Instead, it consists of several independent entities, called agents, which are given certain properties, behaviour and rules to change this behaviour. These agents interact with each other and their environment during a simulation run and produce the overall outcome of the system. A typical application for this sort of modelling is the simulation of epidemics.

An important step in developing a model, agent-based or other, is the calibration. It consists of adjusting the different parameters used in the simulation such that the simulated results match a given outcome. When

the model is capable of reproducing a reference system, it can be used to test the outcomes of alternatives strategies in this reference system or to make predictions by simulating the reference system in the future. The nature of agent-based simulation models induces different problems regarding the calibration process. Since the result of the simulation emerges from the interaction between the agents, the outcome is hard to estimate. Therefore, it is difficult to say in what way the different parameters affect the simulation outcome. Only by running the simulation, the effects may be observed and appropriate parameter changes can be made. If the model requires only a few parameters, these adjustments can be made manually. With an increasing number of parameters, calibration algorithms are needed. Since a calibration problem consists of minimizing the distance between the simulated data and the reference system, it can be seen as an optimization problem and algorithms from this application area can be used. The nature of agent-based simulation models requires calibration algorithms which regard the simulation as a black box and only have informations on the outcome of the simulation and not

on the internal processes and calculations. The evolutionary algorithm and simulated annealing are two algorithms meeting this criterion. They are presented in section 2.

A common problem in performing a calibration is the long computation time required to produce the results. Section 2 proposes varying the amount of agents used in the simulation as a solution to this problem. Different configurations of this method are applied to a test model. The configuration with the best performance is then used to calibrate a more complex agent-based simulation model. Both models are described in section 3.

2 Methods

In a calibration process, the agent-based simulation model acts as a function: given a specific parameter set, it produces the simulated data points. These are passed to an error function which calculates the distance between the simulated data points and the data points that should be matched by the model. Often, a weighted Euclidean distance is used. This allows the error function to put more emphasis on the characteristic elements of the data of the reference system. The aim of a successful calibration is to find a parameter set which minimizes this distance. Thus a calibration can be considered as an optimization problem and the respective algorithms can be used.

In this paper, two different optimization algorithm are applied for model calibration: simulated annealing [4] [5] and evolutionary algorithm [6][7]. Flowcharts of these algorithms can be seen in figure 1 and 2.

In simulated annealing, accepting a point with a larger error should allow the algorithm to escape local minima and converge to a global minimum. The acceptance probability depends on the temperature and is decreased during the calibration process. The cooling schedule applied, as well as an appropriate choice of the neighbourhood of a point, have a large influence on the convergence of the algorithm. The evolutionary algorithm uses several points simultaneously to determine the global minimum. By choosing different methods of selecting the points and combining them to form new candidates, evolutionary algorithms can be adapted to suit the needs of many calibration

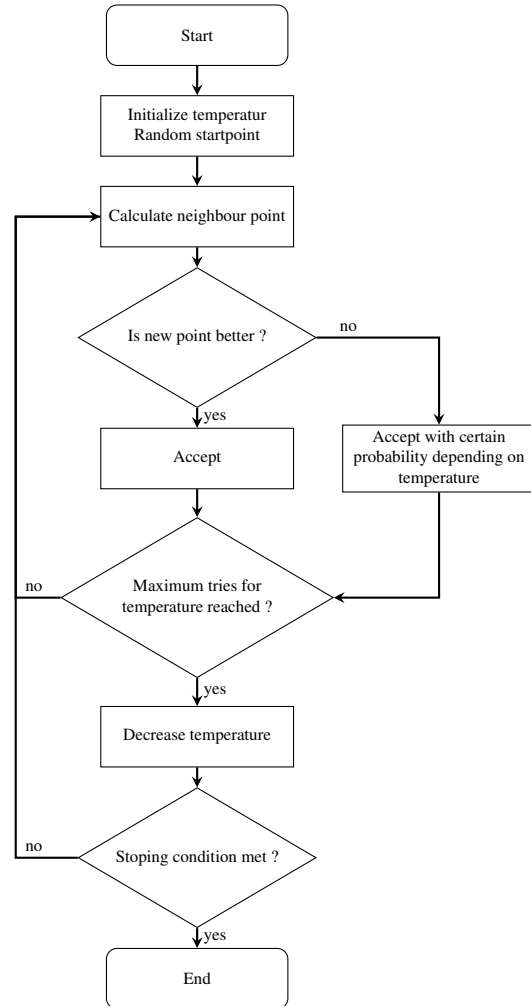


Figure 1: Flowchart of simulated annealing based on Kong et al. [8]

problems. However, finding the optimal configuration is often difficult.

Agent-based models often have long computation times caused by high agents numbers. This is an important issue during calibration, when the model is simulated hundreds of times. The runtime can be reduced with a lower number of agents. However, the agents behaviour usually depends on random decisions. Hence, the simulation results underlie a variability. For high agent numbers, the variability is naturally low due to the law of large numbers. Lower agent numbers lead to an unwanted higher variability of the results.

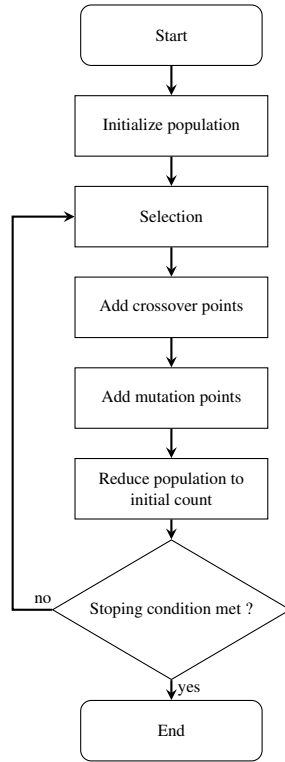


Figure 2: Flowchart of the evolutionary algorithm based on Kong et al. [8]

A possible solution to this problem consists of varying the number of agents throughout the calibration procedure. Agent-based models can be scaled by simulating them with reduced agent numbers. This does not affect its functionality but increases uncertainty of the results due to a required upscale to the actual problem. At the start, the simulation model is run with a small number of agents, allowing to test many parameter sets in a short time period. During the calibration the number of agents is gradually increased until the targeted agent count is reached.

A calibration performing an increase of the agent count requires the following information:

- the starting amount of agents
- the targeted amount of agents
- the number of agent count updates
- the growth behaviour
- the number of simulation runs

After every update, the error of a newly considered

parameter set is calculated with the new agent count. Normally, this error is only calculated once and then referred to in the further calibration process. However, this might cause a problem. If the error is calculated with a small agent count, it is possible that a low value has only been reached by accident and does not represent the usual outcome of the simulation performed with this parameter set. It is important to eliminate these false error values in time and not base every further search on the corresponding parameter sets. A simple solution would consist of recalculating the error of all the current parameter sets at every update of the agent count. This increases the number of simulation runs during a calibration, especially when an evolutionary algorithm is used. As a trade-off, the parameter sets are not re-evaluated directly at the next agent count update but after two agent count updates. This way, the chances are higher that the parameter set gets discarded by the algorithm before it needs to be recalculated. However, during the last update which increases the agent count to the targeted amount, every parameter set is re-evaluated to ensure that the best error has been calculated with the full agent count.

The increase of the agent count during the calibration has also an effect on the stopping conditions of the calibration algorithm. Normally, a calibration would terminate, if the error has reached a certain value. But, as mentioned above, if this error has been calculated with a small agent count, it might not be valid. Therefore the calibration is not allowed to terminate prematurely but has to perform the full amount of simulation runs.

3 Models

First, the calibration methods are tested on a simple SIR model which simulates the spreading of an infectious disease. In such a model, the agents represent people who can be in one of three different states: susceptible, infected or resistant. Every time, a susceptible person comes in contact with a person already infected the disease may be transmitted. After a certain amount of time, an infected person recovers from the disease and becomes resistant. This means, the person can not be infected a second time. In our model, there are two parameters that need to be calibrated: the probability p_i that a healthy person is in-

ected when they comes in contact with an infected person and the probability p_r that an infected person is recovering. To create a reference system, the model is run with a known parameter set $p_i = 0.4$, $p_r = 0.05$. The output consists of the number of infected agents at a given time step. The goal of the calibration is to reproduce this curve. Previous tests have shown, that the variability of the results is negligibly small, if 10 000 agents are used, making this the target population.

Ultimately, the calibration algorithm is used to determine the parameters of a more complex agent-based model simulating an influenza epidemic. A known influenza epidemic in the year 2007 in the Austrian population is used as a reference system. The data that needs to be reproduced consists of eleven data points representing the number of newly infected people per week during the eleven week long influenza wave. For this model, the calibration needs to determine five parameters describing the contact rate between the agents, the probability of infection and development of mild or severe symptoms and the ratio of the population which is naturally immune.

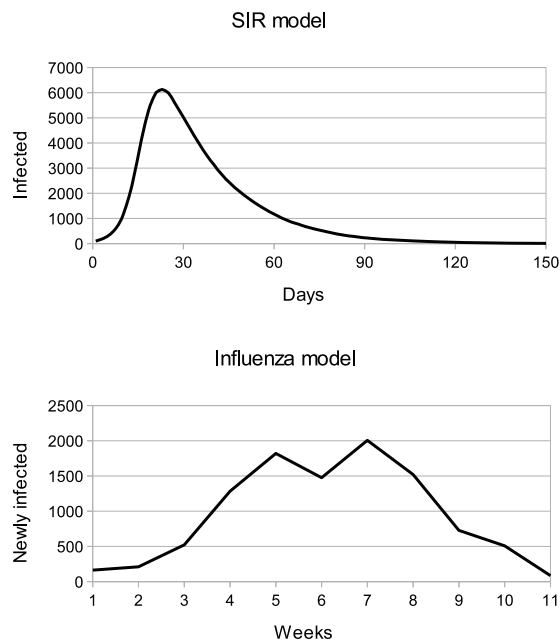


Figure 3: Reference systems of the SIR and the influenza model that need to be reproduced during the calibration

4 Results

Both of the models described above, as well as the calibration algorithms have been implemented in Java and all the following calibrations have been calculated on a laptop running with Windows 8.1 using an Intel^(R) Core^(TM) i5-4200U processor and 8,00 GB RAM.

4.1 Results of calibration the SIR model

In order to compare the results of a calibration using an increasing amount of agents, the calibration has been performed with a constant agent count. The error is calculated using an Euclidean distance which puts a larger weight on the peak of the epidemic. In order to scale this error, the value of every data point is divided by the current agent count.

In this paper, 15 different update schedules are tested as shown in Table 1.

	Agents at Start	Growth	Updates
1	1000	no	0
2	3000	no	0
3	10 000	no	0
4	1000	geom.	2
5	1000	geom.	4
6	1000	geom.	8
7	3000	geom.	2
8	3000	geom.	4
9	3000	geom.	8
10	1000	linear	2
11	1000	linear	4
12	1000	linear	8
13	3000	linear	2
14	3000	linear	4
15	3000	linear	8

Table 1: Updating schedules

The calibration is terminated after 1000 model runs. For each updating schedule the calibration is performed with three different calibration algorithm configurations providing good results in previous tests using a constant agent count. For each of these configurations, the calibration is run 10 times. For the

simulated annealing, these configurations use a geometrical cooling schedule with the temperature being lowered every 10 loops by a factor 0.8, 0.85 resp. 0.9. In the evolutionary algorithm the population consists of 40 agents. A ranking based selection is applied [9]. During the crossover, 10 parameter sets are formed by calculating the mean of the two parent parameter sets and during mutation 8, 12 resp. 16 parameter sets are created by replacing one parameter with a random value.

Figure 4 shows the results of these calibrations. The bars represent the mean error of the 30 calibration results performed with the update schedule and the line represents the mean time to perform these calibrations. Note that the update schedule 3 represents a calibration performed with the targeted amount of agents throughout the whole process, making it the method applied during a standard calibration. Update schedules 1 and 2 represent calibrations using a constantly low agent count.

For the simulated annealing it can be seen, that only certain update schedules yield comparable results as a calibration performed with a constantly high amount of agents. Generally, it can be observed, that a higher number of agents at the start is preferable, as well as a smaller number of updates. The evolutionary algorithm produces overall better results than simulated annealing. The error obtained by updating the number of agents is even generally smaller than the one calculated by a standard calibration. There is no significant difference between the results of the different update schedules and no trend can be observed.

As expected, the computation time is much smaller for calibration updating the number of agents. In general, the running time is about half of the time required by a standard calibration represented by the update schedule 3. Calculations using geometrical growth or a small number of starting agents require less time than those using linear growth and a higher amount of starting agents. Furthermore, the computation time decreases slightly with the number of updates performed. However it is expected that this decrease in running time is not an ongoing trend. At some point, the benefits of calculating with a lower agent count will be outweighed by the costs of re-evaluating the current population of parameter sets at every update. To verify this presentiment a new series of tests have been performed. Calibrations using ge-

ometrical growth and a starting agent count of 1000 are calculated using different numbers of updates, extending the calibrations 4, 5 and 6 from figure 4. The mean error and running time of these calibrations are shown in figure 5. It can be seen, that the running time does increase with a larger number of updates without producing significantly better results.

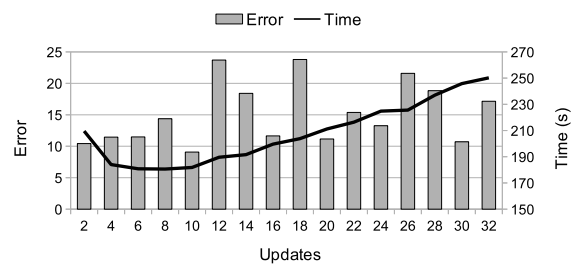


Figure 5: Error and running time for a calibration starting with 1000 agents and applying geometrical growth depending on the number of updates performed

It has been mentioned above that calibrations performed with simulated annealing provide worse results when the number of updates increases. Concerning this observation, further analysis of the error produced by these calibrations have revealed the following phenomenon: there are two different ways in which the error evolves. During some calibrations a relatively small error is already achieved using only a low agent count. The rest of the calibration process is then used for the fine tuning of the parameter set. However, if the error produced with a small amount of agents is not small, the current parameter set is not replaced with a better solution for the most part of the calibration. It is not until the simulation runs with the targeted amount of agents, that the calibration algorithm is effective and starts to lower the produced error. If the calibration process is performed using a high number of updates, the amount of simulation runs performed with the targeted agent count is too low to reach an acceptable error in time. Figure 6 shows the two different evolutions of the error during a calibration performed by simulated annealing using 4, resp. 8 updates. Each line represents one of the two typical behaviours of the error.

