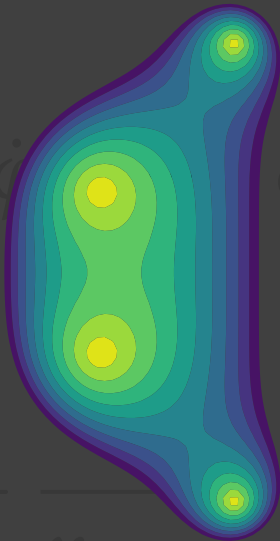


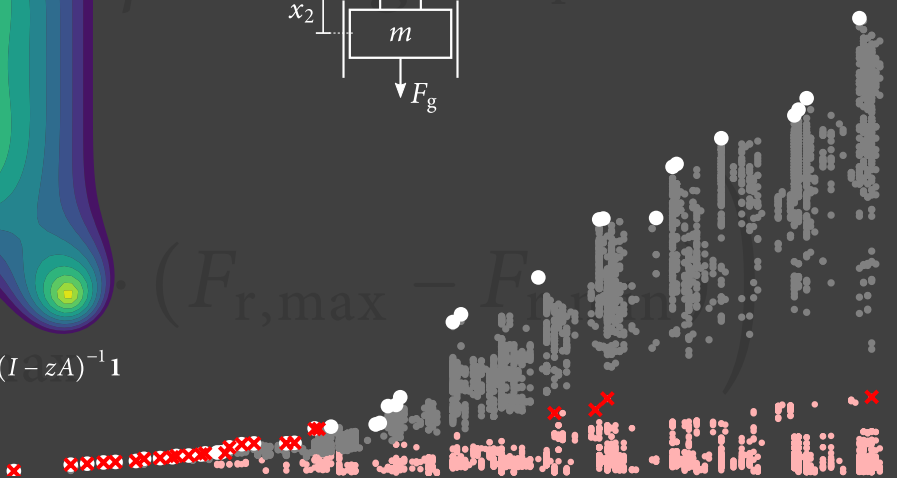
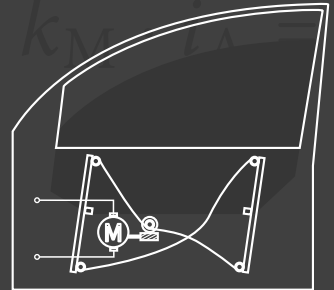
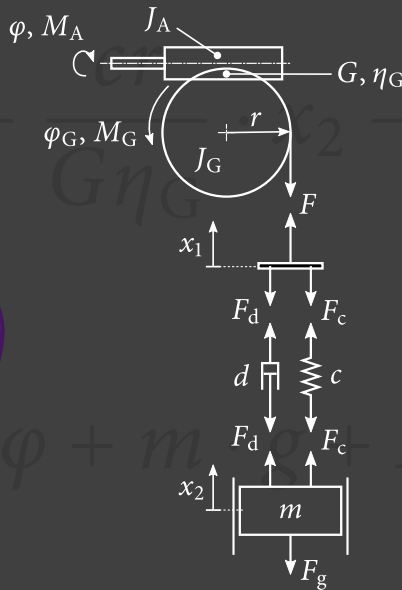
Dennis Kähler

Multiskalensimulation von mechatronischen Komponenten zur Anwendung in Hardware-in-the-Loop Tests

$$S = \{z \in \mathbb{C} : |R(z)| \leq 1\}$$



$$R(z) = 1 + zb^T (I - zA)^{-1} \mathbf{1}$$



Multiskalensimulation von mechatronischen Komponenten zur Anwendung in Hardware-in-the-Loop Tests

**Vom Promotionsausschuss der
Technischen Universität Hamburg**
zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

von
Dennis Kähler

aus
Pinneberg

2021

1. Gutachter: Prof. Dr.-Ing. G. Ackermann
 2. Gutachter: Prof. Dr.-Ing. F. Thielecke
- Prüfungsvorsitzender: Prof. Dr.-Ing. R. Seifried

Tag der mündlichen Prüfung: 21.08.2020

Lizenz:



Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Das Werk steht unter der Creative-Commons-Lizenz Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International (CC BY-SA 4.0, <https://creativecommons.org/licenses/by-sa/4.0/legalcode.de>). Ausgenommen von der oben genannten Lizenz sind Teile, Abbildungen und sonstiges Drittmaterial, wenn anders gekennzeichnet.

DOI: <https://doi.org/10.15480/882.3171>

ORCID: Dennis Kähler
<https://orcid.org/0000-0001-5524-9661>

Für Kathi und Frida

Kurzfassung

Multiskalensimulation von mechatronischen Komponenten zur Anwendung in Hardware-in-the-Loop Tests

Aufgrund immer mehr und immer komplexeren elektronischen Steuergeräten im Automobil steigt der Bedarf an angemessenen Testmöglichkeiten. In dieser Arbeit wird ein Konzept erarbeitet, welches den hochdynamischen Anforderungen aufgrund der automobilen Verbraucher gerecht wird. Dies wird mit einer parallelen heterogenen Simulationsarchitektur erreicht, bei der die unterschiedlichen Frequenzanteile des Eingangssignals auf unterschiedlicher Hardware parallel simuliert werden. Zusätzlich werden einfach zu parametrierende generische Modelle entwickelt. Mit einer exemplarischen Umsetzung wird eine Schrittweite von 385 ns und eine gute Übereinstimmung mit realen Messwerten erreicht.

Abstract

Multiscale simulation of mechatronic components for use in hardware-in-the-loop tests

Due to more and more complex electronic control units in automobiles, the need for appropriate test capabilities is increasing. In this work a concept is developed, which meets the highly dynamic requirements due to the automotive devices. This is achieved with a parallel heterogeneous simulation architecture, where the different frequency components of the input signal are simulated in parallel on different hardware. Additionally, easy to parameterize generic models are developed. With an exemplary implementation a step size of 385 ns and a good agreement with real measurement data is achieved.

Inhaltsverzeichnis

Symbolverzeichnis	iv
Abkürzungen	vii
1 Einleitung	1
2 Stand der Wissenschaft und Technik und Ziel der Arbeit	5
2.1 Rechenhardware	5
2.1.1 Arten von Prozessoren	5
2.1.2 Rechenbeschleunigung	6
2.1.3 Entwicklung der Leistungsfähigkeit	7
2.2 HIL-Simulationen	8
2.2.1 Heterogene HIL-Co-Simulation	10
2.2.2 Leistungselektronik	11
2.2.3 Analog-Digital-Wandler	12
2.3 Ziele und Vorgehen der Arbeit	12
3 Konzept	15
3.1 Anforderungen	15
3.1.1 Schrittweite	15
3.1.2 Genauigkeit	16
3.1.3 Implementierung von Modellen	16
3.1.4 Parametrierung von Modellen	16
3.1.5 Bedienung der HIL-Simulation	17
3.2 Gesamtkonzept	17
3.2.1 Mögliche Umsetzungen	18
3.2.2 Detaillierte Architektur	20
3.3 Simulationsmodelle	22
3.4 Numerische Lösung	23
4 Filterung	25
4.1 Berechnung der Filterparameter	26
4.2 Architektur Frequenzweiche	27
4.2.1 Verwendung von zwei Filtern	30
4.2.2 Verwendung von einem Filter	31
4.3 Ergebnisse	34
4.4 Digitale Filterung	36

5	Modellbildung	39
5.1	Messungen an repräsentativen Verbrauchern	39
5.2	Anforderungen an die Modelle	43
5.3	Modellierung und Parametrierung des Fensterhebers	44
5.3.1	Motor (PMDCM)	44
5.3.2	Mechanisches System Fensterheber	46
5.3.3	Reibung	49
5.3.4	Seillose	52
5.3.5	Kommutierung	53
5.3.6	Zustände	55
5.4	Parametrierung des Fensterhebers	56
5.4.1	Elektrischer Widerstand des Ankerkreises	57
5.4.2	Induktivität des Ankerkreises	59
5.4.3	Motorkonstante	59
5.4.4	Reibkoeffizient des Motors	60
5.4.5	Massenträgheitsmoment des Ankers	61
5.4.6	Getriebe	62
5.4.7	Federsteifigkeit und Dämpfungsmaß des Seilzugsystems 63	63
5.4.8	Masse des Fenstersystems	64
5.4.9	Reibkräfte	64
5.4.10	Parameter Seillose	65
5.4.11	Kommutierungsparameter	66
5.4.12	Modellierung von Störfällen	66
5.4.13	Validierung und Ergebnisse	69
6	Numerische Lösung	71
6.1	Fehler	72
6.2	Stabilität	75
6.3	Modellanalyse	80
6.4	Laufzeitbestimmung	82
6.5	Automatische Auswahl des Löasers	82
6.6	Modellnotation	83
7	Exemplarische Implementierung und Ergebnisse	85
7.1	Komponenten der Simulationshardware	85
7.2	Zeitverhalten	86
7.3	Genauigkeit	89
7.4	Leistungselektronik	89
7.5	Ergebnisse	89
7.5.1	HIL-Simulation R-L-Glied	90
7.5.2	PHIL-Simulation Fensterheber	91

8	Zusammenfassung und Ausblick	95
8.1	Zusammenfassung	95
8.2	Ausblick	97
A	Anhang	99
A.1	Erklärung Abbildung: CPU Leistungsfähigkeit bei Fließkom- maberechnungen	99
A.2	Benötigte Schrittweite aufgrund der PWM	101
A.3	Zustände Fensterhebermodell	102
A.4	Zusammenfassung der Modellgleichungen des Fensterhebers .	103
A.5	Zusammenfassung der Modellparameter des Fensterhebers . .	104
A.6	Modellbeschreibung einer Halogenlampe	105
A.7	Entwicklung der Samplerate von SAR ADCs	106
	Literatur	107

Symbolverzeichnis

Symbol	Bezeichnung	Einheit
α	Umrechnungsfaktor	–
α_e	Ideelle Polbedeckung	rad
ε^*	Bezogener numerischer Fehler	–
ε	Numerischer Fehler der Stromsimulation	A
η	Wirkungsgrad	–
η_G	Getriebewirkungsgrad	–
λ	Eigenwert	s^{-1}
ω	Winkelgeschwindigkeit	rad s^{-1}
φ_L	Rotationswinkel Seillose	rad
φ	Rotationswinkel	rad
τ	Zeitkonstante	s
τ_p	Polteilung	–
A	Systemmatrix	–
a	Halbe Anzahl paralleler Wicklungszweige	–
a	Stromeinflussfaktor (Rippelparameter)	–
A_d	Amplitude der Dämpfung	dB
A_E	Systemmatrix aus Linearisierung	–
B	Eingangsmatrix	–
b	Reibkoeffizient	N m s rad^{-1}
B_{2000}	Baselinewert aus <i>SPEC</i> [®] 2000 CPU-Test	–
B_{2006}	Baselinewert aus <i>SPEC</i> [®] 2006 CPU-Test	–
$B_{\delta,m}$	Konstantes maximales Luftspaltfeld	T
C	Ausgangsmatrix	–
C	Kapazität	F
c	Federkonstante	N m^{-1}
D	Dämpfung	dB
D	Durchgangsmatrix	–
d	Dämpfungsverhalten des Filters	dB/Dekade
d	Federkonstante	N s m^{-1}
E	Einheitsmatrix	–
F	Vektor von Funktionen	–
F_b	Losbrechkraft bei vereistem Fenster	N
F	Kraft	N

Symbol	Bezeichnung	Einheit
f_b	Normierte Losbrechkraft bei vereistem Fenster	–
f	Funktionen	–
f	Frequenz	Hz
f_c	Grenzfrequenz des Filters	Hz
f_{mod}	Modulationsfrequenz der PWM	Hz
f_{pwm}	PWM Frequenz	Hz
F_r	Reibkraft	N
f_s	Abtastfrequenz	Hz
G	Getriebeübersetzung	–
G	Übertragungsfunktion	–
H	Umrechnungsfaktor Strom / Kraft	N A^{-1}
h	Schrittweite	s
h	Höhe	m
h_m	Vektor der maximalen stabile Schrittweiten je Eigenwert	s
H_{max}	Vektor der maximalen stabilen Schrittweiten für alle Entwicklungspunkte	s
h_{max}	Maximale stabile Schrittweite Gesamtsystem	s
I_N	Nennstrom	A
i	Strom	A
I	Strom	A
i_M	Motorstrom	A
i_r	Rippelstrom	A
$I_{R\infty}$	Scheitelwert Rippelstrom bei hoher Drehzahl	A
ω_x	Eck-Winkelgeschwindigkeit	rad s^{-1}
J	Jacobi-Matrix	–
J	Massenträgheitsmoment	kg m^2
K	Anzahl Kommutatorsegmente	–
K	Verstärkungsfaktor	–
k	Anzahl diskreter Punkte je Zustandsvariable	–
k_M	Motorkonstante	N m A^{-1}
L	Induktivität	H
L_A	Ankerinduktivität	H
l_e	Ideelle Eisenlänge	m
M	Moment	N
m	Anzahl Differentialgleichungen	–
m	Masse	kg
M_L	Motormoment	N
M_G	Moment nach Getriebe	N

Symbol	Bezeichnung	Einheit
M_L	Lastmoment	N
N	Filterordnung	–
N_c	Spulenwindungszahl	–
O	Ordnung des Lösers	–
P_N	Nennleistung	W
p	Polpaarzahl	–
P_{FP}	normierte Punkte aus dem SPEC-Test	–
R	Stabilitätsfunktion	–
R	elektrischer Widerstand	Ω
r	Radius	m
R_A	Ankerwiderstand	Ω
R_u	Elektrischer Widerstand bei Umgebungstemperatur	Ω
s	komplexe Frequenz	–
T	Temperatur	K
t	Zeit	s
t_L	Maximal erlaubte Latenz der Simulation	s
T_u	Umgebungstemperatur	K
U_N	Nennspannung	V
u	normiertes Eingangssignal	–
u	Eingangsgrößenvektor	–
u	elektrische Spannung	V
u_A	Ankerspannung	V
u_{hp}	Spannung nach Hochpassfilter	V
u_{hp}	normiertes Ausgangssignal Hochpassfilter	–
u_{tp}	Spannung nach Tiefpassfilter	V
u_{tp}	normiertes Ausgangssignal Tiefpassfilter	–
x	Zustandsvektor	–
x	Ortskoordinate	m
x_2	Ortskoordinate Fensterscheibe	m
y	Ausgangsgrößenvektor	–
z	Anzahl Kommutatorlamellen	–
z	komplexe Frequenz der z-Transformation	–

Abkürzungen

$\Delta\Sigma$	Delta-Sigma
μC	Mikrocontroller
μP	Mikroprozessor
ADC	Analog Digital Converter
BLDC	Brushless Direct Current
BLDCM	Brushless Direct Current Motor
Bus	Binary Unit System
CAN	Controller Area Network
CPU	Central Processing Unit
DAC	Digital Analog Converter
DAE	differential-algebraic system of equations
DGL	Differentialgleichung
DSP	Digital Signal Processor
DUT	Device Under Test
ECU	Electronic Control Unit
FEM	Finite-Elemente-Methode
FFT	Fast Fourier Transformation
FIFO	First In First Out
FIR	Finite Impulse Response
FLOPS	Floating Point Operations Per Second
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input / Output
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HDD	Hard Disk Drive
HDL	Hardware Description Language
HF	Hochfrequenz
HIL	Hardware-in-the-loop
HLS	High-Level-Synthese
HPS	Hard Processor System
HS	High Speed
I ² C	Inter-Integrated Circuit
IC	Integrated Circuit
IDE	Integrated Development Environment
IIR	Infinite Impulse Response
IO	Input / Output
Kfz	Kraftfahrzeug
LS	Low Speed

LSB	Least Significant Bit
LUT	Lookup Table
NASA	National Aeronautics and Space Administration
NF	Niederfrequenz
NLDAGS	nicht-lineares differential-algebraisches Gleichungssystem mit Schaltbedingungen
OPV	Operationsverstärker
PC	Personal Computer
PCI	Peripheral Component Interconnect
PHIL	Power Hardware-in-the-loop
PMDC	Permanent Magnet Direct Current
PMDCM	Permanent Magnetized Direct Current Motor
PWM	Pulsweitenmodulation
RAM	Random Access Memory
RK	Runge-Kutta
RPi	Raspberry Pi
RPi3	Raspberry Pi 3B+
RT	Realtime
SAR	Successive-approximation-Register
SoC	System on Chip
SPEC	Standard Performance Evaluation Corporation
SPI	Serial Peripheral Interface
SSD	Solid State Drive
TUHH	Technische Universität Hamburg
USB	Universal Serial Bus

1 Einleitung

Bei Hardware-in-the-loop (HIL) Tests wird ein Teil eines Hardwaresystems durch eine Simulation ersetzt. Die Simulation besitzt einen oder mehrere Ein- und Ausgänge, welche mit der restlichen Hardware verbunden sind. Diese können sowohl digital als auch analog sein. Da eine HIL-Simulation immer mit realen Komponenten interagiert, muss sie echtzeitfähig sein. HIL-Tests dienen bei der Entwicklung in der Regel als Zwischenstufe. Viele Komponenten und Systeme werden bei der Entwicklung zunächst vollständig simuliert. Bevor abschließende Tests der Komponenten in einer realen Umgebung durchgeführt werden, können HIL-Tests während der Entwicklung hilfreich sein. Ein großer Vorteil dabei ist, dass die Testparameter schnell und in einem weiten Rahmen angepasst werden können. Somit sind nicht nur Tests unter gewöhnlichen Bedingungen möglich, sondern es können auch Extremfälle simuliert werden. HIL-Simulationen sind besonders nützlich, wenn die Testbedingungen real nur schwer umzusetzen sind oder diese zu einem Schaden oder einer Gefahr führen können.

Die erste einfache Form einer HIL-Simulation ist wahrscheinlich auf Edward A. Link zurückzuführen. Im Jahr 1929 patentierte dieser den ersten elektromechanischen Flugsimulator. Ziel war es, die Instrumente in einem festen Cockpit zu simulieren [15]. Die Piloten konnten so für den ersten realen Flug trainieren. Später wurde ein bewegliches Cockpit ergänzt, welches entsprechend der simulierten Flugzeugbewegung reagierte.

In den 50er bis 60er Jahren kamen die ersten HIL-Simulationen mit digitalen Computern in Flugsimulatoren zum Einsatz [29]. Ab den 60er bis 70er Jahren fanden HIL-Tests verbreitete Anwendung bei Raketentests. Zeitgleich nutzte die National Aeronautics and Space Administration (NASA) HIL-Simulationen zur Entwicklung von Flugzeugtechnologien [11].

Mit der Entwicklung von digitalen Regelungssystemen in Automobilen wurden HIL-Simulationen auch in der Automobilbranche populär [29]. Durch die Elektrifizierung der Automobile in den letzten 10 bis 20 Jahren werden immer mehr und immer komplexere elektronische Steuergeräte (ECUs, Electronic Control Units) verbaut. Die darin enthaltenen Steuer- und Regelungsalgorithmen werden zunehmend umfangreicher.

Im Automobil wird eine Vielzahl elektrischer Kleinmotoren eingesetzt. Beispiele dafür sind Fensterheber, Umwälzpumpen, Lüfter, Sitzverstellungen oder Spiegelverstellungen. Dabei handelt es sich in der Regel um Gleichstrommotoren.

Es werden sowohl permanenterregte mechanisch kommutierte Gleichstrommotoren (PMDCM) als auch bürstenlose elektronisch kommutierte Gleichstrommotoren (BLDCM) verbaut. Oft werden die mechanischen Verbraucher der Motoren über Schneckengetriebe mit einer hohen Übersetzung angetrieben. Dadurch haben die Motoren selber eine Drehzahl in einer Größenordnung von bis zu $20\,000\text{ min}^{-1}$. Teilweise wird die Geschwindigkeit der Motoren über eine Pulsweitenmodulation (PWM) der Versorgungsspannung gesteuert. Die genutzten Frequenzen sind dabei in der Regel in einem Bereich über 20 kHz. Ansonsten können Vibrationen, welche durch die PWM hervorgerufen werden, zu Schall im für das menschliche Gehör wahrnehmbaren Frequenzbereich führen [36]. Die aus diesen Gründen benötigten Zeitschrittweiten für die HIL-Simulation sind auch für moderne Hochleistungs-PCs unter Berücksichtigung der Echtzeitanforderung eine Herausforderung.

Wird beispielsweise die Simulation eines Fensterhebermechanismus betrachtet, spielen sich die physikalischen Vorgänge auf stark unterschiedlichen Zeitskalen ab. Während das Fenster innerhalb mehrerer Sekunden von Anschlag zu Anschlag fährt, befindet sich die Periodendauer der Stromrippel (bedingt durch die Kommutierung) in der Größenordnung von wenigen Millisekunden. Die aus der PWM resultierenden Spannungspulse können bei geringen Pulsbreiten eine Dauer von unter einer Mikrosekunde haben. Aus diesem Verhältnis der Zeiten von etwa 10^6 ergibt sich nicht nur eine hohe Anforderung an die Rechenleistung. Auch Rundungsfehler können bei einer entsprechend kleinen Schrittweite und längerer Simulationsdauer zu Ungenauigkeiten führen. Alternativ kann die Simulation als Co-Simulation mit unterschiedlichen Schrittweiten durchgeführt werden.

Um die Komponenten für die Simulation ausreichend genau modellieren zu können, ist oftmals ein detailliertes Wissen über den Verbraucher nötig. Dies wird in der Regel nicht mit den Herstellerangaben abgedeckt. Für die Modellierung sind deshalb unter Umständen aufwendige Messungen oder andere Arten der Analyse nötig. Das kann auch numerische Rechnungen wie zum Beispiel mit der Finite-Elemente-Methode (FEM) beinhalten.

Analoge Signale in einem System müssen für die digitale Simulation gewandelt werden. Ein Digital-Analog-Wandler (DAC, Digital Analog Converter) hat vergleichsweise geringe zeitliche Verzögerungen. Eine Latenz von einigen Nanosekunden ist ohne großen Aufwand realisierbar [4]. Ein Analog-Digital-Wandler (ADC, Analog Digital Converter) hingegen hat bei den benötigten Schrittweiten der Simulation eine nicht zu vernachlässigende zeitliche Verzögerung in der Größenordnung von einer Mikrosekunde.

In der vorliegenden Arbeit wird die Umsetzung der Simulation mit einer parallelen Architektur untersucht. Dabei werden die hochdynamischen elektrischen Vorgänge separat von den restlichen Vorgängen simuliert. Die langsamere komplexe Simulation wird auf einer Central Processing Unit (CPU) durchgeführt. Simulationsmodelle können dafür relativ umfangreich sein und auch externe Programme oder Bibliotheken nutzen. Die verfügbaren Entwicklungswerkzeuge ermöglichen eine relativ einfache Programmierung und Parametrierung. Die sehr schnellen Vorgänge können durch stark vereinfachte Modelle nachgebildet werden, da diese lediglich einen Einfluss auf die elektrischen Vorgänge haben. Außerdem sind die Anforderung an die Genauigkeit der Simulation für diese schnellen Vorgänge in der Praxis weniger stark zu gewichten, als die für die langsameren Vorgänge. Deshalb kann die schnelle Simulation entweder mit einer analogen Hardware, mit einem Field Programmable Gate Array (FPGA) oder einer anderen schnellen Hardware durchgeführt werden.

2 Stand der Wissenschaft und Technik und Ziel der Arbeit

2.1 Rechenhardware

Für die Echtzeitsimulation physikalischer Prozesse können sich je nach Anforderungen verschiedene Hardwarelösungen eignen. Im Folgenden werden die gängigsten Möglichkeiten dargestellt. Es gibt außerdem kombinierte Lösungen, welche zur Beschleunigung der Rechenleistung beitragen können. Anschließend wird die Entwicklung der Leistungsfähigkeit und die damit einhergehenden Probleme aufgezeigt.

2.1.1 Arten von Prozessoren

Es gibt verschiedene Arten von Prozessoren, welche für die Simulation in Frage kommen. Neben den anschließend dargestellten Arten, gibt es noch digitale Signalprozessoren (DSPs, Digital Signal Processors). Diese haben einen vergleichsweise hohen Entwicklungsaufwand und sind nach der Herstellung nicht flexibel einsetzbar, weshalb sie in dieser Arbeit nicht betrachtet werden. Die Entwicklung der Leistungsfähigkeit von Prozessoren wird exemplarisch anhand von CPUs in Abschnitt 2.1.3 dargestellt.

Mikrocontroller

Ein Mikrocontroller (μC) ist prinzipiell gut für den Einsatz in einer HIL-Umgebung geeignet, wenn die Anforderungen an die Rechenleistung gering sind. Sie sind echtzeitfähig und besitzen diverse Input / Output (IO) Möglichkeiten. Neben universellen Inputs / Output (GPIOs, General Purpose Inputs / Outputs) sind in der Regel auch gängige Datenbus Protokolle implementiert. In vielen Modellen sind außerdem integrierte ADCs und DACs vorhanden. Aktuelle Modelle, zum Beispiel basierend auf einem *ARM[®]Cortex[®]-M7* können bis zu 300 MHz Systemtakt, 144 GPIOs, 24 ADC Kanäle mit insgesamt bis zu 2 MHz Abtastrate und 2 DACs Kanäle mit bis zu 1 MHz Umwandlungsrate haben [26].

Mikroprozessor

Mikroprozessoren (μPs) sind darauf ausgelegt universell einsetzbar zu sein und

sind somit nicht für eine spezielle Aufgabe spezialisiert. Sie sind deutlich leistungsfähiger als μCs . Ein μP benötigt immer diverse Peripherie, um betrieben werden zu können. Schnittstellen wie GPIOs oder Wandler müssen extern realisiert werden und beispielsweise über USB oder PCIe angebunden werden. Es sind Echtzeit (RT, Realtime) Betriebssysteme mit einigen Einschränkungen verfügbar. Nicht-RT-Betriebssysteme können bei geringen Anforderung ausreichend sein, wenn sie mit einem externen Taktgeber arbeiten. Dies hat den Vorteil, dass viele Standardprogramme verfügbar sind. Somit ist die Verwendung eines μPs in einer HIL-Umgebung sinnvoll, wenn ein μC nicht ausreicht und die Anwendungsumgebung gewisse Toleranzen bei den Echtzeitbedingungen zulässt.

FPGA

Ein FPGA ist ein spezieller Typ von integrierten programmierbaren Schaltungen, bei welchem sich Logikelemente, Speicherelemente und andere Komponenten miteinander verschalten lassen. Die Hardware ist somit im Gegensatz zu einem μC oder μP veränderbar. Die Elemente werden je nach Applikation miteinander verschaltet, so dass die zu bewältigende Aufgabe auf Hardwareebene realisiert wird. Somit ist es möglich, so viele Vorgänge wie gewünscht zu parallelisieren und einzelne Operationen zeitlich exakt aufeinander abzustimmen. Viele Modelle besitzen auch dedizierte Einheiten, mit welchen sich Fließkommaoperationen effizient durchführen lassen. Damit ist ein FPGA ausgehend von seinen Qualitäten ideal für den Einsatz in einer HIL-Umgebung geeignet. Allerdings ist der Aufwand der Programmierung (bzw. Erstellung eines Hardware Designs) deutlich größer, als bei einem μC oder CPU. Da alles auf Hardwareebene umgesetzt wird, gibt es teilweise auch starke Begrenzungen was den Umfang des Designs angeht. Auch kann nicht alles, was gewöhnlich programmiert werden kann, synthetisiert werden [45]. Eine gewöhnliche Software benötigt lediglich genügend Speicher, um ein Programm bereitzustellen, ein FPGA hingegen benötigt genügend Logikelemente etc. um das Design zu realisieren.

2.1.2 Rechenbeschleunigung

Neben dem Einsatz einer einzelnen Prozessorart gibt es auch kombinierte Konzepte. Eine übliche Variante ist die Nutzung einer CPU als übergeordnete Recheneinheit und die Auslagerung bestimmter rechenintensiver Subroutinen auf eine andere Hardware.

Graphics Processing Unit (GPU)-Beschleunigung

GPUs sind Prozessoren, welche für die Berechnung von (3D-) Grafiken optimiert sind. Damit einhergehend ist ihre Architektur im Gegensatz zu CPUs auf starke Parallelisierung ausgelegt. Lassen sich Teile eines Problems parallelisieren, können diese von der CPU auf eine GPU ausgelagert werden. Dies wird häufig bei physikalischen Berechnung mit örtlicher Diskretisierung genutzt. Dabei können die Berechnungen beispielsweise für alle Volumenelemente parallel auf der GPU ausgeführt werden [47, 50].

FPGA-Beschleunigung

Die FPGA-Beschleunigung funktioniert analog zur GPU-Beschleunigung. Spezielle Rechnungen werden von der CPU auf den FPGA ausgelagert. Der Unterschied ist dabei, dass die Hardware genau auf die auszulagernde Routine angepasst werden kann und damit noch effizienter arbeitet. Außerdem können die IOs auf den FPGA verlagert werden, wodurch bessere Timings erzielt werden können [31, 39, 56].

2.1.3 Entwicklung der Leistungsfähigkeit

Es ist nicht möglich, eindeutige Aussagen über die Leistungsfähigkeit von Prozessoren zu treffen, da dabei immer Bezug auf bestimmte Eigenschaften des Prozessors genommen werden muss. Wie schnell ist dieser bei Fließkommaberechnungen, Integerberechnungen oder Datendurchsatz? Außerdem spielt auch die Art der zu berechnenden Aufgabe eine große Rolle. Lässt sich die Aufgabe beispielsweise parallelisieren und somit auf mehrere Prozessorkerne verteilen? Um die Leistungsfähigkeit von Prozessoren zu beurteilen gibt es diverse Testsoftware. Diese bilden mit bestimmten Aufgaben eine Punktzahl, um Prozessoren gegeneinander vergleichen zu können. Einige davon sind sehr spezialisiert, andere testen über ein großes Spektrum an Aufgaben. Die Standard Performance Evaluation Corporation (SPEC) ist eine gemeinnützige Organisation, welche standardisierte Benchmark-Suits für verschiedene Hardware bereitstellt. Dazu gehört auch der *SPEC[®]fp* Test, welcher die Geschwindigkeit bei Fließkommaberechnungen misst. Dabei werden verschiedene Tests kombiniert, welche von der Art der Aufgaben hauptsächlich aus dem Physik- und Ingenieursbereich kommen. In Abbildung 1 sind die Ergebnisse dieses Tests für diverse Intel Xeon Serverprozessoren im Zeitraum von 2000 bis 2018 dargestellt. Diese stammen aus den Testergebnissen von dem SPEC2000 und dem SPEC2006 Test. Die Skalierung der Ergebnisse wurde angeglichen und auf die niedrigste Punktzahl

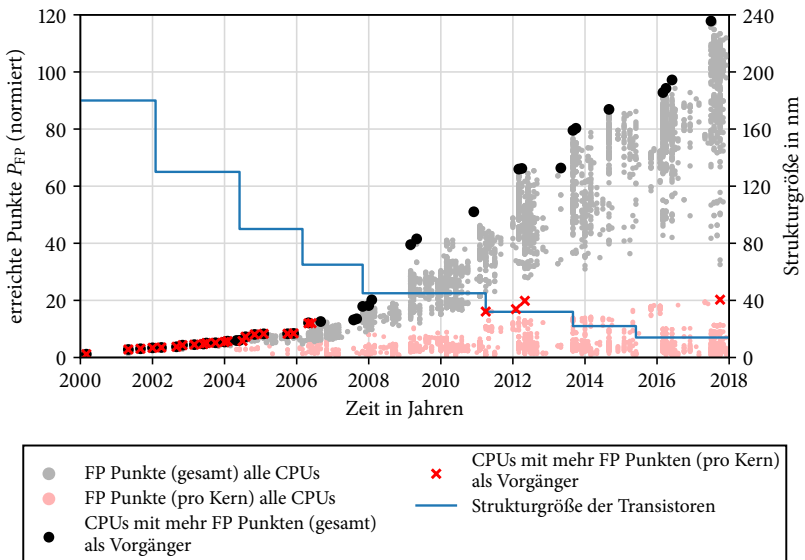


Abbildung 1: Entwicklung der Leistungsfähigkeit von Fließkommaberechnungen der Intel Xeon Prozessoren nach dem SPEC-Test [54] und die zugehörigen Strukturgrößen [27]

normiert¹. In den dargestellten 18 Jahren hat sich die Gesamtleistungsfähigkeit in etwa um den Faktor 12 erhöht. Die auf einen Kern bezogene Rechenleistung ist seit 2006 nur sehr langsam gestiegen. Sie ist teilweise sogar wieder für mehrere Jahre gesunken. Rechner für Probleme, welche sich nicht oder nur begrenzt parallelisieren lassen, haben seitdem somit kaum einen Geschwindigkeitszuwachs erfahren.

2.2 HIL-Simulationen

Aufgrund der weiten Verbreitung von HIL-Simulationen und deren starker Individualität ist es schwierig, eine generelle Aussage über die Leistungsfähigkeit und Qualität aktueller HIL-Simulationen zu treffen. Während mit dem Einsatz von FPGAs theoretisch Schrittweiten von einigen Nanosekunden denkbar sind, hängt die tatsächlich Schrittweite stark von der Applikation ab. Sowohl die

¹Genauere Informationen zu Abbildung 1 sind in Anhang Abschnitt A.1 zu finden

Komplexität als auch die Schnittstellen zum Device Under Test (DUT) spielen dabei eine große Rolle. Müssen zunächst analoge Signale gewandelt werden, ist der ADC oft die beschränkende Komponente. Dabei muss auch berücksichtigt werden, dass oftmals nicht nur die Schrittweite eine wichtige Größe der HIL-Simulation darstellt, sondern auch die Latenz zwischen Ein- und Ausgangssignal, welche je nach Anwendung durchaus mehrere Zeitschritte dauern kann.

