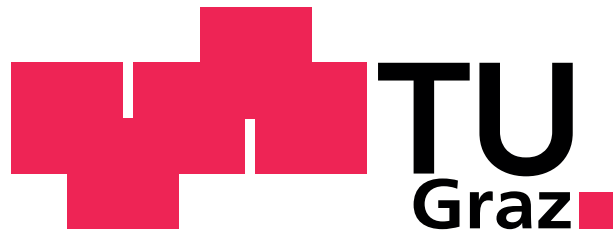


Technische Universität Graz  
Institut für Regelungs- und Automatisierungstechnik



Kompetenzzentrum - Das Virtuelle Fahrzeug Forschungsgesellschaft mbH  
Area E - Vehicle Electrics / Electronics and Software



Masterarbeit

# Steuerung und Regelung eines Rollenprüfstandes

Martin Kirchengast

17. Oktober 2012

Gutachter: Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Martin Horn  
Betreuer: Petr Micek, MSc. und Dipl.-Ing. Dr. techn. Allan Tengg



# Kurzfassung

Die vorliegende Masterarbeit befasst sich mit der Momenten- und Drehzahlregelung eines elektrischen Rollenprüfstandes im Maßstab 1:5, der zur Simulation von unterschiedlichen Belastungssituationen des zugehörigen Elektrofahrzeuges dient. Messdatenerfassung, Steuerung und Kommunikation erfolgen jeweils über einen Mikrocontroller, auf dem die entworfenen Algorithmen implementiert wurden. Im ersten Teil der Arbeit wird die verwendete Softwarestruktur erläutert, in welche die zeitdiskret entworfenen Regler integriert wurden. Steuerung und Auswertung der am Prüfstand durchgeführten Versuche werden von einem PC aus bewerkstelligt. Die Kommunikation zwischen PC, Prüfstand und Fahrzeug erfolgt über ein Flexray-Netzwerk. Für dessen einfache Nutzung war die Implementierung einer Softwareschnittstelle in einer Programmbibliothek notwendig. Mit deren Hilfe wurde auch eine Integration des Rollenprüfstandes in Matlab/Simulink realisiert. Den nächsten Schritt bildete die Erstellung und Parametrierung eines mathematischen Modells des Prüfstandes. Das entworfene nichtlineare Modell berücksichtigt die Besonderheiten der Leistungselektronik der Gleichstrommaschine und diente als Grundlage für den Reglerentwurf. Der Hauptteil der Arbeit befasst sich mit dem Entwurf von unterschiedlichen Regelungsalgorithmen für Strom und Drehzahl. Während zur Stromregelung ein einschleifiger Regelkreis ausreichend ist, wird zur Geschwindigkeitsregelung eine kaskadierte Struktur benutzt und dadurch die Einhaltung der Strombegrenzung des Motors gewährleistet. Die Momentenregelung kann auf einfache Weise auf eine Stromregelung zurückgeführt werden. Diese Vorgehensweise hat den Vorteil, dass ein und derselbe Regler zur Momenten- und Drehzahlregelung verwendet werden kann. Aufgrund teils großer Unsicherheiten bei der Bestimmung der Systemparameter bot es sich nach dem Entwurf von Standard-PI-Reglern an, auf robuste Algorithmen zurückzugreifen. Dabei fiel die Wahl auf zwei Formen von Second Order Sliding Mode Control: den *Sub-Optimal Algorithm* und den *Super Twisting Algorithm*. Beide wurden für Strom- und Geschwindigkeitsregelung herangezogen und die Resultate aus Simulationen und Versuchen am realen Prüfstand verglichen. Wegen der begrenzten Ressourcen des 8-Bit Mikrocontrollers war es nicht möglich, eine niedrigere Abtastzeit als eine Millisekunde zu realisieren, weshalb der lineare Regler für die Stromregelung eindeutig die bessere Wahl ist. In der Simulation konnte jedoch demonstriert werden, dass bei kürzeren Abtastperioden die Sliding Mode Regler eine bessere Performance als ein vergleichbarer PI-Regler liefern. Betreffend die Geschwindigkeitsregelung zeigte sich, dass der Super Twisting Regler selbst bei den vorliegenden, schwierigen Verhältnissen am realen Prüfstand hinsichtlich Anregelzeit und Störgrößenunterdrückung dem PI-Regler überlegen ist.



# Abstract

This master's thesis deals with torque- and speed control of an electric dynamometer on a scale of 1:5 which is used for simulation of different load conditions of an electric car. Measurement data acquisition, control and communication is done on a microcontroller on which all developed algorithms were implemented. The first part of this thesis describes an appropriate software structure which enables the integration of the developed discrete-time controllers. Since control and data analysis of the experiments on the dynamometer are performed on a PC, the definition and implementation of a software interface for communicating over a Flexray network had to be done. By the means of that software library it was possible to integrate the dynamometer control into Matlab/Simulink. The next step was the development and parameter identification of a mathematical model representing the dynamometer. This nonlinear model especially takes into account the specific characteristics of the power electronics controlling the DC motor and is utilized as a testbed for the designed controllers. The main portion of this work is concerned with the development of multiple control algorithms for motor current and speed. While for the purpose of current control one single control loop is sufficient, a cascaded structure is better suited to handle constraints on the maximum motor current during speed control. The torque control problem can easily be reduced on current control. The advantage of this approach is the possibility to use the same controller for torque regulation and inside the current loop of the cascaded speed control. Due to significant uncertainties of some plant parameters the application of robust control techniques seemed to be a reasonable choice after finishing the design of industrial standard PI controllers. Two methods belonging to the class of second order sliding mode control were chosen: the *Sub-Optimal Algorithm* and the *Super Twisting Algorithm*. Both were used as current and speed controllers and their performance was compared in simulations and in experiments carried out on the real dynamometer. Because of the limited resources of the 8-bit microcontroller it was not possible to reach smaller sampling times than one millisecond. Hence the linear current controller was obviously superior on the dynamometer. In simulation it was demonstrated that the sliding mode controllers outperform comparable PI controllers at higher sampling rates. In case of speed control the Super Twisting Algorithm provides better rise time and disturbance rejection despite the difficult conditions on the real dynamometer.



### **Eidesstattliche Erklärung**

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am \_\_\_\_\_  
Ort, Datum

\_\_\_\_\_  
Unterschrift

### **Statutory Declaration**

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, \_\_\_\_\_  
Place, Date

\_\_\_\_\_  
Signature





# Inhaltsverzeichnis

<b>Kurzfassung</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Abkürzungsverzeichnis</b>	<b>xi</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Co-Simulation . . . . .	2
1.3 Funktionalität und Projektübernahmestatus . . . . .	2
1.3.1 Übernahmestatus . . . . .	2
1.3.2 Probleme . . . . .	2
1.4 Ziele . . . . .	3
1.5 Gliederung . . . . .	4
<b>2 Mikrocontroller Architektur</b>	<b>5</b>
2.1 Übersicht . . . . .	5
2.2 Mikrocontroller . . . . .	6
2.3 Software . . . . .	6
2.3.1 Fast Task . . . . .	7
2.3.2 Zeitverhalten . . . . .	8
<b>3 Kommunikation</b>	<b>9</b>
3.1 Anforderungen an die Kommunikationsschnittstelle . . . . .	9
3.2 Flexray . . . . .	9
3.2.1 Buszugriff . . . . .	9
3.3 Übertragungsformat . . . . .	10
3.3.1 Fehlercodes . . . . .	11
3.3.2 Betriebsmodi und Befehle . . . . .	12
3.4 Software-Interface des PCs . . . . .	12
3.4.1 DynamometerLibrary . . . . .	13
<b>4 Matlab/Simulink-Anbindung</b>	<b>15</b>
4.1 Erweiterung der Matlab-Funktionalität . . . . .	15
4.2 Erweiterung der Simulink-Funktionalität . . . . .	15
4.2.1 Zeitliche Synchronisierung . . . . .	16
4.2.2 Simulink Anbindung . . . . .	17
<b>5 Modellbildung</b>	<b>19</b>
5.1 Permanentmagneterregte Gleichstrommaschine . . . . .	19
5.1.1 Aufbau . . . . .	19
5.1.2 Momentenerzeugung . . . . .	19

5.1.3	Gegenspannung . . . . .	20
5.1.4	Ersatzschaltung . . . . .	21
5.2	Gleichstromsteller . . . . .	22
5.2.1	Tiefsetzsteller . . . . .	22
5.2.2	Vierquadrantsteller . . . . .	23
5.3	Zahnriemengetriebe und Rollen . . . . .	26
5.4	Modell des Prüfstandes . . . . .	26
5.4.1	Modellierung der Sperrdioden . . . . .	26
<b>6</b>	<b>Sensorik</b>	<b>29</b>
6.1	Strommessung . . . . .	29
6.2	Winkelgeschwindigkeitsmessung . . . . .	29
6.2.1	Periodendauermessung . . . . .	30
6.2.2	Frequenzmessung . . . . .	30
6.2.3	Vergleich der Quantisierungsfehler beim Einsatz am Prüfstand . . . . .	30
6.2.4	Implementierung . . . . .	31
6.2.5	Drehrichtungserkennung . . . . .	32
<b>7</b>	<b>Parameteridentifikation</b>	<b>33</b>
7.1	Motorkonstante . . . . .	33
7.1.1	Versuchsbeschreibung . . . . .	33
7.1.2	Ergebnisse . . . . .	33
7.2	Reibungsmoment . . . . .	34
7.2.1	Versuchsbeschreibung . . . . .	34
7.2.2	Ergebnisse . . . . .	35
7.3	Ankerwiderstand . . . . .	35
7.3.1	Versuchsbeschreibung . . . . .	35
7.3.2	Ergebnisse . . . . .	36
7.4	Induktivität und Trägheitsmoment . . . . .	37
7.4.1	Vorgangsweise . . . . .	37
7.4.2	Ergebnisse . . . . .	37
7.5	Simulation . . . . .	37
7.5.1	Lineares Motormodell . . . . .	38
7.5.2	Nichtlineares Modell . . . . .	38
<b>8</b>	<b>Theoretische Grundlagen</b>	<b>41</b>
8.1	PID-Regler . . . . .	41
8.1.1	Anti-Windup . . . . .	41
8.1.2	Zeitdiskrete Realisierung . . . . .	42
8.2	Kaskadenregelung . . . . .	43
8.3	Symmetrisches Optimum . . . . .	44
8.3.1	Einstellung beim offenen Kreis mit zwei Integralanteilen . . . . .	44
8.3.2	Einstellung bei ungünstigem Zeitkonstantenverhältnis . . . . .	46
8.4	Sliding Mode Control . . . . .	47
8.4.1	First Order Sliding Mode Control (1-SMC) . . . . .	47
8.4.2	Second Order Sliding Mode Control (2-SMC) . . . . .	51

<b>9</b>	<b>Implementierung Strom- und Momentenregelung</b>	<b>57</b>
9.1	I-Stromregler . . . . .	57
9.1.1	EMK-Kompensation . . . . .	58
9.1.2	Entwurf . . . . .	58
9.2	Verbesserung der Stromregelung . . . . .	59
9.3	2-SMC Algorithmen . . . . .	61
9.3.1	Sub-Optimal Algorithm . . . . .	62
9.3.2	Super Twisting Algorithm . . . . .	62
9.4	Momentenregelung . . . . .	63
<b>10</b>	<b>Implementierung Geschwindigkeitsregelung</b>	<b>65</b>
10.1	Kaskadenstruktur mit PI-Regler . . . . .	65
10.1.1	Entwurf . . . . .	65
10.2	2-SMC Algorithmen . . . . .	67
10.2.1	Sub-Optimal 2-SMC . . . . .	67
10.2.2	Super Twisting 2-SMC . . . . .	68
<b>11</b>	<b>Ergebnisse</b>	<b>69</b>
11.1	Strom- / Momentenregelung . . . . .	69
11.1.1	I-Regler . . . . .	69
11.1.2	Second Order Sliding Mode Regler . . . . .	70
11.2	Geschwindigkeitsregelung . . . . .	75
11.2.1	PI-Kaskadenregelung . . . . .	75
11.2.2	Second Order Sliding Mode Regler . . . . .	76
11.3	Fahrzyklus . . . . .	79
11.4	Zusammenfassung . . . . .	80
11.5	Mögliche zukünftige Erweiterungen . . . . .	81
	<b>Literaturverzeichnis</b>	<b>83</b>



# Abkürzungsverzeichnis

1-SMC .....	First Order Sliding Mode Control
2-SMC .....	Second Order Sliding Mode Control
ADC .....	Analog to Digital Converter
API .....	Application Programming Interface
CAN .....	Controller Area Network
COM .....	Component Object Model
ECU .....	Electronic Control Unit
EMK .....	Elektromotorische Kraft
FIBEX .....	Field Bus Exchange Format
HILCOS .....	Hardware In The Loop and Co-Simulation
I/O .....	Input/Output
IC .....	Integrated Circuit
JTAG .....	Joint Test Action Group
MEX .....	Matlab Executable
MIMO .....	Multiple Input Multiple Output
MOSFET .....	Metal Oxide Semiconductor Field Effect Transistor
PWM .....	Pulsweitenmodulation
RISC .....	Reduced Instruction Set
SRAM .....	Static Random Access Memory
TDMA .....	Time Division Multiple Access
UART .....	Universal Asynchronous Receiver Transmitter
VSC .....	Variable Structure Control



# 1 Einleitung

## 1.1 Motivation

Diese Masterarbeit ist Teil des Projektes HILCOS, dessen Ziel die Durchführung einer Hardware Integration in eine Co-Simulations Plattform ist. Die Hardware besteht dabei aus einem Rollenprüfstand und einem Fahrzeug, beide aufgebaut im Maßstab 1:5. Durch Aufschalten eines Antriebs- oder Bremsmomentes auf die Rollen ermöglicht der Prüfstand die Simulation verschiedener Belastungssituationen des Fahrzeuges. Dazu gehören beispielsweise Steigungen der Fahrbahn, Gefälle oder der während der Fahrt auftretende Luftwiderstand. Auf diese Weise können genormte Fahrzyklen und Teststrecken unter reproduzierbaren Bedingungen abgefahren werden. Parallel zu den Versuchen am Prüfstand läuft eine (Co-)Simulation, deren Ergebnisse am Ende mit den realen Messwerten des Prüfstandes verglichen und validiert werden können.

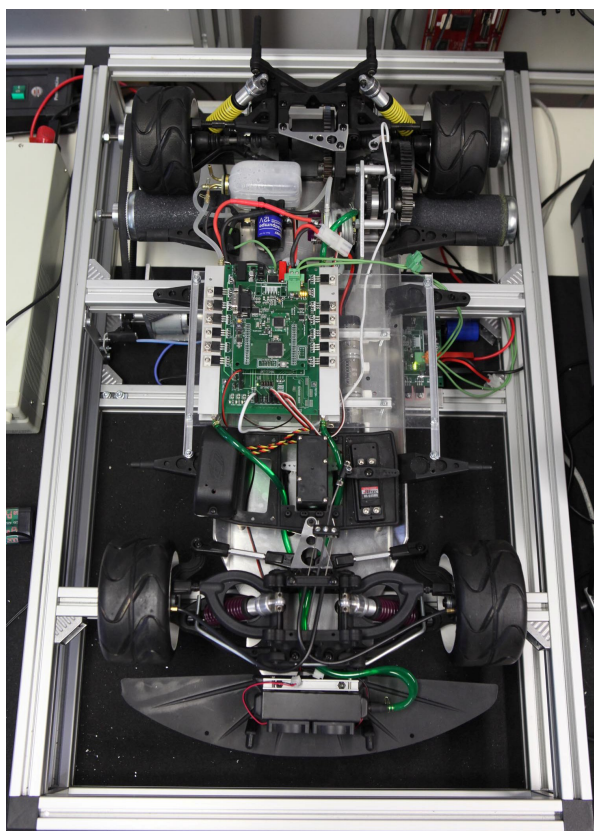


Abbildung 1.1: Rollenprüfstand und Fahrzeug von oben.

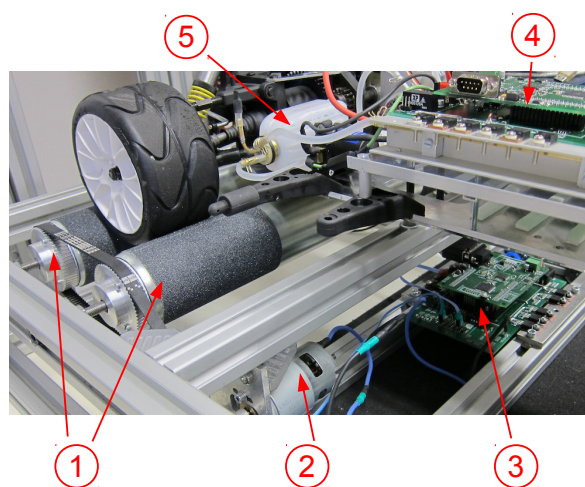


Abbildung 1.2: Seitenaufnahme des Rollenprüfstands:

- 1: Rollen mit Haftbeschichtung
- 2: Elektromotor
- 3: Steuerelektronik des Prüfstands
- 4: Steuerelektronik des Fahrzeuges
- 5: Kühlmittel tank

## 1.2 Co-Simulation

Simulationssoftware wird durch den zunehmenden Komplexitätsgrad technischer Systeme immer mehr zu einem unverzichtbaren Bestandteil deren Entwicklungsprozesses. Durch ihren Einsatz ist es möglich, das Verhalten komplexer Maschinen am Rechner nachzustellen und so etwaige Fehler frühzeitig zu erkennen. Innerhalb der zahlreichen Ingenieursdisziplinen haben sich im Laufe der Jahre die unterschiedlichsten Simulationsprogramme etabliert, jedes mitunter hochspezialisiert auf den jeweiligen Anwendungsbereich. Die mit der steigenden Komplexität einhergehende wachsende Verflechtung der Disziplinen erfordert auch die gemeinsame Simulation der oftmals zahlreichen Teilmodelle, aus denen sich das Gesamtsystem zusammensetzt. Dies kann nun entweder durch geschlossene Simulation mit verteilter Modellierung oder durch verteilte Simulation mit verteilter Modellierung geschehen. Der erste der beiden Ansätze sieht vor, dass die Modellierung der Teilsysteme zwar mit Hilfe der jeweils darauf spezialisierten Software durchgeführt wird, die Simulation an sich jedoch nur von einem Programm vorgenommen wird. Beim zweiten Ansatz hingegen ist jedes der Modellierungswerkzeuge selbst an der Berechnung der Lösung beteiligt, dies bezeichnet man als *Co-Simulation* oder auch *Co-Integration* (aufgrund der verteilten Lösung der Gleichungen). Die beteiligten Programme kommunizieren während des Simulationsablaufes über definierte Schnittstellen miteinander und sind auf diese Weise in der Lage, die berechneten Größen auszutauschen. Die zeitliche Synchronisierung der Software spielt hierbei eine entscheidende Rolle [Geimer u. a. (2006)].