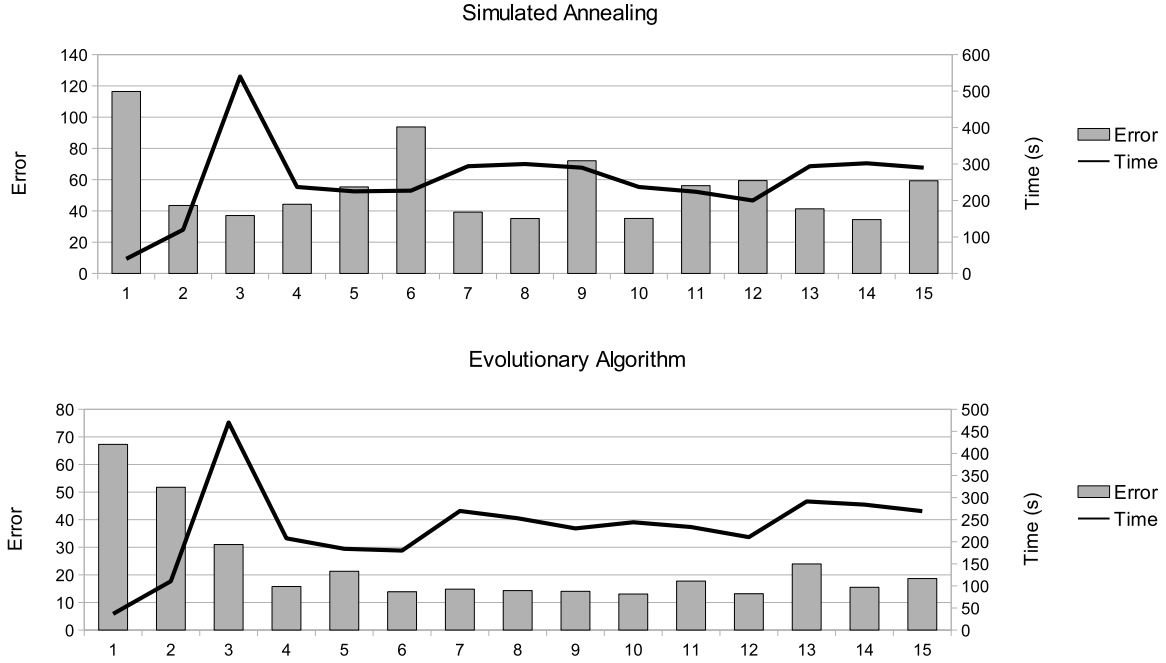


Figure 4: Error and running time of the calibration performed with simulated annealing and an evolutionary algorithm applying the different updating schedules listed in Table 1

4.2 Results of calibration the influenza model

The findings of these tests are now used to calibrate the more complex influenza model. The error function used is similar to the one described at the beginning of this section. Since the data point at week six is most probably inaccurate, the weight posed on this point is very small. One simulation of the influenza model with 800 000 agents takes 300s on average. To improve the running time of the calibration, parallel computing on three kernels is applied. Since 1000 calibrations are performed, the total running time of a calibration without agent count updates would amount to approximately 28 hours. To further reduce this computation time, a calibration using 4 agent count updates is applied. The starting agent count consists of 50 000 agents which is increased geometrically to reach the targeted count of 800 000 agents. Due to the better results with the SIR model, the evolutionary algorithm is used for the calibration. The result can be seen in figure 7. The running time of this calibra-

tion consisted of about 560 minutes, only a third of the estimated time required by a standard calibration.

5 Discussion

This paper briefly describes the approach of an agent-based simulation model and the procedure of calibrating such a model. The variability of the results of an agent-based model complicate the task of calibration and the usual methods of reducing this variability lead to an increase in the running time of the calibration procedure. By gradually increasing the number of agents used in a simulation, this paper proposes a possible solution to this problem. This strategy is then tested on a simple agent-based simulation model. The performance looks very promising, but leaves a few open questions that require further research. For example, it might be possible to improve the performance of the simulated annealing with a cooling schedule adapted to the increase in the number of agents. Furthermore, the optimal number of

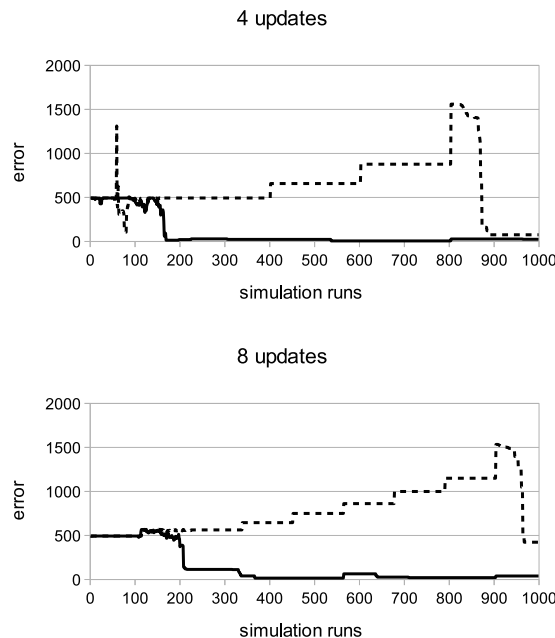


Figure 6: Two possible evolutions of the error calculated with simulated annealing during a calibration using geometrical growth, 1000 agents at the start and 4, resp. 8 agent count updates

updates needs to be determined. This number may depend on the simulation model used. In this paper, the strategy has only been applied to one type of agent-based simulation model. It needs to be tested if the results are similar with another type of model and what factors are beneficial to a good performance of this calibration method.

References

- [1] C. Macal and M. North. *Tutorial on Agent-Based Modeling and Simulation PART 2: How to Model with Agents*. Proceedings of the 2006 Winter Simulation Conference, p. 73-83, 2006.
- [2] P. Siebers, C. Macal, J. Garnett, D. Buxton and M. Pidd. *Discrete-Event Simulation is Dead, Long Live Agent-Based Simulation!*. Journal of Simulation, **4**, no. 3, p. 204-210, 2010.

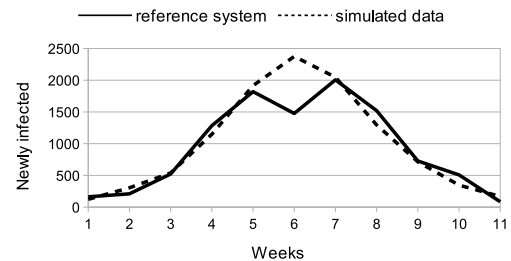


Figure 7: Results of a calibration of the influenza model using an evolutionary algorithm, 4 agent count updates and geometrical growth

- [3] E. Bonabeau. *Agent-based modeling: Methods and techniques for simulating human systems*. Proceedings of the National Academy of Sciences, **99** no. Supplement 3, p. 7280-7287, 2002.
- [4] S. Kirkpatrick and C. D. Gelatt and M. P. Vecchi. *Optimization by simulated annealing*. SCIENCE **220** no.4598, p. 671-680, 1983.
- [5] V. Cerny. *Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm*. Journal of Optimisation Theory and Application **45** no.1, p. 41-51, 1985.
- [6] J.H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, USA, 1975.
- [7] I. Rechenberg. *Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Verlag, Deutschland, 1973.
- [8] C.Y. Kong, P.M. McMahon and G.S. Gazelle. *Calibration of Disease Simulation Model Using an Engineering Approach*. Value in Health, **12** no.4, p. 521-529, 2009.
- [9] D. Goldberg and K. Deb. *A comparative analysis of selection schemes used in genetic algorithms*. Foundations of genetic algorithms **1**, p. 69-93, 1991.

Modellierungsansätze und Fallbeispiele in der Ausbildung für Modellbildung und Simulation

Andreas Körner¹, Stefanie Winkler¹, Felix Breiteneker¹

¹Technische Universität Wien

Institut für Analysis und Scientific Computing

Wiedner Hauptstraße 8-10

1040 Wien

andreas.koerner@tuwien.ac.at

Modellbildung und Simulation ist als wissenschaftliches Gebiet sehr schwierig zu charakterisieren. Es gibt ein weites Feld an mathematischen Ansätzen und Methoden aber mindestens zu gleichen Teilen einen hohen interdisziplinären Aspekt. Aufgrund dieser Problematik gestaltet sich die Ausbildung in diesem Gebiet umso schwieriger. Modellbildung und Simulation ist an vielen Fakultäten und Instituten unterschiedlicher Grundausbildung, wie Mathematik, Informatik, Elektrotechnik, Mechanik, Physik, etc. als Unterrichtsfach zu finden. Problematik dabei ist es, dass eine grundlegende Ausbildung für Modellbildung und Simulation oft eine Querschnittseinführung in verschiedenen Fächern mit sich bringt. Angefangen bei mathematischen Methoden wie Differentialgleichungen für Zustandsraummodelle dynamischer Systeme oder die Laplace-Transformation für Verhaltensmodelle, über elektrotechnische Problemstellungen, bis hin zu mechanischen Modellen für einfache als auch komplexe Systeme. Spätestens bei der Simulation sind viele informatische Aspekte zu berücksichtigen.

Ausbildung in Modellbildung und Simulation muss also eine Brücke zwischen Mathematik und den Anwendungen derart bauen, dass grundsätzliche Ansätze in unterschiedlichen Fallbeispielen erkannt werden und auf ähnliche Situationen angewandt werden können. Das vorliegende Paper stellt ein mögliches didaktisches Konzept vor und zeigt einige Fallbeispiele aus der Umsetzung.

1 Struktur und Aufbau des Ausbildungskonzeptes

Die grundlegende Eigenschaft im vorgestellten Konzept ist die theoretischen Aspekte der Modellbildung und Simulation anhand von fundierten Grundlagen zu vermitteln aber gleichsam aussagekräftige und illustrierende Fallbeispiele zu transportieren. Diese beiden Aspekte müssen aufeinander abgestimmt und passend miteinander verschränkt werden. In diesem Zusammenhang spielt natürlich die Vorbildung der Hörerinnen und Hörer oft eine wichtige Rolle, insbesondere bei der Auswahl der Fallbeispiele.

Das Ausbildungskonzept der Modellbildung und Simulation bezieht sich auf das Fach Systemsimulation, welches also im Wesentlichen auf dynamische Prozesse und Systeme abzielt.

1.1 Struktur der akademischen Themen

Die akademischen Themen bezeichnen in diesem Zusammenhang die Themengebiete der Modellbil-

dung und Simulation, welche abstrakt vermittelt werden sollen. Dabei handelt es sich in der Ausbildung zur Systemsimulation um folgende Themen:

- Modellbildungs- und Simulationskreislauf
- Eingangs- Ausgangsmodelle und Verhaltensmodelle
- Differentialgleichungs- und Zustandsraummodelle
- Physikalische Modellierung
- Differenzgleichungen und zeitdiskreter Zustandsraum
- Zelluläre Automaten
- Agentenbasierte Modellierung
- Diskrete Event Simulation
- Modellierung mit Ereignisgraphen
- State Event Modelling

1.2 Struktur der Fallbeispiele

Die zweite Säule des didaktischen Aufbaus sind aussagekräftige und exemplarische Beispiele. Diese sind elementar im grundlegenden Verständnis von Modellbildungsansätzen. Ausgehend von dem Problem, dass unterschiedliche physikalische Domäne in der Modellbildung und Simulation oft Fachkenntnisse benötigen, müssen Beispiele gewählt werden, deren Umfeld kurz aber doch umfassend einem akademischen Kreis näher gebracht werden können. Im angesprochenen Kurs der Systemsimulation sind das die folgenden Fallbeispiele:

- Populationsdynamik – Lotka Volterra Räuber Beute Modell
- Lineares und nichtlineares mathematisches Pendel
- Kardiovaskular Kompartiment Modell
- Fahrrad Ergometer Verhaltensmodell
- Epidemiologie Modelle
- Modell kommunizierender Gefäße
- Resonanzschwingkreis

Die genannten Fallbeispiele sind keineswegs immer an ein konkretes Modellierungskonzept gebunden. So ist es z.B. im Fall der Epidemiologie Modelle. Dieses Fallsbeispiel wurde als Differenzialgleichungsmodell, zellulärer Automat und mit Hilfe eines Agentenbasierten Ansatzes implementiert. Diese unterschiedlichen Ansätze können anschließend weiter verglichen und gegenübergestellt werden.

Einige der genannten Fallbeispiele sind an konkrete Modellierungsansätze gebunden, wie zum Beispiele der Resonanzschwingkreis oder das Fahrrad Ergometer Verhaltensmodell. Diese Beispiele werden aufgrund ihres einführenden Charakters eher zu Beginn eines thematischen Abschnittes verwendet.

2 Abbildung der Kursstruktur

An die Vorstellung des Konzeptes anschließend wird in diesem Kapitel nun auf die Kursstruktur eines derartigen Kurses eingegangen. Die Kursstruktur sollte die Kombination aus Theorie und Fallbeispielen hinreichend unterstützen. Typischerweise wird mit dem Simulationskreislauf begonnen, um die grundsätzliche Aufgabe der Modellbildung und Simulation vorzustellen. Daran anschließend wird die Reihenfolge der akademischen Themen abgestimmt abgearbei-

tet. Je weiter der Kurs fortschreitet, desto mehr werden die Aufgaben in die akademischen Themen eingebunden und als Motivation oder konkrete Untersuchung angeboten. Die Kursstruktur ist damit zu Beginn eine Vorlesung mit einer Übung oder einem Praktikum gepaart, wird zum Ende hin aber mehr und mehr zu einer integrierten Vorlesungsübung. Eine Veranschaulichung dieser strukturellen Idee wird in Abbildung 1 gegeben.

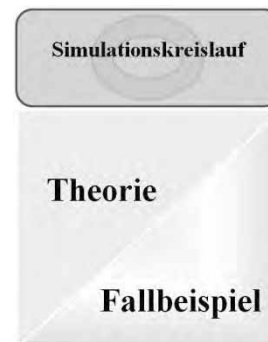


Abbildung 1. Veranschaulichung der Kursentwicklung hinsichtlich der Verteilung Theorie-Fallbeispiele

Die in Abbildung 1 angedeutete Gerade als Trennung der beiden Bereiche ist variabel. Abhängig von den besprochenen Themen und dem nötigen theoretischen Hintergrund kann dieser Verlauf angepasst werden.

3 Testung und Evaluierung

Um einen Lernfortschritt für die Kursteilnehmerinnen und Teilnehmer zu gewährleisten muss natürlich auch eine ständige Evaluierung des Lernfortschritts gewährleistet werden. Diese ist durch die Fallbeispiele im laufenden Kurs sehr einfach abzubilden. Selfevaluation so wie institutionalisierte Überprüfungen können einfach mit Blended Learning Methoden implementiert werden und gewinnbringend in die Kursstruktur integriert werden. Mehr zu diesem Thema ist in [1] veröffentlicht.