Bei Schrittweiten mit einer Größenordnung von 100 μs oder mehr sind übliche Computer von Vorteil. Die Simulationsmodelle können komfortabel mit gängiger Simulationssoftware (beispielsweise *Simulink*) implementiert werden. Dabei sind viele Vorgänge automatisiert und die Entwicklungszeit ist minimal. Bis zu einer Schrittweite mit einer Größenordnung von etwa 10 μs ist es möglich, mit optimierten, spezifisch entwickelten Simulationsprogrammen eine CPU für HIL-Tests zu nutzen [44]. Diese können beispielsweise in der Programmiersprache C/C++ oder *Rust* geschrieben werden. Mit diesen sind die Vorteile höherer Programmiersprachen nutzbar und gleichzeitig ist eine hardwarenahe Programmierung möglich. Somit ist immer noch eine vergleichsweise schnelle Entwicklung möglich. Sollen noch kleinere Schrittweiten erreicht werden, ist dies meistens nur mit dem Einsatz eines FPGAs zu erreichen. Der Große Nachteil ist dabei die aufwendige Programmierung, durch welche die Entwicklungszeit vergleichsweise lang ist. Um diese zu verkürzen gibt es immer mehr Möglichkeiten für eine High-Level-Synthese (HLS). Dabei können die Funktionen des FPGAs in einer Hochsprache wie zum Beispiel C++ beschrieben werden, wobei spezielle Bibliotheken zum Einsatz kommen. Der so erzeugte C++ Programmcode wird anschließend in eine Hardwarebeschreibungssprache (HDL, Hardware Description Language) übersetzt und synthetisiert. Außerdem gibt es vermehrt Entwicklungsumgebungen (IDEs, Integrated Development Environments), welche durch HLS die Entwicklung vereinfachen. Dazu gehört zum Beispiel das Programm *HDL Coder*[™] von *MathWorks*[®]. Mit diesem ist es möglich, aus *Simulink*[®]-Blöcken, *MATLAB*[®]-Funktionen und *Stateflow*[®]-Diagrammen HDL-Code zu generieren. Trotz dieser Möglichkeiten ist die Programmierung einer HIL-Simulation auf einem FPGA deutlich aufwendiger, als auf einem herkömmlichen Prozessor. Gerade für komplexe Modelle kann die Entwicklung eines auf einem FPGA basierenden HIL-Tests unwirtschaftlich sein.

Bei aktuellen Veröffentlichungen von HIL-Simulationen verschiedener Elektromotoren mit einer reinen FPGA-Lösung werden Schrittweiten in der Größenordnung von etwa 1 μs erreicht [2, 3, 30, 40].

Herstellerangaben für erreichbare Schrittweiten mit kommerziellen Lösungen zur HIL-Simulation sind Tabelle 3 zu entnehmen.

Tabelle 3: *Minimal erreichbare Schrittweiten von kommerziell erhältlichen HIL-Simulationen [20, 43, 51]*

Hersteller	Schrittweite CPU in μs	Schrittweite FPGA in μs
dSpace	20	2,5
RTDS Technologies	25...50	1...4
Opal RT	10...100	0,2...2

2.2.1 Heterogene HIL-Co-Simulation

Bei einer heterogenen Co-Simulation werden Berechnungen parallel auf unterschiedlicher Hardware ausgeführt. Verbreitete Varianten sind die in Abschnitt 2.1.2 beschriebene GPU- und FPGA-Beschleunigung.

Es gibt bereits Ansätze, bei welchen eine heterogene HIL-Co-Simulation auf einem Prozessor und einem FPGA ausgeführt wird. Dabei kann diese Kombination sowohl mit separaten Komponenten oder auch als sogenannter System on Chip (SoC)-FPGA ausgeführt sein. Solche Modelle sind von großen FPGA-Herstellern verfügbar. Dabei kommt jeweils eine Kombination aus FPGA und Mikroprozessor zum Einsatz, welche in einem Chip eingebettet sind. Meistens wird der FPGA speziell für das entsprechende Simulationsmodell programmiert. Beispielsweise ist in [53] eine HIL-Simulation beschrieben, in der die elektrischen Vorgänge in den Windungen eines BLDCM in einem FPGA simuliert werden und die restlichen Vorgänge des Motors auf einem Mikroprozessor. Der FPGA arbeitet dabei mit einer deutlich kleineren Schrittweite als der Prozessor. Als Schnittstelle zwischen den parallel arbeitenden Modellen dienen die Werte des Stromes und der in den Wicklungen induzierten Spannung. Diese Anordnung ist in Form eines Blockschaltbildes in Abbildung 2 dargestellt. Obwohl der FPGA die Spannung mit 10 MHz abtastet, liegt die Verzögerung zwischen der Erfassung der Spannung und der analogen Einstellung des simulierten Stromwertes bei etwa 5 μs . Dies resultiert zum einen aus der Pipelineverzögerung des verwendeten ADCs und des FPGAs und zum anderen aus den dynamischen Eigenschaften der analogen Komponenten der Leistungselektronik, welche über die Simulation angesteuert wird und den entsprechenden Strom einstellt.

Bei einem solchen Konzept muss für jedes neue Modell sowohl die Simulation an sich, als auch die Implementierung der Schnittstellen neu vorgenommen werden, da diese je nach Modell verschiedenste Größen übertragen müssen.

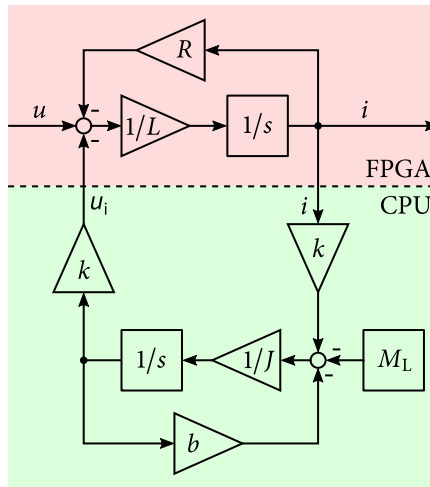


Abbildung 2: Blockschaltbild einer modellspezifischer Implementierung bei einer CPU-FPGA Co-Simulation

2.2.2 Leistungselektronik

Wenn die Ausgangsgröße der HIL-Simulation ein elektrischer Strom ist, wird eine Leistungselektronik benötigt, welche meistens als Last und seltener als Generator arbeitet. Die dynamischen Eigenschaften müssen bei dem Entwurf des Simulationskonzeptes berücksichtigt werden. In Tabelle 4 sind die dynamischen Eigenschaften von einigen kommerziellen elektronischen Lasten für Gleichstrom gelistet. Die Stromanstiegsrate ist für die meisten Verbraucher im Automobil ausreichend wie in Abschnitt 5.1 gezeigt wird. Die Aktualisierungsraten sind hingegen gerade für eine pulsweitenmodulierte Ansteuerung unter Umständen zu gering [32].

Tabelle 4: Maximale Stromanstiegsraten und Aktualisierungsraten von kommerziellen elektronischen Lasten [19, 28, 33, 34, 37]

Hersteller	Stromanstiegsrate in $A \mu s^{-1}$	Aktualisierungsrate in kHz
Keysight Technologies	10	10
Intepro Systems	40	20
MDL Technologies	10	20
Chroma Systems Solutions	8	50
Kikusui America	50	50

2.2.3 Analog-Digital-Wandler

In vielen HIL-Anwendungen werden ADCs benötigt, da das Eingangssignal analog ist. Eine pauschale Aussage über die optimale Auswahl von ADCs kann nicht getroffen werden. Je nach Anwendung gibt es stark abweichende Anforderungen an Auflösung, Abtastrate und Latenz. Für die Anwendung in einer HIL-Simulation ist gerade die Latenz ein entscheidender Faktor. Pipelinewandler haben prinzipbedingt eine Latenz, welche ein Vielfaches der Konversionszeit beträgt [41, 46] und kommen somit in der Regel nicht in Frage. Auch Wandler der Delta-Sigma ($\Delta\Sigma$)-Architektur haben eine relativ hohe Latenz von mehreren Messzyklen. Wenn intern kein Tiefpassfilter nachgeschaltet ist, muss dieser außerdem extern realisiert werden, was zusätzlichen Aufwand bedeutet. Flash Wandler haben in der Regel eine maximale Auflösung von 8 bit und sind daher für viele Anwendungen zu ungenau. Die beste Wahl für HIL-Simulationen sind somit oft ADCs der Successive-approximation-Register (SAR)-Architektur. Diese sind bei einer Auflösung von 14 bit bis 16 bit mit Abtastraten von bis zu 10 MHz bis 15 MHz erhältlich [8, 60] und haben keine Pipelineverzögerung. Bei solchen Wandlern liegt die zeitliche Verzögerung bei minimal 100 ns bis 200 ns. Bei niedrigeren Abtastraten ist auch eine Auflösung bis zu 24 bit realisierbar.

2.3 Ziele und Vorgehen der Arbeit

Ziel dieser Arbeit ist es, ein Konzept für HIL-Tests automobiler Steuergeräte von mechatronischen Komponenten zu erarbeiten. Dieses soll schnelle und reproduzierbare Tests während der Entwicklung der Steuergeräte ermöglichen. Bei typischen Komponenten in diesem Bereich finden relevante physikalische

Vorgänge auf sehr unterschiedlichen Zeitskalen statt. Während der gesamte Vorgang einer Ansteuerung mehrere Sekunden dauern kann, können beispielsweise durch eine PWM verursachte Reaktionen im Bereich von Mikrosekunden stattfinden. Eine Echtzeitsimulation mit diesen geringen Schrittweiten kann nicht mit herkömmlichen Mitteln umgesetzt werden. Es gilt daher zu untersuchen, wie eine HIL-Simulation aufgebaut sein muss, damit sie in der Lage ist, die Vorgänge in allen relevanten Zeitskalen mit ausreichender Genauigkeit abzubilden. Die Steuergeräte geben eine Spannung vor und messen in der Regel den Strom, um daraus Informationen über den Zustand der jeweiligen Komponente abzuleiten. Als Eingangssignal der Simulation dient somit die Spannung. Es soll untersucht werden, wie eine passende Signalverarbeitung für die Spannungserfassung aufzubauen ist. Eine solche HIL-Simulation ist außerdem nur dann sinnvoll, wenn diese mit vertretbarem Aufwand durchführbar ist. Dazu wird untersucht, ob es möglich ist, generische Modelle typischer Verbraucher zu entwickeln, welche mit geringem Aufwand für die jeweilige Anwendung zu parametrieren sind. Aufgrund der hohen zeitlichen Anforderungen und dem Echtzeitkriterium einer HIL-Simulation ist auch die Wahl eines geeigneten numerischen Lösers und der zugehörigen Schrittweite von Bedeutung. Dazu wird untersucht, wie diese Auswahl automatisiert werden kann. Die Funktionsweise des entwickelten Konzeptes soll anhand einer beispielhaften Umsetzung validiert werden.

Um Zeitschrittweiten von unter $1\ \mu\text{s}$ mit einer geringen Latenz bei gleichzeitig einfacher Implementierung zu erreichen, wird eine parallele heterogene Architektur verwendet, bei der physikalische Vorgänge in unterschiedlichen Frequenzbereichen parallel auf unterschiedlicher Hardware simuliert werden.

Dazu muss das Eingangssignal, in diesem Fall die anliegende (PWM-) Spannung, in einen niederfrequenten und einen hochfrequenten Anteil getrennt werden. Dafür ist eine spezielle Filterung erforderlich. Hier stellt sich insbesondere die Frage, wie diese prinzipiell aufzubauen ist und wie die Grenzfrequenz gewählt wird.

Um das Testen zu erleichtern, soll eine Modellbibliothek gängiger Komponenten implementiert sein. Diese müssen für den Anwender leicht zu parametrieren sein. Trotz der vorhandenen Modellbibliothek muss es auch möglich sein neue Modelle mit geringem Aufwand zu implementieren ohne dabei genaue Kenntnisse der Software und dem Aufbau der HIL-Simulation zu haben.

Auch komplexere Modelle wie zum Beispiel ein Fensterhebermechanismus müssen mit geringem Aufwand und mäßigen Kenntnissen über den zu simulierenden Fensterheber parametrierbar sein. Dazu ist es notwendig, dass die

Modelle so wenig Parameter wie möglich aufweisen und diese außerdem ohne großen Aufwand zu ermitteln oder abzuschätzen sind.

Aufgrund von Verbindungen zu anderer Hardware muss vor Beginn der Simulation sichergestellt sein, dass das Modell stabil ist. Dabei stellen sich mehrere Fragen. Haben die Parameter einen Einfluss auf den Löser? Welche Schrittweite kann mit dem jeweiligen Modell auf dem μP realisiert werden und reicht diese aus, damit das Modell stabil simuliert werden kann? Welche Schrittweite ist geeignet in Hinsicht auf den numerischen Fehler und die entstehende Latenz? Reicht die Implementierung eines Löser oder muss je nach Modell aus mehreren implementierten Lösern ein geeigneter ausgewählt werden?

Es sollen Modelle gebildet werden, welche für die Modellbibliothek genutzt werden können. Ein relativ komplexes Beispiel ist die Modellierung eines Fensterhebermechanismus. Dabei soll nicht nur das grundsätzliche Verhalten des Fensterhebers modelliert werden, sondern auch spezielle Betriebszustände, wie zum Beispiel eine festgefrorene Scheibe. Die Parametrisierung und Validierung des Modells soll anhand von Messungen an einem realen Fensterheber stattfinden.

Letztendlich soll gezeigt werden, dass sich das erarbeitete Konzept auch für komplexere Systeme wie dem Fensterheber praktisch umsetzen lässt. Dazu wird ein eigenes Hardwarekonzept realisiert, auf dem die parallele HIL-Simulation durchgeführt wird. Um die Funktion der entwickelten Hardware zu validieren, wird diese mit einer geeigneten Leistungselektronik kombiniert und der simulierte Strom wird mit dem tatsächlichen Strom eines Fensterhebers unter gleichen Bedingungen verglichen.

3 Konzept

Im Folgenden wird ein Konzept erarbeitet, welches geeignet ist, die genannten Ziele zu erreichen. Zunächst wird gezeigt, für welche Art der Anwendung das Konzept bestimmt ist und welche Anforderungen im Detail erfüllt werden sollen. Die HIL-Simulation soll in der Entwicklungsphase von elektronischen Steuergeräten eingesetzt werden. Die Qualität der Simulationsmodelle soll dabei ausreichend sein, um die prinzipielle Funktion von Steuer- und Regelalgorithmen testen zu können, ist aber nicht dafür gedacht, Endkontrollen fertiger Steuergeräte durchzuführen.

3.1 Anforderungen

Die HIL-Simulation soll verschiedene mechatronische Verbraucher aus dem Automobilbereich simulieren können. Fensterheber, Scheibenwischer, Stellantriebe, Lüfter, Pumpen oder Halogenlampen sind beispielsweise typische Anwendungen. Bei den motorischen Verbrauchern kommen in der Regel permanenterregte Gleichstrommotoren (PMDCMs, Permanent Magnetized Direct Current Motors) oder bürstenlose Gleichstrommotoren (BLDCMs, Brushless Direct Current Motors) zum Einsatz. Viele davon werden über Schneckengetriebe mit dem mechanischen System gekoppelt. So können die Motoren mit geringerem Gewicht und geringerem Bauraum realisiert werden. Dadurch bedingt sind die Drehzahlen solcher Motoren verhältnismäßig hoch und können im Bereich von bis zu $20\,000\text{ min}^{-1}$ liegen (Abschnitt 5.1). Bei den betrachteten Verbrauchern stehen die Produktionskosten im Vordergrund. Somit muss beispielsweise bei PMDCMs berücksichtigt werden, dass diese oft nur mit zwei Polen, einer geringen Lamellenanzahl von drei bis zehn und einer geringen Polabdeckung ausgeführt sind. Daraus resultiert unter anderem eine starke Stromschwankung (Stromrippel) aufgrund der Kommutierung. Dies wird genauer in Abschnitt 5.3.5 untersucht.

3.1.1 Schrittweite

Wird zur Orientierung die oben genannte Drehzahl und eine Lamellenzahl von zehn angenommen, ergibt sich somit eine Kommutierungsfrequenz von 1,7 kHz. Um das periodische Signal nach dem Abtasttheorem darstellen zu können, wird eine maximale Schrittweite von $300\text{ }\mu\text{s}$ benötigt [14].

Wird ein Verbraucher mit einer PWM-Spannung betrieben, kommt es auch aufgrund dieser zu Stromrippeln. Sollen auch diese in der Simulation realisiert werden, kommt es zu noch geringeren Schrittweiten. Die Frequenz der PWM befindet sich in der Regel mindestens knapp über dem hörbaren Frequenzspektrum, typisch bei etwa 25 kHz [36]. Soll die Pulsweite in 5 % Schritten aufgelöst werden, wird eine Schrittweite von maximal 1 μ s benötigt, bei 1 % sogar 200 ns (siehe Anhang A.2). Dies ist mit einem herkömmlichen μ P unter Einhaltung des Echtzeitkriteriums nicht zu realisieren.

3.1.2 Genauigkeit

Eine allgemeine Aussage über die benötigte Genauigkeit zu machen ist nicht möglich. Typischer Weise komme es zum einen auf den mittleren Strom der Verbrauchers an, aber auch auf die Amplitude und die zeitlichen Eigenschaften von dynamischen Vorgängen, die abgebildet werden sollen. Voruntersuchungen vergleichbar zu den Messungen in Abschnitt 5.3.5 haben gezeigt, dass eine Genauigkeit von 1 mA in jedem Fall ausreichend ist.

3.1.3 Implementierung von Modellen

Die Implementierung von Modellen soll möglichst rationell und mit minimalen Kenntnissen des HIL-Teststands, der verwendeten Programmiersprachen und von numerischen Lösern möglich sein. Neben gewöhnlichen linearen Differentialgleichungssystemen muss es auch möglich sein, nichtlineare Systeme und Systeme mit Schaltbedingungen zu implementieren. Damit das möglich ist, muss ein einheitliches Schema geschaffen werden, nach dem Gleichungen und Schaltbedingungen abgelegt werden und anschließend automatisiert mit einem geeigneten Löser in Echtzeit berechnet werden (Abschnitt 6).

3.1.4 Parametrierung von Modellen

Mit der HIL-Simulation sollen Steuer- und Regelalgorithmen bei der Entwicklung getestet werden. Es ist für dieses Konzept daher nicht zielführend, wenn detaillierte Simulationsmodelle mit vielen und zum Teil schwer zu ermittelnden Parametern entwickelt werden. Es sollen lediglich die physikalischen Prozesse abgebildet werden, die zum Testen der Steuer- und Regelalgorithmen relevant sind, damit das Modell möglichst einfach zu parametrieren ist. Es muss somit ein Kompromiss zwischen der Anzahl der Parameter, der Ermittelbarkeit der

Parameter und der Einstellbarkeit des Modells gefunden werden. Wenn beim Testen der ECUs zum Beispiel Parametervariationen durchgeführt werden sollen, muss verständlich sein, welche Parameter des Modells welchen Effekt bei der Simulation haben (Abschnitt 5.4).

3.1.5 Bedienung der HIL-Simulation

Die Wahl des zu simulierenden Modells und der zugehörigen Parameter soll über eine grafische Benutzeroberfläche (GUI, Graphical User Interface) realisiert sein, damit die Bedienung schnell und übersichtlich ist. Außerdem muss die Eingabe abgesichert sein. Das heißt, dass dabei sowohl Fehleingaben als auch Werte, welche außerhalb eines sinnvollen Bereiches liegen erkannt werden sollen. Wenn die Stabilität der Simulation mit den gewählten Parametern nicht sichergestellt werden kann, soll dem Nutzer mindestens eine Warnung vor dem Start der Simulation gegeben werden.

3.2 Gesamtkonzept

In Abbildung 3 sind die übergeordneten Komponenten des HIL-Teststands dargestellt. Im Wesentlichen besteht er aus den vier Komponenten

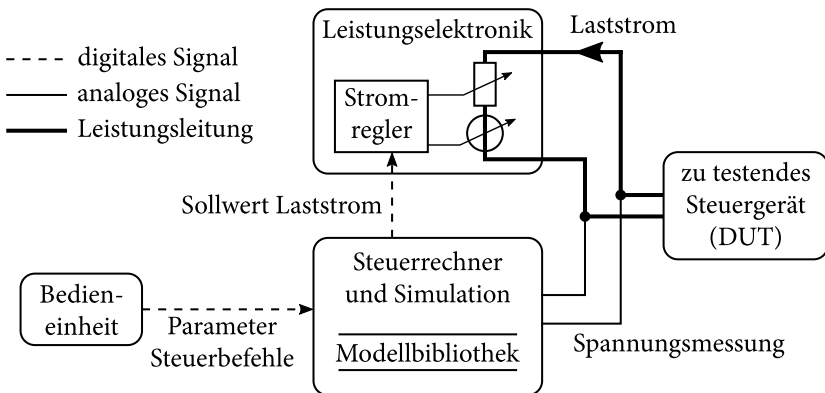


Abbildung 3: Überblick des Aufbaus der HIL-Simulation

- ECU (DUT),
- Simulationshardware / -software,

- Leistungselektronik,
- Bedieneinheit (PC, Notebook, Tablet).

Das DUT ist an eine Leistungselektronik angeschlossen, die in allen vier Quadranten arbeiten kann. Die Simulationshardware ist ebenfalls an das DUT angeschlossen und erfasst die Spannung an den Anschlüssen. Als Ausgangsgröße der Simulation wird der Strom als Vorgabewert an die Leistungselektronik weitergegeben. Für dieses Konzept gilt also $i = f(u, t)$. Dies ist real allerdings nicht immer der Fall. Ein Beispiel ist ein motorischer Verbraucher, der aus einer H-Brückenschaltung mit vier Transistoren jeweils mit einer antiparallelen Freilaufdiode versorgt wird. Sperren die Transistoren, nimmt der Strom entsprechend des ohmsch-induktiven Verhaltens ab und fließt über die Freilaufdioden der Transistoren. Auch wenn der Strom Null erreicht, wird noch Spannung induziert werden, sofern der Rotor des Motors noch dreht. Es gibt zum Beispiel Vorschläge um eine Blockade, beziehungsweise eine eingeschränkte Bewegungsfreiheit eines Schrittmotors mit Hilfe der induzierten Spannung im Steuergerät zu erkennen [42]. Die Spannung an den Klemmen des DUT hängt dann von der Drehzahl ab. Diese Informationen sind in der Simulation enthalten und könnten gegebenenfalls genutzt werden. Für eine entsprechende Simulation müsste also eine Spannung bei einem Strom von Null vorgegeben werden und die Leistungselektronik müsste von einer steuerbaren Stromquelle in eine steuerbare Spannungsquelle umschaltbar sein.

Die HIL-Simulation wird von einem beliebigen Endgerät (PC, Notebook, Tablet) gesteuert. Die Modellbibliothek und die zugehörigen Parametersätze sind zentral auf der Simulationshardware abgelegt und das Endgerät des Benutzers dient lediglich als Eingabegerät.

3.2.1 Mögliche Umsetzungen

Abbildung 4 zeigt als erste Alternative, wie die Spannung $u(t)$ des DUT von einer analogen Filterschaltung in ein hochfrequentes und ein niederfrequentes Signal zerlegt wird. Diese Signale werden jeweils digital gewandelt und auf einem μP , bzw. FPGA simuliert. Die Schrittweite der High Speed (HS)-Simulation hängt von der Leistungsfähigkeit des FPGAs und des ADCs ab. Anschließend werden beide Ergebnisse addiert und ausgegeben. Die Art der Ausführung der Addition ist davon abhängig, ob die Schnittstelle zur Leistungselektronik analog oder digital ausgeführt ist.

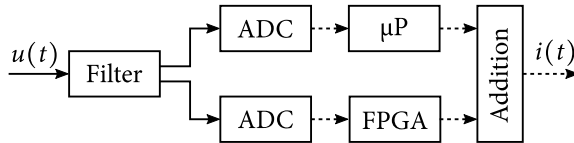


Abbildung 4: Konzept mit analoger Filterung und HS-Simulation auf FPGA

Wie in Abbildung 4 wird auch in dem zweiten Konzept (Abbildung 5) zunächst analog gefiltert. Die HS-Simulation findet hier jedoch schon vor der Digitalwandlung durch eine analoge Schaltung statt. Damit ist die Schrittweite der HS-Simulation hauptsächlich durch die Wandlungsrate des ADCs begrenzt.

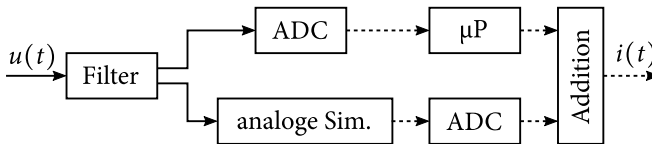


Abbildung 5: Konzept mit analoger Filterung und mit analoger HS-Simulation

Bei der dritten Architektur in Abbildung 6 wird das Eingangssignal direkt digital gewandelt. Der FPGA teilt das Signal anschließend mit einem digitalen Filter auf und überträgt das niederfrequente Signal an die Low Speed (LS)-Simulation auf dem μP . Dieser überträgt die Ergebnisse der Simulation wiederum an den FPGA, welcher das addierte Gesamtergebnis ausgibt.

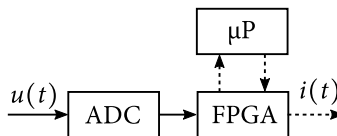


Abbildung 6: Konzept mit Filterung und HS-Simulation auf FPGA

Jedes der gezeigten Konzepte hat Vor- und Nachteile. Es gilt abzuwägen, welche Anforderungen bei der Umsetzung priorität haben. In der aktuellen Anwendung sind geringe Schrittweiten und Latenzzeiten besonders wichtig. Dafür ist die Alternative in Abbildung 5 wegen der analogen Simulation der schnellen Vorgänge die beste.

3.2.2 Detaillierte Architektur

Bei dem Großteil der zu simulierenden Komponenten, die mit PWM angesteuert werden, handelt es sich um motorische Verbraucher. Das grundsätzliche elektrische Verhalten bezüglich der PWM-Frequenz ist somit ohmsch-induktiv und kann mit dem Ersatzschaltbild aus Abbildung 7 dargestellt werden. Die

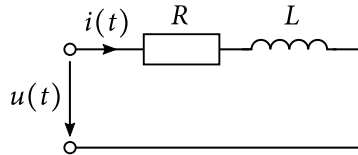


Abbildung 7: R-L-Glied als Ersatzschaltbild für motorische Verbraucher bei hohen Frequenzen

Übertragungsfunktion für den Strom i zu der Spannung u ist

$$G_{RL}(s) = \frac{i(s)}{u(s)} = \frac{1}{R} \cdot \frac{1}{\frac{L}{R} \cdot s + 1} = K \cdot \frac{1}{\tau \cdot s + 1} \quad (1)$$

mit $K = 1/R$ und $\tau = L/R$. In der analogen Schaltung für die HS-Simulation soll das Ergebnis eine Spannung sein, welche proportional zum simulierten Strom ist, da diese von einem ADC für den FPGA gewandelt wird. Ohne die Verstärkung K in Gleichung (1) kann die Übertragungsfunktion mit einem Widerstand und einem Kondensator wie in Abbildung 8 realisiert werden. Da

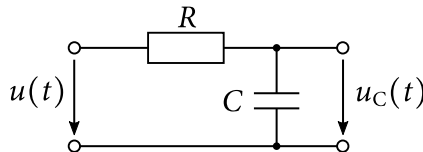


Abbildung 8: RC-Glied

die Werte für die weitere Verarbeitung im FPGA sowieso noch skaliert werden müssen, kann die Multiplikation mit K auch digital in einem Schritt mit der Skalierung vorgenommen werden. Für die Addition der Simulationsergebnisse und die Ausgabe an die Leistungselektronik genügt ein einfacher FPGA, um die geforderte Abtastrate und Latenz zu realisieren. Der FPGA empfängt über eine digitale Schnittstelle das Simulationsergebnis des μ Ps, addiert diese mit den skalierten Werten von i_{hp} und gibt diese über eine digitale Schnittstelle

aus (Abbildung 9). Auf dem μP kann beispielsweise ein Linux-Betriebssystem mit RT-Kernel zum Einsatz kommen. Zur zeitlichen Synchronisation gibt der FPGA ein Taktsignal an den μP aus. Die Filterung ist hier als Blackbox dar-

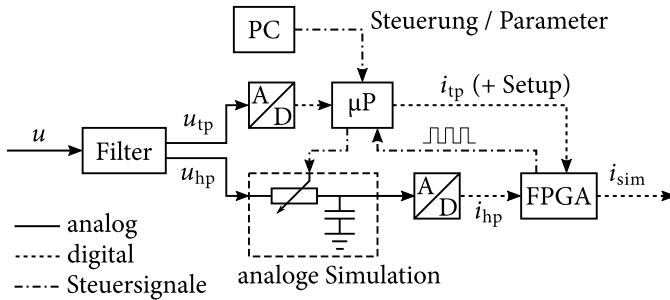


Abbildung 9: Detailliertes Hardwarekonzept

gestellt und wird in Abschnitt 4 detailliert untersucht. Der Widerstand des R-C-Gliedes aus der HS-Simulation kann als digitales Potentiometer ausgeführt werden. Dadurch kann die Zeitkonstante der HS-Simulation entsprechend der gewählten Modellparameter von dem μP eingestellt werden. Die Verstärkung K wird dem FPGA abhängig von den Modellparametern übermittelt. Somit ist es auch mit der analogen Simulation möglich, ohmsch-induktive Verbraucher mit unterschiedlichen Kombinationen von Widerstand und Induktivität abzubilden.

Dieses Konzept setzt somit auf eine getrennte parallele Simulation. Die LS- und HS-Simulation laufen weitgehend unabhängig voneinander ab. Im Gegensatz zu Ansätzen wie sie in der Literatur zu finden sind (Abschnitt 2.2.1), werden bei dieser Architektur die schnellen physikalischen Prozesse nicht vollständig getrennt. Das Modell für die LS-Simulation enthält auch die im HS-Modell enthaltenen Anteile, die aber wegen des vorgeschalteten Tiefpasses ein nur schwaches Eingangssignal enthalten. Signalanteile mit Frequenzen im Übergangsbereich zwischen Tiefpass und Hochpass werden somit bezüglich des Systemverhaltens bei hohen Frequenzen in beiden Simulatoren parallel berechnet und die Ergebnisse werden addiert. Deshalb ist es vorteilhaft, ein Filterpaar zu verwenden, für das die Summe der Übertragungsfunktionen über den gesamten Frequenzbereich möglichst konstant ist. In Abbildung 10 sind das Einschwingverhalten der Filtersignale und die dazugehörigen Ausgangsgrößen der Simulationen beim Einschalten eines Rechtecksignals für ein Filterpaar gemäß Abschnitt 4.2.2 dargestellt. Im stationären Zustand wird das Gleichsignal ausschließlich von der LS-Simulation erzeugt und die durch die PWM

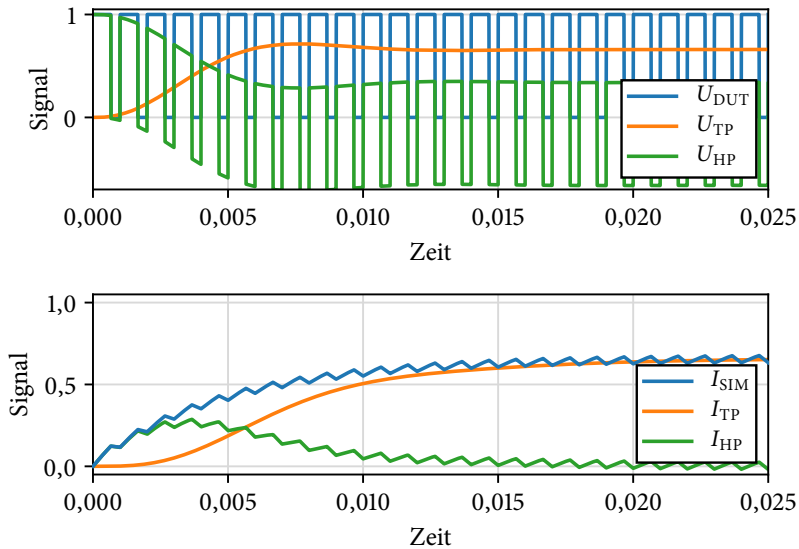


Abbildung 10: *Prinzipieller Signalverlauf bei paralleler Simulation
(Signalnamen entsprechend Abbildung 9)*

erzeugten hochfrequenten Anteile durch die HS-Simulation. Dem gegenüber wird unmittelbar nach dem Einschalten der Einschwingvorgang von beiden Zweigen zum Teil simuliert und die Summe ergibt den zu erwartenden Anstieg entsprechend einem Verzögerungsglied erster Ordnung mit den überlagerten Anteilen mit PWM-Frequenz.

3.3 Simulationsmodelle

Die Modellbibliothek enthält generische Simulationsmodelle gängiger Komponenten, welche jeweils für eine möglichst große Anzahl spezifischer Typen der Komponente parametrierbar ist. Dabei werden physikalische Prozesse, die für die Testanwendung nicht relevant sind, nur vereinfacht oder gar nicht abgebildet. Damit besitzen die Modelle nur wenige Parameter, die aus üblichen Messungen auf einem Prüfstand oder aus Informationen des Datenblattes zu ermitteln oder abzuschätzen sind.

Um dies zu realisieren werden zunächst komplexe Modelle erstellt, welche auf

realen Bauteilen basieren und mit ausführlichen Messungen parametrisiert und validiert werden. Nachdem somit detaillierte Kenntnisse über die Komponente gewonnen wurden, wird das Modell in Hinblick auf mögliche Testszenarien weitgehend vereinfacht. Dieser Vorgang wird exemplarisch in Abschnitt 5.3 für das Modell eines Fensterhebermechanismus dargestellt. Durch die Vereinfachungen entstehen auch oft algebraische Zwangsbedingungen. Daraus resultieren Differential-algebraische Gleichungssysteme (DAEs, differential-algebraic system of equations).

Bei vielen HIL-Tests sollen bestimmte Testfälle simuliert werden. Bei einem Fensterheber kann das beispielweise eine eingefrorene Scheibe oder eine Hand zwischen Scheibe und Rahmen sein. Um solche Testfälle implementieren zu können, muss es möglich sein verschiedene Zustände des Modells und dazugehörige Schaltbedingungen zu definieren. Auch wenn keine Testfälle definiert sind, können Schaltbedingungen auftreten. Zum Beispiel findet ein Übergang zwischen zwei Zuständen statt, wenn die Fensterscheibe aus dem normalen Fahrbetrieb gegen einen Anschlag fährt und blockiert.

Auch nicht lineare Gleichungen können bei den berücksichtigten Komponenten auftreten. Zum Beispiel wird bei einer Halogenlampe der Hauptteil der Leistung durch Wärmestrahlung abgegeben. Die Temperatur als Zustandsvariable geht mit vierter Potenz in die Leistungsbilanz ein.

