

40

Karlsruher Schriftenreihe
Fahrzeugsystemtechnik

Florian Netter

**Komplexitätsadaption integrierter
Gesamtfahrzeugsimulationen**



Scientific
Publishing

Florian Netter

**Komplexitätsadaption integrierter
Gesamtfahrzeugsimulationen**

**Karlsruher Schriftenreihe Fahrzeugsystemtechnik
Band 40**

Herausgeber

FAST **Institut für Fahrzeugsystemtechnik**

Prof. Dr. rer. nat. Frank Gauterin

Prof. Dr.-Ing. Marcus Geimer

Prof. Dr.-Ing. Peter Gratzfeld

Prof. Dr.-Ing. Frank Henning

Das Institut für Fahrzeugsystemtechnik besteht aus den eigenständigen Lehrstühlen für Bahnsystemtechnik, Fahrzeugtechnik, Leichtbautechnologie und Mobile Arbeitsmaschinen

Eine Übersicht aller bisher in dieser Schriftenreihe erschienenen Bände finden Sie am Ende des Buchs.

Komplexitätsadaption integrierter Gesamtfahrzeugsimulationen

von
Florian Netter

Dissertation, Karlsruher Institut für Technologie (KIT)
Fakultät für Maschinenbau, 2015

Impressum



Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark of Karlsruhe
Institute of Technology. Reprint using the book cover is not allowed.

www.ksp.kit.edu



*This document – excluding the cover, pictures and graphs – is licensed
under the Creative Commons Attribution-Share Alike 3.0 DE License
(CC BY-SA 3.0 DE): <http://creativecommons.org/licenses/by-sa/3.0/de/>*



*The cover page is licensed under the Creative Commons
Attribution-No Derivatives 3.0 DE License (CC BY-ND 3.0 DE):
<http://creativecommons.org/licenses/by-nd/3.0/de/>*

Print on Demand 2016

ISSN 1869-6058

ISBN 978-3-7315-0414-6

DOI: 10.5445/KSP/1000048183

Vorwort des Herausgebers

Die Fahrzeugtechnik ist gegenwärtig großen Veränderungen unterworfen. Klimawandel, die Verknappung einiger für Fahrzeugbau und -betrieb benötigter Rohstoffe, globaler Wettbewerb, gesellschaftlicher Wandel und das rapide Wachstum großer Städte erfordern neue Mobilitätslösungen, die vielfach eine Neudefinition des Fahrzeugs erforderlich machen. Die Forderungen nach Steigerung der Energieeffizienz, Emissionsreduktion, erhöhter Fahr- und Arbeitssicherheit, Benutzerfreundlichkeit und angemessenen Kosten finden ihre Antworten nicht aus der singulären Verbesserung einzelner technischer Elemente, sondern benötigen Systemverständnis und eine domänenübergreifende Optimierung der Lösungen.

Hierzu will die Karlsruher Schriftenreihe für Fahrzeugsystemtechnik einen Beitrag leisten. Für die Fahrzeuggattungen Pkw, Nfz, Mobile Arbeitsmaschinen und Bahnfahrzeuge werden Forschungsarbeiten vorgestellt, die Fahrzeugsystemtechnik auf vier Ebenen beleuchten: das Fahrzeug als komplexes mechatronisches System, die Fahrer-Fahrzeug-Interaktion, das Fahrzeug in Verkehr und Infrastruktur sowie das Fahrzeug in Gesellschaft und Umwelt.

Die Forschung an und die Entwicklung von Fahrzeugen profitieren erheblich von Modellbildung und Simulation. Es ist heute aufgrund stark gesteigener Rechenleistung möglich, immer komplexere Modelle zu erstellen, die eine realitätsnahe Simulation erlauben. Allerdings stoßen umfangreiche Modelle regelmäßig an die Grenzen der Rechnerkapazität, so dass ein Modell nicht komplexer als nötig zu gestalten ist. Dies ist nicht nur zur Verkürzung der Rechenzeiten zu fordern sondern auch um eine Überanpassung der zur Validierung des Modells verwendeten Messdaten und damit eine verringerte Vorhersagequalität des Modells zu vermeiden. Komplexe Gesamtfahrzeugmodelle sollten also die Möglichkeit bieten, ihre Komplexität an den Zweck der Simulation anzupassen, etwa durch eine Reduzierung der Ausdifferenzierung des Gesamtfahrzeugs in seine Einzelkomponenten oder durch eine Vereinfachung der Komponentenmodelle.

Hierzu liefert die vorliegende Arbeit einen Beitrag, indem sie Ansätze zu einer geeigneten Modularisierbarkeit komplexer Gesamtsystemmodelle, zur Quantifizierung von Komplexität, Genauigkeit und Extrapolationsvermögen von Modellen und zu Möglichkeiten einer rechnerausführbaren Auswahl von Modellen entsprechend des Simulationsziels vorschlägt.

Karlsruhe,
im Mai 2015

Prof. Dr. rer. nat. Frank Gauterin
Karlsruher Institut für Technologie (KIT)

Komplexitätsadaption integrierter Gesamtfahrzeugsimulationen

Zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
der Fakultät für Maschinenbau des
Karlsruher Instituts für Technologie (KIT)

genehmigte
Dissertation
von

Dipl.-Ing. Florian Netter

Tag der mündlichen Prüfung:	13. Mai 2015
Hauptreferent:	Prof. Dr. rer. nat. Frank Gauterin
Korreferent:	Prof. Dr.-Ing. Eric Sax

Kurzfassung

Komplexitätsadaption integrierter Gesamtfahrzeugsimulationen

Steigende Rechenleistung und sinkende Hardwarekosten fördern den Einsatz von Simulationen, mit deren Hilfe es möglich ist, das statische und dynamische Verhalten technischer Systeme und seiner Komponenten automatisiert, gefahrlos und wiederholbar zu untersuchen. Sogenannte integrierte Gesamtsystemsimulationen gestatten es selbst das Verhalten hoch vernetzter, simulationswerkzeugübergreifender Systeme nachzuahmen und zu prognostizieren. Dieses Potential ermöglicht heute im Rahmen der zunehmenden Virtualisierung einen Paradigmenwechsel im Produktentstehungsprozess. Dabei nehmen mit fortschreitender Detaillierung virtueller Prototypen sowohl die Anzahl an Untersuchungsmöglichkeiten als auch die Genauigkeit der Ergebnisse und Vorhersagen zu.

Im Rahmen dessen steigt zwar mit zunehmender Detaillierung die Anpassungsgüte an gegebene Messdaten. Jedoch wächst dabei auch die Gefahr der Überanpassung wodurch das Extrapolationsvermögen sinkt. Darüber hinaus nehmen aufgrund der Zunahme an Einflussfaktoren auch Testaufwand und Simulationsdauer zu, deren Beherrschung mit hohem Personal- und Kostenaufwand verbunden sind. Dementsprechend stellt sich die Frage, wie detailliert und damit komplex die ausführbaren Simulationsmodelle innerhalb einer Gesamtsystemsimulation abgebildet werden müssen. Gemäß dem heuristischen Forschungsprinzip der Sparsamkeit soll die einfachste und zugleich passendste Erklärung den komplizierteren vorgezogen werden. Um infolgedessen in einer virtuellen Entwicklung neben einer hohen Ergebnissgüte und damit einer hohen Effektivität der Simulation trotz steigender Rechenleistung auch eine kurze Simulationsdauer, also eine hohe Effizienz, gewährleisten zu können, muss in Abhängigkeit des jeweiligen Simulationszwecks ein geeigneter Kompromiss identifiziert werden.

Zu diesem Zweck werden im Rahmen dieser Arbeit Kenngrößen für Simulationsmodellkomplexität, Anpassungsgüte und Extrapolationsvermögen

erforscht, deren Zusammenspiel die Identifikation der jeweils optimalen Simulationsmodellkomplexitäten aus einer Vielzahl konkurrierender Modelle gestatten. Um für die Erstellung virtueller Prototypen bei einer integrierten Gesamtsystemsimulation die daraus resultierenden Quantifizierungsmerkmale zur Identifikation eines solchen Kompromisses nutzen zu können werden diese zudem auf Basis eines skalierenden Modularisierungsansatzes in einem Prozess zur Kopplung simulationswerkzeugübergreifender Systeme verortet.

Abstract

Complexity adaptation of simulation models in entire system simulations

Increasing computational power and decreasing hardware costs are supporting the application of simulation, enabling an automatized, dangerless and reproducible analysis of highly interconnected technical systems and all of its components. By simulating entire systems, their behavior can be imitated and predicted. Thus, a paradigm shift in virtual product engineering processes can be achieved. With an increase in detail, the number of possibilities for analysis and the accuracy for predictions are growing.

In this context indeed, with a rising level of detail, the goodness of fit to given measured data increases. However the danger of overfitting rises which is effecting a decrease in generalizability. Additionally also the efforts for testing and the duration of simulation are increasing, effecting high costs and personnel expenses. Against this backdrop the question has to be answered, how detailed and thus how complex the executable simulation models within an entire system simulation have to be designed. Following the heuristical research principle of economy, the simplest and most suitable explanation should be preferred over more complex ones. In order to achieve a high goodness of fit, namely high effectivity, and in spite of increasing computational power still maintaining a short simulation period, or in other words a high efficiency, an optimal compromise has to be identified in respect to simulation purpose.

For this reason quantification attributes for simulation model complexity, goodness of fit and generalizability are researched in this thesis, enabling model selection. To be able to use the attributes for the adaptation of simulation model complexity to simulation purpose during the creation of virtual prototypes at an entire system simulation they are additionally inserted in an existing process for coupling integrated multidomain systems on the basis of a scaling modularization approach.

Vorwort des Autors

Die Vorliegende Arbeit entstand in den Jahren 2010 bis 2013 während meiner Tätigkeit als Doktorand in der Abteilung „Tools, Frameworks und Mobile Applications“ bei der Audi Electronics Venture GmbH in Gaimersheim bei Ingolstadt. Den zahlreichen Menschen, die mich während dieser Zeit und auch danach unterstützt haben, möchte ich an dieser Stelle meinen Dank aussprechen.

Besonders danken möchte ich meinem Doktorvater Herrn Prof. Dr. rer. nat. Frank Gauterin, Leiter des Lehrstuhls für Fahrzeugtechnik des Instituts für Fahrzeugsystemtechnik (FAST) am Karlsruher Institut für Technologie (KIT), der die Arbeit wissenschaftlich betreut und mich in hohem Maß unterstützt hat. Die konstruktiven Diskussionen und sehr wertvollen Anregungen haben wesentlich zum Gelingen dieser Arbeit beigetragen.

Des Weiteren gilt mein Dank Herrn Prof. Dr.-Ing. Eric Sax, Leiter des Instituts für Technik der Informationsverarbeitung (ITIV) am Karlsruher Institut für Technologie (KIT), für die freundliche und unkomplizierte Übernahme des Korreferates, sein hohes Interesse an dieser Arbeit und die damit verbundenen Bemühungen.

Ebenso bedanke ich mich bei Herrn Gerhard Wagner und seinem Nachfolger, Herrn Roland Pfänder, Abteilungsleiter „Softwareentwicklung“ bei der Audi Electronics Venture GmbH, für das entgegengebrachte Vertrauen, mir dieses spannende Themengebiet überantwortet und ein akademisches Vorgehen unterstützt zu haben. Des Weiteren möchte ich mich bei meinen Vorgesetzten Herrn Jörg Flerlage und Herrn Stefan Andreas Bauer bedanken, die mich während und nach meiner Doktorandenzeit unterstützt und den für die Ausarbeitung notwendigen Freiraum eingeräumt haben.

Mein Dank gilt ebenso allen Kollegen, die meine Arbeit unterstützt und begleitet haben. Stellvertretende danke ich Herrn Dr. Karl Gruber und Herrn Stefan Bauer aus der Abteilung „Prozess-/Methodenentwicklung der technischen Entwicklung“ für die Übertragung des Themas. Herrn Thomas Neumayer und Herrn Andreas Kern aus der Abteilung „Entwicklung

Software-Entwicklungstools/-methoden“ danke ich für die Unterstützung bei der Realisierung der virtuellen Umfeldsimulation. Meinen Weggefährten Mathias Schrapf, Dr. rer. nat. Pedro Sebastião Correia, Dr.-Ing. Andreas Blattner, Dr. Christina Steiner, Dr. Thomas Ganslmeier, Nico Stüber und meinem Namensvetter Florian Netter danke ich sowohl für die fachlichen Diskussionen als auch die gemeinsamen überfachlichen Aktivitäten, die meine Zeit als Doktorand bereichert und mich in meinem Wesen nachhaltig geprägt haben. Meinen Bacheloranden, Masteranden und Diplomanden Herrn Steffen Burger, Herrn Björn Butterer, Herrn Johannes Schreyer, Herrn Xu Chu und Herrn Yassine Ghorbel danke ich für ihr Engagement und die jeweils sehr guten Arbeiten.

Mein herzlicher Dank gilt meiner Familie und meinen Freunden für die Unterstützung und Geduld während meines Studiums und der Doktorarbeit. Besonders hervorheben möchte ich meine Eltern, die mich während meiner Ausbildungszeit uneingeschränkt unterstützt und den notwendigen Beistand gegeben haben. Ohne ihre Zuversicht und ihren Glauben an mich wäre dieser Werdegang nicht möglich gewesen. In Dankbarkeit ist Ihnen diese Arbeit gewidmet.

Nicht zuletzt möchte ich meiner Lebensgefährtin Martina für Ihre Geduld und Unterstützung insbesondere während der Ausarbeitung dieser Arbeit danken. Sie gab mir die nötige Kraft auch nach meiner offiziellen Zeit als Doktorand neben dem Berufsalltag die zahllosen Urlaubstage und Wochenenden für deren Fertigstellung aufzuwenden.

Ingolstadt, im Dezember 2015

Florian Netter

Inhaltsverzeichnis

Vorwort des Herausgebers	I
Kurzfassung	V
Abstract	VII
Vorwort des Autors	IX
1 Einleitung	1
1.1 Problemstellung und Zielsetzung	3
1.2 Struktur der Arbeit	6
2 Grundlagen der Modellbildung und Simulation	9
2.1 Grundlagen der Systemtheorie	10
2.1.1 Definition von Signalen	10
2.1.2 Definition von Systemen	13
2.2 Modellbildung und Simulation	17
2.2.1 Modellbildung	18
2.2.2 Simulation	21
2.3 Parameteridentifikation	24
2.3.1 Methode der höchsten Wahrscheinlichkeit	25
2.3.2 Methode der kleinsten Quadrate	28
2.4 Modellverifikation und -validierung	31
2.4.1 Modellverifikation	31
2.4.2 Modellvalidierung	32
3 Modellbildung und Simulation übergreifender Systeme	35
3.1 Kopplungsvarianten simulatorübergreifender Systeme	36
3.1.1 Geschlossene Modellbildung	36
3.1.2 Verteilte Modellbildung	38

3.2	Modularisierung und Kopplung simulatorübergreifender Systeme	40
3.2.1	Methoden zur Modularisierung bei verteilter Modellbildung	41
3.2.2	Methoden zur Kopplung bei verteilter Modellbildung	42
3.3	Erstellung simulatorübergreifender integrierter Gesamtsystemsimulationen	45
3.3.1	Anwendungen basierend auf Co-Simulation	45
3.3.2	Co-Simulationsprozess zur Erstellung virtueller Prototypen	47
4	Strukturierung und Absicherung technischer Gesamtsysteme	53
4.1	Modularisierungsansatz	54
4.2	Strukturierung technischer Gesamtsysteme	55
4.2.1	Definition der Strukturelemente	56
4.2.2	Definition der Teilsystemgrenzen	59
4.2.3	Definition der Signale und Schnittstellen	60
4.3	Absicherung technischer Gesamtsysteme	62
4.3.1	Absicherung der Gesamtsystemarchitektur	62
4.3.2	Absicherung der Simulationsmodelle	63
4.4	Evaluation anhand einer Gesamtfahrzeugmodellarchitektur	66
4.4.1	Strukturierung einer Gesamtfahrzeugmodellarchitektur	67
4.4.2	Absicherung einer Gesamtfahrzeugmodellarchitektur	74
5	Quantifizierung der Simulationsmodellkomplexität	77
5.1	Definition der Simulationsmodellkomplexität	78
5.2	Quantifizierung der Modellkomplexität	79
5.2.1	Parametrische Komplexität	80
5.2.2	Stochastische Komplexität	80
5.2.3	Geometrische Komplexität	82
5.3	Applikation der Quantifizierungsmerkmale auf Simulationsmodelle	85
5.3.1	Komplexitätsquantifizierung bei unabhängiger Messung	86
5.3.2	Komplexitätsquantifizierung bei abhängiger Messung	94
5.4	Evaluation anhand mehrerer Teilsystemsimulationsmodelle	99

5.4.1	Komplexitätsquantifizierung verschiedener Batteriemodelle	100
5.4.2	Komplexitätsquantifizierung verschiedener Längsdynamikmodelle	103
6	Quantifizierung der Simulationsergebnisgüte	109
6.1	Definition der Anpassungsgüte und des Extrapolationsvermögens	110
6.2	Quantifizierung der Anpassungsgüte	112
6.3	Quantifizierung des Extrapolationsvermögens	114
6.3.1	Akaikes Informationskriterium	114
6.3.2	Bayessches Informationskriterium	116
6.3.3	Theorie der kürzesten Beschreibungslänge	120
6.4	Evaluation anhand mehrerer Teilsystems simulationsmodelle	121
6.4.1	Gütequantifizierung einer Gesamtfahrzeugmodell- architektur	122
6.4.2	Gütequantifizierung von Teilsystems simulations- modellen	124
6.4.3	Gütequantifizierung bei noch unbekanntem Extrapolationsbereich	128
7	Komplexitätsadaption integrierter Gesamtsystems simulations	135
7.1	Identifikation von Simulationszwecken	136
7.2	Ontologiebasierte Komplexitätsadaption	137
7.2.1	Definition der Ontologie	137
7.2.2	Aufbau einer Ontologie	140
7.3	Demonstration anhand einer Gesamtfahrzeugsimulation .	144
7.3.1	Integration einer Ontologie in den Prozess der Co-Simulation	144
7.3.2	Durchführung einer Gesamtfahrzeugsimulation . .	145
8	Zusammenfassung und Ausblick	149
8.1	Zusammenfassung	149
8.2	Ausblick	153
A	Anwendung der theoretischen Modellbildung	155
A.1	Erstellung eines physikalischen Ersatzmodells	155
A.2	Implementierung eines ausführbaren Modells	156

A.3	Überführung einer Differentialgleichung in die Zustandsraumdarstellung	158
A.3.1	Theoretische Grundlagen	158
A.3.2	Applikation auf das physikalische Ersatzmodell	160
B	Durchführung einer Parameteridentifikation	163
B.1	Parameteridentifikation mit der Methode der höchsten Wahrscheinlichkeit	163
B.1.1	Erstellung einer Wahrscheinlichkeitsverteilung	163
B.1.2	Erstellung einer Likelihood-Funktion	166
B.2	Parameteridentifikation mit der Methode der kleinsten Quadrate	168
C	Implementierung der Gesamtfahrzeugmodellarchitektur	171
C.1	Aufbau der Gesamtfahrzeugmodellarchitektur	171
C.2	Speicherung der Gesamtfahrzeugmodellarchitektur	174
D	Berechnung der Simulationsmodellkomplexität	181
Glossar	185
Abkürzungsverzeichnis	191
Symbolverzeichnis	195
Abbildungsverzeichnis	199
Tabellenverzeichnis	203
Literaturverzeichnis	205
Quellenverzeichnis	205
Betreute Studienarbeiten	214
Publikationen	215

1 Einleitung

„Die Geschichte der Computersimulation beginnt mit dem Wunsch, komplexe Modelle einer wissenschaftlichen Erforschung zugänglich zu machen“ [GRAMELSBERGER 2003, S.7].

John von Neumann entwickelte 1946 in seinem Positionspapier „*On the Principles of Large Scale Computing Machines*“ eine Differenzenmethode für die numerische Simulation partieller Differentialgleichungen. Damit verfolgte er das Ziel, die analytische Behandlung nichtlinearer Probleme mit Hilfe einer numerischen Methode, welche von einem Digitalrechner verarbeitet werden kann, zu überwinden. Diese Von-Neumann-Methode lässt sich als Verfahren beschreiben, das durch Variation der Variablen und Konstanten ein Gleichungssystem mit Hilfe numerischer Integratoren (engl. *Solver*) für spezifische Rand- und Anfangsbedingungen approximativ berechnet [GRAMELSBERGER 2003, S.7]. Damit schuf von Neumann die erste Version einer computergestützten Simulation. In Verbindung mit der bereits 1965 von Gordon Moore in seinem Zeitschriftenaufsatz zum Thema „*Cramming more components onto integrated circuits*“ prognostizierten drastischen Kostenreduktion der hardwareseitig benötigten Schaltkreiskomponenten war der Weg für den Erfolg der digitalen Lösung analytischer Problemstellungen geebnet.

Aufgrund der eingeschränkten Rechenleistung lag der Fokus in den frühen Jahren der Computersimulation auf der Untersuchung des Verhaltens dedizierter Teilsysteme, um diese separat zu optimieren. Mittlerweile ist es durch die gestiegene Rechenleistung möglich, die wechselseitige Einflussnahme heterogener, multiphysikalischer Teilsysteme aus unterschiedlichen Domänen wie Elektronik, Mechanik, Thermodynamik oder Informationstechnik transparent zu machen [LÜDECKE 2003, S.1]. Um diese analysieren zu können, wird durch die Nachbildung eines realen Systems ein Modell erzeugt, welches in der Lage ist, dessen Verhalten in Abhängigkeit von Untersuchungsgegenstand und Modellierungsgrenzen nachzuahmen und

mit Hilfe der Simulation zu prognostizieren [NOLLAU 2009, S.1]. Damit ist es möglich, das statische und dynamische Verhalten komplexer technischer Systeme und seiner Komponenten automatisiert, gefahrlos und wiederholbar zu untersuchen. Dieses Potential der Simulation ermöglicht heute durch die zunehmende Virtualisierung einen Paradigmenwechsel im Produktentstehungsprozess. So ermöglicht eine Virtuelle Entwicklung (VE) durch eine frühzeitige Auslegung auf Basis eines virtuellen Prototyps, also eines Modells eines neuen Produktes, bereits in der Konzeptphase einen höheren Reifegrad (vergleiche linke Hälfte in Abbildung 1.1). Durch diese höhere Produktreife können Entwicklungszeiten verkürzt (vergleiche frühere *Time to Market* und damit früherer *Start of Production* (SoP) in der rechten Hälfte von Abbildung 1.1), die Erprobungsqualität realer Prototypen erhöht und damit sowohl Hardwareschleifen eingespart als auch Untersuchungsmöglichkeiten erweitert werden [SEIFFERT 2008, S.7].

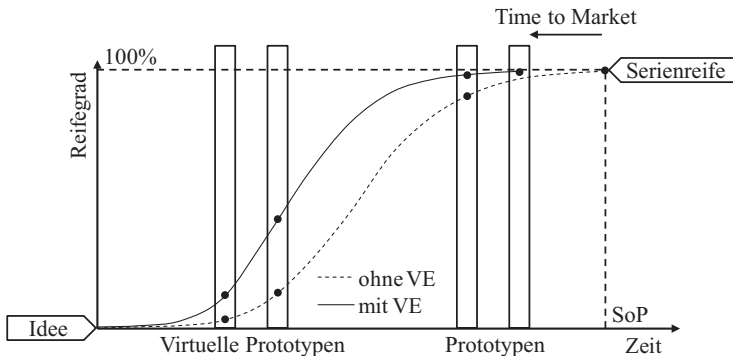


Abb. 1.1: Erhöhung Reifegrad durch virtuelle Entwicklung nach [SEIFFERT 2008, S.8]

Auch in der Kraftfahrzeugtechnik spielt der Einsatz virtueller Prototypen eine immer größere Rolle, um die steigende Anzahl an Produkten und Derivaten in kürzer werdenden Innovationszyklen beherrschen zu können. Als hilfreich hat sich dabei der Einsatz realitätsnaher, computergenerierter Versuchsmodelle erwiesen, deren Aufbaustufen zu definierten Meilensteinen im virtuellen Produktentstehungsprozess die Absicherung von Bauteilgestalt und Produktstruktur gestatten. Diese werden auch als *Digital Mock-Up* (DMU) bezeichnet. In Kombination mit den physikalischen Werkstoffeigenschaften ermöglichen DMUs eine teilweise Substitution der zeit- und kostenintensiven Bauteilprüfung an realen Prototypen.

Wie anhand Abbildung 1.2 ersichtlich, existieren neben der gestaltungsorientierten Modellierung und Analyse weitere Aspekte eines virtuellen Prototyps. Davon gewinnen insbesondere die Modellbildung und Modellanalyse zunehmend an Bedeutung, die den Aspekt der Gestalt um weitere Aspekte wie Verhalten und Funktion erweitern. Eine solche umfassende digitale Produktbeschreibung wird auch als *Functional Mock-Up* (FMU) bezeichnet [IPARRAGUIRRE 2013]. Hintergrund für diesen Trend ist die wachsende Anzahl domänenübergreifender Funktionen, wie die Auslegung einer Betriebsstrategie für Hybridfahrzeuge oder die Applikation neuer Fahrerassistenzsysteme. Dabei gestatten FMUs die Berücksichtigung einer Vielzahl an Einflussfaktoren, die ansonsten sowohl mit physischen Prototypen als auch mit den bisherigen Methoden und Werkzeugen der virtuellen Produktentwicklung kaum mehr zuverlässig abgesichert werden können.

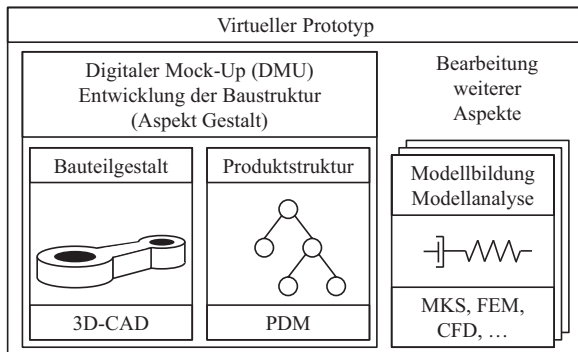


Abb. 1.2: Komponenten eines virtuellen Prototyps nach [SCHÄPPI 2005]

1.1 Problemstellung und Zielsetzung

Mit steigender Anzahl an Einflussfaktoren wachsen Testaufwand und Simulationsdauer, deren Beherrschung mit hohem Personal- und Kostenaufwand verbunden ist. Dementsprechend stellt sich vor dem Hintergrund des jeweiligen Simulationszwecks die Frage, wie viele Einflussfaktoren für eine hohe Ergebnisgüte benötigt werden. Gemäß dem heuristischen Forschungsprinzip der Sparsamkeit, welches nach dessen geistigem Vater, dem Mönch Wilhelm von Ockham, auch als „*Occam's Razor*“ bezeichnet wird, sollte

ein Modell immer so genau wie nötig sein und nicht so genau wie möglich. Damit soll die einfachste und zugleich passendste Erklärung den komplizierteren vorgezogen werden [NOLLAU 2009, S.2]. Um infolgedessen in der virtuellen Entwicklung neben einer hohen Ergebnisgüte und damit einer hohen Effektivität der Simulation trotz steigender Rechenleistung auch eine kurze Simulationsdauer, also eine hohe Effizienz, gewährleisten zu können, muss ein Simulationszweck unter möglichst geringem Ressourceneinsatz erreicht werden können. Allerdings fehlen den Simulationsmodellen die dafür benötigten Quantifizierungsmerkmale und damit die Attribute, die unter Berücksichtigung einer hohen Simulationsergebnisgüte bei gleichzeitig kurzer Simulationsdauer eine Anpassung der Komplexität an einen Simulationszweck gestatten.

Zur Analyse von Entwicklungszeit, -aufwand und -kosten haben Halstead und McCabe bereits Mitte der 70er Jahre Quantifizierungsmerkmale für Softwarekomplexität eingeführt, um daraus Vorhersagen für Aufwände von Test und Wartung abzuleiten (siehe [MCCABE 1976] und [HALSTEAD 1977]). Dabei wird davon ausgegangen, dass sich Softwarekomplexität mit dem Grad an Anwesenheit oder Abwesenheit bestimmter Charakteristiken objektivieren lässt [KOKOL 1996, S.2862]. Der IEEE-Standard 1061 von 1992 definiert die Metriken einer Software als eine Größe, welche die Qualität der einzelnen Programmteile in einem numerischen Wert abbildet. Die wichtigsten Metriken zur Quantifizierung von Programmgröße und Komplexität sind die Zeilenmetriken, die Halstead-Metriken und die McCabe Cyclomatic Number [CULLMANN 2007, S.36]. Durch die Anwendung der Metriken wird das Ziel verfolgt, Softwarekomplexität zu analysieren, zu bewerten und zu vergleichen [STÜRMER 2010, S.5]. Allerdings kann mit diesen traditionellen Softwaremetriken ausschließlich der Softwarecode analysiert werden. Um damit also Simulationsmodelle quantifizieren zu können, müssten diese zunächst kompiliert werden. Da verschiedene Simulatoren, also unterschiedliche Simulationswerkzeuge, mit Compilern unterschiedlicher Qualität arbeiten, würde deren Güte in die Quantifizierung einfließen und das Bewertungsergebnis verfälschen. Im Gegensatz zu den traditionellen Metriken ermöglichen die Verfahren zur Modellselektion (engl. *Model Selection*) eine direkte Bewertung konzeptueller Modelle. Dabei wird versucht, aus einer Vielzahl an konkurrierenden Nachbildungen von Datensätzen die bestmögliche auszuwählen [PITT 2002, S.472]. Field fasst dafür im Rahmen seiner Arbeit die wichtigsten Ansätze zusammen, um aus den Klassen an

Kandidatenmodellen auf Basis verfügbarer Messungen das optimale Modell zu identifizieren (siehe [FIELD 2004]). Allerdings handelt es sich dabei stets um eine situative Bewertung, weswegen für eine Entscheidung, welche Modelle in Abhängigkeit von einem Simulationszweck für welche Teilsysteme am besten geeignet sind, sämtliche Modelle aufwändig überprüft werden müssen. Um eine nachhaltige Quantifizierung zu ermöglichen, stellen Myung und Pitt im Rahmen mehrerer Arbeiten dedizierte Kennzahlen vor (siehe [MYUNG 1999], [PITT 2002] und [MYUNG 2005]). Dabei wird analog zu Field aus einer Klasse von Kandidatenmodellen auf Basis verfügbarer Messungen das jeweils optimale Modell identifiziert. In Ergänzung dazu wird mit Hilfe mathematischer Verfahren die Simulationsmodellkomplexität und Simulationsergebnisgüte bewertet und in definierte Attribute gespeichert, die eine relative Bewertung mehrerer Modelle gestatten. Im Rahmen dieser Arbeit soll das Potential dieser Quantifizierungsmerkmale für eine Komplexitätsadaption technischer Gesamtsysteme erforscht werden.

Um damit die Komplexität von Gesamtsystemsimulationen an einen damit zu erfüllenden Simulationszweck anpassen zu können, sollen die Attribute in einem Prozess zur Entwicklung virtueller Prototypen verortet werden. Grundlage für die virtuelle Entwicklung sind Gesamtsystemarchitekturen, mit deren Hilfe die Systemgrenzen und Schnittstellen sämtlicher Teilsysteme bereits vor der Modellerstellung strukturiert und aufeinander abgestimmt werden können. Damit können die in den einzelnen Teilsystemen enthaltenen Simulationsmodelle an dedizierten Meilensteinen zu virtuellen Prototypen aggregiert werden und dafür deren Quantifizierungsmerkmale berücksichtigt werden. Da dabei zwei oder mehrere Teilsysteme, die untereinander in Beziehung stehen, zu einer gemeinsamen Simulation zusammengefasst werden sollen, handelt es sich um eine sogenannte integrierte Simulation.

Mit dem Standard *AUTomotive Open System ARchitecture* (AUTOSAR) existiert bereits seit 2005 ein Vertreter einer solchen Architekturdefinition. Er stammt aus einer Entwicklungspartnerschaft aus Automobilherstellern, -zulieferern sowie Tool- und Halbleiterlieferanten mit dem Ziel einer standardisierten Softwarearchitektur mit einheitlichen Beschreibungs- und Konfigurationsformaten für Steuergeräte im Automobil [RÜPING 2014, S.1]. Es werden darin Methoden zur Beschreibung von Software im Fahrzeug beschrieben, die sicherstellen, dass Softwarekomponenten ausgetauscht, skaliert, wiederverwendet und integriert werden können. Allerdings werden

bei diesem Standard ausschließlich Anforderungen für Softwarekomponenten der Steuergeräteentwicklung adressiert, wovon FMUs als Bestandteil der virtuellen Entwicklung unberücksichtigt bleiben. Explizit für diese FMUs wurde 2008 das Projekt *Information Technology for European Advancement* (ITEA)² MODELISAR initiiert. Zielsetzung des Projektes ist seitdem die Entwicklung eines applikationsunabhängigen offenen Schnittstellen-Standards, dem *Functional Mock-Up Interface* (FMI), um damit sowohl die Kopplung von Simulatoren als auch den Austausch von Simulationsmodellen im Bereich der Fahrzeugsystemdynamik zu unterstützen [BUSCH 2012, S.3]. Allerdings setzt der Prozess der Standardisierung erst bei den bereits fertigen Simulationsmodellen an, weswegen für eine virtuelle Entwicklung kein umfassender Modularisierungsansatz existiert, der einen koordinierten Aufbau einer Gesamtsystemarchitektur unter Berücksichtigung der Vielfalt an physikalischen und funktionalen Abhängigkeiten vorab definiert. Dementsprechend soll im Rahmen dieser Arbeit neben der Erforschung der Quantifizierungsmerkmale auch ein solcher Ansatz für einen prozesssicheren und beliebig skalierenden Aufbau einer integrierten Gesamtsystemsimulation entwickelt werden. Darin sollen die Quantifizierungsmerkmale eingebettet werden um diese auch in der industriellen Praxis nutzen zu können.

1.2 Struktur der Arbeit

Mit den Grundlagen der Modellbildung und Simulation wird in Kapitel 2 zunächst ein einheitliches Verständnis für die in den nachfolgenden Kapiteln verwendeten Begriffe vermittelt. Kapitel 3 erweitert diese Grundlagen um Aspekte, die für die Modellbildung und Simulation simulatorübergreifender Systeme benötigt werden. Im Rahmen dessen werden verschiedene Modularisierungs- und Kopplungsansätze vorgestellt und der aktuelle Stand der Wissenschaft und Technik beleuchtet. Um die Systemgrenzen und Schnittstellen sämtlicher Teilsysteme aufeinander abzustimmen und damit den Aufbau des FMUs eines virtuellen Prototyps zu koordinieren, werden in Kapitel 4 durch die Strukturierung und Absicherung technischer Gesamtsysteme die dafür benötigten Strukturelemente entwickelt und Maßnahmen zur Absicherung der Struktur vorgestellt. Diese werden am Ende des Kapitels anhand eines Beispiels für ein zu simulierendes FMU demonstriert. Kapitel

5 behandelt dann die mathematischen Grundlagen, die zur Bewertung der Komplexität und damit der Effizienz der in den Strukturelementen gekapselten Simulationsmodelle benötigt werden. Diese werden am Ende des Kapitels exemplarisch auf verschiedene Teilsysteme angewendet. Der Aspekt der Effektivität und damit die Güte der Simulationsergebnisse, sowohl gegebene als auch unbekannte Daten abzubilden, wird in Kapitel 6 auf Basis geeigneter Quantifizierungsmerkmale erfasst und ebenfalls anhand verschiedener Teilsysteme evaluiert. Durch die Verwendung einer Ontologie wird in Kapitel 7 anhand des FMUs eines exemplarischen virtuellen Prototyps die Verknüpfung der Attribute und damit die Anpassung der Simulationsmodellkomplexität sämtlicher Teilsysteme innerhalb einer integrierten Gesamtsystems simulation an einen damit zu erfüllenden Simulationszweck ermöglicht. Das Potential dieser Methode für die ingenieurwissenschaftliche Praxis wird abschließend in der Zusammenfassung mit Ausblick in Kapitel 8 bewertet.

2 Grundlagen der Modellbildung und Simulation

„Modellierung und Simulation haben mit der Einführung der Informationstechnik in den letzten fünfzig Jahren einen Großteil der Ingenieurs- wie auch der Natur-, und Sozialwissenschaften durchdrungen. Egal ob es um den Entwurf der Radaufhängung für ein Automobil, den Metabolismus einer Bakterie oder die Einführung einer neuen Rentenformel geht, immer wieder werden Modelle dieser realen Systeme herangezogen, um die inneren Zusammenhänge eines Systems zu begreifen und Prognosen über dessen Verhalten zu machen“ [PELZ 2001, S.5].

Gemäß diesem einleitenden Zitat aus dem Buch „*Modellierung und Simulation mechatronischer Systeme*“ eröffnet die Nachbildung realer Systeme in den unterschiedlichsten Disziplinen neue Perspektiven der Analyse und Prognose. Schlüssel dafür ist die Abstrahierung des Betrachtungsgegenstands im Rahmen der Modellbildung und Simulation.

Für ein einheitliches Verständnis der in dieser Arbeit verwendeten Begriffe werden zunächst die Grundlagen der Systemtheorie eingeführt, welche die Basis für die nachfolgenden Kapitel bereitstellt. Dabei stützt sich diese Arbeit auf die Definitionen von [DÖRING 2011], [FREY 2008], [KÜBLER 2000] und [PELZ 2001], wissend, dass in der Literatur auch anderslautende Definitionen existieren, die von den hier genannten abweichen können. Darauf aufbauend wird der Prozess der Modellbildung und Simulation technischer Systeme erläutert, innerhalb dessen die theoretischen Grundlagen der Entstehung und der computergestützten Ausführung von Simulationsmodellen vorgestellt werden. Um den Prozess abzuschließen, werden Verfahren zur Parameteridentifikation und Absicherung der in dieser Arbeit eingesetzten Simulationsmodelle erläutert.

2.1 Grundlagen der Systemtheorie

Die Systemtheorie beschäftigt sich mit der mathematischen Beschreibung und Berechnung von mehr oder weniger komplexen Anordnungen aus allen, insbesondere jedoch technischen Bereichen des Lebens. Diese reagieren auf äußere Anregungen oder Einflüsse in einer bestimmten und damit definierten Art und Weise [FREY 2008, S.1]. Dabei ist der Systembegriff ein abstrakter Ausdruck, der z. B. in Soziologie, Politik, Naturwissenschaft und Technik sehr unterschiedliche Ausprägungen annimmt. So dient der Begriff System z. B. in der Soziologie der Beschreibung von Rentensystemen, wohingegen er in der Technik stellvertretend für technische Systeme oder Anlagen steht, die mit mathematischen Gleichungen beschrieben werden können. Dabei repräsentiert ein System S immer ein abstrahiertes Modell der realen Welt, welches in Abbildung 2.1 schematisch dargestellt ist [DÖRING 2011, S.1].

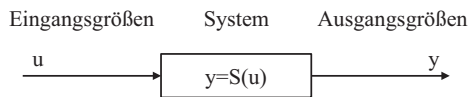


Abb. 2.1: System mit Eingangs- und Ausgangsgrößen nach [DÖRING 2011, S.1]

2.1.1 Definition von Signalen

Die Bezeichnung „Signal“ kommt aus dem Lateinischen und bedeutet „dazu bestimmt, ein Zeichen zu geben“ [DÖRING 2011, S.1]. Vor dem Hintergrund technischer Systeme sind Signale physikalische Größen, bei denen ein oder mehrere Parameter Informationen über eine oder mehrere variable Größen tragen [DKE 2009, S.23]. Im mathematischen Sinn stellen Signale Zeitfunktionen dar und beschreiben damit den zeitlichen Verlauf einer Größe. Gemäß Abbildung 2.1 wird zwischen Eingangssignalen u_i und Ausgangssignalen y_i unterschieden. Signale können anhand ihrer Eigenschaften klassifiziert werden. Eine Übersicht der wichtigsten Signaleigenschaften ist in Abbildung 2.2 dargestellt. Dabei weist ein Signal immer mehrere Merkmale auf [DÖRING 2011, S.2].

Eine wichtige Unterscheidung ist die Einteilung in zeit-kontinuierliche und zeit-diskrete (diskontinuierliche) Signale. Dabei wird ein zeit-kontinuierliches Signal durch eine reelle oder komplexe Funktion mit einer reellen

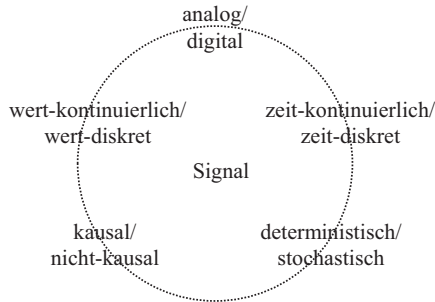


Abb. 2.2: Ausgewählte Signaleigenschaften nach [DÖRING 2011, S.2]

Variable (Veränderlichen) t_k dargestellt (vergleiche Formel 2.1a). Im Gegensatz dazu kann ein zeit-diskretes Signal gemäß Formel 2.1b als eine Folge reeller oder komplexer Zahlen mit einer ganzzahligen Veränderlichen t_d dargestellt werden. Die hier und im Folgenden am Beispiel des Eingangssignals definierten Formeln gelten analog dazu auch für Ausgangssignale [FREY 2008, S.1]:

$$u(t_k) \in \mathbb{R}(\mathbb{C}), \quad t_k \in \mathbb{R} \quad (2.1a)$$

$$u[t_d] \in \mathbb{R}(\mathbb{C}), \quad t_d \in \mathbb{Z} \quad (2.1b)$$

So stellt z. B. ein Sprachsignal und der damit an einem Mikrophon hervorgerufene Spannungsverlauf ein zeit-kontinuierliches Signal dar, während es sich beim Börsenschluss um ein zeit-diskretes Signal handelt. Zeit-kontinuierliche Signale werden mathematisch durch Funktionen, zeit-diskrete Signale durch Folgen beschrieben. Das Sprachsignal ist darüber hinaus auch wert-kontinuierlich, da es zu jedem Zeitpunkt auch einen dazugehörigen reellen Wert gibt. Im Gegensatz dazu werden beim Börsenschluss zu einem dedizierten Zeitpunkt nur eine begrenzte Anzahl an Nachkommastellen gespeichert. Aus diesem Grund kann das Signal nur abzählbare Werte innerhalb eines Wertebereichs annehmen und ist damit wert-diskret [FREY 2008, S.2]. Generell handelt es sich bei physikalischen Signalen weitestgehend um zeit- und wert-kontinuierliche Funktionen von Raum und Zeit. Wegen des Aufwandes ist es nicht möglich (und oft auch nicht sinnvoll), den Wert eines Signals an jedem Raumpunkt zu jedem Zeitpunkt t zu ermitteln. Dementsprechend wird gedanklich ein Netz von Sensoren im

Raum verteilt (engl. *Sensor Fusion*) und deren Messergebnisse nur zu bestimmten Zeitpunkten abgefragt und verarbeitet (engl. *Data Fusion*). Diesen Vorgang zur künstlichen Erzeugung zeit-diskreter Signale aus zeit-kontinuierlichen Signalen nennt man Diskretisieren. Dabei wird das Signal mit der Abtastperiodendauer Δt_s (engl. *Sampling Period*) abgetastet und mit dem Quantisierungsintervall Δu_q (engl. *Quantisation Interval*) quantisiert [RUHM 2008, S.1]. Insgesamt resultieren daraus vier verschiedene Fälle, die in Abbildung 2.3 zusammengefasst dargestellt sind. Davon haben insbesondere die analogen Signale, die sowohl zeit- als auch wert-kontinuierlich sind, sowie die digitalen Signale, die sowohl zeit- als auch wert-diskret sind, praktische Bedeutung [FREY 2008, S.2].

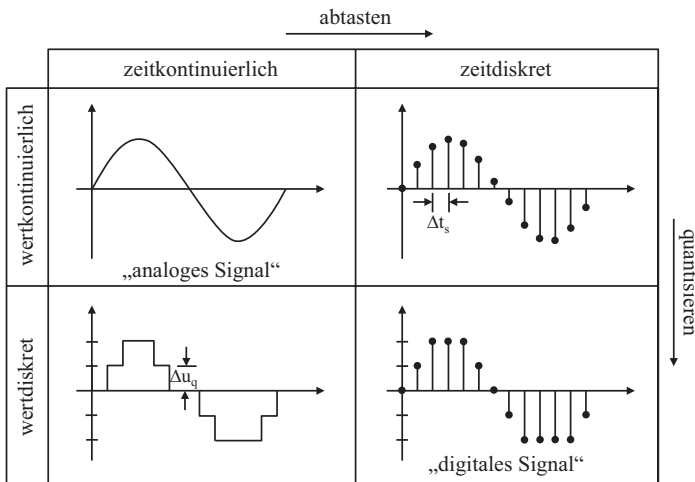


Abb. 2.3: Übersicht kontinuierliche und diskrete Signale nach [FREY 2008, S.2]

Des Weiteren wird zwischen kausalen und nicht-kausalen Signalen unterschieden. Kausale Signale nehmen für $t < 0$ den Wert null und für $t > 0$ Werte ungleich null an [DÖRING 2011, S.1]. Dadurch wird die Beziehung zwischen Ursache und Wirkung festgelegt. Demnach darf eine Änderung des Wertes eines Signals nur für Zeiten $t > 0$ erfolgen.

Ferner existiert gemäß Abbildung 2.2 eine Unterscheidung in deterministische und stochastische Signale. Dabei wird von einem deterministischen Signal gesprochen, wenn der Signalverlauf für alle Zeiten bekannt ist und durch eine mathematische Vorschrift beschrieben werden kann. Bei einem

stochastischen oder zufälligen Signal kann hingegen der exakte Verlauf nicht angegeben werden, sondern nur seine statistischen Signaleigenschaften wie Mittelwert oder Dichtefunktion [FREY 2008, S.3].

2.1.2 Definition von Systemen

Ein System ist eine Menge miteinander in Beziehung stehender Elemente, die in einem bestimmten Zusammenhang als Ganzes angesehen und als von ihrer Umgebung durch eine gedachte Hüllfläche abgegrenzt betrachtet werden können. Ein System wird dabei im Allgemeinen hinsichtlich seiner Zielsetzung, zum Beispiel der Ausführung einer bestimmten Funktion, definiert [DKE 2009, S.11]. Gemäß Abbildung 2.4 bestehen Systeme aus einer Menge an Elementen (Gebilden, Teilen, Komponenten), die in einer bestimmten Beziehung zueinander stehen und zu einem bestimmten Zweck durch interne Funktionszusammenhänge und physikalische Gesetze miteinander interagieren [HILLER 1983, S.1].

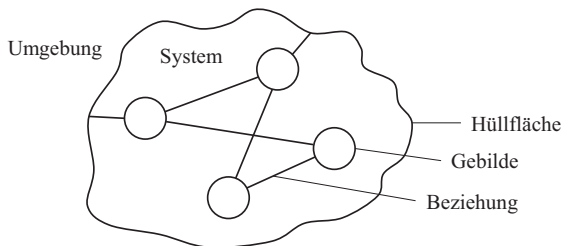


Abb. 2.4: Systembegriff [KÜBLER 2000, S.14]

Systeme, welche mindestens eine solche Interaktionsbeziehung zu einem umgebenden System unterhalten, nennt man offene Systeme. Beziehungen mit Elementen anderer Systeme jenseits der gedachten Hüllfläche legen die Schnittstellen des Systems nach außen hin fest. Diese sind als Grenze zwischen zwei Funktionseinheiten definiert, die durch funktionelle Merkmale, Signalkenngrößen oder andere Merkmale als geeignet festgelegt worden ist [DKE 2009, S.17]. Im Gegensatz zu offenen Systemen interagieren geschlossene Systeme nur mit den Elementen innerhalb der eigenen Hüllfläche und sind damit unabhängig von der Umgebung. Dementsprechend besitzen geschlossene Systeme nach außen hin keine Schnittstellen. Je nach Abstraktionsgrad können die Elemente eines Systems wiederum Systeme sein, bzw.

Systeme können Elemente übergeordneter Systeme sein. Man spricht in diesem Fall von Systemhierarchie [PAGE 1991, S.2]. Um die verschiedenen Ebenen unterscheiden zu können, werden im Folgenden die Elemente eines Systems als „Subsysteme“ oder „Teilsysteme“ bezeichnet.

Das Verhalten eines Systems wird durch die Systemstruktur und durch die Systemparameter definiert. Dabei wird die Systemstruktur durch die Gesamtheit der Beziehungen zwischen den Elementen des Systems festgelegt, wohingegen Systemparameter variable Größen des Systems darstellen, deren Werte das Verhalten des Systems bei gegebener Systemstruktur beschreiben [DKE 2009, S.12]. Neben den Systemparametern sind die Eingangs-, Ausgangs- und Zustandsgrößen weitere Größen zur Beschreibung des Systems. Während die Eingangs- und Ausgangsgrößen über die Systemgrenze hinweg wirken, beschreiben die Zustandsgrößen den inneren Zustand des Systems. Dabei repräsentieren Eingangsgrößen verursachende Größen der Umgebung, die auf das System einwirken, ohne selbst von ihm beeinflusst zu werden [DKE 2009, S.7]. Im Gegensatz dazu sind Ausgangsgrößen messbare Größen, die nur vom System und dessen Eingangsgrößen beeinflusst werden [DKE 2009, S.7]. Zustandsgrößen sind diejenigen zeitveränderlichen Größen des Systems, mit deren Kenntnis zu jedem Zeitpunkt das Verhalten des Systems bei gegebenen Eingangsgrößen eindeutig bestimmt wird [KÜBLER 2000, 14f]. Im Folgenden werden die Eingangsgrößen u_i , Ausgangsgrößen y_i und Zustandsgrößen x_i zu Vektoren zusammengefasst und als „Eingangsvektor u “, „Ausgangsvektor y “ und „Zustandsvektor x “ bezeichnet.

Analog zu Signalen lassen sich auch Systeme, wie in Abbildung 2.5 dargestellt, hinsichtlich ihrer Eigenschaften differenzieren.

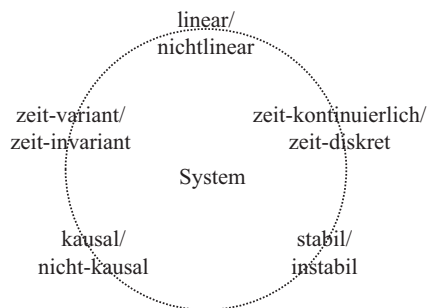


Abb. 2.5: Ausgewählte Systemeigenschaften

Dabei wird auch bei Systemen zwischen zeit-kontinuierlichen und zeit-diskreten Systemen unterschieden. Vor dem Hintergrund des Systemgedankens ist somit ein zeit-kontinuierliches System eine Abbildung, die gemäß Formel 2.2a einem zeit-kontinuierlichen Eingangssignal ein zeit-kontinuierliches Ausgangssignal zuordnet. Demnach bildet, gemäß Formel 2.2b, ein zeit-diskretes System einen zeit-diskreten Eingangsvektor auf einen zeit-diskreten Ausgangsvektor ab. Ein System mit Eingangsvektor u , Ausgangsvektor y und Übertragungsverhalten S wird auch Übertragungsglied oder Block genannt [KÜBLER 2000, S.19]. Für die mathematische Beschreibung verwendet man im Allgemeinen bei zeit-kontinuierlichen Systemen Differentialgleichungen und bei zeit-diskreten Systemen Differenzgleichungen. Diese können mit Hilfe folgender Schreibweisen dargestellt werden [FREY 2008, S.3]:

$$y(t_k) = S \{u(t_k)\} \quad (2.2a)$$

$$y[t_d] = S \{u[t_d]\} \quad (2.2b)$$

Eine weitere elementare Eigenschaft von Systemen ist deren Linearität. Bei linearen Systemen entspricht die Antwort auf eine Linearkombination von Eingangssignalen der dazugehörigen Linearkombination der einzelnen Systemantworten. Dieser Sachverhalt wird durch Formel 2.3 abstrahiert, wobei der Parameter p darin eine beliebige reelle oder komplexe Zahl repräsentiert:

$$S \{p_1 u_1(t) + p_2 u_2(t)\} = p_1 S \{u_1(t)\} + p_2 S \{u_2(t)\}, \quad p_1, p_2 \in \mathbb{R}(\mathbb{C}) \quad (2.3)$$

Dieser Zusammenhang wird auch als „Superpositionsprinzip“ bezeichnet und ist in Abbildung 2.6 graphisch dargestellt.

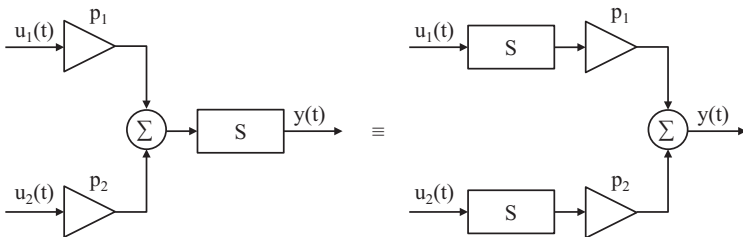


Abb. 2.6: Superpositionsprinzip bei linearen Systemen S nach [FREY 2008, S.5]

Kennt man nun die Antwort zweier Signale eines linearen Systems, kennt man dadurch auch die Antwort aller Linearkombinationen dieser Signale. Umgekehrt lassen sich damit auch komplizierte Signale als Linearkombination einfacher Signale darstellen, die dann einzeln lösbar sind und deren Ergebnisse sich wieder zu den Gesamtergebnissen der komplizierten Signale rekombinieren lassen [FREY 2008, S.4].

Reagiert ein System gemäß Formel 2.4 auf ein verzögertes Eingangssignal mit einem verzögerten Ausgangssignal, so bezeichnet man das System als „zeit-invariant“. Damit ist die Systemantwort nicht von der Zeit abhängig. Es gilt das Verschiebungsprinzip, welches besagt, dass die Gleichungen, die das System beschreiben, und ihre Koeffizienten zeitunabhängig sind [DKE 2009, S.14]:

$$y(t - t_0) = S \{u(t - t_0)\} \quad (2.4)$$

Damit ist die Antwort auf ein Eingangssignal unabhängig vom Anregungszeitpunkt, womit sich die Systemeigenschaften zeitlich nicht ändern.

Systeme, die sowohl linear als auch zeit-invariant sind, werden als „lineare zeit-invariante Systeme“ (engl. *Linear Time-Invariant (LTI) Systems*) bezeichnet und spielen in der Systemtheorie eine wichtige Rolle. Grund dafür ist, dass sowohl die Erstellung als auch die Analyse von Systemen ohne diese beiden Eigenschaften sehr viel komplizierter bzw. überhaupt nicht möglich wären. Aus diesem Grund wird bei der Modellbildung und Simulation technischer Systeme, bei denen diese Voraussetzungen nicht oder nur teilweise zutreffen, durch Einschränkungen oder Näherungen versucht, diese Eigenschaften so weit wie möglich zu erreichen. So kann man bei Nichtlinearitäten wie z. B. bei einem Regelkreis durch eine Taylor-Entwicklung eine Linearisierung um den Arbeitspunkt durchführen. Bei zeitvarianten Systemen wie z. B. dem Mobilfunkkanal kann man Zeiträume suchen, innerhalb derer das System zeit-invariant ist [FREY 2008, S.4f].

Neben diesen zentralen Systemeigenschaften gibt es noch weitere wichtige Eigenschaften eines Systems. So soll es z. B. auch kausal sein, was bedeutet, dass gemäß Formel 2.5 das Ausgangssignal zu einem bestimmten Zeitpunkt t_0 nur von dem Verlauf des Eingangssignals bis einschließlich zu diesem Zeitpunkt abhängen darf:

$$y(t_0) = S \{u(t \leq t_0)\} \quad (2.5)$$

Zwar gehorchen alle System diesem Naturgesetz, allerdings können Idealisierungen in der Systemtheorie zu nicht-kausalen Systemen führen.

Des Weiteren sollen Systeme stabil sein. Demnach darf bei einem beschränkten Eingangssignal die Systemantwort nicht über alle Grenzen anwachsen. Reagiert ein System gemäß Formel 2.6 auf ein beschränktes Eingangssignal mit einem ebenfalls beschränkten Ausgangssignal, wird es als stabil bezeichnet [FREY 2008, S.5]:

$$|u(t)| < \infty \Rightarrow |S\{u(t)\}| < \infty \quad (2.6)$$

2.2 Modellbildung und Simulation

Bei der Modellbildung und Simulation handelt es sich, wie in Abbildung 2.7 dargestellt, um einen Prozess, der auf den in der Systemtheorie getroffenen Definitionen aufsetzt.

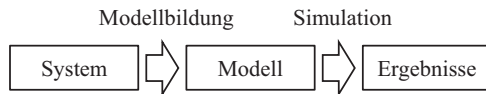


Abb. 2.7: Zusammenhang zwischen Modellbildung und Simulation [ZEITZ 1987]

Dabei wird, ausgehend von einem realen oder hypothetischen System, in einem ersten Schritt durch die Modellbildung ein Modell erstellt, welches die Eigenschaften des Systems bezüglich eines Simulationszwecks hinreichend genau abbildet. Die Simulation ermöglicht es dann im zweiten Schritt, das Experiment nicht am System selbst, sondern an dessen Modell durchzuführen. Im Rahmen dessen ist ein Modell eine mathematische oder physikalische Darstellung eines Systems oder Prozesses, welches das System oder den Prozess aufgrund bekannter Gesetzmäßigkeiten, einer Identifikation oder getroffener Annahmen genügend genau abbildet [DKE 2009, S.17]. Durch die Simulation des Modells werden Daten und Ergebnisse erzeugt, die bei entsprechender Modellgüte Rückschlüsse auf das Verhalten des Originalsystems zulassen [KÜBLER 2000, S.16]. Damit verfolgt die Modellbildung oder auch Modellierung das Ziel, ein reales System zu verstehen und nachzubilden, wohingegen die Simulation basierend auf dem nachgebildeten System versucht, das System zu optimieren und dessen zukünftiges Systemverhalten vorherzusagen [DRONKA 2004, S.14].

Aus der Möglichkeit, Experimente an realen Systemen durch die Simulation virtueller Prototypen zu ersetzen, resultieren in Ergänzung zu den bereits in der Einleitung vorgestellten Potentialen weitere Vorteile für die virtuelle Produktentwicklung technischer Systeme [PELZ 2001, S.5]:

- Zeit- und Kosteneinsparung durch eine automatisierte Analyse virtueller Prototypen
- Zerstörungsfreie und reproduzierbare Untersuchung von Systemverhalten und -zuständen
- Gezielte Parameteridentifikation von Modellen und Transparenz der zu beobachtenden Größen

Diesen Vorteilen stehen auch Nachteile gegenüber:

- Notwendigkeit der Sicherstellung einer vollständigen, validierten und verifizierten Modellbildung
- Begrenzung der Detailtreue durch den Modellbildungsaufwand und die Leistungsfähigkeit der Rechner

2.2.1 Modellbildung

Unter Modellbildung werden diejenigen Arbeiten verstanden, die notwendig sind, um ausgehend von einem realen technischen System eine computergerechte Darstellung des davon abgeleiteten Modells zu erstellen [DRONKA 2004, S.14]. Dabei ist es Ziel der Modellbildung, durch Abstraktion des zu simulierenden Systems (Struktur, Funktion und Verhalten) eine mathematisch-analytische Beschreibung zu erhalten. Diese gestattet es für die daran anschließende Simulation, relevante Aspekte des Systemverhaltens und des Einflusses der Systemkomponenten zu untersuchen, deren Interpretation Rückschlüsse auf das zu simulierende System ermöglicht. Für die Modellbildung lassen sich grundsätzlich zwei Herangehensweisen unterscheiden: Zum einen die theoretische Modellbildung, bei der mathematische Modelle aus physikalischen Gesetzen abgeleitet werden, und zum anderen die experimentelle Modellbildung, bei der ausgehend von einer mathematisch formulierten Modellstruktur parametrisierte Modelle durch die Berücksichtigung gemessener Ein- und Ausgangssignale bestimmt werden [SCHRAMM 2009, S.10].

Theoretische Modellbildung

Bei der theoretischen Modellbildung (White-Box-Modellbildung) wird die Realität als ein System verstanden, das durch eine Simulation nachgebildet wird. Deren Modellbildung lässt sich, wie in Abbildung 2.8 gezeigt, als zweistufiger Prozess darstellen.

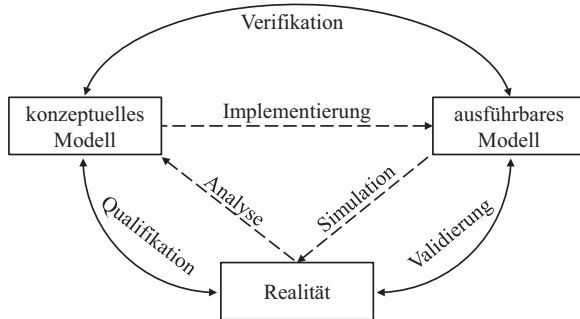


Abb. 2.8: Modellqualifizierung/-validierung und -verifikation nach [PELZ 2001, S.7]

Im ersten Schritt analysiert man die Realität und modelliert sie mit verbalen Beschreibungen, Gleichungen, Relationen oder Naturgesetzen, woraus ein semi-formales konzeptuelles Modell resultiert. Für dieses ist ein Anwendungsbereich zu definieren, für den das Modell die Realität korrekt abbildet [PELZ 2001, S.6]. Für technische Systeme geht man für das zu erstellende konzeptuelle Modell im Allgemeinen von den physikalischen Grundgesetzen aus. Diese sind bei elektrischen Systemen z. B. die Kirchhoffschen Gesetze und bei mechanischen Systemen die Erhaltungssätze für Energie und Impuls [LUNZE 2013, S.58]. Dabei obliegt es dem Entwickler, anhand des Systemaufbaus und des Ziels der Simulation eine sinnvolle Systemgrenze und ein geeignetes Ersatzmodell zu wählen.

Im zweiten Schritt wird das konzeptuelle Modell im Rahmen einer Implementierung durch einen Satz von Anweisungen in eine streng formale, konsistente und widerspruchsfreie Beschreibung des Betrachtungsgegenstands gebracht [BUNGARTZ 2009, S.5]. Daraus resultiert ein ausführbares und damit simulierbares Modell. Die darin definierten Anweisungen beschreiben dabei im Wesentlichen die Antwort des Systems auf von außen eingebrachte Stimuli [PELZ 2001, S.6]. Bei technischen Systemen wird dafür die physikalische in eine mathematische Modellbeschreibung überführt.

Dieser Übergang kann entweder durch den Einsatz eines Modellierungswerkzeugs (automatische Gleichungsgenerierung) oder unter Verwendung der Gesetzmäßigkeiten der jeweiligen Disziplin von Hand durchgeführt werden. Der dafür verwendete Algorithmus wird auch als „Formalismus“ bezeichnet [DRONKA 2004, S.16]. Zur Formalisierung der identifizierten Wechselwirkungen und Abhängigkeiten können gemäß Bungartz [BUNGARTZ 2009, S.8f] diverse Beschreibungsmittel verwendet werden. Dazu zählen unter anderem algebraische Gleichungen oder Ungleichungen, Systeme gewöhnlicher Differentialgleichungen mit nur einer unabhängigen Variablen (engl. *Ordinary Differential Equations* (ODE)) und Systeme partieller Differentialgleichungen mit mehreren unabhängigen Variablen (engl. *Partial Differential Equations* (PDE)). Das häufigste Beschreibungsmittel ist die gewöhnliche Differentialgleichung [LUNZE 2013, S.59]. Diese beschreibt den dynamischen Zusammenhang zwischen der Eingangsgröße $u(t)$ und der Ausgangsgröße $y(t)$ in der Form:

$$\begin{aligned}
 a_r \frac{d^r y(t)}{dt^r} + a_{r-1} \frac{d^{r-1} y(t)}{dt^{r-1}} + \dots + a_1 \frac{dy(t)}{dt} + a_0 y(t) = \\
 b_q \frac{d^q u(t)}{dt^q} + b_{q-1} \frac{d^{q-1} u(t)}{dt^{q-1}} + \dots + b_1 \frac{du(t)}{dt} + b_0 u(t)
 \end{aligned}
 \tag{2.7}$$

Darin sind a_i und b_i reellwertige Koeffizienten, die aus den physikalischen Parametern des betrachteten Systems berechnet werden und q und r die Ordnung der Ableitung von u und y . Der theoretische Modellbildungsprozess zur Erstellung eines physikalischen Ersatzmodells und deren Implementierung in ein ausführbares Modell werden im Anhang in den Kapiteln A.1 und A.2 anhand eines Feder-Masse-Schwingers beschrieben.