4 Referenzen

- [1] M. Bicher, I. Hafner, A. Bauer, C. Pöll, N. Popper und F. Breiteneker: *A Web-based Platform for E-Learning and Blended Learning in Modelling and Simulation*. Tagungsband International Conference on Business, Technology and Innovation, in Durres, Albania, S. 100 – 109, 2013.

Vergleich von Blended Learning Tools für Modellbildung und Simulation

Stefanie Winkler¹, Martin Bicher², Andreas Körner¹

¹Technische Universität Wien, Institut für Analysis und Scientific Computing
Wiedner Hauptstraße 8-10, 1040 Wien

²dwh GmbH, Neustiftgasse 57-59, 1070 Wien
stefanie.winkler@tuwien.ac.at

Dieser Artikel befasst sich mit dem Vergleich verschiedener online Systeme für Blended Learning. Hierbei geht es im Speziellen um Lern- und Lehrunterstützungen für Lehrveranstaltungen in Richtung Modellbildung und Simulation. An der Technischen Universität Wien in der Forschungsgruppe Mathematische Modellbildung und Simulation ist die online unterstützte Lehre schon seit vielen Jahren ein großes Thema. Begonnen hat es mit einem selbst-entwickelten Server der für verschiedene Verwendungen eingesetzt wird. In den erwähnten Lehrveranstaltungen geht es um verschiedene Aspekte die unterstützt werden sollen. Zum einen wird das Tool eingesetzt, um in den Vorlesungen wichtige Prinzipien in Modellbildung und Simulation zu vermitteln. Ein weiterer Aspekt ist aber auch das individuelle Arbeiten mit Simulations Software und die Überprüfung der Kenntnisse am Ende der Lehrveranstaltungen. In diesem Beitrag werden die verschiedenen Blickwinkel und deren Ausführung beleuchtet und diskutiert.

1 Einleitung

Die Forschungsgruppe Mathematische Modellbildung und Simulation an der Technischen Universität Wien beschäftigt sich schon sehr lange mit einer online unterstützten Lehre. Seit ca. 10 Jahren gibt es an der Universität eine an angepasste Version einer Moodle Plattform, namens TUWEL. Bereits im Semester nach der Einführung des Systems wurden Kurse der Forschungsgruppe auf dieser Plattform administriert. Diese Kurse beinhalteten eine Beschreibung der Kursinhalte sowie eine thematische Gliederung der Vorlesungsblöcke. In jedem dieser Blöcke wurde die Vorlesungszeit und der Ort bekannt gegeben. Zusätzlich wurden von den Vortragenden die verwendeten Folien hochgeladen.

Um zielführende und pädagogisch sinnvolle E-Learning Unterstützung anzubieten reicht es allerdings nicht aus, nur Materialien online verfügbar zu machen. Zur damaligen Zeit gab es von Mathworks noch das Angebot eines MATLAB-Servers. Dieser war online zugänglich und auch Studenten ohne MATLAB Lizenz konnten dort MATLAB verwenden.

Die von der Forschungsgruppe entwickelten Beispiele wurden dann auf den MATLAB-Server geladen und konnten sowohl vom Vortragenden während der Vorlesung als auch von den Studierenden von zuhause aus zum Üben verwendet werden. Der Zugang zu den Beispielen war für jede Lehrveranstaltung durch einen eigenen Account geregelt, der den Studierenden im Semester für die Dauer der Lehrveranstaltung zur Verfügung gestellt wurde.

Dieser Server wurde leider nach wenigen Jahren von Mathworks eingestellt. Daher hat sich die Forschungsgruppe entschieden, selbst eine eigene Plattform namens MMT (Mathematics, Modelling and Tools) auf die Beine zu stellen um die verlorene Funktionalität wieder herzustellen. [1]

2 Einsatz von MMT

Die rudimentären Anforderung war eine Website zu entwickeln, bei der es möglich war, MATLAB Beispiele aufrufen zu lassen und deren Ergebnisplots

wieder zusammen zugeben. Eine wichtige Funktion hinsichtlich Modellbildung und Simulation war auch die Variation der Parameter.

In den vergangenen Jahren wurden immer wieder Verbesserungen am System durchgeführt. War es Anfangs nur möglich MATLAB Beispiel auf der Plattform vorzuführen wurde das Spektrum mittlerweile stark vergrößert. Es sind nun Beispiele in MATLAB, Simulink, Octave, Anylogic als auch Java verfügbar. Die Erweiterung der Simulink Beispiele brachte eine Schwierigkeit mit sich. Im Gegensatz zu standard MATLAB oder Octave Beispielen müssen bei der Ausführung dieser Beispiele mindestens zwei Files im Hintergrund aufgerufen werden. Im Falle von Anylogic Modellen wurden mithilfe der Anylogic-eigenen Schnittstelle zu Java als sogenannte Java-Applets in den Server integriert.

Betrachtet man nun den Beispielaufbau genauer so besteht jede Aufgabe aus 5 Abschnitten, siehe Abbildung 1 und 2.

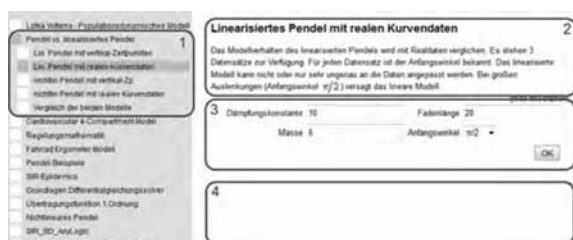


Abbildung 1: Schematische Struktur des MMT Aufbaus der Beispiele

Zum einen sind die Beispiel thematische zugeordnet. Zum Beispiel gibt es eine große Anzahl an unterschiedlichen Beispielen, die sich mit der Analyse des physikalischen Pendels beschäftigen. Diese sind alle in einem Ordner "Pendel", wie in Abbildung 1 Abschnitt 1 zu sehen, zusammengefasst. Weiters kann daher eine allgemeine Beschreibung des grundlegenden mathematischen Modells bereits für alle Modelle gemeinsam gegeben werden. Darunter folgt dann eine genaue Erläuterung der einzelnen Realisierungen im Angabetext, siehe Abschnitt 2. Um die Parameterabhängigkeit dieser Modelle zu testen, besteht im Abschnitt 3 die Möglichkeit, einige der im Modell verwendeten Parameter zu variieren. Mit einem Klick auf OK wird die Simulation ausgeführt. Anschließend erscheinen in Abschnitt 4 die Ausgabeparameter bzw.

der Ergebnisplot wie in Abbildung 2 ersichtlich.

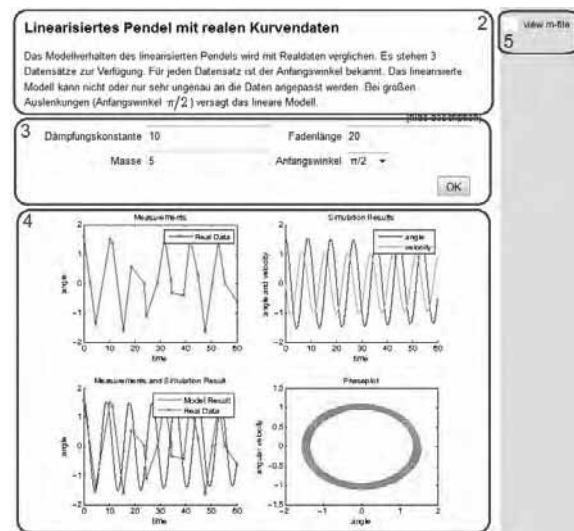


Abbildung 2: Schematische Struktur der MMT Beispiele

Abschnitt 5 stellt das File der Implementierung zur Verfügung um Studierenden einerseits zu zeigen, wie so ein Simulationsbeispiel aufgebaut ist. Andererseits haben Studierende damit auch die Möglichkeit den Programmcode zu adaptieren und Experimente am eigenen Rechner durchzuführen.[2]

Insgesamt umfasst das System an die 700 Beispiele zu verschiedensten Modellen mit unterschiedlichen Umsetzungsmethoden für Modellbildung im kontinuierlichen wie auch im diskreten Bereich. Seit einigen Jahren inkludiert der Leistungsnachweis dieser Lehrveranstaltungen auch einen Test. In diesem Test werden theoretische Fragen zu verschiedenen mathematischen Modellbildungsgrundlagen als auch praktische Fragen zu bestimmter Parameterwahl einzelner Modelle geprüft. Dies wurde bisher mit in Moodle abgebildet.

3 Einsatz von Maple

Seit 2008 betreut die Forschungsgruppe Mathematische Modellbildung und Simulation die Umsetzung einer weiteren online Unterstützung. Diese Plattform ist eine Kombination aus dem Computeralgebrasystem Maple und einem Interface welches vor allem für Testing und Assessment eingesetzt wird.

Das System unterstützt den Lehrenden dabei, Fragen in lehrveranstaltungsbezogenen Klassen zusammenzufassen und diese dann in kleineren Assignments, nach Themenblöcken sortiert, den Studenten zur Verfügung zu stellen.[3]

3.1 Maple T.A.

In den Anfängen wurde diese Plattform ausschließlich für die Lehre von Grundkenntnissen in Mathematik verwendet. Angefangen bei den Auffrischkursen in Mathematik in 2008 wurde der Beispielpool stetig erweitert. Dieser Kurs wiederholt in recht kurzer Zeit für eine große Anzahl an Studierenden den Stoff der in den Schulen durchgenommen wurde bzw. als Maturastoff von den Universitäten als Voraussetzung gesehen wird.

Seit 2009 wird Maple T.A. auch in den grundlegenden Mathematiklehrveranstaltungen für Ingenieure eingesetzt. Dabei umfasst die Lehrveranstaltung die Themengebiete Analysis, Linear Algebra und mehrdimensionale Analysis. Maple T.A. wird vor allem als Übungstool der Fertigkeiten, die nicht nur in Mathematik selbst sondern auch in anderen Lehrveranstaltungen benötigt werden, herangezogen. Studierende haben so die Möglichkeit, selbstständig aber mit ständigem Feedback den Stoff tiefgehend zu verarbeiten.

Seit 2012 wird Maple T.A. nicht nur für die Übungen und kleine Zwischentest eingesetzt, sondern auch für die Abwicklung der Abschlussprüfungen in den Lehrveranstaltungen eingesetzt. Aufgrund der sehr schnellen und fairen Bewertung durch das System, entscheiden sich bereits mehr als 50% für den computerunterstützten Prüfungsantritt.

3.2 Mathapps

Maplesoft arbeitet ständig an Erweiterungen und Verbesserungen dieser E-Learning Plattform. Seit einigen Jahren ist es auch möglich, Mathapps in das System zu inkludieren. Ein Mathapp ist eine spezielle Form eines Maple Sheets. Es kann nicht nur Maple Befehle ausführen sondern verfügt zusätzlich noch über interaktive Elemente. Zum Beispiel ist es möglich, Regler in das File einzubauen. Bei bewegen

des Reglers verändert sich der Wert des angezeigten Parameters und damit auch der Graph der Abbildung. Diese Erweiterung macht es möglich, die Anforderungen des MMT Servers in Maple T.A. nachzubilden.

4 Beispielvergleich

Die Beispielstruktur kann nun in MMT und Maple T.A. analog abgebildet werden. Abbildung 3 zeigt das Model im MMT und Abbildung 4 die Realisierung in Maple T.A.

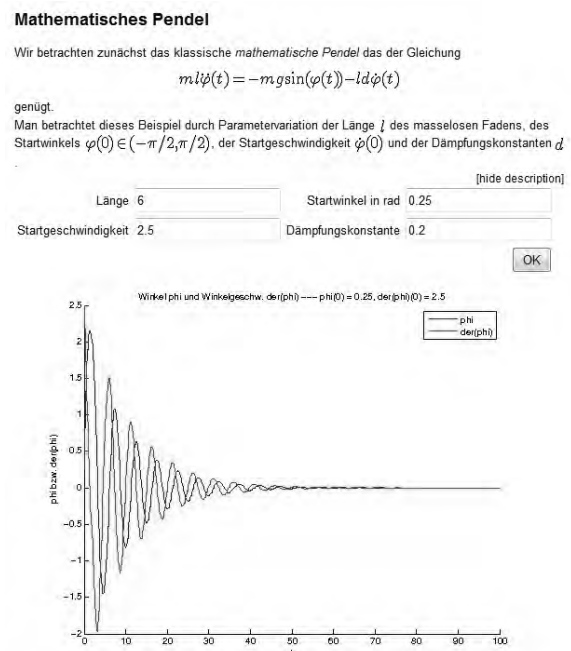


Abbildung 3: Das mathematische Pendel als Beispiel in MMT.

Natürlich sind Differentialgleichungen, wie hier das Beispiel des Pendels, leichter in einer Numerikumgebung zu implementieren als in einem Computeralgebrasystem. Trotz allem lässt sich das qualitative Verhalten, abhängig von den Parametern, in beiden Beispielen sehr gut darstellen. In beiden Systemen ist es möglich, Defaultwerte der Parameter zu bestimmen. In MMT hat dies aber keine direkte Wirkung, denn erst nach Bestätigung der Parameter erscheint die Graphik. In Maple T.A. wird das abgespeicherte Beispiel mit den zuletzt verwendeten Parametern

und dem zugehörigen Output geladen. Veränderung der Parameter müssen danach, gleich wie im MMT, bestätigt werden bevor das Beispiel neu geladen wird. Das Layout des Outputs in Maple T.A. weist allerdings noch etwas Potenzial auf.

Wie schon erwähnt ist es natürlich einfacher, Differentialgleichung in MATLAB zu implementieren. Der Vorteil von Maple T.A ist allerdings eine einfach Übersicht über die Bereiche der Parameter. Bei Eingabe eines unzulässigen Parameters am MMT Server wird beim Neuladen eine Fehlermeldung produziert. Dies kann in Maple T.A. gar nicht erst passieren da der Bereich durch die Regler von Anfang an eingeschränkt ist.