## 1.3 Funktionalität und Projektübernahmestatus

Sowohl der Rollenprüfstand als auch das Fahrzeug werden von einer permanentmagneterregten Gleichstrommaschine angetrieben. Die Versorgung selbiger geschieht auf Seiten des Prüfstandes mit einem Gleichspannungsnetzgerät und beim Auto mittels Akkus. Die elektrischen Maschinen können im Vierquadrantbetrieb eingesetzt werden, beim Fahrzeug ist ein Rückspeisen von Bremsenergie in die Akkus möglich (*Rekuperation*). Das Fahrzeug ist weiters mit einem Wasserkühlkreislauf ausgestattet, um auch thermische Aspekte behandeln zu können. Auto und Prüfstand sind mit separater Steuer-, Mess- und Kommunikationselektronik ausgestattet, deren Herzstück ein *ATmega128* Mikrocontroller der Firma Atmel bildet.

### 1.3.1 Übernahmestatus

- Mechanischer und elektrischer Aufbau abgeschlossen
- Steuerung und Regelung über State-Machine und PI-Regler implementiert
- Kommunikation über Flexray möglich

### 1.3.2 Probleme

Wie bereits erwähnt war ein geregelter Betrieb des Prüfstandes bereits möglich, dabei zeigten sich jedoch einige Probleme:

#### 1. Überschwingen

Sowohl die Geschwindigkeitsregelung des Autos als auch die Momentenregelung des Prüfstandes tendierten zu starkem Überschwingen. Dies zeigte sich insbesondere bei schnellen und großen Lastwechseln.



## 2. langsame Regelung

Zusätzlich zur naturgemäß beschränkten Stellgröße der Gleichstrommaschinen ist auch auf eine Strombegrenzung zu achten, da zu hohe Ströme zu einer starken Erwärmung führen. Um diese Strombegrenzung einzuhalten, waren die Regler eher langsam eingestellt.

## 3. Drehzahlmessung

Die Drehzahl wurde mit einer selbstgefertigten optischen Messvorrichtung gemessen, die aus einer Gabellichtschranke und einer Lochscheibe bestand und 16 Impulse pro Umdrehung lieferte. Die Ausführung war einkanalig und erlaubte somit keine Feststellung der Drehrichtung.

## 4. Strommessung

Die Strommessung wurde mittels Spannungsmessung an einem Shunt-Widerstand durchgeführt. Um die Verluste gering zu halten, war der Widerstand sehr klein, was sich negativ auf die Genauigkeit der zugehörigen Spannungsmessung auswirkte. Weiters erlaubte die Position des Messpunktes keine vorzeichenbehaftete Messung und führte zu beträchtlichen Messfehlern des Betrages.

## 5. Wartbarkeit

Die Software war unzureichend strukturiert und daher nicht wartbar.

# 1.4 Ziele

Die nun folgenden Zielsetzungen wurden zu Beginn dieses Projektes definiert:

**Kommunikationsschnittstelle** Die Kommunikationsschnittstelle zwischen Rollenprüfstand, Fahrzeug und PC ist zu definieren und zu implementieren. Dabei soll auf Wiederverwendbarkeit und Modularisierung besonderer Wert gelegt werden.

**Anbindung an Matlab / Simulink** Die beiden Geräte sollen während einer laufenden Simulation von Simulink aus gesteuert werden können.

**Einsatz neuer Sensoren** Ein neuer Drehgeber und ein neuer Strommess-IC sind am Prüfstand in Betrieb zu nehmen.

**Modellbildung** Für den Rollenprüfstand ist ein dynamisches Modell zu erstellen. Anschließend sind die zugehörigen Parameter zu identifizieren.

**Rollenprüfstandsregelung** Der Prüfstand soll im geschwindigkeits- und im momentengeregelten Betrieb verwendet werden können. Es sind jeweils zwei Regelungsalgorithmen zu vergleichen.

**Fahrzeug** Die Software des Autos soll ebenfalls neu strukturiert und an die neue Kommunikationsschnittstelle angepasst werden. Eine Geschwindigkeitsregelung des Autos soll möglich sein.

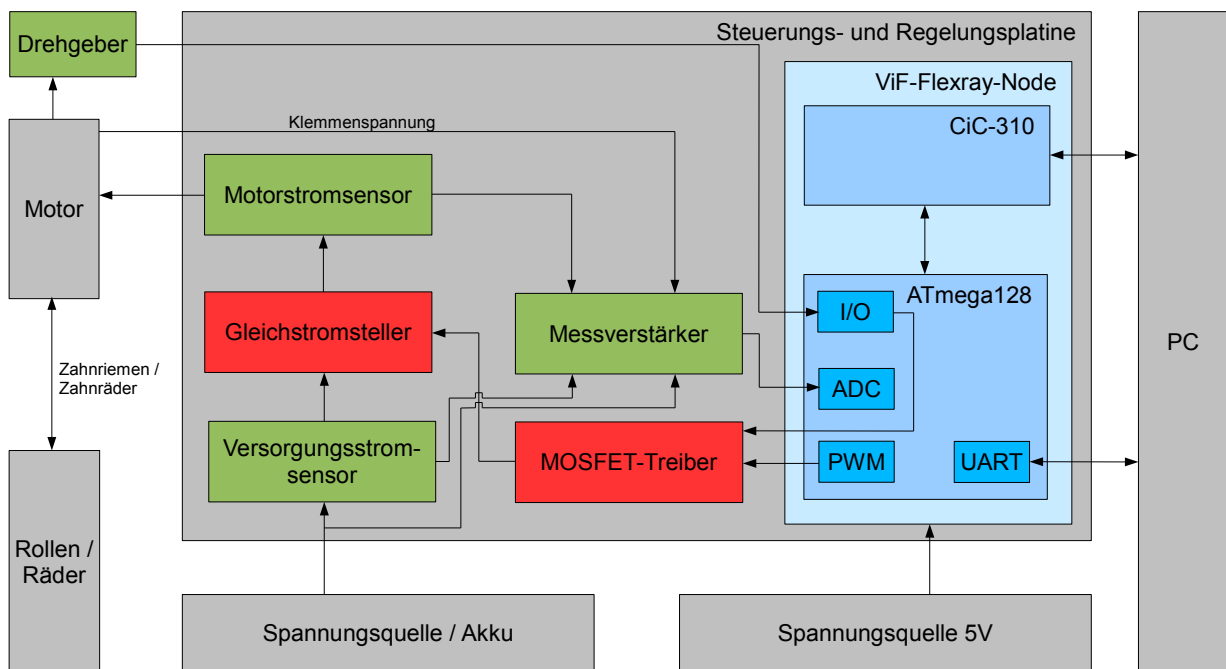
## 1.5 Gliederung

Der erste Abschnitt (Kapitel zwei bis vier) der Arbeit befasst sich in erster Linie mit Themen aus dem Bereich der Softwareentwicklung. Er behandelt die Kommunikationsschnittstelle sowie die Strukturierung der Mikrocontroller Software. Anschließend widmet sich der zweite Abschnitt, der aus den Kapiteln fünf bis sieben besteht, der Modellierung des Systemverhaltens. Dazu werden zunächst kurz grundlegender Aufbau und Funktionsweise der Gleichstrommaschine sowie deren Ansteuerung dargestellt. Darauf aufbauend wird ein Gesamtmodell erstellt und dessen Parameter durch geeignete Versuche bestimmt. Weiters werden zwei mögliche Messverfahren für die Drehzahl erläutert und Überlegungen zur Messgenauigkeit angestellt. Der letzte Abschnitt befasst sich mit der Regelung des Prüfstandes, wobei zuerst die theoretischen Grundlagen der zum Einsatz kommenden Verfahren beschrieben werden. Danach folgen Details zur Implementierung und eine Gegenüberstellung der erzielten Ergebnisse.

## 2 Mikrocontroller Architektur

### 2.1 Übersicht

Der Hardwareaufbau, der für Prüfstand und Fahrzeug weitgehend übereinstimmt, konnte größtenteils übernommen werden, nur für die neuen Sensoren mussten geringfügige Modifikationen vorgenommen werden. Abbildung 2.1 zeigt schematisch das Zusammenspiel der wichtigsten Hardwarekomponenten.



**Abbildung 2.1:** Darstellung der wichtigsten Hardwarekomponenten von Prüfstand und Auto als Blockdiagramm.

Die Software für Steuerung, Regelung und Kommunikation läuft am *ATmega128* Mikrocontroller, der den Kern des ViF-Flexray-Nodes bildet. Im Normalbetrieb findet die Kommunikation ausschließlich über Flexray statt, nur zu Debugging-Zwecken und zum Einstellen der Reglerparameter kann die Nutzung der UART-Schnittstelle mitunter hilfreich sein. Über den Gleichstromsteller kann der Motor in allen vier Quadranten betrieben werden. Der zwischengeschaltete Treiber-IC stellt ein ausreichend schnelles Umschalten der Leistungstransistoren sicher, wie es beim Betrieb mit PWM (Frequenz:  $\approx 7.8$  kHz) nötig ist. Der Motorstromsensor misst den gesamten durch den Motor fließenden Strom, der Versorgungsstromsensor nur den Teil, der aus der Spannungsquelle entnommen (zur Erläuterung siehe Abschnitt 5.2) bzw. im Falle des Fahrzeuges im Rekuperationsbetrieb zurückgespeist wird. Über die Messverstärkerschaltungen werden die unterschiedlichen Messgrößen in einen vom ADC des Mikrocontrollers erfassbaren Spannungsbereich gewandelt. Die beiden Signale des Drehgebers können direkt von den digitalen Eingangspins des *ATmega128* verarbeitet werden. Die Kraftübertragung vom Motor auf die

beiden Rollen bzw. die Räder erfolgt über einen Zahnriemen bzw. ein Zahnradgetriebe.

## 2.2 Mikrocontroller

Der *ATmega128* ist ein 8-Bit Mikrocontroller basierend auf einer RISC Architektur. Am ViF-Flexray-Node wird er mit seiner maximalen Taktfrequenz von 16 MHz betrieben, wobei der überwiegende Teil der Instruktionen nur einen Taktzyklus zur Ausführung benötigt. Als Arbeitsspeicher stehen 4 KB SRAM zur Verfügung, Programm- und Datenspeicher sind als 128 KB Flash ausgeführt. Die Programmierung erfolgt über ein JTAG -Interface. Die Verarbeitung der Messsignale geschieht über einen 10-bit ADC mit acht Eingangskanälen oder über die digitalen Eingangs-Pins bzw. die externen Interrupt-Eingänge. Insgesamt stehen jeweils zwei 8-Bit und 16-Bit Timer zur Verfügung, die auch zur Erzeugung des PWM-Signals für die Motoransteuerung in der Lage sind [Atmel Corporation (2011)].

## 2.3 Software

Die am Mikrocontroller laufende Software ist naturgemäß simpel aufgebaut. Nach einem Reset werden zunächst die Kommunikationsmechanismen UART und Flexray initialisiert.

Anschließend erfolgen die ADC-, Timer- und I/O Konfiguration des Microcontrollers. Sobald die Flexray-Kommunikation problemlos funktioniert, startet die Programmhauptschleife, die bis zum nächsten Reset oder zu einem möglichen Kommunikationsfehler ausgeführt wird. Innerhalb dieser Hauptschleife werden die Kommunikationsdaten gelesen und geschrieben, sowie die beiden Tasks mit der zuvor konfigurierten Frequenz ausgeführt (siehe Abbildung 2.2). Zunächst werden die Eingabedaten vom Flexray-Netzwerk gelesen und verarbeitet. Der *Fast Task* beinhaltet die gesamte Messsignalverarbeitung, Steuerung und Regelung der elektrischen Maschine sowie gegebenenfalls die UART-Datenübertragung. Er wird mit einer Frequenz von  $f_{Tf} = 1KHz$  ausgeführt. Diese kann leider nicht mehr weiter angehoben werden, da die Rechenleistung des ATmega128 dafür nicht ausreicht. Im nächsten Schritt werden die Messwerte zur Übertragung vorbereitet. Sekundlich läuft im Anschluss daran der *Slow Task*, der für die Temperatur- und Versorgungsspannungsüberwachung zuständig ist. Das Setzen der Flags, die die Ausführung der beiden Tasks aktivieren, geschieht über den Interrupt-Handler eines der beiden 8-Bit Timer. Die Flexray-Interrupts werden vom CiC-310 alle fünf Millisekunden ausgelöst und führen ebenfalls zum Setzen eines Flags. In der Hauptschleife erfolgt daraufhin der eigentliche Datentransfer mit dem Flexray-Controller.

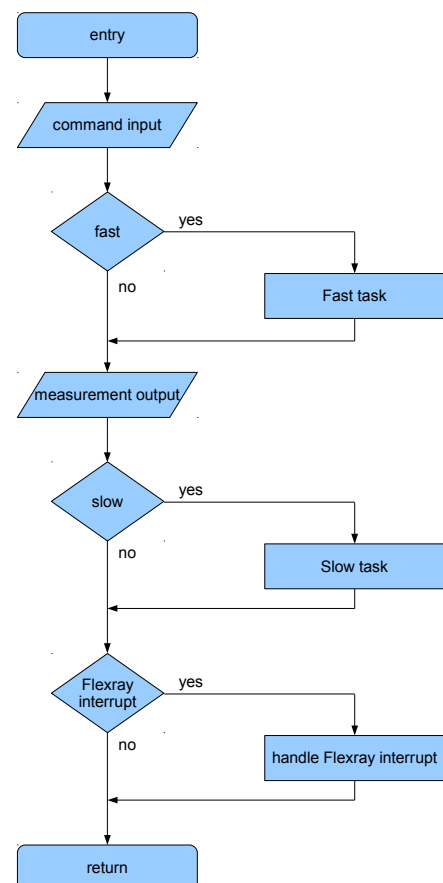


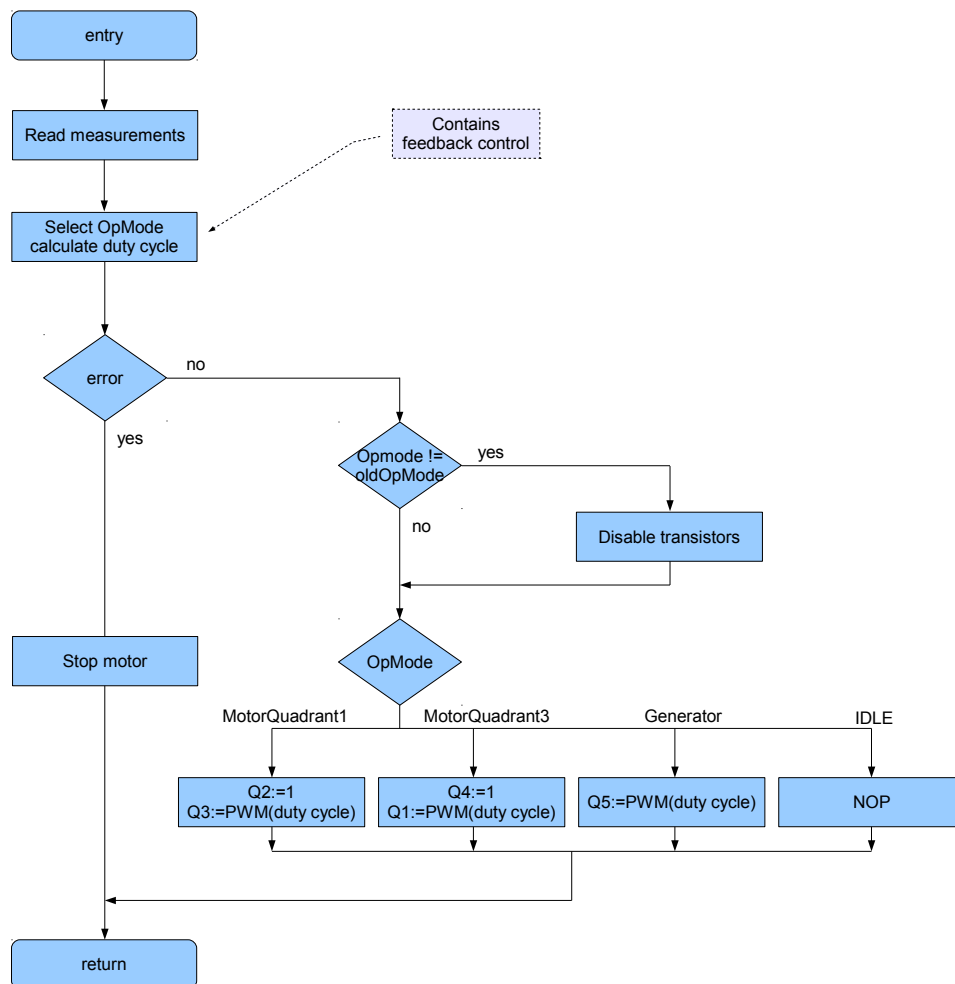
Abbildung 2.2: Inhalt der Hauptschleife des Microcontrollers.