Für die computergestützte Verarbeitung wird gegenüber der Darstellung als gewöhnliche Differentialgleichung häufig das Zustandsraummodell mit dem Zustandsvektor x bevorzugt. Um dies zu veranschaulichen werden im Anhang in Kapitel A.3 sowohl die theoretischen Grundlagen zur Erstellung einer Zustandsraumdarstellung aus bereits ermittelten Differentialgleichungen als auch deren Applikation auf den Feder-Masse-Schwinger erläutert.

Experimentelle Modellbildung

Alternativ unterstützt die experimentelle Modellbildung (Black-Box-Modellbildung) die Modellbildung durch Verfahren zur Modellidentifikation.

Diese Herangehensweise wird insbesondere bei unbekanntem oder nur mit erheblichem Aufwand aus physikalischen Gesetzen ableitbaren Prozessen eingesetzt [SCHRAMM 2009, S.10]. Im Gegensatz zur theoretischen Modellbildung spielt dabei die zugrunde liegende Physik und damit auch das Innenleben des Systems nicht notwendigerweise eine Rolle.

Im einfachsten Fall kommen Tabellenmodelle zum Einsatz, bei denen für eine vorgegebene Eingangsgröße die Ausgangsgröße gemessen wird und in einer stufigen oder abschnittsweise linearen Kennlinie gespeichert wird. Problematisch sind dabei Sprung- oder Knickstellen, die aufgrund der endlichen Anzahl an Stützstellen durch alternierendes Anfahren verschiedener Teilstücke zu numerischen Oszillationen führen können und damit die Konvergenz der Simulation beeinträchtigen oder verhindern. Eine Verbesserung kann durch Kennlinienglättung z. B. durch die Chebychev- oder Spline-Approximation erzielt werden [PELZ 2001, S.23].

Ein alternatives Verfahren zur experimentellen Modellbildung ist die Systemidentifikation. Dabei wird in einem ersten Schritt das zu identifizierende System über einen ausreichend langen Zeitraum mit einem geeigneten Testsignal wie Sprungfunktionen, Rechteck- und Dreieckimpulsen angeregt und die Systemantwort gemessen. Mit Prozeduren der Signalvorverarbeitung wie Mittelwertbildung, Effektivwertbildung, aber auch Fourier-, Korrelations- und Spektralanalysen können Störsignalkomponenten beseitigt werden. Im zweiten Schritt wird, basierend auf dem identifizierten Systemverhalten, ein Modell auf Basis vorgefertigter und zugeschnittener Strukturen ausgewählt und deren Parameter entsprechend angepasst. Die Modellstruktur dient dabei als Vehikel, um das beobachtete Verhalten wiederzugeben [PELZ 2001, S.23].

2.2.2 Simulation

Simulation ist das Nachbilden eines Systems mit seinen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind [VDI 2014, S.3]. Ziel der Simulation ist somit, bestimmte Eigenschaften eines tatsächlichen oder auch gedachten technischen, wirtschaftlichen oder biologischen Systems nicht am Original, sondern ersatzweise an einem geeigneten Modell des Originals zu untersuchen. Darüber hinaus können durch eine Simulation für ein betrachtetes Subjekt oder Objekt reale Lastfälle nachempfunden und künstliche Lastfälle

geschaffen werden. Da die Simulation nur mit zeit-diskreten Werten umgehen kann, während die Natur so gut wie nur zeit-kontinuierliche Signale kennt, muss das ausführbare Modell für die numerische Verarbeitung und Analyse diskretisiert werden [RUHM 2008, S.1]. Dabei wird über Diskretisierungsformeln das mit Hilfe von Differentialgleichungen beschriebene zeit-kontinuierliche Modell auf eine zeit-diskrete Beschreibung mit Differenzgleichungen gebracht. Auf diese Weise kann durch wiederholtes Aufrufen der Differenzgleichungen mit fortschreitender Simulationszeit das approximierte Systemverhalten über der Zeit berechnet werden [DRONKA 2004, S.17]. Die Rechtfertigung des Abtastens beruht auf den typischen Eigenschaften zeit-kontinuierlicher Signale, dass sie hohe Redundanz besitzen und sich nicht beliebig schnell ändern. Dementsprechend findet zwischen zwei geschickt gewählten Abtastpunkten (engl. *Sampling Points*) keine oder nur eine unwesentliche Informationsveränderung statt [RUHM 2003, S.2]. Somit ist es möglich, mit einer Folge von zeit-diskreten Zahlen eine zeit-kontinuierliche Funktion exakt zu erfassen, wenn diese Werte der Funktion in hinreichend kleinen Abständen entnommen werden. Dabei werden die Verfahren zur numerischen Differentiation, Integration oder Interpolation angewendet, um unter Verwendung der bekannten Abtastwerte der Eingangsfunktion näherungsweise oder exakt die Zahlenwerte der jeweils gewünschten Ausgangsfunktion als Lösung einer Differenzgleichung zu liefern [SCHÜSSLER 2008, S.2]. Eine Übersicht über die Einteilung der ge-läufigsten Diskretisierungsformeln numerischer Integrationsverfahren ist in Abbildung 2.9 dargestellt. Diese Verfahren eignen sich zur numerischen Lösung von Anfangswertproblemen für Systeme gewöhnlicher Differentialgleichungen (ODEs).

Dabei wird zwischen Einschritt- und Mehrschrittverfahren unterschieden. Bei den Einschrittverfahren wird für die Berechnung des nächsten Wertes jeweils nur der letzte berechnete Wert berücksichtigt. Im Gegensatz dazu verwenden Mehrschrittverfahren die Lösungspunkte mehrerer zurückliegender Zeitpunkte und verfügen damit im Vergleich zu Einschrittverfahren über ein „Gedächtnis“. Integrationsverfahren verwenden unterschiedliche Approximationsmethoden zur Annäherung der gesuchten Lösung und der Berechnung des neuen Wertes durch Extrapolation [DRONKA 2004, S.18].

Da dabei die Ordnung, also die Höhe der Ableitung, der Approximationsmethoden der Ordnung des Verfahrens entspricht, wird neben den Einschritt- und Mehrschrittverfahren auch noch nach der Ordnung des

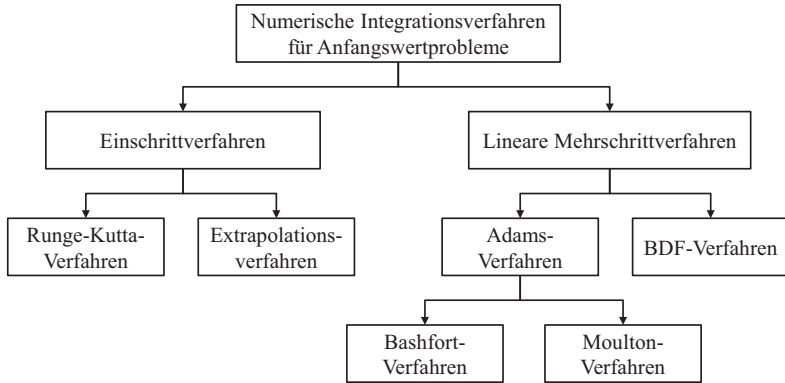


Abb. 2.9: Einteilung von Integrationsverfahren [DRONKA 2004, S.19]

Verfahrens unterschieden. Mit zunehmender Ordnung steigt die Genauigkeit des Verfahrens und der Diskretisierungsfehler sinkt. Die Berechnung des Fehlers kann z. B. auf der Berechnung der Differenz der Ergebnisse zweier Verfahren unterschiedlicher Ordnung basieren. Durch eine Ordnungssteuerung, d. h. die Anpassung der Ordnung an die lokalen Eigenschaften, kann die Genauigkeit der Ausgangsfunktion erhöht werden [DRONKA 2004, S.18].

Bezüglich der Wahl der Abtastpunkte wird zwischen Verfahren mit fester und variabler Schrittweite unterschieden. Bei Verfahren mit fester Schrittweite wird unabhängig vom Ergebnis der letzten Berechnung eine konstante Schrittweite verwendet. Da hierbei keine Fehlerauswertung erfolgt, muss darauf geachtet werden, dass die Schrittweite klein genug gewählt wird, damit die Stabilität der Integration zu jedem Zeitpunkt gewährleistet ist. Im Gegensatz dazu wird bei einem Verfahren mit variabler Schrittweite eine Schrittweitensteuerung verwendet, die für jeden berechneten Zeitschritt analog der Fehlerberechnung bei der Ordnungssteuerung einen Fehler berechnet. Dieser wird mit einer vorgegebenen Genauigkeitsgrenze abgeglichen. In Abhängigkeit davon, ob der Fehler innerhalb des zulässigen Toleranzbereichs liegt, wird das Ergebnis des aktuellen Zeitschritts akzeptiert oder, falls nicht, die Schrittweite verkleinert und die Berechnung wiederholt [DRONKA 2004, S.17].

Ein abschließendes Unterscheidungsmerkmal ist die Unterteilung in explizite (d. h. nicht-iterative) und implizite (d. h. iterative) Verfahren. Bei den expliziten Verfahren werden ausschließlich bereits berechnete Werte für

die Berechnung des neuen Wertes verwendet. Im Gegensatz dazu nutzen implizite Verfahren auch den aktuell zu berechnenden Wert in der Diskretisierungsformel. Da sich damit Gleichungen ergeben, die zu jedem Zeitpunkt iterativ gelöst werden müssen, ist keine explizite Beschreibung für die Berechnung des neuen Wertes möglich. Die Iteration wird bei Unterschreiten einer vorgegebenen Fehlerschranke abgebrochen [DRONKA 2004, S.18].

Problematisch ist für alle Arten von Integrationsverfahren die Behandlung von Unstetigkeitsstellen. Verfahren mit fester Schrittweite können solche Stellen übergehen und durch eine genügend kleine Schrittweite die Stabilität der Integration gewährleisten, während Verfahren mit variabler Schrittweite mit speziellen Unstetigkeitsindikatoren an die Unstetigkeitsstelle heranrechnen und ab dieser die Berechnung neu initialisieren müssen. Dies ist besonders bei Mehrschrittverfahren problematisch, da hier durch das ab dieser Stelle fehlende Gedächtnis eine neue Startrechnung notwendig wird [DRONKA 2004, S.19]. In der Praxis werden fast ausschließlich Einschrittverfahren und lineare Mehrschrittverfahren mit variabler Schrittweite eingesetzt [KÜBLER 2000, S.56]

2.3 Parameteridentifikation

Wie bereits in Kapitel 2.1.2 erläutert, wird das Verhalten eines Systems neben seiner Systemstruktur auch durch seine Systemparameter definiert. Um also die Modellbildung abzuschließen, müssen die Parameter des Modells (wie z. B. die reellwertigen Koeffizienten a_i und b_i der gewöhnlichen Differentialgleichung aus Kapitel 2.2.1) bestimmt werden. Durch das sogenannte Parametrieren werden die beim Parametrisieren definierten Parameter identifiziert, mit deren Hilfe das Simulationsmodell konfiguriert bzw. an beobachtete Daten angepasst werden kann. Einige dieser Parameter, z. B. geometrische Abmessungen wie Länge, Flächen und Volumina, sind messtechnisch sehr einfach erfassbar, wohingegen z. B. die Dichte der Umgebungsluft von verschiedenen Bedingungen abhängt und der jeweils korrekte Wert aus Tabellen bestimmt werden muss. Problematisch wird es z. B. bei Wärmeübergangszahlen, die trotz umfangreicher empirischer Ansätze für unterschiedliche Szenarien stark unterschiedliche Werte liefern können. Für solche freien Parameter ist es sinnvoll, diese über eine Parameteridentifikation zu ermitteln. Dabei muss darauf geachtet werden,

dass die Werte für die freien Parameter innerhalb ihrer zulässigen und in der Regel vorab spezifizierten Grenzen liegen. Auf diese Weise kann vermieden werden, dass das Simulationsergebnis eines Simulationsmodells durch eine unzulässige Parametrierung verbessert wird. Darüber hinaus kann dadurch die Vergleichbarkeit der Bewertungsergebnisse mit ähnlichen bereits parametrisierten Simulationsmodellen gewährleistet werden, da bei allen Simulationsmodellen deren theoretisches Potential für eine hohe Simulationsergebnisgüte voll ausgeschöpft wird. Für die Parameteridentifikation können verschiedene Verfahren angewendet werden, wobei im Folgenden mit der Methode der maximalen Wahrscheinlichkeit und der Methode der kleinsten Quadrate die im Rahmen dieser Arbeit eingesetzten Verfahren erläutert werden.

2.3.1 Methode der höchsten Wahrscheinlichkeit

Bei der Methode der maximalen Wahrscheinlichkeit oder auch Maximum-Likelihood-Methode (engl. *Maximum Likelihood Estimation* (MLE)) handelt es sich um die in vieler Hinsicht allgemeinste Methode der Parameterschätzung [KOLANOSKI 2008, S.78]. Dabei werden diejenigen Parameter θ_i als Schätzung ausgewählt, gemäß deren Verteilung die Repräsentation der beobachteten Daten am plausibelsten erscheint. Im Rahmen dessen stellt der Datenvektor $z = (z_1, \dots, z_n)$ die n -fache Beobachtung (Stichprobe, Realisation) einer als zufällig aufgefassten Prozedur der Größe n dar. Da es sich bei den Ergebnissen eines Zufallsvorgangs nicht zwangsläufig um einen verbal beschreibbaren Sachverhalt handelt, bedient man sich des Konzepts der Zufallsgröße (zufällige Größe, Zufallsvariable), um die Ergebnisse mit Hilfe reeller Zahlen beschreiben zu können [KRAPP 2007, S.51]. Bei einer solchen Zufallsgröße handelt es sich um eine Funktion, mit der jedem Ergebnis ω aus der Ergebnismenge Ω eines Zufallsexperiments eine reelle Zahl $Z(\omega)$ zugeordnet wird [BEWERSDORFF 2011, S.49]. Für die Ausprägung einer Beobachtung werden kleine Buchstaben wie z. B. z verwendet. Besitzt eine Zufallsgröße abzählbar viele mögliche Funktionswerte (z. B. die Augen eines Würfels), wird sie als „wert-diskrete Zufallsgröße“ bezeichnet. Wert-kontinuierliche Zufallsgrößen werden als Grenzwerte wert-diskreter Zufallsgrößen für infinitesimal kleine Intervalle angesehen. Dabei gibt die Wahrscheinlichkeitsverteilung an, mit welchen Wahrscheinlichkeiten eine Zufallsgröße ihre möglichen Werte annimmt. Im Fall einer wert-diskreten Zufallsgröße

wird die Beobachtung von einer dazugehörigen Wahrscheinlichkeitsfunktion (WF) (engl. *Probability Mass Function* (PMF)) beschrieben, die festlegt, mit welcher Wahrscheinlichkeit Z einen bestimmten Wert annimmt. Da für den Grenzfall einer wert-kontinuierlichen Zufallsgröße die Wahrscheinlichkeit, dass die Zufallsgröße einen bestimmten Wert annimmt, gleich null ist, wird in diesem Fall angegeben, mit welcher Wahrscheinlichkeit Z in ein bestimmtes, infinitesimal kleines Intervall fällt. Die daraus resultierende Verteilung wird als „Wahrscheinlichkeitsdichtefunktion (WDF)“ (engl. *Probability Density Function* (PDF)) bezeichnet.

Das Ziel der Datenanalyse ist es, daraus die Werte der k Parameter $\theta = (\theta_1, \dots, \theta_k)$ zu identifizieren, die die Wahrscheinlichkeit, die Beobachtung erzeugt zu haben, maximieren. Damit ist es möglich, Rückschlüsse auf eine unbekannt Population zu treffen. Dabei existiert für jede Wahrscheinlichkeitsverteilung ein festgelegter Wert der Modellparameter. Durch die Änderung der Parameterwerte werden verschiedene Wahrscheinlichkeitsverteilungen erzeugt. Ein Modell wird somit durch die Gesamtheit an Wahrscheinlichkeitsverteilungen definiert, die von den Modellparametern festgelegt werden [MYUNG 2003, S.91].

Die Funktion $f(z|\theta)$ beschreibt eine Familie von Wahrscheinlichkeitsverteilungen. Sie legt im Fall wert-diskreter Zufallsgrößen die Wahrscheinlichkeitsfunktion und im Fall wert-kontinuierlicher Zufallsgrößen die Wahrscheinlichkeitsdichtefunktion fest, nämlich dass der Datenvektor z bei einem gegebenen Parametervektor θ erzeugt wird. Falls einzelne Beobachtungen z_i voneinander statistisch unabhängig sind, kann die WDF bzw. die WF für den Datensatz z durch die Multiplikation einzelner Beobachtungen folgendermaßen ausgedrückt werden [MYUNG 2003, S.91]:

$$f(z_1, \dots, z_n|\theta) = \prod_{i=1}^n f(z_i|\theta) \quad (2.8)$$

Die Erstellung einer WF wird im Anhang B.1.1 für Erfolge in einer Folge von 10 Bernoulli-Versuchen erläutert. Dabei gibt die WF Auskunft darüber, inwieweit bei einem festgelegten Wert der Parameter einzelne Beobachtungen wahrscheinlicher sind als andere.

Bei der im Rahmen der Parameteridentifikation betrachteten Wahl der freien Parameter tritt allerdings das umgekehrte Problem auf. Da hier für ein vorhandenes Simulationsmodell bereits beobachtete Daten vorliegen,

muss unter all den möglichen Wahrscheinlichkeitsverteilungen diejenige Wahrscheinlichkeitsverteilung gefunden werden, die die beobachteten Daten am wahrscheinlichsten erzeugt hat. Zur Lösung dieses Problems kann wie im Folgenden aufgeführt die Likelihood-Funktion $L(\theta)$ herangezogen werden, die die Rollen des Datenvektors z und des Parametervektors θ vertauscht [MYUNG 2003, S.91]:

$$L(\theta|z) = f(z|\theta) = \prod_{i=1}^n f(z_i|\theta) \quad (2.9)$$

Damit repräsentiert $L(\theta|z)$ die Likelihood-Funktion der Parameter θ als Antwort auf die beobachteten Daten z und ist deshalb eine Funktion von θ . Um daraus mit der Methode der maximalen Wahrscheinlichkeit diejenige Likelihood-Funktion zu ermitteln, die die Beobachtung am wahrscheinlichsten erzeugt hat, muss derjenige Parametervektor $\hat{\theta}$, bestehend aus denjenigen Parametervariablen $\hat{\theta}_i$, gefunden werden, der die Likelihood-Funktion maximiert. Dieser wird auch als „MLE Estimate“ bezeichnet und hat die Form $\hat{\theta} = (\hat{\theta}_1, \dots, \hat{\theta}_k)$. Die dadurch maximierte Likelihood-Funktion wird als $L(\hat{\theta})$ notiert [MYUNG 2003, S.93]. In Kapitel B.1.2 im Anhang ist beschrieben, wie aus den vorab im Anhang B.1.1 ermittelten Wahrscheinlichkeitsfunktionen für die 10 Bernoulli-Versuche eine Likelihood-Funktion ermittelt werden kann.

Für eine größere Anzahl an Parametern stößt eine solche Multiplikation einzelner Wahrscheinlichkeitsverteilungen schnell an ihre Grenzen, weswegen die analytische Berechnung von $\hat{\theta}$ bevorzugt wird. Zu diesem Zweck wird von einer differenzierbaren Likelihood-Funktion ausgegangen. Aus numerischen Gründen benutzt man dabei meistens den Logarithmus der Likelihood-Funktion, die sogenannte Log-Likelihood-Funktion $\ln L(\theta)$ [KOLANOSKI 2008, S.78]:

$$\ln L(\theta|z_1, \dots, z_n) = \sum_{i=1}^n \ln f(z_i|\theta) \quad (2.10)$$

Die Maximierungsbedingung einer solchen Log-Likelihood-Funktion lautet für einen Parameter θ (bei kontinuierlichen Parametern) wie folgt [KOLANOSKI 2008, S.78]:

$$\begin{aligned} \frac{\partial \ln L(\hat{\theta}|z)}{\partial \theta} &= 0 \\ \frac{\partial^2 \ln L(\hat{\theta}|z)}{\partial \theta^2} &< 0 \end{aligned} \tag{2.11}$$

Für mehrere unabhängige Parameter kann diese wie folgt formuliert werden [KOLANOSKI 2008, S.78] bzw. [MYUNG 2003, S.93]:

$$\begin{aligned} \frac{\partial \ln L(\hat{\theta}|z)}{\partial \theta_i} &= 0, \quad i \in \{1, \dots, k\} \\ \frac{\partial^2 \ln L(\hat{\theta}|z)}{\partial \theta_i \partial \theta_j} &= H(\hat{\theta}) \text{ negativ definit,} \quad i, j \in \{1, \dots, k\} \end{aligned} \tag{2.12}$$

Die Hesse-Matrix repräsentiert darin ein Analogon zur zweiten Ableitung einer Funktion, mit deren Hilfe die Krümmung einer differenzierbaren Funktion bestimmt wird [LEYDOLD 2000, S.166]. Diese ist für k Parameter wie folgt definiert:

$$H(\hat{\theta}) = \left(\frac{\partial^2 f(\hat{\theta})}{\partial \theta_i \partial \theta_j} \right)_{i,j \in \{1, \dots, k\}} = \begin{pmatrix} \frac{\partial^2 f(\hat{\theta})}{\partial \theta_1 \partial \theta_1} & \frac{\partial^2 f(\hat{\theta})}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 f(\hat{\theta})}{\partial \theta_1 \partial \theta_k} \\ \frac{\partial^2 f(\hat{\theta})}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 f(\hat{\theta})}{\partial \theta_2 \partial \theta_2} & \cdots & \frac{\partial^2 f(\hat{\theta})}{\partial \theta_2 \partial \theta_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\hat{\theta})}{\partial \theta_k \partial \theta_1} & \frac{\partial^2 f(\hat{\theta})}{\partial \theta_k \partial \theta_2} & \cdots & \frac{\partial^2 f(\hat{\theta})}{\partial \theta_k \partial \theta_k} \end{pmatrix} \tag{2.13}$$

Die Matrix ist negativ definit, wenn alle Eigenwerte kleiner null sind [KOLANOSKI 2008, S.78].

2.3.2 Methode der kleinsten Quadrate

Ein weiteres Verfahren zur Identifikation eines optimalen Parameter-Settings für beobachtete Daten ist die Methode der kleinsten Quadrate (engl. *Least Squares Estimation* (LSE)). Dabei handelt es sich um ein mathematisches Standardverfahren zur Ausgleichsrechnung in der Regressionsanalyse. Im Gegensatz zur MLE werden bei der LSE nicht diejenigen Parameterwerte gesucht, die mit höchster Wahrscheinlichkeit die beobachteten Daten erzeugt haben, sondern diejenigen Parameterwerte, die die genaueste Beschreibung der beobachteten Daten ermöglichen. Dabei werden mit

Hilfe der Summe der quadratischen Fehler (engl. *Sum of Squared Errors* (SSE)) über die n Messpunkte die freien Parameter des Modells so lange variiert, bis die Abweichung zwischen den gemessenen Realdaten z_i und Modellausgangsgrößen y_i minimal wird [RE 2009, S.6f]:

$$SSE = \sum_{i=1}^n (z_i - y_i)^2 \quad (2.14)$$

Die lineare, zeit-diskrete Funktion der Modellausgangsgrößen y_i ist dabei eine Funktion der Modelleingangsgrößen u_{ij} und der zu schätzenden freien Parameter θ_j mit i und j als deren Zählindizes:

$$y_i = u_{i1}\theta_1 + \dots + u_{ij}\theta_j, \quad i \in \{1, \dots, n\} \quad j \in \{1, \dots, m\} \quad (2.15)$$

Dieser Zusammenhang zwischen den Modelleingangsgrößen und -ausgangsgrößen kann auch in Form einer Matrix dargestellt werden:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1k} \\ u_{21} & u_{22} & \dots & u_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \dots & u_{nk} \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_k \end{pmatrix} \quad (2.16)$$

Für diese werden zunächst für den Fall einer Modelleingangsgröße $u = (u_1, \dots, u_n)$ diejenigen Parameter $\hat{\theta} = (\hat{\theta}_1, \dots, \hat{\theta}_k)$ gesucht, die die Abweichung zwischen den gemessenen Beobachtungen $z = (z_1, \dots, z_n)$ und den durch die Modelleingangsgrößen und Parameter definierten Modellausgangsgrößen $y = (y_1, \dots, y_n)$ minimieren [RE 2009, S.35]. Zu diesem Zweck muss zunächst für jeden Parameter die auf der SSE basierende Kostenfunktion (Zielfunktion) V wie folgt formuliert werden:

$$V(\hat{\theta}) = \sum_{i=1}^n (z_i - u_i^T \hat{\theta})^2 \quad (2.17)$$

Um dafür das Minimum zu finden, muss die notwendige Bedingung für eine Extremstelle identifiziert und die erste Ableitung gleich null gesetzt werden:

$$\begin{aligned}\frac{\partial V(\hat{\theta})}{\partial \hat{\theta}} &= 2 \sum_{i=1}^n (z_i - u_i^T \hat{\theta}) \frac{\partial}{\partial \hat{\theta}} (z_i - u_i^T \hat{\theta}) \\ &= -2 \sum_{i=1}^n u_i (z_i - u_i^T \hat{\theta}) \\ &= -2 \sum_{i=1}^n (u_i z_i - u_i u_i^T \hat{\theta}) = 0\end{aligned}\tag{2.18}$$

Um gewährleisten zu können, dass es sich dabei um ein Maximum handelt, muss zudem als hinreichende Bedingung erfüllt sein, dass die zweite Ableitung kleiner null ist:

$$\frac{\partial^2 V(\hat{\theta})}{\partial \hat{\theta}^2} = 2 \sum_{i=1}^n u_i u_i^T < 0\tag{2.19}$$

Gleichung 2.18 kann nun nach dem zu schätzenden Parameter $\hat{\theta}$ aufgelöst werden. Wird dabei wieder die in Gleichung 2.16 dargestellte Matrix mit den k zu schätzenden Parametern $\theta = (\theta_1, \dots, \theta_k)$ berücksichtigt, wird aus dem in den Gleichungen 2.17 bis 2.19 betrachteten Vektor für die Modelleingangsgrößen $u = (u_1, \dots, u_n)$ durch Transponieren die Eingangsgrößenmatrix U [RE 2009, S.38]:

$$\begin{aligned}\hat{\theta} &= \left(\sum_{i=1}^n u_i u_i^T \right)^{-1} \sum_{i=1}^n u_i z_i \\ &= (U^T U)^{-1} U^T z\end{aligned}\tag{2.20}$$

Das auf dieser Gleichung basierende Anlernen von Parametern (engl. *Training*) ist im Anhang in Kapitel B.2 am Beispiel eines Chemiewerkes veranschaulicht. Für dieses sind über einen längeren Zeitraum die Eingangs- und Ausgangsgrößen erfasst und die Korrelation zwischen den Werten über eine Gleichung hergestellt worden.

Für den Fall, dass die Messwerte z_i aus einer normal verteilten Wahrscheinlichkeitsverteilung stammen, entspricht die bereits vorgestellte Methode der maximalen Wahrscheinlichkeit der Methode der kleinsten Quadrate (siehe dazu [KOLANOSKI 2008, S.91f]). In einem solchen Fall kann die Likelihood-Funktion auch analytisch mit Hilfe der Methode der kleinsten Quadrate ermittelt werden.

2.4 Modellverifikation und -validierung

Die Verlässlichkeit und damit die Prognosesicherheit einer Simulation hängt entscheidend von den darin eingebetteten Modellen ab. Grund dafür ist, dass es sich bei der Modellbildung um einen fehlerträchtigen Prozess handelt [PELZ 2001, S.6]. Gemäß Großmann [GROSSMANN 2000] kann zwischen folgenden Fehlerarten unterschieden werden:

- Modellfehler
 - Effekte (fehlende, fehlerhafte oder ungenügende Abbildung aller aussagerelevanten physikalisch-technischen Wirkmechanismen)
 - Struktur/Topologie (unzureichende oder fehlerhafte Auflösung in der Abstraktion und Elementarisierung bezüglich der Ziele der Simulation)
 - Parameter (ungenauere oder fehlerhafte Identifikation freier Parameter)
- Berechnungsfehler
 - Lösungsverfahren (explizit/implizit, Einzschritt-/Mehrschritt-Methoden)
 - Lösungsbedingungen (vorgegebene Fehlertoleranzen, Schrittweiten, Anfangsbedingungen)
- Auswertungsfehler
 - Verarbeitung (Filterung, Skalierung, Transformation)
 - Interpretation

Dementsprechend sind Methoden und Werkzeuge gefragt, die in der Lage sind, sowohl konzeptuelle als auch ausführbare Modelle abzusichern. Gemäß Abbildung 2.8 können dafür die Modellverifikation und -validierung verwendet werden.

2.4.1 Modellverifikation

Verifikation bedeutet gemäß VDI-Richtlinie 3633 den formalen Nachweis der Korrektheit des Simulationsmodells, also die Übereinstimmung zwischen Umsetzung und Spezifikation. Im Bereich der Simulation bedeutet

Verifikation beispielsweise, dass das ausführbare Modell das konzeptionelle Modell korrekt umsetzt [RABE 2008, S.14].

Da die Verifikation damit, wie in Abbildung 2.8 bei der Einführung der Modellbildung in Kapitel 2.2.1 dargestellt, die Übereinstimmung zwischen einem konzeptuellen Modell und einem ausführbaren Modell beschreibt, setzt die direkteste Form der Verifikation schon bei der Implementierung des ausführbaren Modells an. Hierbei wird durch modulare bzw. objektorientierte Modellierung, modulares Testen, exakte Definition und laufende Überprüfung des Geltungsbereichs versucht, zu identifizierende Fehler erst gar nicht entstehen zu lassen [PELZ 2001, S.27].

Eine weitere Möglichkeit liefert die Verifikation anhand alternierender Modelle. Falls sich ein Modell über zwei in Methodik und Umsetzung verschiedene Ansätze darstellen lässt, können diese für eine gegenseitige Verifikation herangezogen werden. Dadurch sinkt die Wahrscheinlichkeit eines Fehlers, da es unwahrscheinlich ist, dass verschiedene Fehler innerhalb der Ansätze die gleichen Auswirkungen zeigen [PELZ 2001, S.28].

Eine dritte Möglichkeit der Verifikation bietet die visuelle Inspektion (engl. *Eyeballing* [KLEIJNEN 1995, S.147]). Durch Einsatz eines sogenannten *Debuggers* oder vergleichbarer Werkzeuge ist es möglich, einen sequentiellen Modellcode darzustellen und zu überprüfen. Alternativ können hier auch die Verläufe der Simulationsergebnisse für die Beurteilung verwendet werden [PELZ 2001, S.29].

2.4.2 Modellvalidierung

Die VDI-Richtlinie 3633 beschreibt als Ziel der Validierung, das Verhalten des realen Systems mit Hilfe eines Modells genau genug und fehlerfrei widerzuspiegeln. Validierung versucht zu klären, ob die Übereinstimmung von Modell und realen System hinreichend genau ist. Die geforderte Genauigkeit der Übereinstimmung hängt dabei vom jeweiligen Anwendungsfall des Modells ab [RABE 2008, S.15].

Generell kann dabei der Ansatz der Validierung anhand der Modellhierarchie verfolgt werden. Falls sich Teilmodelle im Gesamtmodell wiederholen, werden die Teilmodelle losgelöst validiert, woraus auf die Validität der übrigen, identischen Komponenten geschlossen wird [PELZ 2001, S.34f].

Da gemäß Abbildung 2.8 aus der Einführung der Modellbildung in Kapitel 2.2.1 mit der Validierung die Korrespondenz zwischen dem ausführbaren

Modell und der Realität gewährleistet werden soll, eignet sich neben einem direkten visuellen Vergleich die Quantifizierung auf Basis von Messdaten, die aus Messungen am realen System gewonnen werden können. Gemäß Murray-Smith [MURRAY-SMITH 1995, S.77] lassen sich dafür diverse Funktionen definieren, die an den Stützstellen ein Maß für die Übereinstimmung zwischen der gemessenen Antwort z_i und dem Ergebnis der Simulation y_i darstellen.

Eine weitere Möglichkeit bietet die Validierung durch den Model-Distorsion-Ansatz. Dabei wird für jeden Zeitpunkt die „Verzerrung“ von Parametern berechnet, die notwendig sind, um zu exakt mit den Messungen übereinstimmenden Simulationsergebnissen zu gelangen. Die daraus resultierende Spanne an sich von Zeitschritt zu Zeitschritt ändernden Nominalparametern ist dann ein Maß für die Güte des Modells [PELZ 2001, S.33].

Eine dritte Möglichkeit ist die Validierung anhand einer Empfindlichkeitsanalyse. Da Parameter linear in das Modellverhalten eingehen, ist der Wert eines Modells fraglich, wenn die Variation eines Parameters innerhalb eines Intervalls zu erheblichen Variationen der Simulationsergebnisse führt. Zusätzliche Indizien können mit der Validierung anhand einer Monte-Carlo-Simulation oder auch Risikoanalyse gefunden werden. Im Gegensatz zur Empfindlichkeitsanalyse werden hier nicht nur die Auswirkungen eines Parameters, sondern, aufgrund der Abhängigkeit der Simulationsergebnisse von mehreren Parametern, auch die Auswirkungen einer Kombination von statistisch streuenden Parametern geprüft. Zu diesem Zweck werden Intervalle und statistische Verteilungen für „verdächtige Parameter“ vorgegeben und eine größere Anzahl an Simulationen durchgeführt [PELZ 2001, S.33f].

3 Modellbildung und Simulation simulatorübergreifender Systeme

„Der steigende Komplexitätsgrad technischer Systeme, bedingt durch die zunehmende Verflechtung unterschiedlicher Fachdisziplinen, macht es erforderlich, die in domainspezifischen Simulationsprogrammen erstellten Teilmodelle einer gemeinsamen Simulation zugänglich zu machen, um so das Verhalten eines Gesamtsystems analysieren zu können“ [GEIMER 2006, S.572].

Mit den Grundlagen der Modellbildung und Simulation aus Kapitel 2 wurde die Basis für das weitere Vorgehen erarbeitet. Sie werden im Rahmen dieses Kapitels um Möglichkeiten zur Modularisierung und Kopplung simulatorübergreifender Gesamtsysteme erweitert.

Hintergrund dafür ist, gemäß dem einleitenden Zitat aus einem Zeitschriftenaufsatz zum Thema „*Co-Simulation, gekoppelte Simulation oder Simulatorkopplung?*“ die zunehmende Verflechtung unterschiedlicher Domänen. Diese macht einen simulationswerkzeugunabhängigen Aufbau integrierter Gesamtsystemsimulationen erforderlich. Die Kopplung solcher simulatorübergreifender Systeme bietet die Möglichkeit, über die Betrachtung von Einzelkomponenten hinausgehend, simulationswerkzeugübergreifend Wechselwirkungen zwischen Teilsystemen abzubilden. Damit wird der Aufbau von FMUs unterstützt und gleichzeitig das dafür benötigte Verfahren einem breiten Anwendungsbereich zur Verfügung gestellt. Mit einem auf diese Weise erstellten FMU können Funktionen abgeprüft und das Verhalten des Gesamtsystems prognostiziert und optimiert werden [KOSSEL 2010, S.103].

Um dieses Ziel zu erreichen, werden in diesem Kapitel zunächst die möglichen Kopplungsvarianten simulatorübergreifender Systeme eingeführt. Darauf aufbauend werden die Methoden zur Modularisierung und Kopplung

vorge stellt. Im Rahmen dessen wird auch die Entscheidung für das in dieser Arbeit relevante Verfahren herausgearbeitet. Abschließend wird der aktuelle Stand der Wissenschaft und Technik beleuchtet, wobei in diesem Zusammenhang auch der Prozess eingeführt wird, auf den der später vorgestellte Modularisierungsansatz aufsetzt.

3.1 Kopplungsvarianten simulatorübergreifender Systeme

Für die Modellbildung und Simulation simulatorübergreifender Systeme sind durch die zunehmende Komplexität der zu simulierenden Problemstellungen in den letzten Jahren verschiedene Entwicklungstendenzen sichtbar geworden. Dazu zählen die Erweiterung ehemals spezialisierter Simulatoren um Modellbibliotheken zur Systemsimulation, die Realisierung von Schnittstellen zur Datenübergabe an Werkzeuge anderer Bereiche und die Entwicklung sogenannter *Multi-Domain*-Werkzeuge, die einen universellen mathematischen Ansatz der Modellierung in verschiedenen Domänen zulassen [DRONKA 2004, S.28]. Die daraus resultierenden Ansätze bei getrennter Betrachtung von Modellbildung und Simulation unter Einsatz eines bzw. mehrerer Werkzeuge sind in Abbildung 3.1 dargestellt. Auf der Abszisse ist darin die Anzahl der Integratoren und auf der Ordinate die Anzahl der Modellierungswerkzeuge dargestellt.

3.1.1 Geschlossene Modellbildung

Bei der geschlossenen Modellbildung (siehe untere Hälfte von Abbildung 3.1) wird genau ein Modellierungswerkzeug verwendet, in welchem das Gesamtsystem abgebildet wird. Infolgedessen ist eine Verwendung domänenspezifischer Werkzeuge für eine simulatorübergreifende Modellbildung nicht möglich, da sowohl die Modellbildungsumgebung als auch die Simulations- und Analysefunktionalitäten an die jeweilige physikalische Disziplin angepasst sind [DRONKA 2004, S.29]. Es wird unterschieden, ob nur ein Integrator (geschlossene Simulation) oder mehrere Integratoren (verteilte Simulation) zum Einsatz kommen [GEIMER 2006, S.572]. Im einfachsten Fall, der „klassischen“ Simulation, in Bild 3.1 als III gekennzeichnet, finden Modellbildung und Simulation innerhalb eines Modellierungswerkzeugs statt.

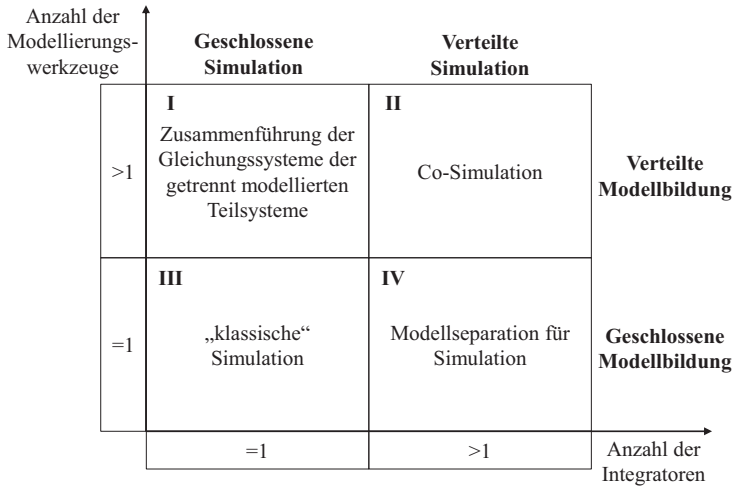


Abb. 3.1: Kopplungsvarianten nach [GEIMER 2006, S.572]

Im Gegensatz dazu wird bei der „Modellseparation für Simulation“ aus Variante IV das Modell zwar ebenfalls mit nur einem Modellierungswerkzeug erstellt, aber verteilt simuliert. Zu diesem Zweck wird es in Teilsysteme zerlegt und mit mehreren Integratoren gelöst. Auf diese Weise kann für verschiedene Teilsysteme der jeweils optimale Solver eingesetzt werden und damit die Lösungsgenauigkeit erhöht und wegen der verteilten Simulation Rechenzeit eingespart werden.

Praktische Bedeutung hat davon insbesondere der Ansatz der multidisziplinären Sprache, der im Wesentlichen dem Ansatz der „klassischen“ Simulation analog Variante III aus Abbildung 3.1 entspricht. Dabei wird das System komplett innerhalb einer Simulationsumgebung in einer multidisziplinären Modellierungssprache wie z. B. Modelica oder *Advanced Continuous Simulation Language (ACSL)* beschrieben. Diese Sprachen basieren darauf, dass Systeme verschiedener physikalischer Domänen in einer einheitlichen, abstrakten Form dargestellt werden können [BUSCH 2012, S.1]. Eine solche Implementierung ist in Abbildung 3.2 exemplarisch anhand der Definition eines Kondensators in der Programmiersprache Modelica dargestellt.

Darin wird zunächst im oberen Drittel von Abbildung 3.2 eine Steckverbindung definiert, diese im mittleren Drittel auf zwei Steckverbindungen

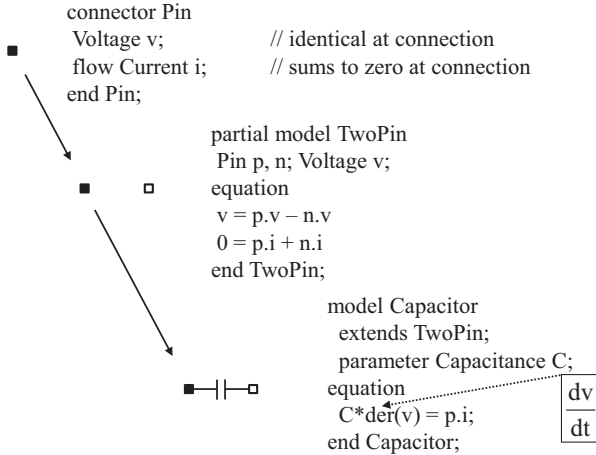


Abb. 3.2: Exemplarische Definition eines Kondensators in Modelica [OTTER 2009]

erweitert und im unteren Drittel um die Kapazität ergänzt. Dieses elektrische Teilsystem kann dann innerhalb der gleichen Modellierungssprache um weitere Teilsysteme anderer physikalischer Domänen erweitert werden. Anschließend werden, wie in Kapitel 2 erläutert, die Gleichungssysteme für die multidisziplinären Teilsysteme erarbeitet, die Abhängigkeiten zwischen den Eingangs- und Ausgangsgrößen identifiziert und in eine geeignete Form für die numerische Lösung, wie die Zustandsraumdarstellung, gebracht [VALASEK 2009, S.87]. Die daraus resultierenden Gleichungen des multidisziplinären Gesamtsystems können dann automatisiert erstellt und von einem optimierten Integrator gelöst werden. Dabei ist der Prozess der Erstellung eines solchen Gesamtsystems verhältnismäßig einfach. Allerdings kann ein automatisiert erstelltes Gleichungssystem komplex und unstrukturiert sein, so dass eine aufwändige algebraische Transformation notwendig ist, um eine stabile numerische Simulation gewährleisten zu können [BUSCH 2012, S.2].

3.1.2 Verteilte Modellbildung

Bei der verteilten Modellbildung (siehe obere Hälfte von Abbildung 3.1) kommen zwei oder mehrere Modellierungswerkzeuge zum Einsatz. Somit ist für eine Analyse des Gesamtsystems in jedem Fall die Kopplung der dabei

entstehenden Teilsysteme erforderlich [DRONKA 2004, S.29]. Dabei wird bei der „Zusammenführung der Gleichungssysteme der getrennt modellierten Teilsysteme“ in Variante I aus Abbildung 3.1, im Gegensatz zu der bereits beschriebenen Modellseparation für die Simulation aus Variante IV, das Modell verteilt modelliert, aber geschlossen simuliert. Es handelt sich somit um den inversen Fall, bei dem die getrennt modellierten Teilsysteme oder deren Gleichungssysteme für die Simulation zusammengefasst und mittels eines gemeinsamen Integrators gelöst werden. Die Einbindung „fremder“ Systemanteile erfolgt dabei in der Regel als Quellcode über die zur Verfügung stehenden Programmierschnittstellen [DRONKA 2004, S.29]. Im Gegensatz dazu werden bei der verteilten Simulation die verteilt modellierten Teilsysteme mit verschiedenen Integratoren berechnet. Dabei handelt es sich dann um eine „Co-Simulation“ (engl. *Co-Simulation*) [BUSCH 2010, S.2]. Diese wird auch als „gekoppelte Simulation“ (engl. *Coupled Simulation*) [GEIMER 2006, S.572] bzw. „Simulatorkopplung“ (engl. *Simulator Coupling*) [KÜBLER 2000, S.93] bezeichnet und ist in Abbildung 3.1 durch die Variante II repräsentiert [GEIMER 2006, S.572].

Praktische Bedeutung hat heute insbesondere der Ansatz der Co-Simulation analog Variante II aus Abbildung 3.1. Dabei wird das simulatorübergreifende System zunächst, wie in Abbildung 3.3 dargestellt, in separate Teilsysteme zerlegt.

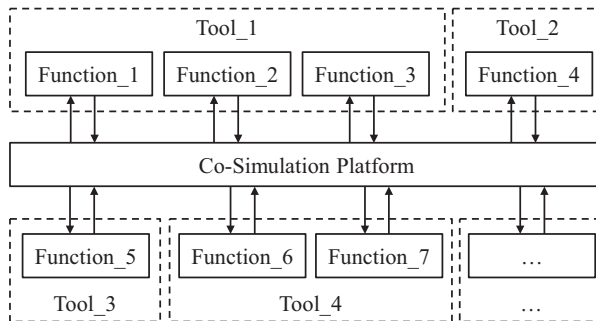


Abb. 3.3: Prinzipskizze Co-Simulation [NETTER 2013b, S.2]

Jedes Teilsystem, welches in der Darstellung als Funktion (engl. *Function*) bezeichnet wird, kann dann in dem jeweils dafür vorgesehenen Simulationswerkzeug (engl. *Tool*) modelliert und, falls gewünscht, im gleichen

Simulationswerkzeug simuliert werden. Die Kopplung dieser Teilsysteme erfolgt dabei über die Systemeingangs- und Ausgangsgrößen mit Hilfe einer neutralen Kopplungssoftware (engl. *Middleware*), die in Abbildung 3.3 als Co-Simulationsplattform (engl. *Co-Simulation Platform*) dargestellt ist. Diese synchronisiert die Zeitschrittweiten für den Informationsaustausch zwischen den in den Simulatoren eingebundenen Modellen. Für die Interaktion der auf teilweise unterschiedlichen Programmiersprachen basierenden Simulationswerkzeuge und den Eingangs- und Ausgangsgrößen der darin eingebundenen Modelle sind entsprechende Schnittstellen erforderlich, die in Darstellung 3.3 schematisch als Pfeile abgebildet sind. Der Vorteil einer auf diese Weise gekoppelten Simulation ist, dass diese von den zumeist einfachen Gleichungen und den ausgereiften und angepassten Integratoren innerhalb der Simulationswerkzeuge profitiert. Allerdings kann bei dem Ansatz der gekoppelten Simulation bereits die Erstellung solcher Schnittstellen für die Interaktion zwischen unterschiedlichen Simulatoren wegen deren Nicht-Uniformität (z. B. durch unterschiedliche Benutzeroberflächen oder verschiedene Programmiersprachen) sehr anspruchsvoll sein [BUSCH 2012, S.1f].

Für die virtuelle Entwicklung innerhalb eines Industrieunternehmens spielt es eine entscheidende Rolle, dass die für die verschiedenen Anwendungsgebiete spezialisierten und optimierten Simulatoren auch für die Modellbildung und Simulation virtueller Prototypen weiterhin genutzt werden können [STÜBER 2011, S.76]. Da die im Rahmen dieser Arbeit erforschten Verfahren, wie in der Problemstellung und Zielsetzung aus Kapitel 1.1 gefordert, auch in der industriellen Praxis Anwendung finden sollen, wird hier ausschließlich der Ansatz der verteilten Modellbildung weiter verfolgt.

3.2 Modularisierung und Kopplung simulatorübergreifender Systeme

Für die Modellbildung und Simulation simulatorübergreifender System kann es erforderlich sein, das zu modellierende Gesamtsystem in mehrere Teilsysteme zu untergliedern. Grund hierfür kann zum einen sein, dass die Teilsysteme zu unterschiedlichen Funktionsbereichen gehören und durch eine verteilte Modellbildung mit unterschiedlichen Werkzeugen dargestellt

werden sollen. Zum anderen können durch eine verteilte Simulation Teilsysteme mit unterschiedlichen Methoden bzw. Schrittweiten berechnet werden oder eine Parallelisierung der Berechnung für den Einsatz auf einem Mehrprozessorsystem erreicht werden [DRONKA 2004, S.15].

3.2.1 Methoden zur Modularisierung bei verteilter Modellbildung

Für eine verteilte Modellbildung wird das Gesamtsystem zunächst in koppelbare Teilsysteme zerlegt. Die dafür verwendeten Techniken unterscheiden sich primär in der Wahl der Koppelgrößen. In der Mechanik kommen das sogenannte *Applied-Force Coupling* und das *Constraint Coupling* zum Einsatz. Beim *Applied-Force Coupling* erfolgt die Kopplung der Teilsysteme über eingeprägte Kräfte wie z. B. Feder- oder Dämpferkräfte, wohingegen beim *Constraint Coupling* die Kopplung der Teilsysteme über Reaktionskräfte wie z. B. Kontaktbedingungen oder algebraische Nebenbedingungen erfolgt. Das *Constraint Coupling* kann auch als Grenzfall des *Applied-Force Coupling* angesehen werden, für den Fall, dass unendlich steife Federn verwendet werden. Der Ansatz des *Applied-Force Coupling* kann weiterhin in Kraft-Weg-Kopplung (engl. *Force/Displacement Coupling*) und Weg-Weg-Kopplung (engl. *Displacement/Displacement Coupling*) unterteilt werden. Bei der Kraft-Weg-Kopplung liefert eines der Subsysteme kinematische Koppelgrößen (Positionen und Geschwindigkeiten), während das andere Subsystem eine Kraft bereitstellt. Bei der Weg-Weg-Kopplung liefern beide Subsysteme jeweils kinematische Größen. Die hier beschriebenen Kopplungsarten lassen sich auch auf andere physikalische Fachdisziplinen übertragen. So entspricht die Kraft-Weg-Kopplung z. B. in der Schaltkreistheorie der Kopplung über Spannung und Strom [BUSCH 2012, S.4].

Zur Modularisierung des Gesamtsystems wird von einem simulatorübergreifenden Gesamtsystem, bestehend aus zwei Teilsystemen mit jeweils einer Ein- und Ausgangsgröße, ausgegangen. Durch die Verwendung der Systemeingangs- und Ausgangsgrößen u_1 , u_2 , y_1 und y_2 als Koppelgrößen, kann das Gesamtsystem analog Abbildung 3.4 in zwei separate Teilsysteme zerlegt werden.

In der dort dargestellten Kopplungsschleife hängen die beiden nicht-linearen Vektorfunktionen f_i und g_i von den Zustandsvektoren x_i , den Eingangsvektoren der Subsysteme u_i und der Zeit t ab [BUSCH 2012, S.4].

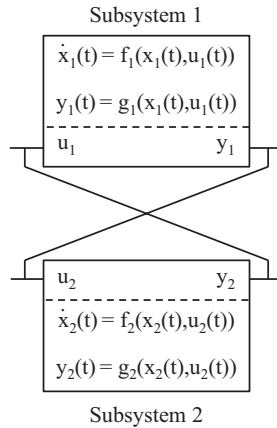


Abb. 3.4: Kopplungsschleife nach [BUSCH 2012, S.5]

Diese Zusammenhänge können gemäß der Gleichung A.14 aus Anhang A.3 durch ein Gleichungssystem erster Ordnung in der expliziten nichtlinearen Zustandsraumdarstellung folgendermaßen beschrieben werden:

$$\dot{x}_1(t) = f_1(x_1(t), u_1(t)) \qquad \dot{x}_2(t) = f_2(x_2(t), u_2(t)) \quad (3.1a)$$

$$y_1(t) = g_1(x_1(t), u_1(t)) \qquad y_2(t) = g_2(x_2(t), u_2(t)) \quad (3.1b)$$

Mit den reellen Matrizen L_{ij} gelten zwischen den Koppelgrößen, also den Systemeingangs- und Ausgangssignalen, folgende Kopplungsbedingungen:

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad (3.2)$$

Häufig sind $L_{11} = L_{22} = 0$ Null- und $L_{12} = L_{21} = I$ Einheitsmatrizen.

Bei der hier dargestellten Kopplung zweier Subsysteme handelt es sich um einen Spezialfall, welcher aber bereits wesentliche Aspekte der Kopplung einer beliebigen Anzahl an Subsystemen erfasst [BUSCH 2012, S.4].

3.2.2 Methoden zur Kopplung bei verteilter Modellbildung

Um die Teilsysteme des verteilt modellierten Gesamtsystems simulieren zu können, müssen diese gekoppelt werden. Anhand der Anzahl der für

die Kopplung verwendeten Integratoren wird zwischen stark und schwach gekoppelten Teilsystemen unterschieden. Kopplungstechniken, die nur einen Integrator verwenden, werden als „stark gekoppelte Systeme“ (engl. *Strong Coupling Approach*) bezeichnet. Bei schwach gekoppelten Systemen (engl. *Weak Coupling Approach*) kommen mehrere Integratoren zum Einsatz. Die Schnittstellenansätze beider Varianten sind in Abbildung 3.5 dargestellt.

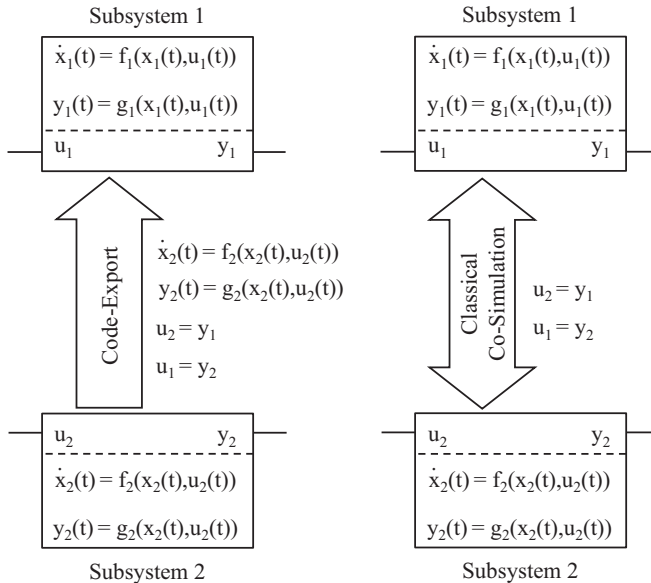


Abb. 3.5: Schnittstellenansätze für verschiedene Kopplungen nach [BUSCH 2012, S.6]

Bei stark gekoppelten Systemen wird nur ein Teilsystem-Integrator für die gekoppelte Simulation verwendet. Dementsprechend werden die Gleichungen von Subsystem 2, analog Variante I aus Abbildung 3.1, über einen Code-Export extrahiert und in das Subsystem 1 eingebettet (vergleiche linke Hälfte in Abbildung 3.5). Anschließend erfolgt die Lösung des auf diese Weise gekoppelten Systems durch den Integrator von Subsystem 1. Die Kopplungsbedingungen aus Formel 3.2 werden dabei zu jedem Integrationszeitpunkt erfüllt. Die Stabilität der starken Kopplung wird hauptsächlich durch die Stabilitätseigenschaften des Integrators bestimmt. Da in den Simulationswerkzeugen meist stabile Integratoren verwendet werden, führt die starke Kopplung zu einer stabilen Form der gekoppelten Simulation.

Werden damit allerdings steife mit weniger steifen Subsystemen gekoppelt, also Systeme, bei denen die Simulationsschrittweite durch die Stabilitätsanforderungen und nicht durch die Genauigkeitsanforderungen bestimmt wird, mit Systemen, bei denen die Genauigkeitsanforderungen und nicht die Stabilitätsanforderungen die Zeitschrittweite definiert, so bestimmt das steifere Subsystem die Schrittweite des gemeinsamen Integrators. Dementsprechend wird das weniger steife Subsystem häufiger ausgewertet als zur Abbildung der Dynamik notwendig wäre. Diese Ineffizienz starker Kopplungen wird durch die Kopplung verhältnismäßig großer und steifer Subsysteme mit kleinen und weniger steifen Subsystemen verstärkt [BUSCH 2012, S.6].

Da bei schwach gekoppelten Systemen analog Variante II aus Abbildung 3.1 beide Teilsystem-Integratoren für die Lösung des gekoppelten Systems verwendet werden, wird die Kopplungsschleife aus Abbildung 3.4 gedanklich aufgetrennt. Damit können beide Subsysteme unabhängig voneinander gelöst werden (vergleiche rechte Hälfte in Abbildung 3.5). Während der Integration müssen die Subsysteme zu zeit-diskreten Makrozeitpunkten (Kommunikationszeitpunkten) Informationen austauschen. Als Makrozeitgitter werden in aller Regel äquidistante Zeitgitter verwendet. Dementsprechend müssen ausschließlich zu diesen Zeitpunkten die in Gleichung 3.2 definierten Kopplungsbedingungen erfüllt sein. Zwischen diesen Zeitpunkten sind die Kopplungsbedingungen im Allgemeinen nicht erfüllt und anstelle der tatsächlichen Koppelgrößen werden approximierte Größen für die Integration der Subsystemgleichungen verwendet. Die dabei anfallenden Zwischenergebnisse werden nicht kommuniziert. Schwach gekoppelte Subsysteme können jeweils einen eigenen problemspezifischen Integrator verwenden. Das kann z. B. bei unterschiedlich steifen Problemen vorteilhaft sein, da jedes Subsystem mit einer angepassten Zeitschrittweite integriert werden kann und damit die Effizienz der Kopplung im Vergleich zur starken Kopplung verbessert wird. Da bei der Co-Simulation die Subsysteme unabhängig voneinander gelöst werden, eröffnet die Variante der schwachen Kopplung darüber hinaus die Möglichkeit des geteilten Speichers (engl. *Shared Memory*) und damit der Parallelisierung auf einem Rechner oder einer verteilten Simulation auf mehreren Rechnern. Üblicherweise erfolgt die Kommunikation über TCP/IP-Netzwerkprotokolle [BUSCH 2012, S.7].

Wie in der Problemstellung und Zielsetzung aus Kapitel 1.1 gefordert, spielt im Rahmen dieser Arbeit die Effizienz der Simulation und damit auch der Kopplung eine entscheidende Rolle. Dementsprechend wird im

Folgenden ausschließlich die Kopplungstechnik der schwachen Kopplung weiter verfolgt. Dessen Schnittstellenansatz ist in der rechten Hälfte von Abbildung 3.5 dargestellt und entspricht der Variante II der gekoppelten Simulation oder auch Co-Simulation aus Abbildung 3.1.

3.3 Erstellung simulatorübergreifender integrierter Gesamtsystemsimulationen

Für die Erstellung simulatorübergreifender integrierter Gesamtsystemsimmulationen muss der klassische Prozess der Modellbildung und Simulation aus Kapitel 2.2 um Aspekte zur Modularisierung und Kopplung erweitert werden. Für die Variante der Co-Simulation wird im Folgenden zunächst der bereits verfügbare Stand der Wissenschaft und Technik beleuchtet. Anschließend wird ein auf der Co-Simulation basierender Prozess für eine virtuelle Entwicklung vorgestellt, auf den der später vorgestellte Modularisierungsansatz aufbaut.

3.3.1 Anwendungen basierend auf Co-Simulation

Die Co-Simulation ist bereits seit den 1980er/1990er Jahren eine etablierte Simulationstechnik im Bereich der Strömungs-Strukturkopplung. In diesem Zusammenhang ist auch die für die simulatorübergreifende Kopplung standardisierte Schnittstelle *Mesh-based parallel Code Coupling Interface* (MpCCI) am Fraunhofer SCAI entwickelt worden [BUSCH 2012, S.2f]. Diese unterstützt mit Hilfe der MpCCI-Kopplungsumgebung die Berechnung von Netznachbarn und die Interpolation physikalischer Mengenwerte.

Um die Jahrtausendwende fand die gekoppelte Simulation verschiedener Simulationswerkzeuge auch vermehrt Anwendung im Bereich der Fahrzeugsystemdynamik. Darin wurde zur Realisierung eines simulationswerkzeugunabhängigen Datenaustauschs 2008 das ITEA2-Projekt MODELISAR initiiert. Zielsetzung des Projektes ist seitdem die Entwicklung eines applikationsunabhängigen offenen Schnittstellen-Standards, dem FMI, um damit sowohl die Kopplung von Simulatoren als auch den Austausch von Simulationsmodellen im Bereich der Fahrzeugsystemdynamik zu unterstützen [BUSCH 2012, S.3]. Zu diesem Zweck werden klassische Simulationsmodelle um eine Schnittstellenbeschreibungsdatei im *Extensible-Markup-Language*

(XML)-Format ergänzt. Die beiden Dateien werden zu einer sogenannten *Functional Mock-Up Unit* zusammengefasst, wobei auch weitere Informationen mit eingebunden werden können [BLOCHWITZ 2011, S.6]. Mit den darin standardisierten Schnittstellen ist es möglich, eigene Simulationsmodelle über die beschriebenen Schnittstellen mit fremden Simulationsmodellen zu koppeln. Allerdings setzt der Prozess der Standardisierung erst bei den bereits modellierten Simulationsmodellen an, weswegen es damit nicht möglich ist, wie in der Problemstellung und Zielsetzung aus Kapitel 1.1 gefordert, eine verteilte Modellbildung über bereits im Vorfeld definierte Schnittstellen zu koordinieren. Dementsprechend kann damit nicht die Konsistenz der daraus resultierenden Konfiguration gewährleistet werden.

Um das Potential einer gekoppelten Simulation simulatorübergreifender Systeme voll ausschöpfen zu können, wurde von Stüber 2011 ein Prozess veröffentlicht, der in Ergänzung zu MODELISAR neben standardisierten Schnittstellen auch einen koordinierten Aufbau einer Gesamtsystemarchitektur mit vorab definierten Schnittstellen berücksichtigt (siehe [STÜBER 2011]). Damit kann noch vor der eigentlichen Aggregation der Teilsysteme in einem virtuellen Prototyp die Konsistenz der Schnittstellen sichergestellt werden. Die dafür erforderlichen vorgelagerten Prozessschritte „*System Design*“ und „*Model Design*“ sind in Abbildung 3.6 dargestellt.

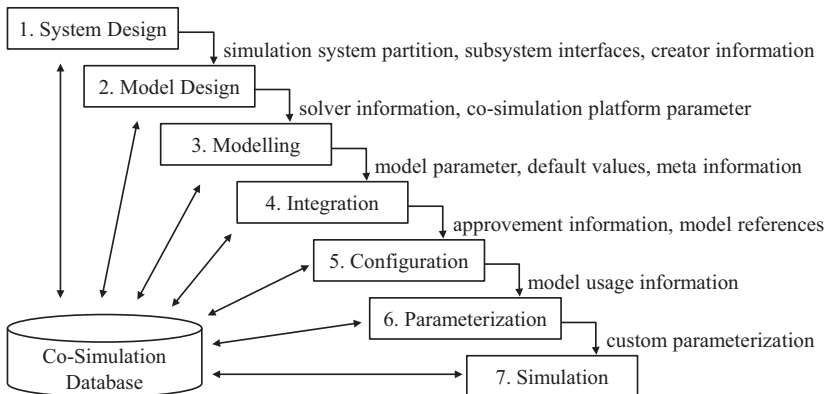


Abb. 3.6: Prozess zur Erstellung einer Co-Simulation nach [STÜBER 2011, S.79]

Darin greifen sämtliche Prozessschritte auf die in Abbildung 3.6 links unten skizzierte Co-Simulations-Datenbasis (engl. *Co-Simulation Database*) zu.

Dadurch wird die Konsistenz der Struktur (engl. *Framework*) und der Modelle auch bei mehreren Anwendern gewährleistet [STÜBER 2011, S.79].

3.3.2 Co-Simulationsprozess zur Erstellung virtueller Prototypen

Da sich der Prozess zur Erstellung virtueller Prototypen nach Stüber aus Abbildung 3.6 als Grundlage für den Modularisierungsansatz in Kapitel 4 eignet, werden im Folgenden die einzelnen Schritte detaillierter erläutert.

Analog zu den *Functional Mock-Up Units* in MODELISAR werden sämtliche Dateien, die zur Informationsweitergabe generiert werden, im XML-Format gespeichert. Damit wird innerhalb des Prozesses ein einfacher und simulationswerkzeugunabhängiger Datenaustausch sichergestellt [BRAY 2008]. Eine Übersicht sämtlicher Dateitypen, die während dieses Co-Simulationsprozesses erzeugt werden, ist in Tabelle 3.1 aufgelistet.