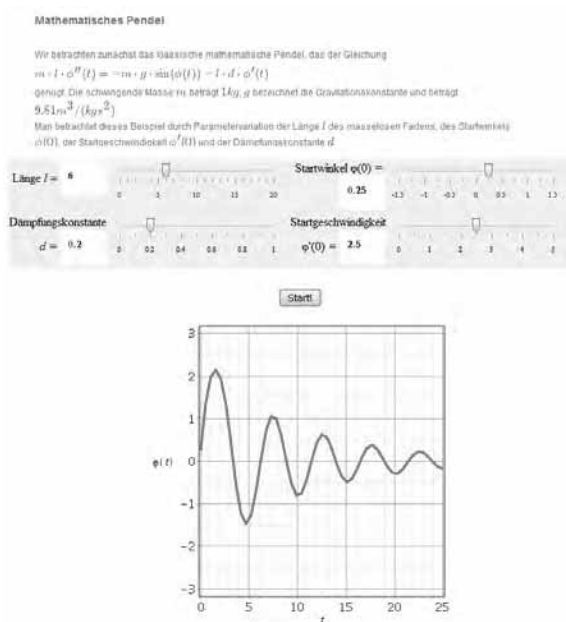


Abbildung 4: Das mathematische Pendel als Beispiel in MTA.

In Abschnitt 2 wurde bereits der Zwischentest in den Lehrveranstaltung angesprochen. Maple T.A. bietet hier offensichtliche Vorteile. Das Tool wurde dafür entwickelt, Leistungen von Studierenden mittels Test zu bewerten. Daher ist es hier auch einfacher Testbeispiele zu entwickeln. Die MMT Beispiele benötigen zusätzliche Implementierung um als Testbeispiele fungieren zu können. In Maple T.A. kann der Wert der Parameter, welcher aus dem Regler ausgelesen wird, direkt bewertet werden, was am MMT nur in

mühsamer Zusammenarbeit mit Moodle möglich ist. Dies war auch einer der Hauptgründe die Simulationsbeispiele nach Maple T.A. zu transferieren.

Ein allerdings offensichtlicher Nachteil besteht natürlich darin, dass in Maple T.A. die Verwendung eines Maple Produkts. Es wird also in Zukunft nicht möglich sein, Beispiele in Simulink, Anylogic oder Java zu realisieren.

5 Ausblick

Abschnitt 4 hat gezeigt, dass in Maple T.A. sicher noch einiges an Potenzial hinsichtlich Layout und Outputs steckt. Man kann natürlich sagen, dass die Beschränkung der Software ein großes Defizit des Systems ist allerdings besteht die Möglichkeit auch Beispiele bzw. Mathapps in MapleSim zu erstellen. Dadurch erreicht man größere Flexibilität in der Implementations von Simulationsbeispielen.

References

- [1] S. Winkler, A. Körner und I. Hafner. *MMT - A Web-Based E-Learning System for Mathematics, Modelling and Simulation using MATLAB*. Proceedings of the 7th Congress on Modelling and Simulation, in Prag, ISBN: 978-80-01-04589-3, S. 1215 - 1221, 2010.
- [2] M. Bicher, I. Hafner, A. Bauer, C. Pöll, N. Popper und F. Breitenecker. *A Web-based Platform for E-Learning and Blended Learning in Modelling and Simulation*. International Conference on Business, Technology and Innovation, in Durres, Albanien, S. 100 - 109, 2013.
- [3] V. Urbonaite, S. Winkler und A. Körner. *Various Usage of Maple T.A. in Mathematics, Modelling and Simulation*. ERK - International Electrotechnical and Computer Science Conference, in Portoroz, Slovenien, S. 173 - 176, 2013.

Integration of Simulation-based Training for Welders

Benjamin Knoke¹, Klaus-Dieter Thoben¹

¹BIBA – Bremer Institut für Produktion und Logistik GmbH at the University of Bremen

kno@biba.uni-bremen.de

Simulation-based training for welders is continuously gaining importance. However, research on the integration of welding simulators into existing structures and processes is still scarce. In order to contribute towards a greater understanding, this paper collects concepts from areas with a long history of simulation-based training, such as medicine and aviation. These concepts are applied to provide a structured evaluation of two case studies that were conducted within two pioneer organisations. It was observed that certain levels of physical and functional fidelity were necessary for experienced trainers to accept welding simulators. Currently, the expected features are best provided by welding simulators based on augmented reality. Optimal results were yield in groups of 3-4 trainees, when every trainee is engaged into the simulation with a certain role (welding, correcting, filming, taking notes). Simulation-based training can be successfully applied towards a range of skills that includes technical and functional expertise training, problem-solving and decision-making skills, as well as interpersonal and communications skills.

1 Introduction

Vocational education and training in the field of welding is currently undergoing a transformation through the integration of simulation-based training. Although the transformation is gaining momentum from advances in multimedia technologies, the implementation of simulation-based training is still in its infancy. Much less research is available than in areas with a long history of simulation-based training, such as the medical sector. In order to contribute towards a greater understanding of the possibilities that these simulation technologies have to offer, this paper discusses the applicability of findings from other sectors towards simulation-based training for welders.

For an educated design and the integration of simulation technologies for welders, it is necessary to understand the benefits and relevant characteristics of training simulations. Although it seems that a training simulation simply improves the closer it comes to reality, the relation is instead more complex [1]. Not only can unnecessary details increase the costs of simulators [2] [3], they also bear the risk to divert the focus from the intended training [4]. Therefore, Shirts (1992) recommends to “look past the details to the essence of reality” [4].

This paper discusses concepts that contribute to the success of simulation-based training in various sectors, such as simulation fidelity, definition of training objectives, and benefits of simulation technologies. The currently available welding simulators and two case studies are analysed to investigate the current

state-of-the-art and the applicability of the aforementioned concepts into welding training.

2 Related Work

2.1 Simulation Fidelity

Early research initiated by Thorndike & Woodworth in 1901 argued that a simulated environment must have the same elements and surface features than its real counterpart in order to allow a transfer of problems [5] and to evoke engagement of the trainee [6]. This demand influenced the definition of the term “simulation fidelity” in the middle of the 20th century. Simulation fidelity is used to describe the degree to which the real operational equipment or situation is resembled by a simulator [1] [2].

Subsequent studies indicate that the correlation between simulation fidelity and training effectiveness relies on multiple dimensions. Initially, Kinkade and Wheaton (1972) defined three dimensions of simulation fidelity [7]:

- *Equipment fidelity*: the degree to which the simulator looks and feels like the original operational equipment.
- *Environmental fidelity*: the degree to which the simulator resembles the sensory stimulation and control feel of the original task situation.
- *Psychological fidelity*: the degree to which the trainee perceives the simulator as a du-

plicate of the original operational equipment and task situation.

This definition has been refined by Fink and Shriver (1978), who defined two key dimensions [8]:

- *Physical fidelity* as the degree to which a simulation represents the appearance and feel of the original equipment (previously: equipment fidelity).
- *Functional fidelity* as the degree to which the original stimulus and response options are implemented in the equipment (previously: environmental fidelity).

Hays (1980) emphasises the use of the terms physical and functional fidelity [9]. He also states that psychological fidelity and corresponding approaches can be derived from physical and functional fidelity and should therefore be discarded. Although physical fidelity and functional fidelity also show interdependencies, the functional fidelity of a training simulation is mainly considered to determine its performance [10] [11].

Although the term fidelity is widely used since its introduction, Hamstra et al. (2014) argue that the differentiation between functional and physical dimensions is not made consistently across literature [12]. The authors also emphasise the importance of functional task alignment (functional fidelity) over physical resemblance (physical fidelity), which is always determined in context of the underlying instructional goals [12].

The knowledge to differentiate between important and unnecessary fidelity is one of the key objectives during the design of simulation-based training, because minimal costs can only be achieved by selecting just the fidelity that is necessary to meet the training objectives [13].

The conclusions drawn for the methodology of this paper are (i) to distinguish between physical from functional fidelity, (ii) to analyse if the importance of functional over physical fidelity can be confirmed for welding simulators, and (iii) to benchmark functional fidelity in consideration of specific training objectives. Approaches to describe the skills that can be achieved through simulation-based training are described in the following section.

2.2 Skills that can be trained through simulation-based training

The main objective of training is the acquisition, improvement or testing of skills [14] [15] [16]. The skills that can be trained through simulations have been classified by Lateef (2010) into three main types [16]:

- Technical and functional expertise training.
- Problem-solving and decision-making skills.
- Interpersonal and communications skills or team-based competencies.

A similar categorization has been published by Larnpotang et al. (2013), who defined a skills triangle to emphasise the possible training of multiple dimensions in a single training simulation, as shown in Figure 1 [17].

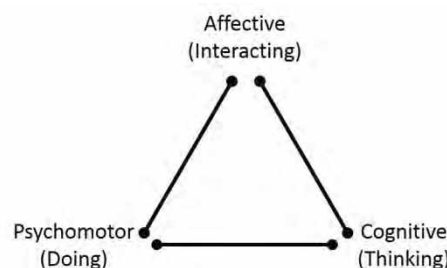


Figure 1. The skills triangle of simulation-based training [17]

In addition to an improved acquisition of skills, higher fidelity in simulation-based training can offer several benefits in comparison to regular training.

2.3 Features and uses of fidelity in simulation-based training

Issenberg et al. (2005) conducted an extensive literature review on the features and uses of successful high-fidelity medical simulations with the following results, sorted by weight [18]:

- Feedback is provided during learning experience
- Learners engage in repetitive practice
- Simulator is integrated into overall curriculum
- Learners practice with increasing levels of difficulty
- Adaptable to Multiple Learning Strategies

- Capture Clinical Variation
- Operate in a controlled Environment
- Individualised Learning
- Defined Outcomes and Benchmarks Clearly
- Simulator validity

Although the study has been conducted in the medical field, the results are expected to be applicable as a framework to examine the features and uses of fidelity in training simulations of welders.

2.4 Simulation technologies in welding simulators

Simulation technologies can be defined as materials and devices created or adapted to solve practical problems and create simulations [19]. The current key simulation technologies in welding are virtual and augmented reality. During the past decade these technologies were applied to create welding simulators and made the transition towards practical application [20] [21]. The commercial solutions that are currently available are listed in Table 1.

Manufacturer	Product	Characteristics
GSI SLV Halle	GSI SLV Halle Schweißtrainer	Real low-power arc for melt run, requires shielding gas
Seabery	Soldamatic	Augmented Reality over an artificial workpiece
Fronius International	Virtual Welding	Virtual Reality, artificial workpiece in fixed position for haptic feedback
Lincoln Electric	VRTEX	Virtual Reality, artificial workpiece in fixed position for haptic feedback
123 Certification	ARC+	Virtual Reality, no haptic feedback
EWM Hightec Welding	EWM Virtual Welding Trainer	Virtual Reality on screen, no head-gear, no haptic feedback

Table 1. Overview on available welding simulators

All of the listed simulators rely on optic measurement to capture the position and movement of a welding torch or electrode. The measured characteristics include stick out, work angle, travel angle, travel path, and travel speed.

The listed welding simulators differ greatly in physical and functional fidelity, as well as in price. Currently, the augmented reality system (Soldamatic) seems to be dominant, which has also been observed within the two case studies that are described in the following section.

3 Application of Welding Simulators

Within the course of the German research project “MESA – Medieneinsatz in der Schweißausbildung” (media applications in welding training), two diverse case studies have been conducted to analyse the implementation of welding simulators in industrial practice. The case studies include a plant for the construction of car chassis of a car manufacturer and an organisation for joint training of metal processing SMEs.

3.1 Case study 1: Car chassis manufacturing plant

The first case study was conducted at a car chassis manufacturing plant of a German car manufacturer. On average, 50 trainees periodically perform a welding training with a duration of four weeks. This includes mostly apprentices, but also advanced training sessions and individual coaching for employees.

The introduction of welding simulators started with two VRTEX-systems that were mostly used as demonstrator and had low impact on the practical training. In 2015 four Soldamatic systems have been purchased. The decision to change the system was made due to the considerably lower price and a better physical and functional fidelity of the simulation. The augmented reality system was preferred due to a more robust mode of operation and constraints of motion sickness through limited environmental perception within the virtual reality simulation. It was observed that VRTEX-systems create strong electromagnetic fields and are prone to failure, if positioned within 4-5 metres of other sources of electromagnetic fields or each other.

In collaboration with the simulator manufacturer, an individual workpiece was integrated in the simulation to enable the welding training along a curved outline

on a control arm that is actually manufactured within the chassis plant and used in welding training. The control arm was replicated as a plastic workpiece with reference markers. It is used within the welding training to increase the simulators acceptance among the trainers through an increased connection to reality and to save resources.

Each simulator is used by a group of 3-4 trainees. After the initial enthusiasm fades, it is considered important to engage every trainee in the simulation through a specific task. While one trainee welds within the simulation, the second trainee analyses an external screen of the simulation and communicates corrections. The third trainee films the process and focusses on ergonomic aspects. To make the posture more visible, a white adhesive strip is vertically taped on the back of the first trainee. If present, an optional fourth trainee is instructed to fill an evaluation sheet.

After an initial scepticism and fear for their occupational safety, the simulators are now greatly appreciated by all welding trainers. They see the simulators as an easy way to provide individual feedback that is not contested by the trainees. To further increase the acceptance among experienced welding trainers, it is important to state that the simulation should not be seen as a perfect replication. The physical fidelity is lowered through characteristics such as a low travel speed that is considered to be more like “Tai-Chi” by the trainers. However, the trainers generally observe a “significant learning effect” through the simulation-based training, implicating a high functional fidelity.

3.2 Case study 2: Joint training organisation

The second case study concerns a regional joint training organization that trains apprentices and experienced employees for 65 metal processing SMEs in central Germany.

The practical welding training is usually conducted in groups of three trainees. In addition to the 15 welding booths, the organization purchased one ARC+-system in 2013 and also one Soldamatic-system in 2016. Complaints were made that the ARC+-system is rather fragile and had to be sent to Canada multiple times for repairs. Also it allows less customization than the Soldamatic-system and provides no haptic feedback through lack of a physical workpiece.

The simulators have been integrated into basic welding training for beginners. The objectives were to ensure an ergonomic posture, train the correct param-

eters, such as travel speed and work angle, and to generate direct and individual feedback. The simulation-based training resulted in “much better results” during the trainees’ first workpieces and a focus on ergonomic aspects that was not possible in the conventional welding booths, due to blinds and personal safety gear. The welding tasks can be reset quickly and do not require the preparation of workpieces, leading to much faster exercises and a steeper learning curve. A time saving of up to 40-50% through a combination of simulation-based and conventional training was observed to teach basic welding skills to beginners. The trainers also noticed an increased level of engagement during the group sessions in simulation-based training. This appears to be linked to the gamification aspect of the simulations, as trainees contest each other for higher scores.

In addition to training, the mobile design and operational safety of the simulators also allows to use them to attract attention on local trade fairs and in events for occupational orientation.

4 Discussion of results

During the following discussion, the concepts described in section 2 are evaluated in consideration of results from the two reported case studies.