### 2.3.1 Fast Task

Hier werden die über Flexray erhaltenen Kommandos ausgewertet und gegebenenfalls einer der Regler aktiviert. Diese liefern neben dem *Duty Cycle* für die PWM-Ansteuerung der Transistoren auch den Betriebsmodus der elektrischen Maschine, der folgende Werte annehmen kann:

- MODE\_MOTOR\_QUADRANT1: Motorischer Betrieb in positive Richtung<sup>1</sup>
- MODE\_MOTOR\_QUADRANT3: Motorischer Betrieb in negative Richtung
- MODE\_GENERATOR: Dieser Modus dient ausschließlich zum Bremsen, hier wird der Versorgungsspannungsquelle keine Energie entnommen.
- MODE\_IDLE: Der Stromkreis des Motors ist getrennt, folglich treibt er weder an, noch bremsst er.

Einen Überblick über die wichtigsten Abläufe innerhalb des *Fast Task* zeigt Abbildung 2.3.

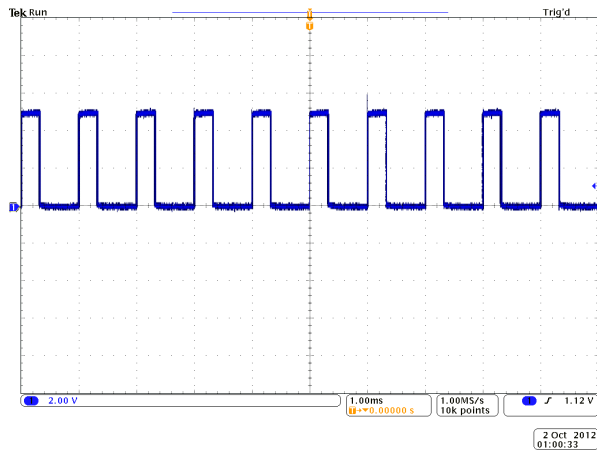


**Abbildung 2.3:** Programmablaufplan des Fast Task: Die Regelungsalgorithmen sind in „calculate duty cycle“ gekapselt. Im Falle eines Fehlers wird der Motor sofort angehalten. Um einen Kurzschluss der Versorgungsspannung auszuschließen, werden die Transistoren der Motoransteuerung (siehe Kapitel 5.2) vor jedem Quadrantenwechsel kurz abgeschaltet.

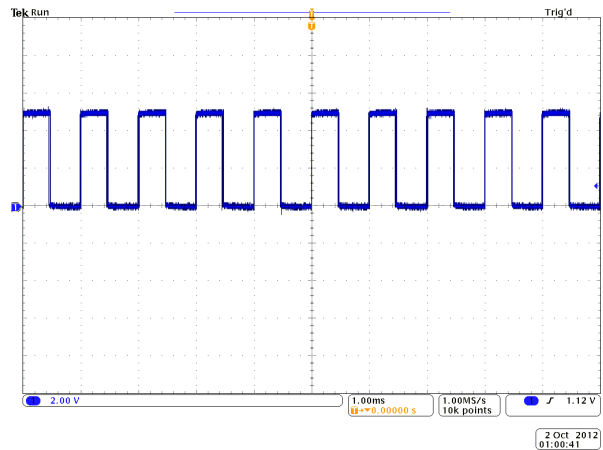
<sup>1</sup>In diesem Bereich sind definitionsgemäß Strom und Eingangsspannung ebenfalls positiv.

### 2.3.2 Zeitverhalten

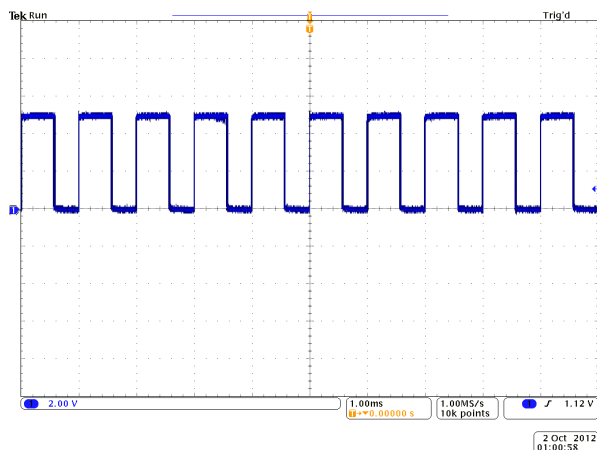
Um Abschätzungen bezüglich der minimal erreichbaren Abtastzeit durchführen zu können, wurde der Rechenzeitbedarf gemessen. Durch Aktivieren von I/O-Pins am Anfang bzw. Deaktivieren am Ende von Codepassagen konnte der Zeitverlauf am Oszilloskop gemessen werden. Die nun folgenden Messungen zeigen den Zeitbedarf des *Fast Task*, also des für die Regelung zuständigen Softwareteiles. Man erkennt die eingestellte Abtastperiode von einer Millisekunde.



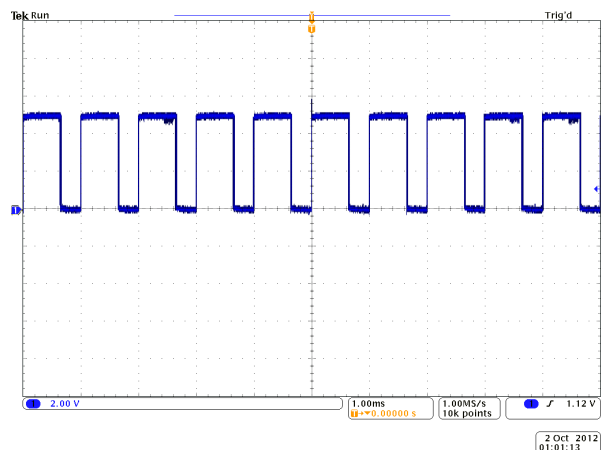
**Abbildung 2.4:** Zeitbedarf im IDLE-Betrieb:  $\approx 300\mu s$



**Abbildung 2.5:** Zeitbedarf bei Datenübertragung per UART:  $\approx 450\mu s$



**Abbildung 2.6:** Zeitbedarf im Stromregelungsbetrieb:  $550 - 600\mu s$



**Abbildung 2.7:** Zeitbedarf mit Geschwindigkeitskaskadenregelung:  $750 - 800\mu s$

Die im Anschluss an den *Fast Task* durchzuführende Behandlung des Flexray-Interrupts dauert etwa 150 Mikrosekunden, wobei innerhalb eines Kommunikationszyklus (siehe Abschnitt 3.2) zwei Interrupts auftreten (jeweils einer für Input und Output). Diese können jedoch innerhalb von zwei Reglerzyklen ( $T_s$ ) abgearbeitet werden, solange die Ungleichung  $T_s \leq \frac{T_{flexray}}{2}$  erfüllt ist.

# 3 Kommunikation

## 3.1 Anforderungen an die Kommunikationsschnittstelle

Die Kommunikationsschnittstelle verbindet den Rollenprüfstand und das Fahrzeug mit einem PC und bietet dabei die folgende Funktionalität:

- Auslesen der aktuellen Messdaten von Prüfstand und Fahrzeug
- Vorgabe des gewünschten Betriebsmodus
- Veränderung der Reglerparameter

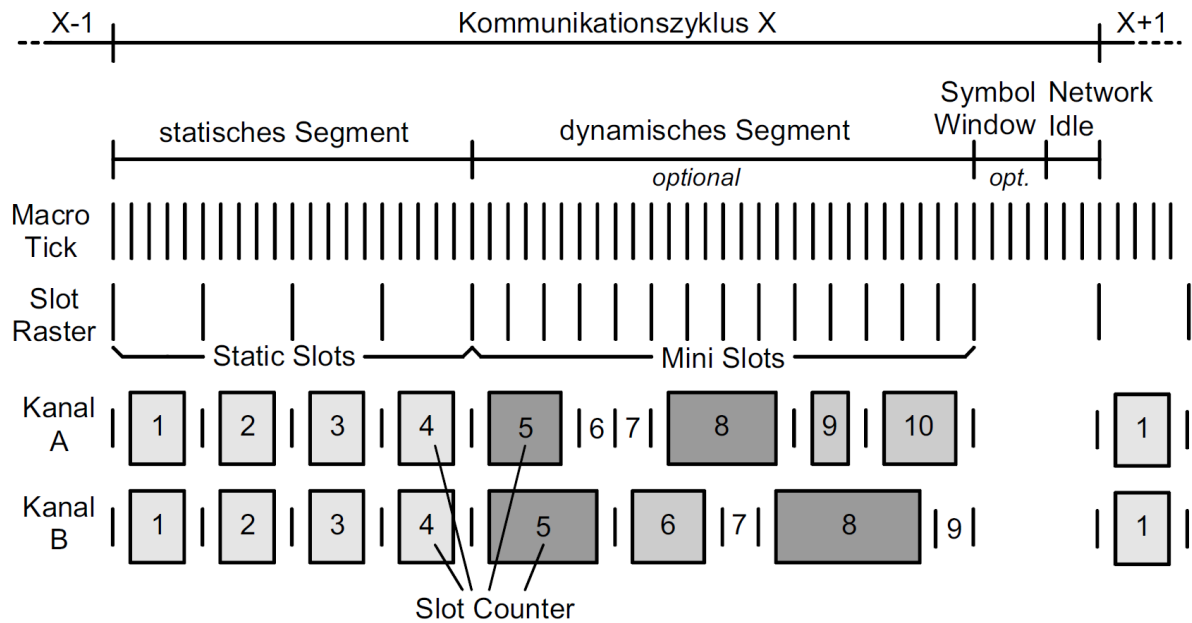
Neben der Ausgabe der aktuellen Messdaten (z.B. Motorstrom, Winkelgeschwindigkeit, Temperaturwerte, ...) kann im Fehlerfall auch ein zugehöriger Code inklusive Beschreibung ausgelesen werden. Bei der Vorgabe des Betriebsmodus besteht die Möglichkeit, zwischen manueller Steuerung unter Angabe eines PWM-Wertes und Momenten- oder Geschwindigkeitsregelung zu wählen. Wird eine Regelung gewählt, so erlaubt die Schnittstelle die Auswahl eines speziellen Regelalgorithmus und die Spezifikation der Regelgröße. Die Kommunikation zwischen den beiden ECUs und dem Steuerungs- und Auswertungs-PC erfolgt über Flexray.

## 3.2 Flexray

Während der letzten Jahre wurde mit Flexray ein speziell auf den Automobilbereich ausgerichtetes Bussystem entwickelt, um das bisher verwendete CAN allmählich zu ersetzen. Hintergrund dafür sind in erster Linie dessen Einschränkungen im Bereich der Bandbreite sowie der nicht streng deterministische Buszugriff, der die Garantie von maximalen Latenzzeiten bei der Nachrichtenübertragung erschwert. Flexray arbeitet mit maximal zwei Übertragungskanälen (A und B), deren Topologie linien- oder sternförmig aufgebaut sein kann. Der zusätzliche Kanal kann entweder zur redundanten Übertragung sicherheitskritischer Botschaften oder zur Erhöhung der Bandbreite verwendet werden. Die maximale Geschwindigkeit beträgt 10 Mbit/s pro Kanal [Zimmermann und Schmidgall (2007)].

### 3.2.1 Buszugriff

Die Datenübertragung ist in sich periodisch wiederholende Kommunikationszyklen unterteilt. Ein Kommunikationszyklus besteht wiederum aus jeweils einem statischen und einem dynamischen Segment, dem „Symbol Window“ und der „Network Idle Time“. Zur Vermeidung von Kollisionen ist das statische Segment in eine fixe Anzahl von Zeitschlitzes (sogenannten *Slots*) gegliedert, von denen jeder durch eine eindeutige Nummer (siehe Bild 3.1) gekennzeichnet ist. Während eines Zeitschlitzes besitzt eine ECU das exklusive Senderecht auf dem Bus (TDMA Prinzip). Das statische Segment dient der Übertragung zeitkritischer Informationen. Im dynamischen Segment erfolgt ebenfalls eine zeitliche Aufteilung, hier in sogenannte *Minislots*, während



**Abbildung 3.1:** Unterteilung eines Kommunikationszyklus in unterschiedliche Segmente und Zeitschlitze (Slots). [Quelle: Zimmermann und Schmidgall (2007), Seite 60]

der ein Busteilnehmer die alleinige Sendeberechtigung besitzt. Im Gegensatz zum statischen Segment darf hier eine ECU aber mehrere aufeinanderfolgende Minislots desselben Segmentes nutzen. Dadurch erfolgt eine Priorisierung von Busteilnehmern mit niedrigen Minislot-Nummern. Die Datenpakete bezeichnet man in beiden Segmenten als *Frames*, wobei sich die Übertragung eines statischen Frames genau über einen Zeitschlitz erstreckt, die eines dynamischen hingegen mehrere Minislots beanspruchen kann [Zimmermann und Schmidgall (2007)].

### 3.3 Übertragungsformat

Fahrzeug und Prüfstand verwenden ein identisches Format zum Empfang von Befehlen, das innerhalb eines Frames des statischen Segmentes Platz findet. Zur Ausgabe von Daten stehen dem Fahrzeug zwei Frames zur Verfügung, während beim Rollenprüfstand mit einem das Auslangen gefunden wird. Dies konnte durch die geringere Anzahl an Temperatursensoren und durch die Umrechnung der meisten Messwerte in Fixkommazahlen erreicht werden. Ein Frame enthält 128 Bit an Nutzdaten, sodass eingabeseitig noch weitere 80 Bit genutzt werden können und beim Fahrzeug ausgabeseitig 16 Bit im zweiten Frame ungenutzt bleiben (siehe Tabellen 3.1 bis 3.3). Der Ausgabe-Frame des Prüfstandes ist dagegen vollständig belegt. Die Periodendauer eines Kommunikationszyklus liegt bei fünf Millisekunden.

Bedeutung	Frame	Datentyp	Breite
Spannung am Motor	1	float	32 bit
Strom durch den Motor	1	float	32 bit
Versorgungsspannung	1	float	32 bit
Winkelgeschwindigkeit der Rollen bzw. Räder	1	float	32 bit
Drehmoment an Rollen bzw. Rädern oder Reglerparameter	2	float	32 bit
Versorgungsstrom	2	float	32 bit
Temperaturwerte 1 - 5	2	unsigned byte	5x8 bit
Fehlercode	2	unsigned byte	8 bit

**Tabelle 3.1:** Übertragene Daten des Fahrzeuges an den PC.



Bedeutung	Datentyp	Breite
Spannung am Motor	short	16 bit
Strom durch den Motor	short	16 bit
Versorgungsspannung	short	16 bit
Winkelgeschwindigkeit der Rollen bzw. Räder	short	16 bit
Drehmoment an Rollen bzw. Rädern oder Reglerparameter	float	32 bit
Versorgungsstrom	short	16 bit
Temperaturwert	unsigned byte	8 bit
Fehlercode	unsigned byte	8 bit

**Tabelle 3.2:** Übertragene Daten des Rollenprüfstandes an den PC.

Bei den von der ECU an den PC gesendeten Daten handelt es sich hauptsächlich um Messwerte sowie den gegenwärtigen Fehlerzustand. Die 16-Bit Variablen auf Seiten des Prüfstandes enthalten die Messwerte in Festkommadarstellung auf drei Nachkommastellen genau. Da die Variable für die Drehmomentübertragung auch zum Auslesen von Parametern verwendet wird, ist in ihrem Falle eine Beibehaltung der Gleitkommadarstellung erforderlich.

Bedeutung	Datentyp	Breite
Betriebsmodus oder Befehl	unsigned byte	8 bit
Reglernummer zur Algorithmusauswahl	unsigned byte	8 bit
Regelgröße oder Reglerparameter	float	32 bit

**Tabelle 3.3:** Übertragene Daten des PCs an die ECUs.

Die Verwendung von nur einem Frame zur Datenausgabe führt zur Verfügbarkeit von mehr Rechenzeit für die Regelung, da pro Frame ein Interrupt vom Flexray-Controller erzeugt wird.