3.4 Numerische Lösung

Bei HIL-Simulationen gibt es deutlich mehr Einschränkungen für den Differentialgleichungslöser, da dieser immer in Echtzeit arbeiten muss. Das heißt, dass eine bestimmte gewählte Schrittweite nicht überschritten werden darf. Somit muss unter Umständen die Komplexität des Löser niedrig gehalten werden. Auch Löser mit variablen Schrittweiten können in der Regel nicht eingesetzt werden. Unter diesen Bedingungen muss immer sichergestellt werden, dass der Löser das jeweilige Modell mit den eingestellten Parametern stabil lösen kann. Um dies zu garantieren, wird eine automatische Modellanalyse durchgeführt, die vor Simulationstart verschiedene Analysen durchführt und basierend auf den Ergebnissen einen geeigneten Löser wählt. Gegebenenfalls wird die Simulation abgebrochen oder eine Warnung ausgegeben, wenn das Stabilitätskriterium nicht erfüllt werden kann oder die voraussichtliche Genauigkeit gering ist. Bei der Modellanalyse müssen nicht-lineare differential-algebraische Gleichungssysteme mit Schaltbedingungen (NLDAGS) berücksichtigt werden [55]. Damit eine Modellanalyse und eine automatische Wahl und Anwendung eines Löser

realisiert werden kann, wird eine allgemeine Darstellungsform für Modelle mit den zuvor genannten Eigenschaften definiert.

4 Filterung

Für die parallele Simulationsarchitektur muss das Eingangssignal u , also typisch die PWM-Spannung des Steuergerätes, mit einer Frequenzweiche in ein niederfrequentes Signal u_{tp} und ein hochfrequentes Signal u_{hp} aufgeteilt werden. Ein PWM-Signal enthält verschiedene Frequenzanteile. Zum einen die PWM-Frequenz f_{pwm} und deren Harmonischen und die Modulationsfrequenz f_{mod} . In Abbildung 11 werden die erwarteten Frequenzanteile von u dargestellt. Es gibt

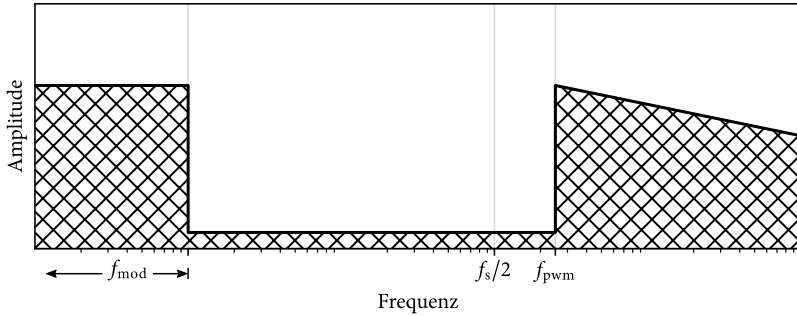


Abbildung 11: Frequenzanteile des Eingangssignals

also einen Frequenzbereich zwischen f_{mod} und f_{pwm} , in welchem nur geringe oder keine Signalamplituden erwartet werden. Dieser Bereich bietet sich für die Grenzfrequenz der Frequenzweiche an.

Tabelle 5: Wichtige Frequenzbereiche für die Auslegung

Bezeichnung	Variable	Frequenzbereich in kHz	Auslegungswert in kHz
Modulationsfrequenz	f_{mod}	$\leq 0,1$	0,1
PWM-Frequenz	f_{pwm}	≥ 25	25
halbe Abtastrate (LS)	$f_s/2$	10... 50	10

In Tabelle 5 sind die für die Auslegung wichtigen Frequenzbereiche aufgelistet. Die maximale Modulationsfrequenz ergibt sich aus der Annahme, dass eine Modulation mit einer Frequenz, welche deutlich größer ist, als die aus den mechanischen Zeitkonstanten der Verbraucher resultierenden Frequenzen nicht sinnvoll ist. Die minimale mechanische Zeitkonstante der simulierten

Verbraucher liegt in einer Größenordnung von 100 ms, wie in Abschnitt 5.1 gezeigt wird. Es werden somit keine Modulationsfrequenzen deutlich größer 10 Hz auftreten, eine Grenze von 100 Hz gewährleistet, dass die Signale inklusive Modulationsfrequenz nahezu unverfälscht übertragen werden. Aufgrund des Abstaththeorems ist auch die halbe Samplefrequenz $f_s/2$ zu berücksichtigen. Diese ergibt sich aus den in Abschnitt 7.2 ermittelten Schrittweiten. Die Auslegungswerte der Frequenzweiche in Tabelle 5 ergeben sich für den kritischen Fall, dass der kleinste Abstand zwischen Modulations- und PWM-Frequenz auftritt und dass die Abtastrate am kleinsten ist.

4.1 Berechnung der Filterparameter

Zunächst erfolgt eine getrennte Analyse der Anforderungen für Hoch- und Tiefpasssignal.

Der Tiefpass muss eine ausreichende Dämpfung für $f \geq f_s/2$ aufweisen, da andernfalls Aliasing-Effekte zu Fehlern in der LS-Simulation führen würden. Ist $f_s/2 < f_{\text{pwm}}$ reicht eine mäßige Dämpfung aus, da in diesem Frequenzbereich nur geringe Amplituden erwartet werden. Für $f \geq f_{\text{pwm}}$ muss eine möglichst hohe Dämpfung vorliegen, da hier hohe Amplituden vorliegen, welche nicht von der LS-Simulation verarbeitet werden können. Aus diesen Anforderungen ergibt sich, dass die Grenzfrequenz f_c des Tiefpassfilters möglichst klein und dessen Ordnung möglichst groß gewählt werden sollte, um eine möglichst hohe Dämpfung zu erreichen. Dem gegenüber stehen weitere Forderungen. Es soll eine möglichst schnelle Sprungantwort von u_{tp} realisiert werden. Außerdem soll die Dämpfung und die Phasenverschiebung für $f \leq f_{\text{mod}}$ möglichst gering sein. Für die geringe Dämpfung reicht eine Grenzfrequenz etwas größer f_{mod} aus. Für die geringe Phasenverschiebung wird eine möglichst große Grenzfrequenz und eine möglichst kleine Ordnung benötigt. Aufgrund der sich widersprechenden Anforderungen muss ein Kompromiss für f_c gefunden werden. Als wichtigste Anforderung wird eine Dämpfung für $f \geq f_{\text{pwm}}$ mit mindestens $A_d = -60$ dB festgelegt.

Der Hochpass soll eine möglichst große Dämpfung für $f \leq f_{\text{mod}}$ aufweisen, da diese Frequenzanteile einen signifikanten Einfluss auf die mechanischen Vorgänge haben und diese in der HS-Simulation nicht modelliert sind. Somit führen Signalanteile mit diesen Frequenzen im Hochpass zu einem Fehler der Gesamtsimulation. Darauf basierend müsste die Grenzfrequenz und die Ordnung möglichst groß gewählt werden. Dem gegenüber stehen die Forderungen nach möglichst geringer Dämpfung und Phasenverschiebung für $f \geq f_{\text{pwm}}$. Für

eine geringe Dämpfung reicht eine Grenzfrequenz etwas kleiner als f_{pwm} aus. Für die geringe Phasenverschiebung muss eine möglichst kleine Grenzfrequenz und Ordnung gewählt werden. Somit muss auch für den Hochpassfilter ein Kompromiss gefunden werden. Als wichtigste Anforderung wird die geringe Dämpfung und Phasenverschiebung für $f \geq f_{pwm}$ festgelegt. Denn kommt es hier zu Fehlern, treten diese während der gesamten Simulationslaufzeit auf, so lange das Eingangssignal in Form einer PWM anliegt und nicht konstant ist. Eine nicht ausreichende Dämpfung der Modulationsfrequenzen wirkt sich nur in einem begrenzten Zeitraum aus, da eine Modulation in der Regel nicht permanent stattfindet.

In Tabelle 6 sind die Anforderungen für den Tiefpassfilter (TP) und für den Hochpassfilter (HP) zusammengefasst. Es ist vorteilhaft, wenn die Addition der

Tabelle 6: Anforderungen an die Filter

Filter	Anforderung	Bedeutung Grenzfrequenz	Bedeutung Ordnung	Relevanz
TP	hohe Dämpfung für $f \geq f_{pwm}$	möglichst klein	möglichst groß	sehr hoch
	hohe Dämpfung für $f \geq f_s/2$	möglichst klein	möglichst groß	hoch
	geringe Phasenverschiebung für $f \leq f_{mod}$	möglichst groß	möglichst klein	mittel
	geringe Dämpfung für $f \leq f_{mod}$	ausreichend größer f_{mod}	-	hoch
	schnelle Sprungantwort	möglichst groß	möglichst klein	hoch
HP	hohe Dämpfung für $f \leq f_{mod}$	möglichst groß	möglichst groß	mittel
	geringe Phasenverschiebung für $f \geq f_{pwm}$	möglichst klein	möglichst klein	sehr hoch
	geringe Dämpfung für $f \geq f_{pwm}$	ausreichend kleiner f_{pwm}	-	hoch

Übertragungsfunktionen von Hoch- und Tiefpass im gesamten Frequenzbereich in etwa eins entspricht und so alle Frequenzen in der Simulation gleichermaßen berücksichtigt werden. Es kann eine geringe Abweichung im Bereich zwischen f_{mod} und f_{pwm} in Kauf genommen werden, da in diesem Bereich keine großen Signalanteile erwartet werden. Trotzdem sollten die Grenzfrequenzen von Hoch- und Tiefpassfilter in etwa gleich sein, um Fehler zu vermeiden. Somit muss ein Kompromiss zwischen den Forderungen an Tief- und Hochpassfilter gefunden werden.

4.2 Architektur Frequenzweiche

Bei dieser Anwendung spielt die Phasenlinearität im Durchlassbereich eine wichtige Rolle, damit sprungförmige Signale, wie die PWM ihre Form beibehalten. Somit sind gängige Filter mit steilem Amplitudenverlauf im Sperrbereich, wie zum Beispiel Filter nach der Chebyshev oder der elliptischen Approximation, weniger geeignet. Es kommen also nur Filterapproximationen wie Butterworth

oder Bessel in Frage, da diese eine gute Phasenlinearität besitzen [18]. Diese haben ab der Grenzfrequenz eine Dämpfung von

$$D = N \cdot d \quad (2)$$

mit der Filterordnung N und $d = 20$ dB/Dekade.

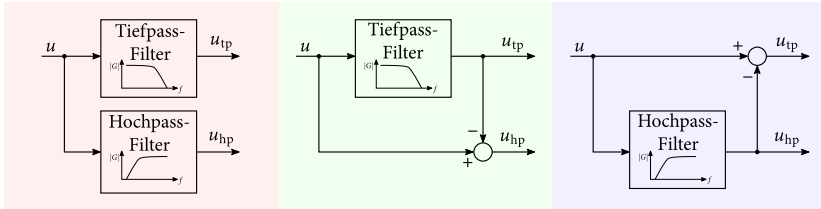


Abbildung 12: Mögliche Filterarchitekturen

Es gibt prinzipiell die drei in Abbildung 12 dargestellten Möglichkeiten die Frequenzweiche zu realisieren. Idealerweise sollte die Bedingung

$$G_{\text{TP}}(s) + G_{\text{HP}}(s) = 1 \quad (3)$$

erfüllt sein. Ist dies der Fall, kann die Frequenzweiche sowohl mit zwei getrennten Filtern, als auch durch Differenzbildung realisiert werden. Bei der Variante mit Differenzbildung wird diese Bedingung immer erfüllt, bei der Variante mit zwei Filtern, muss eine entsprechende Auslegung dafür vorgenommen werden. Die Nutzung von nur einem Filter hat einen geringeren Implementierungsaufwand und wird deshalb bevorzugt.

Bei den in Frage kommenden üblichen Filterapproximationen (Bessel, Butterworth, ...) können die Übertragungsfunktionen für Hoch- und Tiefpass mit der Ordnung $N = n$ mit

$$G_{\text{TP}}(s) = \frac{b_0}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \dots + a_0}, \quad \text{mit } b_0 = a_0 \quad (4)$$

$$G_{\text{HP}}(s) = \frac{b_n \cdot s^n}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \dots + a_0}, \quad \text{mit } b_n = a_n \quad (5)$$

dargestellt werden. Wird der jeweils andere Filter hingegen durch die Differenz wie in Abbildung 12 gebildet, folgt für die Realisierung mit einem Hochpassfilter

$$G_{\text{TP,d}}(s) = 1 - G_{\text{HP}} = \frac{b_{n-1} \cdot s^{n-1} + \dots + a_0}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \dots + b_0} \quad (6)$$

und für die Realisierung mit einem Tiefpassfilter

$$G_{\text{HP,d}}(s) = 1 - G_{\text{TP}} = \frac{b_n \cdot s^n + b_{n-1} \cdot s^{n-1} + \dots + b_1 \cdot s}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \dots + a_0} \quad (7)$$

Die Ordnung des Dämpfungsverhaltens eines Filters aus einer solchen Differenzbildung ist somit immer eins.

Allgemein können Filter, welche ein Verhalten nach Gleichung (2) aufweisen, mit

$$G_{\text{TP}}(s) = \frac{b_m \cdot s^m + b_{m-1} \cdot s^{m-1} + \dots + b_0}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \dots + a_0}, \quad \text{mit } m < n \quad (8)$$

$$G_{\text{HP}}(s) = \frac{b_n \cdot s^n + b_{n-1} \cdot s^{n-1} + \dots + b_k \cdot s^k}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \dots + a_0}, \quad \text{mit } k > 0 \quad (9)$$

beschrieben werden. Die zugehörigen Ordnungen für das Dämpfungsverhalten sind

$$N_{\text{TP}} = n - m \quad (10)$$

$$N_{\text{HP}} = k \quad (11)$$

Für eine Frequenzweiche, bei welcher Gleichung (3) gelten soll, muss

$$k \leq m + 1 \quad (12)$$

erfüllt sein, damit alle im Nenner vorkommende Ordnungen von s auch im Zähler vorhanden sind. Bei gleicher Ordnung in beiden Filterteilen und für eine minimale Anzahl an Pol- und Nullstellen in Bezug auf die Ordnung wird $k = m + 1$ und $m = (n - 1)/2$ für ungerade n festgelegt:

$$G_{\text{TP}}(s) = \frac{b_m \cdot s^m + b_{m-1} \cdot s^{m-1} + \dots + b_0}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \dots + a_0} \quad (13)$$

$$G_{\text{HP}}(s) = \frac{b_n \cdot s^n + b_{n-1} \cdot s^{n-1} + \dots + b_{m+1} \cdot s^{m+1}}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \dots + a_0} \quad (14)$$

Die Filter haben dann jeweils die Ordnung $N = (n + 1)/2$. Damit Gleichung (3) gilt, muss

$$b_i = a_i, \quad \text{für } i = 0 \dots n \quad (15)$$

erfüllt sein. Die Koeffizienten a_i und b_i müssen durch eine numerische Optimierungsrechnung bestimmt werden um das gewünschte Filterverhalten anzunähern, da diese nicht mit den üblichen Approximationen bestimmt werden können.

Die Auslegung und die Implementierung einer solchen Frequenzweiche ist durch die erhöhte Anzahl an Pol- und Nullstellen in Bezug auf die Ordnung und einer nötigen numerischen Optimierung der Parameter im Vergleich zu den anderen Optionen recht umfangreich. Sofern nicht eine größere Flankenteilheit als erste Ordnung unbedingt nötig ist, ist somit ein üblicher Filter vorteilhafter.

4.2.1 Verwendung von zwei Filtern

Bei der Verwendung von zwei Filtern ist es sinnvoll, die Grenzfrequenz auf das geometrische Mittel von f_{mod} und f_{pwm} festzulegen. Der Tiefpassfilter dämpft dann Signale mit f_{pwm} ebenso stark wie der Hochpassfilter Signale mit f_{mod} dämpft. Für die in Tabelle 5 definierten Frequenzen ergibt sich die in Abbil-

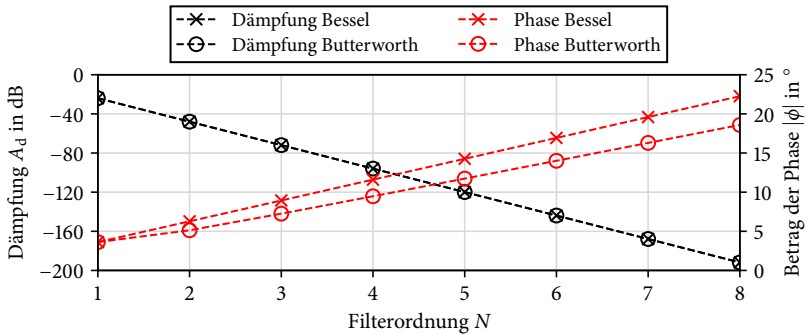


Abbildung 13: Dämpfung im Sperrbereich und Phasenverschiebung im Durchlassbereich des jeweiligen Filters bei f_{mod} und f_{pwm} für $f_{\text{pwm}}/f_{\text{mod}} = 250$

ding 13 dargestellte Abhängigkeit der Dämpfung und Phasenverschiebung von der Ordnung. Mit höherer Ordnung ergibt sich eine bessere Dämpfung, aber auch eine größere Phasenverschiebung. Deshalb ist nur ein Kompromiss zwischen hoher Dämpfung und geringer Phasenverschiebung möglich. Die Dämpfung verhält sich für die Butterworth- und die Besselapproximation annähernd identisch. Die geforderten 60 dB Dämpfung werden ab einer Ordnung

von $N = 3$ erreicht. Die Phasenverschiebung hingegen ist für die Besselapproximation größer, weshalb die Butterworthapproximation zu bevorzugen ist. Aber auch bei dieser beträgt die Phasenverschiebung für $N = 3$ bereits $7,3^\circ$. Dies führt vor allem für das pulsformige Signal des Hochpasses zu einer deutlichen Formänderung, wie in Abbildung 14 dargestellt ist. Außerdem ergibt sich deshalb für $u_{\text{tp}} + u_{\text{hp}}$ das schwingende Verhalten und somit eine deutliche Abweichung zu u beim Einschalten der PWM.

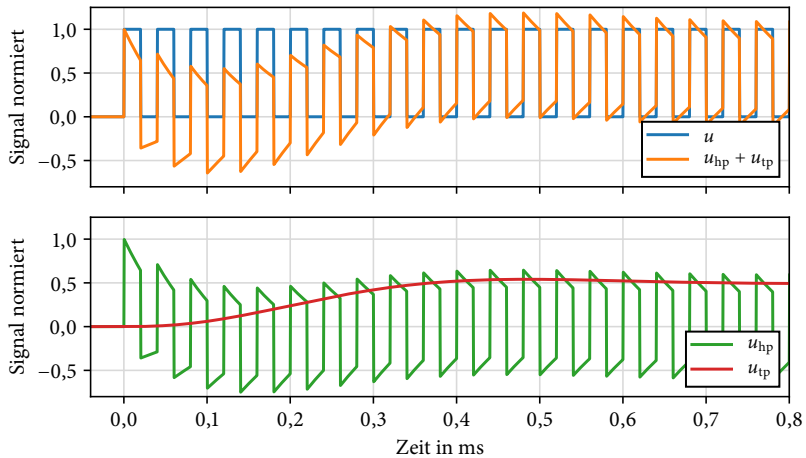


Abbildung 14: Mit Hoch- und Tiefpassfiltern gefilterte Signale einer PWM mit $f_{\text{pwm}} = 25 \text{ kHz}$
(Butterworth Approximation, Ordnung $N = 3$, $f_c = 1581 \text{ Hz}$)

Bei der Verwendung von zwei getrennten Filtern ergibt die Addition der Übertragungsfunktionen beider Filter im Übergangsbereich nur bei speziellen Filtertypen (zum Beispiel Linkwitz-Riley) den Wert eins [35].

4.2.2 Verwendung von einem Filter

Bei der Verwendung von nur einem Filter und einer üblichen und geeigneten Approximation wird das Signal aus der Differenzbildung nur mit einem Verhalten erster Ordnung, also mit 20 dB/Dekade ab der Grenzfrequenz gedämpft. Für die Auslegungswerte hat f_{pwm} weniger als drei Dekaden Abstand zu f_{mod} . Somit ist es nicht möglich für u_{tp} und u_{hp} gleichzeitig eine Dämpfung von mindestens 60 dB zu erreichen. Allerdings ist es mit diesen Varianten möglich, dass u_{tp}

und u_{hp} in den entscheidenden Frequenzbereichen eine Phasenverschiebung von nahezu null aufweisen, wie Abbildung 15 exemplarisch darstellt. Dies ist möglich, da das Signal aus der Differenzbildung eine Phase von annähernd null bis in die Nähe der Grenzfrequenz aufweist. Dies trifft für alle Ordnungen des Filters zu.

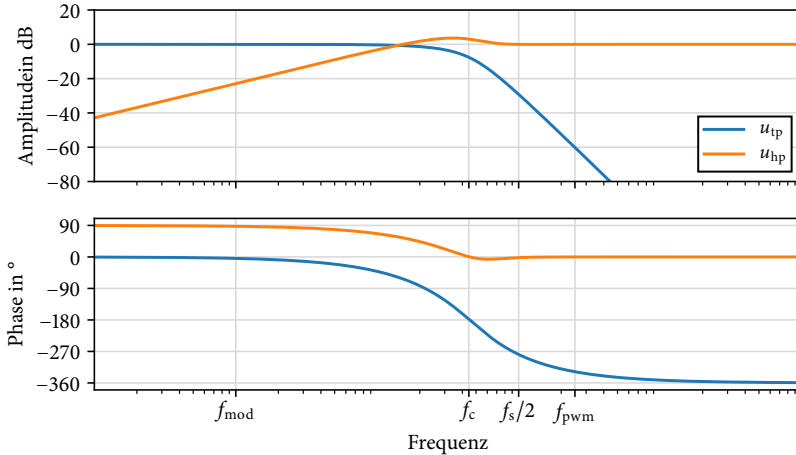


Abbildung 15: Bode Diagramm für eine Filterung mit einem Tiefpassfilter und Differenzbildung
(Bessel: $N = 4$, $f_c = 4446$ Hz)

Hochfrequente Signale können nicht von der LS-Simulation verarbeitet werden und beispielsweise zu Aliasing-Effekten führen. Gelangen niederfrequente Signale in die HS-Simulation, kann dies ebenfalls zu einem Fehler führen, da hier nur ein vereinfachtes Modell simuliert wird. Dieser Effekt ist allerdings weniger kritisch. Somit sollte zumindest die festgelegte Dämpfung von 60 dB für u_{tp} bei f_{pwm} erreicht werden. Die Variante mit einem Hochpassfilter und Differenzbildung kann mit diesem Ansatz ausgeschlossen werden, da für $f_c > f_{mod}$ die Dämpfung für u_{tp} nicht erreicht werden kann.

Für die Variante mit einem Tiefpassfilter und Differenzbildung sollte die Grenzfrequenz so gewählt werden, dass die Dämpfung $A_d = -60$ dB von u_{tp} bei f_{pwm} entspricht. Dies hat zwei Vorteile: Zum einen wird die erforderliche Dämpfung im Hochpass bei f_{mod} erreicht und zum anderen wird eine schnelle Sprungantwort von u_{tp} erreicht. In Abhängigkeit von der Ordnung N ergibt sich die

Grenzfrequenz zu

$$f_c(N) = f_{\text{PWM}} \cdot 10^{A_d/(d \cdot N)}. \quad (16)$$

In Abbildung 16 sind die aus Gleichung (16) berechneten Grenzfrequenzen über der Ordnung aufgetragen. Die größte Erhöhung der Grenzfrequenz erfolgt in etwa bis zu einer Ordnung von $N = 4$. Der erhöhte Implementierungsaufwand

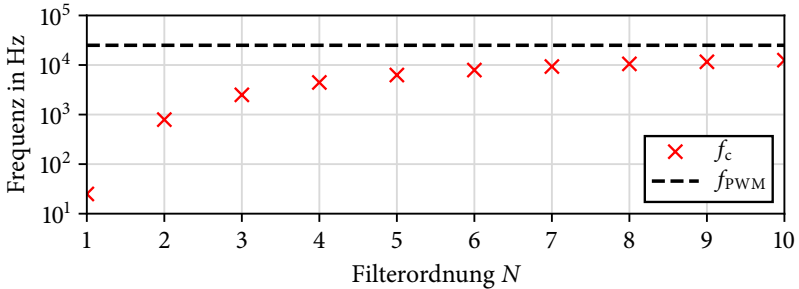


Abbildung 16: Grenzfrequenzen f_c aus Gleichung (16) mit $f_{\text{PWM}} = 25$ kHz, $A_d = -60$ dB und $d = 20$ dB/Dekade.

ist deshalb für eine Filterordnung größer vier nicht mehr gerechtfertigt.

Abbildung 17 zeigt das Zeitverhalten von u_{tp} und u_{hp} beim Einschalten eines PWM-Signals mit einem Filter vierter Ordnung und der aus Gleichung (16) berechneten Grenzfrequenz von $f_c = 4446$ Hz. Im Gegensatz zu dem Verhalten mit zwei Filtern in Abbildung 14 ist ersichtlich, dass die Summe von u_{tp} und u_{hp} gleich u ist und somit der Fehler der Gesamtsimulation verringert wird. Außerdem benötigt u_{tp} nach dem Einschalten nur etwa die halbe Zeit bis zum Erreichen des eingeschwungenen Wertes im Vergleich zu zwei Filtern aufgrund der höheren Grenzfrequenz. Dies verbessert die Dynamik und somit die Qualität der LS-Simulation. Das dazugehörige Bode-Diagramm der Filterung ist in Abbildung 15 dargestellt. In diesem Fall ist es auch möglich die Besselapproximation zu nutzen, da auch mit dieser aufgrund der höheren Grenzfrequenz eine Phasenverschiebung von u_{tp} von etwa null bei f_{mod} erreicht werden kann. Durch den linearen Phasengang und der daraus resultierenden weitgehend konstanten Gruppenlaufzeit im Durchlassbereich, werden Signalformen mit der Besselapproximation am besten erhalten [52].

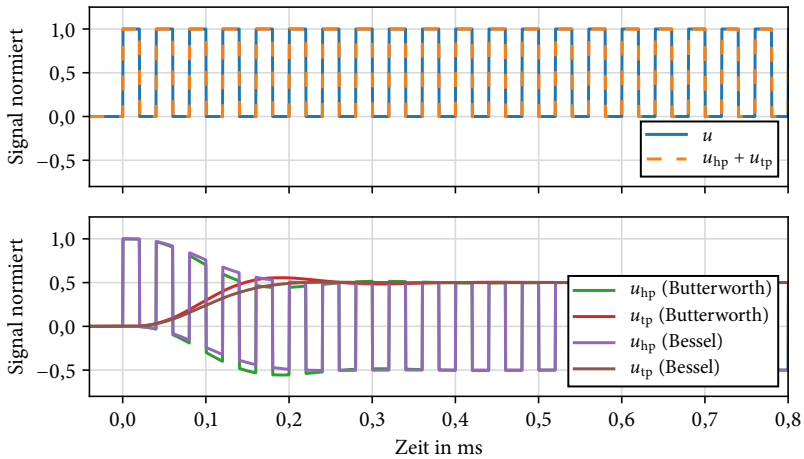


Abbildung 17: Gefiltertes PWM-Signal mit $f_{\text{pwm}} = 25 \text{ kHz}$. u_{tp} und u_{hp} aus Tiefpassfilter und Differenzbildung. (Bessel Approximation, Ordnung $N = 4$, $f_c = 4446 \text{ Hz}$)

4.3 Ergebnisse

Für den allgemeinen Fall einer solchen Anwendung kann der Einsatz zweier getrennter Filter sinnvoll sein, wenn die zu trennenden Frequenzbereiche weit genug auseinander liegen und eine Dämpfung und Phasenverschiebung für die relevanten Frequenzen von etwa null erreicht werden kann.

Der beste Kompromiss für diese spezielle Anwendung ist die Verwendung einer Kombination aus Tiefpassfilter und Differenzbildung für das Hochpasssignal. Eine sinnvolle Wahl der Ordnung für den Tiefpassfilter ist $N = 4$. Die Grenzfrequenz ist nach Gleichung (16) $f_c = 4446 \text{ Hz}$, so dass gerade die minimal geforderte Dämpfung für die hochfrequenten Signale der PWM im Tiefpass erreicht wird.

Der genaue Verlauf der Pulse ist in Abbildung 17 zu sehen. In Abbildung 18 ist der Verlauf von u_{tp} im Vergleich zu der eingestellten Pulsweite einer PWM dargestellt. In der normierten Darstellung entspricht die Pulsweite dem Verlauf für ein ideal gefiltertes Tiefpasssignal. Oben ist eine Modulation mit der maximalen Modulationsfrequenz von 100 Hz dargestellt. Unten wird die Pulsweite in 5 ms linear von null bis eins gesteigert. Es ist zu erkennen, dass die Abweichung von u_{tp} zu der Pulsweite in beiden Fällen sehr gering ist. Für die sinus-förmige Mo-

dulation beträgt diese maximal 3,7 % und für die lineare Modulation maximal 2,4 %.

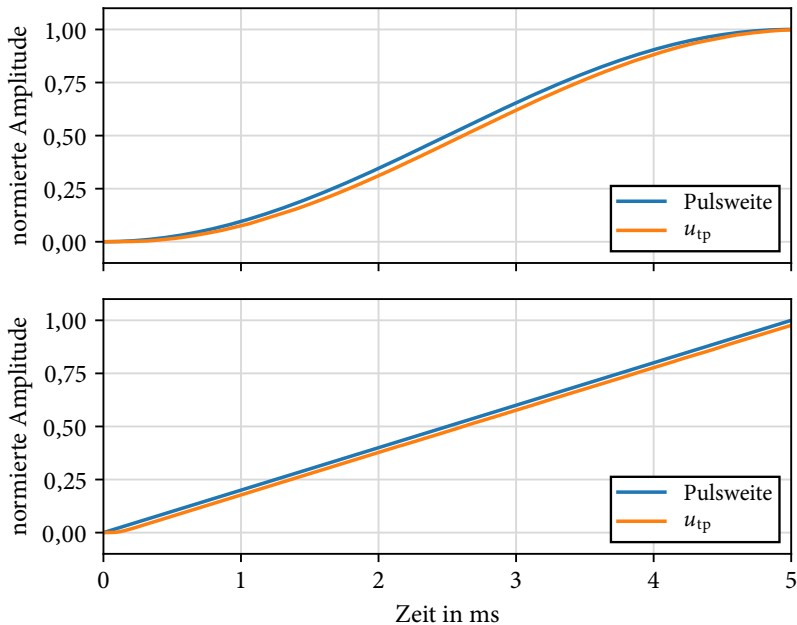


Abbildung 18: Ausgang des Tiefpassfilters u_{tp} für ein PWM-Signal mit $f_{pwm} = 25$ kHz im Vergleich mit der Pulsweite. Oben: Sinusförmige Modulation der Pulsweite mit $f_{mod} = 100$ Hz. Unten: Rampenförmiger Anstieg der Pulsweite. (Filter: Bessel Approximation, Ordnung $N = 4$, $f_c = 4446$ Hz)

Die Auswirkung des Filters auf die Simulation eines R-L-Gliedes ist in Abbildung 19 dargestellt. Bei großen elektrischen Zeitkonstanten τ_{max} des ohmsch-induktiven Verhaltens (rechts) ist nahezu keine Abweichung zu erkennen. Bei einer kleinen elektrischen Zeitkonstante τ_{min} (links) gibt es deutliche Abweichungen zwischen dem simulierten Strom mit und ohne Filter. Die betrachteten Größenordnungen für die Zeitkonstante τ ergeben sich aus Messungen an realen Komponenten. Die schraffierte Fläche ist proportional zur elektrischen Energie, welche im Verhältnis zu einer idealen Simulation weniger aufgenommen wird. Bei der Simulation eines reinen R-L-Gliedes wird der entsprechende Teil in der HS-Simulation abgebildet und das Gesamtergebnis der parallelen Simulation ist korrekt. Wird aber ein Verbraucher simuliert, bei welchem der Strom in

weiteren Gleichungen genutzt wird, zum Beispiel für mechanische Vorgänge, kommt es zu einem Fehler. Beispielsweise wird einem Motor beim Anlauf so wenig Energie zugeführt, was das Anlaufverhalten verzögert. Das gefilterte Signal erreicht bei τ_{\min} in etwa 0,2 ms später die volle Amplitude. Die Anlaufzeit von den betrachteten Motoren (Abschnitt 5.1) ist allerdings um einen Faktor von 100 bis 1000 größer. Dieser Effekt hat somit keinen signifikanten Einfluss auf das Simulationsergebnis.

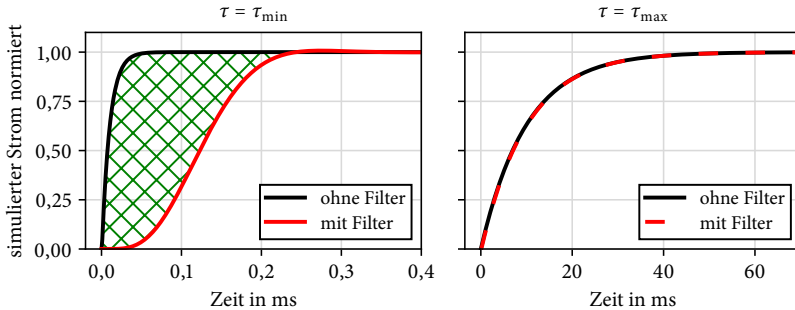


Abbildung 19: Simulierter Strom eines R-L-Gliedes in der LS-Simulation bei einem normierten Spannungssprung mit und ohne Tiefpassfilter und unterschiedlichen Zeitkonstanten $\tau = L/R$ ($\tau_{\min} = 10 \mu\text{s}$, $\tau_{\max} = 10 \text{ms}$)

4.4 Digitale Filterung

In dem Konzept wie in Abbildung 6 erfolgt die Filterung digital. Bei digitalen Filtern gibt es grundsätzlich die Unterscheidung zwischen Finite Impulse Response (FIR) und Infinite Impulse Response (IIR) Filtern. Für die Auslegung von digitalen IIR Filtern werden in der Regel analoge Filterapproximationen genutzt und mit einer z -Transformation in den diskreten Bereich überführt. Prinzipiell kann die Auslegung eines digitalen Filters auf der beschriebenen analogen Auslegung basieren. Die meistens verwendete Bilineartransformation kann die speziellen Eigenschaften des analogen Besselfilters aufgrund der nicht linearen Kompression nicht abbilden. Alternativ kann aber die impulsinvariante Transformation genutzt werden [52].

FIR-Filter haben die besondere Eigenschaft, dass sie im Gegensatz zu (analogen) IIR Filtern einen durchgehend linearen Phasengang und somit eine konstante Gruppenlaufzeit haben können. Dadurch wird die Form der Eingangssignale

bestmöglich erhalten. Außerdem sind FIR-Filter immer stabil und reagieren auf Rundungen der Koeffizienten toleranter als IIR-Filter [38].