4.1 Fidelity requirements

A certain level of *physical fidelity* appears to be beneficial in order to support the integration of simulators into existing structures and processes. The case studies showed that an experienced welding trainer is more likely to approach a simulator that resembles traditional welding equipment. Therefore, the processing equipment of the most successful welding simulators is integrated into cases of traditional welding equipment and applies a similar set of controls.

Although the welding simulators perform well for their intended purpose, an experienced welder requires 3-5 “test runs” in order to generate good results on a welding simulator. This shows a certain lack of *functional fidelity* that is usually accepted, if the difference between simulation and replication is explained to the welder. The level of acceptance is higher if the simulator is seen as equipment that allows to train skills that be transferred into welding.

4.2 Skills that can be trained through welding simulators

The positive learning effect through (partial) integration of simulators in the training of welders has been

proven in multiple studies [22] [23] [24] [25] and was also observed in both case studies. The skills that can be trained through welding simulators are described in the following along the three main categories [16].

Technical and functional expertise

Technical and functional expertise comprises the main objective of practical welding training. The key parameters that are monitored by all welding simulators are stick out, work angle, travel angle, travel path, and travel speed.

In addition to the parameters that impact the weld, welding simulators have shown great potential in training of an ergonomic posture. Instead of isolated training in welding booths, where welding apprentices are mostly rated by their results, simulation-based training can be conducted under supervision or in groups to put the focus on the process. Currently, ergonomic training is not directly implemented in welding simulators, but both case studies showed great interest and applied work-arounds. Group training and video recording can be applied to create awareness for posture during welding training.

Problem-solving and decision-making skills

The current state-of-the-art welding simulators create complex simulations that depend on various parameters, such as voltage, current, shielding gas type and flow rate. These parameters have to be configured for training sessions and trainees can also experiment with the outcomes of parameter manipulation without suffering serious safety constraints.

The simulations can be applied to provide a link from theory to practice and support classroom situations through combination with an external video projector, e. g. in vocational training schools.

Most simulators include some ability to display learning material and conduct tests. However, this function has not been used within the case studies, because the simulators are usually operated in groups of 3-5 trainees, which contradicts traditional test situations.

Interpersonal and communications skills

In comparison to traditional training, the simulators are usually used in small groups of trainees. Within the chassis manufacturer plant, each group member was given a specific task during the individual training sessions (welding, correcting, filming, taking notes), which led to an increase in communication and team-work. Both case-studies also reported in-

formal competitions between the trainees as a positive effect that increases engagement.

Furthermore, the welding sector is characterised by a relatively high level of cultural diversity, which can cause language barriers. Visualisations and multi-language support are implemented in most simulators and can be applied to facilitate the teaching of technical terms.

Overall, skills in all three categories are impacted by training with welding simulators. However, interaction and communication are merely seen as a corollary of psychomotor and cognitive skills.

4.3 Features and uses of fidelity in simulation-based training

The features and uses of fidelity in simulation-based training that have been defined by Issenberg et al. (2005) within the medical sector [18] are discussed in consideration of the characteristics of welding simulators and the performed case studies.

Feedback is provided during learning experience

All of the currently successful welding simulators include feedback concerning the monitored parameters that can be shown during welding sessions. They also feature an evaluation screen that shows the course of parameters along the weld after completion.

In combination with group work, the feedback feature was considered highly important in both use cases and is expected to be a main cause for skill improvement.

Learners engage in repetitive practice

The ability to quickly restart a welding session is seen as a great advantage over traditional training, which would require preliminary work, such as cutting and grinding of the workpiece. Instead, simulation-based training sessions can be repeated under identical circumstances within the press of a button.

Simulator is integrated into overall curriculum

Multiple cases of successful implementation of welding simulators in vocational training are existing. However, the curriculum integration is still in a preliminary state and differs greatly between organisations in scope and evaluation. The MESA-project is currently performing research towards a curriculum integration of welding simulators and cooperates with the German Welding Society (DVS) to develop a structured guideline.

Learners practice with increasing levels of difficulty

Implemented difficulty levels in welding simulations vary through more or less strict evaluation, variation of tasks, or (partially) turned off feedback during the exercise. The variation of feedback was considered helpful, but not necessary during the case studies. It has been characterised as a viable approach to further prepare beginners in advance of their first actual welding training.

Adaptable to Multiple Learning Strategies

Most welding simulators can be used in learning situations of single users or groups with or without the presence of instructors. The simulation is usually displayed within a head-mounted display that resembles a welding helmet, and a small screen that is placed on the processing unit. Through connection of external video equipment, the simulations can also be used in classroom settings.

Capture (Clinical) Variation

The high variation of products and processes that confront welders in industrial practice are difficult to transfer into simulation-based training and seem to be a general barrier for simulation-based training in production.

An integration of individual products showed to have a positive effect on the acceptance of simulation-based training, and to reduce waste during the production of those products. During the case study of the car chassis manufacturer, considerable effort was spent to integrate a specific control arm into the augmented reality simulation of the Soldamatic-system. However, a low-cost or short-term variation of workpieces remains impossible, as long as the systems rely on digital representations of workpieces and an on-the-fly digitalization is not implemented.

A promising feature of augmented reality-based simulators is that the simulation is integrated into the real environment. During the simulation of MAG welding, the Soldamatic-system enforces a certain position and orientation of the headgear and welding torch towards the workpiece. This allows to mount the workpiece into confined spaces or underneath a table to simulate extraordinary processes (e. g. Figure 2), such as the welding of pressure tanks or heat exchangers. This can be applied to provide ergonomic assistance during constrained postures and to improve confidence towards complex task or unanticipated events (as in [16]).



Figure 2. Welding simulation in confined spaces through Augmented Reality

Operate in a controlled Environment

Welding simulators provide the possibility to train in groups and to freely varyate process characteristics, which is usually not possible in conventional training due to safety constraints. Withing the case studies, welding trainers expressed great delight towards the opportunity to provide support during the process and to improve consciousness for an ergonomic posture.

Individualised Learning

Individual requirements can be met through variation of time spent in the simulation and through adjustment of difficulty levels. Additionally, the simulators provide several sets of standard tasks with the most prominent types of manual welding processes: MIG, GTAW, and SMAW.

Defined Outcomes and Benchmarks Clearly

The task and benchmarks are clearly defined by all observed welding simulators and the monitored parameters are graded in percent that were achieved during the training sessions. However, beginners require a short introduction to the simulation to correctly interpret the provided feedback. A suitable tutorial is missing.

Also, welding trainers reported the neutral feedback through sensor data as a strong improvement over traditional training, where their feedback is sometimes either not taken seriously or causes unnecessary strain.

Simulator validity

The validity of welding simulators corresponds with their functional fidelity and has been proven in both use cases. System resilience was described as an

important criteria by both use cases for their decision to purchase specific welding simulators.

5 Conclusion and Outlook

During the past decade, simulation-based training for welders continuously gained importance. Since extensive research on the integration of simulators has already been conducted in other areas, such as medicine or aviation, this paper reviewed established concepts to evaluate two case studies on the integration of welding simulators that were performed in pioneer organisations.

The evaluation proved that welding simulators are greatly appreciated by welding trainers, once certain requirements are met. The expectations towards physical fidelity mostly address intuitive controls that simulate those of conventional welding equipment. It was also observed to be important that welding simulators are presented as training equipment and not as a perfect replication of reality. Otherwise, experienced welders, who usually do not perform well on their first tries, tend to get upset and focus on system flaws.

In accordance with previous research, both case studies reported a 30-50% decrease in time that is necessary to develop basic welding skills, when simulators are integrated into welding training for beginners. Best results were achieved during simulation-based training in groups of 3-4 persons, when each person was engaged into the simulation through an individual task (welding, correcting, filming, taking notes), or during individual training with an instructor.

Most features and uses of fidelity that contribute towards the success of simulations in the medical sector, also apply for welding simulators. Though, the individuality of products and processes seems to be a general barrier towards the customization of simulations in production. A demand of customization showed, as one of the two organisations spent considerable effort to integrate a specific workpiece into a welding simulator.

A promising application of augmented reality is the opportunity to fit the available standard workpieces into confined spaces. This allows the simulation of complex welding tasks and the support of ergonomic postures with relatively low effort. Following research may include experiments to transfer welding simulations into specific environments, such as pressure tanks or heat exchangers.

6 References

- [1] G. G. Miller. *Some considerations in the design and utilization of simulators for technical training (AFHRL TR-74-65)*. Air Force Human Resources Laboratory, Brooks Air Force Base, 1974.
- [2] R. T. Hays and M. J. Singer. *Simulation fidelity in training system design: Bridging the gap between reality and training*. Springer Science & Business Media, 2012.
- [3] D. H. Holding. *Transfer of training*. In: Training for performance: principles of applied human learning, editor: J. E. Morrison, John Wiley, New York, pp. 93-125, 1991.
- [4] R. G. Shirts. *10 Secrets of Successful Simulations*. Training, 29(10), pp. 79-83, 1992.
- [5] E. L. Thorndike and R. S. Woodworth. *The influence of improvement in one mental function upon the efficiency of other functions*. Psychological Review, 8, pp. 247-261, 1901.
- [6] P. Bradley. *The history of simulation in medical education and possible future directions*. Med Education, 2006(40), pp. 254-262, 2006.
- [7] R. G. Kinkade and G. R. Wheaton. *Training device design*. In: Human engineering guide to equipment design, editor: H. P. Van Cott and R. G. Kinkade, US Government Printing Office, Washington, pp. 668-699, 1972.
- [8] C. D. Fink and E. L. Shriver. *Simulators for maintenance training: Some issues, problems and areas for future research*. Kinton Inc, Alexandria, 1978.
- [9] R. T. Hays. *Simulator fidelity: A concept paper (No. ARI-TR-490)*. Army Research Institute for the Behavioral and Social Sciences. Alexandria, 1980.
- [10] J. A. Allen, R. T. Hays and L. C. Buffardi. *Maintenance training simulator fidelity and individual differences in transfer of training*. Human Factors: The Journal of the Human Factors and Ergonomics Society, 28(5), pp. 497-509, 1986.
- [11] L. Davidovitch, A. Parush and A. Shtub. *The impact of functional fidelity in simulator-based learning of project management*. International

- Journal of Engineering Education, 25(2), pp. 333-340, 2009.
- [12] S. J. Hamstra, R. Brydges, R. Hatala, B. Zendejas and D. A. Cook. *Reconsidering fidelity in simulation-based training*. Academic Medicine 89(3), pp. 387-392, 2014
 - [13] J. D. Fletcher. *Using networked simulation to assess problem solving by tactical teams*. Computers in human behaviour, 15(3), pp. 375-402, 1999.
 - [14] D. M. Gaba, S. K. Howard, B. Flanagan, B. E. Smith, K. J. Fish, and R. Botney. *Assessment of clinical performance during simulated crises using both technical and behavioral ratings*. The Journal of the American Society of Anesthesiologists, 89(1), pp. 8-18, 1998.
 - [15] A. Gupta, B. Peckler, and D. Schoken. *Introduction of hi-fidelity simulation techniques as an ideal teaching tool for upcoming emergency medicine and trauma residency programs in India*. Journal of emergencies, trauma, and shock, 1(1), pp. 15-18, 2008.
 - [16] F. Lateef. *Simulation-based learning: Just like the real thing*. Journal of Emergencies, Trauma, and Shock, 3(4), pp. 348-352, 2010.
 - [17] S. Larnpotang, D. Lizdas, D. Rajon, I. Luria, N. Gravenstein, Y. Bisht, W. Schwab, W. Friedman, F. Bova, and A. Robinson. *Mixed simulators: augmented physical simulators with virtual underlays*. Conference Proceedings IEEE Virtual Reality 2013, in Orlando, Florida, pp. 7-10, 2013.
 - [18] S. B. Issenberg, W. C. Mcgaghie, E. R. Petrusa, D. L. Gordon and R. J. Scalese. *Features and uses of high-fidelity medical simulations that lead to effective learning: a BEME systematic review*. Medical teacher, 27(1), pp. 10-28, 2005.
 - [19] D. A. Cook, R. Hatala, R. Brydges, B. Zendejas, J. H. Szostek, A. T. Wang, P. J. Erwin, J. Stanley and S. J. Hamstra. *Technology-enhanced simulation for health professions education: a systematic review and meta-analysis*. Jama 306(9), pp. 978-988, 2011.
 - [20] K. Fast, T. Gifford, and R. Yancey. *Virtual training for welding*. Conference proceedings ISMAR 2004 - Mixed and Augmented Reality Conference (IEEE), in Arlington, VA, pp. 298-299, 2004.
 - [21] S. White, M. Prachyabrued, D. Baghi, A. Aglawe, D. Reiners, C. Borst, and T. Chambers. *Virtual welder trainer*. Conference proceedings VR 2009 - Virtual Reality Conference 2009 (IEEE), in Lafayette, Louisiana, pp. 303-303, 2009.
 - [22] R. T. Stone, K. P. Watts, P. Zhong and C. S. Wei. *Physical and cognitive effects of virtual reality integrated training*. Human Factors: The Journal of the Human Factors and Ergonomics Society, 53(5). pp. 558-572, 2011.
 - [23] Y. Wang, Z. Nan, Y. Chen and Y. Hu. *Study on welder training by means of haptic guidance and virtual reality for arc welding*. Conference proceedings ROBIO'06 - International Conference on Robotics and Biomimetics (IEEE), in Kunming, China, pp. 954-958, 2006.
 - [24] M. L. L. Okimoto, P. C. Okimoto and C. E. Goldbach. *User Experience in Augmented Reality Applied to the Welding Education*. Procedia Manufacturing, 3, pp. 6223-6227, 2015.
 - [25] K. Fast, T. Gifford and R. Yancey. *Virtual training for welding*. Conference Proceedings ISMAR 2004 - Mixed and Augmented Reality Conference (IEEE), in Arlington, VA, pp. 298-299, 2004.
 - [26] D. Mavrikios, V. Karabatsou, D. Fragos and G. Chrysosolouris. *A prototype virtual reality-based demonstrator for immersive and interactive simulation of welding processes*. International Journal of Computer Integrated Manufacturing, 19(3), pp. 294-300, 2006.

7 Acknowledgment

This work has been funded by the German Federal Ministry of Education and Research (BMBF) through the research project MESA – Medieneinsatz in der Schweißausbildung (media applications in welding training, <http://mesa-projekt.de>). The authors wish to acknowledge the ministry for their support. We also wish to acknowledge our gratitude and appreciation to MESA project partners for their contribution during the development of various ideas and concepts presented in this paper.