Tab. 3.1: Übersicht der im Prozess zur gekoppelten Simulation erzeugten Dateitypen nach [STÜBER 2011, S.79]

	File Extension	Name	Based On	Additional Information Stored
System Abstraction	.muxml	Mock-Up xml	–	Abstract Subsystem Partition Subsystem Interfaces Creator Information
	.fmuxml	Functional Mock-Up xml	.muxml	Concrete Subsystem Implementations
	.scaxml	Simulation Configuration Assembly xml	.fmuxml	Global Parameter Definitions Simulation Status Approval Information Co-Simulation Platform Information
Model Abstraction	.amxml	Abstract Model xml	–	Subsystem Interface Internal Subsystem Structure Creator Information
	.mtxml	Model Template xml	.amxml	Solver Parameter Co-Simulation Parameter
	.smxml	Simulation Model xml	.mtxml	Parameter Information Default Values
	.psmxml	Parameterized Simulation Model xml	.smxml	Custom Parameterization Global Parameter Information

Grundsätzlich wird darin zwischen zwei Typen unterschieden: zum einen den Abstraktionsdateien des Simulationssystems (engl. *System Abstraction*), die Informationen über die Struktur des kompletten Simulationssystems beherbergen (vergleiche obere Hälfte in Tabelle 3.1), und zum anderen den Abstraktionsdateien des Modells (engl. *Model Abstraction*), die Informationen der darin eingebundenen Teilsysteme speichern (vergleiche untere Hälfte in Tabelle 3.1). Die Bedeutung der einzelnen Dateien wird im Folgenden im Rahmen der Prozessbeschreibung erläutert.

Um einer aufwändigen Anpassung und Korrektur von Schnittstellen entgegenzuwirken, unterstützt der Co-Simulationsprozess im Prozessschritt „*System Design*“ den Aufbau einer Gesamtsystemarchitektur mit definierten Schnittstellen. Zu diesem Zweck wird das Gesamtsystem in sogenannte Funktionen unterteilt (siehe linker Teil in Abbildung 3.7).

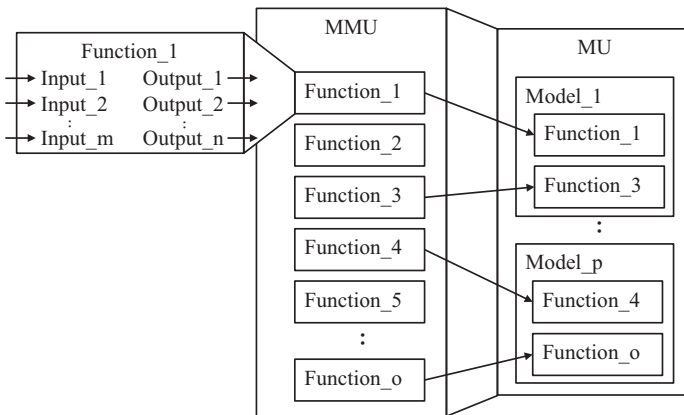


Abb. 3.7: Prozessschritt „*System Design*“ nach [STÜBER 2011, S.80]

Gegenüber anderslautenden Definitionen repräsentiert im Rahmen dieser Arbeit eine Funktion ein Element innerhalb eines Systems. Dieses wird im Verlauf einer virtuellen Entwicklung mit einem ausführbaren Modell befüllt, welches mit seiner Funktionalität zur Aufgabe des Systems beiträgt. Dem Begriff der Systemhierarchie folgend stellt eine Funktion die niedrigste, sinnvoll separierbare Ebene innerhalb der Systemhierarchie dar und verfügt über fest definierte Schnittstellen in Form von Eingangs- und Ausgangsgrößen. Sinnvoll bedeutet in diesem Kontext, dass eine Funktion detailliert genug sein soll, um einen separierbaren Teilaspekt technischer

Systeme zu beherbergen, und grob genug, um mit angepasstem Parametersatz auf gegenstandsverwandte Systeme portiert werden zu können. Solche gegenstandsverwandte Funktionen werden in einem sogenannten *Maximal Mock-Up* (MMU) zusammengefasst, welches einen maximalen Verbund an Funktionen und ihrer Schnittstellen zusammenfasst. Es repräsentiert somit alle möglichen Ausprägungen eines Systems (z. B. verschiedene Fahrzeugmodelle eines Herstellers). Da es sich dabei um ein maximales Setting an Funktionen handelt, welches nicht nur sämtliche Bestandteile eines einzelnen, sondern mehrerer unterschiedlicher Systeme beherbergt, wird es auch als „150 %-System“ bezeichnet. Dieses ist im mittleren Teil von Abbildung 3.7 dargestellt. Daraus können dann in Abhängigkeit von einem Simulationszweck virtuelle Prototypen, sogenannte *Mock-Up* (MU) aufgebaut werden, die teilweise auf identischen Funktionen basieren können. Dadurch wird die Modellwiederverwendung der in der Co-Simulations-Datenbasis abgelegten Simulationsmodelle unterstützt [STÜBER 2011, S.80]. Darüber hinaus können selbst Systeme geschaffen werden, die nicht im MMU abgelegt worden sind (z. B. kann ein Hybridfahrzeug aus der Kombination von Funktionen eines konventionell und eines elektrisch betriebenen Fahrzeugs zusammengestellt werden).

Zur Erstellung von FMUs können aus dem MMU relevante Funktionen ausgewählt und deren Schnittstellen in Abhängigkeit von einem Simulationszweck angepasst werden (engl. *Tailoring*). Diese Funktionen werden dann in einem sogenannten 100 %-System aggregiert, welches im rechten Teil von Abbildung 3.7 dargestellt ist. Die erzeugten Informationen werden in einer *Mock-Up*-Beschreibungsdatei (.muxml in Tabelle 3.1) gespeichert, die alle Informationen über enthaltene Funktionen und deren Schnittstellen erfasst. In Ergänzung dazu wird für jede darin enthaltene Funktion automatisch eine separate *Abstract-Model*-Beschreibungsdatei (.amxml in Tabelle 3.1) erzeugt.

Ausgehend davon kann im Prozessschritt „*Model Design*“ jede Funktion einem beliebigen Simulationswerkzeug, welches für die Simulationsaufgabe am besten geeignet ist, zugewiesen werden. Dadurch wird für jede Funktion aus der *Abstract-Model*-Beschreibungsdatei eine dazugehörige *Modeltemplate*-Beschreibungsdatei (.mtxml in Tabelle 3.1) erzeugt, welche mit dem jeweils gewählten Simulationswerkzeug geöffnet werden kann.

Bei dem „*Modelling*“-Schritt werden in dem gewählten Simulationswerkzeug die Simulationsmodelle in die *Modeltemplate*-Beschreibungsdateien

implementiert. Ausnahmen bilden Teilsysteme, für die bereits befüllte *Modeltemplate*-Beschreibungsdateien in der Co-Simulations-Datenbasis abgelegt worden sind und die dementsprechend wiederverwendet werden können. Generell sind für jede Funktion auch mehrere Implementierungen möglich, sofern diese den Anforderungen genügen. Für sämtliche Dateien muss dabei der Gültigkeitsbereich des Simulationsmodells spezifiziert und eine Standardparametrierung vorgegeben werden. Anschließend werden die Eingänge und Ausgänge des erstellten Simulationsmodells mit den Schnittstellen der jeweiligen *Modeltemplate*-Beschreibungsdatei verknüpft. Die Ergebnisse des „*Modelling*“-Schrittes werden anschließend in einer *Simulation-Model*-Beschreibungsdatei (.smxml in Tabelle 3.1) gespeichert, welche zusammen mit dem erstellten Modell einem Systemarchitekten zur Verfügung gestellt werden.

Erst dann können die auf diese Weise erstellten Funktionen und die darin enthaltenen Simulationsmodelle im Prozessschritt „*Integration*“ in die Gesamtsystemarchitektur des projektspezifischen MUs, also des zu erstellenden virtuellen Prototyps, eingebunden werden. Zu diesem Zweck werden die Pakete daraufhin überprüft, ob sie den formalen Anforderungen genügen, indem die *Simulation-Model*-Beschreibungsdatei und die erstellten Modelle automatisch mit der *Modeltemplate*-Beschreibungsdatei verglichen werden. Ein *Approved*-Tag kennzeichnet nach erfolgreichem Abgleich ein zulässiges Paket, wohingegen ein unzulässiges Paket mit einem Bericht über eventuelle Defizite zur Nacharbeit an den jeweiligen Entwickler zurück geschickt wird.

Der Prozessschritt „*Configuration*“ erlaubt dann in einem separaten Konfigurationswerkzeug die Anpassung der konkreten Implementierung. Dort kann aus verschiedenen Implementierungen ein und derselben Funktion die jeweils gewünschte ausgewählt werden (also z. B. ob das Fahrzeug einen Zwei- oder Vierradantrieb hat). Zu diesem Zweck werden dem Anwender für jede *Abstract-Model*-Beschreibungsdatei aus der zu simulierenden *Mock-Up*-Beschreibungsdatei die jeweils gültigen Implementierungen ausgeliefert und angezeigt. Diese Information wird in einer *Functional-Mock-Up*-Beschreibungsdatei (.fmuxml in Tabelle 3.1) gespeichert.

Die „*Parameterization*“ erfolgt im selben Werkzeug und gestattet ausgehend von der *Functional-Mock-Up*-Beschreibungsdatei die Einstellung und Modifizierung der Parameter. Dabei werden für jedes Teilsystem die Parameter (wie z. B. geometrische Abmessungen eines Objektes) an den

jeweiligen Anwendungsfall angepasst und sogenannte globale Parameter (wie z. B. eine Initialisierungstemperatur) definiert, die dann über alle Simulationsmodelle hinweg konsistent sind. Die getätigten Einstellungen werden dann in der *Simulation-Configuration-Assembly*-Beschreibungsdatei (.scaxml in Tabelle 3.1) und der *Parameterized-Simulation-Model*-Beschreibungsdatei (.psmxml in Tabelle 3.1) gespeichert und an die Co-Simulations-Datenbasis geschickt.

Von dort können dann im Prozessschritt „*Simulation*“ verschiedene Anwender aus verschiedenen Abteilungen die Simulation starten. Dazu wird die Co-Simulationsumgebung geöffnet und die benötigten Daten werden geladen. Während der Initialisierung werden die Parameter aus der .psmxml-Datei geladen. Mit dem Start der Simulation synchronisiert eine neutrale Kopplungssoftware die Zeitschrittweiten sämtlicher involvierter Simulatoren. Abschließend erhält der Initiator der Simulation eine Nachricht über den Abschluss der Simulation und die Ergebnisse können ausgewertet werden.

4 Strukturierung und Absicherung technischer Gesamtsysteme

„Für die Berechnung mit einer verteilten Simulation muss das zuvor erstellte Modell in Teilsysteme zerlegt werden. Der Grund für dieses Vorgehen liegt in der Spezialisierung verschiedener Integriertoren für spezifische Anwendungsfälle. So kann zum Beispiel ein mechanisches System im Vergleich zu einem hydraulischen System träge sein, was sich auf die mathematische Berechnung auswirkt“ [GEIMER 2006, S.572].

Mit dem Ziel einer simulatorübergreifenden integrierten Gesamtsystemsimulation wurde der klassische Prozess der Modellbildung und Simulation aus Kapitel 2 innerhalb von Kapitel 3 um Aspekte zur Modularisierung und Kopplung solcher Systeme erweitert. Basierend darauf wurde ein Prozess für gekoppelte Simulationen eingeführt, der einen koordinierten Aufbau eines FMUs mit vorab definierten Schnittstellen berücksichtigt.

Um einen virtuellen Entwicklungsprozess unter Ausschöpfung des darin enthaltenen Potentials verwenden zu können, muss das Gesamtsystem sinnvoll modularisiert werden. Gemäß dem einleitenden Zitat aus einem Zeitschriftenaufsatz zum Thema „*Co-Simulation, gekoppelte Simulation oder Simulatorkopplung?*“ ist eine solche Modularisierung erforderlich, um die Simulationsmodelle innerhalb der daraus resultierenden Teilsysteme unter Einsatz beliebiger Simulatoren abbilden zu können. Im Rahmen dessen spielen die Systemgrenzen und Schnittstellen der Teilsysteme eine entscheidende Rolle, da diese die Architektur für die darauf aufbauende Gesamtsystemsimulation definieren.

Zu diesem Zweck wird im Folgenden für die virtuelle Entwicklung eines FMU ein Ansatz zur Modularisierung bei gleichzeitig vollständiger

Erfassung technischer Gesamtsysteme vorgestellt. Dieser Modularisierungsansatz wird im weiteren Verlauf für die Strukturierung technischer Systeme appliziert, womit durch die Definition einer Gesamtsystemarchitektur die Grundlagen für eine simulatorübergreifende integrierte Gesamtsystems simulation gelegt werden. Darauf aufbauend werden Maßnahmen vorgestellt, um bei der Erstellung eines FMUs bereits in frühen Phasen der virtuellen Produktentwicklung die Struktur der Gesamtsystemarchitektur absichern zu können. Abschließend werden die vorgestellten Definitionen und Verfahren anhand einer konkreten Gesamtfahrzeugmodellarchitektur in mehreren Ausbaustufen evaluiert.

4.1 Modularisierungsansatz

Gemäß den einleitenden Worten ist es für die Modellbildung und Simulation simulatorübergreifender Systeme erforderlich, das zu modellierende Gesamtsystem in mehrere Teilsysteme zu untergliedern. Mit Hilfe eines geeigneten Modularisierungsansatzes wird eine solche Modularisierung bei gleichzeitig vollständiger Erfassung der Systeme ermöglicht. Dazu wird, wie in der Problemstellung und Zielsetzung aus Kapitel 1.1 gefordert, der Fokus auf technische Systeme gelegt, um diese mit Hilfe des Modularisierungsansatzes vollständig erfassen und beschreiben zu können und damit die relevanten Kategorien zur Entwicklung virtueller Prototypen aufzuspannen.

Für eine solche Kategorisierung technischer Systeme wird von den naturwissenschaftlichen Disziplinen ausgegangen. Darunter werden Wissenschaften zusammengefasst, die empirisch arbeiten und sich mit der Erforschung der Natur befassen [HABERMAS 1973]. Diese umfassen gemäß Beitz und Grote alle Bereiche der Herstellung und Anwendung maschinenbaulicher Produkte, angefangen bei der Produktplanung, Forschung, Entwicklung, Konstruktion, Arbeitsvorbereitung, Normung, Materialwirtschaft, Fertigung, Montage und Qualitätssicherung über den technischen Vertrieb bis zur Bedienung, Überwachung, Instandsetzung und deren Recycling [BEITZ 2001, S.VII]. Von diesen Bereichen werden für den hier betrachteten Modularisierungsansatz zur Entwicklung virtueller Prototypen ausschließlich Gebiete der Entwicklung betrachtet, die der Erarbeitung maschinenbaulicher Produkte zugeordnet werden können. Diese umfassen die dargestellten Bereiche:

- Mechanik (Me)
- Thermodynamik (Th)
- Werkstofftechnik (We)
- Strömungslehre (St)
- Elektrotechnik (El)
- Steuerungs- und Regelungstechnik (SR)

Mit Hilfe dieser Bereiche ist eine vollständige Modularisierung von FMUs möglich. Da diese Bereich zur Kategorisierung technischer Produkte eingesetzt werden, werden sie im Folgenden als „Kategorien“ bezeichnet.

4.2 Strukturierung technischer Gesamtsysteme

Zur Strukturierung technischer Gesamtsysteme mit Hilfe der gewählten Kategorien wird von einem projektspezifischen Verbund ausgegangen, welcher im Rahmen des Co-Simulationsprozesses zur Entwicklung virtueller Prototypen aus Kapitel 3.3.2 als „MU“ definiert worden ist. Dabei handelt es sich, wie in Abbildung 4.1 dargestellt, um eine Untermenge des MMUs, welches auf Funktionen als kleinste austauschbare Elemente basiert. Um gewährleisten zu können, dass diese das Gesamtsystem vollständig beschreiben, werden die darin enthaltenen Funktionen anhand der gewählten Kategorien (engl. *Categories*) spezifiziert. Die Funktionen und die jeweiligen Kategorien sind am rechten Rand von Abbildung 4.1 schematisch dargestellt.

Gemäß dem Begriff der Systemhierarchie aus Kapitel 2.1.2 ist es möglich, Funktionen in einer Hierarchie abzubilden, da diese wiederum als Teilsysteme übergeordneter Systeme interpretiert werden können. Aufgrund der Vielzahl möglicher Funktionen in einem MMU ist eine solche Gliederung für die praktische Anwendung sinnvoll, da durch eine entsprechende Struktur die Orientierung der Anwender bei der Erstellung eines projektspezifischen MUs aus dem MMU unterstützt wird. Zu diesem Zweck werden unterhalb des MUs, wie in Abbildung 4.1 schematisch dargestellt, neben den Funktionen auch Domänen (engl. *Domains*) und Kompositionen (engl. *Compositions*) als weitere Ebenen eingeführt. Die daraus resultierenden Strukturelemente werden im Folgenden erläutert.

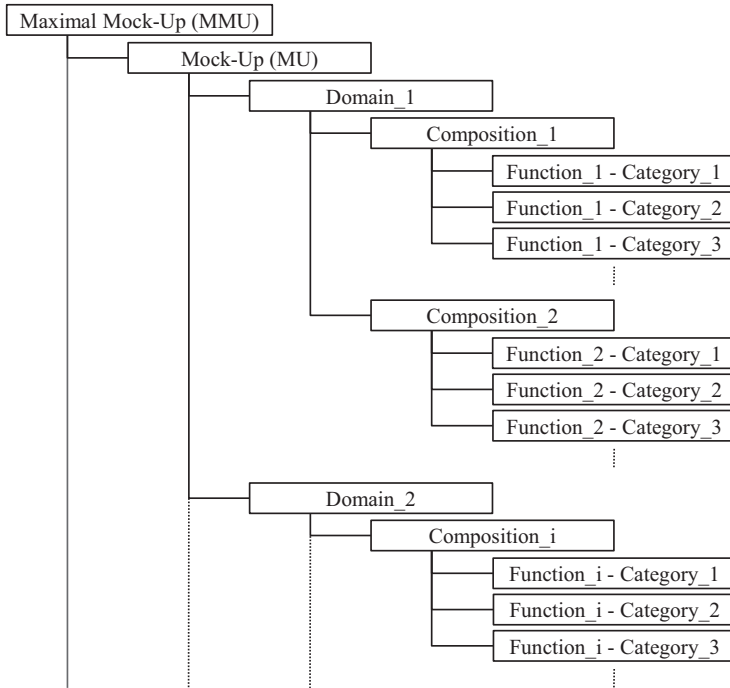


Abb. 4.1: Hierarchie zur Strukturierung von Funktionen nach [NETTER 2013c, S.3]

4.2.1 Definition der Strukturelemente

Domänen repräsentieren im Sinne einer Aufgabe für das Gesamtsystem Funktionsbereiche, welche je nach betrachtetem technischem System unterschiedlich sein können. So wird z. B. in der Kraftfahrzeugindustrie zwischen den Domänen Antrieb, Fahrwerk, Aufbau und Elektrik/Elektronik unterschieden [BRAESS 2003, S.XIIIff]. Im Rahmen dieser Arbeit werden Domänen eingesetzt, um gemäß Abbildung 4.1 die in dem MMU enthaltenen Kompositionen und Kategorien der Funktionen zu strukturieren. Wie dargestellt, kann jede Domäne mehrere Kompositionen enthalten. Dadurch wird die Zuordnung und Wiederauffindung von Funktionen bei der Erstellung eines MUs aus dem MMU unterstützt.

Gemäß Parton [PARTON 2011] ist eine Komposition eine Kombination bestimmter Teile oder Elemente, um daraus ein Ganzes zu formen. Mit diesem Begriff wird hier, wie in Abbildung 4.1 dargestellt, die Bündelung

der verschiedenen Kategorien eines Teilsystems, wie z. B. eines Wärmetauschers, bezeichnet. Dabei erfüllt die Komposition im Sinne einer Aufgabe bestimmte Funktionalitäten für das Gesamtsystem, sobald im Rahmen der virtuellen Entwicklung in den darunter angeordneten Funktionen Modelle implementiert worden sind. Funktionalitäten bezeichnen dabei das Vermögen des daraus resultierenden virtuellen Prototyps, beim Einsatz unter spezifizierten Bedingungen Ergebnisse zu liefern, die festgelegte und vorausgesetzte Erfordernisse erfüllen [DEUTSCHES INSTITUT FÜR NORMUNG 2010]. Kompositionen gestatten es, dass die darunter angeordneten Funktionen in Abhängigkeit von den in Kapitel 4.1 definierten Kategorien beliebigen Simulatoren zugeordnet werden können [NETTER 2013c, S.4]. So können z. B. die thermischen Eigenschaften eines Wärmetauschers mit Hilfe einer Funktion der Kategorie „Thermodynamik“ abgebildet werden, wohingegen dessen Materialeigenschaften einer Funktion der Kategorie „Werkstofftechnik“ zugeordnet werden können. Damit ist es möglich, diese Kategorien später unterschiedlichen Simulationswerkzeugen zuzuweisen, welche dann für die Modellbildung und Simulation z. B. von Stoff- und Wärmeströmen oder auch Festigkeitsanalysen entsprechende Umgebungen bereitstellen. Je nach betrachteter Komposition und Simulationszweck sind für eine vollständige Beschreibung von Funktionen nicht zwangsläufig alle der in Kapitel 4.1 spezifizierten Kategorien erforderlich.

Wie bei der Einführung des Co-Simulationsprozesses in Kapitel 3.3.2 vorgestellt, repräsentiert eine Funktion ein Element innerhalb eines Systems, wie z. B. die bereits genannte Funktion der Kategorie „Thermodynamik“ eines Wärmetauschers. Durch den gewählten Modularisierungsansatz ist eine Funktion stets an eine Kategorie gebunden. Damit kann diese im Verlauf der virtuellen Entwicklung unter Einsatz beliebiger Simulationswerkzeuge mit Modellen befüllt werden, um zur Aufgabe des Systems beizutragen. Dem Begriff der Systemhierarchie folgend stellt eine Funktion die niedrigste sinnvoll separierbare Ebene innerhalb der Systemhierarchie dar und verfügt über fest definierte Schnittstellen in Form von Eingangs- und Ausgangsgrößen [STÜBER 2011, S.80]. Diese Ebene muss detailliert genug sein, um einen separierbaren Aspekt eines Gesamtsystems zu beherbergen, und gleichzeitig grob genug, um durch eine Parameteranpassung für gegenstandsverwandte Systeme verwendet werden zu können. Dabei darf die Granularität, also der Detaillierungsgrad der Funktionen, weder zu grob noch zu fein sein. Dieser Zusammenhang ist in Abbildung 4.2 qualitativ veranschaulicht.

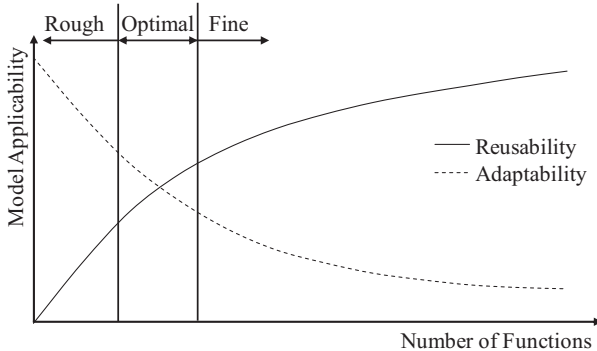


Abb. 4.2: Anpassungsfähigkeit und Wiederverwendbarkeit von Funktionen nach [NETTER 2013c, S.4]

Eine zu grobe Modularisierung des Systems und aufgrund dessen eine geringe Anzahl an durch die Funktionen abgegrenzten Teilsystemen würde zu einer unzureichenden Portierbarkeit der darin eingebetteten Simulationsmodelle führen. Grund dafür ist, dass sich Funktionen bei geringer Anzahl nicht mehr definiert auf eine Komposition innerhalb einer Domäne beschränken lassen. Dementsprechend können solche kompositions- oder sogar domänenübergreifende Elemente nur noch für einen einzelnen konkreten projektspezifischen Verbund (MU) verwendet werden, wodurch das Potential für eine Modellwiederverwendbarkeit sinkt. Dieser Zusammenhang ist in Abbildung 4.2 als Kurve für die Wiederverwendbarkeit (engl. *Reusability*) dargestellt. Demnach steigt das Potential für eine Wiederverwendbarkeit mit einer detaillierteren Modularisierung und einer damit steigenden Anzahl an Funktionen, da sich diese in die Struktur verschiedener MUs einbinden lassen. Im Gegenzug führt jedoch eine sehr feingranulare Modularisierung wegen der hohen Anzahl an resultierenden Teilsystemen zu einer aufwändigen Anpassung der Schnittstellen. Damit steigt der Anpassungsaufwand mit steigender Anzahl an Funktionen und die Anpassungsfähigkeit sinkt. Dieser Zusammenhang ist in Abbildung 4.2 als Anpassungsfähigkeit (engl. *Adaptability*) gekennzeichnet. Aufgrund dessen muss ein Kompromiss zwischen Wiederverwendbarkeit und Anpassungsfähigkeit, in Abbildung 4.2 schematisch als optimaler Bereich (engl. *Optimal*) skizziert, gefunden werden [NETTER 2013c, S.3].

4.2.2 Definition der Teilsystemgrenzen

Dem Prozess zur Erstellung virtueller Prototypen aus Kapitel 3.3.2 folgend, ist es mit den vorgestellten Strukturelementen möglich, jede Kategorie einer jeden Funktion im Prozessschritt „*Model Design*“ einem beliebigen Simulationswerkzeug zuzuweisen, bevor im „*Modelling*“-Schritt die jeweils benötigten Simulationsmodelle implementiert werden. Dabei muss darauf geachtet werden, dass bei der Spezifikation der zu implementierenden Modelle sowohl zwischen den Kompositionen als auch zwischen den Funktionen keinerlei Überschneidungen an den Teilsystemgrenzen stattfinden, um einer Dopplung von Systemteilen vorzubeugen. Da aufgrund der Schnittstellen einer Funktion bereits impliziert wird, innerhalb welcher Funktion welche Funktionalitäten berechnet werden, muss deren Aufteilung bereits vor der Festlegung der Schnittstellen spezifiziert werden. Der Anwender hat somit gewisse Freiheiten zu entscheiden, welches Teilsystem welche Berechnung vornimmt [LUND 2008, S.177]. Dieser Umstand ist in Abbildung 4.3 exemplarisch anhand des Wärmetauschers dargestellt.

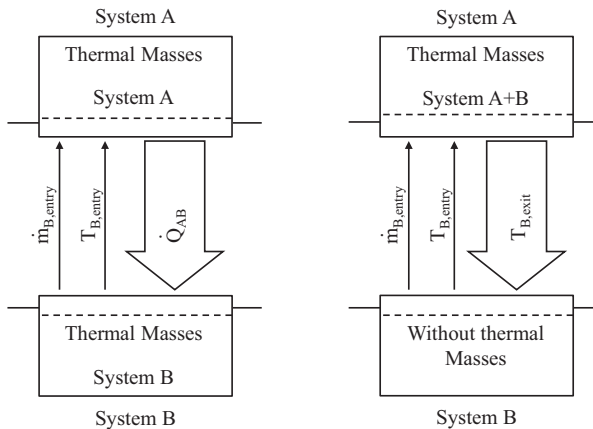


Abb. 4.3: Optionen zur Berechnung der Übergabevariablen nach [LUND 2008, S.176]

Dabei hängen die Übergabevariablen davon ab, innerhalb welcher Funktion und damit innerhalb welches Simulationswerkzeugs die thermischen Massen berechnet werden. Wie in der linken Variante aus Abbildung 4.3 dargestellt, sind die thermischen Massen beider Systeme innerhalb des jeweiligen Systems modelliert. Dementsprechend kann in System A auf

Basis der Eintrittstemperatur und des Massenstroms von System B innerhalb von System A der Wärmestrom berechnet und übergeben werden. Alternativ sind in der rechten Variante aus Abbildung 4.3 beide thermischen Massen in System A modelliert. Damit kann in System A auf Basis der Eintrittstemperatur und des Massenstroms von System B direkt die Austrittstemperatur von System B ermittelt und an System B transferiert werden. In diesem alternativen Szenario hat System A mehr Freiheitsgrade, was dessen Stabilität erhöhen und die Anzahl der notwendigen Iterationen je Zeitschritt verringern kann. Gleichzeitig wird System B vereinfacht, was ebenfalls zu einer Reduzierung der Iterationen führen kann. Somit ist es möglich, Systeme bereits durch eine geeignete Aufteilung numerisch zu stabilisieren und Rechenzeit einzusparen [LUND 2008, S.176f]. Dementsprechend wichtig ist die Wahl geeigneter Teilsystemgrenzen, welche durch deren Schnittstellen festgelegt werden.

4.2.3 Definition der Signale und Schnittstellen

Der Bedarf einer simulatorübergreifenden integrierten Gesamtsystemsimulation bedingt aufgrund der Vielfalt und Vielzahl an physikalischen Abhängigkeiten eine einheitliche und eindeutige Benennung der auszutauschenden Signale. Grund dafür ist, dass es bei den Funktionen innerhalb einer gekoppelten Simulation nur mit erheblichem Aufwand möglich ist, die Bedeutung der Schnittstellen und damit auch der ausgetauschten Signale aus dem Kontext der darin implementierten Funktionalitäten zu interpretieren [NETTER 2013c, S.5]. Um auch bei komplexen FMUs die Interaktion zwischen einzelnen Funktionen und den darin eingebetteten Simulationsmodellen nachvollziehen zu können, wird eine Signalnamenskonvention eingeführt, die die Eigenschaften der Signale berücksichtigt. Deren Struktur ist in Abbildung 4.4 dargestellt.

Jedes Signal innerhalb eines Systems hat nur eine singuläre Quelle, da die gleiche Information nicht aus zwei verschiedenen Funktionen stammen kann. Aus diesem Grund referenziert der erste Teil der Signalnamenskonvention, wie in Abbildung 4.4 dargestellt, die Quelle. Darin repräsentiert die erste Spalte die jeweilige Funktion, aus der die Information stammt, und die zweite Spalte die dazugehörige Kategorie. Das Simulationsmodell innerhalb einer Funktion wandelt Eingangsgrößen in Ausgangsgrößen um. Gemäß Winer [WINER 1968] können alle dabei auftretenden Phänomene durch die

Function	1. Identifier	Phys. Identifier	2. Identifier	Unit
BattHV CarPhy Pedal	PowerSuplyHV DrivePhy DriveControl	P F Angle	Chemical AxleLoadFA Position	W N degree

Abb. 4.4: Signalnamenskonvention für die Schnittstellen [NETTER 2013c, S.4]

Übertragung von Masse, Energie, Informationen oder deren Kombination beschrieben werden. Um diese zu spezifizieren, enthält die dritte Spalte den dazugehörigen physikalischen Bezeichner (engl. *Physical Identifier*) wie Masse, Kraft, Weg oder Position. Die vierte Spalte enthält ein Freitextfeld, innerhalb dem die Funktion z. B. mit Informationen über die Quelle oder weiteren Informationen über die Funktion selbst weiter spezifiziert werden kann. Durch die Eingabe des Wertes *Void* kann dieses Feld auch freigelassen werden, falls die Funktion bis zu diesem Punkt bereits eineindeutig beschrieben ist [SCHREYER 2012, S.27]. Die letzte Spalte enthält Informationen über die Einheit des jeweiligen Signals, welches in dem jeweils zugrunde liegende Einheitensystem wie dem Internationalen Einheitensystem (*Système International d'Unités* (SI)) oder dem englisch/amerikanischen System (*Imperial System of Units* (IPS)) definiert wird. Dadurch wird vermieden, dass versehentlich Signale mit identischem physikalischem Bezeichner, aber unterschiedlichen Einheitensystemen, wie Kilometer und Meilen, miteinander gekoppelt werden. Darüber hinaus wird auch die Einheit festgelegt und damit vermieden, dass verschiedene Potenzprodukte der gleichen Basiseinheit, wie Gramm und Kilogramm, miteinander verknüpft werden.

Als Beispiel repräsentiert die erste Zeile in Abbildung 4.4 ein Signal, welches aus der Funktion „Hochvoltbatterie“ der Kategorie „Elektrotechnik“ stammt. Dieses enthält als physikalischen Bezeichner die Leistung der Batterie. Der freie Bezeichner weist darauf hin, dass es sich um ein chemisches Batteriemodell handelt. Die Einheit der Leistung ist in Watt angegeben [NETTER 2013c, S.5]. Das zweite Beispiel stammt aus der Funktion „Fahrphysik“ der Kategorie „Mechanik“. Dieses überträgt die Kraft

der Vorderachslast in Newton. Das dritte Signal stammt aus der Funktion „Fahrpedal“ der Kategorie „Steuerungs- und Regelungstechnik“. Dieses überträgt den Winkel der Fahrpedalstellung in Grad.

4.3 Absicherung technischer Gesamtsysteme

Im Rahmen des in Kapitel 3.3.2 vorgestellten Prozesses zur Entwicklung virtueller Prototypen definiert die Gesamtsystemarchitektur das Grundgerüst für eine simulatorübergreifende integrierte Gesamtsystems simulation. Die darin im Laufe der Entwicklung virtueller Prototypen implementierten Simulationsmodelle definieren das Potential des Systems, das statische und dynamische Verhalten komplexer technischer Systeme und seiner Komponenten automatisiert, gefahrlos und wiederholbar zu untersuchen. Dementsprechend wichtig ist die fehlerfreie Erstellung der Gesamtsystemarchitektur und der darin implementierten Simulationsmodelle, um eine aufwändige Nacharbeit in den späteren Prozessphasen vermeiden zu können. Zu diesem Zweck wird im Folgenden ein Verfahren vorgestellt, mit dessen Hilfe bereits in frühen Prozessphasen überprüft werden kann, ob die Schnittstellen der darin enthaltenen Funktionen zueinander passen. Darüber hinaus wird ein Ansatz erläutert, mit dem in den späteren Prozessphasen auf Basis aufgezeichneter Messdaten eine Absicherung der in der Struktur implementierten Simulationsmodelle gewährleistet werden kann.

4.3.1 Absicherung der Gesamtsystemarchitektur

Bereits nach dem ersten Prozessschritt „*System Design*“ kann die Durchgängigkeit der gewählten und angepassten Funktionen abgesichert werden, indem die Konsistenz der aus diesem Prozessschritt resultierenden .muxml-Beschreibungsdatei überprüft wird. Dazu wird innerhalb des MUs überprüft, ob alle Eingangsgrößen mit Ausgangsgrößen anderer Funktionen verknüpft sind oder zumindest von einer Parameterdatei mit Daten für die spätere Simulation versorgt werden. Wichtiges Hilfsmittel dafür ist der *Signal Manager* (siehe Abbildung 4.5) in dem Architekturwerkzeug SystemDesk von dSpace.

Dieses Simulationswerkzeug ist für die Planung, Implementierung und Integration komplexer Systemarchitekturen und verteilter Software-Systeme geeignet. Es wird für die modellbasierte Entwicklung von AUTOSAR

Name	Type	Width	Unit	Quantity	Sender List	Receiver List
EModul1_EI_M_eff_Nm	Float	0	Newtonmeter		EModul1_elEn	DiffVA_meEn, AntriebsReg_StRe
EModul1_EI_M_max_Nm	Float	0	Newtonmeter		EModul1_elEn	AntriebsReg_StRe
EModul1_EI_P_auf_W	Float	0	Watt		EModul1_elEn	BattHV_EI
Fahrer_SR_a_Soll_mps2	Float	0	m/s ²		Fahrer_SR	Pedale_StRe
Fahrer_SR_Winkel_Radlenk_rad	Float	0	rad		Fahrer_SR	LenkReg_StRe
FzgLaengsDyn_Me_a_ist_mps2	Float	0	m/s ²		FzgLaengsDyn_meEn	Raeder_meEn, FzgQuerDyn_m...

Abb. 4.5: Signalabhängigkeiten im *Signal Manager* (SystemDesk-Screenshot)

Steuergeräte-Software sowie die graphische Darstellung der Komponenten und ihrer Kommunikation verwendet. Im Rahmen des in Kapitel 3.3.2 beschriebenen Co-Simulationsprozesses wurden diese Anwendungsgebiete um ein Plug-in erweitert, welches einen Eingabedialog für Systembeschreibungselemente, eine Ablage für die Systembeschreibung und angepasste Validierungs- und Analysefunktionen bereitstellt. Damit unterstützt es einen XML-basierten Aufbau eines MMUs und daraus die Ableitung und Verifikation konkreter MUs.

Mit Hilfe des *Signal Managers* kann die Konsistenz der .muxml-Beschreibungsdatei durch den sogenannten Konsistenz-Check toolgestützt durchgeführt werden. Dabei wird die potentielle Simulierbarkeit einer aus der .muxml-Beschreibungsdatei resultierenden Gesamtsystemsimulation sichergestellt, wodurch eine Anpassung der Struktur in den darauf folgenden Prozessschritten vermieden wird. Eine Absicherung der Schnittstellen der Gesamtsystemarchitektur ist dabei ohne Simulationsmodelle möglich.

4.3.2 Absicherung der Simulationsmodelle

Sobald in den späteren Prozessschritten Simulationsmodelle verfügbar sind, können noch vor der eigentlichen Gesamtsystemsimulation die in die Struktur implementierten Simulationsmodelle abgesichert werden. Dies geschieht indem die Signale an den Schnittstellen jeder einzelnen Funktion mit gegebenenfalls bereits verfügbaren korrelierenden Realdaten verglichen werden. Dabei handelt es sich gemäß Kapitel 2.4.2 um eine Form der Validierung, bei der unter Anwendung geeigneter mathematischer Verfahren die Übereinstimmung zwischen ausführbarem Modell und Realität quantifiziert wird. Die erforderlichen Realdaten können aus Vorgängermodellen, Prüfstandversuchen und Technikträgern gewonnen werden. Dabei ist es unerheblich, ob diese an einzelnen Komponenten oder dem Gesamtsystem

aufgezeichnet werden, insofern deren Wertebereiche mit den Spezifikationen der Simulationsmodelle und deren Signale übereinstimmen.

Allerdings stehen in Abhängigkeit von der gewählten Struktur und der daraus resultierenden Schnittstellen nicht immer für alle Signale Messwerte aus den jeweils betrachteten technischen Systemen zur Verfügung. Grund dafür ist, dass es sich dabei um Signale handeln kann, für die keine Messwerte aufgezeichnet worden sind oder die aus technischen Gründen nicht erfasst werden können. Um dennoch eine Quantifizierung zu ermöglichen, wurde im Rahmen dieser Arbeit ein Algorithmus entwickelt, der es ermöglicht, betroffene Funktionen derart mit anderen Funktionen zu kombinieren, dass fehlende bzw. unvollständige Signale ausgeblendet werden können. Aus dieser Kombination resultieren sogenannte Funktionsgruppen (engl. *Function Groups*), deren Schnittstellen vollständig mit den zur Verfügung stehenden Messdaten beaufschlagt werden können. Ein schematisches Beispiel für eine solche Funktionsgruppe (FG) ist in Abbildung 4.6 dargestellt.

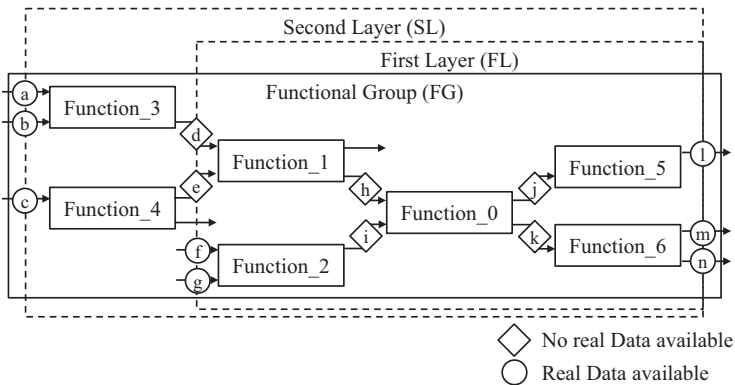


Abb. 4.6: Schema einer Funktionsgruppe [NETTER 2013a, S.44]

Wie darin an *Function 0* veranschaulicht, sind sowohl deren Eingangs- als auch Ausgangsgrößen nicht als Realdaten verfügbar. Um dennoch das darin implementierte Simulationsmodell quantifizieren zu können, werden die Ein- und Ausgänge dieser Funktion mit den Aus- und Eingängen anderer Funktionen verbunden, um das Problem fehlender Schnittstelleninformationen zu übergehen. In dem Beispiel wird das Eingangssignal „h“ von „*Function 1*“ und das Eingangssignal „i“ von „*Function 2*“ bereitgestellt. Die fehlenden Ausgangssignale „j“ und „k“ werden von „*Function 5*“ und „*Function 6*“

übernommen. Da bei „*Function 1*“ wiederum beide Eingangssignale nicht als Messdaten verfügbar sind, werden die fehlenden Eingangssignale „d“ und „e“ von „*Function 3*“ bzw. „*Function 4*“ zur Verfügung gestellt. Um dabei Funktionen, die direkt mit der zu untersuchenden Funktion verbunden sind, von Funktionen, die ausschließlich als Datenlieferanten dienen, unterscheiden zu können, wurden verschiedene Ebenen (engl. *Layers*) eingeführt. In dem gewählten Beispiel gehören die Funktionen 1, 2, 5 und 6 zu der ersten Ebene (engl. *First Layer*) und die Funktionen 3 und 4 zu den Funktionen der zweiten Ebene (engl. *Second Layer*) [NETTER 2013a, S.44].

Aufgrund der Fülle an zu quantifizierenden Funktionen wurde mit Hilfe der Programmiersprache C# ein *Tool* implementiert, welches eine automatisierte Gruppierung zu untersuchender Funktionen ermöglicht. Darin werden beim Öffnen einer .muxml-Beschreibungsdatei sämtliche Informationen über die darin enthaltenen Funktionen und deren Verbindungen ausgelesen. Diese werden anschließend in einer Anzeige (engl. *Graphical User Interface* (GUI)) in alphabetischer Reihenfolge aufgelistet, woraufhin der Anwender die Verfügbarkeit der Signale als Messdatenstrom bestimmen muss. Da einzelne Signale für mehrere Funktionen verwendet werden können und dementsprechend mehrfach auftreten, werden deren Redundanzen ausgeblendet, um eine übersichtliche Anzeige zu gewährleisten. In einem weiteren Schritt muss die zu validierende Funktion ausgewählt werden. Anschließend werden die Angaben eingelesen und ein Suchalgorithmus beginnt mit der Analyse. Dessen Logik ist in Abbildung 4.7 dargestellt.

Im ersten Schritt wird für die zu validierende Funktion überprüft, ob deren Ein- und Ausgänge als Messdaten verfügbar sind (Nr. 1 in Abbildung 4.7). Falls bereits für sämtliche Schnittstellen Daten vorliegen, wird der Suchalgorithmus angehalten, da die Güte der Funktion direkt quantifiziert werden kann (Nr. 5 in Abbildung 4.7). Für den Fall, dass eine oder mehrere Schnittstellen nicht direkt validiert werden können, werden die betroffenen Schnittstellen in separaten Listen abgespeichert (Nr. 2 in Abbildung 4.7). Dies geschieht unabhängig davon, ob es sich dabei um Ein- oder Ausgänge handelt. Für die betroffenen Signale werden im nächsten Schritt Funktionen gesucht, deren Ausgänge mit den nicht vorhandenen Eingangssignalen bzw. deren Eingänge mit den nicht vorhandenen Ausgangssignalen korrelieren (Nr. 3 in Abbildung 4.7). Dabei fährt der Suchalgorithmus so lange fort, bis sämtliche Funktionen für die noch fehlenden Schnittstellen dieser ersten Ebene identifiziert und gespeichert worden sind (Nr. 4 in Abbildung 4.7).

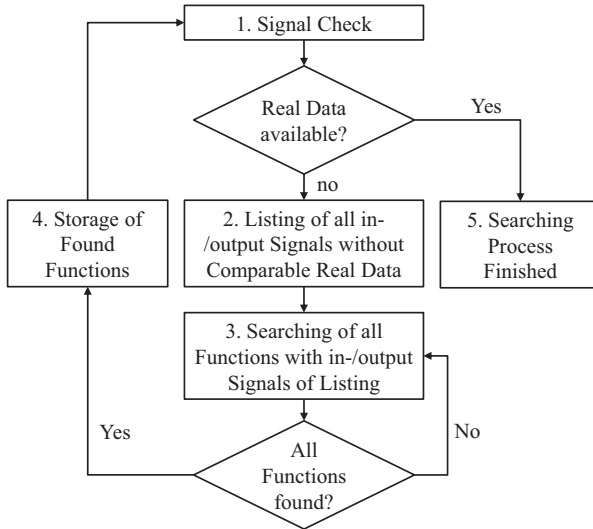


Abb. 4.7: Schematischer Ablauf des Suchalgorithmus [NETTER 2013a, S.45]

Zur Vermeidung von Redundanzen werden mehrfach aufgelistete Funktionen gelöscht. Im weiteren Verlauf beginnt der Suchalgorithmus an den Rändern der soeben identifizierten Funktionen der ersten Ebene erneut mit der Überprüfung der Schnittstellen. Grund dafür ist, dass es vorkommen kann, dass für die soeben identifizierten Funktionen, wie bereits anhand Darstellung 4.6 erläutert, wiederum Realdaten fehlen. In diesem Fall wird die Suche neu initialisiert und so lange fortgeführt, bis sämtliche Signale der daraus resultierenden Ebene mit Messdaten validiert werden können. Aufgrund der Tatsache, dass im Rahmen der Absicherung aus Kapitel 4.3.1 durch den Konsistenz-Check gewährleistet wird, dass sämtliche Eingänge mit einem korrespondierenden Ausgang verbunden sind, ist es immer möglich, eine validierbare Funktionsgruppe zu finden [NETTER 2013a, S.45].

4.4 Evaluation anhand einer Gesamtfahrzeugmodellarchitektur

Zur Evaluation des Modularisierungsansatzes, der daraus abgeleiteten Strukturierung technischer Gesamtsysteme und deren Verfahren zur Ab-

sicherung wird im Folgenden eine Gesamtfahrzeugmodellarchitektur für das FMU eines *Extended Range Electric Vehicle* (EREV)¹ erstellt. Damit sollen in einem ersten Schritt Fahrleistungen und Energieverbrauch eines Audi A1 e-tron abgeschätzt werden können. Diese werden in der Entwicklung genutzt, um Prognosen über das Beschleunigungsverhalten und die dabei verbrauchte Energie anstellen zu können. Da in den frühen Phasen des virtuellen Produktentstehungsprozesses einfache Modelle für generelle Aussagen zum Verhalten des Systems unter bestimmten Bedingungen (z. B. Machbarkeitsstudien) ausreichen, während mit dem Fortschreiten des Entwicklungsprozesses immer detailliertere Simulationsmodelle benötigt werden [DRONKA 2004, S.3], muss gemäß der Problemstellung und Zielsetzung aus Kapitel 1.1 eine beliebige Skalierung der Strukturelemente gewährleistet werden. Um diese ebenfalls evaluieren zu können, wird das System in einem zweiten Schritt um weitere Teilsysteme erweitert, die als zusätzliches Anwendungsbeispiel eine Auslegung, Applikation und Absicherung der Betriebsstrategie für das Zu- und Abschalten eines *Range Extenders*² unter realitätsnahen Bedingungen ermöglichen.

4.4.1 Strukturierung einer Gesamtfahrzeugmodellarchitektur

Gemäß der Einführung des Co-Simulationsprozesses in Kapitel 3.3.2 ist der erste Schritt beim Aufbau eines FMU die Abbildung der Struktur im Co-Simulationsprozessschritt „*System Design*“. Zu diesem Zweck wurden die Elemente des in Abbildung 4.8 dargestellten Schemas einer Gesamtfahrzeugmodellarchitektur in dem Architekturwerkzeug SystemDesk abgebildet. Eine detaillierte Darstellung des Architekturlayouts mit sämtlichen Signalen und Schnittstellen ist im Anhang C.1 dargestellt.

Aufbau der ersten Ausbaustufe

Innerhalb der Gesamtfahrzeugmodellarchitektur sind in der ersten Ausbaustufe sämtliche relevante Funktionen dargestellt, die für eine Abschätzung des Energieverbrauchs des FMU des Elektrofahrzeugs benötigt werden.

¹ *Extended Range Electric Vehicle*: Elektrofahrzeug mit Reichweitenverlängerer

² *Range Extender*: Zusatzaggregat zur Erhöhung der Reichweite elektrisch angetriebener Fahrzeuge

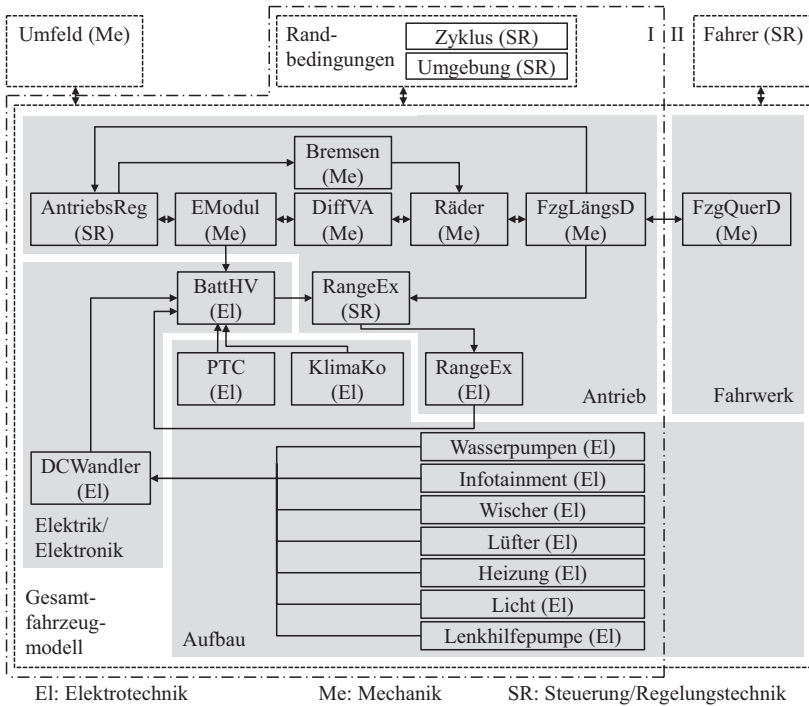


Abb. 4.8: Schema einer Fahrzeugmodellarchitektur nach [NETTER 2013c, S.6]

Diese ist in Abbildung 4.8 durch eine Strichpunktlinie von den aktuell noch nicht benötigten Teilsystemen abgegrenzt. Gemäß dem zu identifizierenden optimalen Bereich der Granularität aus Kapitel 4.2.1, der einen Kompromiss aus hohem Potential für Wiederverwendbarkeit und geringem Anpassungsaufwand bedingt, hat sich bei Gesamtfahrzeugmodellarchitekturen eine Unterteilung auf Fahrzeugkomponentenebene (wie z. B. dem Elektromotor oder der Hochvoltbatterie) bewährt [NETTER 2013c, S.5]. Basierend auf dieser vergleichbaren Granularität sind in Abbildung 4.8 die Funktionen den jeweiligen Domänen zugeordnet und mit Hilfe der jeweiligen Kategorien spezifiziert. Letztere reduzieren sich auf diejenigen Kategorien der Funktionen, die im Sinne des Simulationszwecks zur Ermittlung des Energieverbrauchs und der Fahrleistungen des A1 e-tron relevant sind. Somit sind die darin enthaltenen Funktionen der Kategorien „Elektrotechnik“, „Mechanik“ und „Steuerung/Regelungstechnik“ zur Darstellung des elektrischen Verbrauchs

und zur Ermittlung der Fahrleistung zwingend erforderlich, wohingegen andere Kategorien vernachlässigt werden. So ist in Abbildung 4.8 z. B. für die Funktion des Klimakompressors ausschließlich die Kategorie der „Elektrotechnik“ enthalten, wohingegen dessen Funktion aus der Kategorie „Thermodynamik“ nicht berücksichtigt wird. Die Verbindungen zwischen den dargestellten Funktionen repräsentieren jeweils eine Vielzahl an Signalen, die zwischen den Funktionen ausgetauscht werden [NETTER 2013c, S.6]. Zum besseren Verständnis wird der prinzipielle Signalfuss erläutert.

Die Variablendatei „Zyklus“ ist zusammen mit der Globalparameterdatei „Umgebung“ zu einer großen Datei „Randbedingungen“ zusammengefasst, die damit sämtliche Werte beherbergt, die für alle Simulationsmodelle gleichermaßen vorgegeben werden und nicht aus der Simulation gewonnen werden. Davon werden die globalen Parameter aus der Datei „Umgebung“ für den Co-Simulationsprozessschritt „Parameterization“ aus Kapitel 3.3.2 benötigt, um sämtliche Simulationsmodelle mit identischen Werten zu initialisieren. Die Variablen aus der Datei „Zyklus“ werden als Quelle für die Soll-Werte des zu absolvierenden Fahrzyklus benötigt.

Damit Fahrleistungen und Energieverbrauch analysiert werden können, müssen sämtliche leistungsrelevante Komponenten und Energiewandler im Fahrzeugmodell enthalten sein. Dafür ist es sinnvoll, von der Domäne „Antrieb“ auszugehen, da dieser Domäne die wesentlichen fahrleistungsrelevanten Komponenten und Verbraucher bzw. Stromerzeuger zuzuordnen sind. Darin soll, wie im oberen abgegrenzten Bereich von Abbildung 4.8 dargestellt, die mechanische Energie des Elektromotors, in der Abbildung als „EModul“ bezeichnet, über das Vorderachsdifferential „DiffVA“ an die „Räder“ geleitet werden. Die Funktion der Längsdynamik „FzgLängsD“ soll daraus auf Basis der physikalischen Gesetzmäßigkeiten mit den Fahrwiderstandskräften unter anderem die Ist-Geschwindigkeit berechnen. Die dafür benötigte Steigung wird von der Variablendatei „Zyklus“ bereitgestellt. Aus dieser stammt auch die Soll-Geschwindigkeit, aus deren Abgleich mit der Ist-Geschwindigkeit in der Antriebsregelung „AntriebsReg“ die Fahrpedalstellungen ermittelt werden, die zur Einhaltung der Soll-Geschwindigkeit benötigt werden. In Abhängigkeit davon werden entweder das „EModul“ oder die „Bremsen“ angesprochen. Der Elektromotor soll die Informationen über benötigte bzw. gelieferte, also rekuperierte Energie an die Funktion der Hochvolt (HV)-Batterie „BattHV“ liefern, die Bestandteil der Domäne „Elektrik/Elektronik“ im mittleren abgegrenzten Bereich in Abbildung 4.8

ist. In Abhängigkeit von dem Batteriefüllstand soll die für die Funktion des *Range Extenders* „RangeEx“ aus der Kategorie „Steuerung/Regelungstechnik“ vorgesehene Betriebsstrategie die Zu- bzw. Abschaltung der Funktion „RangeEx“ aus der Kategorie „Elektrotechnik“ ermitteln. Die dabei gegebenenfalls gewonnene Energie soll an die Hochvoltbatterie zurückgeführt werden. Die Domäne „Aufbau“ aus dem unteren abgegrenzten Bereich in Abbildung 4.8 beherbergt abschließend sämtliche Nebenverbraucher. Dabei sollen die beiden HV-Verbraucher, die *Positive Temperature Coefficient* (PTC)³-Heizung, in der Abbildung als „PTC“ bezeichnet, und der Klimakompressor („KlimaKo“) ihre Energie direkt aus der Hochvoltbatterie beziehen. Die restlichen Niedervolt (NV)-Verbraucher sollen ihre Energie aus dem ebenfalls der Domäne „Elektrik/Elektronik“ zugeordneten *Direct Current/Direct Current* (DC/DC)⁴-Wandler („DC-Wandler“) erhalten. Als Beispiel für die Darstellung einer Funktion in dem Architekturwerkzeug SystemDesk ist in Abbildung 4.9 das Architekturlayout der Hochvoltbatterie aus der Kategorie „Elektrotechnik“ während des Co-Simulationsprozessschritts „*System Design*“ veranschaulicht. Dabei handelt es sich um ein maximales Setting aus dem MMU, wovon nicht alle für das projektspezifische MU benötigt werden.

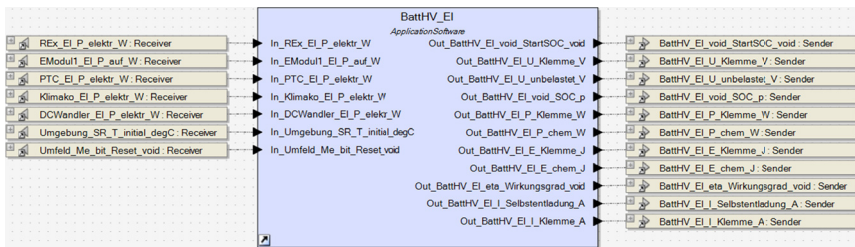


Abb. 4.9: Architekturlayout der Hochvoltbatterie (SystemDesk-Screenshot)

Wie anhand der Fahrzeugmodellarchitektur in Abbildung 4.8 erläutert und in den Eingangssignalen in der linken Hälfte in Abbildung 4.9 dargestellt,

³ *Positive Temperature Coefficient Thermistor*: keramischer Halbleiter mit positivem Temperaturkoeffizienten, also Material, dessen Leitfähigkeit mit steigenden Temperaturen sinkt und damit selbstregulierend agiert. Wird im Kraftfahrzeugbereich als Zusatzheizung verwendet

⁴ DC/DC-Wandler: Gleichspannungswandler, also elektrische Schaltung, die Gleichspannung in Gleichspannung mit höherem, niedrigerem oder invertiertem Spannungsniveau umwandelt

erhält die Funktion Information über die jeweils fließende elektrische Leistung des *Range Extenders* „RangeEx“, dem Elektromotor „EModul“, der PTC-Heizung „PTC“, dem Klimakompressor „KlimaKo“ und dem DC/DC-Wandler „DCWandler“. Diese Funktionen sind jeweils der Kategorie „Elektrotechnik“ zuzuordnen. Das Vorzeichen des übergebenen Wertes definiert, ob es sich dabei um gelieferte oder angeforderte Energie handelt. Bei dem vorletzten Eingangssignal handelt es sich um die Initialisierungstemperatur, die aus der Parameterdatei „Umgebung“ geliefert werden soll. Das letzte Eingangssignal beherbergt ein *Reset-Bit*, dessen Bedeutung im folgenden Teilkapitel erläutert wird. Aus den Interaktionsbeziehungen mit den anderen Funktionen des Architekturentwurfs ergeben sich diverse Anforderungen an die Funktion „BattHV“. So wird z. B., wie in der rechten Hälfte in Abbildung 4.9 dargestellt, vom Elektromotor „EModul“ die Klemmenspannung in Volt und sowohl von den beiden Funktionen „RangeEx“ und „AntriebsReg“ der *State of Charge* (SoC)⁵ in Prozent gefordert. Generell beschränken sich in dem Beispiel die Signale auf Inhalte einer Funktion aus der Kategorie „Elektrotechnik“, wohingegen andere Signale der Komposition „BattHV“ in anderen Funktionen berechnet werden können. So kann z. B. die sich im Betrieb einstellende Batterietemperatur mit Hilfe der Funktion „BattHV“ aus der Kategorie „Thermodynamik“ berechnet werden.

Skalierung der Gesamtfahrzeugmodellarchitektur

Um die Skalierbarkeit des Modularisierungsansatzes, der hierarchischen Strukturelemente und der Signal- und Schnittstellendefinition bewerten zu können, wird die Gesamtfahrzeugmodellarchitektur um weitere Teilsysteme erweitert. Als Evaluationsbeispiel soll eine Auslegung, Applikation und Absicherung der Betriebsstrategie für das Zu- und Abschalten des *Range Extenders* unter realitätsnahen Bedingungen ermöglicht werden. Zu diesem Zweck werden für das Lastprofil nicht mehr Soll-Werte eines normierten Fahrzyklus aus der Variablendatei „Zyklus“ verwendet, sondern Daten, die denen einer realen Testfahrt entsprechen. Um von den Vorteilen einer virtuellen Entwicklung profitieren zu können, werden dafür reale Strecken in einem virtuellen Umfeld reproduziert und von einem virtuellen Fahrer absolviert, um so die elektrischen Komponenten des virtuellen Prototyps computergestützt an reale Gegebenheiten anpassen zu können.

⁵ State of Charge: Ladezustand der Traktionsbatterie eines Elektrofahrzeugs

Dies ist durch eine sogenannte Simulation in einer geschlossenen Schleife (engl. *Closed-Loop Simulation*) möglich, da dort durch eine direkte Interaktion sowohl Einfluss als auch Auswirkung einer Modifikation an einer Fahrzeugkomponente bei verschiedenen Fahrertypen und unterschiedlichen Strecken untersucht werden können. Für einen solchen Aufbau müssen die bei der Definition der Strukturelemente in Kapitel 4.2.1 eingeführten Kraftfahrzeugdomänen des Fahrzeugmodells, wie in Abbildung 4.10 dargestellt, um einen virtuellen Fahrer und ein virtuelles Umfeld erweitert werden.

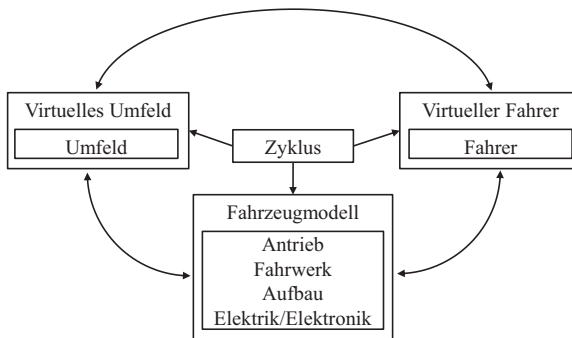


Abb. 4.10: Relevante Komponenten in einer *Closed-Loop Simulation*

In Ergänzung zu den bereits vorhandenen Funktionen des Gesamtfahrzeugmodells wird für die Interaktion mit dem virtuellen Umfeld und dem virtuellen Fahrer ein Modell für die Fahrzeugquerdynamik benötigt, welches innerhalb der zusätzlichen Domäne „Fahrwerk“ angeordnet ist. Diese ist in Ausbaustufe II dargestellt, die die vollständige Architektur aus Abbildung 4.8 umfasst. Dementsprechend müssen die bereits vorhandenen Schnittstellen im Gesamtfahrzeugmodell ergänzt und angepasst werden, um die Kommunikation der vorhandenen Funktionen mit dem Querdynamikmodell zu ermöglichen.

Für eine exakte Positionierung des zu simulierenden Fahrzeugs in dem virtuellen Umfeld benötigt die Umfoldsimulation analog Abbildung 4.11 aus der Kategorie „Mechanik“ von dem Gesamtfahrzeugmodell Informationen über die Soll-Positionen, Geschwindigkeiten und Beschleunigungen an sämtlichen Radaufstandspunkten und dessen Neigungen um die Längsachse, Querachse und Hochachse, welche von der Funktion „Fahrzeugquerdynamik“ bereitgestellt werden.

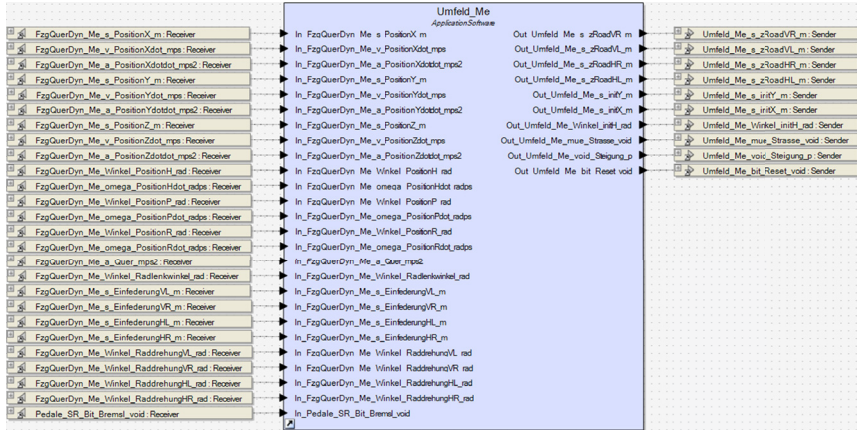


Abb. 4.11: Architekturlayout des virtuellen Umfelds (SystemDesk-Screenshot)

Für eine realitätsnahe Animation des Fahrzeugs werden darüber hinaus auch der Radlenkwinkel und sowohl die Raddrehung als auch die Einfederung an allen Rädern gefordert. Optional können dem virtuellen Umfeld auch Informationen über aktive Beleuchtungselemente zur Verfügung gestellt werden. Dieses liefert im Gegenzug auf Basis der zu absolvierenden Strecke die Ist-Positionen, Steigungen und Hafttreibungsbeiwerte an das Gesamtfahrzeugmodell zurück. Darüber hinaus wird aus der Funktion „virtuelles Umfeld“ ein Reset-Bit an sämtliche übrige Teilsysteme versendet, welches bei Grenzwertüberschreitungen einen zeitsynchronen Reset einer *Closed-Loop Simulation* ermöglicht.