Bei der vorliegenden Anwendung soll die HS-Simulation eine maximale Schrittweite von $1 \mu\text{s}$ aufweisen. Im Gegensatz zur analogen Ausführung gibt es nach dem Konzept in Abbildung 6 nur noch einen ADC. Das heißt, dass die Abtastrate des Filters in dieser Konfiguration mindestens 1 MHz entspricht. Dies führt zu einem sehr kleinen Verhältnis von der Grenzfrequenz zur Abtastfrequenz, was signifikante Nachteile hat. Die Filterkoeffizienten werden sehr klein, was zu größeren Rundungsfehlern in den numerischen Berechnungen führen kann. Mit einer minimalen Abtastfrequenz von $f_{s,\min} = 1,0 \text{ MHz}$ und einer maximalen Grenzfrequenz $f_{c,\max} = 5 \text{ kHz}$ ist die normierte Grenzfrequenz maximal

$$f_{c_n,\max} = \frac{f_{c,\max}}{f_{s,\min}/2} = 0,01. \quad (17)$$

Bei der tatsächlichen Umsetzung des Konzeptes in Abschnitt 7 wird eine Abtastfrequenz von 2,5 MHz genutzt. Die normierte Grenzfrequenz würde für diesen Fall sogar nur bei $f_{c_n} = 0,004$ liegen. Um die vorgegebene Dämpfung einzuhalten, werden so sehr hohe Filterordnungen benötigt. Während bei einem IIR-Filter, basierend auf der analogen Auslegung, schon eine Filterordnung von $N = 3 \dots 4$ ausreichen kann um die geforderte Dämpfung zu erreichen, benötigt ein geeigneter FIR-Filter bei gleichen Anforderungen eine Ordnung zwischen etwa $N = 100 \dots 1000$. Dies ist mit einem sehr hohen Rechenaufwand, bzw. Platzbedarf auf einem FPGA verbunden und kann zu hohen Rundungsfehlern führen.

Die Nachteile eines FIR-Filters bei dieser speziellen Anwendung überwiegen. Deshalb sollte ein digitaler IIR-Filter mit der impulsinvarianten z -Transformation, basierend auf der analogen Auslegung der Abschnitte 4.2.1 und 4.2.2, genutzt werden. Alternativ kann auch die einfachere Bilineartransformation genutzt werden, wenn keine Besselapproximation zum Einsatz kommt und kleine Abstriche bei der Qualität der Phasenlinearität in Kauf genommen werden.

5 Modellbildung

In diesem Kapitel wird anhand von Messungen analysiert, was für Arten von Modellen benötigt werden und was für Anforderungen an diese Modelle gestellt werden. Anschließend wird der Modellbildungsprozess am Beispiel eines Fensterhebers demonstriert. Dabei wird das Modell durch Messungen an realen Bauteilen parametrisiert und validiert.

5.1 Messungen an repräsentativen Verbrauchern

Im Folgenden sind Messungen an den im Kraftfahrzeug (Kfz)-Bereich vorkommenden sehr unterschiedlichen Arten von elektrischen, beziehungsweise mechatronischen Verbrauchern aus einem Kfz dargestellt. Alle Messungen wurden mit einer konstanten Spannung von 12 V mit einem Labornetzteil durchgeführt.

In Abbildung 20 ist der Stromverlauf eines PMDCM mit mechanischer Kommutierung dargestellt, der eine Waschwasserpumpe antreibt. Oben ist das typische Verhalten eines Gleichstrommotors mit annähernd konstanter mechanischer Last zu beobachten. Der Ausschnitt unten zeigt die Stromschwankungen aufgrund der mechanischen Kommutierung. Diese liegen im Bereich von etwa 0,5 A bis 1 A bei einem mittleren Strom von etwa 5,5 A. Die Amplitude der Kommutierungsrippel beträgt etwa 9 % bis 18 % des mittleren Stromes. Die Kommutierungsfrequenz liegt bei 3165 Hz. Der Kommutator besteht aus 10 Lamellen, daraus ergibt sich eine Drehzahl im stationären Betrieb von $18\,987\text{ min}^{-1}$. Die maximale Stromänderungsrate liegt beim Einschalten vor und beträgt 166 A ms^{-1} .

In Abbildung 21 ist der Stromverlauf einer Pumpe (Bosch PAD 12V) dargestellt. Diese ist elektronisch kommutiert und besitzt einen permanent erregten Rotor, auf welchem das Pumpenrad angebracht ist. Sie wird beispielsweise in Kühlmittelkreisen eingesetzt. Der Stator des Motors ist mit zwei Strängen mit jeweils zwei Nuten ausgeführt, der Rotor besitzt sechs Pole. Somit finden pro Umdrehung zwölf Kommutierungen statt. Für den stationären Betrieb ist eine Kommutierungsfrequenz von 499 Hz abzulesen. Daraus resultiert eine Drehzahl von 2495 min^{-1} . Der Strom kann beim Abschalten eines Strangs negativ werden und fast die gleiche negative Amplitude erreichen wie im Positiven. An dieser Stelle ist ein sehr steiler Nulldurchgang durch interne Schaltvorgänge zu beobachten. Bei der Simulation eines entsprechenden Modells muss damit die

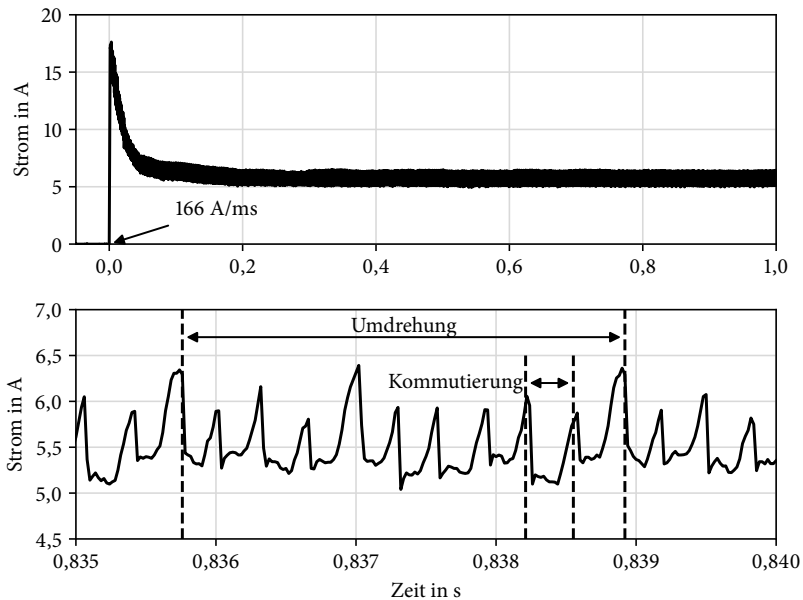


Abbildung 20: Oben: Stromverlauf nach Einschalten und im stationären Betrieb einer Waschwasserpumpe aus einem Kfz; Unten: Detailausschnitt über eine Umdrehung

hohe Steifheit des Systems berücksichtigt werden. Beim Anlaufen des Motors, in Abbildung 21 unten links dargestellt, sind allerdings recht unregelmäßige Schaltvorgänge zu sehen. Dieses Verhalten hat keine Relevanz für Tests und muss somit nicht modelliert werden.

In Abbildung 22 ist der Einschaltvorgang einer Halogenlampe dargestellt. In dem Moment des Einschaltens wird eine hohe Stromänderungsrate von bis zu 314 A ms^{-1} erreicht. Der Verlauf des Stromes wird beim Anstieg des Stromes hauptsächlich von dem elektrischen Widerstand und der Induktivität im kalten Zustand bestimmt und erreicht eine maximalen Strom von ca. 26 A. Dies ist die größte Stromänderungsrate und der größte Maximalstrom aller vermessenen Komponenten. Der anschließende Abfall resultiert aus der Erwärmung des Drahtes auf etwa 2700 °C und der damit verbundenen temperaturabhängigen Erhöhung des elektrischen Widerstandes. Die Wärmeabgabe der Lampe hängt hauptsächlich von der Strahlung ab [13, 21]. Die Temperatur geht somit mit vierter Potenz in den Wärmestrom ein und damit auch in die Berechnung des

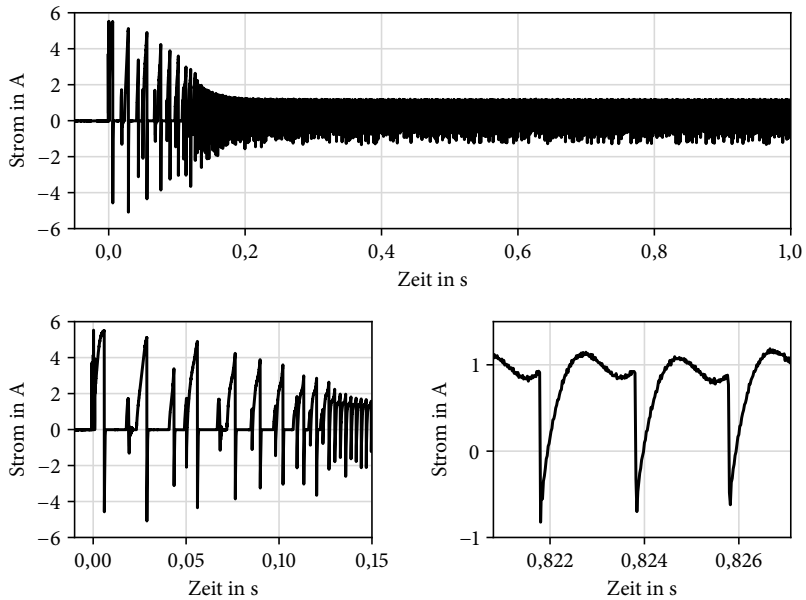


Abbildung 21: Ausschnitt des Stromverlaufs im stationären Betrieb einer Pumpe mit einem BLDCM

Widerstandes [21] (Gleichung (91)).

In Abbildung 23 oben ist der Stromverlauf bei dem gesamten Vorgang des Herunterfahrens eines Fensterhebers dargestellt. An dieser Stelle soll nur ein kurzer Überblick über die Messung des Fensterhebers gegeben werden. Dieser wird im Detail in den nachfolgenden Abschnitten behandelt. Der Fensterhebermechanismus beruht in diesem Fall auf einem Seilzugsystem. Der Antrieb ist ein mechanisch kommutierter Gleichstrommotor. Anhand des Stromverlaufs, kann der Vorgang grob in drei Abschnitte unterteilt werden. In rot ist das Anfahren inklusive Lose im Seil und ansteigender Last markiert. Im grünen Bereich fällt der Strom annähernd linear mit der Zeit und damit auch mit dem zurückgelegten Weg ab. Dies kann durch den abnehmenden seitlichen Eingriff der Fensterscheibe in die Gummidichtungen erklärt werden. Am Ende im blauen Bereich fährt die Fensterscheibe gegen einen Anschlag und blockiert. In Abbildung 23 unten ist der Kommutierungsvorgang dargestellt. Die Amplitude der Stromrippel durch die Kommutierung beträgt etwa 12 % bis 15 %. Im Gegensatz zu dem Motor der Waschwasserpumpe aus Abbildung 20 sind die

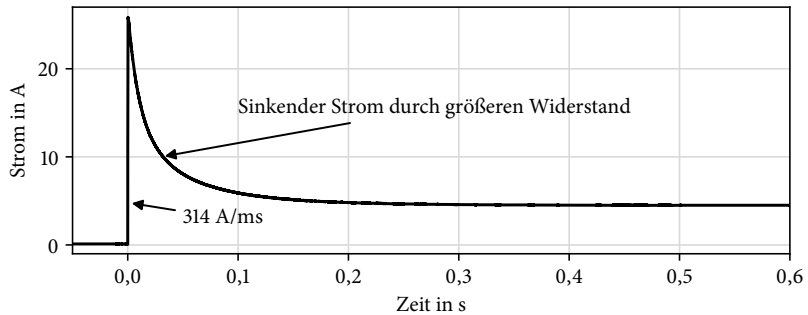


Abbildung 22: *Stromverlauf beim Anschalten einer Halogenlampe*

Stromrippel deutlich abgerundeter. Das Modell eines Fensterhebers muss nicht nur die Kommutierung im Bereich von Millisekunden abbilden, sondern auch unterschiedlich Zustände der Mechanik des Fensterhebermechanismus. Somit müssen Schaltbedingungen in dem Modell abgebildet werden.

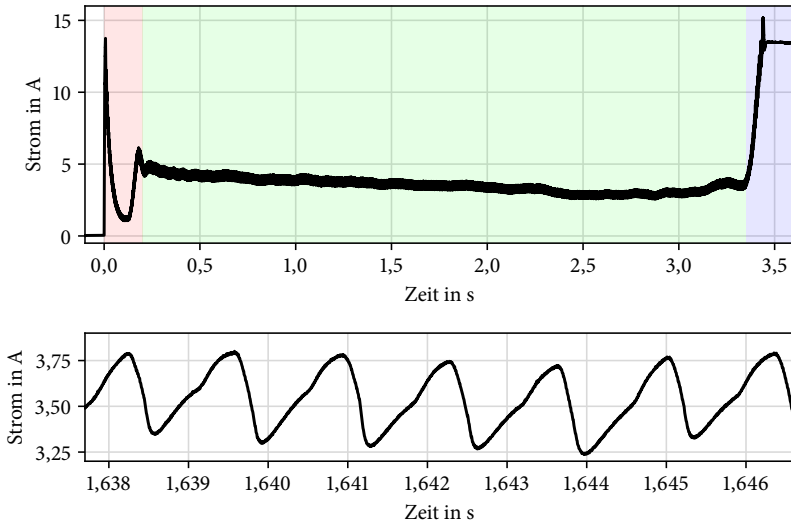


Abbildung 23: *Oben: Stromverlauf beim Herunterfahren eines Fensterhebers mit PMDCM; Unten: Ausschnitt der Stromrippel durch Kommutierung*

5.2 Anforderungen an die Modelle

Die Modelle sollen nur die physikalischen Vorgänge abbilden, welche auch relevant bei Tests der Steuergeräte sind. Dieses hängt im Wesentlichen von den Steuer- und Regelalgorithmen der jeweiligen ECU ab. Für das Steuergerät eines Fensterhebers ist es natürlich unumgänglich, Effekte wie die Anschläge oder die Selbsthemmung des Getriebes zu simulieren. Bei anderen Effekten, wie zum Beispiel der positionsabhängigen Gummireibung der Dichtung ist zu klären, ob diese berücksichtigt werden müssen. Andere Effekte wie zum Beispiel die Erwärmung des Motors sind in den meisten Fällen zu vernachlässigen, da ein Fensterheber in der Regel nur wenige Sekunden am Stück betrieben wird.

Die Kommutierung wird von einigen ECUs zur Positionserkennung genutzt. Daher ist eine Abbildung in der Simulation erforderlich.

5.3 Modellierung und Parametrierung des Fensterhebers

In diesem Abschnitt wird exemplarisch die Modellherleitung für einen Fensterheber durchgeführt. Dabei wird zunächst ein Grundmodell aufgebaut, welches anschließend im Detail ausgearbeitet wird. Die daraus resultierenden Modellparameter sind aus Messungen, analytischen Berechnungen oder Schätzungen bestimmbar. Schätzungen sollten nur genutzt werden, wenn Messungen oder eine analytische Bestimmung sehr aufwendig im Verhältnis zum Einfluss des Parameters auf das Simulationsergebnis sind. Für die Parameterbestimmung in Abschnitt 5.4 werden die in Tabelle 7 gekennzeichneten Verfahren beschrieben.

Tabelle 7: Genutzte Methoden zur Bestimmung der Modellparameter

Parameter / Teilmodell	Analytische		
	Messung	Berechnung	Schätzung
Elektrischer Widerstand	×	-	×
Induktivität	×	-	-
Motorkonstante	×	-	-
Mechanik Motor	-	×	-
Getriebe	-	-	×
Seilzugkinematik	-	×	×
Reibkräfte	-	×	-
Seillose	×	-	-
Kommutierung	-	-	×
Vereisung	-	×	-

Die in diesem Abschnitt genutzten Messungen wurden an einer vorhandenen Autotür einer Mercedes C Klasse durchgeführt.

5.3.1 Motor (PMDCM)

Bei diesen Motoren herrscht ein hoher Kostendruck, deshalb haben sie zum Beispiel eine geringe Anzahl an Wicklungen, bzw. Lamellen am Kommutator. Auch gibt es wenige Maßnahmen um dem Ankerquerfeld entgegenzuwirken. Dadurch entsteht ein sehr ungleichmäßiger Stromfluss, auch bei annähernd konstantem Lastmoment und konstanter Drehzahl. Die Motoren sind aus Platzgründen in der Regel klein und mit hoher Drehzahl ausgeführt und haben ein

Schnecken- oder Planetengetriebe mit einer Untersetzung von etwa 30 bis 400 [17]. Da die angetriebenen mechanischen Systeme im Gegensatz zum Motor selbst verhältnismäßig träge sind, haben die Stromschwankungen einen sehr geringen Einfluss auf die Bewegungen des mechanischen Systems.

Das Ersatzschaltbild für den dynamischen Betrieb eines idealen Gleichstrommotors wird in der Literatur in der Regel wie in Abbildung 24 dargestellt [22] und wird durch die Gleichung

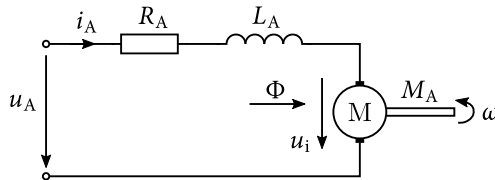


Abbildung 24: Ersatzschaltbild eines permanentenerregten Gleichstrommotors

$$u_A = R_A i_A + L_A \frac{di_A}{dt} + u_i \quad (18)$$

beschrieben, mit der Ankerspannung u_A , dem Ankerstrom i_A , dem Ankerwiderstand R_A , der Ankerinduktivität L_A und der induzierten Spannung

$$u_i = k_2 \Phi \omega. \quad (19)$$

Das erzeugte mechanische Moment am Rotor ist

$$M_A = k_2 \Phi i_A. \quad (20)$$

Die Parameter können unter idealen Umständen berechnet werden, wenn genügend Daten vorhanden sind [16]:

$$k_2 = \frac{2KN_c p}{2\pi a} \quad (21)$$

$$\Phi = \alpha_e \tau_p l_e B_{\delta, m} \quad (22)$$

Diese beiden Größen hängen somit von vielen Parametern ab (Tabelle 8), welche zum Beispiel durch Geometrie oder Art der Wicklung bestimmt werden können. Diese Parameter sind oft nicht ohne Weiteres bekannt oder zu bestimmen. Außerdem stellt diese Beschreibung lediglich eine Näherung dar. Für die in diesem Modell nötigen Berechnungen wird nur das Produkt

$$k_M = k_2 \Phi \quad (23)$$

Tabelle 8: Parameter zur Bestimmung von k_2 und Φ

Parameter	Beschreibung
K	Anzahl der Kommutatorsegmente
N_c	Spulenwindungszahl
p	Polpaarzahl
a	Halbe Anzahl paralleler Wicklungszweige
α_e	Ideelle (äquivalente) Polbedeckung
τ_p	Polteilung
l_e	Ideelle Eisenlänge
$B_{\delta,m}$	Konstantes maximales Luftspaltfeld

benötigt, welches deutlich einfacher zu bestimmen ist. Somit ergibt sich zusammen mit Gleichungen (19) und (20)

$$u_i = k_M \cdot \omega \quad (24)$$

und

$$M_A = k_M \cdot i_A . \quad (25)$$

Das Motormoment setzt sich aus

$$M_A = b\omega + J_A \frac{d\omega}{dt} + M_L \quad (26)$$

zusammen. Dabei ist J_A das Massenträgheitsmoment des Rotors, b der Reibkoeffizient des Rotors und M_L das Lastmoment, welches am Rotor angreift.

Das System des Motors an sich kann also mit Hilfe der Gleichungen (18) bis (20), (23) und (26) durch

$$\frac{di_A}{dt} = \frac{u_A - R_A i_A - k_M \omega}{L_A} \quad (27)$$

$$\frac{d\omega}{dt} = \frac{k_M i_A - b\omega - M_L}{J_A} \quad (28)$$

beschrieben werden.

5.3.2 Mechanisches System Fensterheber

In diesem Abschnitt wird bei der Modellierung von einem System mit Seilzug ausgegangen, da eine Tür aus einem Kfz mit diesem System zur Vermessung

zur Verfügung stand. In Abbildung 25 ist ein solches System dargestellt. Der

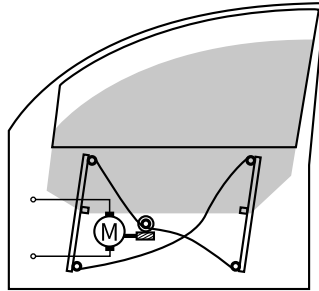


Abbildung 25: Schematische Darstellung des Fensterhebers mit Seilzugsystem

Rotor des Motors triebt ein Schneckengetriebe an, welches mit dem Seilzug verbunden ist. Der Seilzug läuft über mehrere Umlenkrollen und bewegt zwei Führungsschienen auf oder ab, welche mit der Fensterscheibe verbunden sind. Zunächst wird ein ideales System ohne Reibung oder andere Verluste betrachtet. Abbildung 26 zeigt für diesen Fall ein einfaches Modell der Mechanik des Fensterhebers. Die Rotation des Schneckenrades $\dot{\varphi}_G$ setzt sich aus der Rotation des Ankers $\dot{\varphi}$ und der Untersetzung G zusammen.

$$\dot{\varphi}_G = \frac{\dot{\varphi}}{G} \quad (29)$$

Bei der Übertragung des Moments muss neben der Untersetzung und dem Massenträgheitsmoment des Ankers J_A , auch der Getriebewirkungsgrad berücksichtigt werden, da dieser bei Schneckengetrieben deutlich kleiner als eins ist.

$$M_G = G \cdot \eta_G \cdot (M_A - J\ddot{\varphi} - b\dot{\varphi}) = F \cdot r \quad (30)$$

Der Effekt durch das Massenträgheitsmoment des Schneckenrades J_G kann aufgrund der geringen Drehzahl vernachlässigt werden [62]. Am Schneckenrad wirkt mit dem Radius r die Kraft F , welche sich aus der Federkraft F_c und der Dämpferkraft F_d zusammensetzt. Der obere Angriffspunkt von Feder und Dämpfer bewegt sich entlang x_1 , was in Abhängigkeit von φ ausgedrückt werden kann:

$$x_1 = \frac{r}{G} \cdot \varphi \quad (31)$$

Auf der anderen Seite des Feder-Dämpfer-Systems ist die Fensterscheibe mit der Masse m befestigt, welche sich entlang x_2 bewegen kann. Somit können die

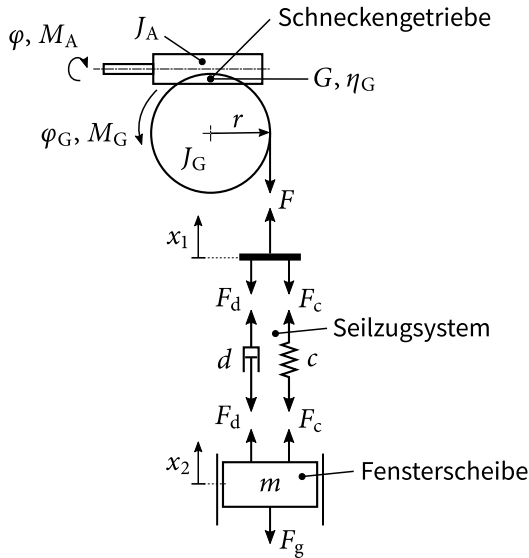


Abbildung 26: Einfaches Modell der Fensterhebermechanik

Kräfte zu

$$F_c = c \left(\frac{r}{G} \varphi - x_2 \right) \quad (32)$$

$$F_d = d \left(\frac{r}{G} \dot{\varphi} - \frac{dx_2}{dt} \right) \quad (33)$$

$$\Rightarrow F = F_c + F_d = c \left(\frac{r}{G} \varphi - x_2 \right) + d \left(\frac{r}{G} \dot{\varphi} - \frac{dx_2}{dt} \right) \quad (34)$$

bestimmt werden. An der Fensterscheibe greift außerdem die Gewichtskraft

$$F_g = m \cdot g \quad (35)$$

mit der Erdbeschleunigung g an. Mit der Bewegungsgleichung des Fensters

$$F_c + F_d = m \cdot \ddot{x}_2 + m \cdot g \quad (36)$$

und mit Gleichungen (29) bis (33) und (35) kann das gesamte mechanische System basierend auf φ und x_2 als Zustandsgrößen

$$J_A \cdot \ddot{\varphi} + \frac{dr^2}{G^2 \eta_G} \cdot \dot{\varphi} + \frac{cr^2}{G^2 \eta_G} \cdot \varphi - \frac{dr}{G \eta_G} \cdot \dot{x}_2 - \frac{cr}{G \eta_G} \cdot x_2 - k_M \cdot i_A = 0 \quad (37)$$

$$m \cdot \ddot{x}_2 + d \cdot \dot{x}_2 + c \cdot x_2 - \frac{dr}{G} \cdot \dot{\varphi} - \frac{cr}{G} \cdot \varphi + m \cdot g = 0 \quad (38)$$

beschrieben werden.

5.3.3 Reibung

In Abbildung 27 ist zum Vergleich der Stromverlauf beim Herunter- und Herauffahren des Fensterhebers dargestellt. Es ist zu erkennen, dass das Niveau

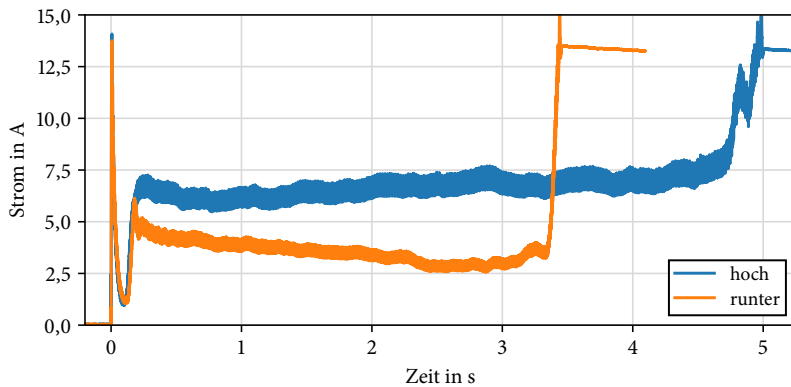


Abbildung 27: Vergleich der Stromverläufe beim Herauf- und Herabfahren der Fensterscheibe

beim Hochfahren größer ist als beim Herunterfahren, da beim Hochfahren die Gravitationskraft dem Antrieb entgegenwirkt. Allerdings ist nicht nur das mittlere Stromniveau unterschiedlich. In beide Richtungen ist eine Veränderung des mittleren Stromes zu erkennen.

In Abbildung 28 (a) ist das Fenster komplett unten. Diese Position ist mit $x_2 = 0$ definiert. Fährt das Fenster nun hoch, fangen die Seiten des Fensters an im Eingriff zu sein (b). Die Länge des Eingriffes an dieser Kante ist linear abhängig von

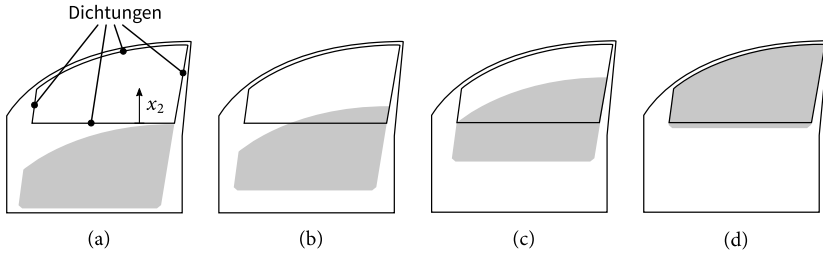


Abbildung 28: Verschiedene Positionen des Fensters

x_2 . Zeitgleich durchfährt das Fenster die untere Dichtung. Hier ist die Abhängigkeit der Berührungslänge von x_2 von der Form der Oberkante des Fensters abhängig. Diese ist oft gebogen. Ab einer gewissen Position des Fensters, wie in (c) dargestellt, ist auch die kurze linke Kante des Fensters im Eingriff mit der Gummidichtung. Ab diesem Punkt nimmt die Länge des Eingriffes dort linear mit x_2 zu. Das Fenster fährt weiter hoch bis es wie in (d) durch die obere Dichtung hindurch bis an den Anschlag fährt, wobei $x_2 = x_{2,\max}$ ist. Alle diese Umstände mind in das Modell einzubeziehen, würde die Parametrierung zu aufwendig machen für einen zeiteffektiven HIL-Test. Man kann die Veränderung des mittleren Stromes im quasi-stationären Bereich wie in Abbildung 29 linear annähern. Die Unterschiede zwischen minimalem und maximalem Strom der jeweiligen dargestellten Gerade ist dabei in etwa gleich. Zu der Gummireibung kommen auch noch andere Reibungseffekte im gesamten System hinzu, zum Beispiel durch die Führung der Schienen. Mit einer minimalen Reibkraft $F_{r,\min}$ und einer maximalen Reibkraft $F_{r,\max}$, kann der Betrag der resultierende Reibkraft für die jeweilige Position mit

$$|F_r(x_2)| = F_{r,\min} + \frac{x_2}{x_{2,\max}} \cdot (F_{r,\max} - F_{r,\min}) \quad (39)$$

bestimmt werden. Die Reibkraft wirkt immer entgegen der Bewegungsrichtung. Somit ist die Reibkraft:

$$F_r(x_2, \dot{x}_2) = \text{sgn}(\dot{x}_2) \left(F_{r,\min} + \frac{x_2}{x_{2,\max}} \cdot (F_{r,\max} - F_{r,\min}) \right) \quad (40)$$

Durch Erweiterung von Gleichung (38) ergibt sich

$$m \cdot \ddot{x}_2 + d \cdot \dot{x}_2 + c \cdot x_2 - \frac{dr}{G} \cdot \dot{\varphi} - \frac{cr}{G} \cdot \varphi + m \cdot g + F_r = 0. \quad (41)$$

In Abbildung 30 ist ein Vergleich des gemessenen Stromverlaufes beim Herunterfahren mit dem Ergebnis einer Simulation des bis hier hergeleiteten Modells

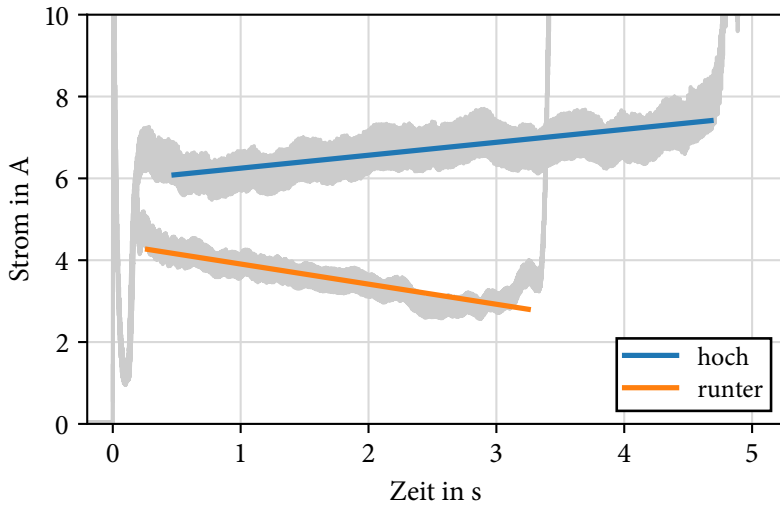


Abbildung 29: Vergleich der Stromverläufe beim Herauf- und Herabfahren der Fensterscheibe mit Linearisierung im quasi-stationären Bereich

abgebildet. Die genutzten Parameter werden ab Abschnitt 5.4 bestimmt. Das generelle Verhalten des Stromverlaufs wird schon relativ gut abgebildet. Im Bereich zwischen 0 s bis 0,25 s ist allerdings noch ein deutlich anderes Verhalten zu beobachten. Dies wird im folgenden Abschnitt modelliert.

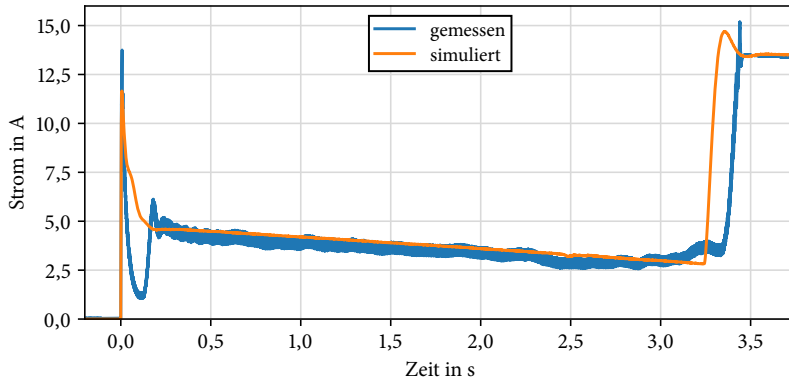


Abbildung 30: Stromverlauf Herunterfahren: Messung und Simulation

5.3.4 Seillose

In Abbildung 31 fällt auf, dass beim Anfahren des Fensterhebers der Strom zunächst auf ein niedriges Niveau abfällt (rot), bevor er wieder steigt und sich auf einem höheren Niveau einpendelt (grün). Das Stromniveau im roten Bereich ab

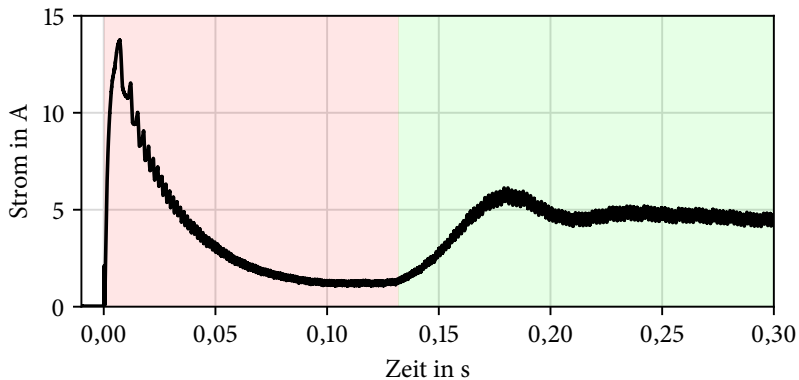


Abbildung 31: Verlauf des Stromes beim Anfahren des Fenster aus geschlossenem Zustand

etwa 0,1 s entspricht dem Leerlaufstrom. Es greift in dem Bereich also scheinbar kein Moment an. Dies resultiert aus der Art der Befestigung der Seilenden an der vom Schneckenrad angetriebenen Rolle. Die Hülsen der Seilenden sind

auf entgegengesetzten Seiten der Rolle eingehängt und je nach Drehwinkel mehr oder weniger um die Rolle gewickelt. Wird die Drehrichtung geändert, muss sich der Einhängpunkt auf der Rolle erst eine gewisse Länge auf dem Umfang der Rolle zurücklegen, bevor Zugkräfte übertragen werden können. Dieser Effekt wirkt sowohl beim Herauf- als auch beim Herunterfahren, da das Seilzugsystem selbsthemmend ist. Wird dieser Punkt erreicht, muss das Seil erst auf Spannung gebracht werden, damit die volle Kraft übertragen werden kann. Dies erklärt den langsamen Anstieg zu Beginn des grünen Bereichs. Durch die Federsteifigkeit c und die Dämpfung d des Seilzugsystems schwingt der Strom nach dem Anstieg zunächst.