### 3.3.1 Fehlercodes

Das Auftreten eines der folgenden Fehlerzustände führt stets zu einer Abschaltung der Motoren von Prüfstand oder Auto (wahlweise auch von beiden: *Coupled Error Mode*, Abschnitt 3.4.1.2):

Fehlercode	Bezeichnung	Bedeutung
0	ERR_NO_ERROR	keine Fehler aufgetreten
1	ERR_FLEXRAY_COM	Ausfall der Datenübertragung für länger als 3 Sekunden
2	ERR_UNKNOWN_MODE	Unbekannter Betriebsmodus oder Befehl
4	ERR_INVALID_VALUE	Ungültiger Wert für gewählten Betriebsmodus oder Befehl
8	ERR_MOSFET_HOT	Temperatur der Leistungselektronik zu hoch
16	ERR_SUPPLY_VOLTAGE	Versorgungsspannung zu niedrig oder zu hoch
32	ERR_HIGH_MOTOR_CUR	Motorstrom zu hoch
64	ERR_WATER_HOT	Kühlflüssigkeitstemperatur zu hoch (nur bei Fahrzeug möglich)
128	ERR_UNKNOWN_CONTROLLER	Ungültige Reglernummer

**Tabelle 3.4:** mögliche Fehlerzustände der ECUs

Im Falle von überschrittenen Höchsttemperaturen erfolgt ein Zurücksetzen des Fehlerzustandes, sobald das Bauteil wieder entsprechend abgekühlt ist. Bei einem zu hohen Motorstrom kann der Fehlerzustand durch Senden des entsprechenden Befehls (siehe Abschnitt 3.3.2) wieder verlassen werden.

### 3.3.2 Betriebsmodi und Befehle

Die Betriebsmodi der ECUs lassen sich grundsätzlich in zwei Kategorien einteilen: aktive und passive. Die aktiven Modi dienen der Steuerung oder Regelung der Gleichstrommaschine, während in den passiven der Motor abgeschaltet ist, sodass Konfigurationsaufgaben gefahrlos durchgeführt werden können. Außerdem besteht die Möglichkeit, die Datenübertragung über die UART-Schnittstelle zu aktivieren. Dies ermöglicht es, in jedem Zeitschritt der Regelung ausgewählte Daten zu übertragen und zu überprüfen. Der Empfang auf Seiten des PCs wurde mittels UART-zu-USB-Konverter realisiert, für den eine Software zur Aufzeichnung, Visualisierung und Speicherung im CSV-Format entwickelt wurde.

Code	Bezeichnung	Kat.	Bedeutung
1	UART_OUTPUT	passiv	UART-Übertragung aktiv
2	DIRECT_VOLTAGE_CONTROL	aktiv	Steuerung des Motors im motorischen Betrieb
4	DIRECT_BRAKE_CONTROL	aktiv	Steuerung des Motors im generatorischen Betrieb
6	FEEDBACK_SPEED_CONTROL	aktiv	Geschwindigkeitsregelung
8	FEEDBACK_TORQUE_CONTROL	aktiv	Momentenregelung (nur bei Prüfstand)
10	SET_CONTROL_PARAM	passiv	Modifikation eines Reglerparameters
12	GET_CONTROL_PARAM	passiv	Auslesen eines Reglerparameters
14	GET_CONTROL_PARAM_NUM	passiv	Auslesen der Anzahl an konfigurierbaren Reglerparametern
254	RESET_HIGH_MOTOR_CUR	passiv	Zurücksetzen des Fehlerzustandes nach zu hohem Motorstrom

**Tabelle 3.5:** Betriebsmodi der ECUs

## 3.4 Software-Interface des PCs

Am PC kommt eine Flexray-Netzwerkschnittstelle der Marke *Decomsys Busdoctor* zum Einsatz. Der mitgelieferte Treiber kann über COM in eigener Software genutzt werden. Der Datenaustausch mithilfe dieses Treibers ist innerhalb der Programmbibliothek *FlexrayLibrary* gekapselt. Das Datenformat wird mithilfe einer FIBEX-Datei spezifiziert und die Library stellt Methoden zum Lesen und Schreiben der darin definierten Variablen zur Verfügung. Darauf aufbauend bietet die *DynamometerLibrary* die Möglichkeit, auf prüfstands- und fahrzeugrelevante Daten zuzugreifen, ohne dabei eine genaue Kenntnis des Übertragungsformates zu erfordern. Sie implementiert die im vorangegangenen Abschnitt definierte Schnittstelle und führt den Datenaustausch über die *FlexrayLibrary* durch. Durch Einbindung dieser beiden Bibliotheken kann die Prüfstandssteuerung und -regelung in beliebigen anderen Programmen verwendet werden.

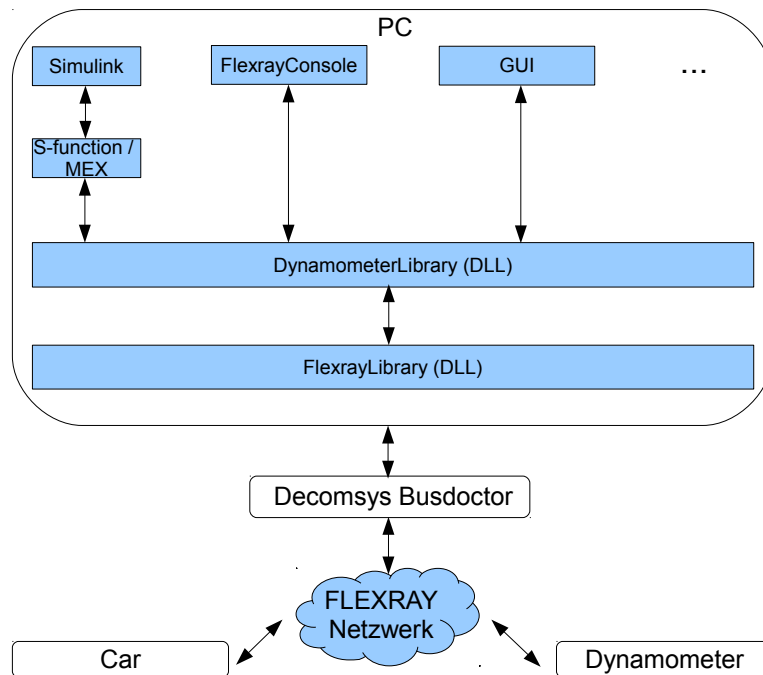


Abbildung 3.2: Interaktion der beteiligten Hardware- und Software-Komponenten

### 3.4.1 DynamometerLibrary

Bei der Implementierung wurde besonderes Augenmerk darauf gelegt, etwaige Fehlerinformationen nicht mittels Exceptions nach außen weiterzugeben, sondern ausschließlich über Funktionsrückgabewerte. Dies beschleunigt die Ausführung insbesondere unter in Simulink.

#### 3.4.1.1 Verbindungsmanagement

```

int Connect(const char *ip, const char *fibexFile, unsigned int ecuNumber);
void Disconnect();
std::vector<std::string> GetEcus();
const char *GetErrorMessage();
  
```

Über die Funktion *Connect(...)* wird der Verbindungsaufbau mit dem Flexray-Netzwerk gestartet. Der Parameter „ip“ beinhaltet die IP-Adresse des *Decomsys Busdoctor* und der Parameter „ecuNumber“ seine Kennung innerhalb der FIBEX-Konfigurationsdatei. Mittels *GetEcus()* erhält man eine Liste aller ECUs des Netzwerkes. *GetErrorMessage()* liefert eine Beschreibung des zuletzt aufgetretenen Kommunikationsfehlers.

#### 3.4.1.2 Konfiguration

```

void RegisterInput(Input input);
void SetMotorReferenceSpeedScaling(bool enable);
void SetCoupledErrorMode(bool enabled);
  
```

Über die Funktion *RegisterInput(...)* kann spezifiziert werden, welche der in den Tabellen 3.1 und 3.2 definierten Messwerte aus dem übertragenen Datenstrom ausgelesen werden sollen. Für die Messwerte sind Konstanten vom Typ *Input* in der Library definiert, die mittels Disjunktion miteinander kombiniert werden können. Die Geschwindigkeitsregler auf den ECUs erwarten als

Referenzwert die Winkelgeschwindigkeit der jeweiligen Motorwelle. Mittels *SetMotorReferenceSpeedScaling* kann man stattdessen die Umdrehungsgeschwindigkeiten von Rollen bzw. Rädern in  $\frac{m}{s}$  angeben. Die entsprechende Umrechnung wird dann automatisch innerhalb der Library durchgeführt. Durch Aktivieren des *Coupled Error Mode* erreicht man, dass im Falle eines Fehlers BEIDE Motoren abgeschaltet werden. Dies ist insbesondere bei einer aktivierten Momentenregelung des Prüfstandes sinnvoll, da dieser bei einem plötzlichen Wegfall des Fahrzeuglastmomentes unter Umständen bis zum Erreichen der maximalen Stellgröße weiterbeschleunigen würde.

### 3.4.1.3 Datentransfer

Da bei Flexray ein zeitsynchrones Buszugriffsverfahren zum Einsatz kommt [Zimmermann und Schmidgall (2007)], kann der Transfer nicht direkt mit dem Funktionsaufruf stattfinden. Stattdessen werden die Daten erst im nächsten Kommunikationszyklus vom Treiber des *Decomsys Busdoctor* übermittelt. Die *DynamometerLibrary* verfügt daher über einen eigenen Zwischenspeicher, der für den eigentlichen Datenaustausch mit dem internen Speicher des Treibers abgeglichen werden muss. Dafür stehen die folgenden Funktionen zur Verfügung:

```
int UpdateInput ();
int UpdateOutput ();
```

Mit *UpdateInput()* werden die Messwerte der ECUs vom Treiberspeicher ausgelesen. Umgekehrt schreibt *UpdateOutput()* die Kommandos und Vorgabewerte für die ECUs in den Treiberspeicher. Die nun folgenden Funktionen lesen bzw. schreiben lediglich den Zwischenspeicher der Library und erst durch den Aufruf der Update-Funktionen kommt es zum eigentlichen Datentransfer. Fahrzeug und Rollenprüfstand stellen dieselbe Schnittstelle für den Datenaustausch zur Verfügung, aus diesem Grund sind hier nur die Funktionen des Rollenprüfstandes aufgelistet.

```
float GetRollerMotorCurrent ();
float GetRollerMotorVoltage ();
float GetRollerSupplyVoltage ();
float GetRollerSupplyCurrent ();
int GetRollerTemperatureX ();
float GetRollerSpeed ();
float GetRollerTorque ();
unsigned int GetRollerInfo ();
std::string GetErrorMessage(unsigned int errorState);
float GetRequestedRollerControlParam ();
int GetRequestedRollerControlParamNum ();
void RequestRollerControlParam(unsigned char number);
void RequestRollerControlParamNum ();
void SetRollerSpeed(float speed, bool uart, unsigned char controllerNumber=0);
void SetRollerTorque(float torque, bool uart, unsigned char controllerNumber=0);
void SetRollerPWM(float value, bool brake, bool uart);
void ResetRollerHighCurrent ();
void SetRollerControlParam(unsigned char number, float value);
```

Aufgrund des zuvor beschriebenen zeitsynchronen Verhaltens kann das Auslesen der Reglerparameter nicht mit einem Funktionsaufruf erledigt werden, sondern benötigt insgesamt vier: *RequestRollerControlParam*, *UpdateOutput*, *UpdateInput* und *GetRequestedRollerControlParam*. Der Rückgabewert der Funktion *GetRollerInfo()* liefert den Status gemäß Tabelle 3.4, welcher über die Funktion *GetErrorMessage(...)* in eine Nachricht übersetzt werden kann. Die beiden Funktionen *SetRollerSpeed(...)* und *SetRollerTorque(...)* aktivieren die Regelung des Prüfstandes.

## 4 Matlab/Simulink-Anbindung

Die Anbindung der implementierten Softwareschnittstelle an Matlab/Simulink ermöglicht es, in Echtzeit Daten vom Prüfstand und Fahrzeug zu empfangen bzw. Kommandos zu schicken, während eine Simulation in Simulink abläuft. Auf diese Weise kann man beispielsweise Teile des Fahrzeugverhaltens in Simulink simulieren, während andere Aspekte mit dem realen Modellfahrzeug am Prüfstand nachgestellt werden.

### 4.1 Erweiterung der Matlab-Funktionalität

Um den Funktionsumfang von Matlab um einen bereits bestehenden Code (z.B. C oder C++) zu erweitern, bietet die Software die Möglichkeit, diesen in Form von dynamischen Programm-bibliotheken zu nutzen. Für die Verwendung unter Matlab müssen diesen Bibliotheken ein definiertes Interface implementieren und als MEX-Datei kompiliert werden. Innerhalb dieser MEX-Funktionen kann auch auf andere bereits bestehende Programmbibliotheken wie beispielsweise die *DynamometerLibrary* zugegriffen werden.

### 4.2 Erweiterung der Simulink-Funktionalität

Simulink ermöglicht die Erweiterung seiner Funktionalität mithilfe der sogenannten „s-function-Blöcke“. Dabei kann das Verhalten eines Blockes in einer der folgenden Programmiersprachen spezifiziert werden:

- Matlab Script Language
- C
- C++
- Fortran

Abgesehen von jenen in der Matlab Script Language müssen alle anderen Blockbeschreibungen als MEX-Datei kompiliert werden. Um eine Nutzung in Simulink zu ermöglichen, muss auch hier ein spezielles Interface bestehend aus diversen Callback-Funktionen implementiert werden. Diese Callback-Funktionen werden in den unterschiedlichen Schritten des Simulationsverlaufes für jeden Block aufgerufen. Die Schritte lassen sich im Wesentlichen in zwei Phasen gliedern: In der *Initialisierungsphase* werden die Blockparameter eingelesen, die Sample Time festgelegt, die Blockausführungsreihenfolge festgelegt usw. Anschließend folgt die *Simulationsschleife*, während der in der zuvor festgelegten Reihenfolge die Ausgangswerte aller Blöcke berechnet werden. Die minimale Anzahl an zu implementierenden Callbacks sowie deren zeitliche Reihenfolge ist in Abbildung 4.1 dargestellt.

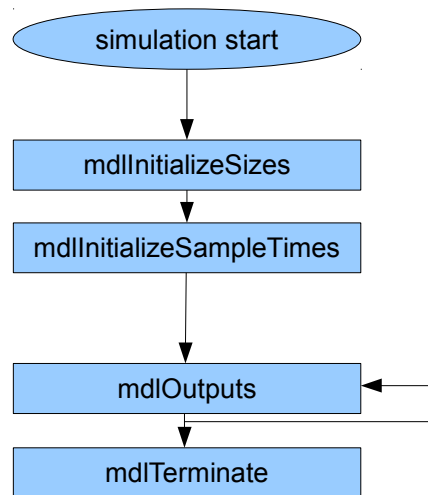


Abbildung 4.1: Aufrufreihenfolge der s-function Callbacks in Simulink

#### 4.2.1 Zeitliche Synchronisierung

Die Simulation am PC, welche die Vorgabewerte der Regler berechnet, muss in Echtzeit ablaufen, damit ein paralleler Betrieb mit Prüfstand und Auto gegeben ist. Die bevorzugte Möglichkeit dies zu erreichen, ist die Verwendung der Matlab-Erweiterung *Real-Time Windows Target*. Dabei handelt es sich um einen eigenständigen kleinen Betriebssystemkernel, der die Echtzeitausführung von Simulationen unter Windows ermöglicht. Dies wird durch eine Modifikation des Timer-Interrupt-Handlers sichergestellt. *Real-Time Windows Target* wird vom Interrupt-Handler aus in fixen zeitlichen Abständen aufgerufen und kann so den nächsten Simulationsschritt ausführen, bevor der Interrupt an Windows weitergeleitet wird. Die Simulation läuft nun in Echtzeit und ist unabhängig vom nicht deterministischen Windows Scheduler. Infolgedessen können nun von den in Echtzeit ausgeführten Programmen keine Windows Systemcalls mehr getätigt und daher auch keine Windows Hardwaretreiber verwendet werden. Für Hardware, die unter *Real-Time Windows Target* benutzt werden soll, ist ein eigens dafür konzipierter Treiber erforderlich. Allerdings steht für den *Decomsys Busdoctor* kein solcher Treiber zur Verfügung, sodass die Verwendung von *Real-Time Windows Target* nicht möglich ist. Um dieses Problem zu umgehen, besteht die Möglichkeit, die Simulation künstlich auszubremsen, damit ein simulierter Zeitschritt der tatsächlich vergangenen Zeitspanne entspricht. Die Simulation läuft in diesem Fall unter Windows und kann folglich auch auf die dort zur Verfügung stehenden Hardwaretreiber zugreifen. Aufgrund des nicht deterministischen Windows Schedulers kann jedoch keine Echtzeitausführung mehr garantiert werden. Je kleiner nämlich die gewählte Schrittweite in der Simulation ist, desto mehr wird die real vergangene Zeitspanne von dieser abweichen, da dann der Funktionsaufrufs-Overhead und die jederzeit möglichen Unterbrechungen der Simulation durch andere Windows-Programme im Rahmen des Multitaskings schwerer ins Gewicht fallen. Die Zeitmessung innerhalb des s-function-Blockes kann durch Nutzung der Timer-Funktionen in der Windows API realisiert werden. Von entscheidender Bedeutung für die Genauigkeit des Verfahrens ist die Frequenz des Windows Schedulers (*Windows Scheduler Time Slice*). Wenn dessen Periodendauer im Vergleich zur gewünschten Simulationsschrittweite zu hoch ist, kann es zu beträchtlichen Ungenauigkeiten kommen, da Simulink unter Umständen nicht genug Prozessorzeit erhält, um die gewünschte Schrittweite (in Echtzeit) einhalten zu können. Daher wird zu Beginn der Simulation die Periode des Scheduler auf eine Millisekunde (Minimum) herabgesetzt. Da eine unnötig hohe Frequenz die Systemperformance des Rechners jedoch negativ beeinflusst

(mehr Overhead durch die häufigen Kontextwechsel), wird diese Änderung am Simulationsende wieder rückgängig gemacht.