Mit Hilfe des in Abbildung 4.12 dargestellten virtuellen Fahrers aus der Kategorie „Steuerung/Regelungstechnik“ werden die Soll-Werte für die Längs- und Querdynamik bestimmt.

Dieser erhält von der Funktion „Fahrzeugquerdynamik“ aus der Kategorie „Mechanik“ ebenfalls Informationen über die Positionen, Geschwindigkeiten und Beschleunigungen an sämtlichen Radaufstandspunkten. Darüber hinaus werden auch hier die Neigungen um die Längsachse, Querachse und Hochachse benötigt. Auf Basis der exakten Lokalisierung des Fahrzeugs kann der virtuelle Fahrer durch einen Abgleich mit der zu absolvierenden Strecke und einem voreingestellten Fahrer-Regelungstyp (z. B. *slow*, *normal*, *hasty*) die Soll-Beschleunigung und den Soll-Radlenkwinkel bestimmen.



Abb. 4.12: Architekturlayout des virtuellen Fahrers (SystemDesk-Screenshot)

Diese können in entsprechende Fahr- bzw. Bremspedalstellungen und Soll-Lenkwinkel umgerechnet und dem Fahrzeugmodell bereitgestellt werden.

4.4.2 Absicherung einer Gesamtfahrzeugmodellarchitektur

Zur Absicherung der erstellten Gesamtfahrzeugmodellarchitektur werden im Folgenden die in Kapitel 4.3 vorgestellten Maßnahmen angewendet. Dazu wird einerseits überprüft, ob die Schnittstellen der darin enthaltenen Funktionen zueinander passen und andererseits diejenigen Funktionsgruppen identifiziert, welche mit Hilfe aufgezeichneter Messdaten theoretisch validiert werden könnten.

Absicherung auf Basis der Gesamtfahrzeugmodellarchitektur

Bei der Erstellung einer Gesamtfahrzeugmodellarchitektur handelt es sich um einen generischen Prozess, bei dem fehlende Funktionsschnittstellen sukzessive in den Funktionen im MMU ergänzt bzw. Funktionen und Schnittstellen bei der Ableitung und Anpassung eines projektspezifischen MUs aus dem MMU entfernt werden. Dementsprechend wichtig ist die Absicherung der Schnittstellen in dem Schritt „*System Design*“. Um den formalen Nachweis der Korrektheit der Struktur zu erbringen, wird gemäß der Beschreibung zur Absicherung der Gesamtsystemarchitektur aus Kapitel 4.3.1 überprüft, ob innerhalb des MUs alle Eingänge mit Ausgängen anderer Funktionen verknüpft sind. Ist dies nicht der Fall, müssen die Eingänge zumindest über eine Parameterdatei, wie den in dem Evaluationsbeispiel verwendeten Randbedingungen, mit Daten versorgt werden.

Existiert nicht für jede Ausgangsschnittstelle eine dazugehörige Eingangsschnittstelle, wird vom *Signal Manager* eine Warnung ausgegeben und die betroffenen überflüssigen Signale werden angezeigt. Ein aus einer solchen Gesamtsystemarchitektur erzeugtes FMU wäre auf Kosten der Simulationsdauer noch simulierbar, da bei einer Simulation die überflüssigen Ausgangsgrößen zwar berechnet, aber nicht abgerufen werden. Wird jedoch nicht jede Eingangsgröße von einer anderen Ausgangsgröße mit Daten versorgt, wäre das System nicht mehr simulierbar, da die für eine Berechnung innerhalb der betroffenen Funktion benötigten Eingangsgrößen fehlen würden. In diesem Fall wird ein Fehler ausgegeben. Dementsprechend müssen weitere Signale aus bereits bestehenden Funktionen bereitgestellt oder gegebenenfalls weitere Funktionen, die die benötigten Inhalte liefern können, in die Gesamtsystemarchitektur eingebunden werden. Ist die Struktur konsistent, kann das MU in eine *Mock-Up*-Beschreibungsdatei und dabei auch jede Funktion in eine *Abstract-Model*-Beschreibungsdatei exportiert werden. Deren Aufbau wird im Anhang in Kapitel C.2 exemplarisch anhand des hier verwendeten Evaluationsbeispiels erläutert.

Absicherung der Simulationsmodelle

Um später die in die Gesamtfahrzeugmodellarchitektur implementierten Simulationsmodelle absichern zu können, wird geprüft, ob für alle Schnittstellen Daten verfügbar sind. Die dafür verwendeten Realdaten stammen aus realen A1 e-tron Prototypen, welche 2011 im Rahmen eines Pilotprojekts mit 20 Fahrzeugen über einen Zeitraum von zwei Monaten von Kunden im Großraum München bewegt worden sind. Zur Erfassung der Daten wurden die Fahrzeuge mit Datenloggern ausgestattet. Um die Quantifizierung unabhängig von dem Simulationswerkzeug, in dem die jeweiligen Simulationsmodelle implementiert sind, durchführen zu können, wurden die Messdaten aus dem proprietären Loggerformat in das allgemein verarbeitbare *Comma-Separated-Value* (CSV)-Format konvertiert. Dabei handelt es sich um ein Freitextformat, welches den Austausch von Daten über eine Vielzahl an Simulatoren unterstützt [BRAY 2008].

Zur Überprüfung wird der in Kapitel 4.3.2 eingeführte Algorithmus zur Absicherung von Simulationsmodellen eingesetzt. Erwartungsgemäß stehen nicht für alle Signale Messdaten aus dem Flottenversuch zur Verfügung, weswegen einzelne Funktionen mit fehlenden Schnittstelleninformationen

zu Funktionsgruppen aggregiert werden [BUTTERER 2012, S.73]. Die Ergebnisse der Gruppierung sind in Abbildung 4.13 dargestellt.

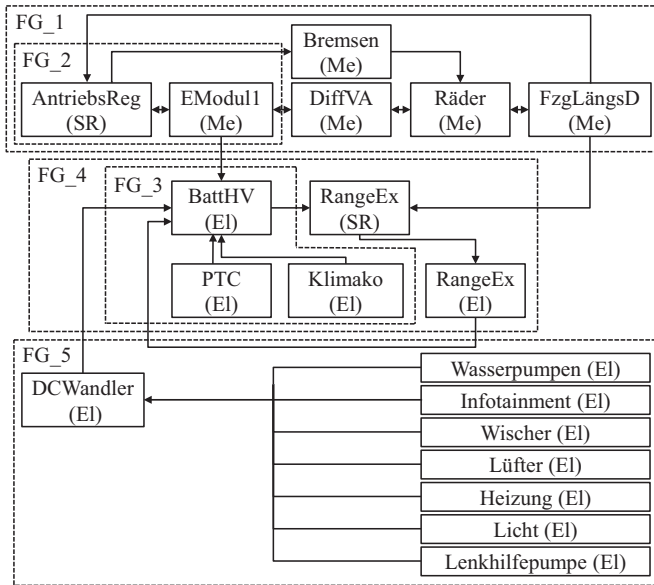


Abb. 4.13: Unterteilung in Funktionsgruppen nach [NETTER 2013a, S.46]

Bei der Analyse der Funktion „Bremsen“ wurde ermittelt, dass diese mit den Funktionen „AntriebsReg“, „EModul“, „DiffVA“, „Räder“ und „FzgLängsD“ gekoppelt werden muss, um validiert werden zu können. Aufgrund dessen wurden alle genannten Funktionen zur Funktionsgruppe „FG_1“ zusammengefasst. Aus der Untersuchung von den Funktionen „AntriebsReg“ resultierte die Notwendigkeit der Kopplung mit der Funktion „EModul“ zur Funktionsgruppe „FG_2“. Dementsprechend bedingt die Analyse der Funktion „EModul“ umgekehrt die Kopplung mit der Funktion „AntriebsReg“. Durch die sukzessive Untersuchungen sämtlicher Funktionen folgte, dass lediglich die Funktionen „RangeEx“ aus der Kategorie „Steuerung/Regelungstechnik“ und „RangeEx“ aus der Kategorie „Elektrotechnik“ direkt validierbar sind und somit keine weiteren Partner für eine Quantifizierung benötigen. Alle weiteren unter Einsatz des Suchalgorithmus identifizierten Funktionsgruppen sind ebenfalls in Abbildung 4.13 eingezeichnet [NETTER 2013a, S.45f].

5 Quantifizierung der Simulationsmodellkomplexität

„The choice of the best model is often viewed as the problem of choosing the appropriate level of detail, and this is considered one of the most difficult aspects of the modelling process, and one which has a major effect on the successfulness of the project“ [BROOKS 1996, S.2].

Nachdem in Kapitel 4 durch die Definition einer Gesamtsystemarchitektur die Grundlagen für ein FMU geschaffen worden sind, können die für die Simulation benötigten Simulationsmodelle in die nun festgelegte Struktur eingebunden werden. Dazu werden, gemäß dem Co-Simulationsprozess aus Kapitel 3.3.2, im Prozessschritt „*Model Design*“ die Funktionen und deren Schnittstelleninformationen aus den *Abstract-Model*-Beschreibungsdateien dem für deren Implementierung am besten geeigneten Simulator zugewiesen. In die daraus resultierenden *Modeltemplate*-Beschreibungsdateien können dann im Prozessschritt „*Modelling*“ die jeweils benötigten Funktionalitäten implementiert werden.

Die Modellerstellung kann analog den Grundlagen der Modellbildung aus Kapitel 2.2.1 nach unterschiedlichen Modellbildungsvarianten erfolgen, weswegen beliebig viele Modelle unterschiedlicher Detaillierung möglich sind [KÜBLER 2000, S.15]. Somit stellt sich im Prozessschritt „*Modelling*“ gemäß dem einführenden Zitat aus dem Beitrag „*Choosing the Best Model: Level of Detail, Complexity, and Model Performance*“ die Frage, wie detailliert und damit komplex die ausführbaren Modelle abgebildet werden müssen. Dieses Problem, aus einer Vielzahl an konkurrierenden Nachbildungen von Datensätzen die bestmögliche auszuwählen, wird als „Modellselektion“ bezeichnet [PITT 2002, S.472]. Um anhand definierter Kennzahlen zwischen einfachen und komplexen Modellen unterscheiden zu können, müssen geeignete Attribute identifiziert werden [BROOKS 1996, S.1].

Zu diesem Zweck wird zunächst der Begriff der Komplexität definiert, um ein gemeinsames Verständnis für die weiteren Schritte zu schaffen. Basierend darauf werden Merkmale identifiziert und appliziert, die eine Quantifizierung der Simulationsmodellkomplexität ermöglichen. Das Potential der Attribute für eine Komplexitätsquantifizierung wird anhand verschieden komplexer Implementierungen zweier Teilsysteme evaluiert.

5.1 Definition der Simulationsmodellkomplexität

Gemäß Myung [MYUNG 1999, S.2] bezeichnet die Modellkomplexität die Flexibilität eines Modells, die es ihm gestattet, sich einem breiten Bereich von Datensätzen (z. B. Wahrscheinlichkeitsverteilungen von Beobachtungen, die von der Modellgleichung erzeugt werden) anzupassen. Dabei wird die Gesamtheit aller Datensätze, die bei einem Versuch beobachtet werden können, als „Datenraum“ (engl. *Data (Model) Space*) bezeichnet. Dieser ist in der oberen Hälfte von Abbildung 5.1 dargestellt.

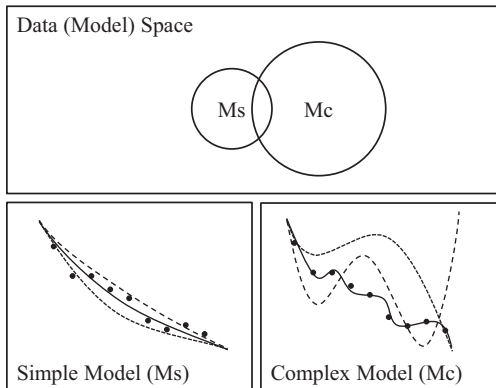


Abb. 5.1: Einfaches und komplexes Simulationsmodell im Datenraum nach [MYUNG 2000, S.192]

Jeder Punkt innerhalb dieses Datenraums entspricht einem festgelegten Datensatz. Verschiedene Modelle überspannen darin unterschiedliche Bereiche innerhalb dieses Datenraums, weswegen es zulässig ist, diesen Raum auch als die Gesamtheit sämtlicher Modelle zu interpretieren. Die Größe des von

einem Modell eingenommenen Bereichs korreliert mit dessen Komplexität [MYUNG 2000, S.192]. Einfache Modelle, in der oberen Hälfte von Abbildung 5.1 als „Ms“ bezeichnet, können nur einen relativ kleinen Bereich des Datenraums einnehmen, da sie aus einfachen spezifischen Strukturen mit wenigen Parametern bestehen. Dementsprechend ermöglicht eine Variation der Parameter, wie in der linken unteren Hälfte von Abbildung 5.1 dargestellt, lediglich eine geringfügige Anpassung an verschiedene Datenpunkte. Nur ähnliche Datensätze, die sich innerhalb des Raums befinden, der von dem einfachen Modell überspannt wird, können exakt reproduziert werden. Somit reicht bereits eine geringe Variation der Daten aus, um festzustellen, ob die Grenzen des von den Systemgleichungen darstellbaren Raums überschritten worden sind, weswegen einfache Modelle leicht validiert werden können. Im Gegensatz dazu umfassen komplexe Modelle, die in der oberen Hälfte von Abbildung 5.1 als „Mc“ bezeichnet werden, einen größeren Bereich im Datenraum. Diese komplexen Modelle bestehen in der Regel aus nichtlinearen Gleichungen, die es ihnen in Kombination mit einer Vielzahl an Parametern gestatten, sich unterschiedlichen Datensätzen anzupassen. Die daraus resultierende große Bandbreite möglicher Funktionsverläufe, mit deren Hilfe auch stärker variierende Datenpunkte repräsentiert werden können, sind in der rechten unteren Hälfte von Abbildung 5.1 veranschaulicht [MYUNG 2000, S.191].

5.2 Quantifizierung der Modellkomplexität

Gemäß der Definition von Systemen in Kapitel 2.1.2 wird das Systemverhalten durch die Systemstruktur und durch die Systemparameter definiert. Diese werden aus statistischer Sicht über die folgenden drei Aspekte der Komplexität festgelegt: die Anzahl der Parameter, deren funktionale Form, also die Art und Weise, in der die Parameter miteinander in der Modellgleichung verknüpft werden [MYUNG 1999, S.2], und der Parameterwertebereich, durch den die Anzahl der berücksichtigten Parameterkombinationen definiert wird [VANPAEMEL 2009, S.1]. Zu deren Quantifizierung können verschiedene Kennzahlen herangezogen werden, wie die parametrische Komplexität (engl. *Parametric Complexity* (PC)), die stochastische Komplexität (engl. *Stochastic Complexity* (SC)) und die geometrische Komplexität (engl. *Geometric Complexity* (GC)).

5.2.1 Parametrische Komplexität

Von diesen Kennzahlen ist die parametrische Komplexität (PC) die gebräuchlichste Größe, die ausschließlich die Anzahl der Parameter k berücksichtigt. Je mehr Parameter ein Simulationsmodell hat, desto höher ist dessen parametrische Komplexität. Damit ist die Korrelation zwischen der PC und dem Begriff der Komplexität leicht nachvollziehbar und analog Formel 5.1 auch leicht zu berechnen [VANPAEMEL 2009, S.2]:

$$PC = k \tag{5.1}$$

Allerdings wird mit der Anzahl der Parameter lediglich ein Aspekt der Komplexität berücksichtigt. Dies kann mitunter zu einer unzureichenden Quantifizierung führen, was an folgenden Gleichungen mit jeweils genau einem Parameter deutlich wird:

$$y = au \tag{5.2}$$

$$y = au^2 \tag{5.3}$$

$$y = au^3 \tag{5.4}$$

Für diese Beispiele resultiert aus der Berechnung mit Formel 5.1 eine identische Komplexität von $PC=1$. Allerdings verfügen die dargestellten Polynome unterschiedlicher Ordnung über eine unterschiedliche Flexibilität, sich gegebenen Daten anzupassen, weswegen, gemäß der Definition der Komplexität aus Kapitel 5.1, von unterschiedlichen Komplexitäten ausgegangen werden muss. Dementsprechend erfüllt eine Quantifizierung mit der PC zwar eine notwendige, aber keine hinreichende Bedingung, da ein Modell mit einer höheren Anzahl an Parametern zwar komplexer sein kann, aber nicht jedes komplexere Modell zwangsläufig eine höhere Anzahl an Parametern besitzt.

5.2.2 Stochastische Komplexität

Die stochastische Komplexität (SC) hat ihren Ursprung in der Muster- und Strukturerkennung. Dabei wird die SC als die kürzestmögliche Länge eines Codes definiert, der die regelmäßigen statistischen Eigenschaften der beobachteten Daten erfasst und gleichzeitig Redundanzen vermeidet

[RISSANEN 1986, S.1080]. Diese wird folgendermaßen formuliert [MYUNG 1999, S.3]:

$$SC = -\ln L(\hat{\theta}|z) + \frac{1}{2} \ln \left\| H(\hat{\theta}) \right\| \quad (5.5)$$

Darin entspricht der Ausdruck $L(\hat{\theta}|z)$ im ersten Term dem MLE *Estimate*, also der optimalen Parameterkombination, die die Likelihood-Funktion auf Basis der beobachteten Daten maximiert. $\left\| H(\hat{\theta}) \right\|$ repräsentiert im zweiten Term den Absolutbetrag der Determinante der Hesse-Matrix von der Minus-Log-Likelihood-Funktion für $\hat{\theta}$. In der Literatur wird der Absolutbetrag der Determinante nicht explizit beschrieben. Da aber die Determinante der Hesse-Matrix durchaus negative Werte annehmen kann und die Logarithmusfunktion im negativen Bereich nicht definiert ist, wird im Rahmen dieser Arbeit der Absolutbetrag für die Berechnung der SC gemäß Gleichung 5.5 verwendet.

Die Ermittlung der Likelihood-Funktion und die Bestimmung des MLE *Estimates*, welches beiden Termen aus Gleichung 5.5 zugrunde liegt, wurde bei der Vorstellung der Parameteridentifikation mit der Methode der maximalen Wahrscheinlichkeit in Kapitel 2.3.1 bereits beschrieben und wird im Anhang in Kapitel B.1 exemplarisch durchgeführt.

Gemäß Myung ist die Minus-Log-Likelihood-Funktion im ersten Term aus Gleichung 5.5 ein Maß für die fehlende Güte eines Modells, beobachtete Daten beschreiben zu können. In Ergänzung dazu stellt die Hesse-Matrix im zweiten Term ein Maß für die Modellkomplexität dar [MYUNG 1999, S.3]. Dieser wird auch von Grünwald in der Abkürzung „*COMP*“ (für *Complexity*) als Komplexitätsmaß verwendet [GRÜNWARD 2005]. Da im Rahmen dieser Arbeit nach einem geeigneten Maß zur Quantifizierung der Simulationsmodellkomplexität gesucht wird, wird im Folgenden ausschließlich der zweite Term berücksichtigt. Um dieses von der ursprünglichen Definition abzuheben, wird folgende Nomenklatur verwendet:

$$SC^* = \frac{1}{2} \ln \left\| H(\hat{\theta}) \right\| \quad (5.6)$$

Damit ermöglicht die SC^* über die Dimensionen der Hesse-Matrix nicht nur die Erfassung der Anzahl der Parameter, sondern ebenfalls eine Beurteilung der funktionellen Darstellungsform. Dementsprechend handelt es sich bei der SC^* im Vergleich zur PC um ein umfassenderes Quantifizierungsmerkmal.

Bei nicht-linearen Wahrscheinlichkeitsverteilungen mit mehreren Parametern kann das MLE *Estimate* nicht analytisch über die bei der Vorstellung der Methode der höchsten Wahrscheinlichkeit in Kapitel 2.3.1 dargestellte Form aus Gleichung 2.12 ermittelt werden [MYUNG 2003, S.93]. In solchen Fällen kann der mehrdimensionale Parameterraum unterteilt und die daraus resultierenden Segmente durchsucht werden. Allerdings hat dabei eine Quantifizierung mit dem SC* den Nachteil, dass für die numerische Berechnung des MLE *Estimate* Stützstellen und damit Schrittweiten für die Parameter vorgegeben werden müssen, weswegen $\hat{\theta}$ nur approximiert werden kann. Zusätzlich handelt es sich bei dieser Stelle lediglich um ein lokales Maximum, da eine numerische Identifikation eines globalen Maximums der Likelihood-Funktion bei unbeschränktem Parameterwertebereich nahezu unmöglich ist [MYUNG 2003, S.93f]. Da die Definition der SC* keine Grenzen für die Parameterwertebereiche vorsieht, besteht die Gefahr, dass bei erneuter Parametrierung des Modells mit variierenden Parameterwertebereichen eine andere optimale Parameterkombination gefunden wird, die die Likelihood-Funktion maximiert. Daraus resultiert dann auch ein anderer Wert für die Determinante der Hesse-Matrix und damit ein anderer Wert für die SC*, weswegen die Kennzahl variant unter Reparametrierung ist. Somit beeinflussen Schrittweite und Parameterwertebereich die stochastische Komplexität.

5.2.3 Geometrische Komplexität

Die geometrische Komplexität (GC) hat ihren Ursprung in der Differentialgeometrie und ist ein Maß für die Anzahl unterscheidbarer Wahrscheinlichkeitsverteilungen. Diese spannen, wie in Abbildung 5.2 dargestellt, einen hyperdimensionalen Raum auf, der als „Riemannsche Mannigfaltigkeit“ bezeichnet wird [MYUNG 1999, S.5].

Die Anzahl der Achsen korreliert darin mit der Anzahl der Parameter. Da in dem hier verwendeten Beispiel nur die beiden Parameter θ_i und θ_j in die Betrachtung einfließen, kann diese als zweidimensionale Fläche dargestellt werden. Innerhalb der Riemannschen Mannigfaltigkeit wird jede Wahrscheinlichkeitsverteilung durch einen Punkt veranschaulicht, welcher aus einem mathematischen Modell unter Verwendung der Parameter, die zu den jeweiligen Koordinaten gehören, resultiert. Exemplarisch sind in der Abbildung 5.2 drei solche Punkte veranschaulicht.

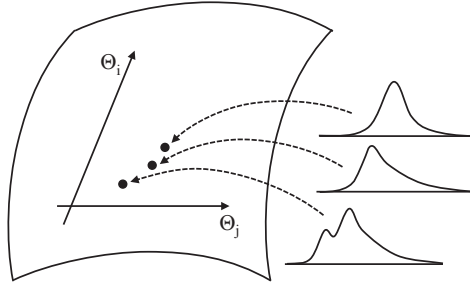


Abb. 5.2: Riemannsche Mannigfaltigkeit [PITT 2002, S.478]

Gemäß Myung [MYUNG 2003, S.91] wird ein Modell durch die Gesamtheit an Wahrscheinlichkeitsverteilungen definiert, die von den Modellparametern festgelegt wird. Dabei ist, der Definition der Komplexität aus Kapitel 5.1 folgend, ein Modell umso komplexer, je breiter der Bereich an Datensätzen ist, an die das Modell durch Parameteranpassung angepasst werden kann. Dementsprechend steigt die Komplexität eines Modells mit der Anzahl an Wahrscheinlichkeitsverteilungen die dafür erstellt werden können. Da diese Anzahl theoretisch unendlich groß ist, betrachtet Pitt lediglich unterscheidbare Wahrscheinlichkeitsverteilungen [PITT 2002, S.478]. Die Ähnlichkeit zweier Wahrscheinlichkeitsverteilungen korreliert mit deren Distanz s innerhalb der Riemannschen Mannigfaltigkeit, deren Entfernung ein Maß für die Unterscheidbarkeit darstellt. Dieser Abstand zwischen zwei Positionen ist wie folgt definiert:

$$ds^2 = \sum_{i,j=1}^k I_{ij}(\theta) d\theta_i d\theta_j \quad (5.7)$$

Darin entspricht k der Anzahl der Parameter $\theta = (\theta_1, \dots, \theta_k)$. $I_{ij}(\theta)$ ist die Fisher-Informationsmatrix (bzw. der Riemannsche Krümmungstensor), die gemäß Myung [MYUNG 1999, S.5] wie folgt definiert wird:

$$I_{ij}(\theta) = E \left[\frac{\partial \ln f(z|\theta)}{\partial \theta_i} \frac{\partial \ln f(z|\theta)}{\partial \theta_j} \right], \quad i, j \in \{1, \dots, k\} \quad (5.8)$$

In dieser Gleichung entspricht E dem Eigenwert der nach allen Parametern abgeleiteten Wahrscheinlichkeitsverteilung $f(z|\theta)$. Die Darstellung der

Fisher-Informationsmatrix ist auch über die in Kapitel 2.3.1 vorgestellte Likelihood-Funktion möglich:

$$I_{ij}(\theta) = E \left[\frac{\partial \ln L(\theta|z)}{\partial \theta_i} \frac{\partial \ln L(\theta|z)}{\partial \theta_j} \right] \quad (5.9)$$

Zwei Wahrscheinlichkeitsverteilungen können gemäß Balasubramanian nicht mehr voneinander unterschieden werden, wenn die Wahrscheinlichkeit, dass eine Wahrscheinlichkeitsverteilung für eine andere Wahrscheinlichkeitsverteilung gehalten wird, nahezu dem Wert eins entspricht [BALASUBRAMANIAN 1997, S.366]. Damit kann das Volumen eines Parameterraums, welcher nicht mehr voneinander unterscheidbare Wahrscheinlichkeitsverteilungen enthält, wie folgt definiert werden [MYUNG 1999, S.5]:

$$V(\theta) = \frac{\epsilon(2\pi)^{\frac{k}{2}}}{\|I(\theta)\|^{\frac{1}{2}}} \quad (5.10)$$

$\|I(\theta)\|$ repräsentiert darin den Absolutbetrag der Determinante der Fisher-Informationsmatrix, ϵ eine infinitesimal kleine positive Konstante und k die Anzahl der Parameter. Auch hier wird der Absolutbetrag der Determinante in der Literatur nicht explizit beschrieben. Da aber die Determinante der Fisher-Informationsmatrix analog zur Hesse-Matrix durchaus negative Werte annehmen kann und das Volumen eines Parameterraums keine negativen Werte annehmen darf, wird über den Absolutbetrag stets ein positiver Wert gewährleistet. Im Umkehrschluss zur Definition des Volumens repräsentiert das inverse Volumen $\frac{1}{V(\theta)}$ die Anzahl unterscheidbarer Wahrscheinlichkeitsverteilungen innerhalb eines Volumens im Parameterraum. Somit kann die Anzahl N sämtlicher unterscheidbarer Wahrscheinlichkeitsverteilungen, die durch ein Modell definiert werden, durch das Integral über den gesamten Parameterraum gewonnen werden [MYUNG 1999, S.5]:

$$N = \int \frac{1}{V(\theta)} = \frac{\int \|I(\theta)\|^{\frac{1}{2}} d\theta}{\epsilon(2\pi)^{\frac{k}{2}}} \quad (5.11)$$

Mit der von Myung getroffenen Festlegung für n Messpunkte $\|I(\theta)\| = n^k \|I_1(\theta)\|$ resultiert daraus unter Anwendung des Logarithmus die geometrische Komplexität GC [MYUNG 1999, S.5]:

$$GC = \ln \epsilon N = \frac{k}{2} \ln \frac{n}{2\pi} + \ln \int \|I_1(\theta)\|^{\frac{1}{2}} d\theta \quad (5.12)$$

Darin ermöglicht k im ersten Term eine Bewertung der Anzahl der Parameter. Mit $I(\theta)$ im zweiten Term wird deren funktionale Form erfasst [VAN-PAEMEL 2009, S.2]. Durch das Integral über $I(\theta)$ wird darüber hinaus auch der Parameterwertebereich als dritter Aspekt der Komplexität definiert, weswegen die GC im Gegensatz zur SC invariant unter Reparametrierung ist. Damit ist die GC in Relation zur PC und SC das umfassendste Quantifizierungsmerkmal, da sie alle drei Aspekte der Komplexität berücksichtigt. Durch den Absolutbetrag der Determinante der Fisher-Informationsmatrix wird zudem gewährleistet, dass stets der Logarithmus einer positiven Zahl gebildet wird, wodurch die gewählte Notation aus Gleichung 5.10 bestätigt wird. Allerdings sei an dieser Stelle darauf hingewiesen, dass beide Terme aufgrund der Logarithmusfunktionen für weniger als $n = 6$ Werte bzw. kleine Beträge von $I(\theta)$ durchaus negative Werte annehmen können. Aufgrund dessen wird der Absolutbetrag der GC als Maß für die Entfernung der Modellparameterkombination im Parameterraum vom Koordinatenursprung definiert, welcher für ein Modell der Komplexität null steht [GHORBEL 2014, S.77].

5.3 Applikation der Quantifizierungsmerkmale auf Simulationsmodelle

Nachdem nun verschiedene Verfahren zur Komplexitätsquantifizierung vorgestellt worden sind, sollen diese auf Simulationsmodelle angewendet werden. Wie bei der Problemstellung und Zielsetzung in Kapitel 1.1 erläutert, wird dazu auf Basis verfügbarer Messungen aus einer Klasse an Kandidatenmodellen das jeweils optimale Modell identifiziert und mit Hilfe der vorgestellten Verfahren die Simulationsmodellkomplexität bewertet. Dabei wird im ersten Teilkapitel zunächst von einer unabhängigen Messung ausgegangen, um die grundlegende Annahme einer Gaußschen Normalverteilung für die Messwerte einzuführen. Da es sich bei den Messwerten technischer Systeme jedoch weitestgehend um zeitlich voneinander abhängige Größen handelt, wird die getroffene Annahme in einem weiteren Beispiel um diesen Aspekt erweitert. In diesem Kontext wird auch der

Zusammenhang zwischen der Methode der höchsten Wahrscheinlichkeit und der Methode der kleinsten Quadrate hergeleitet, der eine analytische Quantifizierung im Fall normal verteilter Messwerte gestattet.

5.3.1 Komplexitätsquantifizierung bei unabhängiger Messung

Die Komplexitätsquantifizierung wird zunächst anhand des in Abbildung 5.3 dargestellten, linearen Modells mit konstanter Systemeingangsgröße $u = 1$ eingeführt. Da die Quantifizierung von SC* und GC die Likelihood-Funktion voraussetzt, wird für dieses Modell eine Parameteridentifikation mit der Methode der höchsten Wahrscheinlichkeit durchgeführt. Mit Hilfe des Parameters θ , der die Eingangsgröße u über die in Abbildung 5.3 dargestellte Systemstruktur mit der Ausgangsgröße y verknüpft, soll dabei die Ausgangsgröße des Modells an eine Reihe von Messwerten angepasst werden, welche von einem realen System erzeugt worden sind und in diesem Beispiel voneinander unabhängig sind. Ziel ist somit zunächst die Identifikation des Parameterwertes θ , der mit höchster Wahrscheinlichkeit die beobachteten Daten erzeugt hat, um mit diesem Zwischenergebnis anschließend die Modellkomplexität zu quantifizieren.

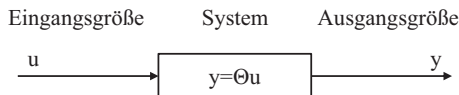


Abb. 5.3: Linearmodell für eine Quantifizierung bei unabhängiger Messung

Analog zu der im Anhang B.1 durchgeführten Parameteridentifikation am Beispiel eines Urnenversuchs muss zunächst eine Zufallsgröße festgelegt werden. Für das hier betrachtete Linearmodell wird das Messergebnis des realen Systems als Zufallsgröße Z mit den n Messpunkten z_i als n -fache Beobachtung interpretiert. Die $n = 15$ Werte sind in Tabelle 5.1 aufgelistet.

Tab. 5.1: Voneinander unabhängige Messdaten zur Parametrierung des Linearmodells

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Wert	5,00	5,02	4,99	5,01	5,00	5,03	4,98	5,02	5,01	5,03	5,00	4,99	5,01	5,02	4,96

Allerdings handelt es sich in diesem Beispiel im Gegensatz zum Urnenversuch nicht um einen Bernoulli-Versuch mit lediglich zwei möglichen Ergebnissen. Analog der Arbeiten von Myung, Pitt und Zhang wird infolge dessen die Annahme getroffen, dass die Messdaten um einen Erwartungswert normal verteilt sind (vergleiche dazu [MYUNG 1999], [MYUNG 2000], [MYUNG 2003] und [PITT 2002]). Für eine solche Gaußsche Normalverteilung kann laut Papula folgende Dichtefunktion angegeben werden [PAPULA 2009, S. 418]:

$$f(z_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z_i - \mu)^2}{2\sigma^2}}, \quad i \in \{1, \dots, n\} \quad (5.13)$$

Darin folgen die Messwerte z_i einer normal verteilten Wahrscheinlichkeitsverteilung mit der Standardabweichung σ um den Erwartungswert μ . Für diesen repräsentiert der Mittelwert den besten erwartungstreuen Schätzer. Somit könnte an dieser Stelle der Erwartungswert aus dem Mittelwert der erfassten Messwerte z_i ermittelt werden und damit für das Modell aus Abbildung 5.3 mit konstanter Systemeingangsgröße $u = 1$ der Wert des Parameters θ direkt abgelesen werden.

Wie eingangs beschrieben setzen die Quantifizierungsmerkmale jedoch eine Parameteridentifikation mit Hilfe der Methode der maximalen Wahrscheinlichkeit voraus, bei der der Parameter über die Likelihood-Funktion bestimmt wird. Allerdings muss dazu der Parameter θ in der Gaußschen Normalverteilung berücksichtigt werden.

Zu diesem Zweck wird als weitere Annahme die Ausgangsgröße und damit das Simulationsergebnis y als Erwartungswert interpretiert. Da die Ausgangsgröße gemäß Abbildung 5.3 vom Parameter θ abhängt, wird dieser dadurch in die Gaußsche Normalverteilung eingebunden. Unter dieser Annahme sind die Messdaten um das Simulationsergebnis normal verteilt, womit die Wahrscheinlichkeitsverteilung $f(z|\theta)$ für einen bestimmten Modellparameter $\theta \in [\theta_{min}, \theta_{max}]$ wie folgt dargestellt werden kann:

$$f(z_i|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z_i - \bar{y})^2}{2\sigma^2}}, \quad i \in \{1, \dots, n\} \quad (5.14)$$

Demnach folgen die Messwerte z_i nicht mehr einer normal verteilten Wahrscheinlichkeitsverteilung mit einer Breite σ um den Erwartungswert μ , sondern um den Mittelwert der Ausgangsgröße y . Da in diesem Beispiel das

System mit der konstanten Eingangsgröße $u = 1$ beaufschlagt wird, handelt es sich auch bei der Ausgangsgröße bei jeweils festen Parameter θ um eine Konstante. Die Standardabweichung σ entspricht im Allgemeinen der quadratischen Wurzel der Varianz und kann wie folgt abgeschätzt werden:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (z_i - \bar{y})^2} = \sqrt{\text{Var}(Z)} \quad (5.15)$$

Allerdings handelt es sich dabei, wie weiter unten im Text erläutert, lediglich um eine asymptotisch erwartungstreue Abschätzung der Standardabweichung, die die tatsächliche Standardabweichung unterschätzt. Um die daraus resultierende Abweichung zu kompensieren, wird die Gleichung für eine erwartungstreue Standardabweichung σ^* um den Korrekturfaktor c_4 erweitert:

$$\sigma^* = \frac{\sigma}{c_4} \quad (5.16)$$

Der aus dem Bereich des Qualitätsmanagements stammende Korrekturfaktor c_4 ist wie folgt definiert [SCLOVE 2005, S.2]:

$$c_4 = \sqrt{\frac{2}{n-1} \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})}} \quad (5.17)$$

Darin steht Γ für die Eulersche Gammafunktion, welche eine Fakultätsfunktion auf reelle nichtnegative Zahlen und komplexe Zahlen mit negativem Realteil erweitert.

Hintergrund für die fehlende Erwartungstreue ist, dass die in σ enthaltene Quadratwurzelfunktion eine nichtlineare und strikt konkave Funktion ist. Für solch eine Funktion $f(\gamma)$ gilt mit $\beta_j > 0 \forall j$ laut der Jensenschen Ungleichung an w Stellen einer Funktion:

$$f\left(\frac{\sum_{j=1}^w \beta_j \gamma_j}{\sum_{j=1}^w \beta_j}\right) \geq \frac{\sum_{j=1}^w \beta_j f(\gamma_j)}{\sum_{j=1}^w \beta_j} \quad (5.18)$$

Daraus folgt für die Quadratwurzelfunktion mit $\beta_j = 1$ und für $\gamma_j = \text{Var}_j(Z)$:

$$\sqrt{\left(\frac{\sum_{j=1}^w \text{Var}_j(Z)}{w}\right)} \geq \frac{\sum_{j=1}^w \sqrt{\text{Var}_j(Z)}}{w} \quad (5.19)$$

Dabei repräsentiert der linke Term die erwartungstreu Standardabweichung σ^* , die aus dem arithmetischen Mittelwert der Varianzen ermittelt wird, wohingegen der rechte Term dem Erwartungswert der abgeschätzten Standardabweichung, also der lediglich asymptotisch erwartungstreuen Abschätzung, entspricht. Mit μ als Erwartungswert gilt:

$$\sqrt{\mu(\text{Var}(Z))} \geq \mu(\sqrt{\text{Var}(Z)}) \quad (5.20)$$

Mit $\text{Var}(z) = \sigma^2$ kann die Gleichung wie folgt vereinfacht werden:

$$\sqrt{\mu(\sigma^2)} \geq \mu(\sqrt{\sigma^2}) \quad (5.21)$$

Daraus folgt, dass der Erwartungswert mehrerer abgeschätzter Standardabweichungen aus unterschiedlichen Beobachtungen derselben Zufallsgröße, welcher aus der Summe der Wurzeln der jeweiligen Varianzen ermittelt wird, immer kleiner oder gleich der erwartungstreuen Standardabweichung σ^* ist, die aus der Wurzel der Summen der jeweiligen Varianzen ermittelt wird [BOURBAKI 2015]. Diese Abweichung ist umso höher, je kleiner die Anzahl der Messpunkte n ist [SCLOVE 2005]. Dementsprechend liefert σ eine Unterschätzung der tatsächlichen Standardabweichung σ^* , weswegen im Folgenden die Berechnungen unter Berücksichtigung des Korrekturfaktors c_4 durchgeführt werden:

$$f(z_i|\theta) = \frac{1}{\sqrt{2\pi\sigma^{*2}}} e^{-\frac{(z_i - \bar{y})^2}{2\sigma^{*2}}}, \quad i \in \{1, \dots, n\} \quad (5.22)$$

Damit kann nun die für die Likelihood-Funktion erforderliche Wahrscheinlichkeitsverteilung $f(z|\theta)$ berechnet werden. Da in diesem Beispiel ein Modell mit nur einer Ausgangsgröße y betrachtet wird, sind die Funktionsverläufe der daraus resultierenden Wahrscheinlichkeitsverteilung zweidimensional, weswegen diese zudem graphisch veranschaulicht werden kann. Für die Berechnung und Darstellung wird MATLAB von The MathWorks verwendet, welche für die numerische Berechnung von Matrizen optimiert ist. Das Skript zur Berechnung der Quantifizierungsmerkmale für das hier betrachtete Linearmodell befindet sich im Anhang D.

Die Anzahl der bei der Ausführung zu berücksichtigenden Stützstellen resultiert in diesem Beispiel aus der Größe des Parameterwertebereichs $\theta_{max} - \theta_{min}$, der dazugehörigen Schrittweite o und der Menge k der dabei

zu identifizierenden Parameter θ . Im Allgemeinen hängt die Anzahl der zu berechnenden Kombinationen und damit der Berechnungsaufwand l auch noch von der Anzahl der zu verarbeitenden Messpunkte n ab, weswegen dieser wie folgt formalisiert werden kann:

$$l = n \prod_{i=1}^k \frac{\theta_{i_{max}} - \theta_{i_{min}}}{o_i} \quad (5.23)$$

Damit wird deutlich, dass der Berechnungsaufwand mit steigender Anzahl an Beobachtungen, steigender Parameteranzahl, steigender Parameterwertebereichsgröße und sinkender Schrittweite steigt. Um den Berechnungsaufwand zu reduzieren, sind Erfahrungswerte bei der Wahl des Parameterwertebereichs und eine geeignete Schrittweite hilfreich. In diesem Beispiel wurde ein Parameterwertebereich $\theta \in [4.96, 5.03]$ bei einer Schrittweite von $o = 0.0005$ verwendet, da dieser die Spannweite möglicher Simulationsergebnisse y vollständig abbildet und die relativ kleine Schrittweite eine genaue Schätzung des MLE *Estimates* ermöglicht. Daraus kann für dieses Beispiel mit $n = 15$ Messpunkten und $k = 1$ Parameter die Anzahl der zu berechnenden Kombinationen ermittelt werden:

$$l_{Linear.} = 15 \frac{5.03 - 4.96}{0.0005} = 2100 \quad (5.24)$$

In diesem Beispiel kann die Zufallsgröße im Gegensatz zu einem Münzwurf oder Würfelversuch beliebige Werte annehmen. Somit handelt es sich hier um eine wert-kontinuierliche WDF. Mit Hilfe von Formel 5.22 werden für jeden Parameterwert für alle Messwerte aus Tabelle 5.1 die dazugehörigen Werte der WDF berechnet. Diese sind in Abbildung 5.4 exemplarisch für den Parameterwert $\theta = 5$ veranschaulicht.

Darin repräsentieren die Kreise die Resultate von $f(z|\theta = 5)$ für die 7 unterscheidbaren Werte der $n = 15$ Messpunkte aus Tabelle 5.1. Diese sind für eine übersichtlichere Darstellung mit den Werten der dazugehörigen WDF hinterlegt, welche über eine Hilfsmessgröße mit fiktiven Messwerten im Wertebereich $z \in [4.8, 5.2]$ bei einer Schrittweite von $o_{Hilf} = 0.001$ erzeugt worden ist. Die Ergebnisse für sämtliche Parameter und den jeweils $n = 15$ Messpunkten werden in einer zweidimensionalen Matrix gespeichert, wovon eine Dimension den Parameterwerten und die andere den Messwerten zugeordnet werden kann. Im Gegensatz zur WF, wie sie bereits im

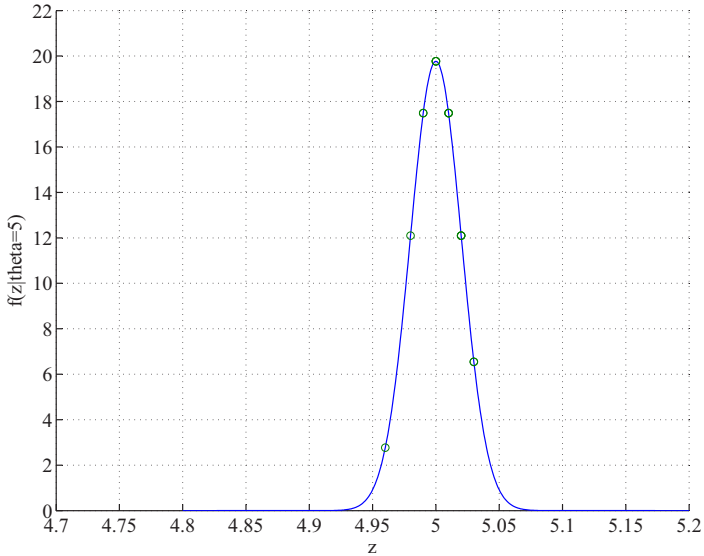


Abb. 5.4: WDF $f(z|\theta = 5)$ für das Linearmodell bei unabhängiger Messung

Anhang in Kapitel B.1.1 am Beispiel der Binomialverteilung dargestellt worden ist, können die Ergebnisse der hier dargestellten WDF Werte $\gg 1$ annehmen, da nur das Integral einer WDF zu 1 normiert ist. Die Verläufe der Wahrscheinlichkeitsdichtefunktionen für sämtliche Parameter aus dem betrachteten Wertebereich $\theta \in [4.96, 5.03]$ sind in dem Plot in Abbildung 5.5 dargestellt.

Da für jeden Wert des Parameters θ und damit unterschiedlichen Werten der Modellausgangsgröße y die erwartungstreue Standardabweichung σ^* gemäß Formel 5.16 neu abgeschätzt wird, variiert der Wert der Streuung, woraus unterschiedlich hohe und breite WDF resultieren. Dabei entspricht das Integral einer jeden WDF gemäß den Erläuterung zu Abbildung 5.4 dem Wert 1.

Die Wahrscheinlichkeit, die Beobachtung zu erhalten, wird über die Likelihood-Funktion beschrieben. Diese erhält man analog der Einführung der Methode der maximalen Wahrscheinlichkeit aus Kapitel 2.3.1 für jeden einzelnen Parameterwert aus dem Produkt seiner Einzelwahrscheinlichkeiten. An dieser Stelle der explizite Hinweis darauf, dass dabei für jeden Parameterwert diejenigen Werte der WDF multipliziert werden, die mit den

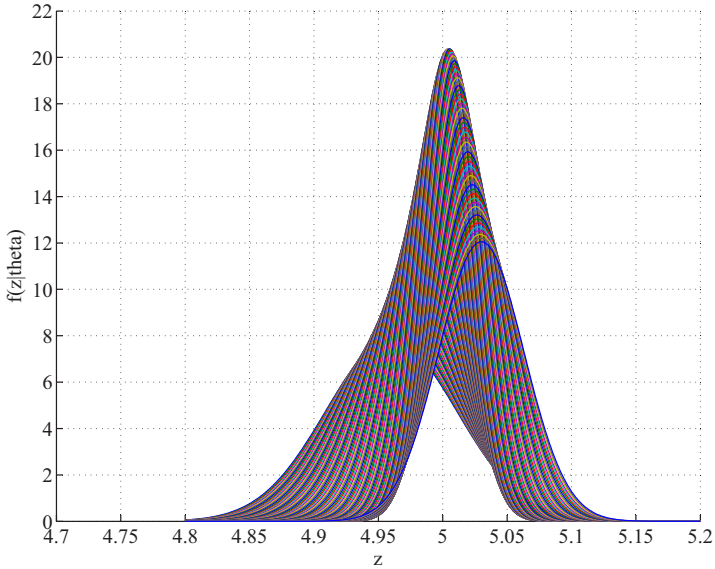


Abb. 5.5: WDF $f(z|\theta)$ für das Linearmodell bei unabhängiger Messung

jeweiligen Messwerten korrelieren, und nicht die Punkte der in Abbildung 5.5 dargestellten Hilfsfunktionen:

$$L(\theta|z) = \prod_{i=1}^n f(z_i|\theta) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z_i - \bar{z})^2}{2\sigma^2}} \quad (5.25)$$

Berechnet man die Likelihood-Funktion $L(\theta|z)$ für alle Werte des Modellparameters θ innerhalb des Wertebereichs und mit der festgelegten Schrittweite, resultieren daraus die Likelihood-Funktionswerte in Abhängigkeit von θ . Da dabei die mit den Messwerten korrelierenden Werte jeder WDF in einen Punkt aggregiert werden, sinkt die Dimension der Ergebnis-matrix um eins. Somit werden die Likelihood-Funktionswerte nicht mehr in einer zweidimensionalen Matrix, sondern in einem Vektor gespeichert. In der Ergebnisdarstellung in Abbildung 5.6 korrelieren die Abstände zwischen den einzelnen Produkten mit der gewählten Schrittweite des Parameterwertes.

Analog den theoretischen Grundlagen aus Kapitel 2.3.1 und dem bereits exemplarisch parametrisierten Urnenbeispiel aus dem Anhang B.1.2 muss der Parameter θ so gewählt werden, dass die Wahrscheinlichkeit,

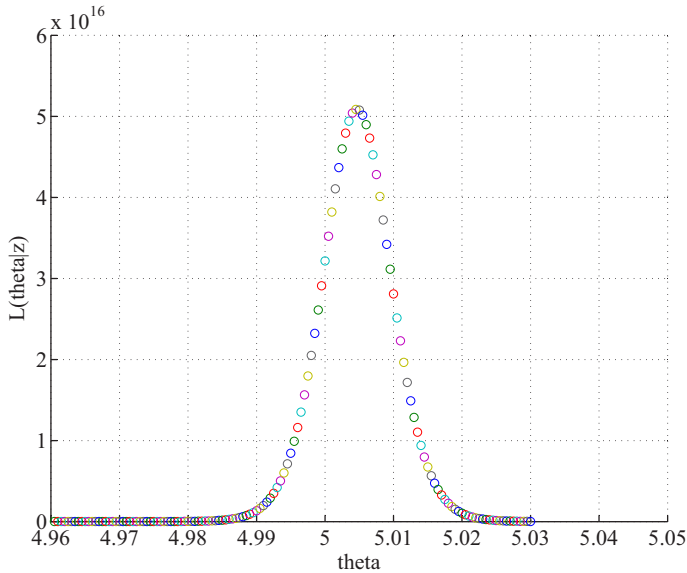


Abb. 5.6: Likelihood-Funktion $L(\theta|z)$ für das Linearmodell bei unabhängiger Messung

die beobachteten Daten zu erhalten, maximiert wird. Zu diesem Zweck wird nach dem Maximum der Likelihood-Funktion $L(\hat{\theta}|z)$ gesucht und der dazugehörige Parameterwert ermittelt. Für dieses Beispiel ergibt sich ein Wert von $5.0857 \cdot 10^{16}$ als maximaler Likelihood-Funktionswert. Analog zur WDF kann auch das MLE *Estimate* Werte $\gg 1$ annehmen, da dieses gemäß Formel 5.25 aus dem Produkt der Einzelwahrscheinlichkeiten resultiert, die selbst ebenfalls Werte $\gg 1$ annehmen können. Dieses MLE *Estimate* korreliert in diesem Fall mit $\hat{\theta} = 5.0045$ als optimalen Modellparameterwert. Bei diesem Wert handelt es sich in etwa um das arithmetische Mittel der Messwerte und damit, aufgrund der linearen Systemgleichung, um ein plausibles Ergebnis. Dieser Wert stützt damit die Zulässigkeit der eingangs für dieses Verfahren getroffenen Annahme. Im Rahmen dessen sei darauf hingewiesen, dass eine kleinere Schrittweite für die Werte des Parameters θ und damit ein kleinerer Abstand zwischen den Stützstellen der Likelihood-Funktion eine noch genauere Schätzung des MLE *Estimates* ermöglichen würde. Allerdings steigt damit auch der Rechenaufwand aufgrund der höheren Anzahl an zu berechnenden Kombinationen.

Nachdem nun die Modellidentifikation abgeschlossen ist und damit der Parameterwert identifiziert worden ist, der mit höchster Wahrscheinlichkeit die Beobachtung erzeugt hat, kann die Komplexitätsquantifizierung durchgeführt werden.

Da in diesem Beispiel lediglich ein Parameter $k = 1$ eingestellt werden muss, resultiert für die parametrische Komplexität auf Basis von Gleichung 5.1 ein Wert von $PC = 1$.

Zur Ermittlung der stochastischen Komplexität muss gemäß der allgemeinen Form aus Gleichung 5.6 zunächst die Hesse-Matrix gebildet werden. Dies geschieht durch zweimaliges Ableiten der Minus-Log-Likelihood-Funktion nach dem Parameter. Daraus wird dann analog Gleichung 2.13 die Hesse-Matrix gebildet, indem das Minimum der abgeleiteten Minus-Log-Likelihood-Funktion ermittelt wird. Da das Linearmodell nur von einem Parameter $k = 1$ abhängig ist, entspricht in diesem Fall die Hesse-Matrix direkt dieser Ableitung. Bei dem Absolutbetrag der Determinante der Hesse-Matrix handelt es sich damit um den Absolutbetrag der zweiten Ableitung, woraus $SC^* = 5.5226$ resultiert.

Die geometrische Komplexität kann für dieses Beispiel mit einem Parameter $k = 1$ und $n = 15$ Messwerten unter Berücksichtigung des Parameterwertebereichs auf Basis der allgemeinen Form aus Gleichung 5.12 ermittelt werden. Zur Berechnung der Fisher-Informationsmatrix muss die Likelihood-Funktion gemäß Formel 5.9 zunächst nach den Parametern abgeleitet werden und davon der Eigenwert gebildet werden. Da in diesem Fall das Linearmodell nur von einem Parameter abhängt, ist die Fisher-Informationsmatrix ein Skalar, weswegen der Eigenwert direkt erfasst werden kann. Zusammen mit dem ersten Term aus Gleichung 5.12 folgt ein Wert von $GC = 3.7472$.

5.3.2 Komplexitätsquantifizierung bei abhängiger Messung

Bei den Datensätzen, die für die Komplexitätsquantifizierung technischer Systeme herangezogen werden, handelt es sich weitestgehend um zeitlich voneinander abhängige Größen. Dementsprechend wird im Folgenden das im letzten Teilkapitel 5.3.1 eingeführte Beispiel eines Linearmodells bei unabhängiger Messung um den Fall einer abhängigen Messung erweitert. In Ergänzung dazu hängen die meisten Simulationsmodelle von mehreren

Eingangsgrößen und Parametern ab, weswegen nun auch diese Fälle mit berücksichtigt werden.

Zu diesem Zweck wird erneut das in Kapitel B.2 im Anhang im Rahmen der Regressionsanalyse mit der Methode der kleinsten Quadrate eingeführte Beispiel eines Chemiekraftwerks verwendet. Dieses ist in Abbildung 5.7 schematisch dargestellt. Da Rissanen dafür bereits Parameter identifiziert hat (siehe [RISSANEN 1987, S.96]), können auch hier die Ergebnisse abschließend plausibilisiert werden.

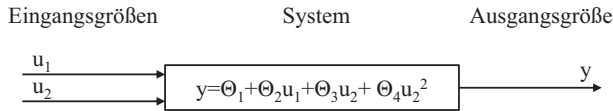


Abb. 5.7: Simulationsmodell für eine Quantifizierung bei voneinander abhängiger Messung

Um die dafür notwendige Vergleichbarkeit zu gewährleisten, wird, analog zur Annahme von Rissanen und dem im Anhang in Kapitel B.2 verwendeten Beispiel, von einer Systemgleichung der Form $y = \theta_1 + \theta_2 u_1 + \theta_3 u_2 + \theta_4 u_2^2$ ausgegangen. Darin entspricht die Ausgangsgröße y_i dem Simulationsergebnis, welches aus den variablen Eingangsgrößen $u_{1,i}$ und $u_{2,i}$ in Kombination mit den zu identifizierenden Modellparametern θ_1 bis θ_4 gebildet wird. Die dazugehörigen Werte für die Eingangsgrößen und das dabei tatsächlich erfasste Messergebnis z_i sind im Anhang in Tabelle B.3 aufgelistet.

Analog zu der im Teilkapitel 5.3.1 für das Linearmodell bei unabhängiger Messung getroffenen Annahme sollen auch hier die Messdaten um das Simulationsergebnis normal verteilt sein. Allerdings hängen in diesem Fall die Werte der Messung z_i zeitlich voneinander ab, da diese für ein System über mehrere Tage aufgezeichnet worden sind, weshalb die nachfolgenden Werte von den vorhergehenden beeinflusst wurden. Somit wird als weitere Annahme davon ausgegangen, dass es sich bei dem erfassten Messergebnis an jedem Messpunkt um einen um das Simulationsergebnis streuenden Wert handelt. Dementsprechend unterscheidet sich der Formalismus zur Berechnung der Wahrscheinlichkeitsverteilung $f(z|\theta)$ um den Ausdruck y_i im Exponenten von dem Fall aus Gleichung 5.22:

$$f(z_i|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z_i - y_i)^2}{2\sigma^2}}, \quad i \in \{1, \dots, n\} \quad (5.26)$$

Zur Berechnung der Likelihood-Funktion wird wieder für jedes von den vier Parametern θ_1 bis θ_4 abhängige Simulationsergebnis an jedem Messwert die dazugehörige WDF berechnet. Dabei wird bei der Berechnung der WDF für jede Parameterkombination eine fünfdimensionale Matrix erstellt, wovon eine Dimension den Messwerten und die anderen vier Dimensionen den Parametern zugeordnet werden können. Da bei einer Visualisierung jede Dimension einer separaten Achse entspricht, lässt sich die daraus resultierende hyperdimensionale Fläche nicht mehr graphisch darstellen.

Die Anzahl der zu ermittelnden Parameterkombinationen korreliert wieder mit der Anzahl der Messwerte, der Parameteranzahl, den Größen der gewählten Parameterwertebereiche und den dazugehörigen Schrittweiten. Um den Rechenaufwand zu begrenzen, dienen die von Rissanen bereits identifizierten Werte als Erfahrungswerte für die vier Parameter [RISSANEN 1987, S.96]:

$$(\hat{\theta}_{1_{Ris.}}, \hat{\theta}_{2_{Ris.}}, \hat{\theta}_{3_{Ris.}}, \hat{\theta}_{4_{Ris.}}) = (2.13, 0.67, -3.23, 0.093) \quad (5.27)$$

Damit können Parameterwertebereich und Schrittweite wie folgt abgeschätzt werden:

$$\begin{aligned} 2 \leq \theta_1 \leq 2.5, & \quad o_1 = 0.01 \\ 0.4 \leq \theta_2 \leq 0.9, & \quad o_2 = 0.01 \\ -3.5 \leq \theta_3 \leq -3, & \quad o_3 = 0.01 \\ 0.08 \leq \theta_4 \leq 0.1, & \quad o_4 = 0.001 \end{aligned} \quad (5.28)$$

Analog Gleichung 5.23 folgt damit für die Anzahl der zu ermittelnden Kombinationen:

$$\begin{aligned} l_{Sim.} &= n \prod_{i=1}^k \frac{\theta_{i_{max}} - \theta_{i_{min}}}{o_i} \\ &= 17 \frac{(2.5 - 2)(0.9 - 0.4)(-3 + 3.5)(0.1 - 0.08)}{(0.01)^3(0.001)} \\ &= 42.5 * 10^6 \end{aligned} \quad (5.29)$$

Der Rechenaufwand ist damit trotz der Einschränkung der Wertebereiche um ein Vielfaches höher als in dem Beispiel mit dem Linearmodell aus Kapitel 5.3.1. Berechnet man die Likelihood-Funktion $L(\theta|z)$ für alle Werte

der Modellparameter θ_1 bis θ_4 innerhalb der festgelegten Wertebereiche und mit den festgelegten Schrittweiten, resultieren daraus die Likelihood-Funktionswerte in Abhängigkeit von den Parametern. Da dabei wieder die Werte jeder WDF in einem Likelihood-Wert aggregiert werden, reduziert sich die Dimension gegenüber der WDF auf eine vierdimensionale Matrix. Analog zu den bereits behandelten Beispielen müssen die Parameter, die die Likelihood-Funktion maximieren, identifiziert werden. In diesem Beispiel muss das Maximum der hyperdimensionalen Fläche und darüber die dazugehörigen Parameterwerte gefunden werden. Aus diesem Vorgehen resultiert für den MLE *Estimate* ein Wert von $1.6888 * 10^{-11}$, der auf folgender optimaler Parameterwertekombination basiert:

$$(\hat{\theta}_{1_{MLE}}, \hat{\theta}_{2_{MLE}}, \hat{\theta}_{3_{MLE}}, \hat{\theta}_{4_{MLE}}) = (2.23, 0.67, -3.23, 0.093) \quad (5.30)$$

Dieser Wert entspricht damit in guter Näherung den in Gleichung 5.27 vorgestellten Ergebnissen nach Rissanen, weshalb an dieser Stelle festgehalten werden kann, dass die Annahme eines um ein Simulationsergebnis normal verteilten Messwertes auch für abhängige Messungen plausible Ergebnisse liefert. Ergänzend sei wieder darauf hingewiesen, dass sich die Genauigkeit der identifizierten Parameterwerte durch eine kleinere Schrittweite für ein exakteres MLE *Estimate* auf Kosten der Rechendauer erhöhen lässt.

Gemäß der Beschreibung im Rahmen der Methode der maximalen Wahrscheinlichkeit aus Kapitel 2.3.1 stößt eine solche Multiplikation einzelner Wahrscheinlichkeitsverteilungen allerdings bei einer größeren Anzahl an Parametern schnell an ihre Grenzen. Wie bei der Vorstellung der Methode der kleinsten Quadrate in Kapitel 2.3.2 erläutert, entspricht im Fall normal verteilter Messdaten die Methode der maximalen Wahrscheinlichkeit der Methode der kleinsten Quadrate. Dabei kann die Likelihood-Funktion wie folgt formuliert werden [KOLANOSKI 2008, S.92]:

$$L(\theta|z) = \prod_{i=1}^n f(z_i|\theta) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^{*2}}} e^{-\frac{(z_i - y_i)^2}{2\sigma^{*2}}} \quad (5.31)$$

Diese hat als Log-Likelihood-Funktion folgende Form:

$$\ln L(\theta|z) = -\sum_{i=1}^n \frac{(z_i - y_i)^2}{2\sigma^{*2}} - \ln \sqrt{(2\pi\sigma^{*2})} = -\frac{1}{2\sigma^{*2}}V - const. \quad (5.32)$$

Dieser Ausdruck ist äquivalent zu der in Gleichung 2.17 aufgestellten Kostenfunktion, weswegen die Likelihood-Funktion auch analytisch über die Methode der kleinsten Quadrate ermittelt werden kann. Da dabei die Likelihood-Funktion aus Formel 5.31 nicht erst erstellt werden muss, sondern die Gleichung für die Fisher-Informationsmatrix ein mal nach allen Parametern bzw. zwei mal für die Hesse-Matrix abgeleitet wird, kann erheblich Rechenzeit eingespart werden. Gemäß dem Beispiel für die Parameteridentifikation mit diesem Verfahren aus dem Anhang B.2 resultieren folgende Werte:

$$(\hat{\theta}_{1_{LSE}}, \hat{\theta}_{2_{LSE}}, \hat{\theta}_{3_{LSE}}, \hat{\theta}_{4_{LSE}}) = (2.125, 0.6726, -3.228, 0.09281) \quad (5.33)$$

Da für die getroffene Annahmen eine analytische Lösung sehr viel genauer und schneller ist, wird die Quantifizierung im Folgenden mit diesem Verfahren durchgeführt.

Nachdem nun die Modellidentifikation abgeschlossen ist und damit die Parameterwerte identifiziert worden sind, die mit höchster Wahrscheinlichkeit die Beobachtung erzeugt haben, kann analog zu dem Beispiel mit dem Linearmodell bei unabhängiger Messung die Komplexitätsquantifizierung über die Kennzahlen durchgeführt werden.

Da in diesem Beispiel vier Parameter $k = 4$ eingestellt werden müssen, resultiert für die parametrische Komplexität auf Basis von Gleichung 5.1 ein Wert von $PC = 4$.

Zur Ermittlung der stochastischen Komplexität muss gemäß der allgemeinen Form aus Gleichung 5.6 zunächst wieder die Hesse-Matrix gebildet werden. Dies geschieht durch zweimaliges Ableiten der Minus-Log-Likelihood-Funktion nach allen Parametern, woraus durch die erste Ableitung vier vierdimensionale Matrizen und durch die zweite Ableitung 16 vierdimensionale Matrizen resultieren. Daraus wird dann für die $k = 4$ Parameter analog Gleichung 2.13 eine symmetrische 4×4 -Hesse-Matrix gebildet, über die das Minimum identifiziert wird. Dazu werden die Werte der 16 vierdimensionalen Matrizen in die Hesse-Matrix übertragen, die sich an den gleichen Stellen wie das MLE *Estimate* in der vierdimensionalen Ergebnismatrix für die Likelihood-Funktionswerte befinden. Aus dem Absolutbetrag der Determinante der Hesse-Matrix und den übrigen Termen aus Formel 5.5 resultiert dann die stochastische Komplexität mit einem Wert von $SC^* = 12.21$.