Physical Simulation Related Exercises for the Education in the STEM Field – Approaches Based on the Physolator Framework

Dirk Eisenbiegler¹, Dietmar Gruber², Thomas Jörg²

¹University of Furtwangen, Germany

²Hector Seminar, Germany

dirk.eisenbiegler@hs-furtwangen.de

This paper presents different kinds of approaches towards using physical simulation based exercises for early teaching of STEM topics at school¹. The approaches presented in this paper are based on the Physolator physics simulation framework. This paper analyzes to which degree these approaches can be used to meet given teaching goals.

Introduction

In our world, computer based physical simulations are omnipresent. They are used for science and engineering as well as in computer games and in animations for the movie industry. Professionals working with computer based physical simulations are aware of the fact that it takes good skills at least in the following fields for producing such computer based applications: physical modeling, geometry, numerical mathematics, programming and graphics programming. The significant skills for producing physical simulations are all located in the STEM fields¹. Work in this field is interdisciplinary. It takes physicists, mathematicians and computer scientist working together in such projects.

The guiding questions of this paper: Could physical simulation be used in early teaching of mathematics, physics and computer science at school? Could this lead to more application oriented understanding of the different topics in mathematics, physics and computer science? Could this strengthen an interdisciplinary thinking for these domains?

Teaching Physics and Physical Simulation

There are numerous publications dealing with the question on how to improve teaching in science and especially in physics [7,8,9]. In a physics lecture, students shall learn the theory of physics and they shall learn how apply this knowledge to real world scenarios. Experiments play an important role when teaching physics. Experiments are used to confirm theoretical models. On the other hand side, experiments are stimulated by theory. Students should learn, how theoretical models are used to describe

nature and how experiments in nature are used to verify theoretical models.

Physical simulation is a supplement to theory and experiment. Working with physical models means applying theoretical knowledge [6]. Just like real world experiments, physical simulations are motivated by theory. The students have to analyze the simulation runs to see if or if not the physical simulation confirms their expectations about the real world behavior. Physical simulations can never be a substitute for experiments. Physical simulations are used for verification and clarification: Is the physical simulation, which is based on the theoretical formulas, consistent with the real world observations from experiments?

A learner who is starting to set up his first physical simulation is confronted mainly with two different challenges: Understanding the physical model and understanding the way of implementing the physical model using a programming language. The following aspects have to be considered when learning programming: learn the syntax of a programming language, learn the fundamental, imperative concepts of programming and optionally learn the concepts of object orientated programming such as classes and instances, inheritance and dynamic binding. From a didactic point of view it is important to keep the different aspects of programming as separated as possible and to teach them in a step-by-step manner.

The approaches presented in this paper are based on the Physolator framework (see www.physolator.de). Physolator uses the Java programming language. Java is taught in many secondary schools. Frameworks such as Mathematica or Matlab could also be used for physical simulations. However, they come with their

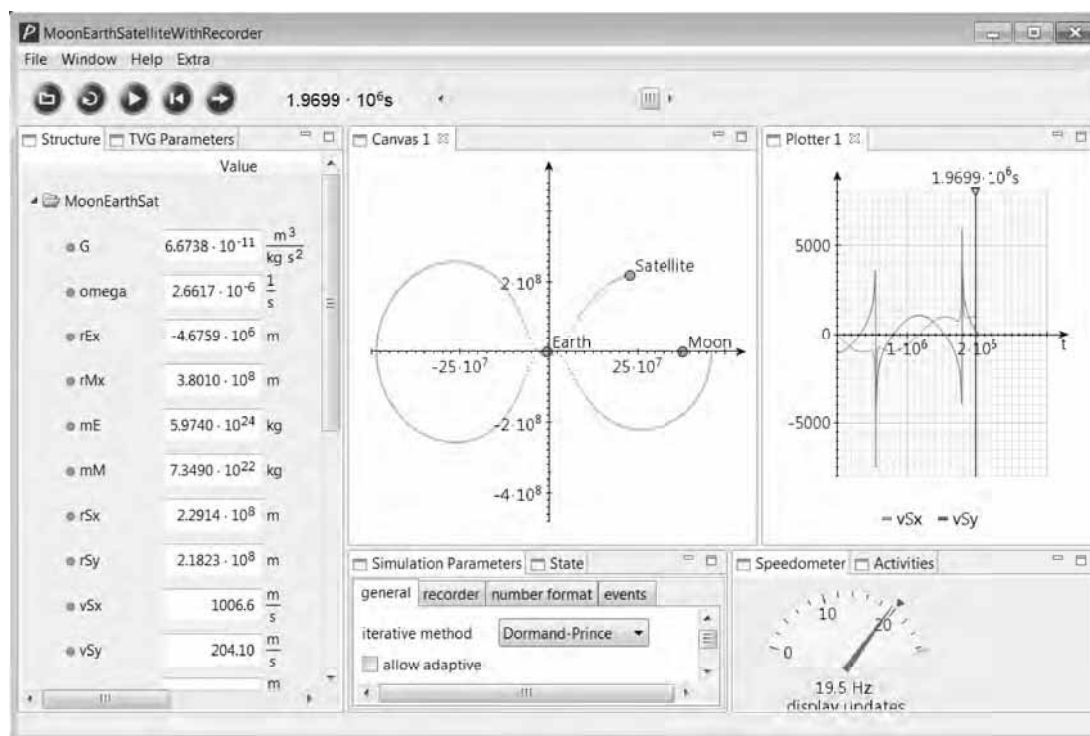


Figure 1. The Physolator Framework

own programming language and the student has to learn this extra programming language to get started with such frameworks. The Physolator is Java based. Students do not have to learn an extra programming language before they get started with physical simulations. Applying Java in the context of physical simulations helps students to acquire a deeper understanding of Java since implementing a physical simulation is an exercise of already taught programming lessons. Furthermore, Physolator is able to encapsulate the higher level OOP parts of Java, therefore entanglements of too many didactic aspects can be avoided effectively.

An exercise on the beginner level should put a focus on one topic only: One exercise for learning the meaning of gravity by playing with a given model, another exercise for learning how to build a physical model, another exercise for learning how to build graphics components. Each of these topics should first be learned in an isolated way. When for example building a first physical model in a science class, students should put a focus on understanding the underlying physical context, and the corresponding formulas. Students should not at the same time have to deal with the complexity of modern object oriented

programming or with numerical mathematics or with graphics programming.

The Physolator Framework

Programming a physical simulation from scratch is considered to be very challenging. The Physolator is designed for physical simulations at the beginner's level. Physical systems are implemented as Java programs. To build your own physical system up and running, you write a piece of Java code, load it to the Physolator framework and then start the simulation by pressing the start button inside the Physolator.

The Physolator framework is based on ODE solvers. From the Physolator's perspective, physical values are initial value problems and it uses ODE solvers for executing the simulations. Besides the ODE solvers, the Physolator framework also supports an event oriented programming style for simulating physical events such as collisions.

Figure 1 shows a snapshot of the Physolator framework. A physical system with a satellite revolving around moon and earth has already been loaded. Physical variables are always displayed on the left. Their values change during simulation. On the middle there is a graphical representation of the physical system and on the right there is a plotter for

displaying the function graphs of selected variables. With the round buttons on the top you can load physical systems, start and stop them. During simulation time the user may also interfere and change the variables values manually. Editing the variables immediately changes the state of the system and its graphical representation.

The Physolator has been designed for modular style of implementing physical systems. With the Physolator one can build a set of basic physical components. Using this set of physical components, one can build more complex components by just joining together the basic components. Also the graphical components for visualizing the simulation results as well as the numerical procedures are developed independently. For every exercise the instructors provide the students with a set of base components: physical components, graphics components and numerical procedures. During the exercise, the students have to build a physical system on top of this infrastructure. They can focus their work on a very specific task. The given infrastructure of components should cover all the aspects the students should not have to take care off.

This paper presents different categories of exercises related to physical simulation with a focus on exercises at the beginner level. The different categories represent different categories of didactic concepts.

With the Physolator one could also define exercises for advanced students such as programming numerical algorithms. Such advanced tasks, however, are not part of the scope of this paper. This paper limits itself to exercises that are well suited for students at school level or at the beginner level at a university.

Category 1

Experimentation with a Given Physical System

Traditionally, physical experiments are based on mechanical or electrical devices. During the exercise, the students build an apparatus by assembling these devices and then execute different runs with varying parameters and conditions. Such experiments pursue the following teaching goals:

- Give the students a practical experience of the theoretical physical concepts presented during

the theoretical parts of the lecture (e.g. gravitation, friction...).

- Make sure, the students can apply theoretical concepts to the real world: relationship between variables and formulas in the theoretical world and observations in the real world.
- Explain to the students the meaning of physical models computer based simulations as a part of the scientific research process of physicists.

Readymade computer simulations can be a complement for such real experiments. For good real world examples you need the right devices. Some of them are costly and assembling an apparatus is time consuming. Some experiments that would be useful to for a better understanding of the physical domain, simply cannot be run in a classroom. One can simulate the orbit of a satellite on the computer (see figure 1) – but do this inside a classroom.

The moon-earth-satellite example is a physical system that is well suited exercises of this category. Exercises: Give the satellite the right initial position and speed and observe the path of its movement! Try find an initial position and speed so the satellite runs on a closed orbit! During simulation time observe the relevant forces and accelerations: gravity, Coriolis force, centrifugal force!

In this kind of experiment, the student loads a readymade physical system to the Physolator. In this virtual experiment interacting with the physical system means “playing” with the physical variables and observing the impact on the behavior of the physical system.

Category 2

Building a Physical Model

In this kind of experiments, mathematical formulas shall be used to build a model of a simple physical system. The students shall write down the physical variables and physical formulas in Java notation and then load and start their physical system.

The learning tasks to be pursued with this kind of exercises are:

- Learn, how to describe a consistent physical model using physical variables, formulas, and

derivation relationships and provide the physical system with an initial state.

- Understand the relationship between formulas defining the behavior and temporal progressions of physical systems following the rules defined by these formulas.
- Learn how to define physical models and learn about the limitations of physical models.

In this kind of exercise, students write Java program code. The programming language Java, however, is used in a very limited way. In this context it is only used for writing down physical variables and formulas. The entire program code only consists of Java variable declarations and value assignments to variables. The Java variables correspond to physical variables. Java variables are a means for representing physical variables in a computer. Variable assignments are a means for representing formulas. The variable assignment in Java assigns a value to the variable. The value is defined by a mathematical expression. The mathematical expression represents the formula.

The following program code gives an example for such a physical system.

```
public class SpringMassPendulum
    extends PhysicalSystem {

    @V(unit = "m")
    public double p0 = 1;

    @V(unit = "m")
    public double g = -9.81;

    @V(unit = "kg")
    public double m = 0.03;

    @V(unit = "N/m")
    public double D = 2;

    @V(unit = "")
    public double k = 0.05;

    @V(unit = "N")
    public double F;

    @V(unit = "m", derivative = "v")
    public double x = 0;

    @V(unit = "m/s", derivative = "a")
    public double v;

    @V(unit = "m/s^2")
    public double a;
```

```
public void f(double t, double h) {
    F = g * m + D * (p0 - x) - k * v;
    a = F / m;
}

public void initPlotterDescriptors(
    PlotterParameters r) {
    r.add("x,v,F", 5, -7, 7);
}
```

The program code above defines a spring mass pendulum with damping. The mass m , the pivot position $p0$, the earth acceleration g , the spring constant D and the coefficient of friction k are given constants. The position of the point mass x and its actual velocity v define the state of the physical system. Annotations $@V$ are used for attaching physical units to the variables and for defining derivation relationships. In this case, v is the first derivative of x and a is the first derivative of v . The formulas are defined inside method f . These formulas define the actual force and the actual acceleration.

This kind of Java program represents a physical system. The students learn, that this kind of notation is used for writing down physical variables, formulas and derivation relationships. At that time, the students do not necessarily have to understand Java. There are no concepts being used that go beyond physical variables and formulas – no control structures, no parameters, no methods, no exception handling etc..

The *initPlotterDescriptors* method declares, that x , v and F shall be plotted during simulation. Figure 2 shows the result of the simulation run.

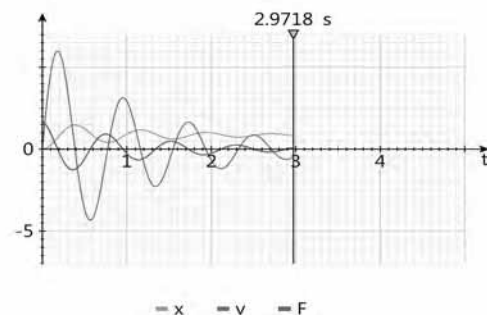


Figure 2. Spring mass pendulum with damping

Category 3 Building a Physical System by composing given Physical Components

In this kind of experiment, physical systems are build up by composing given physical components. The

For such systems, one may also provide graphics components that automatically visually represent the state of the system on the screen. *MechanicsTVG* is a generic 2D graphics component for visualizing simple mechanical systems. Adding the following program code to the above physical system results in an additional graphics component drawing the point masses and springs on the screen (see figure 4).

```
public void initGraphicsComponents(  
    GraphicsComponents g, Structure s,  
    Recorder r, SimulationParameters sp) {  
    g.addTVG(new MechanicsTVG(this, s, r));  
}
```

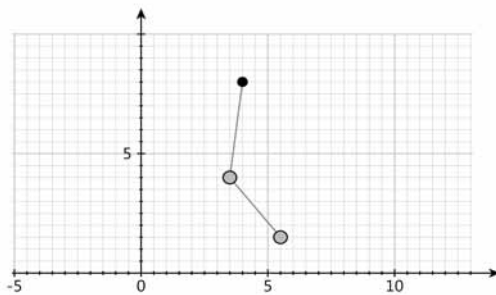


Figure 4. Spring mass double pendulum

Category 4 Graphics Programming

Simple two dimensional graphics programming is well suited for programming exercises at the beginner's level. When the program draws lines and circles on the screen, students get an immediate visual feedback, showing if the program is doing what it is supposed to do. At the same time, the students also have to deal with geometry in an applied manner.

For a simple two dimensional graphics programming, you would not necessarily need the Physolator framework. If your students, however, have already built a physical model such as the examples from category 2 or 3, then it makes perfect sense to define an exercise, where the students program a graphics component that graphically displays the state of the physical system.

During a physical simulation, the graphics are in motion. The graphic represents the physical system state and the physical system state changes during time. Time dependent graphics are nothing but movies. With the Physolator, you can also produce movies without having to deal with any physics:

Build an empty physical system, let this physical system load a graphics component and from inside the graphics component access the actual value of the simulation time t .

The Physolator framework also supports three dimensional graphics based on OpenGL. Three dimensional graphics programming is far more challenging. In such an exercise the students have to learn, how to define a 3D environment with a given camera position and camera direction, certain sources of light, 3D objects of certain shapes and a certain reflection behavior of their surfaces, fog, etc..