## 4.2.2 Simulink Anbindung

Um Simulink die Kommunikation mit Prüfstand und Fahrzeug zu ermöglichen, genügt die im vorangegangenen Abschnitt erwähnte minimale Anzahl von vier implementierten Callbacks. Die einzelnen Funktionen erfüllen dabei die folgenden Aufgaben:

### **mdlInitializeSizes**

- Parameter einlesen
  1. Skalierung der Referenzgeschwindigkeit (0 oder 1)
  2. Coupled Error Mode (0 oder 1)
- Verbindung zu Prüfstand und Fahrzeug herstellen
- sich für die interessierenden Messwerte registrieren
- Windows Scheduler Time Slice auf eine Millisekunde herabsetzen

### **mdlInitializeSampleTimes**

- Simulationsschrittweite einstellen
- Startzeitpunkt in realer Zeit ermitteln

### **mdlOutputs**

- Eingaben von Simulink einlesen
- Kommandos an Prüfstand und Fahrzeug senden
- Daten von Prüfstand und Fahrzeug auslesen und an Simulink weiterleiten
- warten (siehe Abschnitt 4.2.1)

### **mdlTerminate**

- Windows Scheduler Time Slice zurücksetzen
- Stopkommandos schicken
- Verbindung schließen



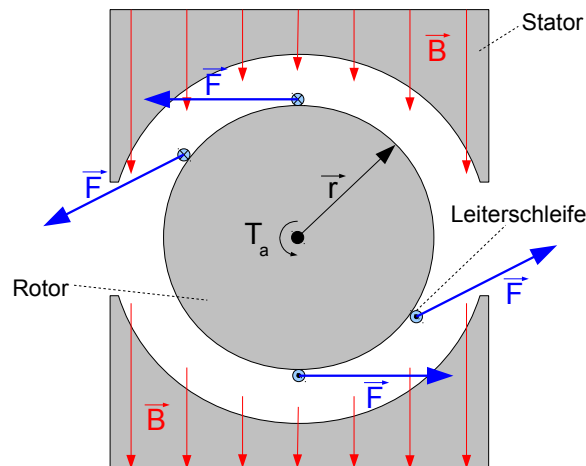


# 5 Modellbildung

## 5.1 Permanentmagneterregte Gleichstrommaschine

### 5.1.1 Aufbau

Die wichtigsten Bestandteile einer Gleichstrommaschine sind der fest stehende *Stator* (oder Ständer) einerseits und der darin drehbar gelagerte *Rotor* (oder Anker) andererseits (siehe Abbildung 5.1). Der Rotor besitzt den Radius  $r$  und die Länge  $s$  und ist von  $N$  Leiterschleifen umwickelt. Zwecks ungehinderter Drehung des Ankers befindet sich zwischen selbigem und dem Stator ein dünner Luftspalt. Bei einer elektrisch erregten Gleichstrommaschine befinden sich auch am Stator stromdurchflossene Wicklungen, um ein magnetisches Erregerfeld zu erzeugen. Im Fall der permanentmagneterregten Gleichstrommaschine entfällt dies, es kommt stattdessen ein Dauermagnet zum Einsatz [Schröder (2009a)].



**Abbildung 5.1:** Prinzipieller Aufbau einer Gleichstrommaschine: durch den permanentmagnetischen Stator entsteht eine magn. Erregung  $\vec{B}$ . Diese führt gemeinsam mit den vom Ankerstrom  $I$  durchflossenen Leitern der Länge  $s$  zur Entstehung der Kraft  $\vec{F}$ . Über den Rotorradius  $\vec{r}$  errechnet sich daraus das Luftspaltpmoment  $\vec{T}_a$ .

### 5.1.2 Momentenerzeugung

Die Ursache des erzeugten Drehmomentes ist auf die *Lorentzkraft* (Gleichung 5.1) zurückzuführen, die durch die Einwirkung eines Magnetfeldes auf einen stromdurchflossenen Leiter entsteht. Die Richtung der entstehenden Kraft ist in Abbildung 5.1 zu erkennen. Der *Kommutator* (ein Schleifkontakt) polt den Strom  $I$  um, sobald sich die Leiterschleife dem anderen Pol nähert. Dadurch wirkt die Kraft  $\vec{F}$  immer in dieselbe Richtung und der Rotor beginnt sich zu drehen [Schröder (2009a)].

$$\begin{aligned}\vec{F} &= I(\vec{s} \times \vec{B}) \\ |\vec{F}| &= I|\vec{s}||\vec{B}|\sin(\alpha)\end{aligned}\tag{5.1}$$

Ein Drehmoment entsteht durch die Kraftwirkung auf einen Hebelarm:

$$\begin{aligned}\vec{T} &= \vec{r} \times \vec{F} \\ |\vec{T}| &= |\vec{r}| |\vec{F}| \sin(\beta)\end{aligned}\tag{5.2}$$

Unter der Berücksichtigung, dass für die beiden Winkel  $\alpha = \beta = \frac{\pi}{2}$  gilt und  $N$  Leiterschleifen um den Rotor gewickelt sind, beträgt das Gesamtmoment am Anker:

$$T_a = \underbrace{(NrsB)}_{k_a} I\tag{5.3}$$

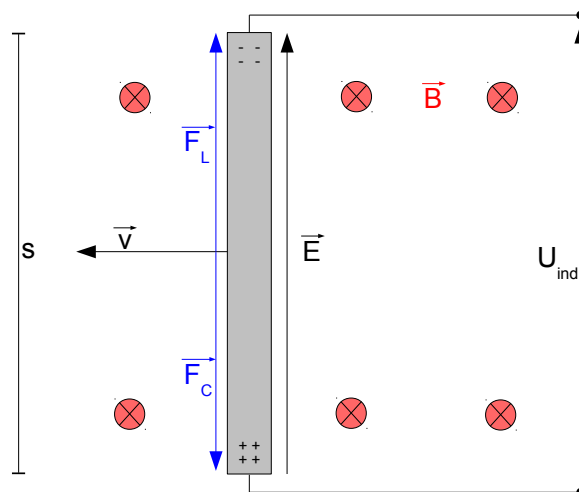
Die Größe  $k_a$  in obiger Gleichung bezeichnet man als *Motorkonstante*. Mit Hilfe des dynamischen Grundgesetzes nach Newton, dem als bekannt vorausgesetzten Trägheitsmoment  $J$  und dem anliegenden Lastmoment  $T_l$  lässt sich nun die Bewegungsdifferentialgleichung des Antriebes angeben [Schröder (2009a)]:

$$J \cdot \frac{d\omega}{dt} = T_a - T_l\tag{5.4}$$

### 5.1.3 Gegenspannung

Durch die Bewegung der Leiterschleifen im Feld der magnetischen Erregung kommt es zur Induktion einer Spannung, die der von außen angelegten Ankerspannung entgegenwirkt und somit den Stromfluss verringert. Dadurch stellt sich bei einer konstanten Eingangsspannung am Anker eine konstante Drehzahl ein. Wie in Abbildung 5.2 zu sehen ist, bewegen sich die Leiter aufgrund der Drehung des Rotors mit der Geschwindigkeit  $\vec{v}$  im magnetischen Erregerfeld  $\vec{B}$ . Die Elektronen (Ladung:  $Q = -e$ ) innerhalb des Leiters erfahren demnach gemäß der *Lorentzkraft* eine Kraftwirkung [Schröder (2009a)]:

$$\vec{F}_L = -e(\vec{v} \times \vec{B})\tag{5.5}$$



**Abbildung 5.2:** Entstehungsprinzip der Gegenspannung: Der Leiter der Länge  $s$  bewegt sich mit der Geschwindigkeit  $\vec{v}$  im Feld der magnetischen Erregung  $\vec{B}$ . Es kommt zur Ladungstrennung, eine Spannung  $U_{ind}$  wird induziert.

Es kommt folglich zu einer Ladungstrennung und zur Entstehung eines elektrischen Feldes  $\vec{E}$ , welches über die *Coulombkraft* auf die Elektronen wirkt:

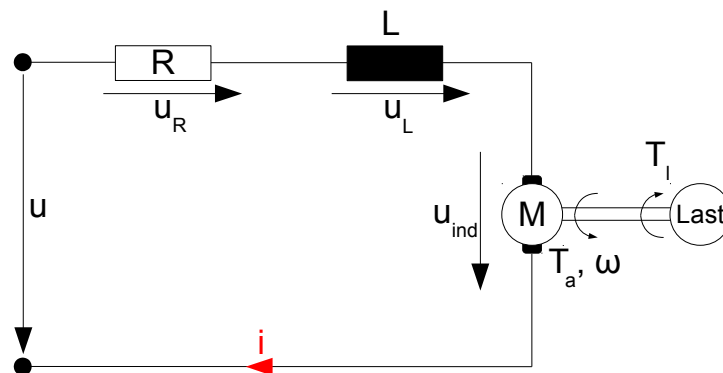
$$\vec{F}_C = -e\vec{E} \quad (5.6)$$

Im stationären Zustand heben sich die beiden Kräfte auf:  $\vec{F}_L = -\vec{F}_C$ . Für das elektrische Feld gilt daher:  $-\vec{v} \times \vec{B} = \vec{E}$ . Da  $\vec{v}$  und  $\vec{B}$  während der Drehung stets senkrecht aufeinander stehen, ergibt sich durch Integration über die  $N$  Leiter der Länge  $s$  für die induzierte Spannung:

$$\begin{aligned} U_{ind} &= NsvB \text{ mit } v = \omega r \\ U_{ind} &= \underbrace{NrsB}_{k_a} \omega \end{aligned} \quad (5.7)$$

### 5.1.4 Ersatzschaltung

Das elektrische Verhalten einer idealen permanentmagneterregten Gleichstrommaschine kann mittels Reihenschaltung des Ankerwiderstandes, der Induktivität der Spulenwicklungen und einer drehzahlabhängigen Spannungsquelle beschrieben werden (siehe Abbildung 5.3). Es gilt daher:  $u = u_R + u_L + u_{ind}$  [Schröder (2009a)].



**Abbildung 5.3:** Die Ersatzschaltung der Gleichstrommaschine besteht aus Ankerwiderstand  $R$ , der Induktivität der Wicklungen  $L$  und der aufgrund der Drehung induzierten Gegenspannung  $u_{ind}$ .

Bei der idealen permanentmagneterregten Gleichstrommaschine handelt es sich also um ein lineares System zweiter Ordnung. Die beiden Zustandsvariablen sind der Ankerstrom  $i$  und die Winkelgeschwindigkeit der Motorwelle  $\omega$ , als Eingangsgrößen fungieren die angelegte Spannung  $u$  sowie das externe Lastmoment  $T_l$ . Das Trägheitsmoment  $J$  umfasst auch die Massenträgheit der gegen den Motor wirkenden Last [Schröder (2009a)]. Meist erweitert man das System noch um ein geschwindigkeitsproportionales Reibmoment, das mit Hilfe der Konstante  $k_r$  berechnet werden kann [Adamy (2009)]. Zusammengefasst lautet die Systembeschreibung:

$$\frac{di}{dt} = \frac{1}{L} \cdot (-Ri - k_a\omega + u) \quad (5.8a)$$

$$\frac{d\omega}{dt} = \frac{1}{J} \cdot (k_a i - k_r \omega - T_l) \quad (5.8b)$$

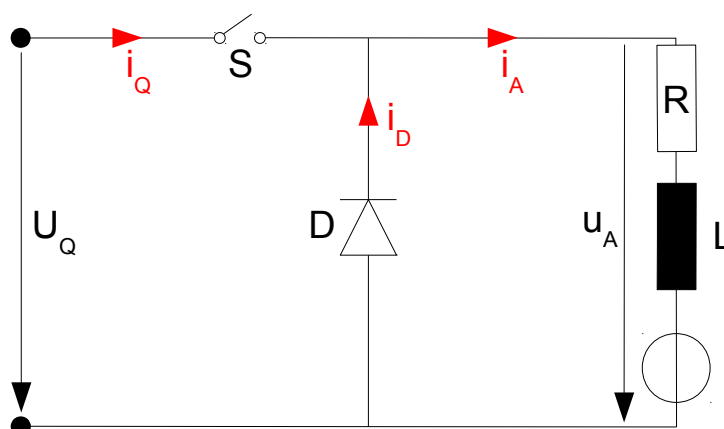
## 5.2 Gleichstromsteller

Gleichstromsteller, auch DC-DC-Wandler genannt, dienen dazu, die konstante Gleichspannung der Quelle  $U_Q$  in eine variable umzuwandeln, wie es für den geregelten Betrieb der Gleichstrommaschine erforderlich ist. Die Schaltungen sind auf sehr hohe Wirkungsgrade ausgelegt [Schröder (2009a)].

### 5.2.1 Tiefsetzsteller

Ein Tiefsetzsteller (auch Abwärts- oder Buck-Wandler genannt) liefert eine Ausgangsspannung kleiner oder gleich groß der anliegenden Quellspannung  $U_Q$ . Erreicht wird dies durch periodisch wechselnden Betrieb eines induktiven Verbrauchers, der entweder an eine Spannungsquelle geschaltet, oder über die Diode  $D$  kurzgeschlossen wird. Abbildung 5.4 zeigt die prinzipielle Schaltung eines Tiefsetzstellers mit nachgeschaltetem Gleichstrommotor als Last.

Wenn der Schalter  $S$  geschlossen ist, fließt der Strom aus der Quelle  $i_Q$  über den Verbraucher (Diode sperrt). Dabei wird magnetische Energie in der Spule gespeichert. Öffnet man den Schalter, so ist der Stromkreis über die nun leitende Diode geschlossen, die gespeicherte magnetische Energie baut sich durch einen allmählich abklingenden Stromfluss  $i_D$  ab. Zur Ansteuerung des meist mittels Transistor realisierten Schalters verwendet man üblicherweise PWM. Dabei variiert man während einer fixen



**Abbildung 5.4:** Schaltung eines Tiefsetzstellers mit einer Gleichstrommaschine als induktivem Verbraucher.

Periodendauer  $T = t_{on} + t_{off}$  das Verhältnis zwischen Zeit im Quellbetrieb  $t_{on}$  und Zeit während Kurzschluss  $t_{off}$ . Für die zeitlichen Mittelwerte der Ausgangsspannung  $u_A$  und des der Quelle entnommenen Stromes  $i_Q$  gilt [Schröder (2009a)]:

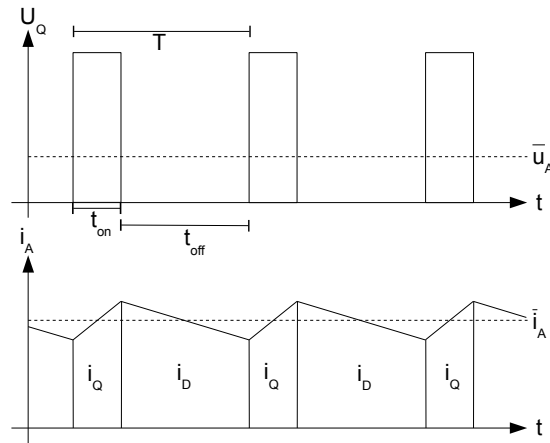
$$\overline{u_A} = U_Q \cdot \frac{t_{on}}{T} \quad (5.9)$$

$$\overline{i_Q} = i_A \cdot \frac{t_{on}}{T} \quad (5.10)$$

In Abbildung 5.5 oben erkennt man den pulsformigen Spannungsverlauf. Darunter ist der in diesem Fall nicht lückende Ankerstrom  $i_a$  abgebildet, der abwechselnd aus der Quelle und aus der Induktivität gespeist wird. Unter der Annahme eines verlustfreien Umschaltvorganges gilt gemäß der Leistungsbilanz: Die der Quelle entnommene Leistung ist gleich der im Verbraucher umgesetzten Leistung [Schröder (2009a)]:

$$P_Q = U_Q \overline{i_Q} = \overline{u_A} i_A = P_{motor} \quad (5.11)$$

Für ein Tastverhältnis kleiner 100% ist der Ankerstrom folglich immer größer als der Quellstrom.



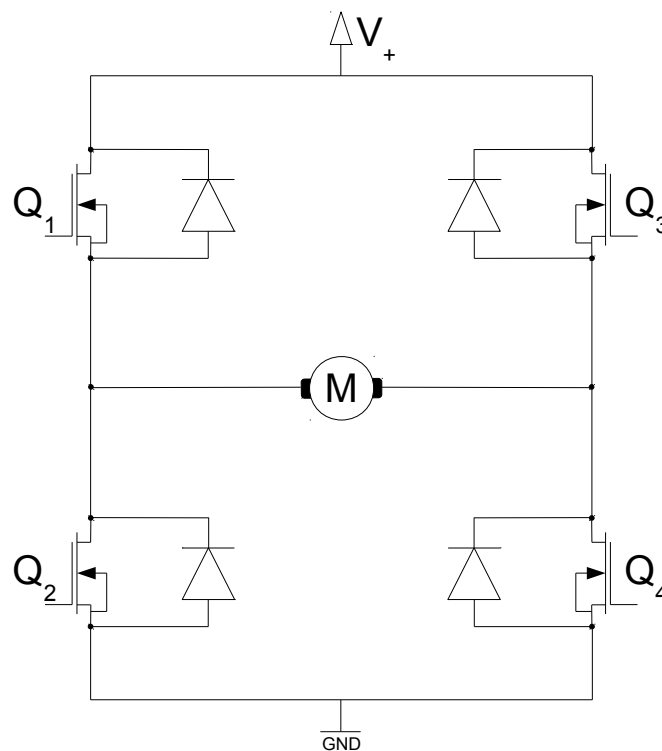
**Abbildung 5.5:** Zeitliche Verläufe von Ausgangsspannung und -strom beim PWM-Betrieb des Tiefsetzstellers mit induktiver Last: Während  $t_{on}$  kommt der Strom aus der Quelle, während  $t_{off}$  aus der Induktivität.