Die geometrische Komplexität kann für dieses Beispiel mit vier Parametern $k = 4$ und $n = 17$ Messwerten unter Berücksichtigung des Parameterwertebereichs auf Basis der allgemeinen Form aus Gleichung 5.12 ermittelt werden. Die Berechnung der Fisher-Informationsmatrix erfolgt darin, analog dem Linearmodell aus Kapitel 5.3.1, durch einmaliges Ableiten der Likelihood-Funktion nach allen Parametern, woraus in diesem Beispiel vier vierdimensionale Matrizen resultieren. Allerdings muss im Gegensatz zu dem bereits behandelten eindimensionalen Fall bei dem hier betrachteten mehrdimensionalen Fall gemäß Formel 5.9 noch jeweils das Produkt aller Elemente untereinander gebildet werden, woraus 16 vierdimensionale Matrizen resultieren. Für die 16 Matrizen wird dann jeweils über die Determinante der Eigenwert berechnet, woraus 16 Werte resultieren. Diese spannen dann die ebenfalls symmetrische 4×4 -Fisher-Matrix auf, von der im Folgenden wiederum über den Absolutbetrag ihrer Determinante der Eigenwert gebildet werden muss. Zusammen mit dem ersten Term aus Gleichung 5.12 und der in Kapitel 5.2.3 getroffenen Definition, für die GC den Absolutbetrag zu nehmen, resultiert in diesem Beispiel ein Wert von $GC = 0.7933$ für die geometrische Komplexität.

Da die Ergebniswerte aufgrund der verwendeten Verfahren vom jeweils verwendeten Datensatz abhängen, können die Ergebnisse für SC und GC nicht mit Ergebnissen anderer Teilsysteme verglichen werden. Dementsprechend eignen sich diese beiden Quantifizierungsmerkmale ausschließlich für einen relativen Vergleich verschieden komplexer Varianten einer Funktion, sofern die Bewertung auf identischen Messwerten basiert.

5.4 Evaluation anhand mehrerer Teilsystems simulationsmodelle

Zur Evaluation der Quantifizierung der Simulationsmodellkomplexität wird das Verfahren zur Bestimmung der Kennzahlen nicht mehr, wie bisher, auf Basis einzelner voneinander losgelöster Simulationsmodelle durchgeführt, sondern für ein und dieselbe Funktion werden mehrere unterschiedliche Simulationsmodelle erstellt und bewertet. Dadurch kann im direkten Vergleich festgestellt werden, ob der Eindruck subjektiv unterschiedlich komplexer Simulationsmodelle durch die eingeführten objektiven Kennzahlen bestätigt werden kann. Zu diesem Zweck werden im Folgenden die Funktionen der

Hochvoltbatterie „BattHV“ und der Fahrzeuglängsdynamik „FzgLängsD“ verwendet, welche im Rahmen des Evaluationsbeispiels zur Strukturierung und Absicherung einer Gesamtfahrzeugmodellarchitektur in Kapitel 4.4 erläutert wurden und in Abbildung 4.8 dargestellt sind. Für diese werden jeweils mehrere unterschiedliche Simulationsmodelle erstellt und deren Komplexität anhand der eingeführten Kennzahlen bewertet.

5.4.1 Komplexitätsquantifizierung verschiedener Batteriemodelle

Motiviert durch die Entwicklung von *Hybrid Electric Vehicle* (HEV), *Plug-in Hybrid Electric Vehicle* (PHEV) und *Battery Electric Vehicle* (BEV) zur Reduzierung des Kraftstoffverbrauchs und der daraus resultierenden CO_2 -Emissionen ist die Hochvoltbatterie und deren Simulation Gegenstand aktueller Forschung. Dabei stehen insbesondere Lithium-Ionen-Batterien im Fokus. Für diesen Batterietyp werden im Folgenden sogenannte Ersatzschaltbild-Batteriemodelle quantifiziert, die auch für Energiemanagement- und Batteriemanagementsysteme verwendet werden [HU 2012, S.359]. Dabei handelt es sich um eine übliche Darstellungsform von Lithium-Ionen-Batterien, um die Leistungsfähigkeit dieser Batteriezellen bei verschiedenen Betriebs- und Lastzuständen vorherzusagen [ANDRE 2008, S.24]. Eine weitestgehend vollständige Zusammenfassung an Ersatzschaltbild-Batteriemodellen findet sich bei Hu (siehe dazu [HU 2012, S.362]). Die davon im Rahmen der Evaluation betrachteten Modelle sind in Abbildung 5.8 dargestellt.

Davon ist das R_i -Modell das einfachste Ersatzschaltbild, mit dessen Hilfe bereits wesentliche Aspekte des Batterieverhaltens erfasst werden. Es besteht, wie in Abbildung 5.8 links oben dargestellt, aus einer Gleichstromquelle der Leerlaufspannung (engl. *Open Circuit Voltage* (OCV)) U_0 und einem Innenwiderstand R_i . Mit steigender Stromstärke I fällt darin ein immer größerer Teil der Spannung am Innenwiderstand R_i ab, weswegen die nutzbare Klemmenspannung U_{Kl} sinkt [ANDRE 2008, S.24]. Mit Hilfe des R_i -Modells kann allerdings nicht das dynamische Verhalten eines zeitverzögerten Spannungsabfalls abgebildet werden, mit dem reale Batterien auf einen Belastungssprung reagieren. Dieser Zusammenhang ist in Abbildung 5.9 am Beispiel einer NiMH-Zelle dargestellt. Andere Batterietypen zeigen hierbei einen grundsätzlich gleichartigen Verlauf.

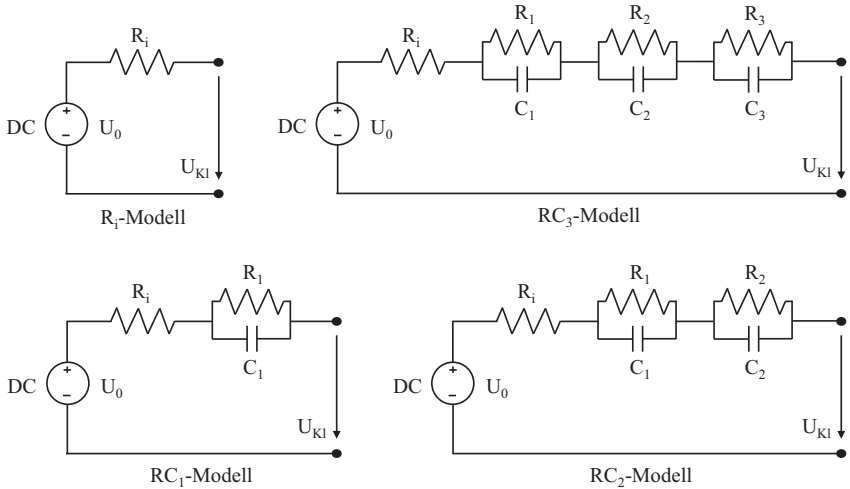


Abb. 5.8: Ersatzschaltbilder der für die Evaluation verwendeten Batteriemodelle

Gemäß dem Spannungsverlauf verursacht eine Belastung einen sofortigen Spannungsabfall durch den ohmschen Widerstand der Zelle. Durch Ladungsträger unterschiedlicher Polarisation im Grenzbereich Elektrolyt/Elektrode, welche eine sogenannte Doppelschicht-Kapazität bilden, verzögert sich der Spannungsabfall im weiteren Verlauf. Der durch Diffusion verzögerte Spannungsabfall wird durch die begrenzte Diffusionsgeschwindigkeit der Ladungsträger verursacht, die von den Elektrolyten an die Elektrode wandern müssen [ANDRE 2008, S.24f].

In der Elektrotechnik werden zeitabhängige Spannungsabfälle durch Widerstands-Kondensator-Parallelschaltungen dargestellt. Diese sogenannten RC-Elemente bestehen aus einer Parallelschaltung von einem Widerstand R und einer Kapazität C . In der einfachsten Ausprägung, dem in Abbildung 5.8 links unten dargestellten RC_1 -Modell, wird das R_i -Modell um genau ein RC-Element erweitert. Damit kann das RC_1 -Modell auch die verzögerte Spannungsreaktion auf einen Belastungssprung darstellen. Eine weitere Parallelschaltung in Reihe zur ersten Parallelschaltung ermöglicht eine separate Erfassung der Effekte, die durch den Ladungsdurchtritt der Doppelschicht und durch die Diffusion verursacht werden [ANDRE 2008, S.25]. Ein solches RC_2 -Modell ist rechts unten in Abbildung 5.8 dargestellt. Da sich die Doppelschichtkapazität an beiden Elektroden ausbildet und

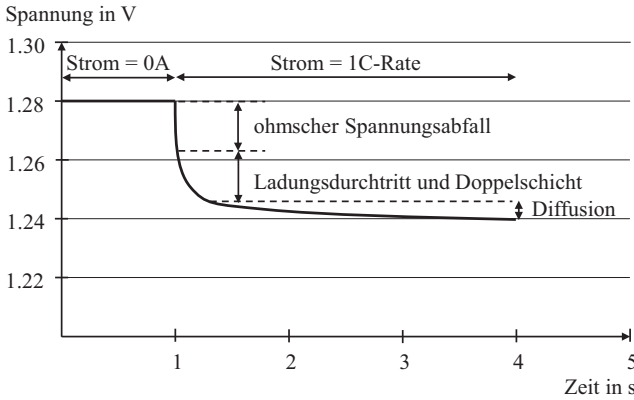


Abb. 5.9: Spannungsverlauf einer Batterie bei einem Belastungssprung nach [JOSSEN 2006, S.19]

die spezifischen Oberflächen beider Elektroden in der Regel unterschiedlich sind, ergeben sich für eine Zelle gegebenenfalls unterschiedliche Zeitkonstanten [JOSSEN 2006, S.19]. Dementsprechend können mit drei parallelen RC-Schaltungen die unterschiedlichen Stufen der verzögerten Spannungsreaktion noch exakter approximiert werden, wovon zwei RC-Elemente auf die Zeitkonstanten der Doppelschicht-Kapazität entfallen und das dritte die Diffusionsgeschwindigkeit berücksichtigt. Ein solches RC3-Modell ist in Abbildung 5.8 rechts oben dargestellt.

Gemäß der Definition der Simulationsmodellkomplexität aus Kapitel 5.1 verfügen die vier Ersatzschaltbild-Batteriemodelle über eine unterschiedliche Flexibilität, sich gegebenen Daten anzupassen, weswegen von unterschiedlichen Komplexitäten ausgegangen wird. Um diesen Eindruck einer subjektiv steigenden Komplexität vom R_i -Modell bis hin zum RC3-Modell objektivieren zu können, werden die Kennzahlen zur Quantifizierung der Modellkomplexität eingesetzt. Um dafür die Anzahl der für die Evaluation zu berücksichtigenden Parameter überschaubar zu halten, handelt es sich bei den hier analysierten vier Ersatzschaltbild-Batteriemodellen lediglich um Teilmodelle der in Abbildung 4.9 dargestellten Funktion „BattHV“.

Analog dem vorgestellten Verfahren werden zunächst diejenigen Parameter identifiziert, die mit höchster Wahrscheinlichkeit die beobachteten Daten erzeugt haben. Eingangsgröße für die Batteriemodelle ist dabei jeweils ein zeit-variabler Stromverlauf $I(t)$, mit dem über Formel 5.34 der SoC und in

Abhängigkeit davon über batteriespezifische Tabellen die Leerlaufspannung U_0 ermittelt wird.

$$SoC = SoC_{Start} - \frac{\int_0^t I(t) dt}{C_{max}} \quad (5.34)$$

Aus dem zeit-variablen Stromverlauf $I(t)$ und der Leerlaufspannung U_0 kann die zeit-variable Klemmenspannung $U_k(t)$ ermittelt werden. Die Parameter und die Formeln, die die Eingangsgröße und die Ausgangsgröße mit Hilfe der Parameter verknüpfen, sind in Tabelle 5.2 in Abhängigkeit von dem jeweils betrachteten Batteriemodell aufgelistet.

Tab. 5.2: Ergebnisse der nach ihrer Komplexität quantifizierten Batteriemodelle nach [GHORBEL 2014, S.68ff]

Typ	Formel	Parameter	PC	SC*	GC
R_i	$U_{K_{R_i}} = U_0 + R_i I$	R_i	1	6,276	4,551
RC_1	$U_{K_{RC_1}} = U_{K_{R_i}} + U_{R_1}$	R_i, R_1, C_1	3	9,639	22,230
RC_2	$U_{K_{RC_2}} = U_{K_{RC_1}} + U_{R_2}$	R_i, R_1, C_1, R_2, C_2	5	13,003	39,866
RC_3	$U_{K_{RC_3}} = U_{K_{RC_2}} + U_{R_3}$	$R_i, R_1, C_1, R_2, C_2, R_3, C_3$	7	16,360	57,457

Die für die Quantifizierung benötigten Daten für den Strom- und Spannungsverlauf stammen aus einem A1 e-tron-Prototyp aus dem in Kapitel 4.4.2 beschriebenen Pilotprojekt. Für das Evaluationsbeispiel wurde aus den digitalisierten zeit-diskreten Informationen der im Batteriesystem verbauten analogen Sensoren ein 70 Sekunden langer Abschnitt eines *Controller Area Network* (CAN)-Traces mit einer Abtastrate von 1,43 Hz verwendet. Dessen Verlauf ist in Abbildung 5.10 dargestellt.

Die Ergebnisse aller Quantifizierungen mit den Daten aus Abbildung 5.10 sind in Tabelle 5.2 eingetragen. Gemäß der subjektiven Einschätzung steigen sowohl die Werte für PC als auch SC* und GC vom R_i -Modell über das RC_1 -Modell und RC_2 -Modell bis hin zum RC_3 -Modell.

5.4.2 Komplexitätsquantifizierung verschiedener Längsdynamikmodelle

Als weiteres Beispiel wird im Folgenden das Fahrzeuglängsdynamikmodell „FzgLängsD“ betrachtet, welches für Fahrdynamikanalysen benötigt wird. Innerhalb dieser Komponente wird die Beschleunigungswiderstandskraft eines Fahrzeugs mit den Fahrwiderstandskräften kombiniert. Dabei setzt

sich die Gesamtfahrwiderstandskraft F_W nach Beitz und Feldhusen aus der Summe aus Rollwiderstandskraft F_R , Luftwiderstandskraft F_L und Steigungswiderstandskraft F_S zusammen [BEITZ 2001, S.Q3]:

$$F_W = F_R + F_L + F_S \quad (5.35)$$

Davon lässt sich die Rollwiderstandskraft als Produkt aus Rollwiderstandsbeiwert f_R , Fahrzeugmasse m_{Fzg} und Erdbeschleunigung g darstellen:

$$F_R = f_R m_{Fzg} g \quad (5.36)$$

Die Luftwiderstandskraft kann aus dem Produkt aus Luftwiderstandsbeiwert c_W , Fahrzeugstirnfläche A_{Fzg} , Luftdichte ρ und Fahrzeuggeschwindigkeit v_{Fzg} ermittelt werden:

$$F_L = c_W A_{Fzg} \rho \frac{v_{Fzg}^2}{2} \quad (5.37)$$

Die Steigungswiderstandskraft entspricht dem Anteil der Fahrzeuggewichtskraft, der längs der Steigung in Hangabtriebsrichtung wirkt:

$$F_S = m_{Fzg} g \sin \alpha \quad (5.38)$$

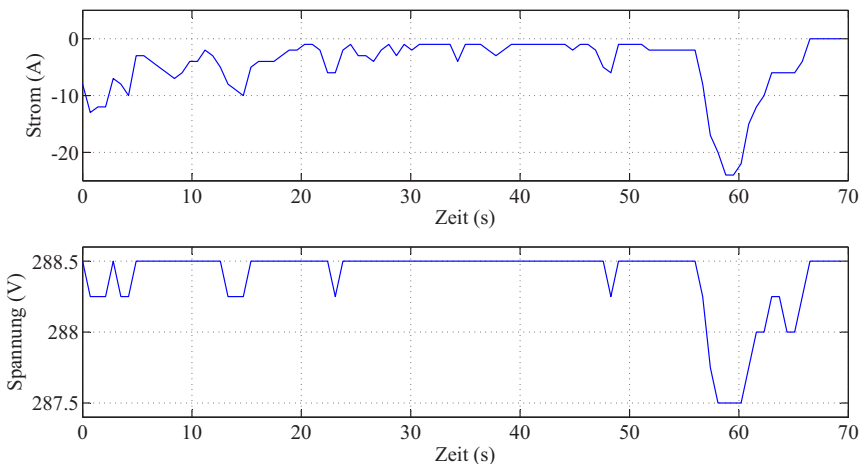


Abb. 5.10: Datensatz zur Batteriemodellquantifizierung [GHORBEL 2014, S.69]

Wird ein Fahrzeug mit der Beschleunigung a_{Fzg} beschleunigt oder verzögert, kommt die Beschleunigungswiderstandskraft F_B als weitere Komponente hinzu. Unter Vernachlässigung der Zuladung und des rotatorischen Anteils kann diese wie folgt vereinfacht werden:

$$F_B = m_{Fzg} a_{Fzg} \tag{5.39}$$

Damit kann die erforderliche Antriebskraft F_A folgendermaßen ermittelt werden:

$$\begin{aligned} F_A &= -F_W - F_B \\ m_{Fzg} a_{Fzg} &= -F_A - F_R - F_L - F_S \end{aligned} \tag{5.40}$$

Durch Umformung kann daraus die Fahrzeuggeschwindigkeit ermittelt werden:

$$v_{Fzg} = \int_0^t a_{Fzg} dt = \int_0^t \frac{-F_A - F_R - F_L - F_S}{m_{Fzg}} dt \tag{5.41}$$

Da die Luftwiderstandskraft im rechten Term der Gleichung 5.41 von der Geschwindigkeit abhängt, fließen die Resultate eines jeden Rechenschritts wieder in die Berechnung des darauf folgenden Rechenschrittes mit ein. Es handelt sich somit um eine rekursive Simulation, bei der Modellausgangsgrößen auch als Eingangsgrößen verwendet werden.

Bei den in Gleichung 5.36 bis 5.38 verwendeten Widerstandsbeiwerten handelt es sich um empirisch ermittelte Größen, die als Konstanten verwendet werden. Allerdings sind davon z. B. der Rollwiderstandsbeiwert f_R und der Luftwiderstandsbeiwert c_W genau genommen Variablen. Letzterer ändert sich z. B. bei variierender Reynolds-Zahl, welche wiederum von der Geschwindigkeit abhängt. Die in einem Windkanal ermittelten Luftwiderstandsbeiwerte für einen Audi A1 sind in Tabelle 5.3 aufgelistet.

Tab. 5.3: Luftwiderstandsbeiwert in Abhängigkeit von der Geschwindigkeit

v_{Fzg} [km/h]	79,1	99,7	121,0	139,1	161,0	181,3	201,1	221,2
c_W	0,3316	0,3316	0,3383	0,3371	0,3398	0,3391	0,3399	0,3415

Um also das Systemverhalten durch die Simulation noch exakter an die Realität anzunähern, können an Stelle eines festen Wertes für den c_W Variablen verwendet werden. Dafür kann der Luftwiderstandsbeiwert durch

ein Polynom beliebiger Ordnung approximiert werden, dessen Genauigkeit mit steigender Ordnung zunimmt. Für den hier betrachteten Fall werden Polynome bis zweiter Ordnung verwendet. Die daraus resultierenden drei Typen zur Darstellung des Luftwiderstandsbeiwertes sind in Tabelle 5.4 aufgelistet.

Tab. 5.4: Ergebnisse der nach ihrer Komplexität quantifizierten Längsdynamikmodelle nach [GHORBEL 2014, S.75ff]

Typ	Formel	Parameter	PC	SC*	GC
$cW_{konst.}$	$cW_{konst.} = cW$	cW	1	4,090	4,407
$cW_{lin.}$	$cW_{lin.} = av + b$	a, b	2	11,341	8,238
$cW_{quadr.}$	$cW_{quadr.} = cv^2 + dv + e$	c, d, e	3	21,502	20,921

Gemäß der Definition der Simulationsmodellkomplexität aus Kapitel 5.1 verfügen die drei Typen über eine unterschiedliche Flexibilität, sich gegebenen Daten anzupassen, weswegen wieder von unterschiedlichen Komplexitäten ausgegangen wird. Um diesen Eindruck einer subjektiv steigenden Komplexität vom konstanten $cW_{konst.}$ -Parameter über das lineare $cW_{lin.}$ -Modell bis hin zum quadratischen $cW_{quadr.}$ -Modell objektivieren zu können, werden wieder die bereits eingeführten Kennzahlen eingesetzt. Auch hier handelt es sich für eine überschaubare Anzahl an Parametern nur um ein Teilmodell der Funktion „FzgLängsD“.

Analog den im Kapitel 5.2 vorgestellten Verfahren müssen zunächst wieder diejenigen Parameter identifiziert werden, die mit höchster Wahrscheinlichkeit die beobachteten Daten erzeugt haben. Die dafür verwendeten Daten stammen auch in diesem Beispiel aus dem A1 e-tron-Prototyp, welcher im Rahmen des Pilotprojekts in München bewegt worden ist. Allerdings standen aus dem Flottenversuch für eine direkte Quantifizierung des Luftwiderstandsbeiwert-Modells keine Realdaten in Form von geschwindigkeits-abhängigen Luftwiderstandsbeiwerten zur Verfügung. Dementsprechend wurde dieses Modell mit dem in Kapitel 4.3.2 vorgestellten Algorithmus zur Gruppierung mit weiteren Modellen untersucht, um anschließend sämtliche Schnittstellen mit Realdaten beaufschlagen zu können. Als Ergebnis wurde eine Funktionsgruppe mit sämtlichen Kräften aus Gleichung 5.40 vorgeschlagen. Die Steigungswiderstandskraft konnte dafür mit Hilfe der Fahrzeugpositionen und der dabei abgefahrenen Straßentopologie ermittelt werden. Die Antriebskraft resultierte aus dem E-Maschinen-Modell. Die

Rollwiderstandskraft wurde als konstant angenommen. Für die Luftwiderstandskraft konnte das enthaltene Luftwiderstandsbeiwert-Modell mit der Geschwindigkeit beaufschlagt, sowie mit der Stirnfläche, der als ebenfalls als konstant angenommenen Luftdichte und dem halben Quadrat der Geschwindigkeit multipliziert werden. Das daraus resultierende Ergebnis konnte mit Hilfe von Gleichung 5.41 in die rekursive Berechnung der Geschwindigkeit eingefügt werden. Damit wurde als Simulationsergebnis nicht direkt der Luftwiderstandsbeiwert, sondern die Geschwindigkeit quantifiziert, wodurch dennoch eine indirekte Identifikation der in Tabelle 5.4 aufgeführten Parameter als Bestandteil des Modells möglich war. Die für die Quantifizierung nötigen Eingangsgrößen von Steigungswiderstandskraft F_S , Antriebskraft F_A und Geschwindigkeit v_{Fzg} sind in Abbildung 5.11 dargestellt. Für dieses Beispiel wurde ein 40 s langer CAN-Trace mit einer Abtastrate von 5 Hz verwendet.

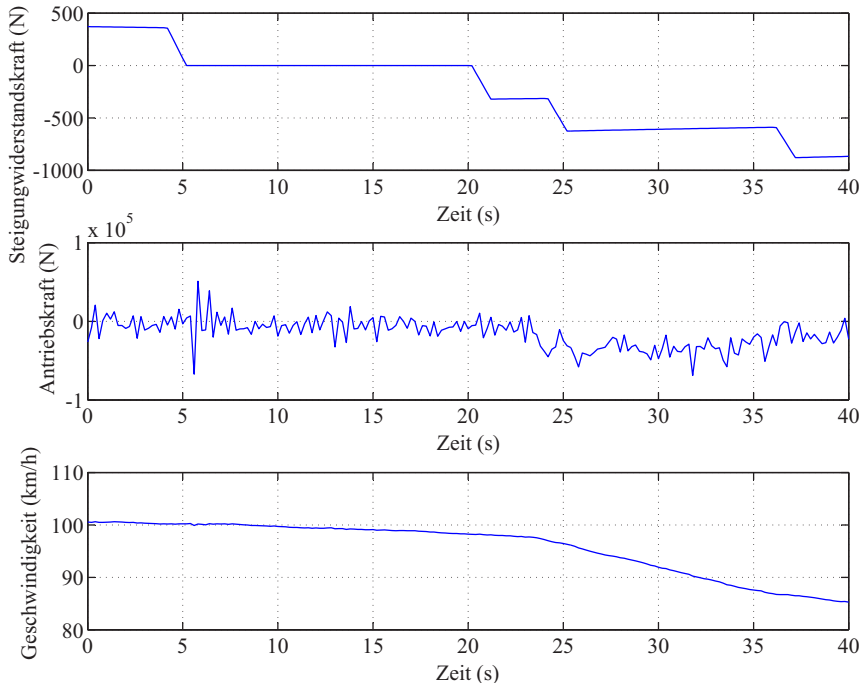


Abb. 5.11: Datensatz zur Längsdynamikmodellquantifizierung

Bei der Berechnung der GC für das linear bzw. quadratische Längsdynamikmodell resultierten jeweils negative Werte, für die dann gemäß der Definition aus Kapitel 5.2.3 der Absolutbetrag verwendet worden ist. Die Ergebnisse der Quantifizierung sind in Tabelle 5.4 aufgelistet. Analog zum Batteriemodell steigen darin die Werte der PC, SC* und GC von dem konstanten über das lineare bis hin zum quadratischen Längsdynamikmodell, was die getroffene Definition, für die GC den Absolutbetrag zu nehmen, stützt. Somit kann auch hier von einer steigenden Komplexität der Simulationsmodelle ausgegangen werden.

6 Quantifizierung der Simulationsergebnisgüte

„A model must be complex enough to capture the underlying regularity yet simple enough to avoid overfitting the data sample and thus losing generalizability“ [PITT 2002, S.474].

In Kapitel 5 wurde der Begriff der Komplexität vor dem Hintergrund der in dieser Arbeit betrachteten Simulationsmodelle definiert. Zusätzlich wurden Attribute identifiziert und mit Hilfe dedizierter Annahmen appliziert, um eine Quantifizierung der Simulationsmodellkomplexität zu ermöglichen. Das daraus resultierende Verfahren wurde anhand verschieden komplexer Simulationsmodelle evaluiert. Damit ist es möglich, bei der Erstellung eines FMU die Modellkomplexität im Prozessschritt „*Modelling*“ zu bestimmen, um diese innerhalb der Funktionen miteinander vergleichen zu können.

Allerdings korreliert die Komplexität der Simulationsmodelle nicht zwangsläufig mit deren Potential, das Systemverhalten exakt zu erfassen. Somit kann es vorkommen, dass die Simulationsergebnisse eines komplexen Simulationsmodells zwar bereits gemessene Daten genau genug abbilden, aber keine korrekte Prognose für noch unbekannte Daten ermöglichen [MYUNG 2000, S.190]. Umgekehrt ist es ebenfalls möglich, dass ein einfaches Modell beobachtete Daten ungenau wiedergibt, aber dennoch gute Prognosen über zukünftiges Verhalten ermöglicht. Dementsprechend müssen Modelle, gemäß dem einführenden Zitat aus dem Zeitschriftenaufsatz „*Toward a Method of Selecting Among Computational Models of Cognition*“ beide Aspekte gleichzeitig berücksichtigen. Damit sollen diese einerseits in der Lage sein, vorhandene Muster abzubilden, und andererseits auch Prognosen über zukünftiges Verhalten erlauben.

Zu diesem Zweck werden im Folgenden mit der Anpassungsgüte (engl. *Goodness of Fit* (GoF)) und dem Extrapolationsvermögen (engl. *Generalizability*) in Ergänzung zu der bereits quantifizierten Komplexität zwei

weitere Attribute eingeführt, mit deren Hilfe die Simulationsergebnisgüte von Modellen erfasst werden kann. Analog dem Vorgehen bei der Quantifizierung der Komplexität in Kapitel 5 werden auch diese zunächst definiert, bevor dafür Merkmale identifiziert werden. Das Potential der Attribute wird abschließend anhand der bereits in Kapitel 5.4 verwendeten Teilsystemsimulationenmodelle evaluiert.

6.1 Definition der Anpassungsgüte und des Extrapolationsvermögens

Die Anpassungsgüte repräsentiert die Übereinstimmung zwischen den Simulationsergebnissen und bereits beobachteten Daten des realen Systems. Im Gegensatz dazu ist das Extrapolationsvermögen bzw. die Vorhersagegenauigkeit ein Maß für die Übereinstimmung zwischen dem Verhalten des Simulationsmodells und dem des realen Systems im noch unbeobachteten Extrapolationsbereich. Für eine hohe Anpassungsgüte bei gleichzeitig hohem Extrapolationsvermögen sollen die Simulationsmodelle weder zu einfach noch zu komplex sein. Dieser Zusammenhang ist in Abbildung 6.1 qualitativ dargestellt.

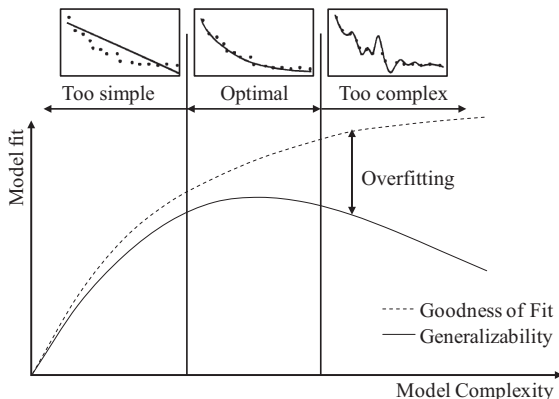


Abb. 6.1: Anpassungsgüte und Extrapolationsvermögen als Funktion der Modellkomplexität nach [MYUNG 2004, S.355]

Ursache dafür ist, dass sehr einfache Modelle in aller Regel die den beobachteten Daten zugrunde liegenden Kausalitäten und Merkmale schlechter

als relativ komplexere Modelle erfassen. Somit können sie das Verhalten des realen Systems tendenziell weniger genau abbilden und die Anpassungsgüte sinkt, je einfacher das Modell ist. Gleichzeitig verliert es durch die einfacheren Simulationsmodellgleichungen die Fähigkeit, weitere Daten, die außerhalb des beobachteten Bereichs liegen, zu prognostizieren, weswegen auch das Extrapolationsvermögen sinkt. Dieses Problem der Unteranpassung ist in dem kleinen Bild in der linken oberen Ecke von Abbildung 6.1 veranschaulicht. Darin bildet die vereinfacht als Gerade dargestellte Simulationsmodellgleichung weder die vorhandenen Messpunkte exakt nach, noch verfügt sie über ausreichend Potential, die außerhalb des dargestellten Bereichs liegenden Messpunkte im Extrapolationsbereich korrekt zu erfassen. Falls das Simulationsmodell sehr komplex ist, kann es in der Regel das Verhalten der beobachteten Daten und damit des realen Systems sehr genau abbilden, wodurch die Anpassungsgüte steigt. Da dabei aber auch fehlerbehaftete Daten im Simulationsmodell berücksichtigt werden, verliert es das Potential, zukünftiges Verhalten korrekt zu prognostizieren, weswegen das Extrapolationsvermögen sinkt. Dieses Problem wird auch als „Überanpassung“ (engl. *Overfitting*) bezeichnet und ist in dem kleinen Bild in der rechten oberen Ecke von Abbildung 6.1 veranschaulicht. Darin wird jeder beobachtete Punkt durch ein hochgradiges Polynom, welches die Simulationsmodellgleichung darstellt, exakt erfasst. Allerdings ist das Polynom nicht mehr in der Lage, Daten, die außerhalb des dargestellten Bereichs im Extrapolationsbereich liegen, abzubilden. Dementsprechend muss ein Kompromiss zwischen diesen beiden Extrema gefunden werden, der in Abbildung 6.1 als optimaler Bereich eingezeichnet ist. Darin wird, wie anhand des kleinen Bildes in der Mitte von Abbildung 6.1 dargestellt, durch eine angepasste Simulationsmodellgleichung zwischen den Messwerten interpoliert und damit eine ausreichend hohe Anpassungsgüte der bereits erfassten Messpunkte bei gleichzeitig hohem Extrapolationsvermögen für die noch nicht erfassten Daten erreicht [MYUNG 2004, S.355].

Die Notwendigkeit eines solchen Kompromisses wird neben Myung [MYUNG 2004] auch von Schoups [SCHOUUPS 2012] gefordert, der unabhängig davon eine eigene Korrelation hergeleitet hat. Diese ist in Abbildung 6.2 veranschaulicht.

In dieser Darstellungsvariante kann die Anpassungsgüte mit einem Kalibrierungsfehler (engl. *Calibration Error*) verglichen werden, welcher bewerten soll, ob das Modell sich den beobachteten Daten adäquat annähert.

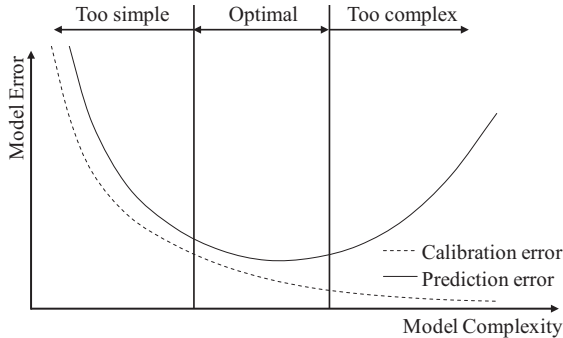


Abb. 6.2: Vorhersagefehler und Kalibrierungsfehler als Funktion der Modellkomplexität [SCHOUUPS 2012]

Gemäß Schoups' Annahme sinkt der Kalibrierungsfehler mit steigender Simulationsmodellkomplexität in einem vergleichbaren Trend, in dem bei Myung die Anpassungsgüte steigt. Darüber hinaus kann das Extrapolationsvermögen mit dem Vorhersagefehler (engl. *Prediction Error*) verglichen werden, welcher angeben soll, ob das Modell eine gute Vorhersage über zukünftige Ereignisse gestattet. Falls das Simulationsmodell sehr einfach oder sehr komplex ist, steigt der Vorhersagefehler relativ zu einer optimalen Simulationsmodellkomplexität in den Bereichen, in denen nach Myung das Extrapolationsvermögen sinkt. Somit handelt es sich bei Schoups [SCHOUUPS 2012] um eine inverse Sicht des gleichen Sachverhaltes und bei Abbildung 6.2 um die inverse Darstellung der Kurven aus Abbildung 6.1.

6.2 Quantifizierung der Anpassungsgüte

Die Anpassungsgüte ist ein Maß für die Übereinstimmung zwischen Simulationsergebnissen und bereits beobachteten Daten. Um diese zu quantifizieren, kann mit Hilfe mathematischer Verfahren die Korrespondenz zwischen dem ausführbaren Modell und der Realität erfasst werden, indem die Simulationsergebnisse y_i mit den Messwerten z_i abgeglichen werden. Dies entspricht einer Form der Validierung aus Kapitel 2.4.2.

Eines der geläufigsten Quantifizierungsverfahren ist die in Kapitel 2.3.2 vorgestellte Summe der quadratischen Fehler SSE, welche auch die Basis für weitere Verfahren liefert. Bei diesem Verfahren werden, wie in Formel 2.14

dargestellt, über sämtliche Messpunkte n die quadratischen Abweichungen zwischen Simulationsergebnissen y_i und gemessenen Daten z_i aufsummiert. Je kleiner die SSE ist, desto besser ist die Anpassungsgüte. Allerdings liefert die SSE als Ergebnis einen absoluten Wert und damit nur eine bedingt interpretierbare Größe. Besser geeignet ist die mittlere quadratische Abweichung (engl. *Mean Squared Error* (MSE)), welche ein relatives Maß darstellt:

$$MSE = \frac{\sum_{i=1}^n (z_i - y_i)^2}{n} = \frac{SSE}{n} \quad (6.1)$$

Dabei werden, analog zur SSE aus Gleichung 2.14, die Abweichungen zwischen Simulationsergebnissen und Messdaten aufsummiert und diese, in Ergänzung dazu, durch die Anzahl der Datenbeobachtungspunkte dividiert. Dementsprechend wächst das Ergebnis aus der MSE nicht wie bei der SSE mit der Anzahl der erfassten Residuen, sondern ist unabhängig von der Anzahl der Messpunkte. Spielen zusätzlich die Einheiten eine Rolle, kann über den MSE die Wurzel gebildet werden, woraus die Wurzel der mittleren quadratischen Abweichungen (engl. *Root Mean Squared Error* (RMSE)) resultiert:

$$RMSE = \sqrt{\sum_{i=1}^n (z_i - y_i)^2} = \sqrt{\frac{SSE}{n}} = \sqrt{MSE} \quad (6.2)$$

Mit Hilfe dieser Darstellungsvariante können die Einheiten, die bei der Bildung der SSE quadriert worden sind, wieder in die ursprünglichen Form gebracht werden.

Eine alternative Möglichkeit zur Quantifizierung der Anpassungsgüte besteht darin, ein normiertes Maß der Ungleichheit festzulegen, worin das Ergebnis einen definierten Wertebereich umspannt. Ein Beispiel dafür ist der Ungleichheitskoeffizient nach Theil (engl. *Theil's Inequality Coefficient* (TIC)), bei dem als Ergebnis immer Werte zwischen null und eins gebildet werden. Werte nahe null weisen darin auf eine starke Gleichheit zwischen Messung und Simulationsergebnis hin, wohingegen Werte nahe eins auf eine starke Ungleichheit schließen lassen [PELZ 2001, S.32]:

$$TIC = \frac{\sqrt{\sum_{i=1}^n (z_i - y_i)^2}}{\sqrt{\sum_{i=1}^n z_i^2} + \sqrt{\sum_{i=1}^n y_i^2}} \quad (6.3)$$

Allerdings stehen für eine solche Quantifizierung der Anpassungsgüte, wie bei der Absicherung der Simulationsmodelle in Kapitel 4.3.2 beschrieben, in Abhängigkeit von der gewählten Struktur und von den daraus resultierenden Schnittstellen nicht immer für alle Signale Messwerte aus den jeweils betrachteten technischen Systemen zur Verfügung. In diesem Fall können die Simulationsmodelle unabhängig von dem hier gewählten Verfahren zu Funktionsgruppen aggregiert werden, um validiert werden zu können.

6.3 Quantifizierung des Extrapolationsvermögens

Anhand der Darstellungen aus Kapitel 6.1 wird deutlich, dass die Betrachtung der Anpassungsgüte allein nicht ausreicht, um aus einer Vielzahl konkurrierender Simulationsmodelle das bestmögliche auszuwählen. Die ausschließliche Berücksichtigung der Anpassungsgüte würde zwangsläufig immer zur Wahl des jeweils komplexesten Modells führen, da dieses in der Regel die den Beobachtungen zugrunde liegende Kausalitäten am besten erfasst [MYUNG 2000, S.195]. Aus diesem Grund werden im Folgenden mit dem Informationskriterien nach Akaike (engl. *Akaike Information Criterion* (AIC)), dem Bayesschen Informationskriterium (engl. *Bayesian Information Criterion* (BIC)) und der Theorie der kürzesten Beschreibungslänge (engl. *Minimum Description Length* (MDL)) Verfahren eingeführt, mit deren Hilfe auch das Extrapolationsvermögen quantifiziert werden kann.

6.3.1 Akaikes Informationskriterium

Das Akaike Informationskriterium (AIC) hat seinen Ursprung in der Informationstheorie, die von Shannon und Wiener unabhängig voneinander Mitte des 20. Jahrhunderts entwickelt worden ist. Der darin definierte Begriff der Information diente als Grundlage für die Arbeiten von Kullback und Leibler, die einen Distanzbegriff zum Vergleich zweier Funktionen formuliert haben [SCHOMAKER 2010, S.30].

Als grundlegende Annahme soll eine Funktion f die volle Realität oder Wahrheit widerspiegeln, wohingegen die Funktion g für ein Modell steht, das versucht die Wahrheit so gut wie möglich, jedoch nur approximativ, auszudrücken. Die Kullback-Leibler-Information (KLI) $I(f,g)$ ist dabei ein

Maß für die verloren gegangene Information bei Verwendung des Modells g anstelle von f . Diese wird auch als „Kullback-Leibler-Divergenz“, „Kullback-Leibler-Diskrepanz“, „Kullback-Leibler-Abstand“ oder „negative Entropie“ bezeichnet. Für stetige Funktionen ist sie wieder mit z als Messdaten und θ als Parameter wie folgt definiert [SCHOMAKER 2006, S.14]:

$$I(f, g) = \int f(z) \ln \left(\frac{f(z)}{g(z|\theta)} \right) dz \quad (6.4)$$

Unter einer Menge von Modellen ist dasjenige das beste, das in Relation zu den anderen am wenigsten Informationen gegenüber dem wahren Modell verliert. Alternativ kann die KLI auch als Pseudodistanz zwischen der Realität und einem Modell interpretiert werden. Die KLI kann in ihrer ursprünglichen Form nicht für die Modellselektion verwendet werden, da sowohl eine umfassende Kenntnis der Realität als auch der Parameter θ der Modelle g erforderlich wäre. Da die Modellparameter jedoch geschätzt werden müssen, wird die erwartete anstelle der ursprünglichen KLI minimiert [BURNHAM 2002]. Zu diesem Zweck ist folgende Umformung nötig [SCHOMAKER 2006, S.15]:

$$\begin{aligned} I(f, g) &= \int f(z) \ln \left(\frac{f(z)}{g(z|\theta)} \right) dz \\ &= \int f(z) \ln(f(z)) dz - \int f(z) \ln(g(z|\theta)) dz \\ &= E_f[\ln(f(z))] - E_f[\ln(g(z|\theta))] \end{aligned} \quad (6.5)$$

Da es sich bei dem ersten Term $E_f[\ln(f(z))]$ um eine Konstante handelt, genügt es als Gütekriterium, die relativ erwartete KLI $E_f[\ln(g(z|\theta))]$ zu schätzen. Dies ist gemäß Akaike [AKAIKE 1973, S.276] mit Hilfe der Likelihood-Theorie möglich, womit eine Verbindung zwischen Informationstheorie und Statistik hergestellt werden konnte. Der Parameter $\hat{\theta}$ repräsentiert demnach die Maximum-Likelihood-Schätzung von θ , basierend auf dem Modell g und den Daten z . Der maximierte Log-Likelihood-Wert wird als verzerrter Schätzer von $E_f[\ln(g(z|\hat{\theta}))]$ interpretiert, wobei die Verzerrung näherungsweise der Anzahl der zu schätzenden Parameter entspricht. Daraus resultiert für die relativ erwartete KLI [SCHOMAKER 2006, S.15]:

$$rel\hat{E}(KL) = \ln L(\hat{\theta}|z) - k \quad (6.6)$$

Der Parameter k steht für die Anzahl der zu schätzenden Parameter. Dieses Ergebnis wird aus historischen Gründen mit -2 multipliziert, einem Wert, der auch als negative Entropie aufgefasst werden kann [RÜGER 1999, S.86]. Als Ergebnis resultiert das Informationskriterium nach Akaike [AKAIKE 1973, S.276]:

$$AIC = -2 \ln L(\hat{\theta}|z) + 2k \quad (6.7)$$

Da es sich bei einer Regressionsanalyse über die Methode der kleinsten Quadrate um ein alternatives Parameteridentifikationsverfahren zur Methode der Maximalen Wahrscheinlichkeit handelt, kann das Akaike Informationskriterium auch über die SSE dargestellt werden. Diese gilt für den Spezialfall eines linearen Regressionsmodells, also für Modelle, innerhalb derer die untersuchten Zusammenhänge linear sind bzw. linearisiert werden können, woraus grundsätzlich gleichartige Verläufe resultieren [LUDWIG-MAYERHOFER 2006, S.18]. Man erhält durch entsprechende Umformung [SCHOMAKER 2010, S.36]:

$$AIC = n \ln(MSE) + 2k \quad (6.8)$$

Der erste Term ist in beiden Darstellungsvarianten ein Maß für die Abweichung der Simulationsergebnisse gegenüber den beobachteten Daten und repräsentiert damit die Anpassungsgüte [MYUNG 1999, S.3]. Wie bei deren Definition in Kapitel 6.1 erläutert, ist diese Abweichung umso kleiner, je größer die Anzahl der Parameter ist. Damit besteht der Trend, dass bei einer Vielzahl konkurrierender Darstellungsformen für einen Betrachtungsgegenstand grundsätzlich immer die komplexeste Form selektiert wird. Um diesem Trend entgegenzuwirken, wird im zweiten Term eine zu große Anzahl an Parametern und damit eine zu hohe Komplexität „bestraft“. Je kleiner der daraus resultierende Wert für die AIC ausfällt, desto kleiner ist die KLI und desto größer ist das Extrapolationsvermögen. Somit wird das Modell mit dem geringsten AIC-Wert selektiert [SCHOMAKER 2010, S.36].

6.3.2 Bayessches Informationskriterium

Das Bayessche Informationskriterium (BIC) ist ein weiteres Maß für das Extrapolationsvermögen, welches auf der bayesianischen Statistik basiert. Dieses wird auch als „Schwarzsches Bayes-Kriterium“ bzw. „Schwarzsches Informationskriterium“ bezeichnet.

Ausgehend von einer A-priori-Verteilung einer Stichprobe für den Parameter θ stellt in der bayesianischen Statistik die A-posteriori-Verteilung den neuen, durch Vorwissen und Beobachtung bestimmten Kenntnisstand über die Verteilung des Parameters θ nach der Beobachtung der Stichprobe dar. Dabei bezeichnet die *Maximum-a-Posteriori* (MaP)-Methode ein Schätzverfahren, das einen vorgegebenen Parameter durch den Modalwert der A-posteriori-Verteilung schätzt. Es besteht somit eine gewisse Ähnlichkeit zur Methode der maximalen Wahrscheinlichkeit, wobei die MAP-Methode Vorwissen in Form des wahrscheinlichsten Parameters unter der A-posteriori-Verteilung verwendet, wohingegen die Maximum-Likelihood-Methode den höchsten Wert ohne Vorwissen einsetzt.

Gegeben seien eine Menge k_{Mod} an Modellen und ein Datensatz z [LANTERMAN 2001, S.188]. Jedes Modell hat eine A-priori-Wahrscheinlichkeit $p_{k_{Mod}}(k_{Mod})$ und eine A-priori-Dichte von $p_l(z|\theta, k_{Mod})$ und $p_p(\theta|k_{Mod})$. Dabei sei θ Element des zu dem Modell gehörenden Parameterraums. Bei einem bayesianischen Ansatz soll der störende Parameter θ durch Integration beseitigt werden, um die Wahrscheinlichkeit von z bei einem gegebenen Modell k_{Mod} identifizieren zu können. Damit soll dasjenige Modell k_{Mod} gewählt werden, das $p(z, k_{Mod})$ maximiert [SCHOMAKER 2006, S.16]:

$$p(z, k_{Mod}) = p_{k_{Mod}}(k_{Mod}) \int p_l(z|\theta, k_{Mod}) p_p(\theta|k_{Mod}) d\theta \quad (6.9)$$

In einigen wenigen Spezialfällen kann das bayesianische Integral direkt berechnet werden. In den meisten Fällen ist es jedoch nötig, sich diesem durch die von Schwarz entwickelte Approximationsmethode anzunähern (siehe dazu [SCHWARZ 1978]):

$$\begin{aligned} L(z|\theta, k_{Mod}) &= \ln p_l(z|\theta, k_{Mod}) \\ P(\theta|k_{Mod}) &= \ln p_p(\theta|k_{Mod}) \\ H(z, \theta|k_{Mod}) &= L(z|\theta, m) + P(\theta|k_{Mod}) \end{aligned} \quad (6.10)$$

Der Term $\hat{\theta}(z) = \operatorname{argmax}_{\theta} H(z, \theta|k_{Mod})$ bezeichnet im Folgenden die MAP-Schätzung. Der Parameter k steht wieder für die Anzahl der zu schätzenden Parameter. Einsetzen von 6.10 in 6.9 ergibt [LANTERMAN 2001, S.189]:

$$p(z, k_{Mod}) = p_{k_{Mod}}(k_{Mod}) \int \exp[H(z, \theta|k_{Mod})] d\theta \quad (6.11)$$

Der Integrand aus Formel 6.11 kann über die Laplace-Methode, mit der eine Taylorreihen-Entwicklung von H um die MAP-Schätzung durchgeführt wird, approximiert werden. Nach entsprechender Umformung erhält man folgenden Ausdruck:

$$H(z, \theta | k_{Mod}) \approx H(z, \hat{\theta} | k_{Mod}) - \frac{1}{2}(\theta - \hat{\theta})^T I_H(z : \hat{\theta} | k_{Mod})(\theta - \hat{\theta}) \quad (6.12)$$

mit

$$I_H(z : \hat{\theta} | k_{Mod}) = -\frac{\partial^2 H(z, \theta | k_{Mod})}{\partial \theta_i \partial \theta_j} \quad (6.13)$$

Wegen der Untersuchung des Maximums verschwindet hierbei die erste Ableitung und damit der zweite Term der mehrdimensionalen Taylorreihen-Entwicklung aus Gleichung 6.12 [LANTERMAN 2001, S.189]. Das Einsetzen von Formel 6.12 in 6.11 ergibt:

$$\begin{aligned} p(z, k_{Mod}) &= p_{k_{Mod}}(k_{Mod}) \int \exp[H(z, \theta | k_{Mod})] d\theta \\ &\approx p_{k_{Mod}}(k_{Mod}) \exp[H(z, \hat{\theta} | k_{Mod})] \cdot \\ &\quad \cdot \int \exp\left[-\frac{1}{2}(\theta - \hat{\theta})^T I_H(z : \hat{\theta} | k_{Mod})(\theta - \hat{\theta})\right] dz \quad (6.14) \\ &= p_{k_{Mod}}(k_{Mod}) \exp[H(z, \theta | k_{Mod})] \frac{(2\pi)^{\frac{k}{2}}}{\sqrt{\det I_H(z : \hat{\theta} | k_{Mod})}} \end{aligned}$$

Der letzte Schritt der Umformung basiert auf der Tatsache, dass der Integrand als die quadratische Form einer Gaußverteilung aufgefasst werden kann [SCHOMAKER 2006, S.17]. Logarithmieren ergibt:

$$\begin{aligned} \ln p(z, k_{Mod}) &\approx \ln p_{k_{Mod}}(k_{Mod}) + H(z, \hat{\theta} | k_{Mod}) + \\ &\quad + \frac{k}{2} \ln 2\pi - \frac{1}{2} \ln \det I_H(z : \hat{\theta} | k_{Mod}) \\ &= \ln p_{k_{Mod}}(k_{Mod}) + L(z | \hat{\theta}, k_{Mod}) + P(\hat{\theta} | k_{Mod}) + \\ &\quad + \frac{k}{2} \ln 2\pi - \frac{1}{2} \ln \det I_H(z : \hat{\theta} | k_{Mod}) \quad (6.15) \end{aligned}$$

Der zweite Term in Gleichung 6.15 kann unter der Annahme, dass N unabhängige und gleich verteilte Daten $z = z_1, z_2, \dots, z_N$ vorliegen, mit der Log-Likelihood-Funktion substituiert werden [LANTERMAN 2001, S.190]:

$$L(z|\theta, m) = \sum_{i=1}^N L(z_i|\theta, k_{Mod}) \quad (6.16)$$

Die empirische Fisher Information hat dabei folgende Form:

$$I_L(z : \hat{\theta}|k_{Mod}) = \sum_{i=1}^N I_L(z_i : \hat{\theta}(z)|k_{Mod}) \quad (6.17)$$

Für die Substitution des letzten Terms aus Gleichung 6.15 sei nun $\hat{\theta}_{\lim} = \lim_{N \rightarrow \infty} \hat{\theta}(z)$. Mit dem Gesetz der großen Zahlen kann gezeigt werden:

$$\frac{E_{p(z|\hat{\theta}_{\lim}, k_{Mod})}[I_L(z_1 : \hat{\theta}(z)|k_{Mod})]}{\frac{1}{N} I_L(z : \hat{\theta}(z)|k_{Mod})} \rightarrow 1 \quad (6.18)$$

Damit kann der letzte Term aus Gleichung 6.15 asymptotisch angenähert werden [LANTERMAN 2001, S.191]:

$$\begin{aligned} & -\frac{1}{2} \ln \det I_L(z : \hat{\theta}|k_{Mod}) \\ & \approx -\frac{1}{2} \ln \det \left\{ N E_{p(z|\hat{\theta}_{\lim}, k_{Mod})}[I_L(z_1 : \hat{\theta}(z)|k_{Mod})] \right\} \\ & = -\frac{1}{2} \ln \det N I_k - \frac{1}{2} \ln \det E_{p(z|\hat{\theta}_{\lim}, k_{Mod})}[I_L(z_1 : \hat{\theta}(z)|k_{Mod})] \\ & = -\frac{k}{2} \ln N - \frac{1}{2} \ln \det E_{p(z|\hat{\theta}_{\lim}, k_{Mod})}[I_L(z_1 : \hat{\theta}(z)|k_{Mod})] \end{aligned} \quad (6.19)$$

Das Einsetzen von Gleichungen 6.16 und 6.19 in Gleichung 6.15 ergibt folgenden Ausdruck:

$$\begin{aligned} \ln p(z, k_{Mod}) & \approx \ln p_{k_{Mod}}(k_{Mod}) + \sum_{n=1}^N L(z_n|\hat{\theta}(z), k_{Mod}) + P(\hat{\theta}|k_{Mod}) + \\ & + \frac{k}{2} \ln 2\pi - \frac{k}{2} \ln N - \frac{1}{2} \ln \det E_{p(z|\hat{\theta}_{\lim}, k_{Mod})}[I_L(z_1 : \hat{\theta}(z)|k_{Mod})] \end{aligned} \quad (6.20)$$

Für $N \rightarrow \infty$ dominieren die Funktionen von N gegenüber den anderen Termen:

$$\ln p(z, k_{Mod}) \approx L(z_n|\hat{\theta}(z), k_{Mod}) - \frac{k}{2} \ln N \quad (6.21)$$

Dieses Ergebnis kann aus historischen Gründen wieder mit -2 multipliziert werden, wodurch man schließlich die Formel für das BIC erhält [LANTERMAN 2001, S.192]:

$$BIC = -2 \ln L(\hat{\theta}|z) + k \ln(n) \quad (6.22)$$

Analog zum AIC kann auch beim BIC für den Spezialfall eines linearen Regressionsmodells folgende Notation verwendet werden:

$$BIC = n \ln(MSE) + k \ln(n) \quad (6.23)$$

Trotz der unterschiedlichen Theorien, die beiden Verfahren zugrunde liegen, unterscheidet sich das Ergebnis für das BIC lediglich um den Faktor $2 \frac{1}{\ln(n)}$ im zweiten Term vom AIC aus Gleichung 6.7. Im Gegensatz zum AIC bestraft das BIC damit auch die Anzahl der verwendeten Messpunkte, da viele Stützstellen eine exaktere Parameteridentifikation unterstützen. Diese exaktere Schätzung führt dazu, dass Modelle mit vielen Parametern den Verlauf der Messdaten sehr genau abbilden können und damit Modelle mit wenigen Parametern relativ benachteiligt werden. Dieser Tendenz wird mit der Anzahl der Messpunkte im Strafterm entgegengewirkt. Auch hier hat das Modell das größte Extrapolationsvermögen, das das BIC minimiert [MYUNG 1999, S.3].

6.3.3 Theorie der kürzesten Beschreibungslänge

Die Theorie der kürzesten Beschreibungslänge (MDL) hat ihren Ursprung in der algorithmischen Kodierungstheorie der Informatik und repräsentiert die kürzeste Codelänge, mit der ein Modell beschrieben werden kann [MYUNG 1999, S.4]. Die MDL ist auf Rissanen zurückzuführen und basiert auf der Idee, durch statistische Schlussfolgerungen eine Regelmäßigkeit in den beobachteten Daten zu identifizieren, wodurch diese komprimiert werden können [GRÜN WALD 2005, S.9] (vergleiche hierzu [RISSANEN 1978]).

Aus statistischer Sicht bedeutend ist dabei der Zusammenhang zwischen Codelängenfunktion und Wahrscheinlichkeitsdichtefunktionen, der mit Hilfe der Kraft-McMillan-Ungleichung hergestellt wird (siehe hierzu [KRAFT 1949]). Die Reduzierung der Codelänge korreliert dabei mit der Wahl geeigneter Wahrscheinlichkeitsfunktionen, wodurch sich die Anzahl der Kandidatenmodelle verringern lässt [SCHOMAKER 2010, S.44]. Bei einer

gegebenen Menge an Modellen muss damit das Modell gefunden werden, welches die beobachteten Daten am stärksten komprimiert [MYUNG 1999, S.3]. Dieses wird wie folgt formalisiert:

$$MDL = -\ln L(\hat{\theta}|z) + \frac{k}{2} \ln \frac{n}{2\pi} + \ln \int \|I_1(\theta)\|^{\frac{1}{2}} d\theta \quad (6.24)$$

Der erste Term ist darin wieder ein Maß für die Abweichung der Simulationsergebnisse zu beobachteten Daten und repräsentiert dementsprechend die Anpassungsgüte [MYUNG 1999, S.3]. Damit nicht zu komplexe Modelle selektiert werden, gibt es auch hier einen zweiten Teil der Modellgleichung, der zu hohe Komplexität bestraft. Auch hier wird die Notation mit dem Absolutbetrag gewählt, um dem Definitionsbereich der Logarithmusfunktion gerecht zu werden. Dessen beide Terme entsprechen, der Definition der geometrischen Komplexität aus Kapitel 5.2.3 folgend, dem Absolutbetrag der GC aus Formel 5.12. Damit kann die MDL wie folgt dargestellt werden:

$$MDL = -\ln L(\hat{\theta}|z) + GC \quad (6.25)$$

Mit Hilfe des Strafterms werden somit, in Ergänzung zum BIC, neben der Anzahl der Parameter und der Anzahl der Messpunkte über die Fisher-Informationsmatrix auch noch die funktionale Form und der Parameterwertebereich berücksichtigt. Dadurch wird nicht nur das Prinzip der Sparsamkeit direkt bei der Konstruktion der Modellwahlkriterien verwendet, sondern auch eine konkrete Ausarbeitung davon präsentiert, wie Sparsamkeit gemessen werden kann [SCHOMAKER 2010, S.13]. Je kleiner die MDL dabei ausfällt, desto höher das Extrapolationsvermögen [MYUNG 1999, S.3].

6.4 Evaluation anhand mehrerer Teilsystemsimulationsmodelle

Für die Quantifizierung der Simulationsergebnisgüte wird als vorbereitende Maßnahme die Anpassungsgüte sämtlicher Simulationsmodelle innerhalb der exemplarischen Funktionen der Gesamtfahrzeugmodellarchitektur aus Abbildung 4.8 bewertet. Dabei werden diejenigen Funktionen identifiziert, deren initiale Parametrierung eine unzureichende Anpassung gestattet, um diese gegebenenfalls gezielt einzeln optimieren zu können. Anschließend

wird die eigentliche Gütequantifizierung anhand der bereits bekannten Funktionen der Hochvoltbatterie („BattHV“) und Fahrzeuglängsdynamik („FzgLängsD“) demonstriert, die in Kapitel 5.4 für die Quantifizierung der Komplexität verwendet worden sind. Um die Wirkmechanismen der Quantifizierungsverfahren zu analysieren, werden auf Basis des dabei eingesetzten CAN-Traces die Attribute der Simulationsergebnisgüte ermittelt. In Ergänzung dazu werden weitere CAN-Traces eingesetzt, um für die Funktion der Hochvoltbatterie zusätzlich eine Gütequantifizierung im noch unbekanntem Extrapolationsbereich zu evaluieren.

6.4.1 Gütequantifizierung einer Gesamtfahrzeugmodellarchitektur

Zunächst wird die Anpassungsgüte sämtlicher Funktionen einer Gesamtfahrzeugmodellarchitektur bewertet. Damit kann innerhalb eines MUs die Qualität initial parametrierter Simulationsmodelle bewertet werden, um so gezielt Modelle mit großen Abweichungen gegenüber den dafür verwendeten Messdaten optimieren zu können. Zu diesem Zweck wird der bei der Quantifizierung der Anpassungsgüte in Kapitel 6.2 eingeführte TIC verwendet, der ein normiertes Maß für die Ungleichheit der Simulationsergebnisse gegenüber Messdaten darstellt. Dabei werden die Eingangsgrößen der Simulationsmodelle mit Messdaten beaufschlagt und die Ergebnisse mit Hilfe des TICs wiederum mit Messdaten verglichen [NETTER 2013a, S.46].

Da, wie bei der Absicherung der Simulationsmodelle in Kapitel 4.3.2 beschrieben, nicht immer für alle Ein- und Ausgänge Messdaten verfügbar sind, werden die Funktionsgruppen aus Abbildung 4.13 verwendet, die in Kapitel 4.4.2 im Fall der in dieser Arbeit als Evaluationsbeispiel gewählten Gesamtfahrzeugmodellarchitektur eines A1 e-tron bereits identifiziert worden sind. Für deren Quantifizierung werden Daten verschiedener Fahrten (engl. *Trips*) aus dem Pilotprojekt mit einer Dauer von jeweils mehr als 20 Minuten verwendet, um eine repräsentativ breite Datenbasis bereitzustellen. Da durch die Anwendung des Suchalgorithmus gewährleistet wird, dass für alle Eingangsschnittstellen Daten zur Verfügung stehen, ist es möglich, jede Funktion zu simulieren. Die Ergebnisse dieser Quantifizierung sind in Tabelle 6.1 aufgelistet. Werte nahe null weisen darin auf eine starke Gleichheit zwischen Messung und Simulationsergebnis hin, wohingegen Werte nahe eins auf eine starke Ungleichheit schließen lassen [PELZ 2001, S.32].

Die Nomenklatur basiert dabei auf der bereits bekannten schematischen Darstellung der Funktionsgruppen aus Abbildung 4.13.

Tab. 6.1: Ergebnisse der nach ihrer Anpassungsgüte quantifizierten Funktionen nach [NETTER 2013a, S.46]

Resulting Compositi- on	Output Signal	<i>TIC</i>				
		Trip 1	Trip 2	Trip 3	Trip 4	Trip 5
RangeEx (SR)	1	0,21	0,19	0,26	0,22	0,27
RangeEx (El)	2	0,07	0,07	0,11	0,15	0,08
	13	0,03	0,05	0,07	0,10	0,05
FG_1	5	0,26	0,14	0,26	0,07	0,08
	6	0,29	0,26	0,36	0,31	0,27
	11	0,32	0,19	0,32	0,09	0,13
	12	0,26	0,14	0,26	0,07	0,08
	4	0,23	0,25	0,30	0,28	0,27
FG_2	3	0,19	0,18	0,46	0,22	0,38
FG_3	7	0,01	0,00	0,01	0,01	0,01
	8	0,02	0,01	0,01	0,02	0,02
	9	0,06	0,07	0,08	0,05	0,08
FG_4	7	0,04	0,01	0,02	0,02	0,04
	8	0,02	0,01	0,02	0,02	0,03
	9	0,16	0,18	0,20	0,12	0,26
	1	0,22	0,21	0,26	0,17	0,31
FG_5	2	0,22	0,21	0,27	0,18	0,31
	10	0,48	0,53	0,47	0,51	0,47

Die erste Spalte beinhaltet die identifizierten Funktionsgruppen. Gemäß den Erläuterungen bei der Absicherung der Gesamtfahrzeugmodellarchitektur in Kapitel 4.4.2 können die beiden Funktionen „RangeEx“ aus den Kategorien „Steuerung/Regelungstechnik“ bzw. „Elektrotechnik“ ohne Gruppierung in einer Funktionsgruppe direkt validiert werden. Dementsprechend sind diese beiden Funktionen separat mit aufgelistet. Die jeweils quantifizierten Ausgangssignale sind zeilenweise in der zweiten Spalte eingetragen. Einzelne Ausgangssignale erscheinen darin mehrfach, da die dazugehörige Funktion Bestandteil mehrerer Funktionsgruppen ist. Die verbleibenden fünf Spalten bieten eine Übersicht über die Ergebnisse, die mit Hilfe des TICs für die verschiedenen Fahrten ermittelt worden sind. Als Beispiel

ist die Abweichung des Ausgangssignals „8“ gegenüber den verwendeten Messdaten durchgängig relativ gering, wohingegen die Ausgangssignale „6“ und „10“ eine relativ schlechte Anpassungsgüte aufweisen [NETTER 2013a, S.46]. Basierend auf den Ergebnissen muss für die Simulationsmodelle mit einer relativ geringen Abweichung nicht zwangsläufig zuerst eine Parameteridentifikation durchgeführt werden, um die Simulationsergebnisgüte quantifizieren zu können.