Category 5 Object Oriented Programming

Teaching object oriented programming concepts is not easy. The teaching goals of such lectures are:

- Learn the basic language constructs and concepts of an object oriented programming language: classes, instance, constructors, encapsulation and inheritance.
- Learn, how to apply these language constructs for developing complex software system and build a software structure that is designed for reusability.

Object oriented programming pays out when developing complex software. Unfortunately, in a programming class the time for the practical exercises is very limited and this is why in such exercises usually only small pieces of program code are produced. For small sized problems, it is hard to explain, that object oriented techniques are superior to the quick and dirty approach without a welldefined object oriented structure.

In a category 5 exercise, students shall use object oriented programming techniques to build their own physical components and use them within physical systems. Before starting with a category 5 exercise, students should first do some category 3 exercises. In a category 3 exercise, students have learn, how to build a physical system by composing given physical components. In a category 3 exercise, the students are using the notations from object oriented programming without necessarily understanding, that this program code is about object oriented programming and that the program code deals with classes, uses constructors and creates instances of classes. As a preparation for a category 5 exercise, students have to

learn the meaning of these language concepts. Then the students shall use these concepts to build their own physical components and use them inside physical systems.

An exercise from this category could ask the students to build the physical components from scratch. As an example, the program code for point masses and springs is easy to implement. Examples with point masses and springs can be found in [4]. This book also thoroughly discusses different kinds of modeling techniques using these examples. Other examples for physical components and the object oriented approaches being used to implement them, can be found in [3] and [5].

An object oriented modeling exercise does not necessarily have to start from scratch. Inheritance can also start with a given example. Sample-Scenario: Build a physical system which is based on the following physical system, but replace the graphics component with your own graphics component and use different simulation parameters. Another example, where inheritance is used on the level of physical components: Build a physical system, that is based on the string double pendulum example from category 3, but replace the given linear springs (Hooke's law) with nonlinear plate springs. The following program code uses inheritance to define *PlateString2D* a son class of *Spring2D*. Due to the fact, that *PlateString2D* inherits from *Spring2D*, one can modify the spring double pendulum code by replacing all occurrences of *Spring2D* by *PlateString2D* and then run the same example with plate springs.

The program code below adds the relevant parameters of a plate springs ($F0$, p and h) and uses overwriting to provide the class with a new implementation of method *computeF* in order to define the physical behavior of the plate spring.

```
public class PlateString2D extends Spring2D {

    public double F0;
    public double p;
    public double h;

    public PlateString2D(double F0, double p,
        double h) {
        this.F0 = F0;
        this.p = p;
        this.h = h;
    }

    public double computeF(double distance) {
```

```
        double sigma = distance / h;
        return F0 * sigma *
            ((1 - sigma) * (1 - 0.5 * sigma) *
            Math.pow(h / p, 2) + 1);
    }
}
```

Category 6 Learning Basics about Physical Simulation

In the exercises from the previous categories, students run physical simulations without necessarily understanding, how physical simulations are executed. There are quite some facts, that students could learn about physical simulations and there are exercises for deepening the understanding. This kind of knowledge is not only applicable to physical simulation, but also gives the students a better understanding about the core principles of computer games and computer based animations.

The teaching goals to be pursued in this category of exercises.

- Understand, that a physical simulation is executed in a time discrete manner.
- Learn, that a numerical simulation has limited accuracy. Understand, that a smaller step width results in a more precise simulation run, but result in a higher computational effort.
- Learn, that there some physical systems such as a simple trajectory, where there are algebraic ways to describe the behavior of a physical system with respect to time (closed solutions). In most other cases, the computer based simulations have to be used.
- Learn about different kinds of numerical procedures: fixed step width vs. flexible step width, single step vs. multi-step procedures.

The exercises from this category shall deepen the understanding about this domain. The Physolator framework provides several simulation parameters. The simulation parameters allow the user to choose, how the simulation is carried out. Among others, the user may choose a step width and the user may also choose among different kinds of numerical procedures (ODE solvers) such as Runge-Kutta, Adam-Bashforth, Cash-Karp and Dormand-Prince. In an exercise, students could be asked to load a given physical system and then find the right simulation parameter settings so the simulation runs with a

highest possible accuracy and with a minimum of computational effort (CPU time consumption). The Moon-Earth-Satellite from figure 1 would be well suited for this purpose – but one could also use any other physical system. In this exercise, students have to choose the right numerical procedure and the right step width. The computational effort for every simulation run can be monitored using the Physolator's built in monitoring tools (see figure 5).

Students working on this exercise will quickly realize, that the accuracy and the computational effort are depending on the underlying numerical procedure as well as the step width. They will first have to produce a very precise result using very small step widths thus decreasing the truncation error. Too small step widths, however, lead to an increase of the round-off error. By working on such an exercise, the student get an awareness of the fact, that simulations are run in a time discrete manner and that the total error increases the longer the simulation runs. In a follow-up lecture one could explain the underlying theoretical problems.

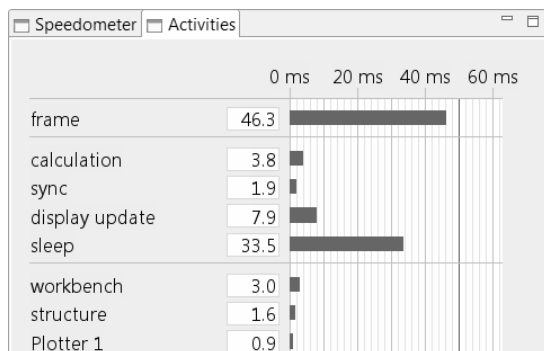


Figure 5. Performance monitor

A variation of this kind of exercise would work with a physical system, where a direct, algebraic solution exists. Examples for such physical systems: simple trajectory, damped point-spring-pendulum. Other than in the previous type of exercise, a precise solution is given and therefore it is easy to compute the error at any time. Working with such an example would also deepen the awareness, that at least for some physical systems, the behavior of physical systems can be described using closed equations and a physical simulation is not necessarily required in these cases.

Summary and Conclusion

This paper has presented different categories of exercises related to physical simulation and it has been explained, how the Physolator simulation

framework can be used as an infrastructure for such exercises.

It has been shown, that the different types of exercises pursue different kinds of teaching goals. All of the teaching goals for such exercises are in the STEM field. In many exercises, several STEM qualifications are needed: physics, programming, mathematics. These qualifications have to be combined when working on the exercise. This is why physical simulation is a domain, where students not only acquire knowledge from different STEM fields, but also learn, how to work in an interdisciplinary manner and combine these skills.

References

- [1] D. Eisenbiegler, "The Software Architecture of the Physolator – a Physical Simulation Framework", MSAM 2015, Atlantis Press, pp. 61-64.
- [2] D. Eisenbiegler, "Object Oriented Modeling and Simulation with the Physolator – Getting Started", <https://opus.hs-furtwangen.de/frontdoor/index/index/docId/614>, 2016.
- [3] D. Eisenbiegler, "A Generic Particle Modeling Library for Fluid Simulation", AMSM 2016, Atlantis Press.
- [4] D. Eisenbiegler, "Objektorientierte Modellierung und Simulation physikalischer Systeme mit dem Physolator", BoD Norderstedt, 2015.
- [5] D. Eisenbiegler, "An Object Oriented Library for Acoustics Simulation Based on the Physolator Simulation Framework", CMSAM 2016, DEStech Publications.
- [6] B. Bloom, "Taxonomy of Educational Objectives", Pearson Education, 1984
- [7] R. Pintó, D. Couso (editors), "Contributions from Science Education Research", Springer, 2014
- [8] N. Papadouris, A. Hadjigeorgiou, C. Constantinou (editors), "Insights from Research in Science Teaching and Learning", Springer, 2013.
- [9] Mikelskis-Seifert, S. et.al., "Physik-Methodik", Cornelson Stuttgart, 2010.

Modellierung und Simulation der langfristigen Bildungspolitik in Deutschland und Europa

Thomas Wiedemann¹

¹ HTW Dresden Fakultät Informatik/Mathematik

wiedem@informatik.htw-dresden.de

Die Bildungspolitik steht in Deutschland und Europa vor sehr großen Herausforderungen. Massive Sparzwänge oder Finanznöte haben in den letzten 25 Jahren insbesondere in den östlichen Bundesländern und im gesamten osteuropäischen Raum zu einer extremen Zurückhaltung bei der Einstellung von jungen Lehrern und Hochschullehrern geführt. Die Folge ist eine massive Überalterung der aktuellen Belegschaft, welche in den nächsten 10 Jahren zu großen Teilen in den Ruhestand gehen wird. Es ist zur Zeit schwer ersichtlich, wie die verantwortlichen Administrationen das entstehende Defizit an Lehrpersonal ausgleichen können – ein massiver Verlust an Lehrpersonal einhergehend mit einem Qualitätsabfall der Lehre, vor allem auch bedingt durch den Einsatz vieler Seiteneinsteiger in Schulen und Hochschulen, ist kaum noch zu vermeiden. Natürlich ist mit den Methoden der Simulation eine ursächliche Lösung der Probleme nicht möglich. Es können aber die langfristigen Entwicklungen auf der Basis der exzellenten Personaldatenlage genau vorhergesagt und mögliche Auswirkungen von politischen und administrativen Maßnahmen in Ihrer Wirkung sehr exakt beurteilt werden. Da sich die genauen Daten und Bildungsregeln in jedem Land oder Bundesland stark unterscheiden, die primären Hochrechnungen zur Alterung der Lehrerschaft jedoch identisch sind, erscheint eine überregionale Zusammenarbeit sehr sinnvoll. Als Werkzeuge werden sowohl eingeführte Tools wie MatLab als auch eine webbasierte Modellierungsumgebung auf der Basis von aktuellen JavaScript-Bibliotheken verwendet.

1 Einleitung und Motivation

Im Rahmen der allgemeinen Finanzpolitik und der jeweiligen regionalen Wirtschaftslage sind in den letzten 25 Jahren seit den großen politischen Veränderungen 1990 in den östlichen Bundesländern Deutschlands und auch in den meisten Ländern Osteuropas nur in sehr geringem Maße Einstellungen von jungen Lehrern und Hochschullehrern erfolgt. Diese Maßnahmen waren eine gewisse Zeit durchaus sinnvoll, da sich durch die wirtschaftlichen Umbrüche auch die Geburtenzahlen mehr als halbiert haben und damit auch nur weniger als halbe Kapazitäten notwendig waren. Nach nun 25 Jahren dieser Nicht-Einstellungspolitik zeigen sich die langfristigen Folgen:

- Es gibt in vielen sächsischen Schulen nur wenig Lehrer unter 50 Jahre (vgl. [1]). Über 50% der sächsischen Lehrerschaft werden in den nächsten 8 Jahren in den Ruhestand gehen. Der Ersatz durch junge Lehramtsabsolventen ist nur zu 65 bis 90% abgedeckt. In einzelnen Fächern wie Mathematik beträgt die Ersatzquote nur etwa 15%, es entsteht also wenigstens in den nächsten 5 Jahren ein jährliches Defizit von bis zu 85% der Ruhestandsabgänge.

- Das Durchschnittsalter der Professorenschaft an der TU Sofia in Bulgarien lag 2011 bei 58 Jahren und wird jedes Jahr ein Jahr höher, weil seit 1990 kaum junge Professoren berufen werden konnten.

Es ist völlig unklar, wie bei diesen öffentlich bekannten Zahlen in 15 Jahren noch eine qualitativ hochwertige Lehre durchgeführt werden kann. Erschwerend kommt hinzu, dass diese Zahlen den aktuellen Istzustand genau beschreiben, jedoch viele Administrationen den 15-Jahres-Forecast unbewusst oder bewusst aus politischen Gründen nicht durchführen. Mit Simulation ist dieser Umstand leider nicht behebbar, es kann aber dieser Forecast genau berechnet und in der Öffentlichkeit publiziert werden, in der Hoffnung, die negativen Folgen noch vermeiden zu können.

2 Anforderungen an ein überregionales Bildungs-Modell

Bei einem strategisch derart bedeutsamen Thema sind die Glaubwürdigkeit und das Vertrauen in die Ergebnisse der Berechnungen von zentraler Bedeutung. Aus aktueller Sicht erscheint es daher sinnvoll, auf die im M&S-Bereich sonst üblichen Vereinfachungen und Abstraktionen zu verzichten und die Modellierung so nah wie möglich an den aktuellen Daten

durchzuführen. Im Extremfall könnte das entstehende Werkzeug so nicht nur zum langfristigen Forecast, sondern auch zur tagesaktuellen Verwaltung der Bildungsressourcen, falls gewünscht und datenschutztechnisch möglich, geeignet sein.

2.1 Das Datenmodell

Bei einer Konzeption eines neuen, sehr komplexen Simulationsmodells erscheint es im Jahr 2016 sinnvoll, von vornherein alle Daten in einer relationalen Datenbank zu speichern. Das primäre Datenbankmodell sowohl in der externen DB wie auch Simulationsmodell besteht nach Abb. 1 aus 3 wesentlichen Teilen bzw. Tabellen, welche alle über M:N-Relationen miteinander verbunden sind.

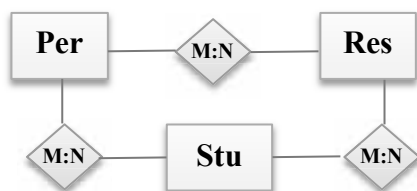


Abbildung 1. Das primäre Datenmodell

Die Entität „Per“ modelliert die aktiven Personen: alle Lehrer und/oder Hochschullehrer und auch alle sonstigen, für den Lehrprozess nötigen Mitarbeiter. Die Entität „Res“ bildet alle nötigen Ressourcen ab, wie z.B. Schulen, Fakultäten, Räume und Laborgeräte, welche auch konkurrierend genutzt werden können. Die Entität „Stu“ soll alle Studenten und Schüler abbilden. Sowohl „Per“ wie auch „Stu“ müssen eine 100%ig genaue Abbildung der vorliegenden Ist-Situation erlauben, es muss also der einzelne, real existierende Lehrer in der Entität identifizierbar sein. Ob dies auch für Studenten bzw. Schüler unbedingt notwendig ist, muss diskutiert werden. Es reicht eventuell auch nur eine klassen- bzw. seminargruppenweise Abbildung mit einer Zahl als Wert und damit ohne einzelne Stu-Individuen.