5.3.5 Kommutierung

Die Stromrippel der Kommutierung werden von einigen Steuergeräten zur Motorsteuerung genutzt und müssen deshalb abgebildet werden. Für eine detaillierte Modellierung des Stromverlaufs einer mechanischen Kommutierung von Gleichstrommotoren müssen viele Effekte berücksichtigt werden:

- Lamellenüberdeckung der Kohlebürsten,
- veränderliche Kontaktflächen zwischen Kohlebürste und Lamellen,
- Verteilung der magnetischen Flussdichte am Luftspalt,
- Einfluss der Stromdichte auf den Widerstand der Kohlebürste,
- Kontaktwiderstand zwischen Bürste und Lamelle,
- Position der Kohlebürsten am Umfang des Kommutators und anderes.

Jeder dieser Faktoren hängt wiederum von vielen Eigenschaften ab.

Bei Stell- oder Komfortantrieben aus dem Automobilbereich sind die Auswirkungen der Kommutierung auf den Stromverlauf besonders deutlich ausgeprägt. In Abbildung 32 ist der Stromverlauf während der Kommutierung für den hier Betrachteten Fensterhebermotor dargestellt. Die Amplitude der Stromrippel hat eine Größenordnung von etwa 10 % des mittleren Stroms. Für die hier dargestellte Anwendung muss eine möglichst einfache Modellierung der Kommutierung erfolgen. Dabei muss diese zugleich einfach zu parametrieren sein.

Ein mögliches elektrisches Ersatzschaltbild, welches die Kommutierung mit abbildet ist in Abbildung 33 dargestellt. Aber auch für die Berechnung nach diesem noch verhältnismäßig einfachen Ersatzschaltbild werden viele Parameter benötigt. Während der Widerstand einer Spule R_s und die Induktivität einer

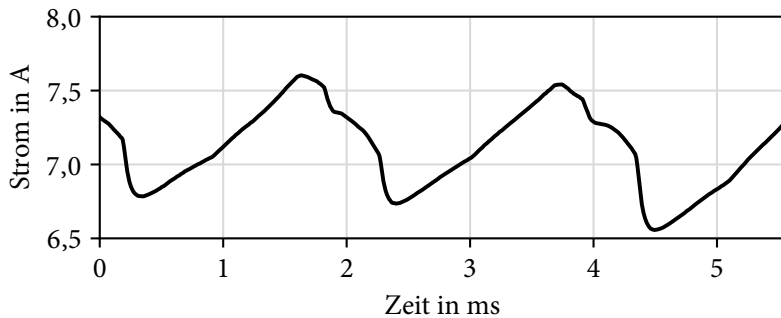


Abbildung 32: Stromverlauf (Rippel) bei der Kommutierung des Fensterhebermotors

Spule L_s relativ einfach bestimmt werden können, ist die Bestimmung von R_{B1} und R_{Bq} deutlich schwieriger. Neben geometrischen Größen der Bürste, wird auch die Leitfähigkeit des Bürstenmaterials, die Drehwinkelabhängige Kontaktfläche von der Bürste zu der jeweiligen Lamelle und die Abhängigkeit des Widerstandes von der Stromdichte benötigt. Dazu kommt bei einer Unter- oder Überkommutierung auch noch die Modellierung des entstehenden Funkens und der damit verbundenen schnellen Stromänderung, welche hier noch nicht berücksichtigt werden. Diese physikalische Modellierung ist somit nicht praktikabel für den verfolgten Ansatz der Modellierung.

Ein deutlich simplerer Ansatz ist die Verwendung eines Parametermodells für die Welligkeit, welches auf einem sinusförmigen Verlauf basiert.

Dabei werden die Modellgleichungen zunächst ohne Kommutierungsrippel mit dem Ankerstrom i_A gelöst (Gleichung (27)). Der Gesamtstrom wird aus der Überlagerung von i_A mit einem sinusförmigen Rippelstrom i_r berechnet. Nach Rechnungen und Messungen [10] hängt die Höhe des Rippelstromes sowohl vom Laststrom i_A als auch von der Drehzahl des Motors ab. Diese Abhängigkeiten können durch

$$i_r = \frac{(i_A \cdot a + 1) \cdot \hat{I}_{R\infty} \cdot \hat{\varphi}}{\sqrt{\omega_x^2 + \hat{\varphi}^2}} \cdot \sin(z \cdot \varphi) \quad (42)$$

angenähert werden. Die Lamellenzahl z ist bekannt. Der Stromeinflussfaktor a , der Scheitelwert des Stromrippel $\hat{I}_{R\infty}$ bei hoher Drehzahl und die Eck-Winkelgeschwindigkeit ω_x müssen durch Messungen bestimmt werden.

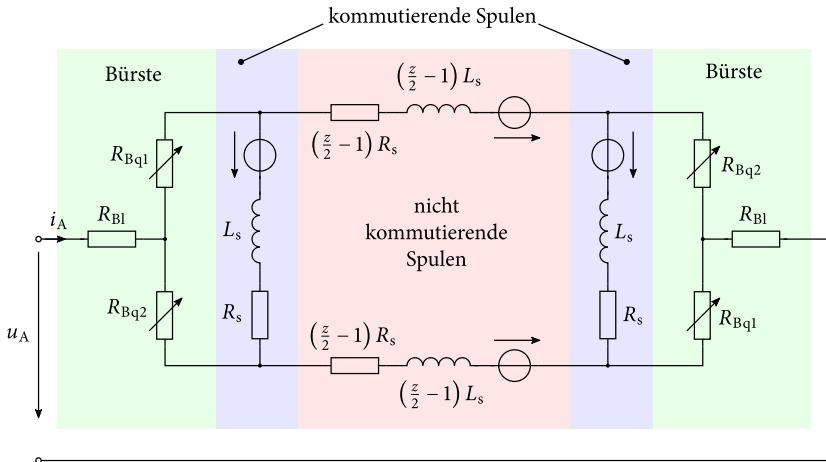


Abbildung 33: Mögliches Ersatzschaltbild für die Kommutierung [25]

5.3.6 Zustände

Das Modell muss verschiedenen Zuständen abbilden, zwischen denen mit bestimmten Schaltbedingungen gewechselt wird.

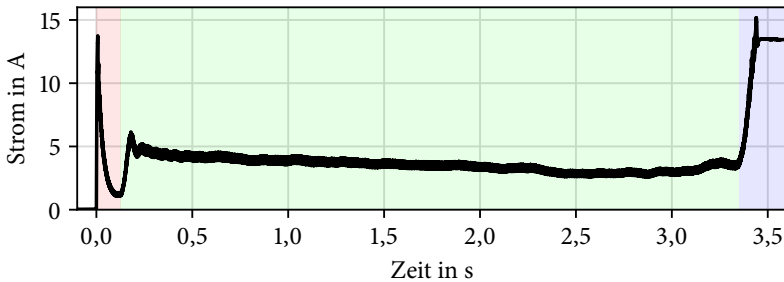


Abbildung 34: Stromverlauf beim Herunterfahren eines Fensterhebers mit verschiedenen Zuständen

In Abbildung 34 sind die hauptsächlich wirksamen Zustände farblich markiert. In grün ist der *Normalbetrieb* dargestellt. In diesem Zustand gelten die bisher hergeleiteten Gleichungen ohne Zwangsbedingungen.

Ein weiterer Zustand ist die Blockierung der Scheibe durch die Anschläge, dies

ist in Abbildung 34 blau markiert. Ist $x_2 = 0$ oder $x_2 = x_{2,\max}$ wird x_2 auf diesem Wert gehalten bis das Fenster sich wieder in die andere Richtung bewegen würde. Für den oberen Anschlag ist dabei zu berücksichtigen, dass so lange in dem Zustand des Anschlages verblieben werden muss, bis das Fenster aktiv durch den Motor bewegt wird. Die Selbsthemmung des Getriebes verhindert eine Abwärtsbewegung auch ohne Motormoment.

Der Bereich der Seillose (Abschnitt 5.3.4) ist in Abbildung 34 rot markiert. Sobald ein Richtungswechsel der Scheibe erfolgt, wird für einen bestimmten Rotationswinkel φ_1 das Lastmoment am Anker $M_L = 0$ gesetzt. Wird dieser Winkel überschritten wird wieder in den normalen Zustand gewechselt.

Die Selbsthemmung des Schneckengetriebe bewirkt, dass der Rotationswinkel φ konstant ist, so lange

$$F \cdot r > M_G \quad (43)$$

gilt (Gleichungen (30) und (34)).

Somit kann das Modell zunächst mit vier Zustände modelliert werden:

- Normalbetrieb,
- Blockieren des Fensters (durch Anschlag),
- Loses Seil,
- Selbsthemmung.

Die Schaltbedingungen zwischen diesen Zuständen beinhalten nicht nur physikalisch begründete Bedingungen. Durch numerische Effekte bei der Simulation kann es ohne zusätzliche Bedingungen zu einem schnellen mehrfachen Wechsel zwischen verschiedenen Zuständen kommen. In Anhang A.3 werden alle Zustände und Schaltbedingungen dargestellt. Für die einfachere Implementierung der Schaltbedingungen wurde der Zustand *Normalbetrieb* in zwei Zustände für den Normalbetrieb beim Herauffahren und für den Normalbetrieb beim Herunterfahren aufgeteilt. Diese Zustände unterscheiden sich nicht beim Lösen der Gleichungen, ermöglichen aber eine simplere Handhabung der Bedingungen.

5.4 Parametrierung des Fensterhebers

Im Folgenden werden die Parameter der in Abschnitt 5.3 hergeleiteten Gleichungen anhand der in Tabelle 7 markierten Methoden bestimmt.

5.4.1 Elektrischer Widerstand des Ankerkreises

Um den über den Drehwinkel gemittelten Widerstand zu bestimmen, wurden Messwerte aufgenommen, während der Anker des Fensterhebermotors mit Hilfe eines zweiten Motors gedreht wurde. Bei der gewählten Drehzahl von etwa $5,5 \text{ min}^{-1}$ sind in Gleichung (18) die induzierte Spannung und die Stromänderungsrate vernachlässigbar klein und es gilt $R_A = U_A/I_A$.

In Abbildung 35 ist der Widerstand zwischen den Motorklemmen bei einem

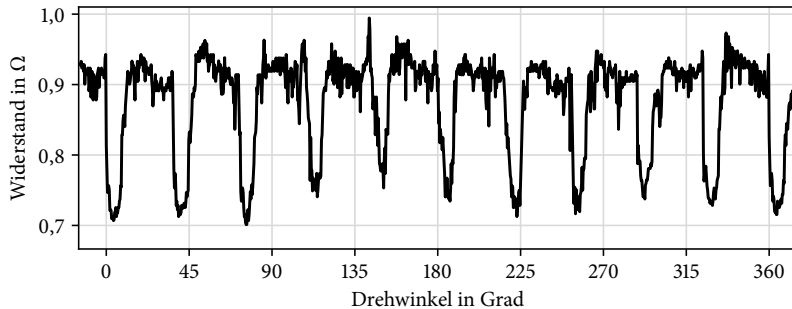


Abbildung 35: Gesamtwiderstand zwischen den Motorklemmen bei einem mittleren Strom von etwa 2 A

mittleren Ankerstrom von etwa 2,07 A über dem Drehwinkel des Rotors dargestellt. Die erste Kommutierung beginnt ab 0° . Der Widerstandswert sinkt wie zu erwarten um etwa 20 % ab, da zwei der zehn Spulen kurzgeschlossen sind. Da die Widerstandswerte der einzelnen Spulen aufgrund von Fertigungstoleranzen deutlich voneinander abweichen können, ergeben sich die dargestellten Unregelmäßigkeiten.

Der Widerstand des Stromübergangs an den Bürsten ist auch abhängig von der Stromdichte und damit von dem mittleren Ankerstrom. In Abbildung 36 wurde dieser Zusammenhang mit denselben Messungen wie für Abbildung 35 ermittelt. Dabei wurden jeweils Messungen genutzt, welche bei abgekühltem Zustand durchgeführt wurden. Außerdem wurden nur die ersten 5 Sekunden betrachtet, um die Temperaturabhängigkeit vernachlässigen zu können. Ab etwa 1,2 A sieht der Widerstandsverlauf qualitativ ähnlich wie in [25] aus.

In Abbildung 37 verändert sich der Widerstand innerhalb der betrachteten Zeit für einen kleineren Strom von etwa 2 A nicht erkennbar. Es findet also keine signifikante Erwärmung statt. Bei höheren Strömen verändert sich der

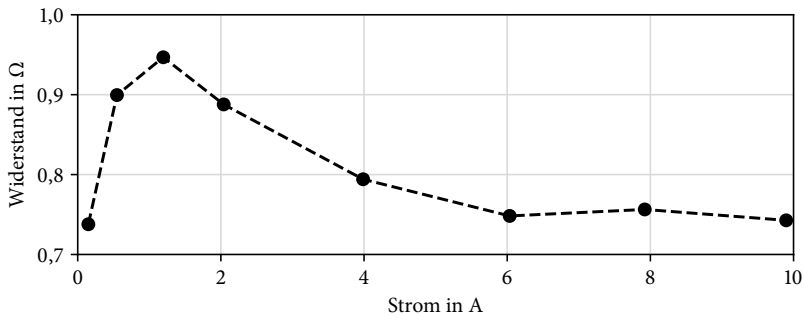


Abbildung 36: Mittlerer Widerstand in Abhängigkeit des mittleren Ankerstroms

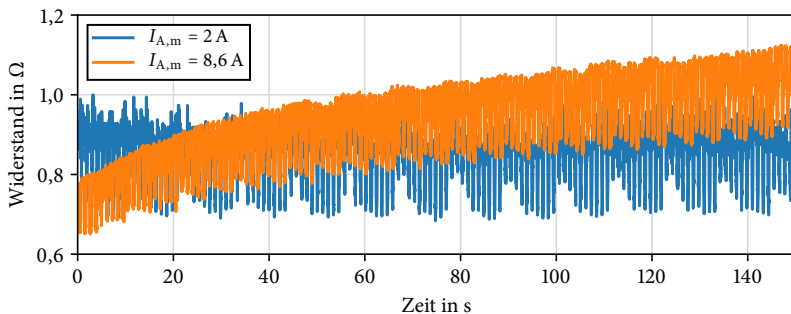


Abbildung 37: Anstieg des Widerstands durch Erwärmung bei einem mittleren Strom von etwa 8,6 A

Widerstand mit der Zeit deutlich. Bei dem dargestellten Widerstandsverlauf bei einem mittleren Strom von 8,6 A dauert es etwa 5 s bis eine Änderung von etwa 5 % erreicht wird. Eine normale Fahrt der Fensterscheibe über die gesamte Distanz liegt in der Regel unter dieser Zeit. Die Erwärmung braucht in dem Modell deshalb nicht nachgebildet werden, da der Fensterheber in der Regel nur wenige Sekunden am Stück in Betrieb ist.

Im normalen Betrieb befindet sich der Strom in der Regel im Bereich zwischen 2 A bis 8 A. Durch Mittlung der Werte aus Abbildung 36 in diesem Bereich ergibt sich der mittlere Ankerwiderstand $R_A = 0,79\ \Omega$.

Ist es nicht ohne weiteres möglich entsprechende Messungen vorzunehmen, kann der Widerstand auch abgeschätzt werden. In Datenblättern sind in der

Regel einige Daten für den Nennbetriebspunkt enthalten. Im Datenblatt des hier verwendeten Motors finden sich zum Beispiel Nennspannung U_N , Nennstrom I_N und Nennleistung (mechanisch) P_N . Ist der Wirkungsgrad des Getriebes η_G bekannt oder wird mit 30 % bis 40 % abgeschätzt (Abschnitt 5.4.6), kann der Ankerwiderstand mit

$$R_A = \frac{U_N I_N - P_N / \eta_G}{I_N^2} \quad (44)$$

näherungsweise berechnet werden. In diesem Fall ergibt sich aus Gleichung (44) mit den Werten aus dem Datenblatt und einem Wirkungsgrad von 35 % ein Ankerwiderstand von $0,8 \Omega$. Dieser Schätzwert ist somit sehr nah an dem weiter oben genauer ermittelten Widerstand.

5.4.2 Induktivität des Ankerkreises

Der Versuchsaufbau für die Messung der Induktivität ist ähnlich wie bei den Messungen zum Ankerwiderstand. Der Fensterhebermotor wird sehr langsam von einem zweiten Motor mit konstanter Geschwindigkeit gedreht. Dabei ist ein LCR-Messgerät der Firma *Hioki* an den Motorklemmen angeschlossen. Bei einer Messfrequenz von 42 Hz ergibt sich eine mittlere Induktivität von $L_A = 0,95 \text{ mH}$ [10].

5.4.3 Motorkonstante

Die Motorkonstante k_M ist oft in Datenblättern zu finden. Ist sie nicht bekannt, kann sie durch Gleichungen (24) und (25) bestimmt werden:

$$k_M = \frac{u_i}{\omega} = \frac{M_A}{i_A} \quad (45)$$

In der Regel ist die Bestimmung von k_M mit minimalen Mitteln am einfachsten, indem der Motor im Generatorbetrieb bei Leerlauf betrieben wird. Im Leerlauf gilt $i_A = 0 \text{ A}$, damit ist $u_A = u_i$. Durch Messung der Winkelgeschwindigkeit ω , beziehungsweise der Drehzahl n und der Ankerspannung an den Motorklemmen u_A , ergibt sich die Motorkonstante aus

$$k_M = \frac{u_A}{\omega} = \frac{u_A}{2\pi n} \quad (46)$$

In Abbildung 38 ist eine Messung für verschiedene Winkelgeschwindigkeiten

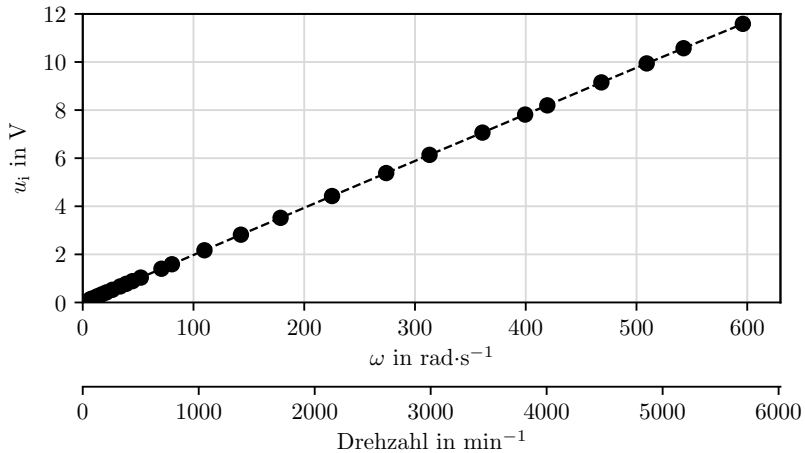


Abbildung 38: Induzierte Spannung im Generatorbetrieb des Motors für verschiedene Drehzahlen

durchgeführt worden. Dabei wurde der gesamte real vorkommende Drehzahlbereich abgedeckt. Der Zusammenhang zwischen u_A und ω ist fast ideal linear. Der Parameter k_M kann also tatsächlich als konstant angenommen werden und es reicht die Messung der Ankerspannung für eine einzelne Drehzahl im generatorischen Leerlauf aus, um diesen zu bestimmen. Der Mittelwert der obigen Messungen ergibt $k_M = 0,0196 \text{ V s}$. Dabei schwankt der Wert lediglich in einem Bereich von 2,5 %, was vermutlich auf Messfehlern beruht.

Die Bestimmung der Motorkonstante über Ankerstrom und Ankermoment kann ebenfalls durchgeführt werden, wenn die Möglichkeit besteht an dem Anker ein definiertes Moment anzulegen. Dies ist zum Beispiel mit Hilfe eines zweiten Motors mit bekannter Kennlinie oder bekannter Motorkonstante möglich. Dann müssen nur die Ströme der beiden Motoren bestimmt werden, um daraus die Motorkonstante abzuleiten.

5.4.4 Reibkoeffizient des Motors

Der Reibkoeffizient b des Motors lässt sich mit Hilfe einer Messung bei Leerlauf bestimmen, wenn die Motorkonstante k_M bekannt ist. Dazu wird der mittlere

Strom im Leerlauf I_0 und die Winkelgeschwindigkeit im Leerlauf ω_0 benötigt:

$$b = \frac{k_M \cdot I_0}{\omega_0} = \frac{k_M \cdot I_0}{2 \cdot \pi \cdot n_0} \quad (47)$$

Die Winkelgeschwindigkeit kann entweder direkt gemessen werden oder aus der Frequenz der Stromrippel mit der Lamellenzahl berechnet werden. Für den untersuchten Motor ergibt sich mit $\omega_0 = 589,4 \text{ rad s}^{-1}$ und $I_0 = 0,831 \text{ A}$ ein Reibkoeffizient von $b = 2,76 \cdot 10^{-5} \text{ N m s rad}^{-1}$.

5.4.5 Massenträgheitsmoment des Ankers

Die Massenträgheit des Ankers lässt sich über die Geometrie und der mittleren Dichte des Ankermaterials berechnen.

Alternativ kann aus der mechanischen Zeitkonstante des Motors beim Anfahren im Leerlauf das Massenträgheitsmoment berechnet werden, wenn die elektrischen Parameter R und L , die Motorkonstante k_M und der Reibkoeffizient b bereits bekannt sind. Ist diese Zeitkonstante nicht bekannt, kann diese aus einem Verlauf der Winkelgeschwindigkeit bei einem Sprung der Ankerspannung ermittelt werden. Ist solch ein Verlauf nicht bekannt, kann dieser aus einer Messung der Ankerspannung und des Ankerstroms mit

$$\omega = \frac{u_A - R \cdot i_A - L \cdot \frac{di_A}{dt}}{k_M} \quad (48)$$

bestimmt werden. Dieses ist zwar nicht der reale Verlauf, nähert diesen aber an. In Abbildung 39 (oben) ist der gemessene Strom beim Anlauf im eingebauten Zustand am Fensterheber dargestellt. Dieser kann hier genutzt werden, da wie in Abschnitt 5.3.4 beschrieben, der Motor quasi im Leerlauf anfährt, wenn es einen Richtungswechsel gibt. In Abbildung 39 (unten) ist der mit Gleichung (48) berechnete Verlauf von ω und eine PT_1 -Annäherung

$$\omega(t) = K \cdot \left(1 - e^{-\frac{t}{\tau}}\right) \quad (49)$$

dessen dargestellt. Die Parameter der PT_1 -Funktion können beispielsweise mit der Methode der kleinsten Quadrate bestimmt werden und ergeben sich in diesem Fall zu $\tau = 27,3 \text{ ms}$ und $K = 594,9 \text{ rad s}^{-1}$. Das Standardmodell für einen Gleichstrommotor ohne Last lässt sich mit dem Zustandsraummodell mit

$$\dot{x} = \underbrace{\begin{bmatrix} -R_A/L_A & -k_M/L_A \\ k_M/J_A & -b/J_A \end{bmatrix}}_A x + \begin{bmatrix} 1/L_A \\ 0 \end{bmatrix} u, \quad x = \begin{bmatrix} i_A \\ \omega \end{bmatrix} \quad (50)$$

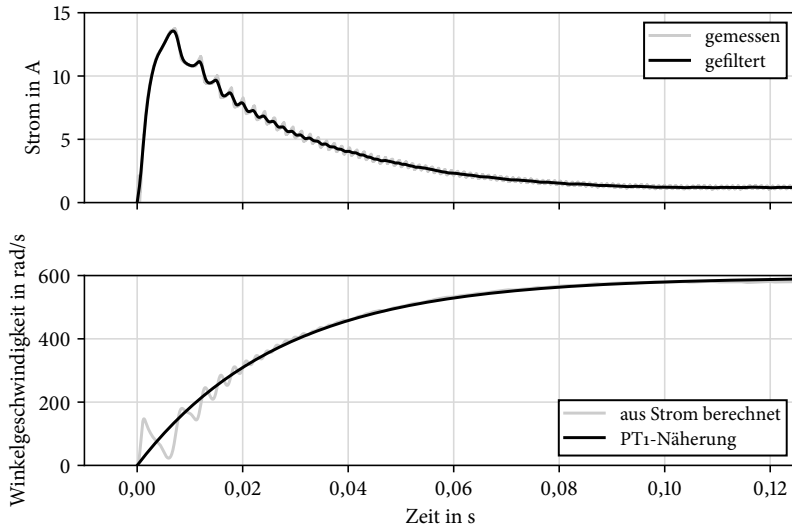


Abbildung 39: Anlaufverhalten ohne Last; oben: gemessener Strom;
 unten: Berechneter Drehzahlverlauf und Approximation mit PT_1 Glied.

darstellen. Dass eine gute Approximation des Drehzahlverlaufes mit einer einzigen e-Funktion möglich ist, zeigt, dass einer der beiden Eigenwerte des Systems in Bezug auf die Winkelgeschwindigkeit sehr dominant ist oder dass beide Eigenwerte etwa gleich sind. In jedem Fall entspricht die berechnete Zeitkonstante recht genau einem der Eigenwerte. Durch Umformung des charakteristischen Polynoms der Systemmatrix A

$$\left(-\frac{R_A}{L_A} - \lambda\right) \left(-\frac{b}{J_A} - \lambda\right) + \frac{k^2}{J_A L_A} = 0 \quad (51)$$

nach der Massenträgheit und durch Einsetzen der gefundenen Zeitkonstante, ergibt sich

$$J_A = -\frac{R_A b + b L \lambda + k^2}{R_A \lambda + L \lambda^2} = 1,46 \cdot 10^{-5} \text{ kg m}^2, \quad \text{mit } \lambda = -\frac{1}{\tau} = -36,7 \text{ s}^{-1}. \quad (52)$$

5.4.6 Getriebe

Die Parameter des Getriebes sind der Getriebewirkungsgrad η_G und die Untersetzung G . Die Untersetzung ist durch die Gangzahl Z_1 der Schnecke und die

Zähnezahl Z_2 des Schneckenrades mit

$$G = \frac{Z_2}{Z_1} \quad (53)$$

bekannt. Der Wirkungsgrad ist oft nicht bekannt. Bei der Form von Schneckengetrieben, wie sie in der Regel in automobilen Komfortantrieben zum Einsatz kommen, kann der Wirkungsgrad mit 30 % bis 40 % abgeschätzt werden.²

Mit den typischer Weise in Datenblättern enthaltenen Werten und dem bekannten Ankerwiderstand, lässt sich der Getriebewirkungsgrad näherungsweise bestimmen. Mit dem Gesamtwirkungsgrad

$$\eta = \frac{P_N}{U_N I_N} \quad (54)$$

und dem theoretischen Motorwirkungsgrad

$$\eta_M = \frac{U_N - I_N R_A}{U_N} \quad (55)$$

lässt sich der Getriebewirkungsgrad mit

$$\eta_G = \frac{\eta}{\eta_M} = 0,3 \quad (56)$$

berechnen.

5.4.7 Federsteifigkeit und Dämpfungsmaß des Seilzugsystems

Federsteifigkeit und Dämpfungsmaß des Systems sind nur schwer messtechnisch genau zu bestimmen, da diese sich aus den Eigenschaften vieler Komponenten des Fensterhebersystems ergeben. Eine genauere Untersuchung wurde in [62] vorgenommen. Daraus wurde die Größenordnung für diese Parameter abgeleitet. Durch Parametervariation und Vergleich der Simulationsergebnisse mit den Messwerten wurden für diesen Fensterheber passende Werte gefunden: Federsteifigkeit $c = 45 \cdot 10^3 \text{ N m}^{-1}$ und Dämpfungsmaß $d = 310 \text{ N s m}^{-1}$.

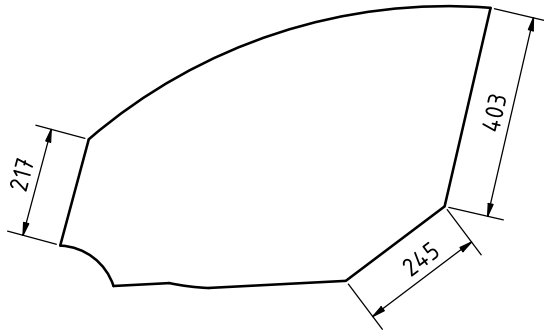


Abbildung 40: Fläche der Fensterscheibe

5.4.8 Masse des Fenstersystems

Die Masse des Fensters kann anhand der Geometrie angenähert werden. Für das untersuchte System ergibt sich die Fläche der Scheibe zu $A_F = 0,336 \text{ m}^2$. Die Glasscheibe besteht hauptsächlich aus Siliciumdioxid und weist somit eine Dichte von $\rho_{\text{SiO}_2} = 2650 \text{ kg m}^{-3}$ auf [49, 59]. Zusammen mit der Dicke der Scheibe von $d_F = 5 \text{ mm}$ ergibt sich die Masse der Fensterscheibe zu

$$m_F = A_F \cdot d_F \cdot \rho_{\text{SiO}_2} = 4,45 \text{ kg}. \quad (57)$$

Hinzu kommen noch die Massen der Metallhalterungen der Führung und des Stahlseils. Mit diesen zusammen ergibt sich ein geschätzte Gesamtgewicht der mit x_2 bewegten Masse von etwa $m = 5 \text{ kg}$.

5.4.9 Reibkräfte

Auf die Reibkräfte kann geschlossen werden, indem der Stromverlauf des gesamten Vorgangs des Herunterfahrens betrachtet wird. Für die Umrechnung des Stromes auf eine äquivalente Kraft, welche am Schneckenrad angreift wird der Umrechnungsfaktor

$$H = G \cdot \frac{k_M^2 \cdot \eta_G + b \cdot R_A}{k_M \cdot r} \quad (58)$$

eingeführt. Damit kann eine dem Strom äquivalente Kraft

$$F_i = H \cdot (i_A - I_0) \quad (59)$$

²abgeschätzt mit Daten aus [17]

bestimmt werden. Zur Berechnung der Reibkraft F_r werden die Kräfte $F_{r,\min}$ und $F_{r,\max}$ benötigt. Diese können zum Beispiel aus der in Abbildung 41 dargestellten

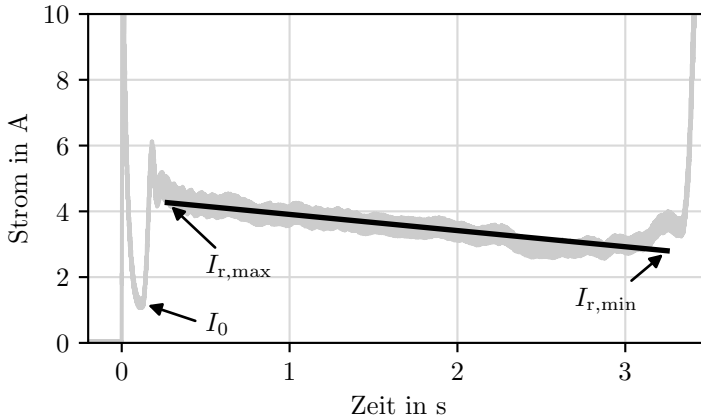


Abbildung 41: Stromverlauf zur Ermittlung der Reibkräfte

Messung bestimmt werden:

$$F_{r,\min} = H \cdot (I_{r,\min} - I_0) , \quad (60)$$

$$F_{r,\max} = H \cdot (I_{r,\max} - I_0) . \quad (61)$$

Somit reicht es zur Parametrierung der Reibkraft die drei Ströme I_0 , $I_{r,\min}$ und $I_{r,\max}$ zu kennen. Mit den Werten aus Abbildung 41 ergibt sich $F_{r,\min} = 30 \text{ N}$ und $F_{r,\max} = 75 \text{ N}$.

5.4.10 Parameter Seillose

Aus einer zeitlich gut aufgelösten Messung des Stromes kann die Anzahl der Rippel N_R bis zum Einsetzen der Zugkraft des Seils gezählt werden. Dies entspricht dem rot markierten Bereich in Abbildung 31. Zusammen mit der Anzahl der Lamellen z ergibt sich der Drehwinkel

$$\varphi_L = \frac{2\pi N_R}{z} , \quad (62)$$

welcher zurückgelegt werden muss, bis das Lastmoment angreift. Mit $N_R = 96$ aus Abbildung 31 ist $\varphi_L = 60,3 \text{ rad}$.

5.4.11 Kommutierungsparameter

Simulationsrechnungen für verschiedene Motoren zeigten, dass die bezogenen Parameter für alle Motoren etwa die gleiche Größenordnung haben [10]. Diese sind als Schätzung in Tabelle 9 angegeben. Eine genauere Parametrierung wird

Tabelle 9: Schätzwerte der Rippelparameter

Größe	Symbol	Schätzwert
Stromeinflussfaktor	a	$10/I_{A,n}$
Scheitelwert der Stromrippel bei hoher Drehzahl	$\hat{I}_{R\infty}$	$0,2 \cdot I_{A,n}$
Eck-Winkelgeschwindigkeit	ω_x	$0,3 \cdot \omega_n$

durch schrittweise manuelle Anpassung der Parameter und detaillierteren Vermessungen der Stromrippel vorgenommen. Somit ergibt sich für diesen Motor: $a = 0,56 \text{ A}^{-1}$, $\hat{I}_{R\infty} = 0,1 \text{ A}$ und $\omega_x = 125 \text{ rad s}^{-1}$.

5.4.12 Modellierung von Störfällen

Bei den Tests von Steuergeräten geht es typisch darum, bestimmte Störungen des normalen Ablaufes zu simulieren. Für das Modell des Fensterhebers wurden zwei solcher Fälle implementiert: Einklemmen eines Fremdkörpers und das Festfrieren der Scheibe im Winter. Der letztere Fall wird an dieser Stelle beispielhaft erläutert.

Bildet sich ein durchgehender Eisfilm zwischen den Dichtungen und der Scheibe, muss dieser beim Anfahren des Fenster zunächst aufgebrochen werden. Außerdem ist es vorstellbar, dass es durch die raue Oberfläche des Eises zu einer erhöhten Reibung zwischen der Fensterscheibe und den Gummidichtungen kommt.