6.4.2 Gütequantifizierung von Teilsystemsimulationsmodellen

Für die Quantifizierung der Simulationsergebnisgüte werden im Folgenden mit der SSE und dem MSE weitere der in Kapitel 6.2 vorgestellten Verfahren zur Quantifizierung der Anpassungsgüte verwendet. Auf eine Normierung, wie sie über den TIC möglich ist, wird dabei verzichtet, um auch hier die Ergebnisse unterschiedlicher Simulationsmodelle einer Funktion miteinander vergleichen zu können. Das Extrapolationsvermögen wird mit Hilfe sämtlicher Verfahren aus Kapitel 6.3 quantifiziert, um die Unterschiede bei einer ergänzenden Betrachtung der Messpunkte im zweiten Term beim BIC gegenüber dem AIC bzw. einer zusätzlichen Betrachtung der funktionalen Form bei der MDL gegenüber dem BIC bewerten zu können.

Bei den für die Quantifizierung verwendeten CAN-Traces handelt es sich um die gleichen Daten, die auch für die Komplexitätsquantifizierung in Kapitel 5.4 eingesetzt worden sind. Diese sind in den Abbildungen 5.10 und 5.11 dargestellt. Die Simulationsmodelle der Hochvoltbatterie („BattHV“) und der Fahrzeuginerpendenzdynamik („FzgLängsD“) bleiben unverändert.

Sowohl AIC als auch BIC werden dabei zunächst mit Hilfe der Standardformeln aus 6.7 bzw. 6.22 ermittelt. Die Werte für MDL folgen aus der Berechnung mit Gleichung 6.25. Die Resultate für die Quantifizierung der Batteriemodelle nach ihrer Simulationsergebnisgüte sind in Tabelle 6.2 aufgelistet, die der Längsdynamikmodelle in Tabelle 6.3. Zusätzlich ist darin der $\ln(\text{MLE})$ mit eingetragen, um die beiden Terme für AIC, BIC und MDL im Folgenden separat bewerten zu können.

Wie in Kapitel 6.2 eingeführt, wird die SSE aus dem Produkt aus dem MSE und der Anzahl an Messpunkten n gebildet. Da die Anzahl der berücksichtigten $n = 100$ Messpunkte beim Batteriemodell und der $n = 200$ Messpunkte beim Längsdynamikmodell jeweils konstant ist, hängen beide

Tab. 6.2: Ergebnisse der nach ihrer Simulationsergebnisgüte quantifizierten Batteriemodelle

Typ	SSE	MSE	RMSE	$\ln(MLE)$	AIC	BIC	MDL
R_i	0,85	0,0085	0,0922	96,419	-190,839	-188,234	-91,868
RC_1	0,80	0,0080	0,0894	99,349	-192,699	-184,883	-77,120
RC_2	0,79	0,0079	0,0889	100,118	-190,356	-177,330	-60,312
RC_3	0,78	0,0078	0,0883	100,750	-187,501	-169,265	-43,294

Tab. 6.3: Ergebnisse der nach ihrer Simulationsergebnisgüte quantifizierten Längsdynamikmodelle

Typ	SSE	MSE	RMSE	$\ln(MLE)$	AIC	BIC	MDL
$cw_{konst.}$	0,241	0,0012	0,0346	393,410	-784,820	-781,517	-389,003
$cw_{lin.}$	0,241	0,0012	0,0346	393,671	-782,734	-776,128	-385,129
$cw_{quadr.}$	0,241	0,0012	0,0346	393,558	-781,116	-771,206	-372,637

Werte linear voneinander ab. Der RMSE entspricht der Wurzel des MSE, weswegen auch hier ein proportionaler Zusammenhang besteht. Somit ist es im Folgenden zulässig, allein den Trend des MSE zu bewerten, da deren qualitativer Verlauf auch Rückschlüsse auf die beiden anderen Quantifizierungsmerkmale gestattet. Die aus Tabelle 6.3 resultierenden Verläufe der Simulationsergebnisgüte sind für die Batteriemodelle in Abbildung 6.3 und für die Längsdynamikmodelle in Abbildung 6.4 dargestellt.

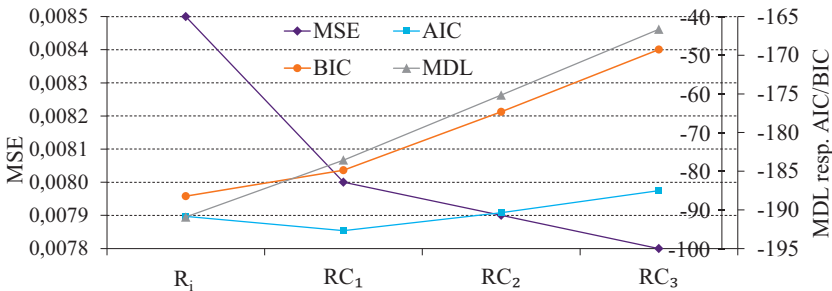


Abb. 6.3: Verläufe der Simulationsergebnisgüte der Batteriemodelle

In Abbildung 6.3 fallen die Werte des MSE vom R_i -Modell zum RC_1 -Modell zunächst um mehr als 5 % und danach bis zum RC_3 -Modell konstant weiter. Generell sinken die Abweichungen zwischen gemessenen und simulierten Daten, weswegen die Anpassungsgüte mit steigender Komplexität zunimmt.

Das erste RC-Element bringt dabei offenbar die größte Verbesserung. Die Verläufe für das Extrapolationsvermögen weisen alle einen weitestgehend positiven Trend auf, weswegen von einem sinkenden Extrapolationsvermögen mit steigender Komplexität ausgegangen werden kann. Dabei sind die Werte des ersten Terms bei allen Simulationsmodellen negativ, da die positiven Log-Likelihood-Schätzungen aus Tabelle 6.2 jeweils mit einer negativen Zahl multipliziert werden. Dieser erste Term ist bei allen Batteriemodellen absolut betrachtet umso negativer, je komplexer das Simulationsmodell ist. Um diesem Trend entgegenzuwirken, wird im zweiten Term eine zu große Anzahl an Parametern und damit eine zu hohe Komplexität bestraft, indem das negative Ergebnis des ersten Terms mit einem positiven zweiten Term verkleinert wird. Dieser Strafterm wächst beim AIC mit steigender Anzahl an Parametern und damit einer steigenden Komplexität. Im Fall des Batteriemodells kommt es dabei zu dem in Abbildung 6.3 dargestellten Richtungswechsel im Trend des AIC, da die geringe Zunahme der Log-Likelihood-Schätzungen mit steigender Komplexität ab dem RC_1 -Modell vom additiven zweiten Term überkompensiert wird. Im Gegensatz dazu steigen die Werte des BICs gleich ab dem R_i -Modell kontinuierlich mit steigender Komplexität. Dieser unterscheidet sich ausschließlich im zweiten Term vom AIC. Im Gegensatz zum AIC korreliert er nicht nur mit einer steigenden Anzahl an Parametern, sondern auch mit der über die Batteriemodelle konstanten Anzahl an Messpunkten. Da in diesem Evaluationsbeispiel der Logarithmus der $n = 100$ verarbeiteten Messpunkte größer als der konstante Faktor zwei beim AIC ist, steigt der Verlauf des BICs mit steigender Komplexität steiler an als der Verlauf des AICs. Aufgrund dessen findet beim BIC auch kein Richtungswechsel statt. Die Werte der MDL nehmen ebenfalls kontinuierlich mit steigender Komplexität zu. Dieses Maß unterscheidet sich um den Faktor $\frac{1}{2}$ im ersten Term von AIC und BIC. Dementsprechend sind die Werte der MDL aus Tabelle 6.2 im ersten Term halb so groß wie die der anderen Quantifizierungsmerkmale, weswegen sich das Endergebnis auch unter Berücksichtigung des zweiten Terms erheblich von den Werten der anderen Quantifizierungsmerkmale unterscheidet. Gegenüber dem BIC berücksichtigt die MDL über die Fisher-Informationsmatrix im zweiten Term auch die funktionale Form der Modellgleichung. In der Summe entspricht der Strafterm dem Absolutbetrag der GC. Deren Werte für die Batteriemodelle sind in Tabelle 5.2 in Kapitel 5.4.1 aufgelistet.

Referenziert man Abbildung 6.3 auf Abbildung 6.2, kann der MSE mit dem darin dargestellten Kalibrierungsfehler verglichen werden. Je kleiner der MSE ausfällt, desto kleiner ist der Kalibrierungsfehler, weswegen bei alleiniger Betrachtung der Anpassungsgüte das RC_3 -Modell selektiert werden würde. Allerdings lässt dessen fallender Trend alleine, wie in Abbildung 6.2 zu sehen ist, keine Aussage darüber zu, ob die vorhandenen Modelle zu einfach, optimal oder zu komplex sind. Zu diesem Zweck können AIC, BIC und MDL als Maß für den Vorhersagefehler herangezogen werden. Je niedriger der Wert für diese Kennzahlen ausfällt, desto kleiner ist der Vorhersagefehler. Damit deutet deren steigender Verlauf in Abbildung 6.3 auf eine zu hohe Komplexität und damit eine Überanpassung hin, weswegen bei alleiniger Betrachtung des Extrapolationsvermögens das R_i -Modell bzw. das RC_1 -Modell selektiert werden würde. Demnach bieten für die vorliegenden Daten die Modelle der mittleren Komplexität den besten Kompromiss aus hoher Anpassungsgüte bei gleichzeitig hohem Extrapolationsvermögen. In diesem Bereich befindet sich auch das Minimum des AIC.

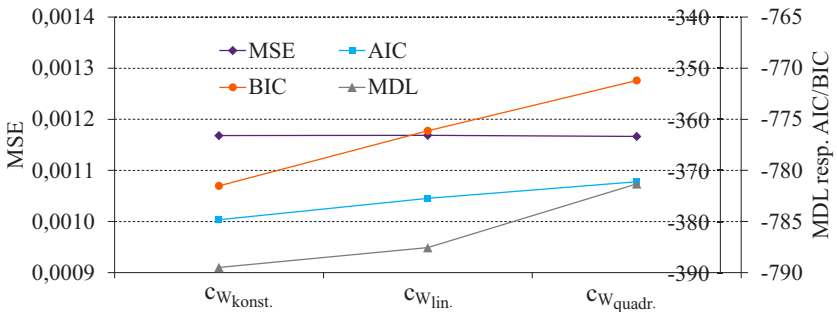


Abb. 6.4: Verläufe der Simulationsergebnisgüte der Längsdynamikmodelle

In der Ergebnisdarstellung der Längsdynamikmodellquantifizierung aus 6.4 bleibt der MSE bei gleicher Skalierung der Ordinate wie in Abbildung 6.3 weitestgehend unverändert. Die jeweils komplexeren Modelle können demnach für den verwendeten Datensatz kaum eine Verbesserung der Anpassungsgüte erzielen. Der Geschwindigkeitsverlauf aus Abbildung 5.11 kann somit von dem konstanten Längsdynamikmodell $c_{W_{konst.}}$ als auch von dem linearen $c_{W_{lin.}}$ - und quadratischen $c_{W_{quadr.}}$ -Modell ähnlich gut approximiert werden. Für das Extrapolationsvermögen resultieren bei allen Merkmalen positive Trends. Da hier alle Log-Likelihood-Werte aus Tabelle

6.3 vergleichbar groß sind, sind die steigenden Werte größtenteils dem Strafterm geschuldet. Auch hier ist beim BIC der Logarithmus der $n = 200$ verarbeiteten Messpunkte größer als der konstante Faktor zwei beim AIC, weswegen der Verlauf des BIC steiler ansteigt als der des AIC. Der Verlauf der MDL wächst proportional zur steigenden GC aus Tabelle 5.4. An dieser Stelle sei darauf hingewiesen, dass ohne die Definition aus Kapitel 5.2.3, nach der die GC den Abstand eines Simulationsmodells zum Koordinatenursprung darstellt und deshalb stets deren Betrag zu nehmen ist, ein inverser Verlauf für die MDL resultiert wäre. Dabei würde die MDL umso negativer werden, je größer die Komplexität des Simulationsmodells wird, weswegen immer das jeweils komplexeste Simulationsmodell selektiert werden würde. Da mit dem Strafterm auf Basis verschiedener Quellen versucht wird, diesem Trend entgegenzuwirken (siehe dazu [SCHOMAKER 2010, S.36] und [MYUNG 1999, S.3]), plausibilisiert dieses Ergebnis die getroffene Definition.

Referenziert man Abbildung 6.4 wieder auf Abbildung 6.2, lässt der weitestgehend konstante Trend des MSE keine Aussage über den Bereich der Simulationsmodellkomplexität zu. Im Gegensatz dazu deutet der kontinuierlich steigende Verlauf auf eine zu hohe Komplexität und damit eine Überanpassung hin, weswegen bei alleiniger Betrachtung des Extrapolationsvermögens das Modell mit c_W als Parameter selektiert werden würde. Demnach bieten das c_W -Modell mit c_W als Parameter für die vorliegenden Daten insgesamt den besten Kompromiss zwischen einer hohen Anpassungsgüte und gleichzeitig hohem Extrapolationsvermögen.

6.4.3 Gütequantifizierung bei noch unbekanntem Extrapolationsbereich

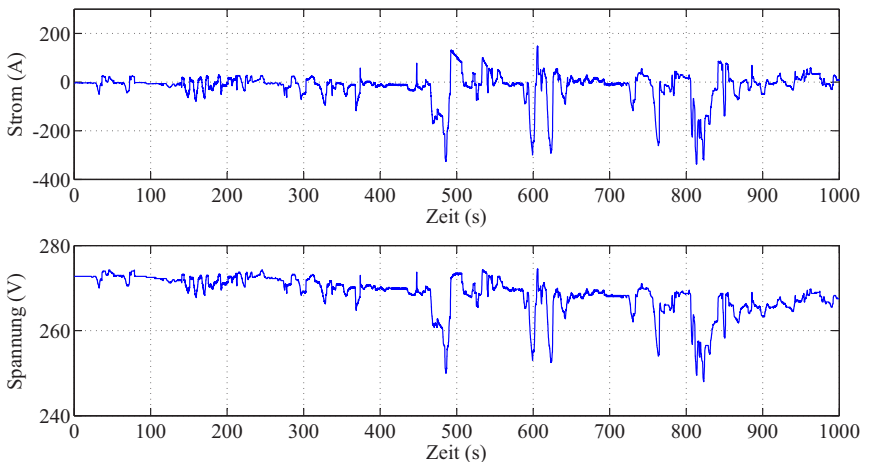
Für eine Quantifizierung der Simulationsergebnisgüte im noch unbekanntem Extrapolationsbereich werden wieder die Batteriemodelle aus Kapitel 5.4.1 eingesetzt. Als Datenquelle werden jedoch andere CAN-Traces verwendet, deren Einteilung in Tabelle 6.4 dargestellt ist.

Darin enthalten die ersten beiden Gruppen 1 und 2 (engl. *Group 1*, *Group 2*) Messdaten aus ein und demselben CAN-Trace. Diese werden der Gütequantifizierung mit jeweils unterschiedlichen Abstraten zur Verfügung gestellt, um den Einfluss der Anzahl der Messpunkte im Strafterm des BIC gegenüber dem AIC bewerten zu können. Der CAN-Trace stammt

Tab. 6.4: Einteilung der Trainings- und Extrapolationsdatensätze nach [NETTER 2013b, S.4]

<i>Simulation</i>	<i>Sampling rate [Hz]</i>	<i>Time Range Training Sample [s]</i>	<i>Time Range Extrapolation Sample [s]</i>
Group 1	1	0-1000	1000-2982
Group 2	10	0-1000	1000-2982
Group 3	1		0-2311
	1		0-1200
	1		0-1689

wieder von einem A1 e-tron aus dem Pilotprojekt, überspannt aber im Gegensatz zu den bisher verwendeten Ausschnitten eine komplette Fahrt mit einer Fahrtdauer von knapp 50 Minuten. Davon wird der Zeitbereich von 0 – 1000 s als Trainingsbereich zum Anlernen der Simulationsmodelle verwendet. Der Zeitbereich zwischen 1000 – 2982 s dient als Extrapolationsbereich, um die Anpassungsgüte jenseits des bekannten Bereichs bewerten zu können. Die Strom- und Spannungsverläufe der verwendeten Messdaten sind in Abbildung 6.5 für den Trainingsbereich und in Abbildung 6.6 für den Extrapolationsbereich veranschaulicht.

**Abb. 6.5:** Datensatz zum Training der freien Parameter nach [CHU 2012, S.54]

Um die Allgemeingültigkeit der angelegten Modelle abzutesten, werden weitere drei Extrapolationsdatensätze verwendet. Diese sind in Tabelle

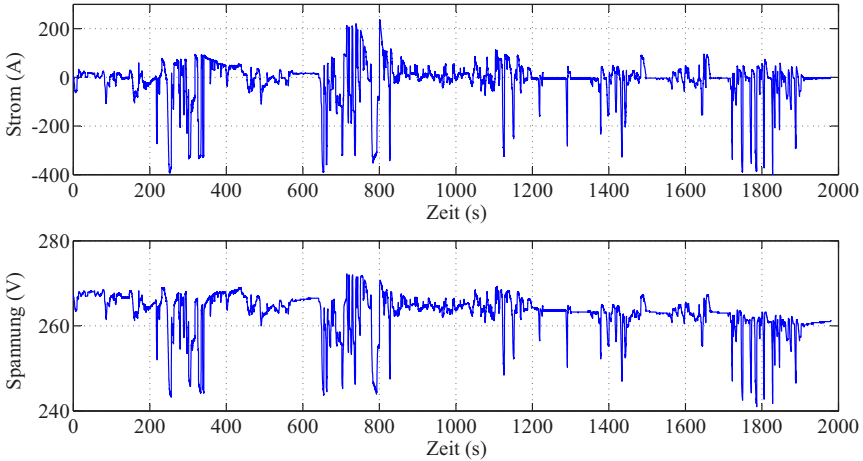


Abb. 6.6: Datensatz zur Plausibilisierung der Parameteridentifikation im Extrapolationsbereich

6.4 in Gruppe 3 (engl. *Group 3*) eingetragen. Die Daten stammen aus drei weiteren Fahrzeugen des gleichen Typs und sind damit der initialen Parameteridentifikation unbekannt [NETTER 2013b, S.4].

In diesem Fall eines linearen Regressionsmodells der Hochvoltbatterie wird die Quantifizierung nun mit den alternativen Formeln für AIC und BIC aus den Gleichungen 6.8 bzw. 6.23 durchgeführt, wodurch allgemein Rechenzeit eingespart werden kann, da die Likelihood-Funktion nicht erstellt werden muss. Entsprechend wird auf eine Erfassung des MDL, deren Berechnung das MLE *Estimate* voraussetzt, verzichtet. Die Ergebnisse dieser Quantifizierung sind in Tabelle 6.5 aufgelistet.

Die Werte des MSE sind darin generell um ein Vielfaches höher als in den Tabellen 6.2 und 6.3 bei der Gütequantifizierung der Teilsystemsimulationen in Kapitel 6.4.2. Somit konnten die Batteriemodelle die Datensätze dort sehr viel exakter approximieren. Dies ist insofern plausibel, als dass der Datensatz für die Komplexitätsquantifizierung der Batterie aus Abbildung 5.10 deutlich homogener, also glatter ist als die Datensätze für die Gütequantifizierung aus den Abbildungen 6.5 und 6.6 und damit weniger „Spitzen“ durch das Polynom abgebildet werden müssen. Dementsprechend sind die Residuen hier größer, weswegen auch der MSE höher ausfällt. Die aus den Ergebnissen resultierenden Verläufe für die

Tab. 6.5: Ergebnisse der nach ihrer Simulationsergebnisgüte quantifizierten Batterie-
modelle

Hz	Typ	Training				Extrapol.
		MSE	$n * \ln(MSE)$	AIC	BIC	MSE
1	R_i	0,510	-673,34	-671,34	-666,44	1,036
	RC_1	0,040	-3211,65	-3205,65	-3190,93	0,090
	RC_2	0,037	-3293,87	-3383,87	-3259,33	0,062
	RC_3	0,035	-3343,87	-3329,87	-3295,52	0,061
10	R_i	0,513	-6679,47	-6677,47	-6670,26	1,045
	RC_1	0,056	-28859,81	-28853,81	-28832,18	0,096
	RC_2	0,051	-29767,14	-29757,14	-29721,09	0,066
	RC_3	0,051	-29762,83	-29748,83	-29698,35	0,068

Quantifizierungsmerkmale der Gruppe 1 sind sowohl für den Trainings- als auch den Extrapolationsbereich bei einer Abtastfrequenz von einem Herz in Abbildung 6.7 dargestellt.

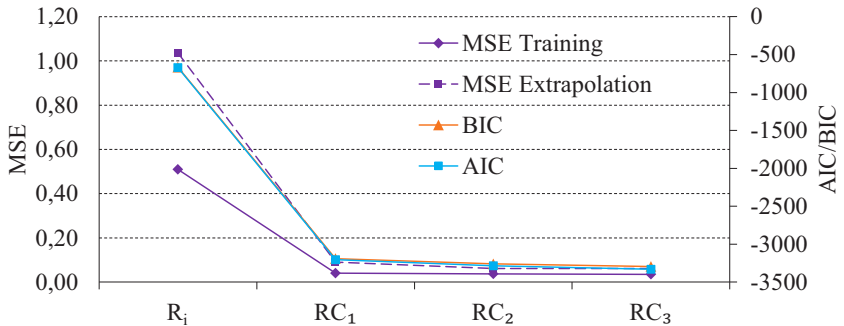


Abb. 6.7: Verläufe der Simulationsergebnisgüte für Gruppe 1 bei 1 Hz nach [NETTER 2013b, S.5]

Darin fällt im Trainingsbereich bei „MSE Training“ die Abweichung zwischen den Messdaten und den Simulationsergebnissen vom R_i -Modell zum RC_1 -Modell um mehr als 90 %. Ab dem RC_1 -Modell fällt der Verlauf um schrittweise kleiner werdende Differenzen weiter. Da somit die Abweichungen zwischen den gemessenen und den simulierten Daten im Trainingsbereich tendenziell sinken, kann daraus eine steigende Anpassungsgüte mit steigender Komplexität gefolgert werden. Analog zum Evaluationsbeispiel aus Kapitel 6.4.2 bringt das erste RC-Element die größte Verbesserung. Allerdings sinken im Gegensatz dazu auch die Verläufe für AIC und BIC.

Diese fallen, wie auch der MSE, vom R_i -Modell zum RC_1 -Modell sehr deutlich um knapp 80 % und danach um schrittweise kleiner werdende Differenzen weiter. Dementsprechend kann entgegen den bisherigen Trends von einem steigenden Extrapolationsvermögen mit steigender Komplexität ausgegangen werden. Infolgedessen verfügen die Batteriemodelle mit vielen Parametern auf Basis der hier verwendeten Daten auch über mehr Potential, Daten im noch unbekanntem Extrapolationsbereich abzubilden. Dieses Potential wird durch den Verlauf der „MSE *Extrapolation*“ bestätigt, der die Abweichungen der im Trainingsbereich angelesenen Simulationsmodelle gegenüber den noch unbekanntem Daten im Extrapolationsbereich repräsentiert. Die damit ermittelten Abweichungen sinken ebenfalls vom R_i -Modell zum RC_1 -Modell um mehr als 90 %. Somit steigt die Anpassungsgüte der komplexeren Modelle, weswegen diese Modelle tatsächlich mehr Potential aufweisen, Daten im noch unbekanntem Extrapolationsbereich abzubilden [NETTER 2013b, S.5]. Hintergrund für dieses Verhalten sind auch hier die sehr inhomogenen Verläufe der für die Gütequantifizierung verwendeten Datensätze aus den Abbildungen 6.5 und 6.6, welche von einem sehr einfachen Modell nur unzureichend abgebildet werden können. Dies ist insofern plausibel, als dass mit dem trainierten RC_1 -Modell bei noch unbekanntem Daten im Extrapolationsbereich mit einem um 80 % kleinerem MSE gegenüber dem R_i -Modell die Daten sehr viel exakter nachgebildet werden können. Dieses Potential steigt mit dem RC_2 -Modell gegenüber dem RC_1 -Modell nochmals um 30 % und ab diesem nur noch geringfügig.

In Referenz auf Abbildung 6.2 sinkt demnach der Vorhersagefehler, ausgehend von dem R_i -Modell mit zu geringem Extrapolationsvermögen. Ab diesem nähert sich die Simulationsmodellkomplexität dem Bereich der Überanpassung, ohne diesen zu erreichen bzw. zu überschreiten. Der fallende Trend des Kalibrierungsfehlers deutet für das R_i -Modell auf eine relativ zu geringe Komplexität des verwendeten Simulationsmodells hin. Relativ dazu weisen die anderen Simulationsmodelle ab dem RC_2 -Modell einen vergleichbar niedrigen Kalibrierungsfehler auf. Demnach bieten für die vorliegenden Daten das RC_2 -Modell und das RC_3 -Modell das größte Potential für eine hohe Anpassungsgüte bei gleichzeitig hohem Extrapolationsvermögen.

Rechnerisch wird das Verhalten des Extrapolationsvermögens durch einen dominanten ersten Term in den Gleichungen für das AIC bzw. BIC gewährleistet, dessen Werte in Tabelle 6.5 aufgelistet sind. Diese sind in Relation zu den Werten des Evaluationsbeispiels aus Kapitel 6.4.2 sehr

viel größer als die jeweiligen Strafterme, weswegen letztere kaum Einfluss auf das Ergebnis haben. Dementsprechend gering sind damit in diesem Beispiel die Auswirkungen durch die Erhöhung der Frequenz und damit eine Erhöhung der Anzahl an Messpunkten n , weswegen die Abbildung für Gruppe 2 bis auf den unterschiedlichen Wertebereich der Ordinate nahezu identisch ist (siehe dazu [NETTER 2013b, S.5]). Generell nimmt beim BIC gegenüber dem AIC durch die zusätzliche Wertung der Anzahl an Messpunkten n der relative Einfluss des Strafterms zu. In diesem Beispiel ist dieser Einfluss des Strafterms beim BIC bei allen Simulationsmodellen knapp viermal so groß wie beim AIC. Die Ergebnisse der noch ausstehenden Gruppe 3 sind in Abbildung 6.8 dargestellt.

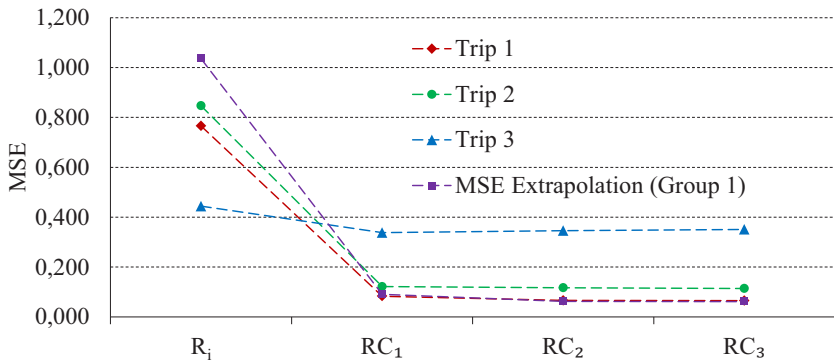


Abb. 6.8: Verläufe der Simulationsergebnisgüte für Gruppe 3 bei 1 Hz nach [NETTER 2013b, S.6]

Darin stimmen die Trends der Verläufe, die aus zusätzlichen Extrapolationsdatensätzen weiterer Fahrzeuge resultieren, mit dem des Referenzverlaufs „MSE Extrapolation“ der Gruppe 1 aus Abbildung 6.7 überein. Tatsächlich können bei diesen Beispielen die mit Hilfe des Trainingsdatensatzes der Gruppe 1 angelegten Batteriesimulationsmodelle die unbekanntenen Daten der Fahrten 1 und 2 teilweise sogar exakter approximieren als den eigenen Extrapolationsdatensatz der „MSE Extrapolation“. „Fahrt 3“ weist beim R_i -Modell eine höhere Anpassungsgüte auf als der Referenzverlauf der „MSE Extrapolation“, jedoch bringt bereits das RC_1 -Modell kaum eine Verbesserung. Ab diesem Modell steigt der Trend leicht an, was ein Indiz dafür ist, dass sich die komplexeren Modelle basierend auf den Daten der „Fahrt 3“ bereits im Bereich der Überanpassung befinden.

7 Komplexitätsadaption integrierter Gesamtsystemsimulationen

„Da die Erstellung eines Modells nach unterschiedlichen Gesetzmäßigkeiten erfolgen kann, (...) kann es beliebig viele Modelle eines Systems geben. Von entscheidender Bedeutung ist dabei die Fragestellung, nach der das betreffende System mit den (...) Modellen untersucht werden soll“ [KÜBLER 2000, S.15].

In den letzten beiden Kapiteln wurden verschiedene Verfahren vorgestellt, mit deren Hilfe sowohl die Simulationsmodellkomplexität als auch deren Potential für eine hohe Anpassungsgüte bei gleichzeitig hohem Extrapolationsvermögen quantifiziert werden kann. Im Rahmen dessen wurden konkrete Attribute identifiziert und anhand verschiedener Evaluationsbeispiele appliziert. Damit ist es möglich, bei der Erstellung eines FMUs die Quantifizierungsmerkmale im Prozessschritt „*Modelling*“ zu bestimmen.

Für eine Beurteilung der für eine Simulation benötigten Simulationsmodellkomplexität und der relevanten Simulationsergebnisgüte liefert, gemäß dem einleitenden Zitat aus dem Buch „*Modulare Modellierung und Simulation mechatronischer Systeme*“, die Fragestellung, auf die das Modell hin untersucht werden soll, das erforderliche Entscheidungskriterium. Anhand derer kann mit den Attributen als Stellgrößen die Komplexität integrierter Gesamtsystemsimulationen an einen solchen Simulationszweck adaptiert werden. Um bei hohem Ertrag, also hoher Ergebnisgüte, eine hohe Wirtschaftlichkeit sicherzustellen, muss gemäß dem Minimumprinzip der Aufwand, also der Ressourceneinsatz, um die hohe Ergebnisgüte zu erreichen, minimiert werden. Damit liefert das daraus resultierende FMU in Abhängigkeit von dem Simulationszweck einerseits ein möglichst genaues Abbild der Realität und nimmt andererseits nur den Grad an Komplexität

in Anspruch, der nötig ist, um die wichtigsten Kausalitäten und Merkmale der beobachteten Daten abzubilden.

Für eine solche Komplexitätsadaption integrierter Gesamtsystemsimulationen werden zunächst exemplarische Simulationszwecke aus dem Bereich der Kraftfahrzeugtechnik vorgestellt. Basierend darauf wird eine Ontologie erstellt, die die Simulationszwecke mit den identifizierten Quantifizierungsmerkmalen verbindet. Das daraus resultierende Potential wird abschließend anhand eines virtuellen Prototyps demonstriert.

7.1 Identifikation von Simulationszwecken

Simulationszwecke beschreiben die Zielsetzung einer Simulation, die stark von den jeweiligen Disziplinen abhängt. So werden z. B. in der Astrophysik Lebenszyklen von Galaxien studiert, in der Geophysik die Entstehung von Erdbeben, in der Chemie die Metabolismen von Bakterien, in der Medizin die Ausbreitung von Seuchen und in der Architektur die Statik von Bauwerken [BUNGARTZ 2009, S.2]. Den bisherigen Evaluationsbeispielen folgend, wird im Rahmen dieser Arbeit von einer Simulation aus der Kraftfahrzeugtechnik ausgegangen. Für diese Fachdisziplin werden die in Abbildung 7.1 im äußeren Kreis dargestellten Simulationszwecke verfolgt.

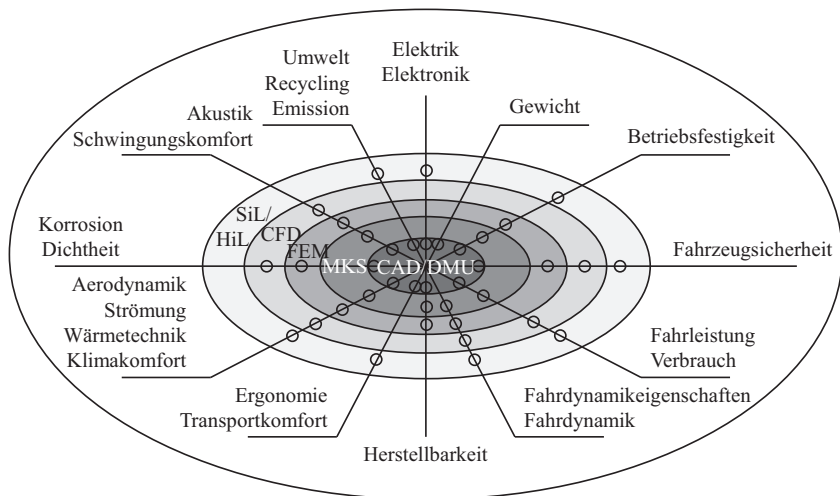


Abb. 7.1: Simulationszwecke in der Automobilindustrie nach [SEIFFERT 2008, S.359]

Die Schichten in der Mitte stellen die Simulationsmethoden dar, für die in der Regel unterschiedliche Simulatoren zum Einsatz kommen. Um das jeweilige Gesamtsystem darzustellen, bedient man sich darin des *Computer-Aided Design* (CAD) bzw. des DMU. Für die Simulation wird zwischen einer Mehrkörpersimulation (MKS), einer Finite Elemente Methode (FEM) Simulation, einer *Computational Fluid Dynamics* (CFD) Simulation und Simulationen in geschlossener Schleife wie *Software in the Loop* (SiL) bzw. *Hardware in the Loop* (HiL) unterschieden. Die kleinen schwarzen Kreise deuten an, welche Simulationsarten für die jeweiligen Simulationszwecke zum Einsatz kommen.

7.2 Ontologiebasierte Komplexitätsadaption

Im Rahmen dieser Arbeit wird eine Ontologie eingesetzt, um die Abhängigkeiten der identifizierten Attribute unter Berücksichtigung des eingangs erwähnten Prinzips der Sparsamkeit von verschiedenen Simulationszwecken abzubilden. Im Folgenden wird zunächst der Begriff der Ontologie erläutert. Anschließend wird eine Ontologie aufgebaut, mit deren Hilfe die Simulationsmodellkomplexität an den jeweiligen Simulationszweck angepasst werden kann.

7.2.1 Definition der Ontologie

Der Begriff „Ontologie“ stammt aus der Philosophie und bezeichnet die Lehre vom Sein. Dabei bedeutet das aus dem Griechischen stammende „*to on*“ das Sein, also das Allgemeine, das man von einem Gegenstand sagen kann [BUSSE 2012, S.6]. Damit fragt die Ontologie nach den grundsätzlichen und allgemeinen Strukturen der Wirklichkeit. Diese Denkmuster wurden in den 90er Jahren in die Informatik übernommen. Darin beschreiben sie Fachterminologien, um Inhalte und Zusammenhänge, also die Semantik, zu erfassen und für die rechnergestützte Verarbeitung zugänglich zu machen. Ziel ist es dabei, Wissen über ein bestimmtes Gebiet formal abzubilden und damit austauschbar und wiederverwendbar zu machen. Zu diesem Zweck werden relevante Begriffe aus der jeweiligen Domäne in einer Modellierungssprache definiert. Da diese Begriffe auf formaler Logik basieren, können sogenannte Inferenzmechanismen genutzt werden, um bereits modelliertes Wissen in Form von logischen Relationen auf Gültigkeit und Konsistenz

abzuprüfen und implizit modellierte Abhängigkeiten explizit zu machen [HESS 2009, S.46]. Ontologien setzen sich dabei aus Klassen, Individuen und Eigenschaften zusammen.

Eine Klasse bzw. ein Konzept fasst darin gleiche Individuen unter einem Begriff zusammen. Mehrere Klassen werden gewöhnlich in sogenannten Taxonomien, also einem hierarchischen Klassifikationsschema, strukturiert [HESS 2009, S.46]. Analog zur Systemhierarchie aus Kapitel 2.1.2 wird auch darin zwischen Ober- und Unterklassen (engl. *Super- and Subclasses*) unterschieden. Mit Hilfe der Klassenhierarchie und der Unterklassenbeziehung ist es möglich, den gesamten Gegenstandsbereich detailliert zu umfassen. So kann man z. B. das Buch als Unterklasse von Printmedien definieren, welche auch Zeitschrift als Unterklasse besitzt. Diese Art der Wissensmodellierung ist vergleichsweise intuitiv, da sie sich sehr nahe am menschlichen Denken in Begriffen orientiert [HITZLER 2008, S.71f]. Individuen sind dabei Elemente einer Klasse und repräsentieren eine bestimmte Ausprägung, indem sie die Klasse instanziierten [HESS 2009, S.46]. Dementsprechend werden diese auch als „Instanzen“ bezeichnet. Eine Klasse kann explizit beschrieben werden, indem man sie benennt und damit eine benannte Klasse definiert (engl. *Named Class*). Die Definition kann auch implizit anhand von Bedingungen erfolgen, die von Individuen erfüllt werden müssen, damit sie zu der Klasse gehören können. Diese werden als „Eigenschaftseinschränkungen“ bezeichnet (engl. *Property Restrictions*).

Eigenschaften werden für Klassen definiert. Diese werden auch als „Relationen“ bezeichnet und genutzt, um Individuen einer Klasse zu definieren [HESS 2009, S.46]. Je nachdem, ob Relationen mit anderen Individuen oder mit Daten bestehen, wird zwischen Objekt- und Dateneigenschaften unterschieden. Objekteigenschaften (engl. *Object Properties*) sind binäre Relationen zwischen zwei Individuen [HORRIDGE 2009, S.11]. Im Gegensatz dazu beschreiben Dateneigenschaften (engl. *Data Properties*) Beziehungen zwischen Individuen und Daten eines bestimmten Typs. Eigenschaften können verschiedene Merkmale aufweisen. In Abbildung 7.2 sind exemplarisch funktionelle, transitive und symmetrische Merkmale dargestellt.

Ist eine Eigenschaft z. B. als „funktionell“ definiert, so kann nur ein einziges Individuum der Zielklasse durch eine Relation entlang dieser Eigenschaft mit einem Individuum der Startklasse verbunden sein. Wie in der linken Abbildung aus 7.2 dargestellt, kann z. B. das Individuum „Julian“ der Klasse „Kind“ nur eine Beziehung entlang der Eigenschaft „hat Mutter“

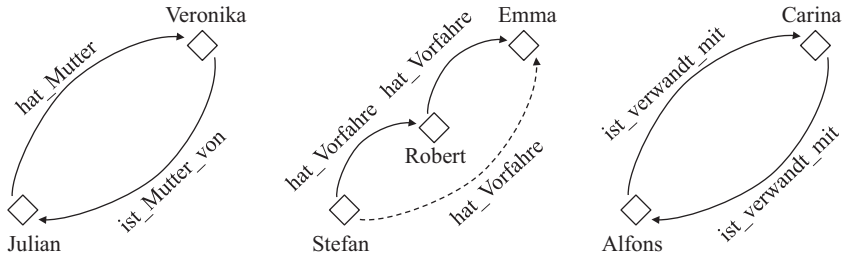


Abb. 7.2: Unterscheidung in funktionelle, transitive und symmetrische Merkmale nach [HORRIDGE 2009, S.29f]

haben und zwar das Individuum „Veronika“ der Klasse „Mutter“. Diese Eigenschaft kann auch invers funktionell zur Eigenschaft „ist Mutter von“ sein. Darüber hinaus können Eigenschaften auch transitiv sein. Sobald ein Individuum mit einem anderen Individuum entlang einer solchen transitiven Eigenschaft verbunden wird und letzteres über dieselbe Eigenschaft mit einem weiteren Individuum verbunden ist, kann daraus eine Abhängigkeit des ersten von dem letzten Individuum gefolgert werden. Dieser Sachverhalt ist im mittleren Teil von Abbildung 7.2 dargestellt. Falls z. B. das Individuum „Stefan“ den Vorfahren „Robert“ hat und dieser wiederum „Emma“ als Vorfahren aufweist, hat automatisch auch „Stefan“ „Emma“ als Vorfahren. Transitive Eigenschaften können nicht funktionell sein. Allerdings sollte das Inverse einer transitiven Eigenschaft ebenfalls transitiv sein. Des Weiteren existieren auch symmetrische Eigenschaften. Sobald eine Eigenschaft symmetrisch ist und die Eigenschaft ein Individuum mit einem anderen verbindet, ist letzteres Individuum automatisch über dieselbe Eigenschaft mit dem ersten Individuum verbunden. Falls, wie in der rechten Abbildung aus 7.2 dargestellt, z. B. „Alfons“ über die Eigenschaft „ist verwandt mit“ mit „Carina“ verbunden wird, ist „Carina“ über die gleiche Eigenschaft auch mit „Alfons“ verbunden. Wenn diese Eigenschaft nicht erwünscht ist, kann die Relation auch als asymmetrisch definiert werden [HORRIDGE 2009, S.29f]. Analog zu Klassen lassen sich auch Eigenschaften hierarchisch definieren, woraus die Möglichkeit resultiert, Ober- und Untereigenschaften (engl. *Super- and Subproperties*) zu spezifizieren. So bezeichnet z. B. die Eigenschaft „ist Veganer“ eine Untereigenschaft der Relation „ist Vegetarier“, da Veganer lediglich eine Teilmenge der Vegetarier darstellen.

7.2.2 Aufbau einer Ontologie

Für die Übersetzung der natürlichen Sprache in eine formale Sprache existieren diverse Ontologie-Modellierungssprachen wie das *Resource Description Framework* (RDF) und die *Web Ontology Language* (OWL), innerhalb derer mit Hilfe verschiedener Entwicklungsumgebungen der jeweilige Betrachtungsgegenstand implementiert werden kann. Im Rahmen dieser Arbeit wurde der Editor Protégé verwendet, welcher am Institut für Medizinische Informatik an der Stanford University in Kalifornien entwickelt worden ist. Die Inferenzmechanismen, die in formalen Sprachen bereits modelliertes Wissen in Form von logischen Relationen auf Gültigkeit und Konsistenz prüfen und implizit modellierte Abhängigkeiten explizit machen können, werden in Protégé als „*Reasoner*“ bezeichnet. Die Struktur der umgesetzten Gesamtsystemarchitektur ist in Abbildung 7.3 dargestellt.

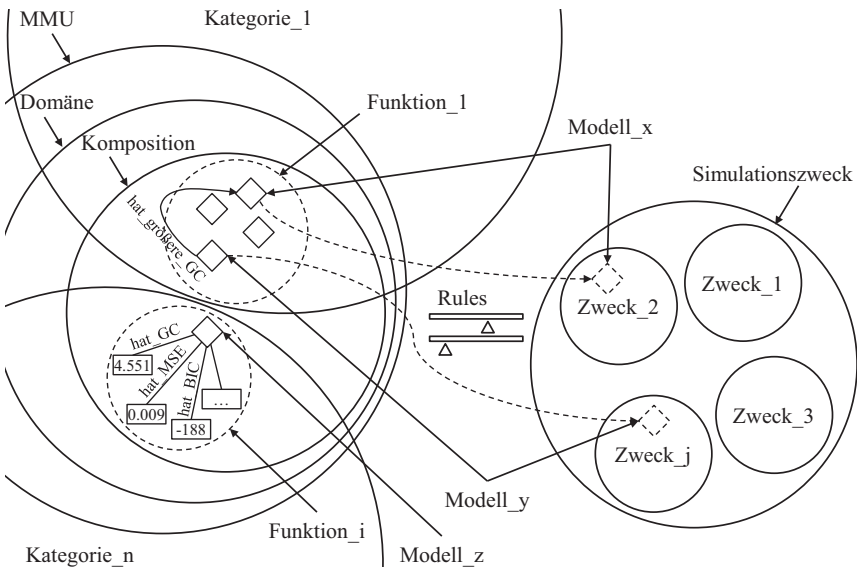


Abb. 7.3: Struktur der Ontologie der Gesamtsystemarchitektur

Der Hierarchie für die Strukturierung technischer Gesamtsysteme aus Abbildung 4.1 folgend, wird von dem „*Maximal Mock-Up*“ als oberster Ebene ausgegangen. Da innerhalb der Ontologie sämtliche Funktionen verschiedener Fahrzeuge abgebildet werden und damit kein projektspezifischer

Verbund betrachtet wird, wird die Ebene des „*Mock-Up*“ übersprungen. Als Teilmengen der Klasse „*Maximal Mock-Up*“ wird somit die „Domäne“ als Unterklasse definiert, unterhalb derer wiederum, wie in Abbildung 7.3 veranschaulicht, die Unterklasse „Komposition“ angeordnet wird. Zur Repräsentation der Funktionen und ihrer Kategorien werden keine weiteren Unterklassen unterhalb der Kompositionen eingerichtet, da sonst in jeder Komposition für jede Kategorie eine eigene Unterklasse definiert werden müsste. Dies hätte eine Vielzahl an Relationen und damit eine unübersichtliche Ontologie zur Folge. Stattdessen wird „Kategorie“ als weitere Oberklasse festgelegt, innerhalb derer die Kategorien des Modularisierungsansatzes aus Kapitel 4.1 als Unterklassen abgebildet werden. Diese sind in der linken Hälfte von Abbildung 7.3 schematisch als große Teilkreise dargestellt, da sie sämtliche Hierarchieelemente in Untermengen aufteilen. Funktionen resultieren somit aus der kombinierten Betrachtung von Komposition und jeweiliger Kategorie, weswegen für dieses Element keine eigene Klasse benötigt wird. Funktionen sind dementsprechend in Abbildung 7.3 lediglich gestrichelt angedeutet.

Innerhalb der Unterklassen der jeweiligen Kategorien werden die quantifizierten Simulationsmodelle als Individuen eingetragen und über Relationen den jeweiligen Kompositionen zugeordnet. Ein und dieselben Modelle unterschiedlicher Attributierung werden infolgedessen als Instanzen einer Kategorie interpretiert. Diese sind in der linken Hälfte von Abbildung 7.3 für „Kategorie 1“ exemplarisch als „Modell x“ und „Modell y“ dargestellt. Den Simulationsmodellen werden abschließend die in den Kapiteln 5 und 6 identifizierten Quantifizierungsmerkmale für Komplexität, Anpassungsgüte und Extrapolationsvermögen als Dateneigenschaften vom Typ „*Double*“ zugewiesen. Dabei handelt es sich um funktionelle Eigenschaften, da in jedes Simulationsmodell jeweils nur ein Wert für PC, SC, GC, MSE, AIC, BIC und MDL eingetragen werden kann. In Abbildung 7.3 sind für das Modell „Modell z“ exemplarisch GC, MSE und BIC entlang der invertierbaren Dateneigenschaften „hat GC“, „hat MSE“ und „hat BIC“ skizziert. Die übrigen Attribute werden analog dazu ergänzt. Zusätzlich werden zwischen den Individuen die Objekteigenschaften „hat größere PC“, „hat größere SC“ und „hat größere GC“ eingeführt. Damit werden die Simulationsmodelle innerhalb der Ontologie-Datenbasis relativ zueinander in Abhängigkeit von der jeweiligen Komplexität sortiert. Bildlich wird dadurch, gemäß der Beziehung zwischen Vorhersagefehler und Kalibrierungsfehler in Abbildung

6.2 aus Kapitel 6.1, die Abszisse für die Komplexitätsadaption definiert. Darin wird bewusst noch keine Entscheidung für eines der Merkmale für die Komplexitätsquantifizierung getroffen, da dies dem Anwender überlassen werden soll. Die Beschreibung des bereits als Evaluationsbeispiel verwendeten R_i -Modells ist in Abbildung 7.4 dargestellt.

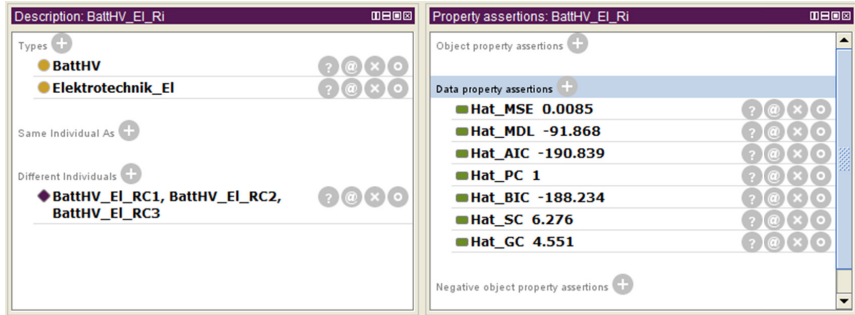


Abb. 7.4: Beschreibung des R_i -Modells (Protégé-Screenshot)

Um die Individuen einem Simulationszweck zuordnen zu können, wird „Simulationszweck“ als weitere Oberklasse eingeführt. Darunter werden als Unterklasse verschiedene „Zwecke“ eingerichtet, welche in der rechten Hälfte von Abbildung 7.3 dargestellt sind. Über Relationen wird dabei einerseits mit Hilfe von Kategorien und Kompositionen festgelegt, welche Funktionen für welchen Simulationszweck benötigt werden. Andererseits werden über sogenannte *Rules* jedem Simulationszweck nur diejenigen Individuen zugeordnet, die für den jeweiligen Simulationszweck geeignet sind. So erfordert z. B. die Funktion der „BattHV“ der Kategorie „Elektrotechnik“ für den Simulationszweck „Verbrauch“ eine hohe Komplexität, wohingegen sie für den Simulationszweck „Fahrleistung“ eine untergeordnete Rolle spielt und durch ein Modell geringerer Komplexität abgebildet werden kann. Dafür werden mit Hilfe der *Semantic Web Rule Language* (SWRL) Schwellwerte für die Komplexitätsquantifizierungsmerkmale in Protégé implementiert. Solche SWRL-Regeln sind in der Mitte von Abbildung 7.3 veranschaulicht. Zusätzlich kann auch der Aspekt der Simulationsergebnisgüte mit berücksichtigt werden, indem ebenfalls über diese Regeln ein erforderlicher Bereich für die Attribute der Anpassungsgüte und des Extrapolationsvermögens vorgegeben wird. Dazu wird analog den Evaluationsbeispielen aus den Kapiteln 6.4.2 und 6.4.3 innerhalb jeder Funktion ein optimaler Bereich identifiziert

und die daraus resultierenden Schwellwerte ebenfalls über die SWRL in Protégé implementiert. Dieser Bereich kann unter Umständen auch mehrere optimale Simulationsmodelle überspannen. Bildlich wird damit, gemäß der Beziehung zwischen Vorhersagefehler und Kalibrierungsfehler in Abbildung 6.2 aus Kapitel 6.1, die Ordinate beschrieben.

Der eingangs erwähnte *Reasoner* stellt abschließend die Gültigkeit und Konsistenz sicher, indem geprüft wird, ob alle Relationen wie eingegeben dargestellt werden können. Zusätzlich ordnet er die Modelle aus der Ontologie-Datenbasis in Abhängigkeit von den Relationen und SWRL-Regeln den Simulationszwecken zu. Eine solche Zuweisung ist in Abbildung 7.3 exemplarisch für die beiden Modelle „Modell x“ und „Modell y“ anhand der gestrichelten Linien veranschaulicht. Die Ergebnisse dieser Zuweisung können über ein Visualisierungs-Plug-in dargestellt werden. In Abbildung 7.5 sind sie für den Simulationszweck „Energieverbrauch“ mit Hilfe des Protégé-Plug-ins OntoGraf dargestellt.

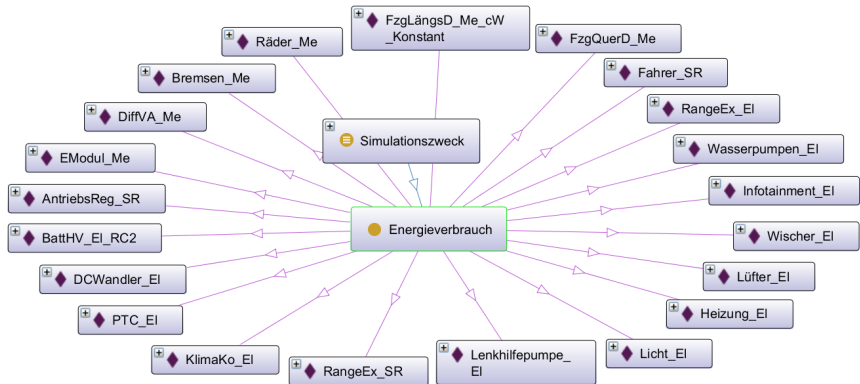


Abb. 7.5: OntoGraf-Darstellung des Simulationszwecks „Energieverbrauch“ (Protégé-Screenshot)

Somit können die Attribute sämtlicher Simulationsmodelle, die im Prozessschritt „*Modelling*“ für ein FMU erstellt worden sind, nach ihrer Quantifizierung in der Ontologie-Datenbasis abgelegt werden. Referenziert man das Ergebnis der Zuweisung auf die jeweils zugrunde liegende Gesamtfahrzeugarchitektur, welche dafür im Prozessschritt des „*System Design*“ erstellt worden ist, wird ersichtlich, ob für alle Funktionen entsprechende Simulationsmodelle die Kriterien erfüllt haben und vorgeschlagen werden.

Fehlen einzelne Funktionen, müssen gegebenenfalls die SWRL-Regeln angepasst werden oder weitere Modelle implementiert werden, welche den gewünschten Anforderungen an Komplexität, Anpassungsgüte und Extrapolationsvermögen entsprechen. Erfüllen mehrere Modelle einer Funktion die genannten Kriterien, hat der Anwender die Wahl oder kann die SWRL-Regeln gegebenenfalls enger fassen.

7.3 Demonstration anhand einer Gesamtfahrzeugsimulation

Um die Komplexitätsadaption einem breiten Anwendungsbereich zur Verfügung zu stellen, wird die Ontologie in den Prozess der Co-Simulation aus Kapitel 3.3.2 integriert. Das daraus resultierende erweiterte Verfahren wird abschließend anhand der Gesamtfahrzeugsimulation eines A1 e-tron für den Simulationszweck „Energieverbrauch“ durchgeführt.

7.3.1 Integration einer Ontologie in den Prozess der Co-Simulation

Als weitere Möglichkeit der Konfiguration einer Gesamtsystemarchitektur wird die Ontologie, der Darstellung des Co-Simulationsprozesses aus Abbildung 3.6 folgend, in den Prozessschritt „*Configuration*“ eingebunden. Dazu wird die Ontologie-Datenbasis in die Co-Simulationsdatenbasis abgelegt und dort auch gepflegt, wodurch die Attribute der Simulationsmodelle und die Schwellwerte der *Rules* allen Nutzern gleichermaßen zugänglich sind. Initial werden sämtliche Simulationsmodelle in der Co-Simulationsdatenbasis mit den vorgestellten Verfahren bezüglich ihrer Komplexität, ihrer Anpassungsgüte und ihres Extrapolationsvermögens quantifiziert. Damit stehen im Prozessschritt „*Configuration*“ sämtliche Attribute der in der Co-Simulationsdatenbasis enthaltenen Simulationsmodelle als Metainformationen zur Verfügung. Funktionen, für die noch keine Simulationsmodelle im MMU abgelegt sind und dementsprechend im Zuge der MU-Erstellung implementiert werden, werden nach dem Approvement im Prozessschritt „*Integration*“ ebenfalls quantifiziert.

In Ergänzung zu dem bisherigen Schritt „*Configuration*“ besteht dadurch die Möglichkeit neben der bisherigen Anpassung der Implementierung nun

auch die Komplexität der Gesamtsystems simulation an den jeweiligen Simulationszweck anzupassen. Dabei steht es dem Anwender frei, über die Auswahl eines vorgegebenen Simulationszwecks von der Ontologie einen Vorschlag der jeweils optimalen Simulationsmodelle abzurufen oder selbst auf Basis der Metainformationen zwischen dem Potential verschiedener Simulationsmodelle zu unterscheiden und dadurch selbstständig eine Auswahl zu treffen. Die Ergebnisse dieser Auswahl werden in der *Functional-Mock-Up*-Beschreibungsdatei gespeichert und stehen damit den nachfolgenden Prozessschritten zur Verfügung.

7.3.2 Durchführung einer Gesamtfahrzeugsimulation

Um das erweiterte Verfahren zu demonstrieren, wird eine komplexitätsadaptierte integrierte Gesamtfahrzeugsimulation auf Basis eines FMUs des A1 e-tron durchgeführt. Dafür wird exemplarisch von der in Kapitel 4.4.1 für den Simulationszweck „Energieverbrauch“ aufgebauten Gesamtfahrzeugmodellarchitektur in der Ausbaustufe II mit virtuellem Fahrer und virtuellem Umfeld ausgegangen. Diese ist in den Abbildungen C.1 und C.2 im Anhang dargestellt. Mit Hilfe des gewählten Layouts können die real befahrenen Strecken aus dem A1 e-tron-Pilotprojekt in München in einem virtuellen Umfeld reproduziert und von einem virtuellen Fahrer mit dem virtuellen Fahrzeug befahren werden. Dem Prozess der Co-Simulation aus Abbildung 3.6 folgend, wird innerhalb dieser Architektur im Prozessschritt „*Modelling*“ für jede Funktion, für die noch keine Simulationsmodelle im MMU enthalten sind, ein Modell in dem dafür jeweils am besten geeigneten Simulationswerkzeug implementiert. Die Anpassungsgüte der neuen Modelle wird über das in Kapitel 4.3 vorgestellte Verfahren zur Absicherung technischer Gesamtsysteme mit Hilfe der im Pilotprojekt aufgezeichneten Daten abgeprüft und Modelle mit großer Abweichung werden nachgebessert. Anschließend werden sämtliche Modelle mit Hilfe der in den Kapiteln 5 und 6 identifizierten Quantifizierungsmerkmale für Komplexität, Anpassungsgüte und Extrapolationsvermögen bewertet. Die daraus resultierenden zusätzlichen Modelle und Werte werden in die Ontologie eingetragen und bei konkurrierenden Modellen innerhalb einer Funktion das optimale Modell bzw. die optimalen Modelle identifiziert. Deren Schwellwerte werden in die SWRL integriert und das *Reasoning* wird durchgeführt. Nach der Parametrisierung mit den Globalparametern wird die Simulation gestartet.

Darin synchronisiert eine neutrale Kopplungssoftware die Zeitschrittweiten sämtlicher involvierter Simulatoren. In diesem Demonstrationsbeispiel wird TISC von TLK Thermo eingesetzt. Dabei handelt es sich um eine *Middleware* mit Client/Server-Umgebung, die es ermöglicht, verschiedene Berechnungsmodelle, z. B. aus MATLAB, Kuli, Dymola oder Flowmaster zu einem Gesamtsystem zu organisieren. Dabei agiert der TISC-Server als Vermittler zwischen den einzelnen Modellen bzw. Clients und sorgt dafür, dass alle Teilnehmer zeitsynchron Informationen austauschen. Dadurch wird gewährleistet, dass die Simulationsmodelle nicht inkohärent simulieren. Die Kommunikation zwischen Client und Server wird über Netzwerkprotokolle realisiert [BURGER 2012, S.25]. Das Beispiel der laufenden Simulation für das Demonstrationsbeispiel ist in Abbildung 7.6 dargestellt.

	Description	IP	Port	Receives	Sends	Synchronization	Sync Rate	Last Time	Total Time	Sim Speed	Version
1	ADTF Connection	127.0.0.1	53295	25 variables	13 variables	Sync Signal	0.02 s	00:00:00,025	00:00:00,020	1.250	2.7
2	Antrieb	10.251.145.225	53300	10 variables	9 variables	Sync Signal	0.001 s	00:00:00,006	00:00:00,163	6.000	2.5
3	Aufbau	10.251.145.225	53297	24 variables	24 variables	Sync Signal	0.001 s	00:00:00,003	00:00:00,046	3.000	2.5
4	Elektrik	10.251.145.225	53298	27 variables	3 variables	Sync Signal	0.001 s	00:00:00,001	00:00:00,035	1.000	2.5
5	Fahrer	10.251.145.225	53299	8 variables	5 variables	Sync Signal	0.001 s	00:00:00,002	00:00:00,032	2.000	2.5
6	Fahrwerk	10.251.145.225	53301	11 variables	26 variables	Sync Signal	0.001 s	00:00:00,003	00:00:00,053	3.000	2.5
7	RB	10.251.145.225	53296	1 variables	26 variables	Sync Signal	0.001 s	00:00:00,001	00:00:00,042	1.000	2.5

Abb. 7.6: Aktive Co-Simulation bei der Gesamtfahrzeugsimulation (TISC-Screenshot)

Bei dem in der obersten Zeile der Abbildung genannten *Automotive Data and Time-Triggered Framework* (ADTF) handelt es sich um ein *Framework* der *Audi Electronics Venture GmbH* (AEV), welches eine Entwicklungs- und Testumgebung für neue Fahrzeugfunktionen, wie z. B. Abstandsregeltempomaten und Spurhalteassistenten, bereitstellt. Die „ADTF Connection“ ermöglicht dabei als einer der Clients die Kopplung mit der virtuellen Umfeldsimulation und dem virtuellen Umfeld.

Als Umfeldsimulation wird *Virtual Test Drive* (VTD) der Firma VIRESS Simulationstechnologie GmbH genutzt. Diese Software wird für die Ausführung und Darstellung von Verkehrssimulationen genutzt. Innerhalb dieser Umgebung können reale Strecken reproduziert und von einem virtuellen Fahrzeug befahren werden. Ein *Image Generator* übernimmt die Visualisierung der 3D-Szene. Der *Dynamics Manager* steuert das in VTD enthaltene Fahrer- und Fahrzeugmodell. Die Visualisierung der Gesamtfahrzeugsimulation in dieser Umgebung ist in Abbildung 7.7 dargestellt.

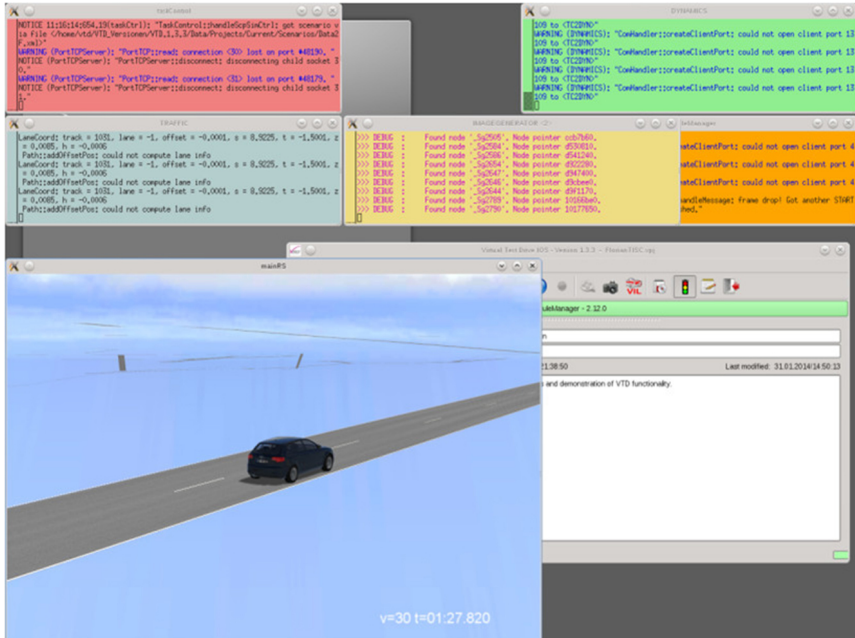


Abb. 7.7: Laufende Umfeldsimulation bei der Gesamtfahrzeugsimulation (VTD-Screenshot)

Das Fahrermodell wird ebenfalls von VTD bereit gestellt. Es ist in der Lage, das virtuelle Umfeld zu interpretieren und damit das Fahrzeug autonom durch die Simulationslandschaft zu navigieren. Es kann durch verschiedene Parameter an reales Fahrverhalten angepasst werden. Je nach Konfiguration beschleunigt und verzögert der Fahrer anders, wodurch unterschiedliche Verbrauchswerte für den Simulationszweck „Energieverbrauch“ provoziert werden können.