## 5.2.2 Vierquadrantsteller

Durch den Einsatz eines Vierquadrantstellers kann die elektrische Maschine in allen Quadranten des Drehmoment-Drehzahl-Diagramms betrieben werden. Eine mögliche Realisierungsform ist die H-Brücke, wie sie in Abbildung 5.6 zu sehen ist [Schröder (2009a)].

### 5.2.2.1 H-Brücke

Der motorische Betrieb in die beiden möglichen Drehrichtungen wird durch Aktivieren der beiden Diagonalzweige ( $Q_1$  und  $Q_4$  oder  $Q_2$  und  $Q_3$ ) erreicht, wobei jeweils einer der beiden Transistoren über PWM angesteuert wird, während der andere dauerhaft leitet. Die vier antiparallel zu den Transistoren geschalteten Dioden bezeichnet man als *Freilaufdioden*. Ihre Aufgabe ist es, den Stromverlauf zu glätten und gefährliche Spannungsspitzen beim Abschalten der Quellspannung während  $t_{off}$  zu verhindern. Wenn beispielsweise  $Q_2$  eingeschaltet ist und das Gate von  $Q_3$  am PWM-Signal liegt, so fließt der Strom während  $t_{on}$  über  $Q_3$  und während  $t_{off}$  über die Freilaufdiode von  $Q_4$ . Dies entspricht

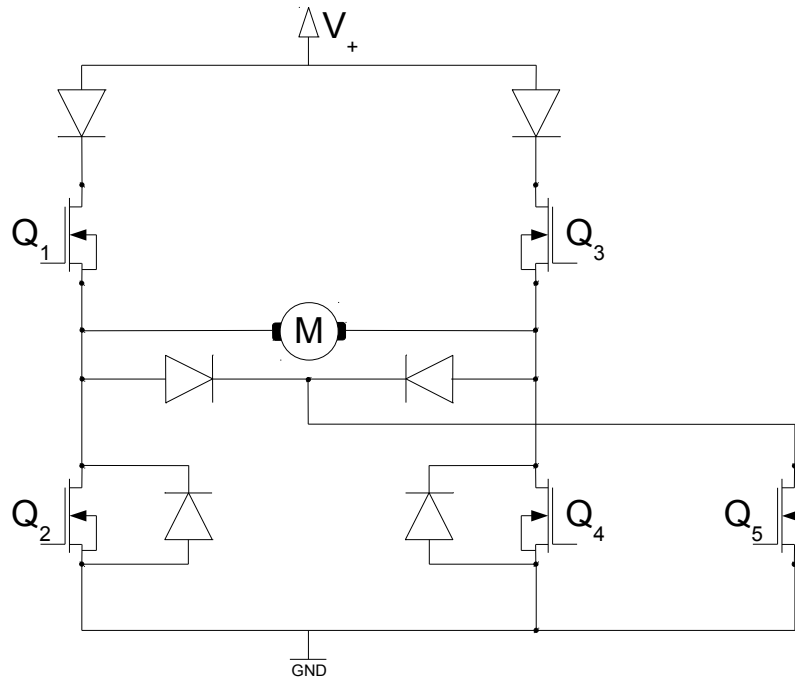


**Abbildung 5.6:** H-Brückenschaltung für den Vierquadrantbetrieb.

dem Verhalten des im vorangegangenen Abschnitt beschriebenen Tiefsetzstellers. Der generatorische Betrieb lässt sich mit Hilfe der beiden unteren Transistoren realisieren, wobei abhängig von der Drehrichtung einer davon über PWM angesteuert wird, um das Bremsmoment steuern zu können. Das durch die Gleichungen 5.8a und 5.8b beschriebene Zustandsraummodell kann unter Verwendung einer H-Brücke unverändert übernommen werden.

### 5.2.2.2 Schaltung am Prüfstand

Der am Prüfstand zum Einsatz kommende Gleichstromsteller enthält einige Modifikationen im Vergleich zur Standardausführung der H-Brücke. Die Schaltung ist in Abbildung 5.7 dargestellt.



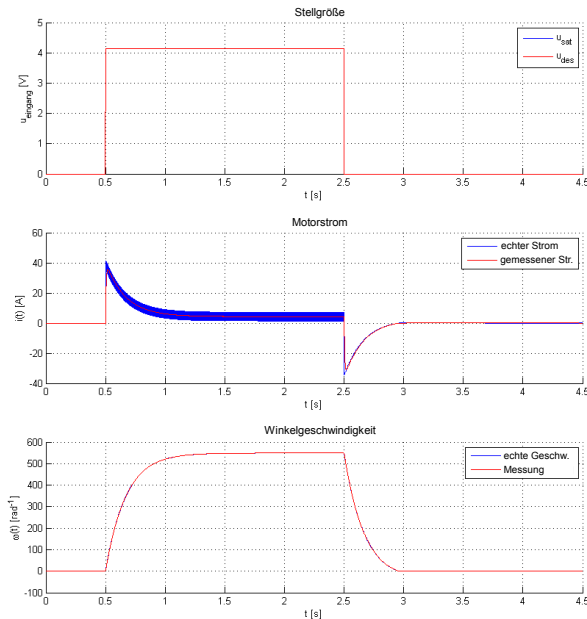
**Abbildung 5.7:** *Modifizierte H-Brückenschaltung, wie sie am Rollenprüfstand zu finden ist.*

Im Gegensatz zum generatorischen Betrieb der Standard-H-Brücke liegt das PWM-Signal hier stets am Gate von Transistor  $Q_5$ , und abhängig von der Drehrichtung wird entweder  $Q_2$  oder  $Q_4$  permanent durchgeschaltet. Dadurch ist es möglich, mit nur einem PWM-Kanal des Mikrocontrollers beide Bremsrichtungen zu steuern. Den Hauptunterschied zur Standardausführung bilden jedoch die fehlenden oberen Freilaufdioden und die statt diesen platzierten Sperrdioden. Deren Einbau verhindert ein Zurückfließen des Stromes in den Plusausgang des Netzteiles, was zu Schäden an selbigem führen könnte. An diesen Eingriff in den Verlauf des Stromes muss die Ankerkreis-Differentialgleichung der Systembeschreibung (Gleichung 5.8a) angepasst werden.

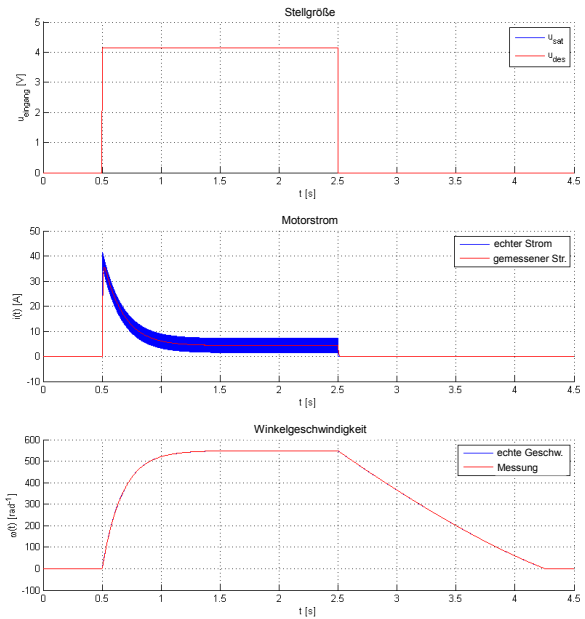
### 5.2.2.3 Auswirkungen der Sperrdioden

Die Sperrdioden schränken die Stromrichtung in Abhängigkeit vom Vorzeichen der Eingangsspannung ein, sodass immer gilt: Stromvorzeichen = Eingangsspannungsvorzeichen. Der Strom wird also bei einem Wert von Null Ampere begrenzt, die Ankerkreisdifferentialgleichung zeigt nichtlineares Verhalten. Dies lässt sich anhand von zwei Betriebssituationen demonstrieren:

**1.) Schnelle betragsmäßige Reduktion der Eingangsspannung:** Ein Vergleich des Verhaltens von linearem Motormodell und jenem des Modells mit Sperrdioden ist in den Abbildungen 5.8 und 5.9 zu sehen.



**Abbildung 5.8:** Verhalten des linearen Systems nach Gleichungen 5.8a und 5.8b bei sprunghafter Reduktion der Eingangsspannung. Ab dem Zeitpunkt  $t = 2.5$  gilt  $u=0$ , daher wird  $\frac{di}{dt} < 0$  und in Folge dessen auch  $i < 0$ . Der Motor bremst bis zum Stillstand.



**Abbildung 5.9:** Abweichendes Verhalten des Prüfstandes beim Wegfall der Eingangsspannung. Aufgrund der Sperrdioden kann der Strom sein Vorzeichen nicht wechseln und der Motor läuft nur durch die Reibung gebremst aus.

**2.) Simulation eines Gefälles:** In diesem Fall treiben die Motoren von Prüfstand und Fahrzeug in dieselbe Richtung an. Angenommen die gewünschte Drehrichtung sei positiv (d.h. das vom Fahrzeug aufgebrachte Lastmoment ist negativ), so kommt es gemäß der linearen Ankerkreis-differentialgleichung erst ab einer gewissen Schwellenspannung, nämlich der durch die Drehung induzierten Gegenspannung  $u_{ind} = k_a \omega$ , zu einem positiven Stromfluss und damit auch beschleunigendem Moment. Unterhalb dieser Schwellenspannung wird lediglich der Bremsstrom verringert, der durch die vom Fahrzeug verursachte Drehung der Rollen entsteht. Tabelle 5.1 zeigt den Unterschied zum Prüfstand.

Eingangsspannung	Standard H-Brücke	Verhalten d. Prüfstandes
$u < k_a \omega$	neg. Strom (brems. Moment)	pos. Strom (beschl. Moment)
$u \geq k_a \omega$	pos. Strom (beschl. Moment)	pos. Strom (beschl. Moment)

**Tabelle 5.1:** Simulation eines Gefälles mit linearem Motormodell und nichtlinearem Modell bei positiver Drehrichtung der Rollen.

Um dieses Phänomen zu erklären, genügt es nicht mehr, allein die Sperrdioden zu berücksichtigen, denn in diesem Fall müsste der Strom bei  $u < k_a \omega$  gleich Null sein. Zusätzlich muss man sich von der Betrachtung der Eingangsspannung als Mittelwert über die PWM-Periode lösen. Denn auch

wenn  $\bar{u} < k_a \omega$  gilt, so sind die einzelnen Spannungsimpulse (zumindest bei den vom Fahrzeug erreichbaren Geschwindigkeiten) nach wie vor größer als die induzierte Spannung. Es kommt daher zu einem positiven, wenn auch lückenden Stromfluss.

### 5.3 Zahnriemengetriebe und Rollen

Ein Getriebe dient zur Wandlung von Drehzahl und Drehmoment einer Antriebsmaschine in einen anderen Bereich. Die wesentliche Kenngröße ist das Übersetzungsverhältnis<sup>1</sup> [Schlecht (2009)]:

$$g = \left| \frac{\omega_{mot}}{\omega_{out}} \right| = \frac{r_{out}}{r_{mot}} = \frac{T_{out}}{T_{mot}} \quad (5.12)$$

Zahnriemengetriebe nutzen das Prinzip der formschlüssigen Kraftübertragung, daher tritt kein Schlupf zwischen Antriebs- und Abtriebsseite auf, d.h. das Übersetzungsverhältnis kann als konstant angesehen werden. Für die Berechnung des antriebsseitig wirkenden Trägheitsmomentes der Last gilt die Formel [Schlecht (2009)]:

$$J_{mot} = \frac{1}{g^2} \cdot J_{out} \quad (5.13)$$

Das Trägheitsmoment des Systems wird durch die Rollen (und abermals durch das daraufstehende Fahrzeug) naturgemäß vergrößert. Es muss folglich ein höheres Moment aufgewendet werden, um die Rollen in Bewegung zu setzen. Daher ist es naheliegend, dem bisher rein geschwindigkeitsproportionalen Reibmoment einen konstanten Term hinzuzufügen. Dieser modelliert das notwendige Anfahrmoment.

$$T_{fric,d}(\omega) = k_1 \omega + k_0 \text{sign}(\omega) \quad (5.14)$$

Der Koeffizient  $k_0$  beschreibt den Anteil der *trockenen* Reibung, während  $k_1$  sich auf die *viskose* Reibung bezieht.

### 5.4 Modell des Prüfstandes

Im motorischen Gesamtmodell ist nun den in den Abschnitten 5.2 und 5.3 begründeten Abweichungen vom Standardmodell der Gleichstrommaschine Rechnung zu tragen.

#### 5.4.1 Modellierung der Sperrdioden

Um das nichtlineare Verhalten der Sperrdioden wiederzugeben, kann der gewöhnlichen Ankerstromdifferentialgleichung (Gleichung 5.8a) ein nichtlinearer „Bestrafungsterm“ hinzugefügt werden. Dieser sorgt dafür, dass der Strom den nicht erlaubten Bereich nicht erreichen kann, bzw. gezwungen wird, diesen sofort wieder zu verlassen. Es entsteht also eine Schwingung um Null Ampere. Tabelle 5.2 fasst die Anforderungen an die Funktion zusammen:

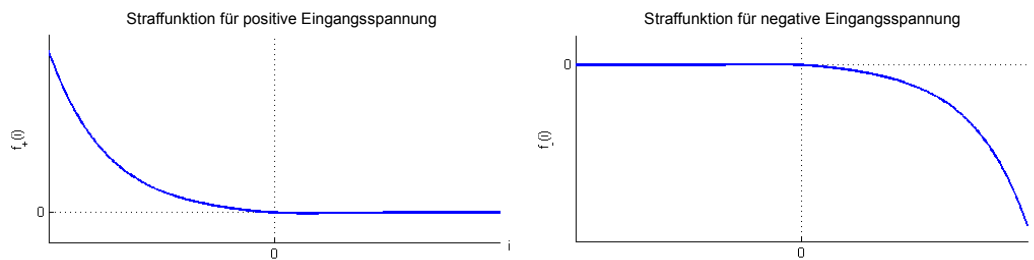
<sup>1</sup>Anmerkung: In einschlägiger Fachliteratur ist das Übersetzungsverhältnis eines einstufigen Getriebes wegen der umgekehrten Drehrichtung von Antrieb und Abtrieb stets negativ. Der Einfachheit halber wird dies hier bewusst ignoriert. Weiters wird „i“ als das übliche Formelzeichen für Übersetzungsverhältnisse durch „g“ (für *gear ratio*) ersetzt, da ansonsten Verwechslungsgefahr mit dem Motorstrom besteht.



$\bar{u}$	$i$	$f_{nlin}(\bar{u}, i)$
$> 0$	$\geq 0$	$0$
$> 0$	$< 0$	$\gg 0$
$< 0$	$\leq 0$	$0$
$< 0$	$> 0$	$\ll 0$
$= 0$	$\neq 0$	$\gg 0$ oder $\ll 0$

**Tabelle 5.2:** Gewünschtes Verhalten des nichtlinearen Terms der Stromdifferentialgleichung.

Der Verlauf einer stetigen Funktion, die obige Forderungen erfüllt, ist in Abbildung 5.10 zu sehen. Sie kann relativ leicht mittels Exponentialfunktionen konstruiert werden.



**Abbildung 5.10:** Möglicher Verlauf des nichtlinearen Terms in Abhängigkeit vom Strom- und Eingangsspannungsvorzeichen.

Um ein zufriedenstellendes Verhalten (sprich eine ausreichend kleine Schwingung um Null) zu erhalten, müssen die Parameter der Funktion so gewählt werden, dass sie schon bei sehr kleinem Strom große Werte liefert. Dies führt jedoch zu einem Problem beim schnellen Wechsel des Eingangsspannungsvorzeichens. Direkt nach dem Wechsel hat der Strom noch das „falsche“ Vorzeichen und zudem auch noch einen relativ hohen Betrag. Die Bestrafungsfunktion liefert in diesem Fall viel zu hohe Werte, das System wird instabil und stimmt daher nicht mehr mit dem realen Prüfstand überein. Der Einsatz eines konstanten Bestrafungsterms schafft hier Abhilfe. Nach Einführung der im vorangegangenen Abschnitt beschriebenen Reibungsfunktion erhält man zusammenfassend folgende Zustandsraumdarstellung:

$$\frac{di}{dt} = -\frac{R}{L} \cdot i - \frac{k_a}{L} \cdot \omega + \underbrace{K [\text{sign}(\bar{u}) - \text{sign}(i)]}_{f_{nlin}(\bar{u}, i)} + \frac{1}{L} \cdot u \quad (5.15a)$$

$$\frac{d\omega}{dt} = \frac{k_a}{J} \cdot i - \frac{k_1}{J} \cdot \omega - \frac{k_0}{J} \cdot \text{sign}(\omega) - \frac{1}{J} \cdot T_l \quad (5.15b)$$

Da der Parameter  $K$  der nichtlinearen Bestrafungsfunktion keine wirkliche physikalische Entsprechung hat, ist seine Bestimmung auch nur mit Hilfe des Simulationsmodells möglich. Er muss groß genug sein, um eine kleine Schwingung um Null zu erhalten, darf aber auch nicht zu groß sein, damit im Falle eines Eingangsspannungsvorzeichenwechsels kein instabiles Verhalten auftritt. In der Simulation hat sich ein Wert von  $K = 140000$  bewährt.