Der Zustand kann also durch die Einführung einer Losbrechkraft F_b und einer eventuellen Anpassung der Reibkraft abgebildet werden. Zunächst wird die Simulation im Zustand des losen Seils gestartet. Von dort geht es beim Einschalten der Spannung nun nicht in den Zustand des normalen Betriebs über, sondern in einen extra Zustand für die angefrorene Scheibe. So lange nun

$$F_c + F_d \leq F_b \quad (63)$$

ist, gilt

$$x_2 = x_{2,\max} , \quad (64)$$

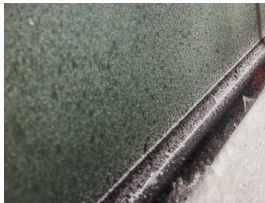
$$\dot{x}_2 = \ddot{x}_2 = 0 . \quad (65)$$

Ist $F_b > F_c + F_d$, wird in den Zustand des Normalbetriebs gewechselt.

Zur Ermittlung der erforderlichen Losbrechkraft wurden in einem Tiefkühlcontainer unterschiedlich starke Eisschichten auf die Fensterscheibe einer Autotür aufgebracht und der Strom- und Spannungsverlauf beim Anfahren gemessen. Abbildung 42 zeigt den Versuchsaufbau in dem Container und verschiedene



Versuchsaufbau



(a) leicht vereist



(b) stärker vereist



(c) sehr stark vereist

Abbildung 42: Versuchsaufbau für angefrorene Scheibe und verschiedene Grade der Vereisung

Grade der Vereisung. In Abbildung 43 sind die zugehörigen Stromverläufe beim Anfahren des Fensterhebers zu sehen. Bei der sehr starken Vereisung schafft es der Motor nicht, die Eisschicht aufzubrechen und das Fenster zu bewegen. Das Seil wird gestrafft, bis das Motormoment nicht mehr ausreicht den Rotor weiterzudrehen. In den anderen Fällen weisen alle Stromverläufe während der Phase der normalen Fensterbewegung ab circa 0,5 s das gleiche Niveau auf. Die Gleitreibung zwischen Fenster und Dichtung ändert sich anscheinend nur in geringem Maße. Die Berechnung der Reibkraft F_r muss somit nicht angepasst werden.

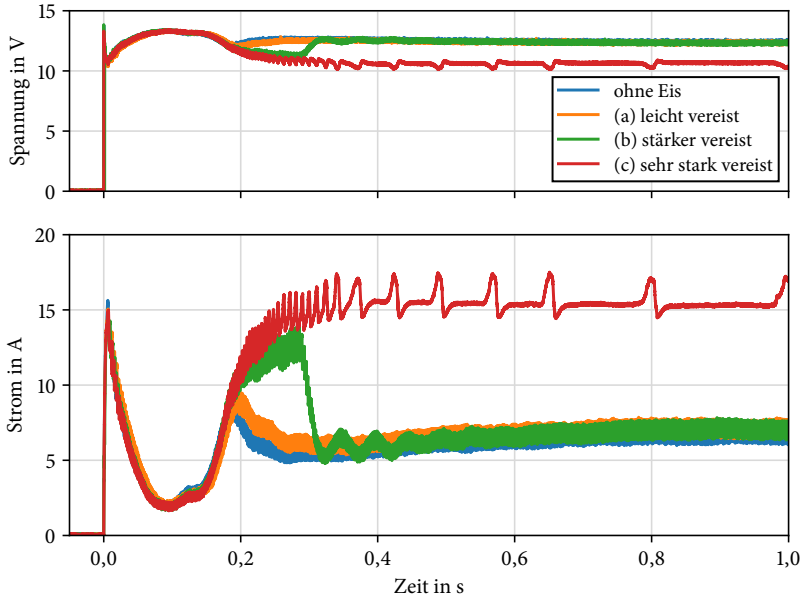


Abbildung 43: Versuchsaufbau für angefrorene Scheibe und verschiedene Grade der Vereisung

Es reicht also aus, eine Losbrechkraft zu definieren. Diese kann anhand von anderen Parametern des Modells und aus dem Strom ermittelt werden. Der maximale stationäre Ankerstrom ergibt sich beim Blockieren des Motors aus

$$i_{A,\max} = \frac{u_A}{R_A}. \quad (66)$$

Mit dem Umrechnungsfaktor H ergibt sich die Kraft

$$F = F_{\text{block}} = H \cdot \frac{u_A}{R_A}, \quad (67)$$

ab der der Motor blockiert. Mit dem dimensionslosen Faktor $f_b = 0 \dots 1$ kann die Losbrechkraft

$$F_b = f_b \cdot H \cdot \frac{u_A}{R_A} \quad (68)$$

auf den Blockstrom normiert eingestellt werden.

5.4.13 Validierung und Ergebnisse

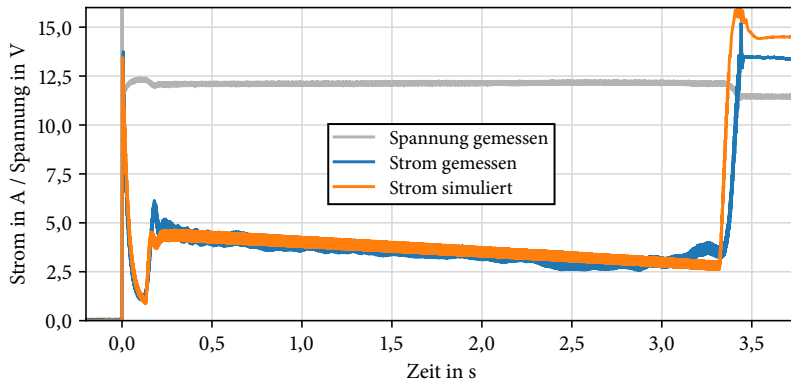


Abbildung 44: Vergleich Messung und Simulation des Gesamtmodells beim Herunterfahren

Eine Zusammenfassung der Modellgleichungen ist in Anhang A.4 zu finden. Mit den in Abschnitt 5.4 ermittelten Werten für die jeweiligen Parameter (Anhang A.5) und den Modellzuständen aus Abschnitt 5.3.6 (Anhang A.3) ergeben sich die in Abbildungen 44 bis 46 dargestellten Simulationsergebnisse. Die dargestellte gemessene Spannung dient auch als Eingangsgröße für die Simulation. Es ist insgesamt eine sehr gute Übereinstimmung gegeben. Der maximal erreichte Einschaltstrom und der Blockstrom nach Erreichen des Anschlages weisen eine signifikante Abweichung zu den Messwerten auf. Diese beiden Größen hängen allerdings stark von dem exakten Drehwinkel des Ankers φ ab. Das ist sowohl für den realen Motor, als auch für die Simulation der Fall. Es sind hier also Abweichungen von bis zu 20 % durch die Lage der Bürsten auf den Lamellen des Kommutators bei Stillstand zu erwarten. Beim Herauffahren gibt es außerdem eine deutliche Abweichung am Ende der Verfahrens. Hier taucht die reale Fensterscheibe in die obere Gummidichtung ein. Dieser Effekt ist nicht modelliert, woraus sich die zu sehende Abweichung ergibt. Auch wird der Blockstrom in der Simulation früher erreicht.

In Abbildung 46 sind Ausschnitte aus Abbildungen 44 und 45 dargestellt. Diese zeigen die Kommutierungsrippel bei verschiedenen Stromniveaus und damit auch bei verschiedenen Drehzahlen. Sowohl Frequenz, als auch Amplitude der simulierten Schwingung stimmen mit den Messwerten überein. Die genaue Form der Schwingung ist nicht relevant für Tests von Steuergeräten. Die Model-

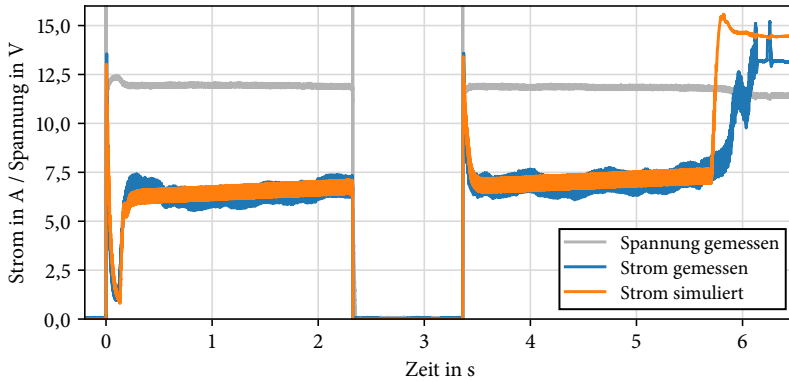


Abbildung 45: Vergleich Messung und Simulation des Gesamtmodells beim Herauffahren mit Zwischenstopp

lierung des Ripplestromes durch eine Sinusfunktion nach Gleichung (42) führt zu einer für Tests brauchbaren Annäherung.

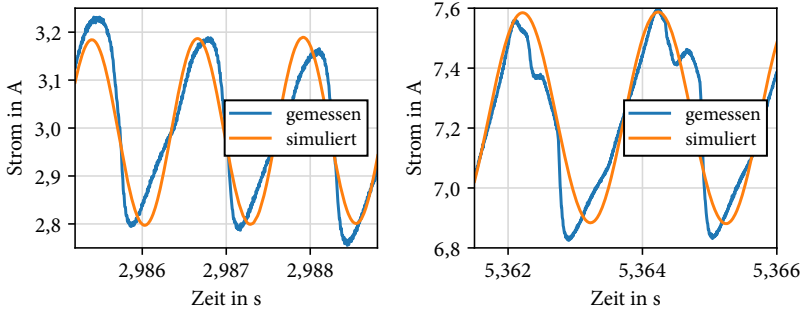


Abbildung 46: Vergleich Messung und Simulation des Gesamtmodells; Ausschnitte der Kommutierungsrippel bei unterschiedlichen mittleren Strömen

6 Numerische Lösung

Die Modellgleichungen werden in Echtzeit numerisch gelöst. Es werden verschiedene Löser implementiert, welche mit einer standardisierten Modellnotation genutzt werden können. Dabei wird sowohl der Löser als auch die Schrittweite automatisiert mit optionalen Vorgaben des Nutzers gewählt.

Die Eigenwerte und die daraus resultierenden Zeitkonstanten der Modelle können um mehrere Größenordnungen voneinander abweichen. Während sich mechanische Vorgänge im Bereich von Sekunden oder Millisekunden abspielen, können elektrische Vorgänge im Bereich von Mikrosekunden liegen. Durch Variation aller Parameter der Differentialgleichungen des Fensterhebermodells in einem sinnvollen Bereich ergibt sich ein Parameterraum. Mit diesem können alle vorkommenden Eigenwerte berechnet werden, woraus sich Zeitkonstanten zwischen $50 \mu\text{s}$ und 1 s ergeben. Es kann somit vorkommen, dass ein Modell in einem Zeitbereich von Sekunden mit Schrittweiten im Bereich von Mikrosekunden gelöst werden muss. Somit könnten Rundungsfehler einen signifikanten Einfluss haben.

Numerische Lösungsverfahren für Differentialgleichungen (DGLs) können in zwei Kategorien eingeteilt werden: Explizite und implizite Löser. Bei expliziten Verfahren können in jedem Zeitschritt die zeitlichen Ableitungen unmittelbar aus den im letzten Zeitschritt ermittelten Größen berechnet werden, bei impliziten Verfahren können die zeitlichen Ableitungen nur iterativ berechnet werden. Bei Echtzeitsimulationen werden in der Regel einfache explizite Verfahren genutzt, da diese einen geringen Rechenaufwand haben und eine vorhersehbare Anzahl an Berechnungen durchführen. Allerdings können auch implizite Löser mit Einschränkung genutzt werden, wenn die Anzahl der Iterationen auf ein Maximum begrenzt wird. Für das hier vorliegende Konzept werden beide Arten von Lösern implementiert, implizite Löser dienen dabei allerdings nur als Ausweichlösung. Eine Modellanalyse entscheidet unter Berücksichtigung der auf der Zielhardware realisierbaren Schrittweiten, welcher Löser am besten geeignet ist.

Desweiteren können die Verfahren jeweils weiter kategorisiert werden, in Verfahren mit konstanter und variabler Schrittweite und in Ein- und Mehrschrittverfahren. Da bei Echtzeitanwendungen eine maximale Latenz zwischen Ein- und Ausgangssignal eingehalten werden muss und möglichst klein sein soll, bieten sowohl eine variable Schrittweite als auch Mehrschrittverfahren eher Nachteile als Vorteile und werden im Weiteren nicht betrachtet.

Der verwendete Löser muss folgende Kriterien einhalten:

- Die Stabilität muss immer gewährleistet sein,
- die Genauigkeit muss in einem angemessenem Bereich liegen,
- die Latenz muss möglichst gering sein und
- das Echtzeitkriterium muss erfüllt werden (Rechenzeit kleiner Schrittweite).

Daraus ergeben sich die zu klärenden Aspekte:

- Wahl des Lösungsverfahrens,
- Wahl der Schrittweite,
- Art der Zahlenrepräsentation.

6.1 Fehler

Bei der numerischen Lösung von Differentialgleichungen treten, abgesehen von unumgänglichen Ungenauigkeiten der Parameter und Eingangsgrößen, im wesentlichen zwei Arten von Fehlern auf. Zum einen der Diskretisierungsfehler, welcher durch das Lösen eines kontinuierlichen mathematischen Problems an diskreten Stützstellen mit endlich genauer Approximation durch einen Löser entsteht. Zum anderen der Rundungsfehler, welche durch die endliche Genauigkeit der Zahlendarstellung auf Maschinen entsteht. Beide Fehler sind sowohl von der Schrittweite h , als auch von dem gewählten Löser abhängig. Während der Diskretisierungsfehler mit abnehmender Schrittweite sinkt, nimmt der Rundungsfehler mit abnehmender Schrittweite zu. Somit ergibt sich ein Optimum mit minimalem Fehler. In dem speziellen Fall der Echtzeitsimulation muss allerdings auch berücksichtigt werden, dass bestimmte Latenzen eingehalten werden müssen und der Löser nur eine bestimmte Rechenzeit zur Verfügung hat. Daraus ergeben sich untere und obere Grenzen für die wählbare Schrittweite.

Je schneller die Änderung der Zustandsvariablen ist, umso kleiner muss die Schrittweite des jeweiligen Lösers sein, um den Diskretisierungsfehler klein zu halten. Somit wird der Diskretisierungsfehler maßgeblich durch den betragsmäßig größten Eigenwert bestimmt. Dieser wird maßgeblich von den elektrischen Parametern bestimmt. Die zugehörige Zeitkonstante kann mit

$$\tau_{\min} = \frac{L}{R} \quad (69)$$

angenähert werden.

Zur Fehlerabschätzung wird deshalb die numerische Lösung der Differentialgleichung

$$u = R \cdot i + L \cdot \frac{di}{dt} \quad (70)$$

mit der analytischen Lösung

$$i(t) = \frac{1}{R} \left(1 - e^{-t \frac{R}{L}}\right) = \frac{1}{R} \left(1 - e^{-\frac{t}{\tau}}\right) \quad (71)$$

für einen Einheitssprung der Spannung u verglichen.

Im Folgenden wird das Maximum des bezogenen Fehlers des simulierten Stromes

$$\varepsilon_{\max}^* = \max \left(\frac{\varepsilon_{\max}}{i_{\infty}} \right) = \max \left(\frac{|i_{\text{ana}} - i_{\text{num}}|}{i_{\infty}} \right) \quad (72)$$

mit der exakten analytischen Lösung i_{ana} , der jeweiligen numerischen Lösung i_{num} und dem Strom nach einer unendlichen Zeit $i_{\infty} = u/R$ für die Betrachtungen verwendet. In der Praxis kommen elektrische Zeitkonstanten von $\tau_{\min} = 10 \mu\text{s}$ bis $\tau_{\max} = 10 \text{ms}$ vor. Dafür ist in Abbildung 47 der bezogene Fehler

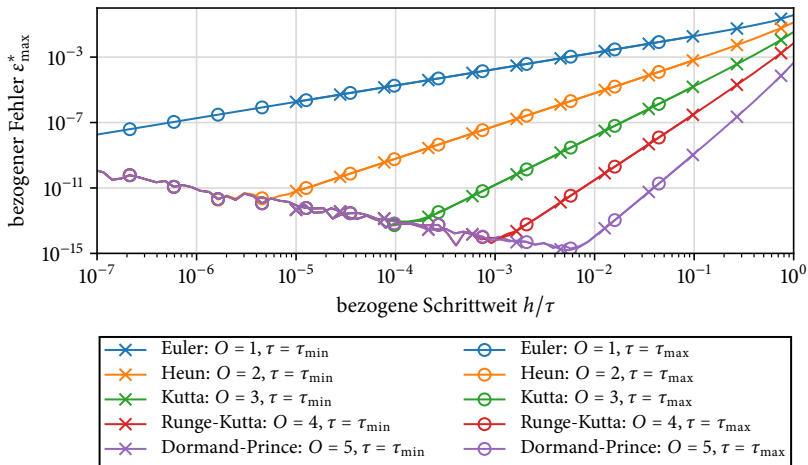


Abbildung 47: Maximaler bezogener Fehler in Abhängigkeit von der bezogenen Schrittweite, der Ordnung O und der Zeitkonstante τ (Rechnungen mit 64 Bit Gleitkommazahlen, $\tau_{\min} = 10 \mu\text{s}$, $\tau_{\max} = 10 \text{ms}$)

aus Gleichung (72) für verschiedene explizite Verfahren unterschiedlicher Ordnung zu sehen. Im Bereich links der Minima dominiert der Rundungsfehler,

im Bereich rechts davon der Diskretisierungsfehler. Dieser ist in Abhängigkeit von der auf die jeweilige Zeitkonstante bezogene Schrittweite dargestellt. Der Fehler in dieser bezogenen Darstellung für τ_{\min} und τ_{\max} ist wegen der Repräsentation von Gleitkommazahlen mit Mantisse und Exponent gleich. Der Rundungsfehler wird je nach Löser erst bei mehreren Größenordnungen unterhalb der Zeitkonstante dominant. Zwar steigt der Rundungsfehler mit der Komplexität des Modells, da bei diesem Konzept aber nur Systeme mit einer geringen Anzahl von Gleichungen betrachtet werden, ist davon auszugehen, dass der Rundungsfehler für die zu realisierenden Schrittweiten immer in einem zu vernachlässigenden Größenbereich liegen.

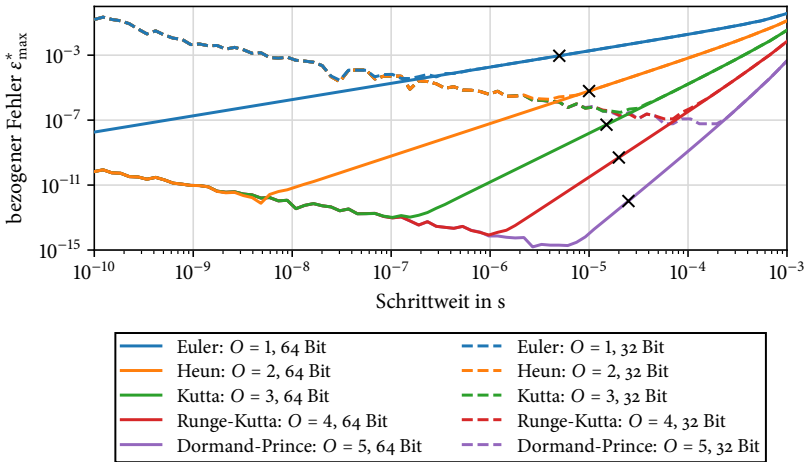


Abbildung 48: Vergleich des maximalen bezogenen Fehlers mit 64 Bit und 32 Bit Fließkommazahlen in Abhängigkeit von der Schrittweite ($\tau = 1$ ms, $i_{\infty} = 1$ A)

Abbildung 48 zeigt, dass bereits bei dieser simplen Simulation mit lediglich einer Differentialgleichung die Berechnung mit 32 Bit Fließkommazahlen zu einem Rundungsfehler führt, welcher mehrere Größenordnungen über dem mit 64 Bit Fließkommazahlen liegt. Bei einem komplexeren Modell kann dieser bereits in einem signifikanten Bereich liegen. Da auch Kompaktrechner mit 64 Bit Architektur gut verfügbar sind, werden die Berechnungen immer mit 64 Bit Fließkommazahlen durchgeführt. Abbildung 48 zeigt außerdem, dass bereits bei diesem simplen Problem der Diskretisierungsfehler im realisierbaren Schrittweitenbereich zwischen 10^{-5} s bis 10^{-4} s bei den Lösern erster und zweiter Ordnung in einem signifikanten Bereich von Milliampere liegen. Es

sollten daher Löser höherer Ordnung eingesetzt werden, wenn die dadurch resultierende höhere Latenz mit der restlichen HIL-Umgebung vereinbar ist.

Der Rechenaufwand steigt bei Runge-Kutta-Verfahren in etwa linear mit der Anzahl der Funktionsauswertungen. Diese stimmen bis zu einer Ordnung von vier mit der Ordnung überein [12]. Zur Einhaltung der Echtzeitanforderung steigt deshalb auch die minimal erreichbare Schrittweite etwa linear mit der Ordnung an. Wird von einer realistischen Schrittweite von $5 \mu\text{s}$ bei einem Löser erster Ordnung und den entsprechenden Vielfachen dieser Schrittweite für höhere Ordnungen ausgegangen, ergeben sich die mit \times in Abbildung 48 gekennzeichneten Fehler des jeweiligen Löser. Der Fehler sinkt trotz zunehmender Schrittweite mit der Ordnung. Allerdings hat die Simulation mit einem Löser höherer Ordnung auch eine mehrfach höhere Latenz, was für einige Anwendungen zu viel sein kann.

Die Untersuchungen zeigen, dass der Rundungsfehler bei einer Rechnung mit 64 Bit Gleitkommazahlen keinen relevanten Einfluss im realisierbaren Schrittweitenbereich hat.

6.2 Stabilität

Das wichtigste Kriterium für die numerische Lösung ist die Stabilität, welche in jedem möglichen Modellzustand garantiert sein muss. Jeder Löser hat ein unterschiedliches Gebiet der (absoluten) Stabilität. Das Stabilitätsgebiet ist

$$S = \left\{ z \in \mathbb{C} : |R(z)| \leq 1 \right\} \quad (73)$$

mit der Stabilitätsfunktion $R(z)$. Die Stabilitätsfunktionen der verschiedenen Löser leiten sich aus den jeweiligen Berechnungsvorschriften des Löser ab. Mit b und A des Butcher-Tableaus [12]

$$\left[\begin{array}{c|c} c & A \\ \hline & b^T \end{array} \right]$$

des jeweiligen Lösungsverfahrens lässt sich die zugehörige Stabilitätsfunktion mit

$$R(z) = 1 + zb^T (I - zA)^{-1} \mathbf{1} \quad (74)$$

bestimmen [55]. Damit ergeben sich die Stabilitätsfunktionen der implementierten Löser zu

$$R(z) = z + 1 \quad (\text{Euler, } O = 1), \quad (75)$$

$$R(z) = \frac{z^2}{2} + z + 1 \quad (\text{Heun, } O = 2), \quad (76)$$

$$R(z) = \frac{z^3}{6} + \frac{z^2}{2} + z + 1 \quad (\text{Kutta, } O = 3), \quad (77)$$

$$R(z) = \frac{z^4}{24} + \frac{z^3}{6} + \frac{z^2}{2} + z + 1 \quad (\text{Runge - Kutta, } O = 4), \quad (78)$$

$$R(z) = \frac{z^6}{600} + \frac{z^5}{120} + \frac{z^4}{24} + \frac{z^3}{6} + \frac{z^2}{2} + z + 1 \quad (\text{Dormand - Prince, } O = 5), \quad (79)$$

$$R(z) = \frac{1}{1 - z} \quad (\text{Euler implizit, } O = 1). \quad (80)$$

Diese werden zur Bestimmung der grenzstabilen Schrittweite benötigt. Die komplexe Zahl z entspricht

$$z = h\lambda \quad (81)$$

mit der Schrittweite h und dem jeweiligen Eigenwert λ . Während die Eigenwerte vom jeweiligen Modell abhängen, kann die Schrittweite so gewählt werden, dass $h\lambda$ sich im Stabilitätsbereich befindet. Außer für die angeführten expliziten Löser ist in Abbildung 49 auch das Stabilitätsgebiet für das implizite Eulerverfahren dargestellt. Dieses hat den Vorteil, dass bei großen Schrittweiten die Lösung zwar ungenau ist, aber zumindest die Stabilität gewährleistet wird. Deshalb wird ergänzend auch dieser Löser implementiert.

Für ein lineares System kann mit Hilfe der Stabilitätsfunktion die maximale stabile Schrittweite $h_{i,m}$ für jeden Eigenwert λ_i ermittelt werden. Grafisch ist die Multiplikation des Eigenwertes λ mit einer Schrittweite eine Verschiebung der Zahl z auf einer Geraden, welche durch den Eigenwert und null hindurch geht, wie in Abbildung 50 dargestellt. Je kleiner die Schrittweite, um so näher wird z in Richtung null verschoben. Für die Gewährleistung der Stabilität muss $|R(\lambda h)| \leq 1$ gelten. Für jeden Eigenwert kann die zugehörige maximale Schrittweite bestimmt werden, indem die Nullstelle der Gleichung

$$|R(\lambda h)| - 1 = 0 \quad (82)$$

bestimmt wird. Die maximal wählbare Schrittweite für eine stabile Lösung ist

$$h_{\max} = \min (h_m) . \quad (83)$$

Anhand von Abbildungen 49 und 50 wird klar, dass bei gleichen Eigenwerten, die maximale Schrittweite h_{\max} bei unterschiedlichen Lösern unterschiedlich ist. Es kann somit unter Umständen mit Sicht auf die Stabilität sinnvoll sein, einen Löser mit mehr Rechenaufwand und somit einer höheren Schrittweite zu wählen, als einen simpleren Löser mit einer geringeren Schrittweite.

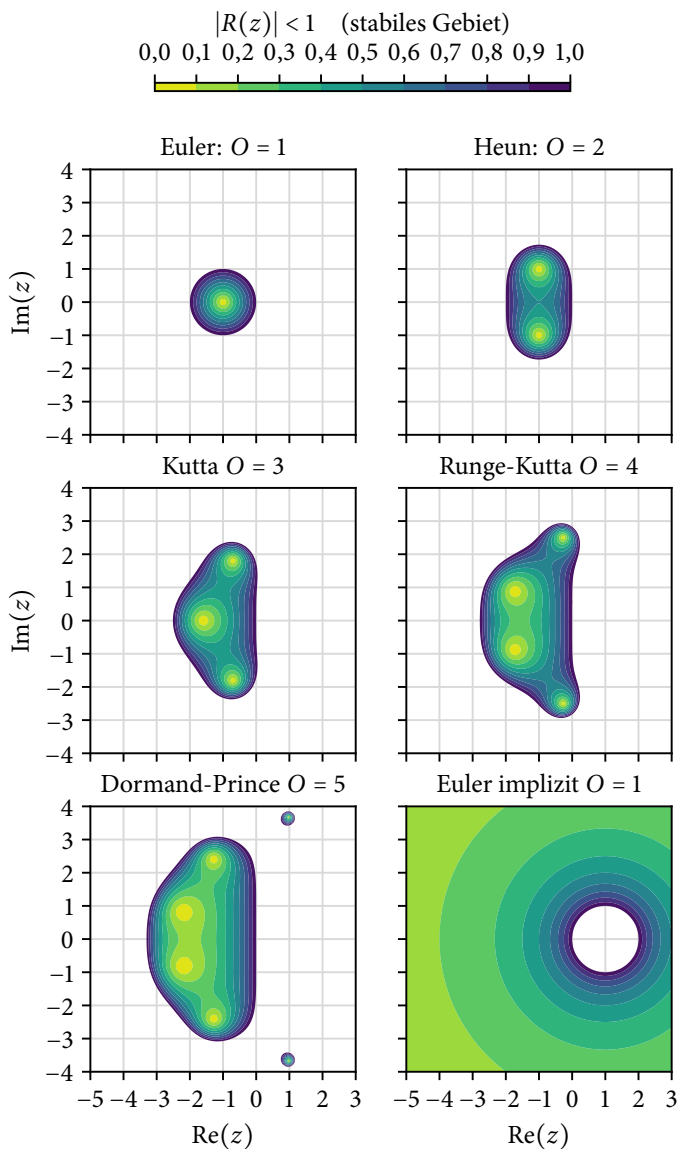


Abbildung 49: Stabilitätsgebiete der einbezogenen Löser aus Gleichungen (75) bis (80) mit der Ordnung O

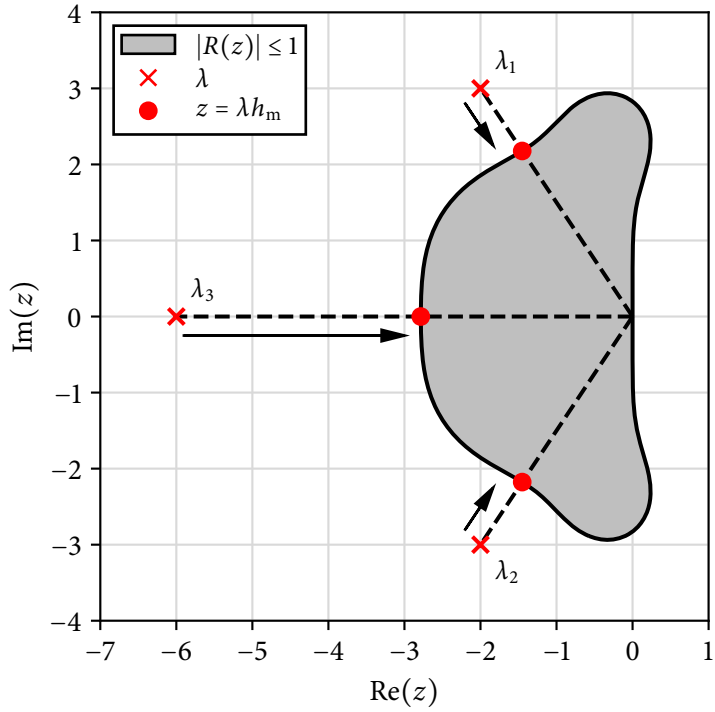


Abbildung 50: Zusammenhang von Schrittweite und Stabilität am Beispiel des Runge-Kutta-Verfahrens 4. Ordnung

6.3 Modellanalyse

Für lineare Systeme

$$\begin{aligned}\frac{d}{dt} x &= A \cdot x + B \cdot u \\ y &= C \cdot x + D \cdot u\end{aligned}\quad (84)$$

können die Eigenwerte aus der Systemmatrix A der Zustandsraumdarstellung [23] bestimmt werden.

Für nicht-lineare Systeme kann das beschreibende Gleichungssystem um einen Entwicklungspunkt linearisiert werden und für dieses linearisierte System die Eigenwerte bestimmt werden. Für ein System von m nicht-linearen Differentialgleichungen in der Form

$$F = \begin{bmatrix} \frac{dx_1}{dt} = f_1(t, x, u) \\ \vdots \\ \frac{dx_m}{dt} = f_m(t, x, u) \end{bmatrix}\quad (85)$$

mit der Zeit t , den Zustandsgrößen x und den Eingangsgrößen u ist die Jacobi-Matrix:

$$J = \nabla F = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}\quad (86)$$

Für einen Entwicklungspunkt x_E ergibt sich daraus die linearisierte Systemmatrix

$$A_E = J(x_E) = \nabla F(x = x_E). \quad (87)$$

Die in Frage kommenden Wertebereiche der jeweiligen Zustandsgrößen x werden mit jeweils k Punkten diskretisiert. Das ergibt einen Zustandsraum mit k^m Entwicklungspunkten und somit auch k^m linearisierte Systemmatrizen. Mit den jeweiligen Eigenwerten und der Vorgehensweise aus Abschnitt 6.2 wird über die Eigenwerte eine m -dimensionale Matrix H_{\max} von maximalen Schrittweiten

für jeden Löser aufgestellt. Die maximal wählbare stabile Schrittweite je Löser ist das Minimum der jeweiligen Matrix

$$h_{\max} = \min(H_{\max}) . \quad (88)$$

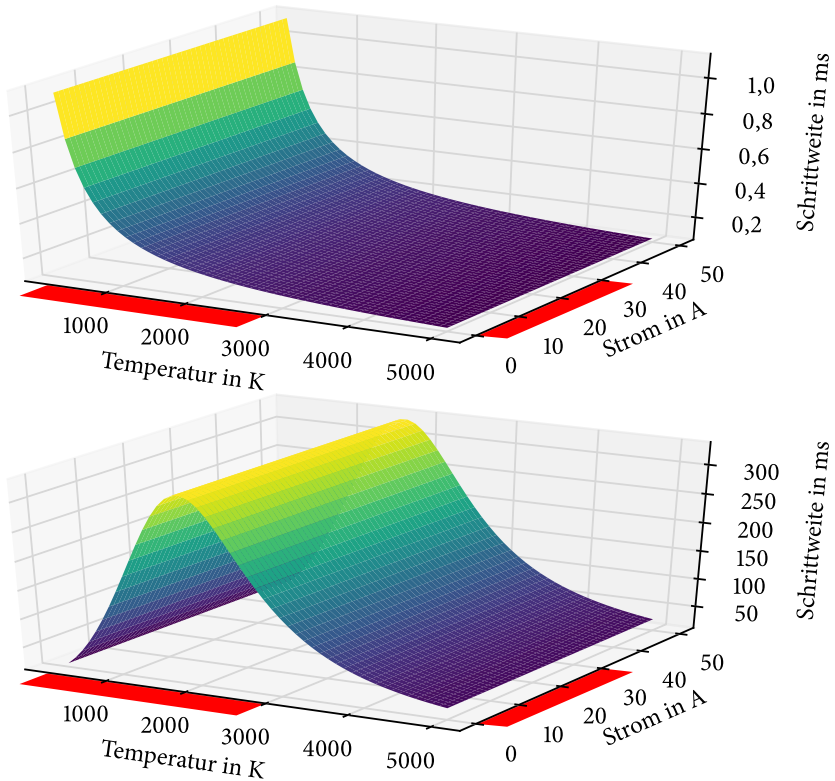


Abbildung 51: Maximale Schrittweite basierend auf dem jeweiligen Eigenwert in Abhängigkeit der Zustandsgrößen eines Modells einer Halogenlampe (Realistischer Arbeitsbereich rot markiert [61])

In Abbildung 51 sind die berechneten maximalen Schrittweiten zu den jeweiligen Eigenwerten eines Modells einer Halogenlampe (Anhang A.6) in Abhängigkeit der Zustandsvariablen Temperatur und Strom zu sehen. Es ist ersichtlich, dass die benötigte Schrittweite bei diesem einfachen Beispiel um eine Größenordnung aufgrund der nicht-linearen Gleichungen variiert. Eine Eigenwertanalyse über den gesamten Arbeitsbereich ist somit notwendig.