Bei dem Fahrzeugmodell in VTD handelt es sich grundsätzlich um ein physikalisches Modell, welches das Verhalten des Fahrzeugs z.B. beim Bremsen in der Kurve korrekt simulieren kann. Im Rahmen dieser Arbeit wurde das Fahrzeugmodell durch die erstellte Gesamtfahrzeugsimulation ersetzt, wodurch dieses beliebig modifiziert werden kann.

Für die gekoppelte Simulation von Fahrzeug, virtuellem Umfeld und virtuellem Fahrer werden die in den einzelnen Domänen innerhalb der Gesamtfahrzeugsimulation berechneten Größen über TISC und ADTF an

VTD und zurück ausgetauscht. In dem dargestellten Demonstrationsbeispiel aus Abbildung 7.6 werden 13 Variablen an VTD gesendet und 25 Variablen erhalten. Diese entsprechen den Signalen aus den Abbildungen C.1 und C.2 im Anhang. In den Zeilen zwei bis sieben aus Abbildung 7.6 sind die Domänen der Gesamtfahrzeugsimulation als weitere Teilnehmer aufgelistet. Diese interagieren sowohl miteinander als auch mit dem virtuellen Umfeld und dem virtuellen Fahrer.

Mit Hilfe der erläuterten Konfiguration kann zum einen z. B. durch Soll-Geschwindigkeitsvorgabe aus den aufgezeichneten Daten das reale Fahrverhalten reproduziert werden, wodurch die Ergebnisse der Gesamtsystems simulation mit Hilfe der aufgezeichneten Daten plausibilisiert werden können. Zum anderen können damit auch virtuell Modifikationen am Fahrzeug durchgeführt werden. Deren Auswirkungen auf Fahrverhalten und Fahrzeugperipherie können nun mit den aufgezeichneten Geschwindigkeitsverläufen, realen Fahrzyklen und unterschiedlichen Fahrertypen auf Basis von Simulationsmodellen mit hoher Anpassungsgüte und gleichzeitig hohem Extrapolationsvermögen analysiert werden.

8 Zusammenfassung und Ausblick

Um für simulatorübergreifende integrierte Gesamtsystemsimulationen aus einer Vielzahl konkurrierender Nachbildungen von Teilsystemen entscheiden zu können, welche Simulationsmodelle in Abhängigkeit von dem Simulationszweck für welche Teilsysteme am besten geeignet sind, wurden dedizierte Kennzahlen identifiziert und appliziert. Die daraus resultierenden Attribute wurden in einen bestehenden Prozess zur Kopplung simulatorübergreifender Systeme integriert, wodurch diese einem breiten Anwendungsbereich zur Verfügung gestellt werden können. Um darin die Systemgrenzen und Schnittstellen sämtlicher Teilsysteme unter Berücksichtigung der Vielfalt an physikalischen und funktionalen Interdependenzen zu definieren, wurde für technische Systeme ein skalierender Modularisierungsansatz entwickelt. Das Potential der Attribute und des Modularisierungsansatzes wurde anhand mehrerer Evaluationsbeispiele analysiert und bewertet. Mit Hilfe eines virtuellen Prototyps wurde abschließend exemplarisch durch die Verwendung einer Ontologie die Verknüpfung der Attribute und damit die Anpassung der Simulationsmodellkomplexität an einen damit zu erfüllenden Simulationszweck demonstriert.

8.1 Zusammenfassung

Mit den Grundlagen der Modellbildung und Simulation wurde in Kapitel 2 zunächst ein einheitliches Verständnis für die in den nachfolgenden Kapiteln verwendeten Begriffe vermittelt. Dabei wurde auf die Eigenschaften von Signalen und Systemen eingegangen und die Systemhierarchie eingeführt. Darauf aufbauend wurde der Prozess der Modellbildung und Simulation technischer Systeme erläutert, um die theoretischen Grundlagen der Entstehung und der computergestützten Ausführung von Simulationsmodellen

vorzustellen. Besonderes Augenmerk galt der Parameteridentifikation, mit der Simulationsmodelle an beobachtete Daten angepasst werden. Darin wurden mit der Methode der maximalen Wahrscheinlichkeit und der Methode der kleinsten Quadrate zwei Verfahren vorgestellt, die elementar für das Verständnis der in dieser Arbeit angewendeten Quantifizierungsverfahren sind. Um den Prozess der Modellbildung und Simulation abzuschließen, wurden Verfahren zur Absicherung der Simulationsmodelle erläutert.

Kapitel 3 erweiterte diese Grundlagen um Aspekte, die für die Modellbildung und Simulation simulatorübergreifender Systeme benötigt werden. Darin wurde auf die verschiedenen Kopplungsvarianten eingegangen und deren Vor- und Nachteile wurden beleuchtet. Damit wurde die Entscheidungsgrundlage für die im Rahmen dieser Arbeit eingesetzte Variante der verteilten Simulation geliefert. Deren Methoden zur Modularisierung und Kopplung wurden infolgedessen näher betrachtet und damit der Einsatz der schwachen Kopplung oder auch Co-Simulation begründet. Für diese Kopplungsvariante wurde abschließend der Stand der Technik zum Aufbau simulatorübergreifender Systeme vorgestellt, wobei auch auf einen bereits bestehenden Prozess eingegangen wurde, auf den der später vorgestellte Modularisierungsansatz aufsetzt.

Um die Systemgrenzen und Schnittstellen sämtlicher Teilsysteme aufeinander abzustimmen und damit den Aufbau eines virtuellen Prototyps zu koordinieren, wurden in Kapitel 4 durch die Strukturierung und Absicherung technischer Gesamtsysteme die dafür benötigten Strukturelemente entwickelt. Zu diesem Zweck wurde ein Ansatz zur Modularisierung bei gleichzeitig vollständiger Erfassung technischer Gesamtsysteme vorgestellt. Dieser wurde für die Strukturierung technischer Systeme appliziert und damit, durch die Definition einer Gesamtsystemarchitektur und einer Signalnamenskonvention, die Grundlagen für eine simulatorübergreifende integrierte Gesamtsystems simulation gelegt. Darauf aufbauend wurden Maßnahmen vorgestellt, um bei der Erstellung eines FMUs bereits in frühen Phasen der virtuellen Produktentwicklung die Struktur der Gesamtsystemarchitektur absichern zu können. Abschließend wurden die vorgestellten Definitionen und Maßnahmen anhand der Gesamtfahrzeugmodellarchitektur eines A1 e-tron evaluiert. Als Ergebnis konnte mit dem gewählten Modularisierungsansatz das Fahrzeug für den Simulationszweck „Energieverbrauch“ vollständig erfasst werden. Darüber hinaus konnte die Fahrzeugmodellarchitektur mit geringem Aufwand beliebig skaliert werden, um bestehende Teilsysteme

zu modifizieren bzw. zu entfernen oder weitere zu integrieren. Darüber hinaus konnte mit Hilfe eines Konsistenz-Checks die Durchgängigkeit der erstellten Gesamtfahrzeugmodellarchitektur abgesichert werden, indem überprüft wurde, ob sämtliche Eingänge der Teilsysteme mit Ausgängen anderer Teilsysteme oder einer Parameterdatei verbunden sind. Auf diese Weise wurde die grundsätzliche Simulierbarkeit einer auf dieser Architektur basierenden Gesamtsystemsimulation gewährleistet. Um darüber hinaus in den späteren Prozessschritten die Anpassungsgüte der dann eingebundenen Simulationsmodelle auf Basis von Messdaten bewerten zu können, wurden die Teilsysteme zu Funktionsgruppen aggregiert, um sicherzustellen, dass sämtliche Schnittstellen mit Messdaten beaufschlagt werden können.

Kapitel 5 behandelte dann die mathematischen Grundlagen, die zur Bewertung der Komplexität und damit der Effizienz von Simulationsmodellen benötigt wurden. Zu diesem Zweck wurde darin der Begriff der Komplexität definiert, um ein gemeinsames Verständnis für die weiteren Schritte aufzubauen. Basierend darauf wurden mit der parametrischen Komplexität, der stochastischen Komplexität und der geometrischen Komplexität Merkmale eingeführt, die eine Quantifizierung der Simulationsmodellkomplexität ermöglichen. Um diese für Simulationsmodelle technischer Systeme anwenden zu können, wurde die dafür benötigte Annahme einer Gaußschen Normalverteilung der zugrunde liegenden Messdaten zunächst mit Hilfe eines Linearmodells bei voneinander unabhängiger Messung eingeführt. Diese Annahme wurde anschließend anhand eines Simulationsmodells um Aspekte bei voneinander abhängiger Messung erweitert und der Zusammenhang zwischen der Methode der maximalen Wahrscheinlichkeit und der Methode der kleinsten Quadrate hergeleitet. Das Potential der Attribute für eine Komplexitätsquantifizierung wurde abschließend anhand verschieden komplexer Implementierungen zweier exemplarischer Simulationsmodelle aus dem A1 e-tron evaluiert. Im Resultat konnten die objektiven Quantifizierungsmerkmale den subjektiven Eindruck einer steigenden Komplexität bei beiden Modellen bestätigen. Allerdings erfordern die dafür verwendeten Verfahren parametrierbare Modelle, damit eine Quantifizierung durchgeführt werden kann. Darüber hinaus hängen die Ergebnisse der Komplexitätsquantifizierung von den dafür verwendeten Daten ab. Dementsprechend bedingen die vorgestellten Verfahren, dass Simulationsmodelle einer Funktion stets auf Basis der gleichen Datensätze quantifiziert werden, so dass deren Komplexität relativ zueinander bewertet werden kann.

Der Aspekt der Effektivität und damit die Güte der Simulationsergebnisse, sowohl gegebene als auch unbekannte Daten abzubilden, wurde in Kapitel 6 auf Basis geeigneter Quantifizierungsmerkmale erfasst. Zu diesem Zweck wurden mit der Anpassungsgüte und dem Extrapolationsvermögen in Ergänzung zu der bereits quantifizierten Komplexität zwei weitere Attribute der Simulationsmodelle eingeführt, mit deren Hilfe die Simulationsergebnisgüte erfasst werden kann. Analog Kapitel 5 wurden auch diese zunächst definiert, bevor dafür Merkmale identifiziert wurden. Das Potential der Attribute wurde auch hier abschließend evaluiert. Dazu wurde mit Hilfe der Funktionsgruppen zunächst die generelle Anpassungsgüte sämtlicher Simulationsmodelle der Gesamtfahrzeugmodellarchitektur aus Kapitel 4.4 quantifiziert. Anschließend wurde die eigentliche Gütequantifizierung anhand der bereits bekannten Beispiele aus Kapitel 5.4 durchgeführt und über deren Trends die Wirkmechanismen der Quantifizierungsmerkmale erläutert. Im Ergebnis steigt in allen vorgestellten Evaluationsbeispielen die Anpassungsgüte mit steigender Komplexität. Das Extrapolationsvermögen hängt bei den verwendeten Beispielen stark von der Homogenität bzw. Inhomogenität des jeweils verwendeten Datensatzes ab und unterscheidet sich mitunter auch durch das jeweils verwendete Quantifizierungsmerkmal. Die Standardformeln bedingen auch hier parametrierbare Modelle. Darüber hinaus muss die Quantifizierung der Simulationsergebnisgüte einer Funktion analog zur Simulationsmodellkomplexität auf Basis der gleichen Datensätze durchgeführt werden, um eine relative Bewertung zu gestatten. Für das AIC und das BIC existieren alternative Formeln, mit deren Hilfe das Extrapolationsvermögen ohne Likelihood-Funktion bestimmt werden kann. Die in den Evaluationsbeispielen erzielten Resultate sind durchweg plausibel und decken gemäß Abbildung 6.2 den kompletten Bereich von „*Too simple*“ bis zu „*Too complex*“ ab. Das Extrapolationsvermögen liefert auch jenseits der für das Parametertraining verwendeten Datensätze gute Prognosen. In der Summe liefern Komplexität, Anpassungsgüte und Extrapolationsvermögen die Möglichkeit, aus einer Vielzahl konkurrierender Simulationsmodelle das bzw. die Modelle zu identifizieren, die den besten Kompromiss hinsichtlich Effizienz und Effektivität gewährleisten.

Durch die Verwendung einer Ontologie wurde in Kapitel 7 die Verknüpfung der Attribute und damit die Anpassung der Simulationsmodellkomplexität an den damit zu erfüllenden Simulationszweck ermöglicht. Über Relationen wurden dafür die Simulationsmodelle in Abhängigkeit von ihrer

Notwendigkeit für einen Simulationszweck mit unterschiedlichen Zwecken verbunden. Zusätzlich wurden über Regeln die jeweils für einen Simulationszweck geforderten und optimalen Wertebereiche in eine Ontologie eingearbeitet, die damit die „Intelligenz“ besitzt, die dafür am besten geeigneten Modelle zu selektieren. Um die dadurch mögliche Komplexitätsadaptation einem breiten Anwendungsbereich zur Verfügung zu stellen, wurde die Ontologie in den Prozessschritt „*Configuration*“ aus dem Prozess der Co-Simulation integriert. Darin steht es dem Anwender nun frei, über die Auswahl eines vorgegebenen Simulationszwecks von der Ontologie einen Vorschlag der jeweils optimalen Simulationsmodelle abzurufen oder selbst auf Basis der Attribute in Form von Metainformationen zwischen dem Potential verschiedener Simulationsmodelle zu unterscheiden und dadurch selbstständig eine Auswahl zu treffen. Zudem wurde bewusst noch keine Entscheidung für eines der jeweils drei Merkmale für die Komplexitätsquantifizierung bzw. das Extrapolationsvermögen getroffen, da diese von der Akzeptanz der Anwender abhängig gemacht werden soll. Eine erste Umsetzung einer mit diesen Verfahren erstellten gekoppelten Gesamtfahrzeugmodellsimulation wurde abschließend anhand eines A1 e-tron unter dem Simulationszweck „Energieverbrauch“ demonstriert.

8.2 Ausblick

Mit Hilfe des vorgestellten Ansatzes wird gewährleistet, dass für die Simulation ausschließlich optimale Simulationsmodelle in Abhängigkeit von dem Simulationszweck selektiert werden können. Allerdings kann es in den frühen Phasen der virtuellen Produktentwicklung für erste Prognosen ausreichend sein, sehr viel einfachere Modelle mit geringer Anpassungsgüte und geringem Extrapolationsvermögen einzusetzen. Darüber hinaus bedingen die fest eingetragenen Schwellwerte einen hohen Pflegeaufwand, da bei der Integration weiterer Simulationsmodelle in die Co-Simulationsdatenbasis stets der optimale Bereich erneut identifiziert und die Werte in der Ontologie angepasst werden müssen.

Um das Potential des vorgestellten Ansatzes voll ausschöpfen zu können, wäre es zielführend, die Ermittlung einer optimalen Simulationsmodellkonfiguration in ein externes Programm auszulagern. In Abhängigkeit von der jeweiligen Phase des virtuellen Produktentstehungsprozesses wären

darin die Schwellwerte nicht mehr statisch vorgegeben, sondern könnten dynamisch angepasst werden. So könnten in den frühen Phasen der virtuellen Produktentwicklung für erste Prognosen des FMUs die Modelle nach einem hohen Extrapolationsvermögen bei geringer Anpassungsgüte selektiert werden, wohingegen in den späten Phasen, z. B. zur Applikation von Regelungsalgorithmen, eine hohe Anpassungsgüte in den Vordergrund tritt. Darüber hinaus kann eine Logik entwickelt werden, die bei einer Vielzahl konkurrierender Modelle innerhalb einer Funktion den optimalen Bereich der Attribute automatisiert identifiziert. Die Speicherung der statischen Modellattribute und die ebenfalls statische Zuweisung der Modelle zu den unterschiedlichen Simulationszwecken könnten weiterhin in der Ontologie erfolgen, um über den *Reasoner* stets die Konsistenz der Datenbasis zu gewährleisten.

A Anwendung der theoretischen Modellbildung

Einem Beispiel aus dem Buch „Regelungstechnik 1“ [LUNZE 2013, S.62ff] folgend soll im Folgenden das physikalische Ersatzmodell eines Feder-Masse-Schwingers als Demonstrator für die theoretische Modellbildung erstellt werden. Das System ist in Abbildung A.1 dargestellt. Auf Basis eines daraus resultierenden konzeptuellen Modells wird ein ausführbares Modell erzeugt, indem durch Umformung die physikalischen Beziehungen in eine Differentialgleichung des Systems überführt werden. Diese wird in einem letzten Schritt in die Form einer Zustandsraumdarstellung gebracht.

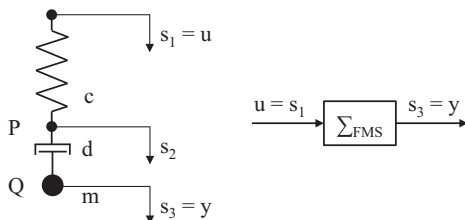


Abb. A.1: Feder-Masse-Schwinger nach [LUNZE 2013, S.63]

A.1 Erstellung eines physikalischen Ersatzmodells

In Abbildung A.1 entsprechen $s_1(t)$, $s_2(t)$ und $s_3(t)$ den Positionen der darin skizzierten Punkte. Es wird angenommen, dass das System nur Bewegungen entlang der vertikalen Achse ausführen kann. Der Schwinger wird durch eine von außen erzwungene Bewegung am oberen Ende der Feder angeregt, weswegen $s_1(t)$ die Eingangsgröße des Systems darstellt.

Beobachtet wird die Position $s_3(t)$ der Masse m am unteren Ende der Feder, die die Ausgangsgröße des Systems darstellt. Die Federkonstante c und die Dämpfung d sind Parameter des Systems. Es wird davon ausgegangen, dass die Größen $s_1(t)$, $s_2(t)$ und $s_3(t)$ so gemessen werden, dass in der Ruhelage $s_i(0) = 0$, ($i = 1, 2, 3$) gilt. Die Erdanziehung wird im Folgenden vernachlässigt. Gemäß dem Newtonschen Gesetz ist die Beschleunigung der Masse m proportional zur Summe der äußeren Kräfte F_m :

$$F_m(t) = m\ddot{s}_3(t) \quad (\text{A.1})$$

Die durch den Dämpfer übertragene Kraft $F_d(t)$ wird bei geschwindigkeitsproportionaler Dämpfung durch die Relativbewegung der beiden Enden des Dämpfers und die Dämpfungskonstante bestimmt:

$$F_d(t) = d(\dot{s}_2(t) - \dot{s}_3(t)) \quad (\text{A.2})$$

Die Federkraft $F_c(t)$ resultiert aus den Positionen der Federenden und der Federkonstanten:

$$F_c(t) = c(s_1(t) - s_2(t)) \quad (\text{A.3})$$

Für die Kopplung der drei Gleichungen erhält man aus den Kräftegleichgewichten an den Punkten P und Q zwei algebraische Beziehungen. Zum einen entspricht unter Vernachlässigung der Erdanziehung die auf die Masse wirkende Kraft $F_m(t)$ am Punkt Q der Kraft $F_d(t)$ vom Dämpfer:

$$F_m(t) = F_d(t) \quad (\text{A.4})$$

Zum anderen gleicht bei P die Federkraft $F_c(t)$ der Dämpferkraft $F_d(t)$:

$$F_d(t) = F_c(t) \quad (\text{A.5})$$

A.2 Implementierung eines ausführbaren Modells

Die Gleichungen A.1 bis A.5 aus Kapitel A.1 beschreiben das Feder-Masse-System vollständig. Auf Basis dieser konzeptuellen Beschreibung kann nun ein ausführbares Modell erstellt werden, indem durch Umformung

die physikalischen Beziehungen in die Differentialgleichung des Systems überführt werden. Ziel der Umformung ist es, die Ausgangsgröße $y(t) = s_3(t)$ in Abhängigkeit von der Eingangsgröße $u(t) = s_1(t)$ darzustellen. Setzt man A.5 in A.4 ein erhält man:

$$F_m(t) = F_c(t) \quad (\text{A.6})$$

Einsetzen von A.1 und A.3 in Gleichung A.6 und Division durch m ergibt:

$$\ddot{s}_3(t) = \frac{c}{m} (s_1(t) - s_2(t)) \quad (\text{A.7})$$

Analog resultiert aus dem Einsetzen von A.1 und A.2 in Gleichung A.4, Division durch d und Umstellung der Gleichung folgender Ausdruck:

$$\dot{s}_2(t) = \frac{m}{d} \dot{s}_3(t) + \dot{s}_3(t) \quad (\text{A.8})$$

Um nun $s_2(t)$ aus Gleichung A.7 zu eliminieren, wird Gleichung A.8 ein Mal nach der Zeit integriert:

$$s_2(t) - s_2(0) = \frac{m}{d} \dot{s}_3(t) + s_3(t) - \left(\frac{m}{d} \dot{s}_3(0) + s_3(0) \right) \quad (\text{A.9})$$

Damit diese Gleichung für alle Zeitpunkte t erfüllt ist, gilt analog den vorab aufgestellten Bedingungen in der Ruhelage $s_i(0) = 0, (i = 1, 2, 3)$:

$$s_2(t) = \frac{m}{d} \dot{s}_3(t) + s_3(t) \quad (\text{A.10})$$

Eingesetzt in Gleichung A.7 resultiert die gesuchte Differentialgleichung:

$$\ddot{s}_3(t) = \frac{c}{m} s_1(t) - \frac{c}{m} \frac{m}{d} \dot{s}_3(t) - \frac{c}{m} s_3(t) \quad (\text{A.11})$$

Durch Umstellen erhält man die in Kapitel 2.2.1 und Gleichung 2.7 definierte Standardform der Differentialgleichung:

$$\frac{m}{c} \ddot{s}_3(t) + \frac{m}{d} \dot{s}_3(t) + s_3(t) = s_1(t) \quad (\text{A.12})$$

Die Konstanten entsprechen dabei folgenden Werten:

$$a_2 = \frac{m}{c}, \quad a_1 = \frac{m}{d}, \quad a_0 = 1, \quad b_0 = 1 \quad (\text{A.13})$$

A.3 Überführung einer Differentialgleichung in die Zustandsraumdarstellung

Als Darstellungsform wird gegenüber der Darstellung in Form einer gewöhnlichen Differentialgleichung häufig das Zustandsraummodell mit dem Zustandsvektor x bevorzugt. Grund dafür ist, dass es sich bei dem dadurch beschriebenen Systemzustand um eine ingenieurtechnisch interpretierbare Größe handelt, die sich darüber hinaus gut für die rechnergestützte Verarbeitung eignet.

A.3.1 Theoretische Grundlagen

Das Zustandsraummodell besteht aus einer Menge von r Differentialgleichungen erster Ordnung, die aus gewöhnlichen Differentialgleichungen erstellt werden können. Es liefert damit eine äquivalente mathematische Beschreibung des Systems [LUNZE 2013, S.72]. Die Zustandsgleichung, dargestellt in Formel A.14a, wird auch als „explizite Darstellung von Differentialgleichungen erster Ordnung“ bezeichnet. Nicht in allen Fällen ist die Beschreibung einer Differentialgleichung ausreichend. Oft müssen bei der Berechnung zusätzliche Bedingungen berücksichtigt werden, die eine algebraische Verknüpfung der Zustandsgrößen beinhalten. Ein solches System wird auch „Differential-Algebraisches System“ (engl. Differential Algebraic Equations (*DAE*)) genannt und hat in der expliziten nichtlinearen Zustandsraumdarstellung die in Gleichung A.14a und A.14b dargestellte Form [DRONKA 2004, S.16]. Die Nebenbedingungen sind in der Ausgangsgleichung A.14b enthalten:

$$\dot{x}(t) = f(x(t), u(t)), \quad x(t_0) = x_0 \quad (\text{A.14a})$$

$$y(t) = g(x(t), u(t)) \quad (\text{A.14b})$$

Handelt es sich um eine lineare Differentialgleichung, kann die Darstellung der linearen Übertragungsglieder auch durch die explizite lineare Zustandsraumdarstellung erfolgen. Diese hat dann für Eingrößensysteme, also Systeme mit nur einem Eingang und einem Ausgang, folgende Form [LUNZE 2013, S.74]:

$$\dot{x}(t) = Ax(t) + bu(t), \quad x(t_0) = x_0 \quad (\text{A.15a})$$

$$y(t) = c^T x(t) + du(t) \quad (\text{A.15b})$$

Zu diesem Zweck wird die in Kapitel 2.2.1 vorgestellte gewöhnliche Differentialgleichung aus Gleichung 2.7 durch die Multiplikation mit $\frac{1}{a_r}$ nach der höchsten Ableitung von $y(t)$ normiert. Anschließend werden sämtliche Ableitungen von $y(t)$ durch die Zustandsvariable $x(t)$ substituiert. Dadurch wird die Differentialgleichung r -ter Ordnung durch r Differentialgleichungen erster Ordnung ersetzt, weshalb für deren Berechnung auch r Integrationen benötigt werden. Die daraus resultierende Zustandsraumdarstellung besteht aus der konstanten r, r -Matrix A , auch als „Begleitmatrix“ oder „Frobeniusmatrix“ bezeichnet, dem r -dimensionalen Eingangsvektor b , dem r -dimensionalen transponierten Ausgangsvektor c^T und dem Durchgangsfaktor d . Diese lassen sich gemäß [LUNZE 2013, S.85] für Differentialgleichungen mit identischer Ordnung $q = r$ des Ein- und Ausgangssignals in folgender Form zusammenfassen:

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 1 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{r-1} \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad (\text{A.16a})$$

$$c^T = (b_0 - b_r a_0 \quad b_1 - b_r a_1 \quad \dots \quad b_{r-1} - b_r a_{r-1}), \quad d = b_r \quad (\text{A.16b})$$

Für $q < r$ können die Elemente der Ausgangsgleichung wie folgt vereinfacht werden:

$$c^T = (b_0 \quad b_1 \quad \dots \quad b_q \quad 0 \quad \dots \quad 0), \quad d = 0 \quad (\text{A.17})$$

Die soeben beschriebene Zustandsraumdarstellung lässt sich gemäß Abbildung A.2 auch graphisch als Zustandsraummodell darstellen.

Dabei handelt es sich um ein vom jeweiligen Anwendungsfall unabhängiges Blockschaltbild, welches den Signalfussplan für ein lineares System r -ter Ordnung mit r Differentialgleichungen erster Ordnung darstellt. Wie aus

der Graphik ersichtlich wird nur ein Integrator benötigt, der das Eingangs- und Ausgangssignal verbindet. Die Signalflüsse von Matrizen und Vektoren werden durch Doppellinien dargestellt.

Bei Mehrgrößensystemen, also Systemen mit mehreren Eingängen und mehreren Ausgängen, wird der einspaltige Eingangsvektor zur Eingangsmatrix und der einzeilige Ausgangsvektor zur Ausgangsmatrix. Falls keine explizite Lösung der Differentialgleichungen angegeben werden kann, können indirekte Verfahren wie die Verfahren zur Stabilitätsuntersuchung oder Näherungslösungen durch Linearisierung des Systems angewendet werden [HILLER 1983, S.2].

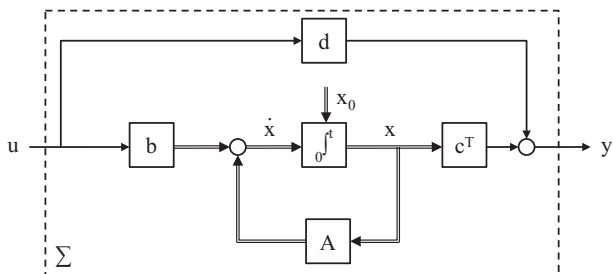


Abb. A.2: Zustandsraummodell eines Eingrößensystems nach [LUNZE 2013, S.76]

A.3.2 Applikation auf das physikalische Ersatzmodell

Um aus der im Kapitel A.2 abgeleiteten Differentialgleichung des Feder-Masse-Schwingers die Zustandsform zu generieren, wird die Differentialgleichung aus Gleichung A.12 mit $\frac{1}{a_2} = \frac{c}{m}$ normiert [LUNZE 2013, S.86]:

$$\ddot{s}_3(t) + \frac{c}{d} \dot{s}_3(t) + \frac{c}{m} s_3(t) = \frac{c}{m} s_1(t) \quad (\text{A.18})$$

Die hier ermittelten Konstanten nehmen analog der Standardform folgende Werte an:

$$a_2 = 1, \quad a_1 = \frac{c}{d}, \quad a_0 = \frac{c}{m}, \quad b_0 = \frac{c}{m} \quad (\text{A.19})$$

Damit ergibt sich mit Hilfe der Gleichungen A.15 bis A.17 für $q < r$ folgende Zustandsform des Feder-Masse-Schwingers:

$$\dot{x}(t) = \begin{pmatrix} 0 & 1 \\ -\frac{c}{m} & -\frac{c}{d} \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(t) \quad (\text{A.20})$$

$$y(t) = \begin{pmatrix} \frac{c}{m} & 0 \end{pmatrix} x(t) \quad (\text{A.21})$$

Dabei entspricht die Zustandsvariable:

$$x_i(t) = \frac{m}{c} \begin{pmatrix} y(t) \\ \dot{y}(t) \end{pmatrix} \quad (\text{A.22})$$

Dementsprechend beschreibt die Zustandsvariable eine mit dem Faktor $\frac{m}{c}$ multiplizierte Position bzw. Geschwindigkeit der Masse des Feder-Masse-Schwingers [LUNZE 2013, S.86].

B Durchführung einer Parameteridentifikation

Um die Modellbildung abzuschließen, müssen die Parameter des Modells (wie z. B. die reellwertigen Koeffizienten a_i und b_i der gewöhnlichen Differentialgleichung aus Kapitel 2.2.1) bestimmt werden, mit deren Hilfe das Simulationsmodell konfiguriert bzw. an beobachtete Daten angepasst werden kann. Zur Parameteridentifikation werden im Folgenden mit der Methode der maximalen Wahrscheinlichkeit und der Methode der kleinsten Quadrate zwei im Rahmen dieser Arbeit eingesetzte Verfahren exemplarisch angewendet.

B.1 Parameteridentifikation mit der Methode der höchsten Wahrscheinlichkeit

Zur Parameteridentifikation mit der Methode der maximalen Wahrscheinlichkeit muss zunächst bei jeweils festgelegten Modellparameter die dazu gehörige Wahrscheinlichkeitsverteilung ermittelt werden. Um unter all den daraus resultierenden Wahrscheinlichkeitsverteilungen diejenige zu identifizieren, die die beobachteten Daten am wahrscheinlichsten erzeugt hat, werden anschließend die Rollen des Datenvektors und des Parametervektors vertauscht, woraus die Likelihood-Funktion resultiert. Diese wird abschließend graphisch dargestellt und erläutert.

B.1.1 Erstellung einer Wahrscheinlichkeitsverteilung

Die Erstellung von Wahrscheinlichkeitsverteilungen soll im Folgenden in Anlehnung an ein Experiment aus dem Zeitschriftenaufsatz „*Tutorial on Maximum Likelihood Estimation*“ [MYUNG 2003, S.91] am Beispiel einer Urne mit insgesamt $i = 10$ schwarzen und weißen Kugeln veranschaulicht

werden. Die Anzahl θ der schwarzen Kugeln ist unbekannt und soll geschätzt werden. Zu diesem Zweck werden ein Mal nacheinander $n = 5$ Kugeln gezogen, darunter zwei schwarze und drei weiße.

Die Kugeln werden nach dem Ziehen jeweils wieder in die Urne zurückgelegt. Dementsprechend handelt es sich um einen sogenannten n -fachen Bernoulli-Versuch bzw. eine Bernoulli-Kette der Länge n , also um eine gleichartige und unabhängige Versuchsserie mit jeweils genau zwei möglichen Ergebnissen. Daraus resultiert eine Binomialverteilung, die in folgender allgemeinen Form dargestellt werden kann:

$$f(z|n, \theta) = \frac{n!}{z!(n-z)!} \frac{\theta^z}{i^z} \left(1 - \frac{\theta}{i}\right)^{n-z}, \quad (0 \leq \theta \leq i; z = 0, 1, \dots, n) \quad (\text{B.1})$$

Für den hier betrachteten Fall der $n = 5$ Kugeln gilt:

$$f(z|n = 5, \theta) = \frac{5!}{z!(5-z)!} \frac{\theta^z}{i^z} \left(1 - \frac{\theta}{i}\right)^{5-z}, \quad (0 \leq \theta \leq i) \quad (\text{B.2})$$

Die Beobachtung bei dieser als zufällig aufgefassten Prozedur wird durch die Zufallsgröße Z beschrieben. In diesem Fall wird die Zufallsgröße als die Anzahl der Erfolge definiert und kann somit folgende diskrete Werte annehmen:

Tab. B.1: Mögliche Werte der Zufallsgröße Z im Urnenversuch

Es wird keine schwarze Kugel gezogen:	$Z(\omega_0) = 0$
Es wird eine schwarze Kugel gezogen:	$Z(\omega_1) = 1$
Es werden zwei schwarze Kugeln gezogen:	$Z(\omega_2) = 2$
Es werden drei schwarze Kugeln gezogen:	$Z(\omega_3) = 3$
Es werden vier schwarze Kugeln gezogen:	$Z(\omega_4) = 4$
Es werden fünf schwarze Kugeln gezogen:	$Z(\omega_5) = 5$

Für die verschiedenen möglichen Mengen an Erfolgen $z \in \{0, 1, \dots, 5\}$ können nun die Wahrscheinlichkeitsverteilungen aufgestellt werden. Da die Zufallsgröße hier nur abzählbar viele mögliche Funktionswerte einnehmen kann, handelt es sich bei den daraus resultierenden Wahrscheinlichkeitsverteilungen um Wahrscheinlichkeitsfunktionen (WFen). Da die tatsächliche Anzahl θ der schwarzen Kugeln unbekannt ist, wird ein Wert für θ angenommen und dafür mit Hilfe von Formel B.2 jeweils die WF ermittelt.

Dieser Wert des Parameters wird so lange variiert, bis alle WFn für den Parameter $\theta \in \{1, \dots, 9\}$ ermittelt worden sind. Die beiden Fälle, dass keine bzw. 10 schwarze Kugeln auftreten, werden nicht betrachtet, da dies implizieren würde, dass nur eine Farbe in der Population enthalten wäre und es sich infolgedessen nicht um einen Bernoulli-Versuch handeln würde. Die Ergebnisse sind in Abbildung B.1 dargestellt.

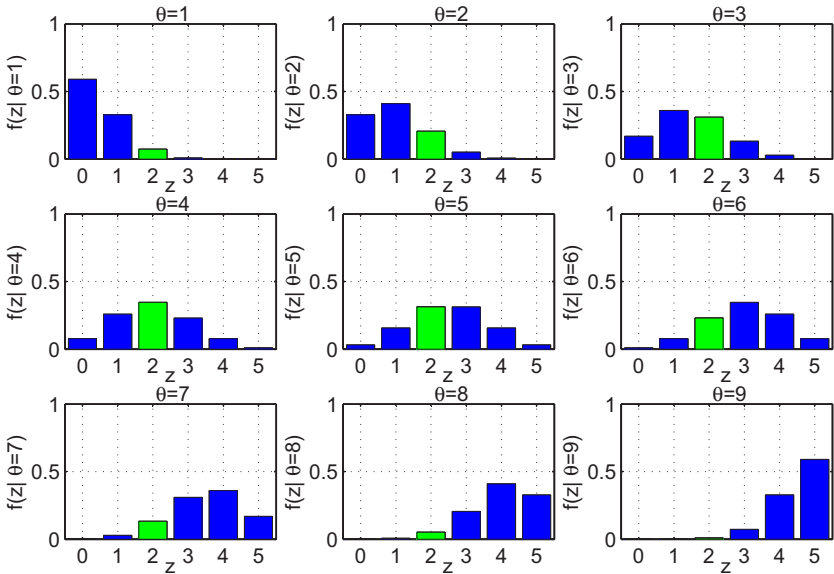


Abb. B.1: Wahrscheinlichkeitsfunktionen für $\theta \in \{1, 2, \dots, 9\}$ beim Urnenversuch

In diesem Plot ist die Wahrscheinlichkeit, keine schwarze Kugel zu ziehen, unter der Annahme $\theta = 1$, also dass sich in der Population nur eine schwarze Kugel befindet, verhältnismäßig am größten. Unter der Annahme, dass sich in der Urne eine oder mehrere schwarze Kugeln befinden, steigt sukzessive die Wahrscheinlichkeit, diese zu ziehen, von den jeweils linken zu den rechten Plots und von oben nach unten. Unter der Annahme, dass die Population fast nur aus schwarzen Kugeln besteht, ist in dem Plot unten rechts die Wahrscheinlichkeit, fünf schwarze Kugeln zu ziehen, am größten. Die Wahrscheinlichkeiten für das tatsächlich aufgetretene Ereignis der zwei gezogenen schwarzen Kugeln ist in Abbildung B.1 jeweils grün hervorgehoben. Diese erscheint bei einer Population von $\theta = 4$ am höchsten.

B.1.2 Erstellung einer Likelihood-Funktion

Da bei der Parameteridentifikation der umgekehrte Fall vorliegt, dass für ein vorhandenes Simulationsmodell bereits beobachtete Daten vorliegen, muss unter all den möglichen WFn diejenige WF gefunden werden, die die beobachteten Daten am wahrscheinlichsten erzeugt hat. Um auf Basis der in Kapitel B.1.1 ermittelten WFn die dafür benötigte Likelihood-Funktion zu erstellen, muss für das Beispiel die Rolle des Parametervektors θ mit dem des Datenvektors z getauscht werden:

$$L(\theta|n = 5, z) = f(z|n = 5, \theta) \tag{B.3}$$

Damit können aus Gleichung B.2 alle Likelihood-Funktionen für $z \in \{0, 1, \dots, 5\}$ erstellt werden. Die daraus resultierenden Verteilungen sind in Abbildung B.2 dargestellt.

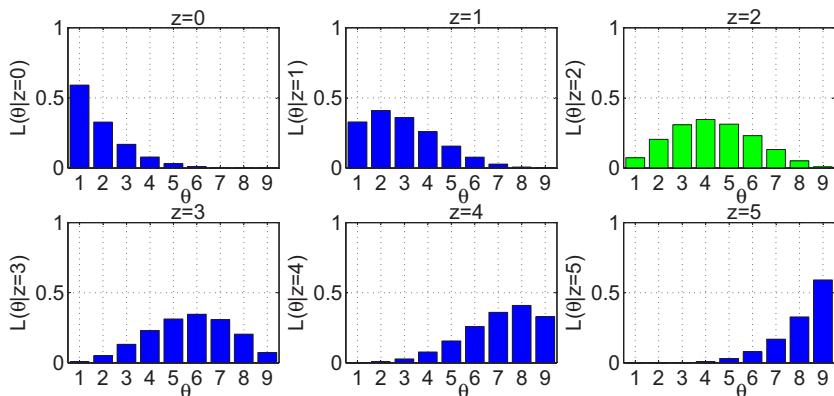


Abb. B.2: Likelihood-Funktionen für $z \in \{0, 1, 2, 3, 4, 5\}$ beim Urnenversuch

Im Gegensatz zu Abbildung B.1 sind darin, wie gewünscht, die Achsen des Parametervektors θ mit dem des Datenvektors z getauscht. Somit wird darin bei jeweils gleichbleibender Beobachtung die Anzahl der Parameter variiert.

In dem Plot oben links in Abbildung B.1 kann abgelesen werden, dass die Wahrscheinlichkeit, dass nur eine schwarze Kugel in der Urne enthalten ist, bei der Beobachtung keiner schwarzen Kugel $z = 0$ am größten ist. Für den Fall, dass eine oder mehrere schwarze Kugeln beobachtet werden,

steigt in Darstellung B.2 sukzessive die Wahrscheinlichkeit, dass sich eine oder mehrere schwarze Kugeln in der Population befinden, von den jeweils linken zu den rechten Plots und von oben nach unten. In dem Fall, dass nur schwarzen Kugeln gezogen werden, ist die Wahrscheinlichkeit, dass die Population fast nur aus schwarzen Kugeln besteht, am größten. Die tatsächlich aufgetretene Beobachtung der zwei gezogenen schwarzen Kugeln $z = 2$ ist in Darstellung B.2 grün hervorgehoben. Für diese können mit Hilfe von Gleichung B.4 die Einzelwahrscheinlichkeiten ermittelt werden:

$$L(\theta | n = 5, z = 2) = \frac{5!}{2!(5-z)!} \frac{\theta^z}{10^5} \left(1 - \frac{\theta}{10}\right)^{(5-z)}, \quad (0 \leq \theta \leq 10) \tag{B.4}$$

Die Ergebnisse dieser Berechnung sind in Tabelle B.2 dargestellt.

Tab. B.2: Likelihood-Funktionswerte für $z = 2$ im Urnenversuch

θ	1	2	3	4	5	6	7	8	9
$L(\theta n = 5, z = 2)$	0.073	0.205	0.309	0.346	0.313	0.230	0.132	0.051	0.008

Anhand der darin erfassten Daten kann die Vermutung aus Kapitel B.1.1, dass die Wahrscheinlichkeit für die Beobachtung der zwei gezogenen schwarzen Kugeln $z = 2$ bei einer Population von $\theta = 4$ am größten ist, bestätigt werden. Dementsprechend kann mit Hilfe der Likelihood-Funktion aus einer Familie an WFn diejenige WF identifiziert werden, die die Beobachtung tatsächlich am wahrscheinlichsten erzeugt hat.

Zusammenfassend ist die WF aus Abbildung B.1 eine Funktion der Daten z bei festgelegten Parameterwert, der im Datenwertebereich definiert ist. Im Gegensatz dazu ist die Likelihood-Funktion eine Funktion des Parameters θ bei einer vorhandenen Beobachtung, die im Parameterwertebereich definiert ist. Verallgemeinert repräsentiert die WF die Wahrscheinlichkeit eines speziellen Datenwertes bei festgelegten Parametern, wohingegen die Likelihood-Funktion die Wahrscheinlichkeit eines speziellen Datenwertes bei einer festgelegten Beobachtung beschreibt. Da in diesem Fall nur ein Parameter existiert, können die Wahrscheinlichkeiten als diskreter Verlauf dargestellt werden. Bei zwei Parametern wäre die Funktion eine Fläche im Parameterraum und bei m Parametern eine m -dimensionale Fläche in einem m -dimensionalen Hyperraum, aufgespannt vom Vektor $\theta = (\theta_1, \dots, \theta_k)$ [MYUNG 2003, S.92f].

B.2 Parameteridentifikation mit der Methode der kleinsten Quadrate

Für die Parameteridentifikation mit der Methode der kleinsten Quadrate soll im Folgenden ein Experiment von Daniel und Wood [DANIEL 1980] als Referenz dienen. Darin wurde von einem Chemiewerk über einen Zeitraum von 17 Tagen täglich ein Ausgangssignal $z_i(t)$ erfasst. Dieses basierte wiederum auf zwei unterschiedlichen Eingangssignalen $u_{1_i}(t)$ und $u_{2_i}(t)$. Die Ergebnisse der Studie sind für jeden einzelnen Tag in Tabelle B.3 dargestellt.

Tab. B.3: Übersicht über die erfassten Daten nach [RISSANEN 1987, S.96]

Sign.	Tag																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
u_1	80	62	62	62	62	58	58	58	58	58	58	50	50	50	50	50	56
u_2	27	22	23	24	24	23	18	18	17	18	19	18	18	19	19	20	20
z	37	18	18	19	20	15	14	14	13	11	12	8	7	8	8	9	15

Gemäß Gleichung 2.16 aus Kapitel 2.3.2 kann damit die Systemgleichung folgendermaßen formuliert werden:

$$\begin{pmatrix} 37 \\ 18 \\ \vdots \\ 15 \end{pmatrix} = \begin{pmatrix} 80 & 27 \\ 62 & 22 \\ \vdots & \vdots \\ 56 & 20 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_k \end{pmatrix} \quad (\text{B.5})$$

Analog dem Experiment von Rissanen wird auch hier von einer hypothetischen Systemgleichung der Form $z = \theta_1 + \theta_2 u_1 + \theta_3 u_2 + \theta_4 u_2^2$ ausgegangen. Zum Anlernen der freien Parameter wurde auf Basis von Gleichung 2.20 aus der Vorstellung der Methode der kleinsten Quadrate in Kapitel 2.3.2 mit Hilfe des Simulationswerkzeugs MATLAB ein Algorithmus implementiert, der die Werte aus Tabelle B.3 sukzessive auf Basis der Systemgleichung verarbeitet. Dieser Prozess des Anlernens der Parameter ist in Abbildung B.3 graphisch veranschaulicht. Darin wird die Schätzung der Parameter mit jedem zusätzlich verarbeiteten Zeitpunkt immer exakter. Generell steigt bei der Methode der kleinsten Quadrate die Genauigkeit einer Parameterschätzung mit jedem zusätzlich zur Verfügung stehenden Messpunkt.

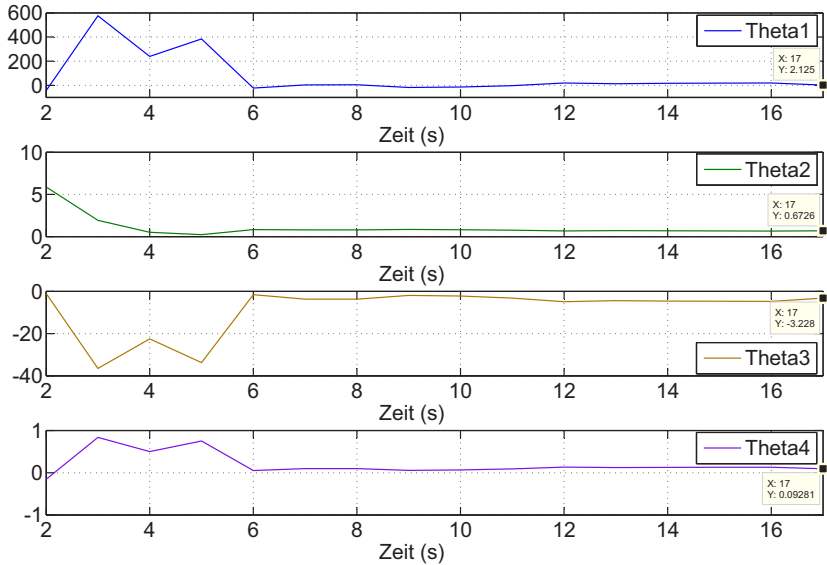


Abb. B.3: Approximation der LSE-Parameterschätzung über die einzelnen Zeitschritte

Am Ende des Anlernprozesses mit den $n = 17$ Datensätzen resultieren die am rechten Rand von Abbildung B.3 dargestellten Werte für die vier Parameter:

$$(\hat{\theta}_{1_{LSE}}, \hat{\theta}_{2_{LSE}}, \hat{\theta}_{3_{LSE}}, \hat{\theta}_{4_{LSE}}) = (2.125, 0.6726, -3.228, 0.09281) \quad (\text{B.6})$$

Diese entsprechen damit in sehr guter Näherung den Ergebnissen, die auch Rissanen [RISSANEN 1987, S.96] bei der Durchführung einer Parameterschätzung für dieses System ermittelt hat:

$$(\hat{\theta}_{1_{Ris.}}, \hat{\theta}_{2_{Ris.}}, \hat{\theta}_{3_{Ris.}}, \hat{\theta}_{4_{Ris.}}) = (2.13, 0.67, -3.23, 0.093) \quad (\text{B.7})$$

Um die Güte der Schätzung visuell bewerten zu können werden die aus der parametrisierten Systemgleichung resultierenden Werte mit den gemessenen Werten verglichen. Der aus der Schätzung mit der Methode der kleinsten Quadrate resultierende Verlauf des Systemausgangs ist in Abbildung B.4 in direktem Vergleich mit den tatsächlich gemessenen Werten dargestellt. Anhand dieses Vergleichs wird deutlich, dass die parametrisierte

Systemgleichung die Realität bereits sehr gut reproduziert. An dieser Stelle der Hinweis dass weitere Messpunkte bzw. eine Systemgleichung höherer Ordnung die Abweichung noch weiter minimieren könnten.

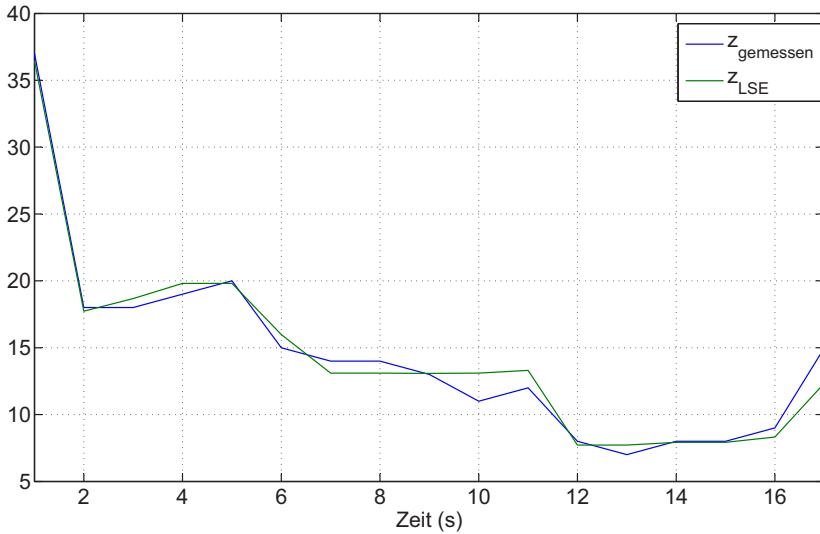


Abb. B.4: Vergleich zwischen gemessenem und über LSE-Schätzung approximiertem Systemausgang

C Implementierung der Gesamtfahrzeugmodellarchitektur

Gemäß Kapitel 3.3.2 ist der erste Schritt beim Aufbau einer Gesamtsystemarchitektur die Abbildung der Struktur im Co-Simulationsprozessschritt „*System Design*“. Als Ergebnis ist im Folgenden das Evaluationsbeispiel eines EREVs aus Kapitel 4.4.1 abgebildet. Dieses repräsentiert einen projektspezifischen Verbund, dessen prinzipieller Signalfluss anhand Abbildung 4.8 erläutert worden ist und welches so als Evaluationsbeispiel im Architekturwerkzeug SystemDesk implementiert worden ist. Nach dem Konsistenz-Check der Schnittstellen wurde dieses für die weitere Verwendung in eine .muxml-Datei exportiert, deren Struktur ebenfalls erläutert wird.

C.1 Aufbau der Gesamtfahrzeugmodellarchitektur

Die Abbildungen C.1 und C.2 zeigen das detaillierte Architekturlayout des Evaluationsbeispiels des EREVs mit virtuellem Umfeld und virtuellem Fahrer. Darin ist die Parameterdatei „Zyklus“ aus Abbildung 4.10 am linken Rand von Abbildung C.1 als separate Funktion dargestellt, da diese die Eingangsgrößen für sämtliche Nebenverbraucher liefert. Obwohl es sich bei einer Parameterdatei um keine Funktion handelt, kann dadurch über den in Kapitel 4.3.1 beschriebenen Konsistenz-Check überprüft werden, ob sämtliche Nebenverbraucher parametrisiert werden. Analog dazu sind die globalen Parameter als weitere Funktion am linken Rand als Parameterdatei „Umgebung“ abgebildet. Abbildung C.2 enthält in der rechten oberen Hälfte den virtuellen Fahrer und in der rechten unteren Hälfte das virtuelle Umfeld. Beide Darstellungen berücksichtigen sämtliche relevante Funktionen mit den jeweils zugeordneten Kategorien. Die ausgetauschten Signale sind gemäß der Signalnamenskonvention aus Kapitel 4.2.3 benannt.

C Implementierung der Gesamtfahrzeugmodellarchitektur

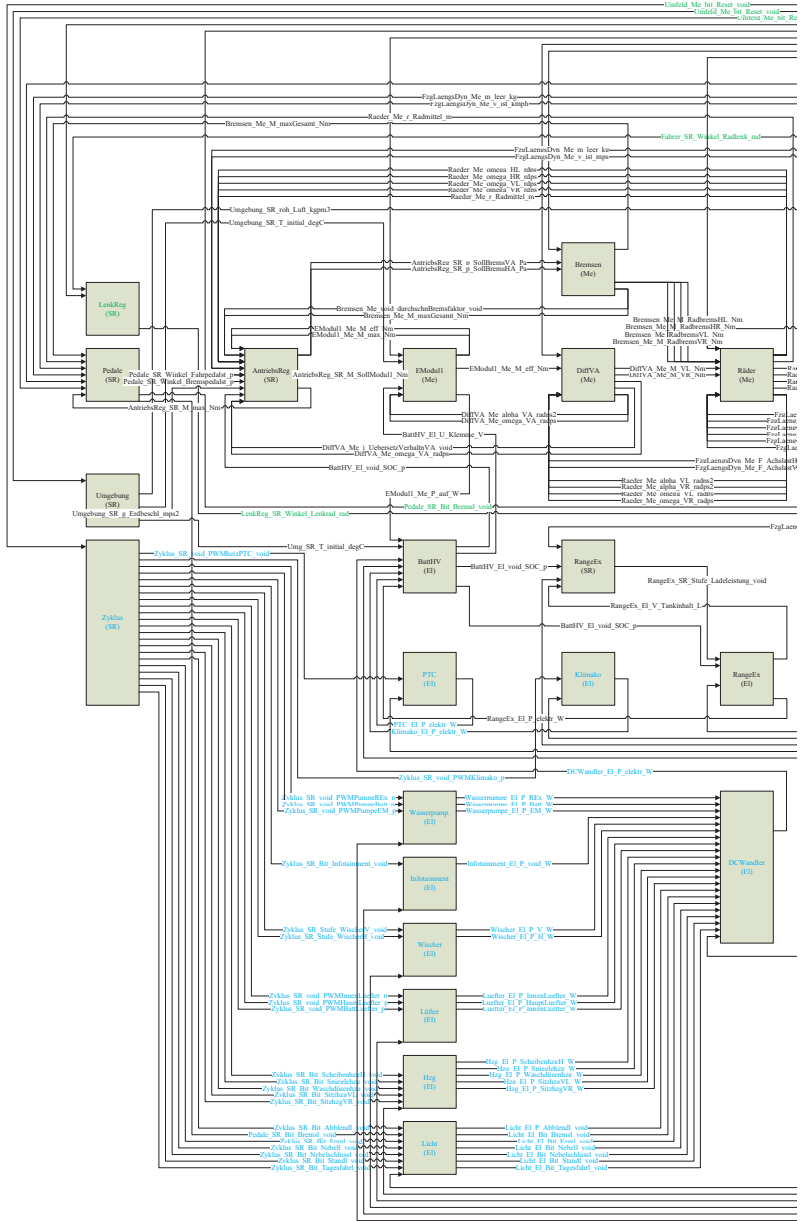


Abb. C.1: Linke Hälfte einer Closed-Loop-Gesamtfahrzeugmodellarchitektur

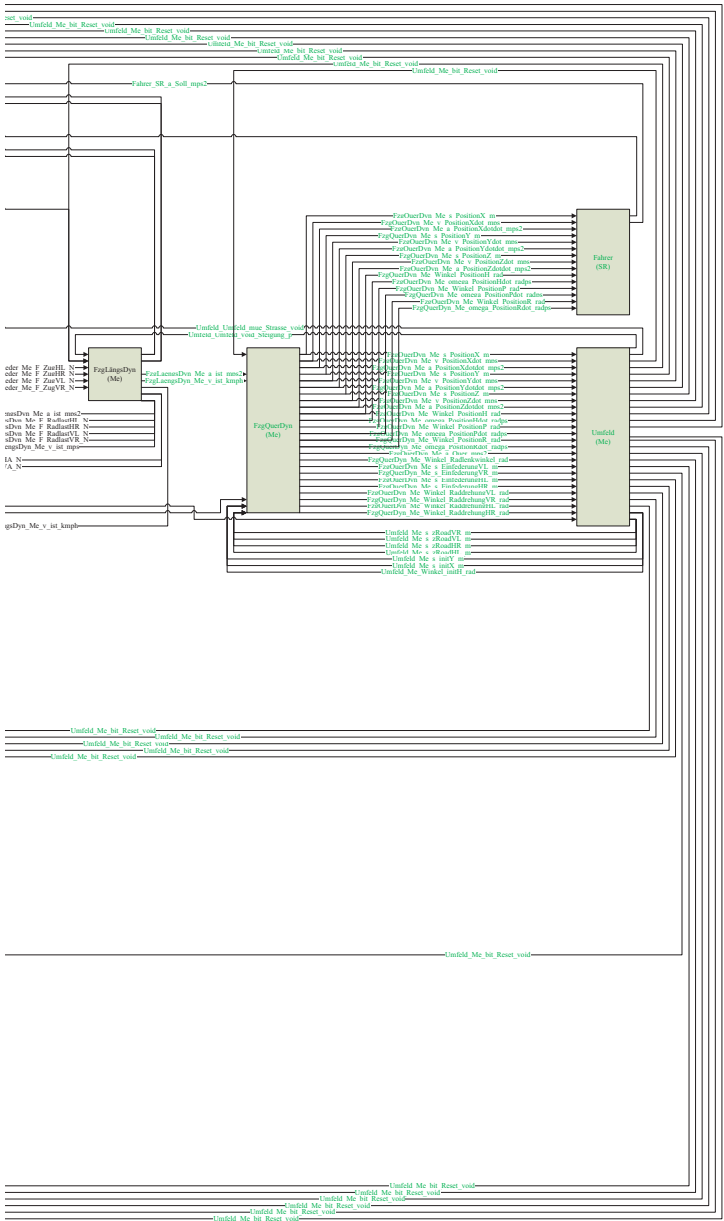


Abb. C.2: Rechte Hälfte einer Closed-Loop-Gesamtfahrzeugmodellarchitektur

C.2 Speicherung der Gesamtfahrzeugmodellarchitektur

In der *Mock-Up*-Beschreibungsdatei werden sämtliche systembeschreibende Informationen der Gesamtfahrzeugmodellarchitektur gespeichert. Dazu zählt, wie in folgendem Quellcode-Auszug aus dem Evaluationsbeispiel des EREVs exemplarisch dargestellt, zunächst die als „*Kind*“ bezeichnete Art des Dokuments (Zeile 3). In der als „*Header*“ bezeichneten Kopfdatei (Zeile 4-29) werden eine eindeutige Referenznummer (engl. *identifier*), die Versionsnummer, der Namen des projektspezifischen Verbunds und Informationen sowohl über den architekturverantwortlichen Entwickler als auch über den Verantwortlichen für deren Überprüfung aufgelistet. Innerhalb des als „*ModelUsages*“ bezeichneten Teils der Systembeschreibung (Zeile 30-221) sind sämtliche Domänen und Funktionen, die in der Gesamtfahrzeugmodellarchitektur verwendet werden, aufgelistet. Darin enthält jede Funktion der Architektur eine separate Referenznummer und eine entsprechende Versionsnummer. Innerhalb der Funktionen sind jeweils die in der Systembeschreibung spezifizierten Eingangs- und Ausgangssignale mit den jeweiligen Signalnamen, dem Typ und der Einheit gespeichert. Um die Darstellung nicht zu überladen, wird für jede Funktion jeweils nur das erste Eingangs- und Ausgangssignal dargestellt. Der Teil „*MockUpContext*“ (Zeile 221-227) gibt im weiteren Verlauf Auskunft über Simulationszweck und Projekt, für die das MU erstellt worden ist. Abschließend wird noch im Referenz-Teil (Zeile 228-232) mit einer Referenznummer vermerkt, aus welchem MMU und auf Basis welcher Version das hier betrachtete MU abgeleitet worden ist.

Gemäß dem Co-Simulationsprozess aus Kapitel 3.3.2 werden aus der *Mock-Up*-Beschreibungsdatei *Abstract-Model*-Beschreibungsdateien erzeugt. Dabei handelt es sich um Freischnitte der einzelnen Funktionen, die für die Erstellung der *Modeltemplate*-Beschreibungsdateien benötigt werden. Deren Struktur entspricht exakt der bereits erläuterten Struktur der *Mock-Up*-Beschreibungsdatei, wobei diese lediglich die Schnittstellen der jeweils betrachteten Funktionen enthält. In deren Referenz-Teil werden das MU und die MU-Version referenziert, aus der sie abgeleitet worden ist, wodurch die einzelnen Freischnitte wieder zu einer Gesamtfahrzeugmodellarchitektur zusammengefügt werden können.

Listing C.1: Gesamtfahrzeugmodellarchitektur als .muxml

```

1  i>¿<?xml version="1.0" encoding="utf-8"?>
2  <MockUpDefinition xmlns:xsi="http://www.w3.org/2001/
   XMLElementSchema-instance" xmlns:xsd="http://www.w3.org/2001/
   XMLElementSchema" xsi:type="AbstractMockUpDefinition"
   xsi:schemaLocation="uri:MSB MU.xsd" xmlns="uri:MSB">
3  <Kind>AbstractMockUp</Kind>
4  <Header>
5  <Id>b0043786-634e-4ebd-a2de-487dfc28ad4a</Id>
6  <VersionId>0</VersionId>
7  <Name>MU_A1_etron_VirtualTestDrive_mit_NV</Name>
8  <Editors>
9  <Editor>
10  <Email>florian1.netter@audi.de</Email>
11  <FirstName>Florian</FirstName>
12  <LastName>Netter</LastName>
13  <Department>I/AEV-22</Department>
14  <Company>AUDI AG</Company>
15  <DateTime>2013-03-08T00:00:00+01:00</DateTime>
16  <ChangeDescription>Erstellt</ChangeDescription>
17  </Editor>
18 </Editors>
19 <Approver>
20 <Email>florian1.netter@audi.de</Email>
21 <FirstName>Florian</FirstName>
22 <LastName>Netter</LastName>
23 <Department>I/AEV-22</Department>
24 <Company>AUDI AG</Company>
25 <DateTime>2014-01-29T00:00:00+01:00</DateTime>
26 <StatusName>approved</StatusName>
27 <StatusLevel>model creation</StatusLevel>
28 </Approver>
29 </Header>
30 <ModelUsages>
31 <ModelUsage>
32 <Id>3160a795-ae5f-4397-a348-463ef687599c</Id>
33 <VersionId>0</VersionId>
34 <Name>Umfeld</Name>
35 <InputSignals>
36 <InputSignal>
37 <Name>FzgQuerDyn_Fahrwerk_a_PositionXdotdot_mps2<
   /Name>
38 <Type>ValueFloat</Type>
39 <Width />
40 <Unit>m/s2</Unit>
41 </InputSignal>

```

```
42     ...
43     </InputSignal>
44 </InputSignals>
45 <OutputSignals>
46   <OutputSignal>
47     <Name>Umfeld_Umfeld_bit_Reset_void</Name>
48     <Type>ValueFloat</Type>
49     <Width />
50     <Unit>Bit</Unit>
51     <Quantity>0,1</Quantity>
52   </OutputSignal>
53   <OutputSignal>
54     ...
55   </OutputSignal>
56 </OutputSignals>
57 </ModelUsage>
58 <ModelUsage>
59   <Id>3f8f1225-c30d-4501-a9d5-5df456424d25</Id>
60   <VersionId>0</VersionId>
61   <Name>RB</Name>
62   <InputSignals>
63     <InputSignal>
64       <Name>Umfeld_Umfeld_bit_Reset_void</Name>
65       <Type>ValueFloat</Type>
66       <Width />
67       <Unit>Bit</Unit>
68       <Quantity>0,1</Quantity>
69     </InputSignal>
70   </InputSignals>
71   <OutputSignals>
72     <OutputSignal>
73       <Name>Umgebung_RB_g_Erdbeschl_mps2</Name>
74       <Type>ValueFloat</Type>
75       <Width />
76       <Unit>m/s2</Unit>
77     </OutputSignal>
78     <OutputSignal>
79       ...
80     </OutputSignal>
81   </OutputSignals>
82 </ModelUsage>
83 <ModelUsage>
84   <Id>aa3665ad-fe9e-44a8-b0a5-4898b5d46fce</Id>
85   <VersionId>0</VersionId>
86   <Name>Fahrer</Name>
87   <InputSignals>
```

```
88     <InputSignal>
89         <Name>AntriebsReg_Antrieb_M_max_Nm</Name>
90         <Type>ValueFloat</Type>
91         <Width />
92         <Unit>Newtonmeter</Unit>
93     </InputSignal>
94     <InputSignal>
95         ...
96     </InputSignal>
97 </InputSignals>
98 <OutputSignals>
99     <OutputSignal>
100         <Name>Pedale_Fahrer_Bit_Bremsl_void</Name>
101         <Type>ValueFloat</Type>
102         <Width />
103         <Unit>void</Unit>
104         <Quantity>0 oder 1</Quantity>
105     </OutputSignal>
106     ...
107 </OutputSignal>
108 </OutputSignals>
109 </ModelUsage>
110 <ModelUsage>
111     <Id>dbed06d1-e6ae-4077-92ac-5c8e09742ae3</Id>
112     <VersionId>0</VersionId>
113     <Name>Antrieb</Name>
114     <InputSignals>
115         <InputSignal>
116             <Name>Umfeld_Umfeld_mue_Strasse_void</Name>
117             <Type>ValueFloat</Type>
118             <Width />
119             <Unit>void</Unit>
120         </InputSignal>
121         <InputSignal>
122             ...
123         </InputSignal>
124     </InputSignals>
125     <OutputSignals>
126         <OutputSignal>
127             <Name>Raeder_Antrieb_r_Radmittel_m</Name>
128             <Type>ValueFloat</Type>
129             <Width />
130             <Unit>Meter</Unit>
131             <Quantity>0 - 0,5</Quantity>
132         </OutputSignal>
133     </OutputSignal>
```

```

134     ...
135     </OutputSignal>
136 </OutputSignals>
137 </ModelUsage>
138 <ModelUsage>
139     <Id>41d648bc-622d-46ec-888f-c2846e41fbb4</Id>
140     <VersionId>0</VersionId>
141     <Name>Fahrwerk</Name>
142     <InputSignals>
143         <InputSignal>
144             <Name>FzgLaengsDyn_Antrieb_a_ist_mps2</Name>
145             <Type>ValueFloat</Type>
146             <Width />
147             <Unit>m/s2</Unit>
148         </InputSignal>
149         <InputSignal>
150             ...
151         </InputSignal>
152     </InputSignals>
153     <OutputSignals>
154         <OutputSignal>
155             <Name>FzgQuerDyn_Fahrwerk_a_PositionXdotdot_mps2<
156 /Name>
157             <Type>ValueFloat</Type>
158             <Width />
159             <Unit>m/s2</Unit>
160         </OutputSignal>
161         <OutputSignal>
162             ...
163         </OutputSignal>
164     </OutputSignals>
165 </ModelUsage>
166 <ModelUsage>
167     <Id>5869dcac-cde6-45ae-ae50-634de575e984</Id>
168     <VersionId>0</VersionId>
169     <Name>Aufbau</Name>
170     <InputSignals>
171         <InputSignal>
172             <Name>Umfeld_Umfeld_bit_Reset_void</Name>
173             <Type>ValueFloat</Type>
174             <Width />
175             <Unit>Bit</Unit>
176             <Quantity>0,1</Quantity>
177         </InputSignal>
178         <InputSignal>
179             ...