# 6 Sensorik

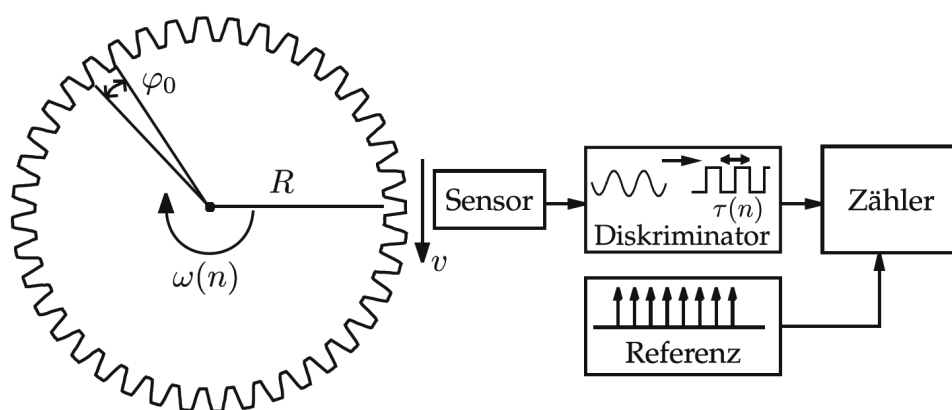
## 6.1 Strommessung

Das Problem der fehlerhaften Strommessung mittels Shunt-Widerstand wurde durch den Einbau von zwei Strommess-ICs *Sensitec CDS4050* gelöst. Einer der beiden Sensoren misst den von der Versorgungsspannungsquelle entnommenen Strom, der andere den Ankerstrom der Gleichstrommaschine. Der Messpunkt für letzteren liegt nun direkt neben dem Motor in der modifizierten H-Brücke (siehe Abbildung 5.7), sodass auch das Vorzeichen des Stromes bestimmt werden kann. Das Messprinzip des Sensors beruht auf dem *Anisotropen Magnetoresistiven Effekt*, sprich der Widerstandsänderung eines Materials aufgrund des Magnetfeldes des zu messenden Stromes. Die Ausgangsspannung des Sensors wird anschließend noch gefiltert und dann an den ADC-Kanal des ATmega128 weitergeleitet.

## 6.2 Winkelgeschwindigkeitsmessung

Zur Messung der Drehzahl bzw. Winkelgeschwindigkeit wird meist auf Inkrementalgeber zurückgegriffen, die das *amplitudenanaloge* Geschwindigkeitssignal in ein *frequenzanaloges* Signal umwandeln. Dies führt zu erhöhter Unempfindlichkeit gegenüber Störungen. Anschließend erfolgt die Bestimmung der Drehzahl selbst über eine Zeitintervallmessung, wie sie mit Mikrocontrollern relativ leicht durchgeführt werden kann (siehe Abbildung 6.1). Die zu messende Winkelgeschwindigkeit ist definiert als [León und Kiencke (2011)]:

$$\omega = \frac{\varphi}{T_\varphi} = \frac{2\pi}{T} \tag{6.1}$$



**Abbildung 6.1:** Prinzip der Drehzahlmessung: Jedes durchlaufene Winkelinkrement  $\varphi_0$  wird vom Sensor (meist optisch oder magnetisch) registriert und löst einen Impuls mit fixer Periodendauer aus. Durch Impulszählung und Zeitmessung kann auf die Winkelgeschwindigkeit rückgerechnet werden. [Quelle: León und Kiencke (2011), Seite 363]

Die gemessene Winkelgeschwindigkeit berechnet sich grundsätzlich immer aus dem Differenzenquotienten von Winkelintervall und dem dafür benötigten Zeitintervall. Zur Bestimmung dieses Differenzenquotienten kann man nun wahlweise eine Periodendauermessung oder eine Frequenzmessung heranziehen. Die Verfahren unterscheiden sich in ihrer Genauigkeit abhängig vom Geschwindigkeitsbereich [León und Kiencke (2011)].

$$\omega_m = \frac{\varphi_2 - \varphi_1}{t_2 - t_1} \quad (6.2)$$

### 6.2.1 Periodendauermessung

Dieses Verfahren beruht auf der Messung der Zeit, die zum Zurücklegen eines Winkelinkrements benötigt wird. In Gleichung 6.2 bleibt der Zähler also konstant, während der Nenner messtechnisch erfasst wird. Man spricht auch von *winkelsynchroner* Abtastung, da mit jedem Winkelschritt ein neuer Geschwindigkeitswert vorliegt. Die quantisierte Winkelgeschwindigkeit beträgt hier [León und Kiencke (2011)]:

$$\omega_q = \frac{\varphi_0}{NT_0} \quad (6.3)$$

In obengenannter Gleichung bezeichnet  $T_0$  die verwendete Zeitauflösung und  $N$  den aktuellen Zählerstand.

### 6.2.2 Frequenzmessung

Bei dieser Methode wird die Anzahl der durchlaufenen Winkelinkremente während einer fixen Messperiode gezählt. Bei dieser sogenannten *zeitsynchronen* Abtastung bleibt der Nenner in Gleichung 6.2 konstant, stattdessen variiert der Zähler. Für die gemessene Winkelgeschwindigkeit gilt folglich [León und Kiencke (2011)]:

$$\omega_q = \frac{N\varphi_0}{T_{ref}} \quad (6.4)$$

### 6.2.3 Vergleich der Quantisierungsfehler beim Einsatz am Prüfstand

Nun folgt eine Gegenüberstellung der Quantisierungsfehler für stationäre Winkelgeschwindigkeiten während einer Messperiode ( $T_{\varphi_0}$  bei Periodendauermessung und  $T_{ref}$  bei Frequenzmessung). Die Abweichung betrifft die *wahre Zeit*  $T_m$  bei der Periodendauermessung und den *wahren Winkel*  $\varphi_m$  bei der Frequenzmessung und beträgt jeweils plus oder minus ein Quantisierungsintervall. Daraus lässt sich, wie bei León und Kiencke (2011) beschrieben, wiederum der maximale relative Fehler der Winkelgeschwindigkeit angeben:

Periodendauermessung:

$$|T_m - NT_0| \leq T_0 \quad (6.5)$$

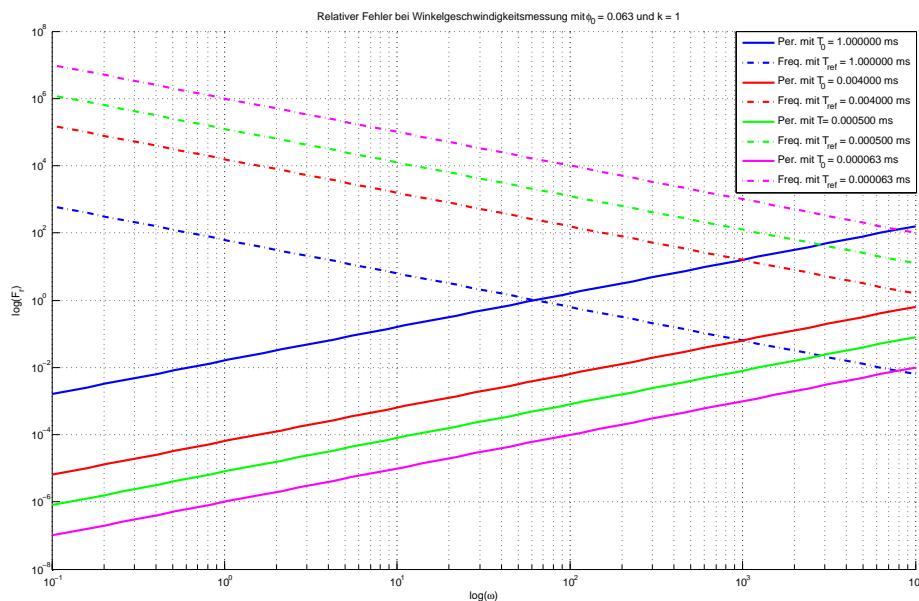
$$F_{\omega,r} \leq \frac{\omega_q T_0}{\varphi_0} \approx \frac{\omega_m T_0}{\varphi_0} \quad (6.6)$$

Frequenzmessung:

$$|\varphi_m - N\varphi_0| \leq \varphi_0 \quad (6.7)$$

$$F_{\omega,r} \leq \frac{\varphi_0}{\omega_m T_{ref}} \quad (6.8)$$

Man erkennt, dass mit steigender Winkelgeschwindigkeit der relative Fehler der Periodendauermessung zunimmt, die Frequenzmessung hingegen bessere Ergebnisse liefert. Die Frequenzmessung arbeitet erwartungsgemäß umso genauer, je höher die Winkelauflösung (kleinere Winkelinkremente  $\varphi_0$ ) ist und je länger ihre Messperiode  $T_{ref}$  ist. Beim Einsatz in einer digitalen Regelung wird deren Maximalwert jedoch durch die Geschwindigkeit des Reglers begrenzt. Die Periodendauermessung dagegen profitiert von einer möglichst kleinen zeitlichen Auflösung  $T_0$ , was für den Timer des ATmega128 mit einer maximalen Frequenz von 16 MHz keinerlei Problem darstellt. Beim verwendeten Sensor handelt es sich um einen *Kübler Inkremental-Drehgeber Typ 2400* mit einer Auflösung von 100 Impulsen pro Umdrehung was einem Winkelinkrement von  $\varphi_0 = 0.0628rad$  entspricht. Gemeinsam mit den Fehlerabschätzungen laut Ungleichungen 6.6 und 6.8 kann nun ein Vergleich beider Verfahren bei verschiedenen Zeitauflösungen bzw. Messperioden angestellt werden.



**Abbildung 6.2:** Vergleich des relativen Fehlers für Periodendauer- (durchgehend) und Frequenzmessung (strichliert) bei stationärer Drehzahl.

Abbildung 6.2 zeigt für die vorliegende Winkelauflösung deutlich die Überlegenheit der Periodendauermessung, deren Ursache wie bereits angedeutet in der hohen zur Verfügung stehenden zeitlichen Auflösung liegt. Um mit der Frequenzmessung ähnlich gute Ergebnisse zu erzielen, müsste die Messperiode bedeutend vergrößert werden, was aber zu einer langsameren Regelung führen würde. Die maximale im Betrieb zu erwartende Winkelgeschwindigkeit liegt bei etwa  $1000 \frac{rad}{s}$ . Bei Verwendung der Periodendauermessung mit  $T_0 = 0.5\mu s$  kann selbst bei dieser Geschwindigkeit noch ein Quantisierungsfehler von unter einem Prozent garantiert werden.

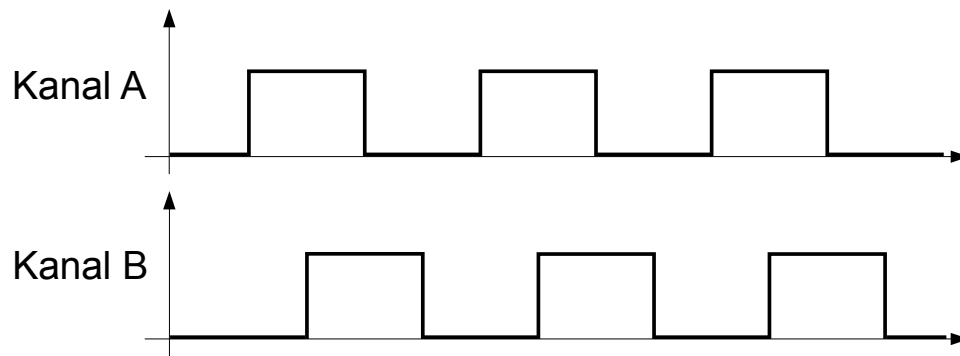
### 6.2.4 Implementierung

Das Rechtecksignal des Drehgebers löst einen Interrupt bei jeder steigenden Flanke aus. Im Interrupthandler wird nun der aktuelle Zählerstand des Timers ausgelesen und mit Hilfe des vorhergehenden die vergangene Zeit berechnet. Allenfalls aufgetretene Overflows des Zählers werden in die Berechnung miteinbezogen. Um die Rechenzeit möglichst kurz zu halten, erfolgt

die Berechnung der Winkelgeschwindigkeit laut Formel 6.3 erst während der Ausführung des Regelungs-Tasks, da Divisionen verhältnismäßig rechenzeitaufwändig sind.

### 6.2.5 Drehrichtungserkennung

Um die Drehrichtung der Motorwelle bestimmen zu können, stellt der eingesetzte Inkrementalgeber ein zweikanaliges Ausgangssignal zur Verfügung. Die Phasenverschiebung zwischen den Kanälen beträgt  $\frac{\pi}{2}$ , sodass sich die Impulse stets überlagern, wie sich aus Darstellung 6.3 ablesen lässt.



**Abbildung 6.3:** Zweikanaliges Drehgebersignal zur Drehrichtungserkennung.

Kanal A löst, wie in Abschnitt 6.2.4 beschrieben, einen Interrupt bei *steigender Flanke* aus und dient der eigentlichen Drehzahlmessung. Im Interrupthandler wird nun zusätzlich der momentane Wert von Kanal B ausgelesen, der je nach Drehrichtung 0 oder 1 ist.

# 7 Parameteridentifikation

## 7.1 Motorkonstante

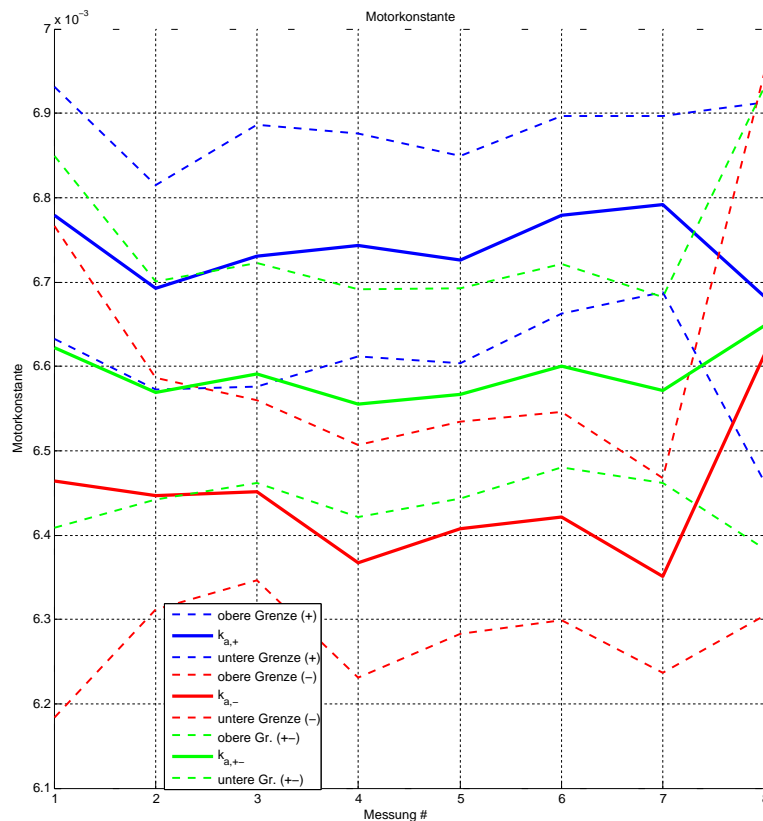
### 7.1.1 Versuchsbeschreibung

Die Bestimmung der Motorkonstante lässt sich recht einfach durch die Messung der induzierten Gegenspannung bewerkstelligen. Das Fahrzeug treibt bei diesem Versuch den Motor des Prüfstandes an, dessen Antriebsstromkreis nicht geschlossen wird. Die am Motor gemessene Spannung entspricht dann der induzierten Gegenspannung, aus der sich bei gegebener Winkelgeschwindigkeit die Motorkonstante berechnen lässt.

$$k_a = \frac{\overline{u_{ind}}}{\overline{\omega}} \quad (7.1)$$

Der Versuch wird in beide Drehrichtungen und bei unterschiedlichen Geschwindigkeiten durchgeführt. Anschließend kann aus den einzelnen Messungen durch Mittelung die Motorkonstante berechnet werden.

### 7.1.2 Ergebnisse



**Abbildung 7.1:** Messungen und Fehlergrenzen für die Motorkonstante: In beiden Richtungen (blau und rot, durchgehend) wurden bei jeweils acht verschiedenen Geschwindigkeiten Messungen vorgenommen. Die Fehlergrenzen sind strichliert dargestellt. Die grüne Kurve zeigt das Ergebnis durch Mittelwertbildung der beiden Drehrichtungen.

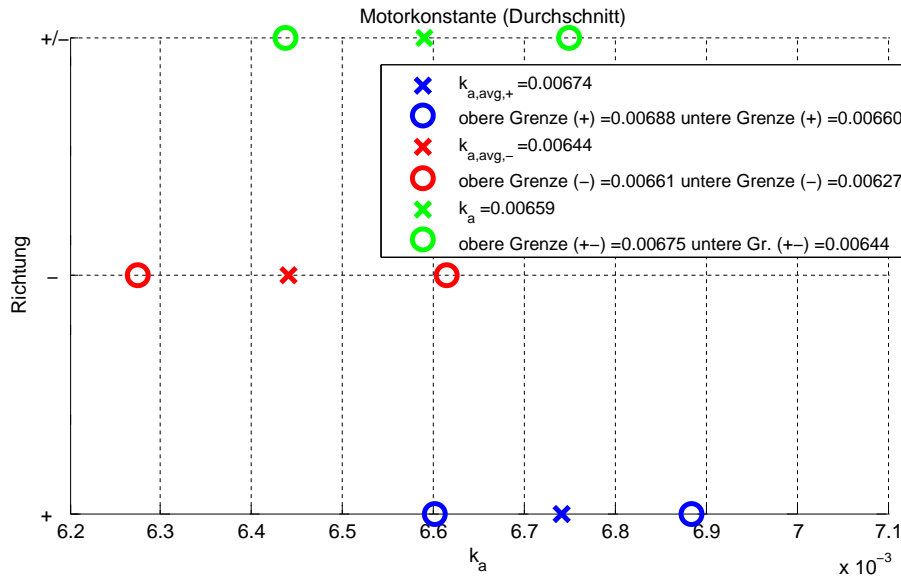


Abbildung 7.2: Messergebnisse für die beiden Drehrichtungen und Gesamtmessergebnis.