Da sowohl „Per“ wie auch „Stu“ Menschen abbilden und sich diese exakt mit der Zeit kohortenweise weiter entwickeln, muss das Datenmodell zur Initialisierung und Ergebnisspeicherung der Simulation den aktuellen Wert auch beliebig viele Werte aus der Vergangenheit und Zukunft speichern können. Beide Tabellen sind damit über die Zeitdimension zu entwickeln. Am Beispiel von „Per“ muss das Datenmodell jeden Lehrer über das gesamte Berufsleben abbilden können. Weiterhin muss auch zwischen Stammdaten

und Verlaufsdaten unterschieden werden, wobei die Attributsausprägungen der Entitäten gleich sein können. Im Sinne eines kanonischen Datenmodells können sowohl Stamm- wie auch Verlaufsdaten in einer Tabelle gespeichert werden, wenn eine Unterscheidung durch entsprechende Attribute möglich ist.

2.2 Die Simulationsdurchführung

Unter dem Gesichtspunkt einer genauen Modellierung einzelner Lehrerindividuen und der anderen Ressourcen entspricht das Modell eher einem diskreten Simulationsmodell, welches aber in gewisser Analogie zur einfach getakteten kontinuierlichen Simulation weiterentwickelt wird.

Da in der Bildungsadministration Prozesse, Konfigurationen und damit Änderungen häufig schulhalbjahr- oder semesterbezogen stattfinden, wäre eine semesterweise Zeitfortschaltung denkbar. Allerdings sind selbst in den deutschen Bundesländern die Ferienzeiten extrem unterschiedliche und auch noch hochdynamisch über die Jahre hinweg veränderlich. Bei einer Konzeption eines überregional einsetzbaren Systems mit einem langfristigen Modellierungsanspruch von Wechselwirkungen zwischen einzelnen Teilmodellen (ein Lehrer oder Schüler wechselt das Bundesland oder den Staat) erscheint daher die „Woche“ als Zeitmaßstab sinnvoll. In Hochschulen ist damit auch der turnusmäßige Wechsel zwischen unterschiedlichen Stundenplänen für die gerade und ungerade Woche abbildbar.

Die Simulationsrechnung erfolgt nach folgendem Algorithmus:

1. Berechnung von möglichen deterministischen (und bei Bedarf auch stochastischen) Änderungen über alle Entitätseinträge (z.B. Lehrerruhestand, starke Grippewelle, Ausfall Rechnerlabor etc.) und Ablage in eine temporäre Zukunftstabelle (oder die nullte Ebene aller Entities wie `stu[0][pid][time]`)
2. Finale Übertragung der neuen Werte für alle Entitätseinträge auf das neue Zeitintervall `time+1`
3. Prüfung auf Simulationsende und ansonsten Wiederholung ab 1.

Bei der Simulation sind die Werte für alle Zeitschritte in der Ergebnisdatenbank zu speichern.

2.3 Die Implementierung des Simulationsmodells

Die Abbildung des Modells auf eine entsprechende Implementierungsbasis ist aufgrund der tabellenorientierten Datenmodellierung relativ flexibel.

Eines der am häufigsten eingesetzten Softwarelösungen ist MatLAB, welches neben der eigentlichen Berechnung auch eine große Anzahl an Schnittstellen und Auswertungsroutinen bereitstellt [2].

Eine zweite Option sind allgemeine Programmiersprachen. Als eine neue Variante kommt dabei auch JavaScript in's Spiel, welches nun sowohl auf dem Client wie auch auf dem Web-Server sehr gute Laufzeitwerte erreicht [3]. Eine der wesentlichen Ursachen ist die von Google entwickelte V8-JavaScript-Engine, welche bei kluger Optimierung des JS-Codes eine Ausführungsgeschwindigkeit in der Größenordnung von C/C++ erlaubt [4].

Beide Toolklassen haben Ihre Vor – und Nachteile:

- Matlab ist ein allgemein anerkanntes und weit verbreitetes Analysetool und unterstützt die Auswertung der Daten mit einer Vielzahl von Zusatzbibliotheken. Allerdings stellt sich die Frage, ob in Ministerien und Schulen die entsprechenden Lizenzen vorhanden sind oder beschafft werden können, was die Akzeptanz etwas einschränken kann, kostenfreie Run-time-Viewer können eine sinnvolle Lösung dieses Problems sein.
- Eine web-basierte Berechnung mit JavaScript-basierten Bibliotheken ist in der heutigen Zeit ein sehr kostengünstiges und sofort, ohne aufwändige Installation im Browser aufrufbares Werkzeug.

Das Simulationsmodell soll in beiden Tools testweise entwickelt werden. Dabei soll anschließend geprüft werden, ob zentrale Teile der Verhaltensbeschreibung nur einmal mit einer abstrahierten Beschreibung definiert und dann automatisiert in beide Tools transformiert werden könnten. Eine derartige automatische Transformation würde auch eine Einbeziehung von weiteren Technologien wie R zur Modellierung erlauben.

3 Simulationsergebnisse

Auf der Basis der statistischen Daten aus [1], welche zukünftig durch eine personengenaue Abbildung eines jedes Lehrers untersetzt werden müssen, wurde eine stochastische Belegung der Tabelle *Per* generiert. Wie das Ergebnis in Abb. 2 zeigt, sind viele sächsische Lehrer über 50 Jahre alt und ein großer Teil wird in den nächsten 5 bis 7 Jahren in den Ruhestand gehen.



Abb. 2 Die sächsische Lehrer-Altersverteilung

Bei der Simulation wird jeder einzelne Lehrer über die Zeit entwickelt und geht im aktuellen Modell mit 66 in den Ruhestand. Die Wiederbesetzung der Stelle wurde mit einer Wahrscheinlichkeit von 80% angenommen, da bereits seit einigen Jahren durch einen generellen Lehrermangel in Sachsen und teilweise bereits auch in ganz Deutschland kaum noch arbeitslose Lehrer verfügbar sind. Die Ausbildungszahlen sind damit nicht ausreichend. Abb. 3 zeigt die entsprechenden Ergebnisse. Der entstehende Lehrermangel von ca. 6% in 5 Jahren und 10% in 10 Jahren wird wahrscheinlich durch Zusammenlegungen von Klassen und Lehrerabordnungen zwischen Schulen noch beherrschbar sein.



Abb.3 Entwicklung über alle sächsischen Lehrer

Bei der Gruppe der Mathematiklehrer in Sachsen ist die Wiederbesetzungswahrscheinlichkeit P_w leider deutlich geringer, da einem jährlichen Bedarf von ca. 80 Lehrern nur einstellige Bewerberzahlen gegenüber stehen. Bei der Berechnung wurde eher unrealistisch optimistisch von einem 5%-Zugewinn aus anderen Bundesländern oder dem Ausland ausgegangen und damit ein $P_w = 15\%$ angesetzt. Abb. 4 zeigt die Ergebnisse.

Es darf angenommen werden, dass ein Verlust von ca. 40% der Mathematiklehrer in 7 Jahren nicht auszugleichen ist. Entsprechende Notmaßnahmen durch Einstellung fachfremder Seiteneinsteiger oder schulübergreifende Zusammenlegungen zu Mathematikkursen werden notwendig sein. Damit ist zukünftig auch die Frage der Ausbildungsqualität in die Simulation mit einzubeziehen.

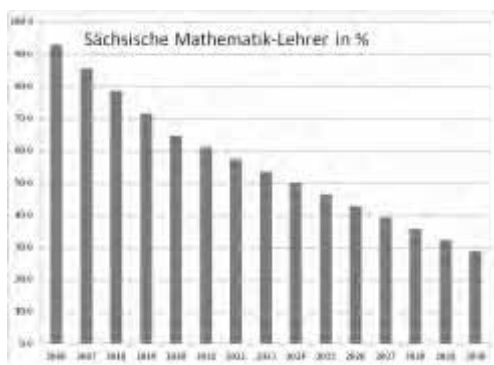


Abb.4 Entwicklung der sächsischen Mathematik-Lehrer

Noch dramatischer ist die Situation bei MINT-Hochschulprofessoren in Südost-Europa (SOE). Aus vorliegenden (leider inoffiziellen) Statistiken ist ersichtlich, dass meist nur noch Professoren über 55 Jahre unterrichten. Alle jüngeren Jahrgänge haben das Land verlassen oder haben aufgrund von zu geringen Gehältern diese Laufbahn nicht angetreten. Bei einer angenommenen $P_w=5\%$ (real ist dieser Wert nahe 0%) ergeben sich Ergebnisse nach Abb. 5.

Aus Abb. 5 ist ersichtlich, dass bereits in 5 Jahren nur noch 50% der bisherigen Professoren verfügbar sein werden und wahrscheinlich ebenfalls Notmaßnahmen ergriffen werden müssen. Im Gegensatz zu einer sofortigen Einstellung von Seiteneinsteigern im Schulbereich als quasi Neu-Lehrer dürfte diese Praxis bei Professoren kaum zielführend sein, wenn man nicht massive Qualitätsverluste in Kauf nehmen will (die Laufbahn eines Professors sollte mindestens eine Promotion und einige Jahre Lehrerfahrung vor der Berufung beinhalten). Da die SOE-Staaten diese

Aufgabe aus eigenen Mitteln kaum bewältigen können, ist die gesamte EU- aufgerufen, entsprechende Programme zur Nachbesetzung von Professorenstellen in den nächsten 5 Jahren zu unterstützen. Andernfalls kann das gesamte MINT-Hochschulwesen in SOE in den nächsten 5 bis 10 Jahren große Probleme bekommen, bis hin zur Schließung ganzer Universitäten oder Fachgebiete.



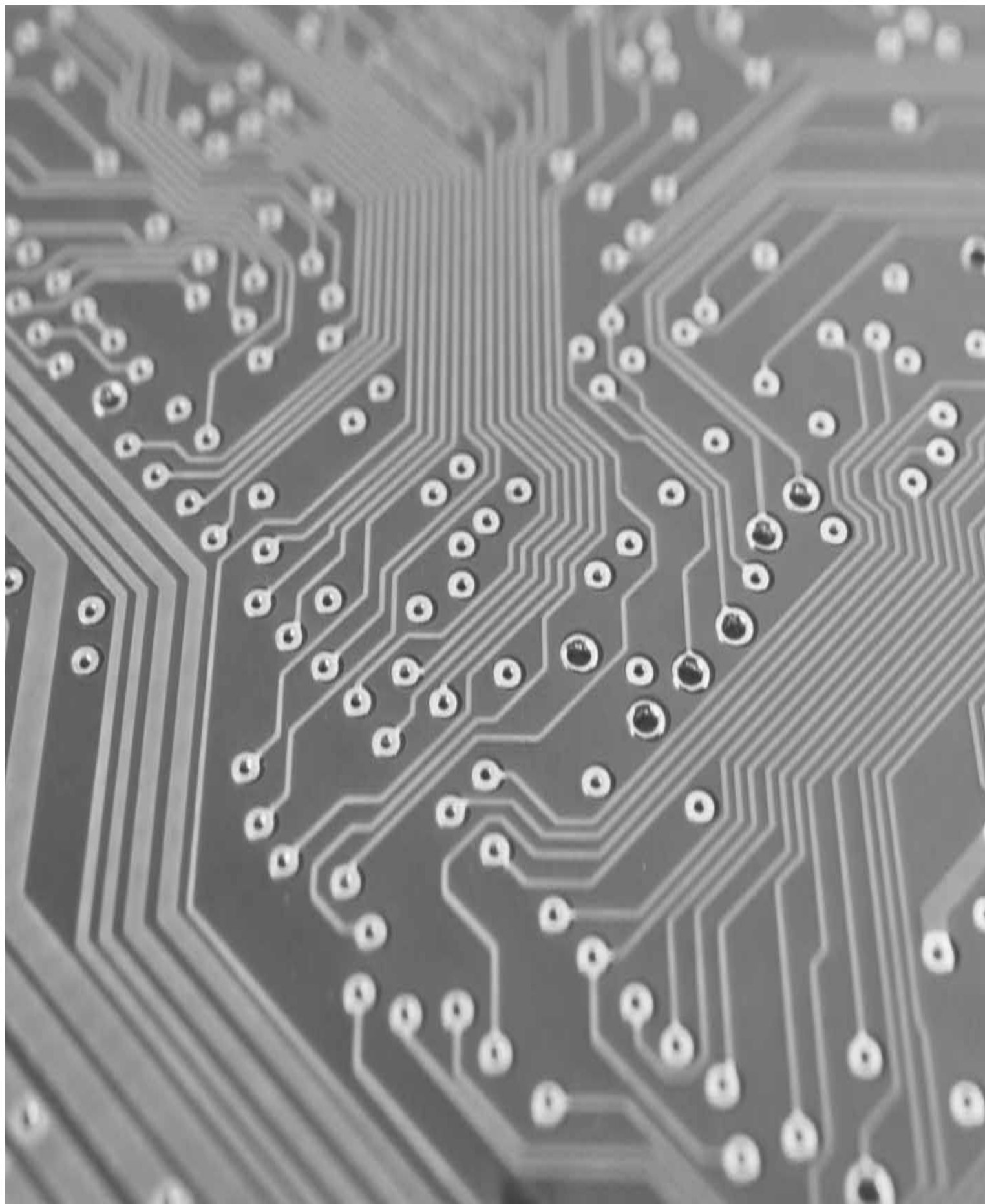
Abb. 5 Entwicklung der MINT-Professoren in SOE

4 Zusammenfassung

Die Entwicklung eines möglichst genauen Modells der Lehrer- und Hochschullehrer stellt einen ersten Schritt zur Beschreibung der zukünftigen Entwicklung im Bildungsbereich dar. Um danach wirksame Maßnahmen veranlassen zu können, sind sehr gute Visualisierungs- und Erklärungsformen zu finden, um Politiker und andere Entscheider von der Brisanz der langfristigen Entwicklung zu überzeugen und zu nachhaltig tragfähigen Maßnahmen zu ermutigen. Es sollte das Ziel der Simulations-Community sein, an diesem Prozess mitzuwirken, um die langfristige Absicherung des eigenen Nachwuchses zu gewährleisten.

References

- [1] *Altersstruktur der hauptberuflichen Lehrpersonen an allgemein bildenden Schulen in Sachsen* (Statist. Landesamt Sachsen) 2008
https://www.statistik.sachsen.de/download/300_Voe-Zeitschrift/2008_04_64-67_Klemm.pdf
- [2] Pietruszka, W.D. *MATLAB und Simulink in der Ingenieurpraxis*. Springer Vieweg. 2014
- [3] Ackermann, P. *Professionell entwickeln mit JavaScript: Design, Patterns und Praxistipps für Enterprise-fähigen Code*. Verlag Rheinwerk Computing; 1. Auflage, 2015
- [4] Google V8-Engine (web presentation).
<https://developers.google.com/v8/>



ISBN 978-3-901608-49-0

ARGESIM/ASIM-Verlag Wien, 2016