6.4 Laufzeitbestimmung

Für die Wahl des Löser muss bekannt sein, welche Schrittweite mit den jeweiligen Lösern auf der Zielhardware realisierbar ist. Dafür wird eine ausreichende Anzahl an Zeitschritten mit einer beliebigen Schrittweite berechnet und die Laufzeit bestimmt. Dadurch kann eine mittlere minimale Schrittweite h_{\min} bestimmt werden. Hat ein Modell verschiedene Zustände, kann es sein, dass die verschiedenen Zustände unterschiedliche Laufzeiten aufweisen. Für solche Modelle muss die mittlere Laufzeit für jeden Zustand bestimmt werden und das Maximum für h_{\min} gewählt werden. Durch einen Test an einem ARM μP , welcher mit einer Linuxdistribution mit Echtzeitkernel betrieben wird, hat sich gezeigt, dass ein Sicherheitszuschlag von 25 % ausreicht, um auch bei Unterbrechung der Rechnung durch das Betriebssystem in über 95 % der Fälle das Echtzeitkriterium zu erfüllen. Die Ausgabe des berechneten Wertes erfolgt über einen externen Taktgeber. Tritt der seltene Fall einer zu langen Störung durch das Betriebssystem auf, sodass das Taktsignal nicht erfasst wird, ist das Ergebnis mit einem geringfügigen Fehler behaftet, da der berechnete Wert für den entsprechenden Zeitschritt erst beim nächsten Takt ausgegeben wird.

6.5 Automatische Auswahl des Löser

Die Modelle sind auf dem Simulationsrechner hinterlegt (Abbildung 3). Vor jedem Simulationsstart werden die Modellparameter von der Bedieneinheit an den Simulationsrechner übermittelt. Auf diesem wird dann die durchschnittliche Laufzeit eines Simulationszeitschrittes inklusive der jeweiligen Ein- und Ausgaben für explizite Löser unterschiedlicher Ordnung ermittelt. Versehen mit einem Sicherheitszuschlag von 25 % ergibt sich die jeweilig minimal mögliche Schrittweite h_{\min} . Außerdem wird für jeden Löser die maximale stabile Schrittweite h_{\max} nach Abschnitt 6.3 ermittelt. Für eine Simulation kommen die Löser in Frage, für welche $h_{\max} > h_{\min}$ erfüllt ist. Zusätzlich kann eine optionale maximale Latenz vom Nutzer hinterlegt werden, welche bei der Auswahl der möglichen Löser berücksichtigt wird. Von den in Frage kommenden Lösern wird derjenige mit der höchsten Ordnung für die Simulation genutzt, da dieser nach Abschnitt 6.1 die genauesten Ergebnisse liefert. Dieser Auswahlprozess ist in Abbildung 52 dargestellt.

Kann kein expliziter Löser das Modell stabil lösen, wird das implizite Eulerverfahren mit einer Begrenzung der Anzahl der Iterationsschritte genutzt. Es hat sich gezeigt, dass in den meisten Fällen bereits wenige Iterationen ausreichen, um die Stabilität zu gewährleisten [24]. Durch die Begrenzung kann aber die

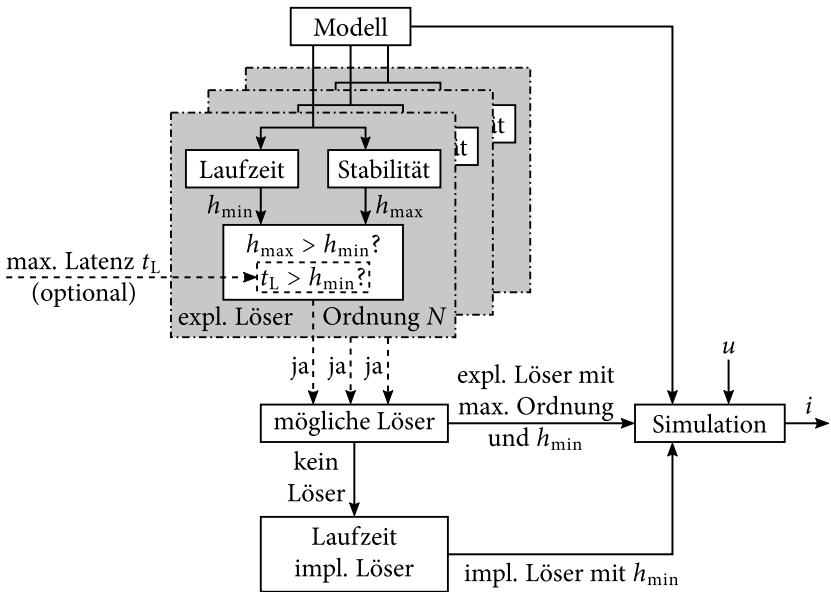


Abbildung 52: Ablaufschema der automatisierten Löserauswahl

Genauigkeit beeinträchtigt werden. Auch wenn das Problem in fast allen Fällen mit einer begrenzten Anzahl an Iterationen stabil gelöst werden kann, ist ein nicht mehr zu vernachlässigender Fehler nicht auszuschließen.

6.6 Modellnotation

Damit die Modellanalyse und die implementierten Löser auf jede Art von Modell angewandt werden können, müssen von linearen Differentialgleichungen bis hin zu NLDAGS, alle Systeme in einer einheitlichen Form notiert werden. Alle hier behandelten Formen von Gleichungssystemen sind Sonderformen von NLDAGS. Somit muss lediglich eine Notation für diese implementiert werden. Diese Notation benötigt folgende Bestandteile:

- Array der Zustandsvariablen,
- Array von Modellparametern,

- Darstellung der Gleichungen in Abhängigkeit von Parametern, Zustandsvariablen und Eingangsgröße wie mit F in Gleichung (85) in einem Objekt oder einer Funktion,
- Funktion zur Auswertung der Jacobimatrix von F ,
- Funktion zur Überprüfung der Zustände und Manipulation der Zustandsvariablen und Parameter.

Für die Implementierung wurde aufgrund der Verfügbarkeit und der Objektorientierung C++ als Programmiersprache gewählt.

7 Exemplarische Implementierung und Ergebnisse

7.1 Komponenten der Simulationshardware

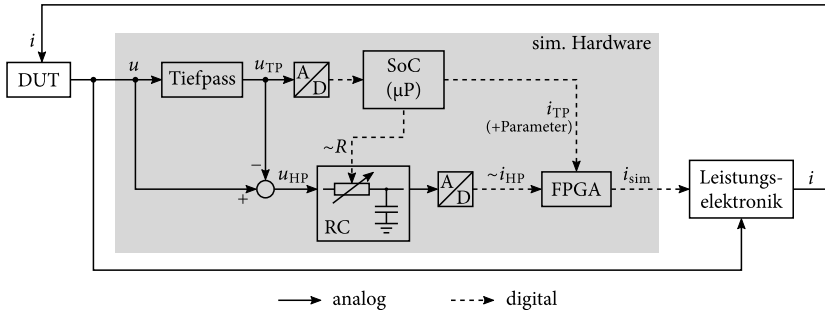


Abbildung 53: Prinzipieller Aufbau der Simulationshardware

Abbildung 53 zeigt den prinzipiellen Aufbau der Implementierung der Simulationshardware.

Der Unterschied zwischen einer Filterung mit Butterworth- und Besselapproximation ist gering (Abbildung 17). Deshalb wird für die Implementierung abweichend von Abschnitt 4 ein verfügbarer Butterworth-Filter dritter Ordnung in Form einer Sallen-Key-Implementierung [57] genutzt. Um die PWM zu unterdrücken wird $f_c = 3 \text{ kHz}$ gewählt. Die entsprechende Schaltung kann mit einem einzigen Operationsverstärker (OPV) realisiert werden.

Die verwendeten ADCs arbeiten nach dem SAR-Prinzip, da diese mit der nötigen Auflösung die geringste Latenz aufweisen. Zum Zeitpunkt der Realisierung waren SAR-Wandler mit 14 Bit bei einer Abtastrate von bis zu 4 MHz verfügbar. Das verwendete Modell [6] arbeitet mit einer Abtastrate von $f_{ADC} = 2,6 \text{ MHz}$. Die Latenz des ADCs liegt bei etwa 300 ns. Bezogen auf den Eingangsbereich von 20 V ergibt sich eine Auflösung von 2,44 mV.

Für die LS-Simulation wird ein Einplatinenrechner mit einem SoC verwendet [48]. Dieser besitzt einen 64 Bit Vier-Kern-Prozessor mit 1,4 GHz Taktung und hat unter anderem GPIOs, über die ADC und FPGA angebunden sind.

Zur Parametrierung der analogen HS-Simulation wird ein digitales Potentiometer [5] verwendet, welches durch den Einplatinenrechner konfiguriert werden

kann. Mit der gewählten Konfiguration können alle üblichen elektrischen Zeitkonstanten der ohmsch-induktiven Verbraucher abgedeckt werden [17].

Der FPGA empfängt die Daten des ADCs der HS-Simulation und des Rechners der LS-Simulation. Diese werden skaliert, addiert und ausgegeben. Für diese Anforderungen reicht ein Modell im unteren Leistungssegment. Es wird ein Modell mit 100 MHz genutzt [1]. Somit können die Prozesse auf 10 ns genau ausgeführt werden, was insbesondere für die Schnittstellen zum ADC und der Leistungselektronik notwendig ist.

Die Anbindung der ADCs erfolgt über eine parallele Schnittstelle. Der FPGA ist in der Lage alle Bits in einem Systemtakt auszulesen. Die Latenz durch das Auslesen liegt somit bei maximal 10 ns. Der SoC hingegen benötigt einige Systemtakte pro Lesevorgang und braucht mehrere Lesevorgänge um alle Bits zu erfassen. Deshalb entsteht eine zusätzliche Latenz von etwa 1,5 μ s beim Auslesevorgang.

Die Übertragung des simulierten Stromes i_{TP} an den FPGA erfolgt über eine serielle Schnittstelle (Serial Peripheral Interface (SPI)). Dieses Protokoll hat keinen Overhead. Somit ist die Datenrate gleich der Taktrate. Mit den gewählten Komponenten kann eine Taktrate bis zu 7,8 MHz realisiert werden. Der Strom wird in einem Format von 17 Bit übertragen. Da die Wortlänge auf acht Bit festgelegt ist, müssen drei Wörter übertragen werden. Somit ergibt sich eine Übertragungszeit von etwa 3 μ s.

Die Ausgabe von i_{HIL} erfolgt mit einer Frequenz von $f_{DAC} = 25$ MHz an die digitale Schnittstelle zur Leistungselektronik.

7.2 Zeitverhalten

Die Latenz der HS-Simulation ist hauptsächlich durch den ADC bedingt. Die gesamte Latenz, inklusive Wandlung, digitaler Verarbeitung und Ausgabe beträgt 385 ns. Davon nimmt die Wandlung den größten Teil von $t_{AD} = 300$ ns in Anspruch.

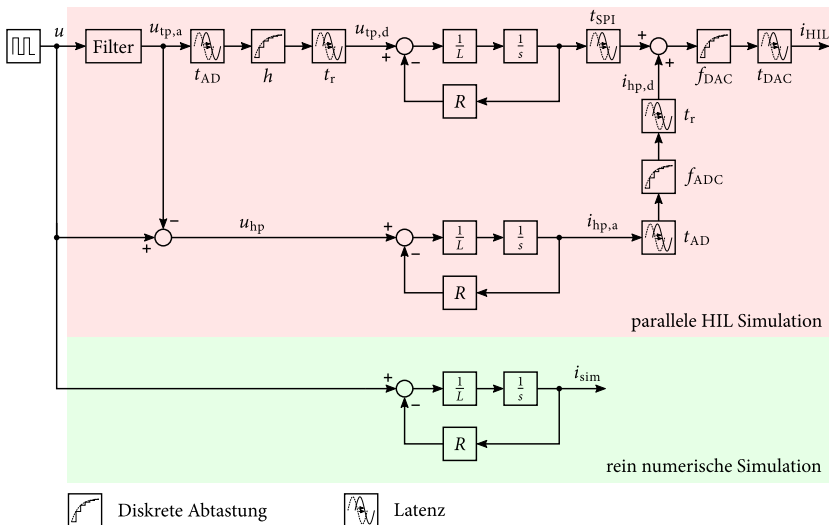
Die Latenz der LS-Simulation ergibt sich aus den Werten in Tabelle 10 zu etwa 10 μ s bis 55 μ s. Dabei hängt die Dauer des Simulationsschrittes stark von der Komplexität des Modells und dem Löser ab.

Um den Einfluss des Zeitverhaltens der implementierten Simulationshardware zu untersuchen, wird die Simulationshardware gemäß Abbildung 54 durch das dargestellte Blockschaltbild modelliert und mit der Eingangsgröße u und

Tabelle 10: Latenzen der einzelnen Vorgänge der LS-Simulation

Vorgang	Bezeichnung	Dauer in μs
Analog-digital-Wandlung	t_{AD}	0,3
Auslesen ADC	t_r	1,5
Simulationsschritt	t_{Sim}	5...50 (üblich)
i_{TP} an FPGA	t_{SPI}	3
i_{TP} Addition und Ausgabe auf FPGA	t_{add}	0,3...0,5

den Ausgangsgrößen i_{HIL} und i_{sim} simuliert. Als Modell wird ein R-L-Glied eingesetzt. Zum Vergleich wird der reale physikalische Vorgang (unterer Zweig in Abbildung 54) simuliert. Beide Simulationen werden mit einem PC und einem üblichen Simulationsprogramm (Matlab/Simulink) durchgeführt. Im oberen Pfad zu i_{HIL} werden die Filterschaltung, die parallele Simulation, die Latenzen und die Abtastraten der implementierten Hardware berücksichtigt.

**Abbildung 54:** Blockschaltbild zum Vergleich zwischen der Simulation eines R-L-Gliedes mit Simulation der HIL-Architektur und rein numerischer Simulation

Aus dieser Simulation ergeben sich die in Abbildung 55 dargestellten Zeitverläufe. Beim Einschwingen des Filters ist durch die Abtastung mit der Schrittweite

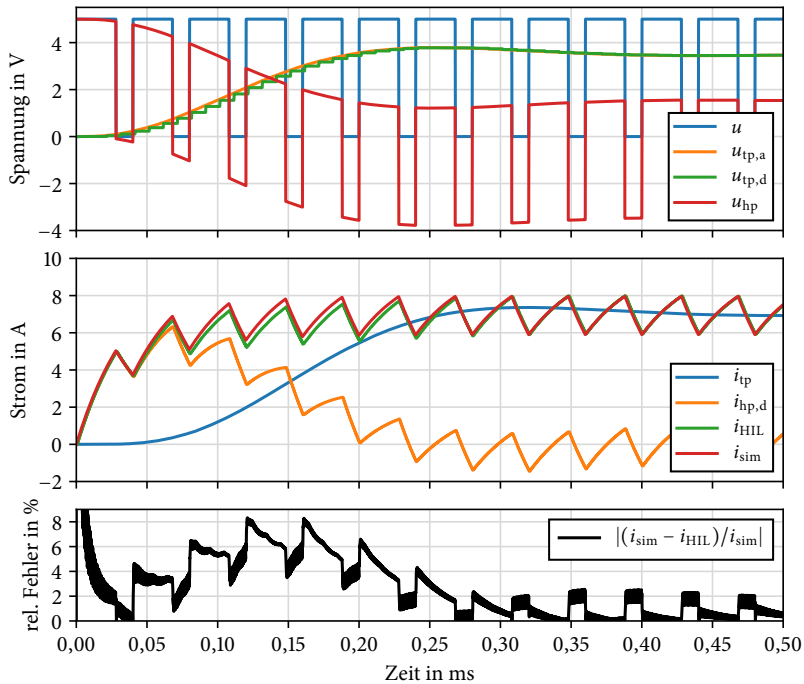


Abbildung 55: Einfluss der HIL-Architektur auf das Simulationsergebnis im Vergleich zu der rein numerischen Simulation ($R = 0,5 \Omega$, $L = 20 \mu\text{H}$, $h = 10 \mu\text{s}$),
Bezeichnungen wie in Abbildung 54

h eine erkennbare Abweichung zwischen der gefilterten analogen Spannung $u_{tp,a}$ und der digitalen Spannung $u_{tp,d}$ gegeben. Dadurch weicht auch der von der HIL-Architektur simulierte Strom i_{HIL} von dem regulär simulierte Strom i_{sim} ab. In Abbildung 55 unten ist die relative Abweichung von i_{HIL} zu i_{sim} dargestellt. Während dem Einschwingvorgang variiert der Fehler zwischen etwa 4 % bis 8 % über der Periode der PWM. Im eingeschwungenen Zustand variiert der Fehler zwischen 0 % bis 2 %. Dies ist die Momentanwertabweichung, die hauptsächlich durch die Latenz und die Abtastung verursacht wird. Die Abweichung der Amplitude der PWM-Schwingungen ist hingegen deutlich geringer und beträgt während dem Einschwingvorgang maximal 1,5 % und ist im stationären Zustand vernachlässigbar.

7.3 Genauigkeit

Der analoge Fehler vom Eingang der Schaltung bis zu der analogen Seite der ADCs kann trotz der Ungenauigkeiten der elektrischen Komponenten durch Kompensationsmaßnahmen bis auf $\pm 0,5$ mV reduziert werden. Zusammen mit dem maximalen Diskretisierungsfehler und dem maximalen Offsetfehler der ADCs ergibt sich eine maximale Abweichung der digitalen Werte von $\pm 14,64$ mV bezogen auf den Eingangsbereich, also eine relative Abweichung von 0,05 % bezogen auf den Skalenendwert.

7.4 Leistungselektronik

Die für die Power Hardware-in-the-loop (PHIL)-Simulation genutzte experimentelle Leistungselektronik empfängt über eine digitale Schnittstelle den Sollwert für einen Strom zwischen $\pm 65,535$ A mit einer Auflösung von 1 mA. Die Aktualisierungsrate der Schnittstelle beträgt 25 MHz. Die maximale Stromänderungsrate ist über den gesamten Bereich konstant und liegt bei $6,4 \text{ A } \mu\text{s}^{-1}$ [32]. Die Leistungselektronik erfüllt somit alle Anforderung an das dynamische Verhalten und an die Auflösung und Amplitude des Stromes.

7.5 Ergebnisse

Im Folgenden wird die Implementierung untersucht und bewertet. Dabei wird zunächst ein Vergleich zwischen der Simulation einer R-L-Reihenschaltung mit der HIL Plattform und der idealisierten numerischen HIL-Simulation aus Abbildung 54 analysiert. Somit wird sowohl in der LS-Simulation als auch in der HS-Simulation das gleiche Modell simuliert. Auf diese Art kann am besten die Abweichung durch das parallele Simulationskonzept von der Realität bestimmt werden. Anschließend wird das komplexere Modell des Fensterhebers mit und ohne Leistungselektronik auf der HIL-Plattform simuliert und mit den tatsächlichen Stromverläufen des realen Fensterhebersystems verglichen. Um die Simulationsergebnisse auch ohne angeschlossene Leistungselektronik einfach auswerten zu können, wird an der Schnittstelle zur Leistungselektronik parallel ein DAC betrieben [7].

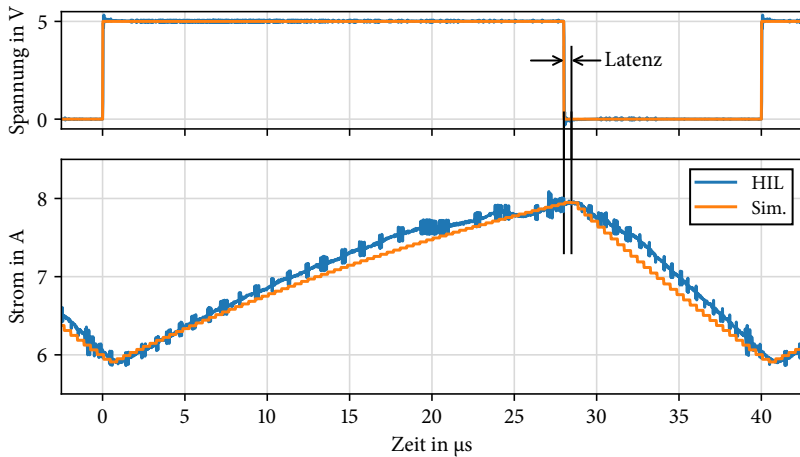


Abbildung 56: Vergleich zwischen realer HIL-Simulation und virtueller HIL-Simulation eines R-L-Reihenschaltung im eingeschwungenen Zustand ($R = 0,5 \Omega$, $L = 20 \mu\text{H}$, $h = 10 \mu\text{s}$)

7.5.1 HIL-Simulation R-L-Glied

In Abbildung 56 ist der Vergleich der realen HIL-Simulation zu der virtuellen numerischen HIL-Simulation (oberer Teil in Abbildung 54) dargestellt, um die Abweichung zur idealen Lösung zu beurteilen. Dabei ist die Simulationshardware nicht an die Leistungselektronik angeschlossen und der simulierte Strom wird stattdessen mit einem DAC gewandelt. Die analogen Signale werden mit einem Oszilloskop mit einer Auflösung von 11 Bit aufgezeichnet. Die Eingangsspannung entspricht einem PWM-Signal mit einer Pulsbreite von 70 % und einer Frequenz von 25 kHz. Die Amplitude der Wechselanteile und der Mittelwert des Stromes stimmen gut überein. Es sind leichte Abweichungen der Form erkennbar. Die Latenz der HIL-Simulation ist in der erwarteten Größenordnung von 385 ns und stimmt in etwa mit der virtuellen HIL-Simulation überein. (Die sichtbaren hochfrequenten Störungen sind durch einen Fehler im Platinenlayout bedingt und wirken sich nur auf den zusätzlich für die Messung angeschlossenen DAC aus.)

7.5.2 PHIL-Simulation Fensterheber

Die PHIL-Simulation wurde zur Vergleichbarkeit der durchgeführten Messungen an einem realen Fensterheber parametrisiert. Zur Ansteuerung wird für den realen und den simulierten Fensterheber die selbe Nachbildung eines Steuergerätes genutzt. Dabei wurden die Messungen direkt hintereinander durchgeführt, sodass auch andere Parameter, wie zum Beispiel der Ladezustand der 12 V Batterie gleich waren. Zur Messung der vom Steuergerät abgegebenen Spannung und des Stromes wurde ein Speicheroszilloskop mit einer maximalen Bandbreite von 1 GHz und einer Auflösung von 11 Bit genutzt. Der Strom wurde dabei mit einer Strommesszange mit einer Bandbreite von 120 MHz und einer Auflösung von 1 mA gewandelt. Die prinzipiellen Versuchsaufbauten sind in Abbildungen 57 und 58 dargestellt. Die Messdaten wurden für die Darstellung mit einem Filter

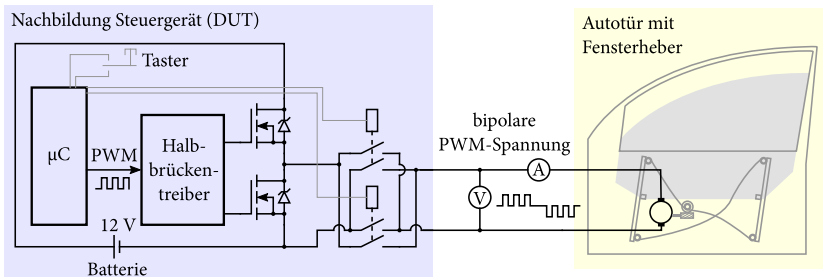


Abbildung 57: Versuchsaufbau zur Vermessung des Fensterhebers (vereinfachte Darstellung)

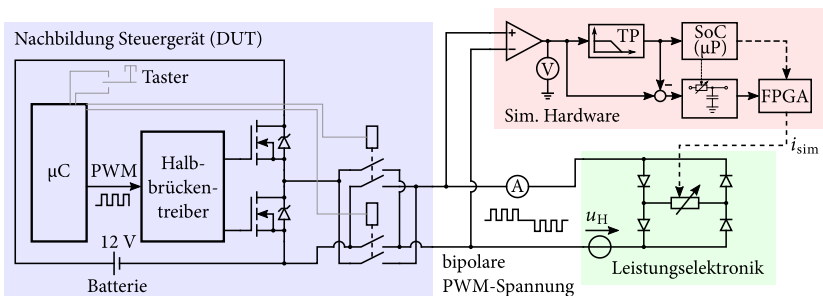


Abbildung 58: Versuchsaufbau zur Vermessung der PHIL-Simulation (vereinfachte Darstellung)

ohne Phasenverschiebung gefiltert. Dabei wurden die Filterparameter so ge-

wählt, dass die für die jeweiligen Abbildungen maßgeblichen Effekte möglichst wenig beeinflusst werden.

In Abbildung 59 sind die Ergebnisse der Simulation und der Messung für das Herunterfahren gegenüber gestellt. Da die PWM-Spannung in diesem Zeitbereich nicht sinnvoll darstellbar ist, wird eine über die Pulse gemittelte Spannung dargestellt. Es gibt zunächst eine kurze Beschleunigungsphase, in der die Pulsbreite von 60 % auf 85 % gesteigert wird. Eine Pulsbreite von 60 % wird benötigt, damit sich die Fensterscheibe in Bewegung setzt. Kurz vor Erreichen des Anschlags wird die Pulsbreite auf 100 % gesetzt. Somit liegt in diesem Bereich eine annähernd konstante Spannung an. Die größte Abweichung im Stromverlauf ist

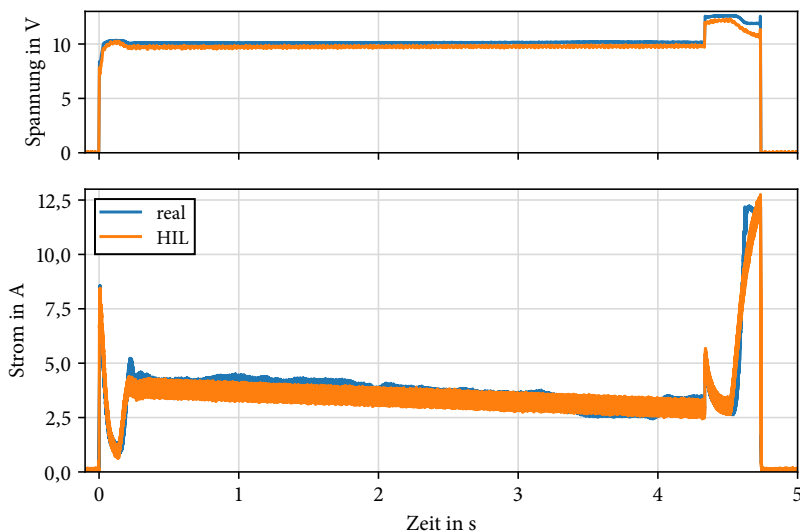


Abbildung 59: Vergleich zwischen realer Tür und PHIL-Simulation beim Herunterfahren; Spannung über PWM-Pulse gemittelt

beim Erreichen des Anschlages ab etwa 4,5 s zu erkennen. Der Anstieg des Stromes der PHIL-Simulation ist im Vergleich langsamer und weist auch qualitativ ein anderes Verhalten auf. Dies ist durch das Feder-Dämpfer-System im Modell zwischen Fenster und Getriebe bedingt. Werden die Parameter so angepasst, dass das Verhalten beim Erreichen des Anschlages besser übereinstimmt, so stimmt das Verhalten beim Anfahren schlechter überein.

Gleiches gilt für das Herauffahren in Abbildung 60. Das Erreichen des Anschlages ist außerdem signifikant verzögert. Dies resultiert hauptsächlich aus der

niedrigeren Spannung. Diese ist bei der PHIL-Simulation bei dieser Messung im Mittel um 8 % kleiner. Die Niveaus der Ströme stimmen auch in diesem Fall

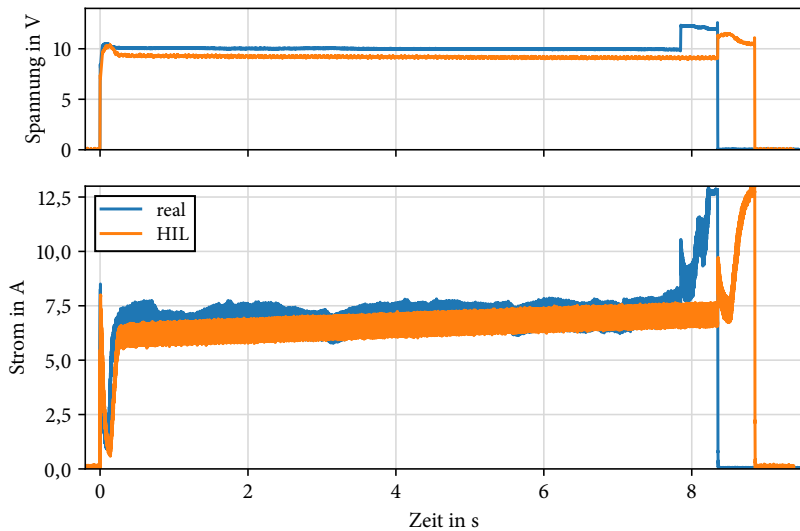


Abbildung 60: Vergleich zwischen realer Tür und PHIL-Simulation beim Herauffahren; Spannung über PWM-Pulse gemittelt

überein. Das Modell bildet die unterschiedlichen mechanischen Lasten beim Herauf- und Herunterfahren somit gut ab.

In Abbildung 61 ist ein Ausschnitt der Kommutierungsrippel zu sehen. Die Grenzfrequenz des Filters wurde mit 3 kHz so gewählt, dass die durch die PWM resultierenden Stromschwankungen unterdrückt werden. Die Amplituden der Schwingungen stimmen gut überein. Die leichte Abweichung in der Frequenz der Schwingungen ist durch eine etwas unterschiedliche Drehzahl des realen Motors und seiner Simulation bedingt.

In Abbildung 62 werden einzelne Spannungspulse der PWM und die zugehörigen Ströme dargestellt. Es ist gut zu erkennen, dass der Verlauf im Mittel qualitativ und quantitativ gut übereinstimmt. Im Gegensatz zu Abbildung 56 ist in dem dargestellten Zeitbereich keine signifikante Verzögerung zu erkennen.

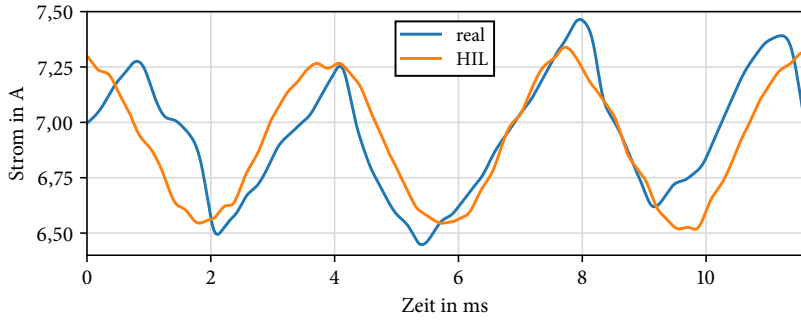


Abbildung 61: Vergleich zwischen realer Tür und PHIL-Simulation (Kommutierungsrippel);
Ströme über PWM-Pulse gemittelt

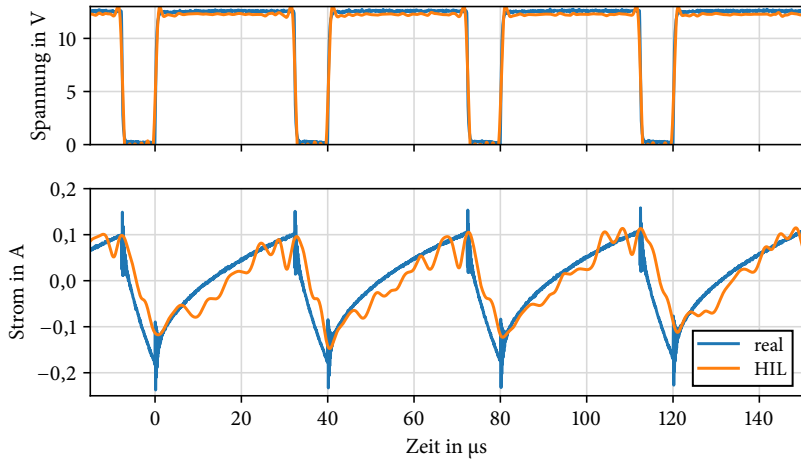


Abbildung 62: Vergleich zwischen realer Tür und PHIL-Simulation (PWM)
(Wechselanteil des Stroms)

8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

Immer mehr und immer aufwendigere Steuer- und Regelalgorithmen in automobilen Steuergeräten erfordern bessere Testmöglichkeiten bei der Entwicklung. Hierfür bieten sich HIL-Simulationen an, da diese mit wenig Aufwand einzusetzen sind und unter anderem reproduzierbare Ergebnisse liefern. Einige der Steuer- und Regelalgorithmen nutzen zeitlich hochaufgelöste Strommessungen des jeweiligen Verbrauchers. Daraus resultiert eine hohe zeitliche Anforderung an die HIL-Simulation, wenn zum Beispiel Kommutierungsrippel von Elektromotoren mit hoher Drehzahl und die Stromschwankungen aufgrund von einer PWM-Spannung abgebildet werden sollen. Auch moderne leistungsfähige CPUs, auf welchen eine einfache Umsetzung von Simulationsmodellen möglich ist, können diesen Anforderungen nicht gerecht werden. Zwar gibt es FPGA basierte Lösungen, welche die zeitlichen Anforderungen erfüllen, allerdings ist es mit einem hohen Aufwand verbunden Simulationsmodelle auf diesen zu integrieren. Auch die Entwicklung von geeigneten Simulationsmodellen und deren Parametrierung ist aufwendig.

Deshalb wird ein Konzept für eine HIL-Architektur entwickelt, das die zeitlichen Anforderungen bei gleichzeitig einfacher Implementierung von Simulationsmodellen ermöglicht. Diese basiert auf einer parallelen Simulation, bei der hochfrequente Spannungssignale mit einem stark vereinfachten Modell analog simuliert werden. Mit den niederfrequenten Signalen wird ein komplexes Modell auf einer gewöhnlichen CPU simuliert. Dafür wird die von dem Steuergerät abgegebene Spannung mit einer Frequenzweiche auf die beiden Simulationspfade aufgeteilt. Die beiden Simulationsergebnisse werden mit einem FPGA zusammengeführt und an eine Leistungselektronik ausgegeben, die den simulierten Strom einstellt.