```

```
179     </InputSignal>
180 </InputSignals>
181 <OutputSignals>
182   <OutputSignal>
183     <Name>Infotainment_Aufbau_P_void_W</Name>
184     <Type>ValueFloat</Type>
185     <Width />
186     <Unit>Watt</Unit>
187     <Quantity>0 - inf</Quantity>
188   </OutputSignal>
189   <OutputSignal>
190     ...
191   </OutputSignal>
192 </OutputSignals>
193 </ModelUsage>
194 <ModelUsage>
195   <Id>6f5d22fa-3bce-46d5-a60c-ec564f543207</Id>
196   <VersionId>0</VersionId>
197   <Name>Elektrik</Name>
198   <InputSignals>
199     <InputSignal>
200       <Name>EModul1_Antrieb_P_auf_W</Name>
201       <Type>ValueFloat</Type>
202       <Width />
203       <Unit>Watt</Unit>
204     </InputSignal>
205     <InputSignal>
206       ...
207     </InputSignal>
208   </InputSignals>
209   <OutputSignals>
210     <OutputSignal>
211       <Name>BattHV_Elektrik_U_Klemme_V</Name>
212       <Type>ValueFloat</Type>
213       <Width />
214       <Unit>Volt</Unit>
215     </OutputSignal>
216     <OutputSignal>
217       ...
218     </OutputSignal>
219   </OutputSignals>
220 </ModelUsage>
221 </ModelUsages>
222 <MockUpContext>
223   <Disciplines>
224     <Discipline>FLV</Discipline>
```

```
225     </Disciplines>
226     <Project />
227 </MockUpContext>
228 <References>
229     <MaxMockUpIdRef refId="75025377-d8cd-4385-82bf-2
230     ad43da1315c" refVersionId="0" />
230     <EntireModelSepcificationPath />
231     <NativePath>file:///D:/Promotion/06_Modelle/
232     Architektur_Netter/Architekturlayout_A1etron/FLV.sdp</
233     NativePath>
232 </References>
233 </MockUpDefinition>
```


D Berechnung der Simulationsmodellkomplexität

Listing D.1: MATLAB-Skript zur Bestimmung der Simulationsmodellkomplexität beim Linearmodell

```
1  x=[4.8:0.001:5.2]; %Festlegung Hilfsvektor zur Darstellung
    der Wahrscheinlichkeitsdichtefunktionen
2
3  z=[5, 5.02, 4.99, 5.01, 5, 5.03, 4.98, 5.02, 5.01, 5.03, 5,
    4.99, 5.01, 5.02, 4.96]; %Festlegung der Messdaten
4
5  Schrittweite=0.0005; %Festlegung Schrittweite des
    Parameterbereichs
6
7  theta=[4.96:Schrittweite:5.03]; %Festlegung des
    Parameterbereichs
8
9  k=1; %Festlegung Anzahl der Parameter
10
11 MLE=0; %Initialisierung der Maximum Likelihood Estimation
12
13 Likelihood_Function=theta; %Initialisierung der Likelihood-
    Funktion
14
15 Optimaler_Modellparameter=0; %Initialisierung des optimalen
    Modellparameters
16
17
18 n=length(z); %Ermittlung Anzahl der Messwerte
19
20 c4=(gamma(n/2)/gamma((n-1)/2))*((2/(n-1))^(0.5)); %
    Berechnung Korrekturfaktor der Standartabweichung
21
22
23
```

```
24 %Berechnung der Wahrscheinlichkeitsdichten und der
    Likelihood-Funktion
25
26 for i=1:length(theta)
27
28     %Abschätzung der Standardabweichung Sigma
29
30     y=1*theta(i); %Simulierte Ergebnisse
31     B=(z-y).^2;
32     Sx=(sum(B)*1/(n-1))^(0.5); %Asymptotisch
    Erwartungstreue Schätzung der Standardabweichung
33     sigma=Sx/c4; %Erwartungstreue Schätzung der
    Standardabweichung
34
35
36
37 %Berechnung der Wahrscheinlichkeitsdichtefunktion
38
39 fa=(1/((2*pi*sigma^2))^(0.5))*exp(B/(-2*(sigma^2)));
40 ga=(1/((2*pi*sigma^2))^(0.5))*exp(((x-1*theta(i)).^2)
    /(-2*(sigma^2)));
41
42
43 %Berechnung der Likelihood-Funktion
44
45 La=prod(fa);
46
47
48 %Darstellung der Wahrscheinlichkeitsdichtefunktion
49
50 figure(1)
51 xlabel('z')
52 ylabel('f(z|theta)')
53 hold all
54 plot(x,ga)
55 grid on
56
57
58 %Darstellung der Likelihood-Funktion
59
60 figure(2)
61 xlabel('theta')
62 ylabel('L(theta|z)')
63 hold all
64 scatter(theta(i),La)
65 grid on
```

```

66
67
68     %Ermittlung der Maximum Likelihood Estimation und des
        dazu gehörigen Parameters
69
70     if La>MLE
71         MLE=La;
72         Optimaler_Modellparameter=theta(i);
73         Maxstelle=i;
74         yy=y;
75     end
76     Likelihood_Function(i)=La;
77
78 end
79 MSE=mean((z-yy).^2);
80
81
82 %Berechnung der Parametrischen Komplexität
83
84 PC=k
85
86
87 %Berechnung der Stochastischen Komplexität
88
89 Log_Likelihood_Function=log2(Likelihood_Function);
90 Minus_Log_Likelihood_Function=-1*Log_Likelihood_Function;
91
92 Ableitung_1=gradient(Minus_Log_Likelihood_Function,
        Schrittweite);
93 Ableitung_2=gradient(Ableitung_1,Schrittweite);
94
95 SC_Stern=0.5*log(abs(Ableitung_2(Maxstelle)))
96 SC=SC_Stern-log(MLE);
97
98
99 %Berechnung der Geometrischen Komplexität
100
101 Ableitung=gradient(Log_Likelihood_Function,Schrittweite);
102
103 Funktion=Ableitung.^2; %Hilfsfunktion zur Berechnung der
        Fisher-Informationsmatrix
104 Fisher=mean(Funktion); %Fisher-Informationsmatrix
105 I1=Fisher/(n^k);
106
107 GC=(k/2)*log2(n/(2*pi))+log2(((I1)^(0.5))*(theta(length(
        theta))-theta(1)))

```

```
108
109
110 %Berechnung der Minimum Description Length
111
112 MDL= -1*log2(MLE)+GC;
113
114
115 %Berechnung des Akaike Information Criterion
116
117 AIC=-2*log2(MLE)+2*k;
118
119
120 %Berechnung des Bayesian Information Criterion
121
122 BIC=-2*log2(MLE)+k*log2(n);
```

Glossar

Attribut

Ein Attribut bezeichnet eine charakteristische Eigenschaft bzw. ein Wesensmerkmal. Im Rahmen dieser Arbeit wird dieser Begriff synonym für ein Quantifizierungsmerkmal verwendet.

Ausgangsgröße

Eine Ausgangsgröße ist eine erfassbare, von einem System erzeugte variable Größe, die nur von diesem und über das System von dessen Eingangsgrößen beeinflusst wird [DKE 2009, S.7].

Domäne

Eine Domäne repräsentiert im Sinne einer Aufgabe einen Funktionsbereich, welcher je nach betrachtetem technischem System unterschiedlich sein kann. So wird z. B. in der Kraftfahrzeugindustrie zwischen den Domänen Antrieb, Fahrwerk, Aufbau und Elektrik/Elektronik unterschieden [BRAESS 2003, S.XIIIff].

Eingangsgröße

Eine Eingangsgröße ist eine variable Größe, die von außen auf ein System wirkt und die von den anderen variablen Größen des Systems unabhängig ist [DKE 2009, S.7].

Funktion

Gegenüber anderslautenden Definitionen repräsentiert im Rahmen dieser Arbeit eine Funktion ein Element innerhalb eines Systems. Dieses wird im Verlauf einer virtuellen Entwicklung mit einem ausführbaren Modell befüllt, welches mit seiner Funktionalität zur Aufgabe des Systems beiträgt. Dem Begriff der Systemhierarchie folgend stellt eine Funktion die niedrigste, sinnvoll separierbare Ebene innerhalb der

Systemhierarchie dar und verfügt über fest definierte Schnittstellen in Form von Eingangs- und Ausgangsgrößen [STÜBER 2011, S.80].

Funktionalität

Funktionalität bezeichnet das Vermögen eines Softwareprodukts, beim Einsatz unter spezifizierten Bedingungen Ergebnisse zu liefern, die festgelegte und vorausgesetzte Erfordernisse erfüllen [DEUTSCHES INSTITUT FÜR NORMUNG 2010].

Komposition

Eine Komposition ist eine Kombination bestimmter Teile oder Elemente, um daraus ein Ganzes zu formen [PARTON 2011].

Modell

Ein Modell ist eine mathematische oder physikalische Darstellung eines Systems oder Prozesses, die das System oder den Prozess aufgrund bekannter Gesetzmäßigkeiten, einer Identifikation oder getroffener Annahmen genügend genau abbildet [DKE 2009, S.17].

Modellbildung

Unter Modellbildung (Modellierung) werden diejenigen Arbeiten verstanden, die notwendig sind, um ausgehend von einem realen technischen System eine computergerechte Darstellung des davon abgeleiteten Modells zu erstellen [DRONKA 2004, S.14].

Modellkomplexität

Modellkomplexität ist die Flexibilität eines Modells, die es ihm gestattet, sich einem breiten Bereich von Datensätzen (z. B. Wahrscheinlichkeitsverteilungen von Beobachtungen, die von einer Modellgleichung erzeugt werden) anzupassen [MYUNG 1999, S.2].

Parametrieren

Beim Parametrieren werden für die bei der Parametrisierung definierten Parameter eines Modells Werte festgelegt.

Parametrisieren

Beim Parametrisieren werden die Parameter eines Modells definiert.

Schnittstelle

Eine Schnittstelle ist eine Grenze zwischen zwei Funktionseinheiten, die durch funktionelle Merkmale, Signalkenngrößen oder andere Merkmale als geeignet festgelegt worden ist [DKE 2009, S.17].

Signal

Ein Signal ist eine physikalische Größe, bei der ein oder mehrere Parameter Informationen über eine oder mehrere Größen tragen [DKE 2009, S.23].

Simulation

Simulation ist das Nachbilden eines Systems mit seinen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind [VDI 2014, S.3]. Bei einer integrierten Simulation werden dabei zwei oder mehrere modellierte Systeme, die untereinander in Beziehung stehen, zu einer gemeinsamen Simulation zusammengefasst. Die dynamisch entstehenden Wechselwirkungen können dabei in einer Simulation ausgewertet werden.

Simulator

Als Simulator wird ein Werkzeug bezeichnet, mit dem die Umgebung für die Untersuchung des Verhaltens eines Objekts oder Subjekts dargestellt wird. Diese Definition umfasst sowohl Simulationswerkzeuge zum Ausführen virtueller Modelle als auch physische Simulatoren wie etwa Versuchseinrichtungen, die reale oder künstliche Lastfälle für ein betrachtetes Subjekt oder Objekt erzeugen. Im Rahmen dieser Arbeit wird der Begriff Simulator synonym für Simulationswerkzeug verwendet.

System

Ein System ist eine Menge miteinander in Beziehung stehender Elemente, die in einem bestimmten Zusammenhang als Ganzes angesehen und als von ihrer Umgebung durch eine gedachte Hüllfläche abgegrenzt betrachtet werden können. Ein System wird dabei im Allgemeinen hinsichtlich seiner Zielsetzung, zum Beispiel der Ausführung einer bestimmten Aufgabe, definiert. Elemente eines Systems

können natürliche oder künstliche Gegenstände ebenso wie Arten von Denkvorgängen und deren Ergebnis sein [DKE 2009, S.11].

Systemparameter

Ein Systemparameter ist eine charakteristische variable Größe, die die Beziehung zwischen variablen Größen innerhalb eines gegebenen Systems kennzeichnet [DKE 2009, S.12].

Systemstruktur

Unter einer Systemstruktur versteht man die Beziehungen zwischen den Elementen eines Systems [DKE 2009, S.12].

Übertragungsglied

Ein Übertragungsglied ist ein Teil eines Systems, für das funktionelle Beziehungen die Abhängigkeiten der Ausgangsgrößen von den Eingangsgrößen festlegen [DKE 2009, S.34].

Validierung

Die VDI-Richtlinie 3633 beschreibt als Ziel der Validierung, das Verhalten des realen Systems mit Hilfe eines Modells genau genug und fehlerfrei widerzuspiegeln. Validierung versucht zu klären, ob die Übereinstimmung von Modell und realem System hinreichend genau ist. Die geforderte Genauigkeit der Übereinstimmung hängt dabei vom jeweiligen Anwendungsfall des Modells ab. Validierung setzt also voraus, dass Simulationsergebnisse mit validen Daten des realen Systems verglichen werden [RABE 2008, S.15].

Verifikation

Verifikation bedeutet gemäß VDI-Richtlinie 3633 den formalen Nachweis der Korrektheit des Simulationsmodells, also die Übereinstimmung zwischen Umsetzung und Spezifikation. Im Bereich der Simulation bedeutet Verifikation beispielsweise, dass das ausführbare Modell das konzeptionelle Modell korrekt umsetzt. Die korrekte Umsetzung zwischen zwei Repräsentationsformen von Modellen wird durch die Korrektheit der Transformationen zwischen diesen überprüft [RABE 2008, S.14].

Wahrscheinlichkeitsdichtefunktion

Ist Z eine kontinuierliche Zufallsgröße, dann heißt die integrierbare, nichtnegative Funktion, die die Wahrscheinlichkeitsverteilung beschreibt, Wahrscheinlichkeitsdichtefunktion oder Dichte von Z [STELAND 2010, S.92].

Wahrscheinlichkeitsfunktion

Ist Z eine diskrete Zufallsgröße, dann heißt die Funktion, die die Wahrscheinlichkeitsverteilung beschreibt, Wahrscheinlichkeitsfunktion oder Zähldichte von Z [STELAND 2010, S.91].

Wahrscheinlichkeitsverteilung

Ordnet man in der Statistik jedem Ereignis eine Wahrscheinlichkeit zu, wird hierdurch die Wahrscheinlichkeitsverteilung festgelegt [STELAND 2010, S.89].

Zufallsgröße

Eine Zufallsgröße (Zufallsvariable, zufällige Größe) ist eine Funktion $Z : \Omega \rightarrow E$, die jedem Ergebnis $\omega \in \Omega$ eines Zufallsexperiments eine reelle Zahl zuordnet [BEWERSDORFF 2011, S.49].

Zustandsgröße

Eine Zustandsgröße ist ein Element eines Größenvektors $x(t)$ in einem System mit den Differentialgleichungen

$$\dot{x}(t) = f(x(t), u(t)) \quad \text{Zustandsgleichung}$$

$$y(t) = g(x(t), u(t)) \quad \text{Ausgangsgleichung}$$

$$\dot{x}(t) = Ax(t) + Bu(t) \quad \text{Zustandsgleichung}$$

$$y(t) = Cx(t) + Du(t) \quad \text{Ausgangsgleichung}$$

Aus diesen Differentialgleichungen kann bei bekannten Anfangsbedingungen zu einem beliebigen Zeitpunkt t_0 (häufig mit $t_0 = 0$) und bekanntem Zeitverlauf der Elemente des Vektors der Eingangsgrößen $u(t)$ ab dem Zeitpunkt t_0 der Zeitverlauf der Elemente des Vektors der Ausgangsgrößen $y(t)$ berechnet werden kann [DKE 2009, S.7].

Abkürzungsverzeichnis

ACSL	<i>Advanced Continuous Simulation Language</i>
ADTF	<i>Automotive Data and Time-Triggered Framework</i>
AEV	<i>Audi Electronics Venture GmbH</i>
AIC	<i>Akaike Information Criterion</i>
AUTOSAR	<i>AUTomotive Open System ARchitecture</i>
BEV	<i>Battery Electric Vehicle</i>
BIC	<i>Bayesian Information Criterion</i>
CAD	<i>Computer-Aided Design</i>
CAN	<i>Controller Area Network</i>
CFD	<i>Computational Fluid Dynamics</i>
CSV	<i>Comma-Separated-Value</i>
DAE	<i>Differential Algebraic Equations</i>
DC/DC	<i>Direct Current/Direct Current</i>
DMU	<i>Digital Mock-Up</i>
EREV	<i>Extended Range Electric Vehicle</i>
FEM	<i>Finite Element Methode</i>
FG	<i>Funktionsgruppe</i>
FMI	<i>Functional Mock-Up Interface</i>
FMU	<i>Functional Mock-Up</i>
GC	<i>Geometric Complexity</i>
GoF	<i>Goodness of Fit</i>
GUI	<i>Graphical User Interface</i>

HEV	<i>Hybrid Electric Vehicle</i>
HiL	<i>Hardware in the Loop</i>
HV	Hochvolt
IPS	<i>Imperial System of Units</i>
ITEA	<i>Information Technology for European Advancement</i>
KLI	Kullback-Leibler-Information
LSE	<i>Least Squares Estimation</i>
LTI	<i>Linear Time-Invariant</i>
MaP	<i>Maximum-a-Posteriori</i>
MDL	<i>Minimum Description Length</i>
MKS	Mehrkörpersimulation
MLE	<i>Maximum Likelihood Estimation</i>
MMU	<i>Maximal Mock-Up</i>
MpCCI	<i>Mesh-based parallel Code Coupling Interface</i>
MSE	<i>Mean Squared Error</i>
MU	<i>Mock-Up</i>
NV	Niedervolt
OCV	<i>Open Circuit Voltage</i>
ODE	<i>Ordinary Differential Equations</i>
OWL	<i>Web Ontology Language</i>
PC	<i>Parametric Complexity</i>
PDE	<i>Partial Differential Equations</i>
PDF	<i>Probability Density Function</i>
PHEV	<i>Plugin Hybrid Electric Vehicle</i>
PMF	<i>Probability Mass Function</i>
PTC	<i>Positive Temperature Coefficient</i>
RDF	<i>Ressource Description Framework</i>
RMSE	<i>Root Mean Squared Error</i>

SC	<i>Stochastic Complexity</i>
SI	<i>Système International d'Unités</i>
SiL	<i>Software in the Loop</i>
SoC	<i>State of Charge</i>
SoP	<i>Start of Production</i>
SSE	<i>Sum of Squared Errors</i>
SWRL	<i>Semantic Web Rule Language</i>
TIC	<i>Theil's Inequality Coefficient</i>
VE	Virtuelle Entwicklung
VTD	<i>Virtual Test Drive</i>
WDF	Wahrscheinlichkeitsdichtefunktion
WF	Wahrscheinlichkeitsfunktion
XML	<i>Extensible-Markup-Language</i>

Symbolverzeichnis

Γ	Eulersche Gammafunktion
Δt_s	Abtastperiodendauer
Δu_q	Quantisierungsintervall
θ	Parametervektor
$\hat{\theta}$	optimaler Parametervektor
θ_i	Parametervariable
$\hat{\theta}_i$	optimale Parametervariable
ρ	Luftdichte
σ^*	erwartungstreue Standardabweichung
Ω	Ergebnismenge
ω	Ergebnis
A_{Fzg}	Fahrzeugstirnfläche
a_{Fzg}	Fahrzeugbeschleunigung
a_i	reellwertiger Koeffizient
b_i	reellwertiger Koeffizient
c	Federsteifigkeit
c_4	Korrekturfaktor
c_W	Luftwiderstandsbeiwert
d	Dämpfungskonstante
E	Eigenwert
F_A	Antriebskraft
F_B	Beschleunigungswiderstandskraft
F_L	Luftwiderstandskraft

F_R	Rollwiderstandskraft
F_S	Steigungwiderstandskraft
F_W	Fahrwiderstandskraft
F_c	Federkraft
F_d	Dämpferkraft
F_m	Kraft an Masse
f_R	Rollwiderstandsbeiwert
f_i	Funktion
g	Erdbeschleunigung
g_i	Funktion
$I(f,g)$	Kullback-Leibler Information
$I_{ij}(\theta)$	Fisher-Informationsmatrix
i	Zählindex
j	Zählindex
k	Anzahl der Parameter
k_{Mod}	Anzahl der Modelle
$L(\theta)$	Likelihood-Funktion
$L(\hat{\theta})$	Maximum-Likelihood-Funktion
L_{ij}	reelle Matrix
$\ln L(\theta)$	Log-Likelihood-Funktion
l	Anzahl der Kombinationen
m_{Fzg}	Fahrzeugmasse
n	Anzahl Messpunkte
o	Schrittweite
p	beliebige reelle oder komplexe Zahl
q	Grad der höchsten Ableitung der Eingangsgröße

r	Grad der höchsten Ableitung der Ausgangsgröße
S	Abbildung eines linearen Systems
s	Position
t	Zeit
t_d	ganzzahlige, diskrete Zeitvariable
t_k	reelle, kontinuierliche Zeitvariable
U	Eingangsmatrix
u	Eingangsvektor
u_i	Eingangsvariable
v_{Fzg}	Fahrzeuggeschwindigkeit
w	Anzahl Stützstellen
x	Zustandsvektor
x_i	Zustandsvariable
y	Ausgangsvektor
y_i	Ausgangsvariable
Z	Zufallsgröße, Zufallsvariable
z	Datenvektor
z_i	Datenvariable

Abbildungsverzeichnis

1.1	Erhöhung Reifegrad durch virtuelle Entwicklung nach [SEIFFERT 2008, S.8]	2
1.2	Komponenten eines virtuellen Prototyps nach [SCHÄPPI 2005]	3
2.1	System mit Eingangs- und Ausgangsgrößen nach [DÖRING 2011, S.1]	10
2.2	Ausgewählte Signaleigenschaften nach [DÖRING 2011, S.2] . .	11
2.3	Übersicht kontinuierliche und diskrete Signale nach [FREY 2008, S.2]	12
2.4	Systembegriff [KÜBLER 2000, S.14]	13
2.5	Ausgewählte Systemeigenschaften	14
2.6	Superpositionsprinzip bei linearen Systemen S nach [FREY 2008, S.5].	15
2.7	Zusammenhang zwischen Modellbildung und Simulation [ZEITZ 1987]	17
2.8	Modellqualifizierung/-validierung und -verifikation nach [PELZ 2001, S.7]	19
2.9	Einteilung von Integrationsverfahren [DRONKA 2004, S.19] . .	23
3.1	Kopplungsvarianten nach [GEIMER 2006, S.572]	37
3.2	Exemplarische Definition eines Kondensators in Modelica [OTTER 2009]	38
3.3	Prinzipskizze Co-Simulation [NETTER 2013b, S.2]	39
3.4	Kopplungsschleife nach [BUSCH 2012, S.5]	42
3.5	Schnittstellenansätze für verschiedene Kopplungen nach [BUSCH 2012, S.6]	43
3.6	Prozess zur Erstellung einer Co-Simulation nach [STÜBER 2011, S.79]	46
3.7	Prozessschritt „ <i>System Design</i> “ nach [STÜBER 2011, S.80] . .	48

4.1	Hierarchie zur Strukturierung von Funktionen nach [NETTER 2013c, S.3]	56
4.2	Anpassungsfähigkeit und Wiederverwendbarkeit von Funktionen nach [NETTER 2013c, S.4].	58
4.3	Optionen zur Berechnung der Übergabevariablen nach [LUND 2008, S.176]	59
4.4	Signalnamenskonvention für die Schnittstellen [NETTER 2013c, S.4]	61
4.5	Signalabhängigkeiten im <i>Signal Manager</i> (SystemDesk-Screenshot)	63
4.6	Schema einer Funktionsgruppe [NETTER 2013a, S.44]	64
4.7	Schematischer Ablauf des Suchalgorithmus [NETTER 2013a, S.45]	66
4.8	Schema einer Fahrzeugmodellarchitektur nach [NETTER 2013c, S.6]	68
4.9	Architekturlayout der Hochvoltbatterie (SystemDesk-Screenshot)	70
4.10	Relevante Komponenten in einer <i>Closed-Loop Simulation</i>	72
4.11	Architekturlayout des virtuellen Umfelds (SystemDesk-Screenshot)	73
4.12	Architekturlayout des virtuellen Fahrers (SystemDesk-Screenshot)	74
4.13	Unterteilung in Funktionsgruppen nach [NETTER 2013a, S.46]	76
5.1	Einfaches und eines komplexes Simulationsmodell im Datenraum nach [MYUNG 2000, S.192]	78
5.2	Riemannsche Mannigfaltigkeit [PITT 2002, S.478]	83
5.3	Linearmodell für eine Quantifizierung bei unabhängiger Messung	86
5.4	WDF $f(z \theta = 5)$ für das Linearmodell bei unabhängiger Messung	91
5.5	WDF $f(z \theta)$ für das Linearmodell bei unabhängiger Messung	92
5.6	Likelihood-Funktion $L(\theta z)$ für das Linearmodell bei unabhängiger Messung	93
5.7	Simulationsmodell für eine Quantifizierung bei voneinander abhängiger Messung	95
5.8	Ersatzschaltbilder der für die Evaluation verwendeten Batteriemodelle	101

5.9	Spannungsverlauf einer Batterie bei einem Belastungssprung nach [JOSSEN 2006, S.19]	102
5.10	Datensatz zur Batteriemodellquantifizierung [GHORBEL 2014, S.69]	104
5.11	Datensatz zur Längsdynamikmodellquantifizierung	107
6.1	Anpassungsgüte und Extrapolationsvermögen als Funktion der Modellkomplexität nach [MYUNG 2004, S.355]	110
6.2	Vorhersagefehler und Kalibrierungsfehler als Funktion der Modellkomplexität [SCHOUPS 2012]	112
6.3	Verläufe der Simulationsergebnisgüte der Batteriemodelle	125
6.4	Verläufe der Simulationsergebnisgüte der Längsdynamikmodelle	127
6.5	Datensatz zum Training der freien Parameter nach [CHU 2012, S.54]	129
6.6	Datensatz zur Plausibilisierung der Parameteridentifikation im Extrapolationsbereich	130
6.7	Verläufe der Simulationsergebnisgüte für Gruppe 1 bei 1 Hz nach [NETTER 2013b, S.5]	131
6.8	Verläufe der Simulationsergebnisgüte für Gruppe 3 bei 1 Hz nach [NETTER 2013b, S.6]	133
7.1	Simulationszwecke in der Automobilindustrie nach [SEIFFERT 2008, S.359]	136
7.2	Unterscheidung in funktionelle, transitive und symmetrische Merkmale nach [HORRIDGE 2009, S.29f]	139
7.3	Struktur der Ontologie der Gesamtsystemarchitektur	140
7.4	Beschreibung des R_i -Modells (Protégé-Screenshot)	142
7.5	OntoGraf-Darstellung des Simulationszwecks „Energieverbrauch“ (Protégé-Screenshot)	143
7.6	Aktive Co-Simulation bei der Gesamtfahrzeugsimulation (TISC-Screenshot)	146
7.7	Laufende Umfeldsimulation bei der Gesamtfahrzeugsimulation (VTD-Screenshot)	147
A.1	Feder-Masse-Schwinger nach [LUNZE 2013, S.63]	155
A.2	Zustandsraummodell eines Eingrößensystems nach [LUNZE 2013, S.76]	160

B.1	Wahrscheinlichkeitsfunktionen für $\theta \in \{1, 2, \dots, 9\}$ beim Urnenversuch	165
B.2	Likelihood-Funktionen für $z \in \{0, 1, 2, 3, 4, 5\}$ beim Urnenversuch	166
B.3	Approximation der LSE-Parameterschätzung über die einzelnen Zeitschritte	169
B.4	Vergleich zwischen gemessenem und über LSE-Schätzung approximiertem Systemausgang	170
C.1	Linke Hälfte einer <i>Closed-Loop</i> -Gesamtfahrzeugmodellarchitektur	172
C.2	Rechte Hälfte einer <i>Closed-Loop</i> -Gesamtfahrzeugmodellarchitektur	173

Tabellenverzeichnis

3.1	Übersicht der im Prozess zur gekoppelten Simulation erzeugten Dateitypen nach [STÜBER 2011, S.79]	47
5.1	Voneinander unabhängige Messdaten zur Parametrierung des Linearmodells	86
5.2	Ergebnisse der nach ihrer Komplexität quantifizierten Batteriemodelle nach [GHORBEL 2014, S.68ff]	103
5.3	Luftwiderstandsbeiwert in Abhängigkeit von der Geschwindigkeit	105
5.4	Ergebnisse der nach ihrer Komplexität quantifizierten Längsdynamikmodelle nach [GHORBEL 2014, S.75ff]	106
6.1	Ergebnisse der nach ihrer Anpassungsgüte quantifizierten Funktionen nach [NETTER 2013a, S.46]	123
6.2	Ergebnisse der nach ihrer Simulationsergebnisgüte quantifizierten Batteriemodelle	125
6.3	Ergebnisse der nach ihrer Simulationsergebnisgüte quantifizierten Längsdynamikmodelle	125
6.4	Einteilung der Trainings- und Extrapolationsdatensätze nach [NETTER 2013b, S.4]	129
6.5	Ergebnisse der nach ihrer Simulationsergebnisgüte quantifizierten Batteriemodelle	131
B.1	Mögliche Werte der Zufallsgröße Z im Urnenversuch	164
B.2	Likelihood-Funktionswerte für $z = 2$ im Urnenversuch	167
B.3	Übersicht über die erfassten Daten nach [RISSANEN 1987, S.96]	168

Literaturverzeichnis

Quellenverzeichnis

- [AKAIKE 1973] H. Akaike (1973): Information Theory and an Extension of the Maximum Likelihood Principle. In: *2nd International Symposium on Information Theory*. Akademiai Kiado, Budapest, S. 267–281.
- [ANDRE 2008] F. Andre (2008): Modellierung einer Li-Ionen Batterie für Hybridfahrzeug-Simulationen: Modellbildung und Echtzeitsimulation technischer Systeme. Projektarbeit. Berlin: Technische Universität Berlin.
- [BALASUBRAMANIAN 1997] V. Balasubramanian (1997): Statistical Inference, Occam’s Razor, and Statistical Mechanics on the Space of Probability Distributions. In: *Neural Computation*. Jg. 9, Nr. 2, S. 349–368.
- [BEITZ 2001] W. Beitz und K.-H. Grote (2001): *Dubbel - Taschenbuch für den Maschinenbau: Mit mehr als 3000 Abbildungen und Tabellen*. 20., neubearb. u. erw. Aufl. Berlin [u.a.]: Springer.
- [BEWERSDORFF 2011] J. Bewersdorff (2011): *Statistik - wie und warum sie funktioniert: Ein mathematisches Lesebuch*. 1. Aufl. Wiesbaden: Vieweg + Teubner.
- [BLOCHWITZ 2011] T. Blochwitz und M. Otter (2011): The Functional Mockup Interface for Tool independent Exchange of Simulation Models. URL: https://trac.fmi-standard.org/export/700/branches/public/docs/Modelica2011/The_Functional_Mockup_Interface.pdf (abgerufen am 23.12.2013).

- [BOURBAKI 2015] N. Bourbaki (2015): Standardabweichung: Schätzung der Standardabweichung aus einer Stichprobe. URL: <http://www.mathepedia.de/Standardabweichung.aspx> (abgerufen am 23.02.2015).
- [BRAESS 2003] H.-H. Braess, Hrsg. (2003): Vieweg-Handbuch Kraftfahrzeugtechnik: Mit 69 Tabellen. 3. Aufl. ATZ-MTZ-Fachbuch. Wiesbaden: Vieweg.
- [BRAY 2008] T. Bray (2008): Extensible Markup Language (XML). URL: <http://www.w3.org/TR/xml/> (abgerufen am 10.09.2012).
- [BROOKS 1996] R. J. Brooks und A. M. Tobias (1996): Choosing the Best Model: Level of Detail, Complexity and Model Performance. In: *Mathematical and Computer Modelling*. Jg. 24, Nr. 4, S. 1–14.
- [BUNGARTZ 2009] H.-J. Bungartz, S. Zimmer, M. Buchholz und D. Pflüger (2009): Modellbildung und Simulation: Eine anwendungsorientierte Einführung. Berlin: Springer.
- [BURNHAM 2002] K. P. Burnham und D. R. Anderson (2002): Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach. New York: Springer.
- [BUSCH 2012] M. Busch (2012): Zur effizienten Kopplung von Simulationsprogrammen. Dissertation. Universität Kassel.
- [BUSCH 2010] M. Busch und B. Schweizer (2010): Numerical Stability and Accuracy of Different Co-Simulation Techniques: Analytical Investigations Based on a 2-DOF Test Model. In: *The First Joint International Conference on Multibody System Dynamics (IMSD 2010)*. Lappeenranta, Finnland.
- [BUSSE 2012] J. Busse et al. (2012): Was bedeutet eigentlich Ontologie? Ein Begriff aus der Philosophie im Licht verschiedener Disziplinen. Darmstadt: Springer. URL: https://www.fbi.h-da.de/fileadmin/personal/b.humm/Publikationen/Busse_et_al_-_Ontologie.pdf (abgerufen am 14.06.2014).

- [CULLMANN 2007] X.-N. Cullmann und K. Lambertz (2007): Software: Komplexität und Qualität von Software. In: *MSCoder*. Jg. 1, Nr. 2, S. 36–43.
- [DANIEL 1980] C. Daniel, F. S. Wood und J. W. Gorman (1980): Fitting Equations to Data: Computer Analysis of Multifactor Data for Scientists and Engineers. New York: Wiley.
- [DEUTSCHES INSTITUT FÜR NORMUNG 2010] Deutsches Institut für Normung (2010): Software-Engineering – Qualitätskriterien und Bewertung von Softwareprodukten (SQuaRE).
- [DKE 2009] DKE (2009): Internationales Elektrotechnisches Wörterbuch - Teil 351: Leittechnik.
- [DÖRING 2011] D. Döring (2011): Eine kurze Einführung in die Systemtheorie: Lehr- und Übungsbuch. 1. Aufl. Studium. Wiesbaden: Vieweg + Teubner.
- [DRONKA 2004] S. Dronka (2004): Die Simulation gekoppelter Mehrkörper- und Hydraulikmodelle mit Erweiterung für Echtzeitsimulation. Dissertation. Dresden: Technische Universität Dresden.
- [FIELD 2004] R. V. Field (2004): Methods for Model Selection in Applied Science and Engineering. Dissertation. Albuquerque: Sandia National Laboratories.
- [FREY 2008] T. Frey und M. Bossert (2008): Signal- und Systemtheorie: Mit 117 Abbildungen, 26 Tabellen, 64 Aufgaben mit Lösungen und 84 Beispielen. 2. Aufl. Studium. Wiesbaden: Vieweg + Teubner.
- [GEIMER 2006] M. Geimer, T. Krügler und P. Linsel (2006): Co-Simulation, gekoppelte Simulation oder Simulatorkopplung? Ein Versuch der Begriffsvereinheitlichung. In: *O+P Zeitschrift für Fluidtechnik - Aktorik, Steuerelektronik und Sensorik*. Jg. 50, Nr. 12, S. 572–576.
- [GRAMELSBERGER 2003] G. Gramelsberger (2003): Computersimulationen - Neue Instrumente der Wissensproduktion: Transdisziplinarität und

Heterogenität der Computational Science. Expertise. Berlin: Helmholtz-Zentrum für Kulturtechnik.

- [GROSSMANN 2000] K. Großmann (2000): Zukunftspotential Simulation. In: *Dresdner Tagung Simulation im Maschinenbau (SIM2000)*. Dresden, S. 2–23.
- [GRÜNWARD 2005] P. D. Grünwald (2005): A Tutorial Introduction to the Minimum Description Length Principle. In: *Advances in Minimum Description Length*. Hrsg. von P. D. Grünwald, I. J. Myung und M. A. Pitt. Cambridge, Massachusetts: MIT Press, S. 5–51.
- [HABERMAS 1973] J. Habermas (1973): Erkenntnis und Interesse. 1. Aufl. Bd. 1. Suhrkamp Taschenbuch Wissenschaft. Frankfurt (am Main): Suhrkamp.
- [HALSTEAD 1977] M. H. Halstead (1977): Elements of Software Science. Bd. 2. Operating and programming systems series. New York: ELSEVIER.
- [HESS 2009] C. Heß, J. Lambertz und T. Syldatke (2009): Ontologien im Praxiseinsatz: Erfahrungen bei der AUDI AG. In: *OBJEKTSpektrum*. Nr. 6, S. 46–50.
- [HILLER 1983] M. Hiller (1983): Mechanische Systeme: Eine Einführung in die analytische Mechanik und Systemdynamik. Berlin, Heidelberg, New York, Tokyo: Springer.
- [HITZLER 2008] P. Hitzler, M. Krötzsch, S. Rudolph und Y. Sure (2008): Semantic Web: Grundlagen. 1. Aufl. Berlin: Springer.
- [HORRIDGE 2009] M. Horridge (2009): A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools. Manchester: University of Manchester. URL: <http://people.cs.vt.edu/~kafura/ComputationalThinking/Class-Notes/Tutorial-Highlighted-Day1.pdf> (abgerufen am 17.06.2014).

- [HU 2012] X. Hu, S. Li und H. Peng (2012): A Comparative Study of Equivalent Circuit Models for Li-Ion Batteries. In: *Journal of Power Sources*. Nr. 198, S. 359–367.
- [IPARRAGUIRRE 2013] P. Iparraguirre (2013): Produktentwicklung mit virtuellen Prototypen. URL: https://wiki.zimt.uni-siegen.de/fertigungsautomatisierung/index.php/Produktentwicklung_mit_virtuellen_Prototypen (abgerufen am 23.04.2014).
- [JOSSEN 2006] A. Jossen und W. Weydanz (2006): Moderne Akkumulatoren richtig einsetzen: 36 Tabellen. 1. Aufl. Neusäß: Ubooks.
- [KLEIJNEN 1995] J. P. Kleijnen (1995): Verification and Validation of Simulation Models. In: *European Journal of Operational Research*. Nr. 82, S. 145–162.
- [KOKOL 1996] P. Kokol, J. Brest und V. Zumer (1996): Software Complexity - An alternative View. In: *IEEE International Conference on Systems, Man and Cybernetics*. Beijing, China, S. 2862–2867.
- [KOLANOSKI 2008] H. Kolanoski (2008): Statistische Methoden der Datenanalyse: Vorlesung im Sommersemester 2008. Berlin: Humboldt-Universität zu Berlin.
- [KOSSEL 2010] R. Kossel, M. Loeffler, M. Bodmann und W. Tegethoff (2010): Verteilte Simulation von Energiesystemen im Fahrzeug. In: *Berechnung und Simulation im Fahrzeugbau*. Düsseldorf: VDI, S. 103–121.
- [KRAFT 1949] L. G. Kraft (1949): A Device for Quantizing, Grouping, and Coding Amplitude-Modulated Pulses. Masterarbeit. Pennsylvania: Massachusetts Institute of Technology.
- [KRAPP 2007] M. Krapp (2007): Methoden der Statistik: Lehr- und Arbeitsbuch der angewandten Statistik für Sozial- und Wirtschaftswissenschaftler. 1. Aufl. Wiesbaden: Vieweg + Teubner.

- [KÜBLER 2000] R. Kübler und W. Schiehlen (2000): Two Methods of Simulator Coupling. In: *Mathematical and Computer Modeling of Dynamical Systems*. Jg. 6, Nr. 2, S. 93–113.
- [KÜBLER 2000] R. Kübler (2000): Modulare Modellierung und Simulation mechatronischer Systeme. Bd. 327. Stuttgart: VDI.
- [LANTERMAN 2001] A. D. Lanterman (2001): Schwarz, Wallace, and Rissanen: Intertwining Themes in Theories of Model Selection. In: *International Statistical Review*. Jg. 69, Nr. 2, S. 185–212.
- [LEYDOLD 2000] J. Leydold (2000): Mathematik für Ökonomen: Formale Grundlagen der Wirtschaftswissenschaften. 2., unwesentlich veränd. Aufl. Wolls Lehr- und Handbücher der Wirtschafts- und Sozialwissenschaften. München: Oldenbourg.
- [LÜDECKE 2003] A. Lüdecke (2003): Simulationsgestützte Verfahren für den Top-Down-Entwurf heterogener Systeme. Dissertation. Duisburg-Essen: Universität Duisburg Essen.
- [LUDWIG-MAYERHOFER 2006] M. Ludwig-Mayerhofer (2006): Das lineare Regressionsmodell. Vorlesung. Siegen: Universität Siegen.
- [LUND 2008] C. Lund, W. Maister, C. Lange und B. Beyer (2008): Innovation durch Co-Simulation. In: *Wärmemanagement des Kraftfahrzeugs*. Hrsg. von P. Steinberg. Renningen: expert, S. 163–187.
- [LUNZE 2013] J. Lunze (2013): Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen. 9. Aufl. Berlin, Heidelberg: Springer-Vieweg.
- [McCABE 1976] T. McCabe (1976): A Complexity Measure. In: *IEEE Transactions on Software Engineering*. Jg. 2, Nr. 4, S. 308–320.
- [MURRAY-SMITH 1995] D. J. Murray-Smith (1995): Advances in Simulation Model Validation: Theory, Software and Applications. In: *Eurosim '95 Simulation Congress: Proceedings of the 1995 EUROSIM Conference*. Hrsg. von F. Breitenacker und I. Husinsky. Wien, Österreich: ELSEVIER, S. 75–84.

- [MYUNG 2000] I. J. Myung (2000): The Importance of Complexity in Model Selection. In: *Journal of Mathematical Psychology*. Nr. 44, S. 190–204.
- [MYUNG 2003] — (2003): Tutorial on Maximum Likelihood Estimation. In: *Journal of Mathematical Psychology*. Nr. 47, S. 90–100.
- [MYUNG 2004] I. J. Myung und M. A. Pitt (2004): Model Comparison Methods. In: *Methods in Enzymology*. Hrsg. von L. Brand und M. L. Johnson. London: Academic Press, S. 351–366.
- [MYUNG 2005] I. J. Myung, M. A. Pitt und W. Kim (2005): Model Evaluation, Testing and Selection. In: *Handbook of Cognition*. Hrsg. von K. Lamberts und R. L. Goldstone. London: SAGE, S. 422–436.
- [MYUNG 1999] I. J. Myung, S. Zhang und M. A. Pitt (1999): Geometric Complexity and Minimum Description Length Principle. In: *Annual Mathematical Psychology Meeting*. Santa Cruz, Kalifornien, USA, S. 1–12.
- [NOLLAU 2009] R. Nollau (2009): Modellierung und Simulation technischer Systeme: Eine praxisnahe Einführung. 1. Aufl. Berlin, Heidelberg: Springer.
- [OTTER 2009] M. Otter (2009): Modelica Overview. URL: <https://www.modelica.org/education/educational-material/lecture-material/english/ModelicaOverview.pdf> (abgerufen am 02.09.2014).
- [PAGE 1991] B. Page (1991): Diskrete Simulation: Eine Einführung mit Modula-2. Springer-Lehrbuch. Berlin, Heidelberg: Springer.
- [PAPULA 2009] L. Papula (2009): Mathematische Formelsammlung für Ingenieure und Naturwissenschaftler: mit zahlreichen Rechenbeispielen und einer ausführlichen Integraltafel. 10. Aufl. Wiesbaden: Vieweg + Teubner.
- [PARTON 2011] J. Parton, Hrsg. (2011): The American Heritage Dictionary of the English language. 5. Aufl. Boston: Houghton Mifflin Harcourt.

- [PELZ 2001] G. Pelz (2001): Modellierung und Simulation mechatronischer Systeme: Vom Chip- zum Systementwurf mit Hardwarebeschreibungssprachen. Heidelberg: Hüthig.
- [PITT 2002] M. A. Pitt, I. J. Myung und S. Zhang (2002): Toward a Method of Selecting Among Computational Models of Cognition. In: *Psychological Review*. Jg. 109, Nr. 3, S. 472–491.
- [RABE 2008] M. Rabe, S. Spieckermann und S. Wenzel (2008): Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken. Berlin, Heidelberg: Springer.
- [RE 2009] L. del Re (2009): Adaptive und prädiktive Regelung. Vorlesung. Linz: Johannes Kepler Universität Linz.
- [RISSANEN 1978] J. Rissanen (1978): Modeling by Shortest Data Description. In: *Automatica*. Jg. 5, Nr. 14, S. 465–471.
- [RISSANEN 1986] — (1986): Stochastic Complexity and Modelling. In: *The Annals of Statistics*. Jg. 14, Nr. 3, S. 1080–1100.
- [RISSANEN 1987] — (1987): Stochastic Complexity and the MDL Principle. In: *Econometric Reviews*. Jg. 6, Nr. 1, S. 85–102.
- [RÜGER 1999] B. Rüger (1999): Test- und Schätztheorie: Grundlagen. Bd. 1. Lehr- und Handbücher der Statistik. München, Wien: Oldenbourg.
- [RUHM 2003] K. Ruhm (2003): Abtasten kontinuierlicher Signale - Qualitative Beschreibung. Zürich, Schweiz. URL: <http://www.mmm.ethz.ch/dok01/d0000100.pdf> (abgerufen am 04.07.2014).
- [RUHM 2008] — (2008): Kontinuierliche, Diskrete Signale. Zürich, Schweiz. URL: <http://www.mmm.ethz.ch/dok01/d0000348.pdf> (abgerufen am 04.07.2014).
- [RÜPING 2014] T. Rüping (2014): AUTOSAR: Basisinformationen. URL: http://www.autosar.org/fileadmin/files/basic_information/

- AUTOSARBasicInformationShortVersion_DE . pdf (abgerufen am 11.02.2015).
- [SCHÄPPI 2005] B. Schäppi, M. M. Andreasen, M. Kirchgeorg, Radermacher und F.-J., Hrsg. (2005): Handbuch Produktentwicklung. München, Wien: Carl-Hanser.
- [SCHOMAKER 2006] M. Schomaker (2006): Neue Ansätze für Kriterien zur Modellselektion bei Regressionsmodellen unter Berücksichtigung der Problematik fehlender Daten. Diplomarbeit. München: Ludwig-Maximilians-Universität München.
- [SCHOMAKER 2010] — (2010): Selektieren und Kombinieren von Modellen unter Berücksichtigung der Problematik fehlender Daten. Dissertation. München: Ludwig Maximilian Universität München.
- [SCHOUPS 2012] G. Schoups (2012): Model structural uncertainty. URL: <http://www.citg.tudelft.nl/en/about-faculty/departments/watermanagement/sections/water-resources/leerstoelen/wrm/research/all-projects/projects/current-projects/model-prediction-uncertainty/> (abgerufen am 02.09.2014).
- [SCHRAMM 2009] D. Schramm, R. Bardini und M. Hiller (2009): Modellbildung und Simulation der Dynamik von Kraftfahrzeugen. 1. Aufl. Berlin, Heidelberg: Springer.
- [SCHÜSSLER 2008] H. Schüßler (2008): Digitale Signalverarbeitung: Analyse diskreter Signale und Systeme. 5. Aufl. Berlin, Heidelberg: Springer.
- [SCHWARZ 1978] G. Schwarz (1978): Estimating the Dimension of a Model. In: *The Annals of Statistics*. Jg. 6, Nr. 2, S. 461–464.
- [SCLOVE 2005] S. Sclove (2005): The Sample Standard Deviation. URL: <http://www.uic.edu/classes/idsc/ids571/samplvar.pdf> (abgerufen am 23.02.2015).
- [SEIFFERT 2008] U. Seiffert und G. Rainer, Hrsg. (2008): Virtuelle Produktentstehung für Fahrzeug und Antrieb im Kfz: Prozesse, Komponenten, Beispiele aus der Praxis. 1. Aufl. Wiesbaden: Vieweg + Teubner.

- [STELAND 2010] A. Steland (2010): Basiswissen Statistik: Kompaktkurs für Anwender aus Wirtschaft, Informatik und Technik. 2. Aufl. Berlin, Heidelberg: Springer.
- [STÜBER 2011] N. Stüber (2011): Challenges and Solutions for an efficient Use of Co-Simulation. In: *The Role of CAE in System Simulation*. Hrsg. von W. Moretti. Bernau am Chiemsee: NAFEMS, S. 75–83.
- [STÜRMER 2010] I. Stürmer, H. Pohlheim und T. Rogier (2010): Berechnung und Visualisierung der Modellkomplexität bei der modellbasierten Entwicklung sicherheits-relevanter Software. Stuttgart.
- [VALASEK 2009] M. Valasek (2009): Modeling, Simulation and Control of Mechatronical Systems. In: *Simulation Techniques for Applied Dynamics*. Hrsg. von M. Arnold und W. Schiehlen. Wien: Springer, S. 75–140.
- [VANPAEMEL 2009] W. Vanpaemel (2009): Measuring Model Complexity with the Prior Predictive. In: *Advances in Neural Information Processing Systems*. Bd. 22. Vancouver, British Columbia, Canada: Neural Information Processing Systems. Printed from e-media with permission by: Curran Associates, Inc, S. 1–9.
- [VDI 2014] VDI (2014): Simulation von Logistik-, Materialfluss- und Produktionssystemen.
- [WINER 1968] N. Winer (1968): Kybernetik: Regelung und Nachrichtenübertragung in Lebewesen und Maschine. Reinbek b. Hamburg: Rowohlt.
- [ZEITZ 1987] M. Zeitz (1987): Simulationstechnik. In: *Chemie Ingenieur Technik*. Jg. 59, Nr. 6, S. 464–469.

Betreute Studienarbeiten

- [BURGER 2012] S. Burger (2012): Generische Schnittstelle für Fahrzeugmessdaten und deren Verifikation. Bachelorarbeit. Karlsruhe: Hochschule Karlsruhe.

- [BUTTERER 2012] B. Butterer (2012): Validierung, Erweiterung und Potentialanalyse eines funktionsorientierten Elektrofahrzeugmodells. Bachelorarbeit. Karlsruhe: Hochschule Karlsruhe.
- [CHU 2012] X. Chu (2012): Komplexitätsreduzierung und Potentialanalyse eines funktionsorientierten Elektrofahrzeugs. Diplomarbeit. Stuttgart: Universität Stuttgart.
- [GHORBEL 2014] Y. Ghorbel (2014): Ontologiebasierte Selektion von Simulationsmodellen am Beispiel eines Elektrofahrzeugs. Diplomarbeit. Stuttgart: Universität Stuttgart.
- [SCHREYER 2012] J. Schreyer (2012): Gestaltung eines Frameworks zur Funktionsorientierten Modularisierung von Elektrofahrzeugmodellen. Diplomarbeit. München: Hochschule München.

Publikationen

- [NETTER 2013a] F. Netter, F. Gauterin und B. Butterer (2013a): Real-Data Validation of Simulation Models in a Function-Based Modular Framework. In: *6th IEEE International Conference on Software Testing, Verification and Validation (ICST 2013)*. Luxemburg: IEEE, S. 41–47.
- [NETTER 2013b] F. Netter, F. Gauterin und X. Chu (2013b): Complexity Adaptation of Simulation Models in a Function-Based Modular Framework. In: *5th IEEE International Conference on Modeling, Simulation and Applied Optimization (ICMSAO 2013)*. Hammamet, Tunesien: IEEE.
- [NETTER 2013c] F. Netter, F. Gauterin und J. Schreyer (2013c): Model Coupling with a Function-Based Modular Framework for Entire-System Simulation. In: *SAE 2013 World Congress*. Detroit, Michigan, USA: SAE International.

Karlsruher Schriftenreihe Fahrzeugsystemtechnik (ISSN 1869-6058)

Herausgeber: FAST Institut für Fahrzeugsystemtechnik

Die Bände sind unter www.ksp.kit.edu als PDF frei verfügbar
oder als Druckausgabe bestellbar.

- Band 1** Urs Wiesel
Hybrides Lenksystem zur Kraftstoffeinsparung im schweren Nutzfahrzeug. 2010
ISBN 978-3-86644-456-0
- Band 2** Andreas Huber
Ermittlung von prozessabhängigen Lastkollektiven eines hydrostatischen Fahrtriebsstrangs am Beispiel eines Teleskopladens. 2010
ISBN 978-3-86644-564-2
- Band 3** Maurice Bliesener
Optimierung der Betriebsführung mobiler Arbeitsmaschinen. Ansatz für ein Gesamtmaschinenmanagement. 2010
ISBN 978-3-86644-536-9
- Band 4** Manuel Boog
Steigerung der Verfügbarkeit mobiler Arbeitsmaschinen durch Betriebslasterfassung und Fehleridentifikation an hydrostatischen Verdrängereinheiten. 2011
ISBN 978-3-86644-600-7
- Band 5** Christian Kraft
Gezielte Variation und Analyse des Fahrverhaltens von Kraftfahrzeugen mittels elektrischer Linearaktuatoren im Fahrwerksbereich. 2011
ISBN 978-3-86644-607-6
- Band 6** Lars Völker
Untersuchung des Kommunikationsintervalls bei der gekoppelten Simulation. 2011
ISBN 978-3-86644-611-3
- Band 7** 3. Fachtagung
Hybridantriebe für mobile Arbeitsmaschinen. 17. Februar 2011, Karlsruhe. 2011
ISBN 978-3-86644-599-4

Karlsruher Schriftenreihe Fahrzeugsystemtechnik (ISSN 1869-6058)

Herausgeber: FAST Institut für Fahrzeugsystemtechnik

- Band 8** Vladimir Iliev
Systemansatz zur anregungsunabhängigen Charakterisierung des Schwingungskomforts eines Fahrzeugs. 2011
ISBN 978-3-86644-681-6
- Band 9** Lars Lewandowitz
Markenspezifische Auswahl, Parametrierung und Gestaltung der Produktgruppe Fahrerassistenzsysteme. Ein methodisches Rahmenwerk. 2011
ISBN 978-3-86644-701-1
- Band 10** Phillip Thiebes
Hybridantriebe für mobile Arbeitsmaschinen. Grundlegende Erkenntnisse und Zusammenhänge, Vorstellung einer Methodik zur Unterstützung des Entwicklungsprozesses und deren Validierung am Beispiel einer Forstmaschine. 2012
ISBN 978-3-86644-808-7
- Band 11** Martin Gießler
Mechanismen der Kraftübertragung des Reifens auf Schnee und Eis. 2012
ISBN 978-3-86644-806-3
- Band 12** Daniel Pies
Reifenungleichförmigkeitserregter Schwingungskomfort – Quantifizierung und Bewertung komfortrelevanter Fahrzeugschwingungen. 2012
ISBN 978-3-86644-825-4
- Band 13** Daniel Weber
Untersuchung des Potenzials einer Brems-Ausweich-Assistenz. 2012
ISBN 978-3-86644-864-3
- Band 14** **7. Kolloquium Mobilhydraulik.**
27./28. September 2012 in Karlsruhe. 2012
ISBN 978-3-86644-881-0
- Band 15** 4. Fachtagung
Hybridantriebe für mobile Arbeitsmaschinen
20. Februar 2013, Karlsruhe. 2013
ISBN 978-3-86644-970-1

Karlsruher Schriftenreihe Fahrzeugsystemtechnik (ISSN 1869-6058)

Herausgeber: FAST Institut für Fahrzeugsystemtechnik

- Band 16** Hans-Joachim Unrau
Der Einfluss der Fahrbahnoberflächenkrümmung auf den Rollwiderstand, die Cornering Stiffness und die Aligning Stiffness von Pkw-Reifen. 2013
ISBN 978-3-86644-983-1
- Band 17** Xi Zhang
Untersuchung und Entwicklung verschiedener Spurführungsansätze für Offroad-Fahrzeuge mit Deichselverbindung. 2013
ISBN 978-3-7315-0005-6
- Band 18** Stefanie Grollius
Analyse des gekoppelten Systems Reifen-Hohlraum-Rad-Radführung im Rollzustand und Entwicklung eines Rollgeräuschmodells. 2013
ISBN 978-3-7315-0029-2
- Band 19** Tobias Radke
Energieoptimale Längsführung von Kraftfahrzeugen durch Einsatz vorausschauender Fahrstrategien. 2013
ISBN 978-3-7315-0069-8
- Band 20** David Gutjahr
Objektive Bewertung querdynamischer Reifeneigenschaften im Gesamtfahrzeugversuch. 2014
ISBN 978-3-7315-0153-4
- Band 21** Neli Ovcharova
Methodik zur Nutzenanalyse und Optimierung sicherheitsrelevanter Fahrerassistenzsysteme. 2014
ISBN 978-3-7315-0176-3
- Band 22** Marcus Geimer, Christian Pohlandt
Grundlagen mobiler Arbeitsmaschinen. 2014
ISBN 978-3-7315-0188-6
- Band 23** Timo Kautzmann
Die mobile Arbeitsmaschine als komplexes System. 2014
ISBN 978-3-7315-0187-9

Karlsruher Schriftenreihe Fahrzeugsystemtechnik (ISSN 1869-6058)

Herausgeber: FAST Institut für Fahrzeugsystemtechnik

- Band 24** Roman Weidemann
Analyse der mechanischen Randbedingungen zur Adaption der oszillierenden Hinterschneidtechnik an einen Mobilbagger. 2014
ISBN 978-3-7315-0193-0
- Band 25** Yunfan Wei
Spurführungsregelung eines aktiv gelenkten Radpaars für Straßenbahnen. 2014
ISBN 978-3-7315-0232-6
- Band 26** David Schmitz
Entwurf eines fehlertoleranten Lenkventils für Steer-by-Wire Anwendungen bei Traktoren. 2014
ISBN 978-3-7315-0264-7
- Band 27** Christian Schwab
Beitrag zu einer universellen Baggerschnittstelle zur Übertragung elektrischer und hydraulischer Leistung sowie elektronischer Signale für komplexe Anbaugeräte. 2014
ISBN 978-3-7315-0281-4
- Band 28** Peter Dengler
Untersuchung zum effizienten Betrieb von Hydraulikzylindern in Konstantdrucksystemen unter Verwendung einer Zwischendruckleitung. 2015
ISBN 978-3-7315-0295-1
- Band 29** Manuel Bös
Untersuchung und Optimierung der Fahrkomfort- und Fahrdynamikeigenschaften von Radladern unter Berücksichtigung der prozessspezifischen Randbedingungen. 2015
ISBN 978-3-7315-0310-1
- Band 30** 5. Fachtagung
Hybride und energieeffiziente Antriebe für mobile Arbeitsmaschinen
25. Februar 2015, Karlsruhe. 2015
ISBN 978-3-7315-0323-1
- Band 31** Michael Eckert
Energieoptimale Fahrdynamikregelung mehrmotoriger Elektrofahrzeuge. 2015
ISBN 978-3-7315-0332-3

Karlsruher Schriftenreihe Fahrzeugsystemtechnik (ISSN 1869-6058)

Herausgeber: FAST Institut für Fahrzeugsystemtechnik

- Band 32** Martin Scherer
Beitrag zur Effizienzsteigerung mobiler Arbeitsmaschinen. Entwicklung einer elektrohydraulischen Bedarfsstromsteuerung mit aufgeprägtem Volumenstrom. 2015
ISBN 978-3-7315-0339-2
- Band 33** Rinaldo Arnold
Automatische Abstimmung der Sekundärseite eines dreiphasigen Systems zur berührungslosen induktiven Energieübertragung. 2015
ISBN 978-3-7315-0355-2
- Band 34** Johannes Gültlinger
Kraftübertragung und Fahrbahnverschleiß durch Spikereifen. 2015
ISBN 978-3-7315-0358-3
- Band 35** Thorsten Dreher
Energieeffizienz von Konstantdrucksystemen mit sekundärgeregelten Antrieben beim Einsatz in mobilen Arbeitsmaschinen. 2015
ISBN 978-3-7315-0377-4
- Band 36** Steffen Kölling
Konzeptionelle Untersuchung zur Neigekompensation von Stromabnehmern. 2015
ISBN 978-3-7315-0387-3
- Band 37** Michael Fritz
Entwicklungswerkzeuge für die Fahrzeugklimatisierung von Nutzfahrzeugen. 2015
ISBN 978-3-7315-0384-2
- Band 38** Ralf Oberfell
Stochastische Simulation von Energieflüssen im Nutzfahrzeug Ein einsatzorientiertes Bewertungs- und Optimierungsverfahren. 2015
ISBN 978-3-7315-0403-0
- Band 39** Christoph Sturm
Bewertung der Energieeffizienz von Antriebssystemen mobiler Arbeitsmaschinen am Beispiel Bagger. 2015
ISBN 978-3-7315-0404-7

Karlsruher Schriftenreihe Fahrzeugsystemtechnik (ISSN 1869-6058)

Herausgeber: FAST Institut für Fahrzeugsystemtechnik

Band 40 Florian Netter
Komplexitätsadaption integrierter Gesamtfahrzeugsimulationen. 2016
ISBN 978-3-7315-0414-6

Steigende Rechenleistung und sinkende Hardwarekosten fördern den Einsatz von Simulationen. Dabei nehmen mit fortschreitender Detaillierung virtueller Prototypen sowohl die Anzahl an Untersuchungsmöglichkeiten als auch die Genauigkeit der Ergebnisse und Vorhersagen zu.

Gemäß dem heuristischen Forschungsprinzip der Sparsamkeit sollte ein Modell jedoch immer nur so komplex wie nötig und nicht so komplex wie möglich sein. Dadurch kann einerseits eine hohe Anpassungsgüte an gegebene Messdaten bei gleichzeitig hohem Extrapolationsvermögen auf noch unbekannte Daten gewährleistet werden. Andererseits können bei der Simulation virtueller Prototypen Testaufwand und Simulationsdauer reduziert werden.

Zu diesem Zweck werden Kennzahlen für Simulationsmodellkomplexität, Anpassungsgüte und Extrapolationsvermögen erforscht. Das Potential dieser Attribute wird anhand mehrerer Evaluationsbeispiele analysiert und bewertet. Als Ergebnis ist es möglich in Abhängigkeit unterschiedlicher Simulationszwecke aus einer Vielzahl konkurrierender Modelle die für einen solchen Kompromiss jeweils optimalen Simulationsmodellkomplexitäten zu identifizieren. Um die Quantifizierungsmerkmale für virtuelle Prototypen nutzen zu können werden diese zudem auf Basis eines skalierenden Modularisierungsansatzes in einem Prozess zur Kopplung simulationswerkzeugübergreifender Systeme verortet.