Um eine gute Qualität der Simulation zu erreichen, ist es eine Grundvoraussetzung, dass das Eingangssignal sinnvoll mit einer Frequenzweiche auf die parallelen Simulationspfade aufgeteilt wird. In Abschnitt 4 werden zunächst die Eigenschaften des Eingangssignals und die genauen Anforderungen für die gefilterten Ausgangssignale für die parallelen Simulationen ermittelt. Es wird gezeigt, dass sich unterschiedliche Anforderungen an die Filterschaltung widersprechen. Darauf basierend werden zunächst unterschiedliche Möglichkeiten der Realisierung untersucht und bewertet. Dabei wird gezeigt, dass der übliche Aufbau einer Frequenzweiche mit zwei Filtern in diesem Fall kein ausreichendes

Ergebnis liefert. Für diese spezielle Anwendung kommt eine Architektur mit nur einem Filter und einer Differenzbildung zum Einsatz, welche einen guten Kompromiss in Bezug auf die Anforderungen erreicht. Für diese Architektur werden die Filterparameter für den besten Kompromiss bestimmt.

In Abschnitt 5 wird anhand von Messung gezeigt, was für Eigenschaften die zu simulierenden Verbraucher aufweisen und welche Anforderungen sich daraus ergeben. Anhand eines Fensterhebers wird exemplarisch ein Simulationsmodell hergeleitet. Dabei werden die für das Testen wichtigen physikalischen Vorgänge mit ausreichender Qualität abbildet. Es wird gezeigt, wie alle Modellparameter mit einfachen Messungen, Berechnungen oder Abschätzung ermittelt werden können. Anhand dieser einfachen Parametrierung wird das Modell mit echten Messdaten validiert.

Für eine einfache Handhabung der HIL-Simulation wird in Abschnitt 6 eine automatische Auswahl eines geeigneten numerischen Löser und einer geeigneten Schrittweite entwickelt. Diese basiert auf einer Eigenwertanalyse des parametrisierten Modells und wird direkt vor Simulationsbeginn auf der Zielhardware ausgeführt. Diese ist auch für nicht-lineare Modelle anwendbar, indem der Zustandsraum diskretisiert wird und für alle Entwicklungspunkte eine Linearisierung durchgeführt wird. Außerdem wird eine Laufzeitanalyse des Modells auf der Zielhardware durchgeführt, um die minimale Schrittweite der implementierten Löser zu ermitteln. Dieses Verfahren ist für nicht-lineare differential-algebraische Gleichungssysteme mit verschiedenen Zuständen einsetzbar. Zusammen mit einer entwickelten Modellnotation sind somit bei der Implementierung neuer Modelle oder der Nutzung bereits vorhandener Modelle keine Kenntnisse über numerische Lösungsverfahren notwendig.

Abschließend wird in Abschnitt 7 gezeigt, dass das entwickelte Konzept mit aktuell verfügbarer Hardware realisiert werden kann. Es wird der Einfluss der zeitlichen Eigenschaften der verwendeten Komponenten auf die Simulation dargestellt. Die geforderte Schrittweite von $1\ \mu\text{s}$ wird mit einer realisierten Schrittweite von $385\ \text{ns}$ deutlich unterschritten. Beim Erfassen des Eingangssignals wird eine Genauigkeit von etwa $0,5\ \%$ bezogen auf den gesamten Eingangsbereich erreicht. Das Ausgangssignal erreicht eine Auflösung von $1\ \text{mA}$ bei einem maximalen Bereich von etwa $65\ \text{A}$. Zusammen mit einer geeigneten Leistungselektronik werden die Ströme der HIL-Simulation mit denen eines realen Fensterhebers verglichen. Hierbei zeigt sich eine gute Übereinstimmung.

8.2 Ausblick

Die minimale Schrittweite der HS-Simulation wird maßgeblich durch die verfügbaren ADCs bestimmt, wenn eine geringe Latenz eingehalten werden soll. Für eine geringe Latenz bei Auflösungen über 12 Bit kommen SAR ADCs zum Einsatz. Die Zunahme der maximalen Samplerate verläuft in den letzten Jahrzehnten in etwa linear (siehe Anhang A.7) und beträgt weniger als 5 MHz pro Dekade. Eine signifikante Reduzierung der Latenz durch den ADC ist somit in absehbarer Zeit nicht zu erwarten.

Sowohl die Signalfilterung als auch die analoge Simulation können in einer Weiterentwicklung digital in einem FPGA umgesetzt werden. Dadurch ergeben sich einige Vorteile:

- Parametrierbarkeit des Filters,
- komplexere Modelle für die HS-Simulation,
- unterschiedliche Modelle für die HS-Simulation,
- Parametrierbarkeit der HS-Simulation in einem größeren Parameterbereich,
- Reduktion von störanfälligen analoge Bauteilen,
- Verwendung von nur einem ADC.

Die heute verfügbaren Kombinationen aus FPGA und Mikroprozessor, sogenannte SoC-FPGAs, vereinen diese beide Komponenten auf einem Chip. Dadurch kann im Gegensatz zu dedizierten Komponenten eine deutlich schnellere Kommunikation zwischen den beiden Komponenten realisiert werden. So können sich weitere Vorteile durch Verringerung der Übertragungszeiten zwischen Mikroprozessor und FPGA ergeben:

- Reduktion der Latenz der LS-Simulation,
- Möglichkeit der kontinuierlichen Anpassung der Parameter der HS-Simulation während der Laufzeit durch die LS-Simulation.

A Anhang

A.1 Erklärung Abbildung: CPU Leistungsfähigkeit bei Fließkommaberechnungen

Es gibt einige Umstände, welche zu Ungenauigkeiten in den dargestellten Datenpunkten führen:

- Die erreichte Punktezahl hängt nicht nur von der verwendeten CPU ab, sondern auch von der restlichen Hardware des Testsystems. So können mit der gleichen CPU verschiedene Ergebnisse erzielt werden.
- Da der Test nur Ergebnisse für die Nutzung aller Kerne des CPUs liefert, ist die dargestellte Leistung pro Kern lediglich der Quotient aus dem *Baseline* Wert des Tests und der Anzahl der CPU Kerne des Testsystems. Dies entspricht nicht unbedingt der realen Leistung im Betrieb mit einem Kern. Dieser kann durch diverse Faktoren davon abweichen.
- Das Datum der Datenpunkte entspricht dem Wert des Eintrages *Hardware Availability* aus dem *Disclosure* Dokument des jeweiligen Systemtests. Dieses Verfügbarkeitsdatum der Hardware kann von dem Datum der Verfügbarkeit des CPUs abweichen, da dort das Datum von der am spätesten verfügbaren Hardwarekomponente im Testsystem eingetragen ist. Bei Stichproben lagen die Daten allerdings immer in der Nähe des CPU Erscheinungsdatum. Aufgrund der großen Menge an Datenpunkten wäre der Zeitaufwand einer manuellen Ermittlung der realen CPU Verfügbarkeit nicht angemessen gewesen.

Für die Erstellung von Abbildung 1 wurden die aus [54] bezogenen Werte folgendermaßen umgerechnet:

Die Basis der Punktzahl bildet der *Baseline* Wert aus den Ergebnissen der *SPEC[®]fp* Tests. Alle paar Jahre kommt eine neue Version der Testsoftware heraus. Die Punktzahl der verschiedenen Versionen ist nicht miteinander vergleichbar. Um einen genügend großen Zeitraum darzustellen, wurden Ergebnisse aus dem *SPEC[®]2000* und dem *SPEC[®]2006* Test genutzt.

Es gibt einige CPU Typen, welche mit beiden Versionen getestet wurden. Damit die Daten in einem Graphen darstellbar sind, wird der Umrechnungsfaktor

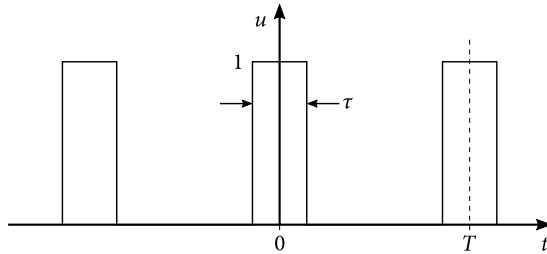
$$\alpha_{\text{SPEC}} = \frac{1}{l} \sum_{k=1}^l \left(\frac{\frac{1}{n_k} \sum_{i=1}^{n_k} B_{2000}(k, i)}{\frac{1}{m_k} \sum_{j=1}^{m_k} B_{2006}(k, j)} \right) = 172,31 \quad (89)$$

eingeführt. Es gibt k CPU Typen, welche in beiden Testversionen vorkommen. Bei jedem dieser Typen gibt es n , bzw. m Ergebnisse für diesen Typen in dem jeweiligem Test. B beinhaltet die *Baseline* Ergebnisse für den jeweiligen Test. Alle Werte werden mit dem Minimalwert normiert und die entsprechenden Werte zusätzlich skaliert. So dass sich der in der Abbildung dargestellte Punktwert

$$P_{\text{FP}} = \begin{cases} \frac{B_{2000}}{\min(B_{2000})} & \text{für } \text{SPEC}^{\circledast}2000 \text{ Test} \\ \frac{B_{2006} \cdot \alpha_{\text{SPEC}}}{\min(B_{2000})} & \text{für } \text{SPEC}^{\circledast}2006 \text{ Test} \end{cases} \quad (90)$$

ergibt.

A.2 Benötigte Schrittweite aufgrund der PWM



Die dargestellte Rechteckfunktion ist symmetrisch und kann mit der Fourierreihe

$$u(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cdot \cos\left(2\pi \cdot k \cdot \frac{t}{T}\right)$$

angenähert werden [58]. Die zugehörigen Koeffizienten sind:

$$a_0 = 2 \cdot \frac{\tau}{T}$$

$$a_k = \frac{2}{T} \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} u(t) \cdot \cos\left(\frac{2\pi \cdot k}{T} \cdot t\right) dt$$

$$\Rightarrow a_k = \frac{2}{\pi \cdot k} \cdot \sin\left(\pi \cdot k \cdot \frac{\tau}{T}\right)$$

Für $k > \frac{T}{\tau}$ sind $a_k < 0,128 \cdot a_0$ und haben somit keinen großen Einfluss auf $u(t)$. Das als Kriterium für die Grenzfrequenz angesetzt und mit

$$\frac{\tau}{T} \geq 0,01$$

ist $k \geq 100$. Mit

$$T = \frac{1}{f_{\text{PWM}}} = 40 \mu\text{s}$$

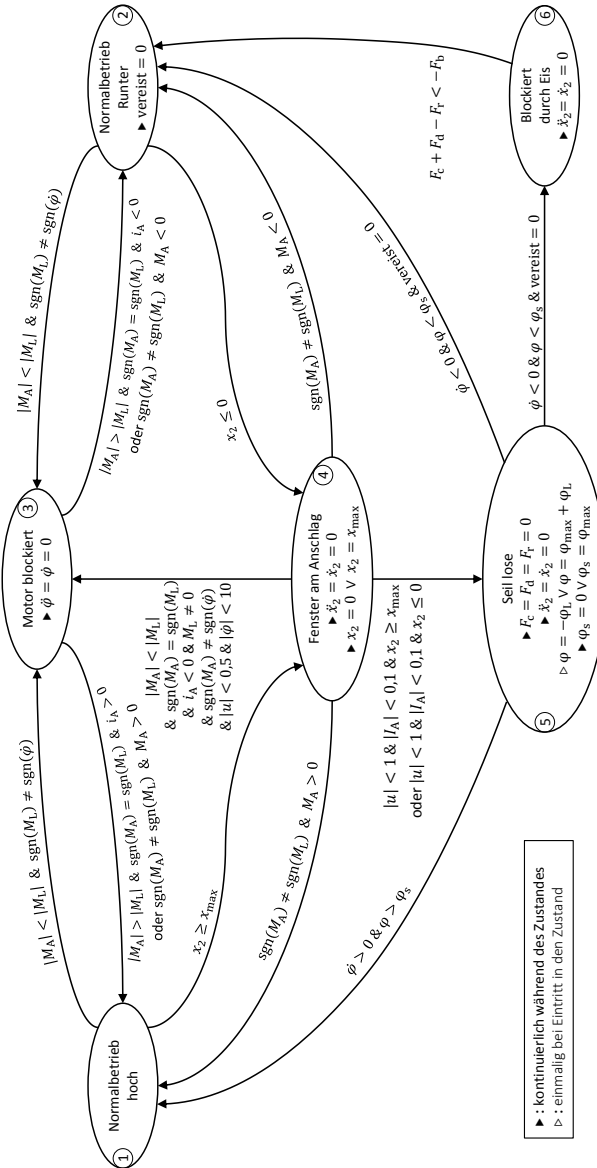
ist

$$f_{\text{Shannon}} \geq 2 \cdot \frac{k}{T} = 5 \text{ MHz}$$

und die dadurch benötigte Schrittweite ist

$$h \leq \frac{1}{f_{\text{Shannon}}} = 200 \text{ ns}$$

A.3 Zustände Fensterhebermodell



A.4 Zusammenfassung der Modellgleichungen des Fensterhebers

$$u_A - R_A \cdot i_A - L_A \cdot \dot{i}_A t - k_M \cdot \dot{\varphi} = 0$$

$$J_A \cdot \ddot{\varphi} + \frac{dr^2}{G^2 \eta_G} \cdot \dot{\varphi} + \frac{cr^2}{G^2 \eta_G} \cdot \varphi - \frac{dr}{G \eta_G} \cdot \dot{x}_2 - \frac{cr}{G \eta_G} \cdot x_2 - k_M \cdot i_A = 0$$

$$m \cdot \ddot{x}_2 + d \cdot \dot{x}_2 + c \cdot x_2 - \frac{dr}{G} \cdot \dot{\varphi} - \frac{cr}{G} \cdot \varphi + m \cdot g + F_r = 0$$

$$F_r(x_2, \dot{x}_2) = \operatorname{sgn}(\dot{x}_2) \left(F_{r,\min} + \frac{x_2}{x_{2,\max}} \cdot (F_{r,\max} - F_{r,\min}) \right)$$

$$i_r = \frac{(i_A \cdot a + 1) \cdot \hat{I}_{R\infty} \cdot \dot{\varphi}}{\sqrt{\omega_x^2 + \dot{\varphi}^2}} \cdot \sin(z \cdot \varphi)$$

$$i = i_A + i_r$$

A.5 Zusammenfassung der Modellparameter des Fensterhebers

- Oberer Abschnitt: Parameter aus den regulären Modellgleichungen
- Unterer Abschnitt: Parameter aus Zuständen (inklusive Testfällen)

Größe	Symbol	Wert	Einheit
R_A	Ankerwiderstand	0,79	Ω
L_A	Ankerinduktivität	$950 \cdot 10^{-6}$	H
k_M	Motorkonstante	0,0196	N m A^{-1}
b	Reibkoeffizient Motor	$2,76 \cdot 10^{-5}$	N m s rad^{-1}
J_A	Massenträgheitsmoment Rotor	$1,46 \cdot 10^{-5}$	kg m^2
G	Getriebeuntersetzung	73	-
η_G	Getriebewirkungsgrad	0,3	-
m	Äquivalente Fenstermasse	5	kg
r	Radius Seilrolle	0,025	m
c	Federkonstante Gesamtsystem	$45 \cdot 10^3$	N m^{-1}
d	Dämpfungsmaß Gesamtsystem	310	N s m^{-1}
z	Anzahl Lamellen	10	-
a	Stromeinflussfaktor	0,56	-
$\hat{I}_{R,\infty}$	Scheitelwert Rippelstrom	0,1	A
ω_x	Eckwinkelgeschwindigkeit	125	rad s^{-1}
$F_{r,\min}$	Minimale Reibkraft	26,5	N
$F_{r,\max}$	Maximale Reibkraft	74,5	N
x_{\max}	Maximaler Weg Fenster	0,5	m
φ_L	Winkeländerung bei Seillose	60,3	rad
F_b	Losbrechkraft Vereisung	0	N
h_B	Höhe Fremdkörper	$0 \dots x_{\max}$	m
V	Vereisung (ja/nein)	1/0	-
B	Fremdkörper (ja/nein)	1/0	-

A.6 Modellbeschreibung einer Halogenlampe

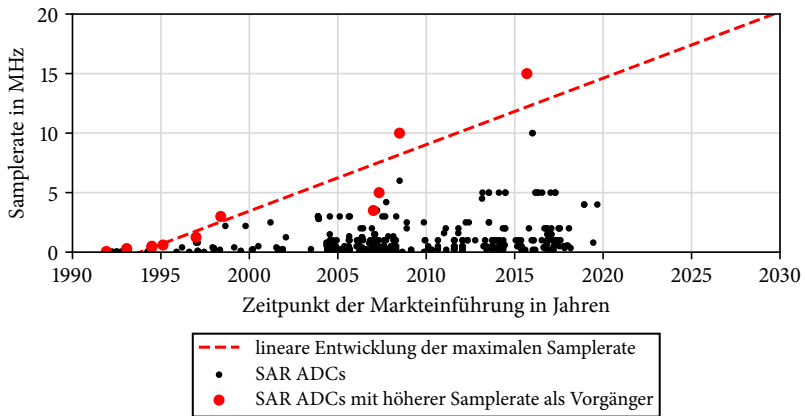
Modellgleichungen basierend auf [13, 21]:

$$F(t, x, u) = \begin{bmatrix} f_1 = \frac{di}{dt} = \frac{u}{L} - \frac{R_u}{LT_u^a} \cdot T^a i \\ f_2 = \frac{dT}{dt} = \frac{T_u^a u^2}{R_u c} \cdot T^{-a} - \frac{b}{c} (T - T_u)^4 \end{bmatrix} \quad (91)$$

$$J = \begin{bmatrix} -\frac{R_u}{LT_u^a} T^a & -\frac{R_u a}{LT_u^a} T^{a-1} i \\ 0 & -\frac{T_u^a u^2 a}{R_u c} T^{-a-1} - \frac{4b}{c} (T - T_u)^3 \end{bmatrix} \quad (92)$$

A.7 Entwicklung der Samplerate von SAR ADCs

Datenpunkte basieren auf [9].



Literatur

- [1] Altera Cooperation. *Cyclone IV EP4CE6 FPGA Datenblatt*. URL: <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-iv/cyiv-51001.pdf> (besucht am 18.10.2019) (siehe S. 86).
- [2] F. Alvarez-Gonzalez und Antonio Griffo. „High-fidelity modelling of permanent magnet synchronous motors for real-time hardware-in-the-loop simulation“. In: *8th IET International Conference on Power Electronics, Machines and Drives (PEMD 2016)* (2016), S. 6.–6. DOI: 10.1049/cp.2016.0308. URL: <http://digital-library.theiet.org/content/conferences/10.1049/cp.2016.0308> (siehe S. 9).
- [3] Fernando Alvarez-Gonzalez u. a. „Real-Time Hardware-in-The-Loop Simulation of Permanent-Magnet Synchronous Motor Drives under Stator Faults“. In: *IEEE Transactions on Industrial Electronics* 64.9 (2017), S. 6960–6969. ISSN: 02780046. DOI: 10.1109/TIE.2017.2688969 (siehe S. 9).
- [4] *Analog Devices - High Speed D/A Converters*. URL: <https://www.analog.com/en/products/digital-to-analog-converters/standard-dac/high-speed-da-converters.html> (besucht am 11.03.2019) (siehe S. 2).
- [5] Analog Devices Inc. *AD5235 Datenblatt*. 2012. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD5235.pdf> (besucht am 18.10.2019) (siehe S. 85).
- [6] Analog Devices Inc. *AD7484 Datenblatt*. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD7484.pdf> (besucht am 18.10.2019) (siehe S. 85).
- [7] Analog Devices Inc. *AD768 Datenblatt*. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD768.pdf> (besucht am 17.10.2019) (siehe S. 89).
- [8] Analog Devices Inc. *Analog Devices, Standard A/D Converters*. URL: <https://www.analog.com/en/products/analog-to-digital-converters/standard-adc.html> (besucht am 11.03.2019) (siehe S. 12).
- [9] *Analog Devices SAR ADCs*. URL: <https://www.analog.com/en/products/analog-to-digital-converters.html> (besucht am 19.12.2019) (siehe S. 106).

- [10] *Arbeiten (auch studentische) am Institut für Elektrische Energiesysteme und Automation im Rahmen der Dissertation* (siehe S. 54, 59, 66).
- [11] M. Bacic. „On hardware-in-the-loop simulation“. In: *Proceedings of the 44th IEEE Conference on Decision and Control* (2005), S. 3194–3198. DOI: 10.1109/CDC.2005.1582653 (siehe S. 1).
- [12] Günter Bärwolff. *Numerik für Ingenieure, Physiker und Informatiker*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. ISBN: 978-3-662-48015-1. DOI: 10.1007/978-3-662-48016-8. arXiv: 1011.1669. URL: <http://link.springer.com/10.1007/978-3-662-48016-8> (siehe S. 75).
- [13] Sam Ben-Yaakov, Mor Mordechai Peretz und Bryce Hesterman. „A SPI-CE compatible behavioral electrical model of a heated tungsten filament“. In: *Conference Proceedings - IEEE Applied Power Electronics Conference and Exposition - APEC 2* (2005), S. 1079–1084. DOI: 10.1109/APEC.2005.1453128 (siehe S. 40, 105).
- [14] Ottmar Beucher. *Signale und Systeme : Theorie , Simulation , Anwendung*. ISBN: 978-3-662-45964-5. DOI: 10.1007/978-3-662-45965-2 (siehe S. 15).
- [15] Roger E. Bilstein. *Flight Patterns: Trends of Aeronautical Development in the United States, 1918-1929*. Univ of Georgia Pr, 1983. ISBN: 978-0820332147 (siehe S. 1).
- [16] Andreas Binder. *Elektrische Maschinen und Antriebe*. Bd. 53. 9. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, S. 1689–1699. ISBN: 978-3-662-53240-9. DOI: 10.1007/978-3-662-53241-6. arXiv: arXiv:1011.1669v3. URL: <http://link.springer.com/10.1007/978-3-662-53241-6> (siehe S. 45).
- [17] Bosch Automotive Aftermarket. *Electric motors*. 2019. URL: https://www.bosch-ibusiness.com/media/images/products/dc_motors/xx_pdfs_2/electric_motors_catalogue_en_2019-2020_low.pdf (besucht am 29.08.2019) (siehe S. 45, 64, 86).
- [18] Wai-Kai Chen. *Passive, Active, and Digital Filters*. CRC Press, 2006. ISBN: 978-0-8493-7277-3 (siehe S. 28).
- [19] Chroma Systems Solutions. *MODEL 63600 SERIES*. URL: <http://www.chromausa.com/pdf/63600-E.pdf> (besucht am 30.11.2018) (siehe S. 12).
- [20] DSpace. *SCALEXIO*. URL: https://www.dspace.com/shared/data/pdf/2018/dSPACE_SCALEXIO_Product-information_01-2018_English.pdf (besucht am 30.10.2018) (siehe S. 10).

- [21] Marija Engelman, Tomislav Barić und Hrvoje Glavaš. „Minimum correct thermo-electric model for transient behaviour of incandescent lamp“. In: *Tehnicki Vjesnik* 21.2 (2014), S. 299–308. ISSN: 1330-3651 (siehe S. 40, 41, 105).
- [22] Rolf Fischer. *Elektrische Maschinen*. München: Carl Hanser Verlag GmbH & Co. KG, 2013. ISBN: 978-3-446-43813-2. DOI: 10 . 3139 / 9783446437678. URL: <http://www.hanser-elibrary.com/doi/book/10.3139/9783446437678> (siehe S. 45).
- [23] Glöckler M. *Simulation mechatronischer Systeme - Grundlagen und technische Anwendung*. 2014. ISBN: 9783658053833 (siehe S. 80).
- [24] Julius Harms. „Parameterabhängige Schrittweitensteuerung für eine HIL-Simulation zur Prüfung automobiler Steuergeräte“. Masterthesis. Technische Universität Hamburg, 2018. DOI: 10 . 15480 / 882 . 2066. URL: <https://tore.tuhh.de/handle/11420/2070> (siehe S. 82).
- [25] Tobias Heidrich. „Kommutierungsberechnung bei permanentmagneterregten Kommutatormotoren kleiner Leistung“. In: (2012), S. 117 (siehe S. 55, 57).
- [26] Microchip Technology Inc. *Datenblatt SAM S70 Family*. 2019. URL: <http://ww1.microchip.com/downloads/en/DeviceDoc/SAM%20E70S70V70V71-Family%20-%2060001527D.pdf> (besucht am 27.02.2019) (siehe S. 5).
- [27] *Intel Product Specifications*. URL: <https://ark.intel.com/> (siehe S. 8).
- [28] Intepro Systems. *EL 2000 HS Series*. 2016. URL: <https://www.inteproate.com/wp-content/uploads/2016/08/EL-2000-HS-1.3.pdf> (besucht am 30.11.2018) (siehe S. 12).
- [29] R. Isermann, J. Schaffnit und S. Sinsel. „Hardware-in-the-loop simulation for the design and testing of engine-control systems“. In: *Control Engineering Practice* 7.5 (1999), S. 643–653. DOI: [https://doi.org/10.1016/S0967-0661\(98\)00205-6](https://doi.org/10.1016/S0967-0661(98)00205-6). URL: <http://www.sciencedirect.com/science/article/pii/S0967066198002056> (siehe S. 1).
- [30] Behzad Jandaghi und Venkata Dinavahi. „Hardware-in-the-Loop Emulation of Linear Induction Motor Drive for MagLev Application“. In: *IEEE Transactions on Plasma Science* 44.4 (2016), S. 679–686. ISSN: 00933813. DOI: 10.1109/TPS.2016.2535460 (siehe S. 9).

- [31] Zheming Jin u. a. „Euro-Par 2017: Parallel Processing“. In: 10417 (2017), S. 664–675. DOI: 10.1007/978-3-319-64203-1. URL: <http://link.springer.com/10.1007/978-3-319-64203-1> (siehe S. 7).
- [32] Dennis Kähler, Günter Ackermann und Horst Brinkmeyer. „Novel Method for Multi-scale Hardware-in-the-Loop Simulation of Automotive Mechatronic Components“. In: *2018 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*. IEEE, 2018, S. 1352–1357. ISBN: 978-1-5386-4941-1. DOI: 10.1109/SPEEDAM.2018.8445299. URL: <https://ieeexplore.ieee.org/document/8445299/> (siehe S. 11, 89).
- [33] Keysight Technologies. *N3300 Series - DC Electronic Loads*. 2017. URL: <https://literature.cdn.keysight.com/litweb/pdf/5980-0232E.pdf?id=1000070562:epsg:dow> (besucht am 30. 11. 2018) (siehe S. 12).
- [34] Kikusui America. *PLZ-4WL Series*. 2012. URL: <https://www.kikusui.co.jp/common/product/pdf/plz-4wl.pdf> (besucht am 30. 11. 2018) (siehe S. 12).
- [35] Siegfried H. Linkwitz. „Passive Crossover Networks for Noncoincident Drivers“. In: *AES: Journal of the Audio Engineering Society* 26.3 (1978), S. 149–150 (siehe S. 31).
- [36] W.C. Lo u. a. „Acoustic noise radiated by PWM-controlled induction machine drives“. In: *IEEE Transactions on Industrial Electronics* 47.4 (2000), S. 880–889. ISSN: 0278-0046. DOI: 10.1109/41.857968 (siehe S. 2, 16).
- [37] MDL Technologies. *Model 6330A Series*. URL: <https://www.mdltechnologies.co.uk/technical-docs/Model-6330A-Series-High-Speed-DC-Electronic-Load.pdf> (besucht am 30. 11. 2018) (siehe S. 12).
- [38] Martin Meyer. *Signalverarbeitung*. 8. Aufl. Wiesbaden: Springer Fachmedien Wiesbaden, 2017. ISBN: 978-3-658-18320-2. DOI: 10.1007/978-3-658-18321-9. URL: <http://link.springer.com/10.1007/978-3-658-18321-9> (siehe S. 37).
- [39] Umar Ibrahim Minhas, Roger Woods und Georgios Karakonstantis. „Exploring Functional Acceleration of OpenCL on FPGAs and GPUs Through Platform-Independent Optimizations“. In: Bd. 10824. Springer International Publishing, 2018, S. 551–563. ISBN: 978-3-319-78889-0. DOI: 10.1007/978-3-319-78890-6_44. URL: http://link.springer.com/10.1007/978-3-319-78890-6_44 (siehe S. 7).

- [40] Sameer Mojlish u. a. „Review of Hardware Platforms for Real-Time Simulation of Electric Machines“. In: 3.1 (2017), S. 130–146 (siehe S. 9).
- [41] Frank Ohnhäuser. *Analog-digital converters for industrial applications including an introduction to digital-analog converters*. 2015, S. 1–333. ISBN: 9783662470206. DOI: 10.1007/978-3-662-47020-6 (siehe S. 12).
- [42] ON Semiconductor. *TND6073/D - Motor Control and Diagnostics for Automotive Adaptive Front-Lighting System (AFS)*. 2018 (siehe S. 18).
- [43] Opal RT. *Hypersim*. URL: <https://www.opal-rt.com/systems-hypersim/> (besucht am 30.10.2018) (siehe S. 10).
- [44] Tarek Ould-Bachir u. a. „Effective floating-point calculation engines intended for the FPGA-based HIL simulation“. In: *IEEE International Symposium on Industrial Electronics* (2012), S. 1363–1368. ISSN: 2163-5137. DOI: 10.1109/ISIE.2012.6237289 (siehe S. 9).
- [45] Aiken Pang und Peter Membrey. *Beginning FPGA: Programming Metal*. Apress, 2017. ISBN: 978-1-4302-6247-3. DOI: 10.1007/978-1-4302-6248-0. URL: <http://link.springer.com/10.1007/978-1-4302-6248-0> (siehe S. 6).
- [46] Marcel J.M. Pelgrom. *Analog-to-Digital Conversion*. New York, NY: Springer New York, 2013. ISBN: 978-1-4614-1370-7. DOI: 10.1007/978-1-4614-1371-4. URL: <http://link.springer.com/10.1007/978-1-4614-1371-4> (siehe S. 12).
- [47] Nileshchandra K. Pikle, Shailesh R. Sathe und Arvind Y. Vyavhare. „GPGPU-based parallel computing applied in the FEM using the conjugate gradient algorithm: a review“. In: *Sadhana - Academy Proceedings in Engineering Sciences* 43.7 (2018), S. 1–21. ISSN: 09737677. DOI: 10.1007/s12046-018-0892-0. URL: <https://doi.org/10.1007/s12046-018-0892-0> (siehe S. 7).
- [48] Raspberry Pi Foundation. *Raspberry Pi 3 Model B+ Datenblatt*. 2015. URL: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf> (besucht am 18.10.2019) (siehe S. 85).
- [49] *RÖMPP Online-Enzyklopädie*. URL: <https://roempp.thieme.de/roempp4.0/> (siehe S. 64).
- [50] Arnold Rosenberg und Alan Sussman. *Topics in Parallel and Distributed Computing*. 2016. ISBN: 9783319931081. DOI: 10.1016/c2015-0-00519-6 (siehe S. 7).

- [51] RTDS Technologies. *Real Time Power System Simulation*. URL: <https://www.rtds.com/real-time-power-system-simulation/> (besucht am 30.10.2018) (siehe S. 10).
- [52] Dietrich Schlichthärle. *Digital Filters*. 2. Aufl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. ISBN: 978-3-642-14324-3. DOI: 10.1007/978-3-642-14325-0. URL: <http://link.springer.com/10.1007/978-3-642-14325-0> (siehe S. 33, 36).
- [53] Thomas Schulte und Jörg Bracker. „Real-time simulation of BLDC motors for hard-ware-in-the-loop applications incorporating sensorless control“. In: *IEEE International Symposium on Industrial Electronics* (2008), S. 2195–2200. DOI: 10.1109/ISIE.2008.4677088 (siehe S. 10).
- [54] *SPEC Test*. URL: <https://www.spec.org/> (besucht am 15.02.2019) (siehe S. 8, 99).
- [55] Karl Strehmel, Rüdiger Weiner und Helmut Podhaisky. *Numerik gewöhnlicher Differentialgleichungen*. Bd. 53. 9. 2012, S. 1689–1699. ISBN: 978-3-8348-1847-8. DOI: 10.1007/978-3-8348-2263-5. arXiv: arXiv:1011.1669v3. URL: <http://link.springer.com/10.1007/978-3-8348-2263-5> (siehe S. 23, 75).
- [56] Nikolaos Tzaniis u. a. „FPGA-Assisted Distribution Grid Simulator“. In: Bd. 10824. Springer International Publishing, 2018, S. 640–646. ISBN: 978-3-319-78889-0. DOI: 10.1007/978-3-319-78890-6_51. URL: http://link.springer.com/10.1007/978-3-319-78890-6_51 (siehe S. 7).
- [57] Lutz v. Wangenheim. *Aktive Filter und Oszillatoren*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. ISBN: 978-3-540-71737-9. DOI: 10.1007/978-3-540-71739-3. URL: <http://link.springer.com/10.1007/978-3-540-71739-3> (siehe S. 85).
- [58] Hubert Weber. *Laplace-Transformation*. Bd. 53. 9. Wiesbaden: Vieweg Teubner, 2007, S. 1689–1699. ISBN: 978-3-8351-0140-1. DOI: 10.1007/978-3-8351-9104-4. URL: <http://link.springer.com/10.1007/978-3-8351-9104-4> (siehe S. 101).
- [59] *Wikipedia Siliciumdioxid*. URL: <https://de.wikipedia.org/wiki/Siliciumdioxid> (besucht am 07.08.2019) (siehe S. 64).

- [60] Steven Xie. *Successive-Approximation ADCs: Ensuring a Valid First Conversion*. URL: <https://www.analog.com/en/analog-dialogue/articles/successive-approximation-adcs.html> (besucht am 11. 03. 2019) (siehe S. 12).
- [61] Vittorio Zanetti. „Temperature of incandescent lamps“. In: *American Journal of Physics* 53.6 (1985), S. 546–548. issn: 0002-9505. doi: 10.1119/1.14410 (siehe S. 81).
- [62] Xin Zhao. „A Contribution to the Computer Aided Development of an Electrical Window Regulator System“. Dissertation. TU Ilmenau, 2015. ISBN: 9783863601218 (siehe S. 47, 63).

$$J_A \cdot \ddot{\varphi} + \frac{dr^2}{G^2 \eta_G} \cdot \dot{\varphi} + \frac{cr^2}{G^2 \eta_G} \cdot \varphi -$$

$$m \cdot \ddot{x}_2 + d \cdot \dot{x}_2 + c \cdot x_2 - \frac{dr}{G}$$

$$F_r(x_2, \dot{x}_2) = \operatorname{sgn}(\dot{x}_2) \left(F_{r, \min} \right)$$