Resultat:  $k_a = 0.00659 \frac{Nm}{A}$

### 7.1.2.1 Fehlerabschätzung

Zur Abschätzung der in den Plots eingezeichneten Grenzen, innerhalb derer sich die Motorkonstante befindet, werden die beiden ungünstigsten Fälle betrachtet:

Maximales  $k_a$ :

$$k_{a,max} = \frac{|\overline{u_{ind}} + \Delta_u|}{|\overline{\omega} - \sigma_\omega} \quad (7.2)$$

Minimales  $k_a$ :

$$k_{a,min} = \frac{|\overline{u_{ind}} - \Delta_u|}{|\overline{\omega} + \sigma_\omega} \quad (7.3)$$

$\Delta_u$  ist die Messunsicherheit des Multimeters<sup>1</sup> und  $\sigma_\omega$  die geschätzte Standardabweichung bei der jeweiligen Geschwindigkeit. Die somit erhaltene Fehlergrenze für das Resultat beträgt etwa  $\pm 2.4\%$ .

## 7.2 Reibungsmoment

### 7.2.1 Versuchsbeschreibung

Der Prüfstand wird nun mit verschiedenen Geschwindigkeiten betrieben, zunächst ohne Fahrzeug. Unter der Voraussetzung einer bekannten Motorkonstante (Abschnitt 7.1) kann im stationären Zustand auf das wirkende Reibmoment geschlossen werden. Die mechanische Differentialgleichung 5.15b kann nun folgendermaßen umgeformt werden:

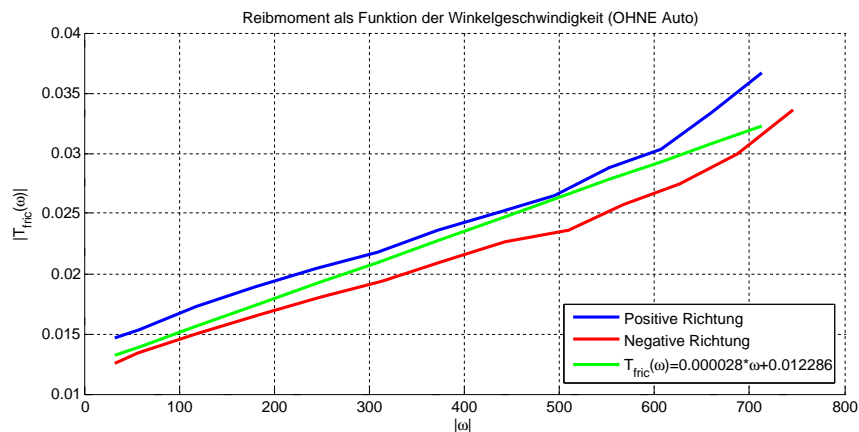
$$T_{fric}(\omega) = k_a \bar{i} \quad (7.4)$$

Auch dieser Versuch wird in beiden Drehrichtungen vorgenommen. Zusätzlich wiederholt man die Messung mit am Prüfstand befindlichem Fahrzeug. Dadurch wird die auf den Prüfstandsmotor zusätzlich wirkende Reibung des Fahrzeuggetriebes ermittelt.

<sup>1</sup>Bei der Berechnung von  $\Delta_u$  wurde das Spannungsvorzeichen miteinbezogen, daher muss die Betragsbildung für den gesamten Zähler erfolgen.



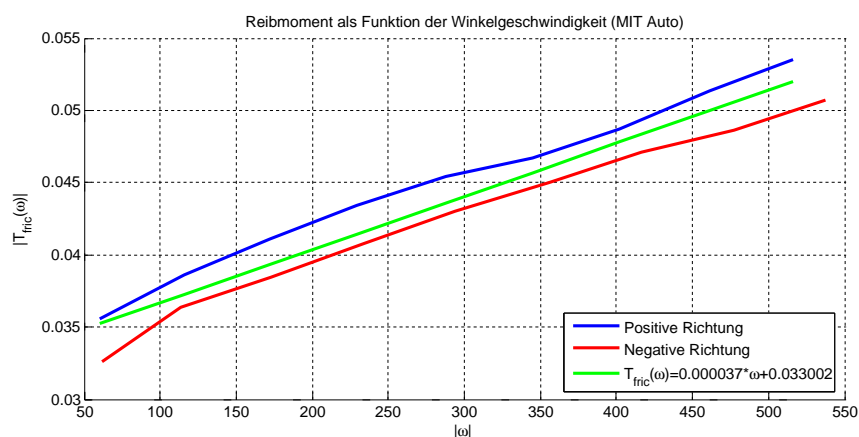
## 7.2.2 Ergebnisse



**Abbildung 7.3:** Reibungsmoment des Prüfstandes (OHNE Fahrzeug) abhängig von der Winkelgeschwindigkeit: Die grüne Kurve ist die lineare Approximation der beiden Drehrichtungen.

Zur Berechnung der Approximation wird der lineare Ansatz laut Gleichung 5.14 verwendet. Das Resultat lautet:

$$T_{fric,d}(\omega) = 0.000028\omega + 0.012286\text{sign}(\omega)$$



**Abbildung 7.4:** Reibungsmoment des Prüfstandes inklusive Fahrzeug..

Das Gesamtreibmoment von Fahrzeug und Prüfstand beträgt:

$$T_{fric,d+c}(\omega) = 0.000037\omega + 0.033002\text{sign}(\omega)$$

## 7.3 Ankerwiderstand

### 7.3.1 Versuchsbeschreibung

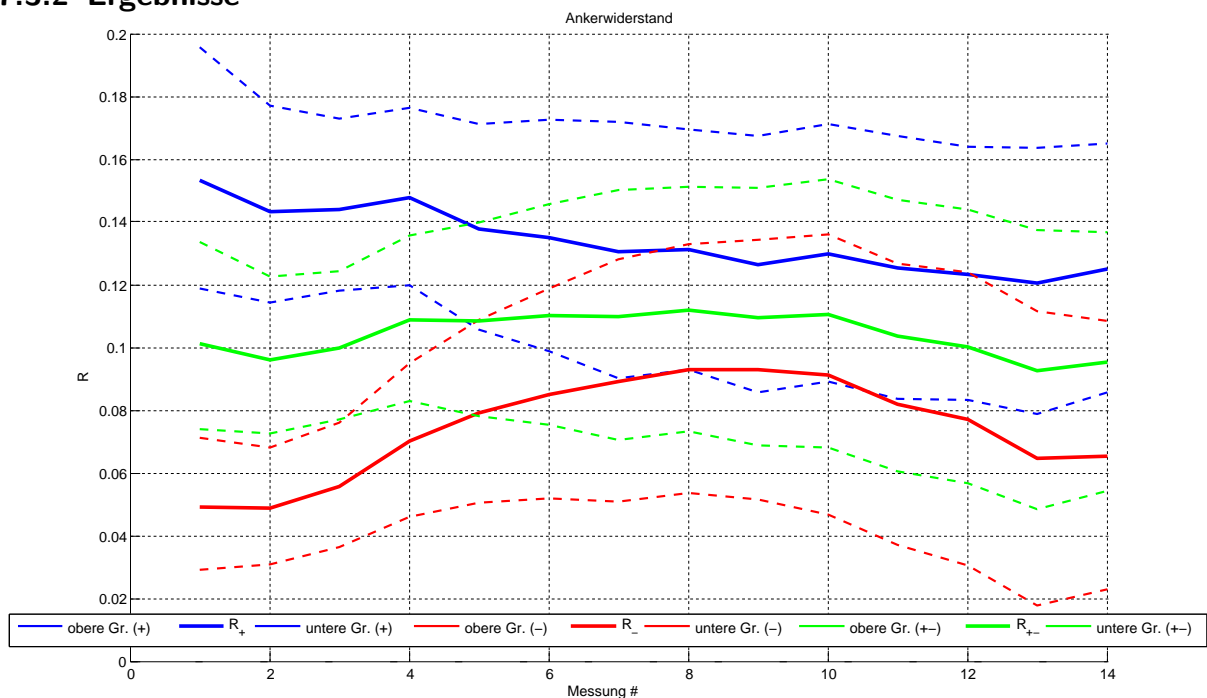
Die übliche Vorgangsweise zur Bestimmung des Ankerwiderstands besteht im Blockieren des Motors, um den generatorischen Effekt der Gegenspannungsinduktion zu verhindern. Anschließend kann bei gegebener Eingangsspannung über das Ohm'sche Gesetz der Widerstand berechnet werden. Da die Motorkonstante bereits berechnet wurde, kann der Widerstand jedoch auch

bei drehendem Anker bestimmt werden. Der Vorteil dieser Vorgehensweise ist, dass die Gleichstrommaschine in dem Betriebszustand vermessen wird, indem sie auch zum Einsatz kommt. Es können dabei alle sich auf den fließenden Strom auswirkenden Größen (z.B. Ummagnetisierungsverluste durch die Drehung, Bürstenspannungsabfälle, Verluste durch die Motoransteuerung, ...) indirekt in den Widerstand miteinbezogen werden.

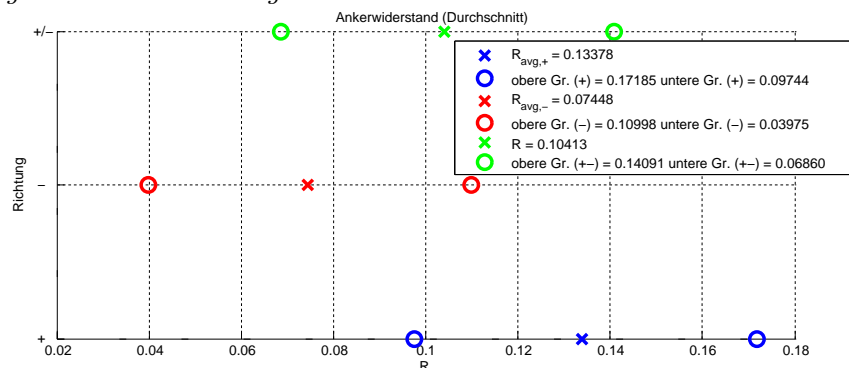
$$R = \frac{\bar{u} - k_a \omega}{\bar{i}} \quad (7.5)$$

Auch diesen Versuch führt man für positive und negative Eingangsspannungen durch und kann somit durch Mittelwertbildung auf den Widerstand schließen.

### 7.3.2 Ergebnisse



**Abbildung 7.5:** Messungen und Fehlergrenzen für den elektrischen Widerstand: In beiden Polungen (blau und rot, durchgehend) wurden bei jeweils 14 Eingangsspannungen bzw. Geschwindigkeiten (in ansteigender Reihenfolge) Messungen vorgenommen. Die Fehlergrenzen sind strichliert dargestellt. Man erkennt Schwankungen des Widerstandes abhängig vom Betrag der Eingangsspannung und von der Polung der Gleichstrommaschine.



**Abbildung 7.6:** Widerstand für negative und positive Drehrichtung sowie Mittelwert über alle Messungen.

Resultat:  $R = 0.104\Omega$

### 7.3.2.1 Fehlerabschätzung

Die Betrachtung der ungünstigsten Konstellationen der Messunsicherheiten erlaubt auch hier Rückschlüsse auf die zu erwartende Genauigkeit. Durch das Miteinbeziehen der Geschwindigkeitsmessung mittels (ebenfalls fehlerbehafteter) Motorkonstante leidet die Genauigkeit der Messung. Es hat sich allerdings gezeigt, dass das Ergebnis im Vergleich zur Messung bei blockiertem Anker viel besser dem Maschinenverhalten im Normalbetrieb entspricht.

Maximaler Widerstand:

Minimaler Widerstand:

$$R_{max} = \frac{|\overline{u_{ind}}| + \sigma_u - (k_a - \Delta_{k_a}) \cdot (|\overline{\omega}| - \sigma_\omega)}{|\overline{i}| - \sigma_i} \quad (7.6) \quad R_{min} = \frac{|\overline{u_{ind}}| - \sigma_u - (k_a + \Delta_{k_a}) \cdot (|\overline{\omega}| + \sigma_\omega)}{|\overline{i}| + \sigma_i} \quad (7.7)$$

Wegen des stark unterschiedlichen Verhaltens abhängig von der Polung bzw. Drehrichtung beträgt die Fehlergrenze etwa  $\pm 35\%$ .

## 7.4 Induktivität und Trägheitsmoment

### 7.4.1 Vorgangsweise

Die direkte Messung dieser Größen gestaltet sich schwierig, da sie ausschließlich die Dynamik der Strecke beeinflussen. Eine Messung bei stationären Betriebspunkten ist daher nicht zielführend. Die beiden Parameter werden deswegen mittels numerischer Optimierung bestimmt. Zu diesem Zwecke wird ein Eingangssignal auf den Prüfstand aufgeschaltet, bei dem der nichtlineare Teil der Stromdifferentialgleichung nicht aktiv wird. Anschließend wird ein Modell der Strecke in Simulink aufgebaut. Die zuvor ermittelten stationären Parameter (Widerstand, Motorkonstante und Reibung) werden als konstant betrachtet, während die Optimierung mit Hilfe der Simulink Parameter Estimation Toolbox über die beiden Größen Induktivität und Trägheitsmoment durchgeführt wird. Das Stromsignal des *Sensitec CDS4050* durchläuft noch einen Tiefpass bevor es dem ADC des Mikrocontrollers zugeführt wird. Daher beinhaltet die auf die hier beschriebene Weise bestimmte Induktivität die Dynamik der Messsignalverarbeitung.

### 7.4.2 Ergebnisse

1. Induktivität:  $L = 40.5\mu H$
2. Trägheitsmoment ohne Fahrzeug:  $J = 6.85 \cdot 10^{-5} kgm^2$

Anmerkung: Dies ist das auf die Motorwelle wirkende Trägheitsmoment. Zur Bestimmung des Trägheitsmomentes der Rollen muss es gemäß Gleichung 5.13 mit dem Quadrat des Übersetzungsverhältnisses multipliziert werden. Das Rollenträgheitsmoment beträgt daher:  $J_{roller} = 2.47 \cdot 10^{-3} kgm^2$

## 7.5 Simulation

Um die Richtigkeit der Parameteridentifikation zu demonstrieren, werden in diesem Abschnitt die Simulationsergebnisse mit den Messungen des realen Prüfstandes verglichen. Als klassisches Testsignal kommt hier die Sprungfunktion mit den Tastverhältnissen 20% und 50% zum Einsatz.

## 7.5.1 Lineares Motormodell

Unter Verwendung der Gleichungen 5.8a und 5.8b als Systembeschreibung und dem Mittelwert des PWM-Signals als Eingangsgröße erhält man folgende Ergebnisse.

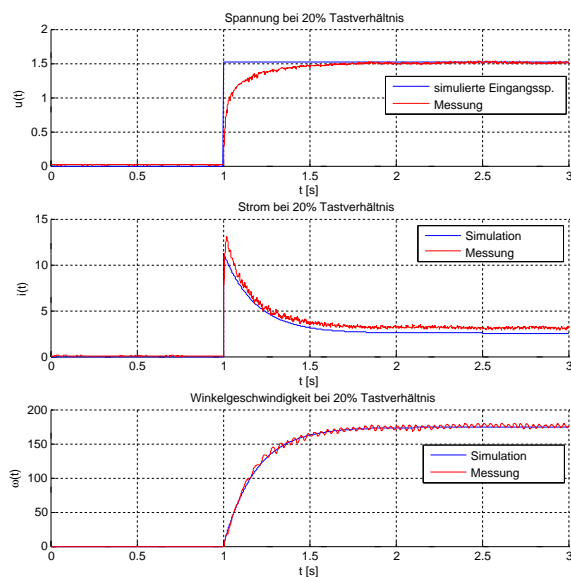


Abbildung 7.7: Sprungantwort des linearen Modells bei 20% Tastverhältnis.

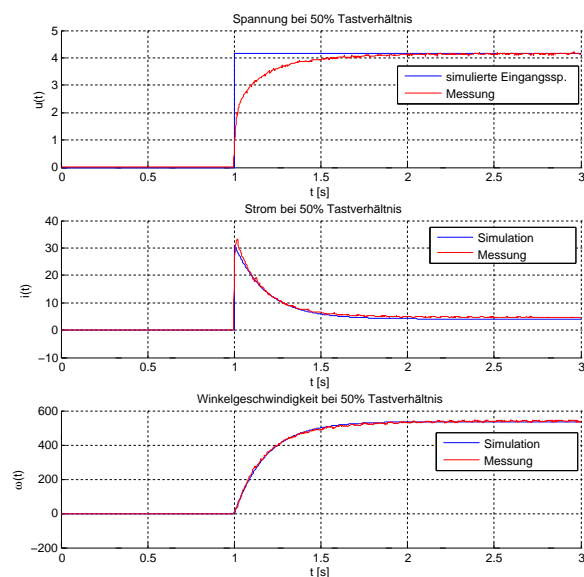


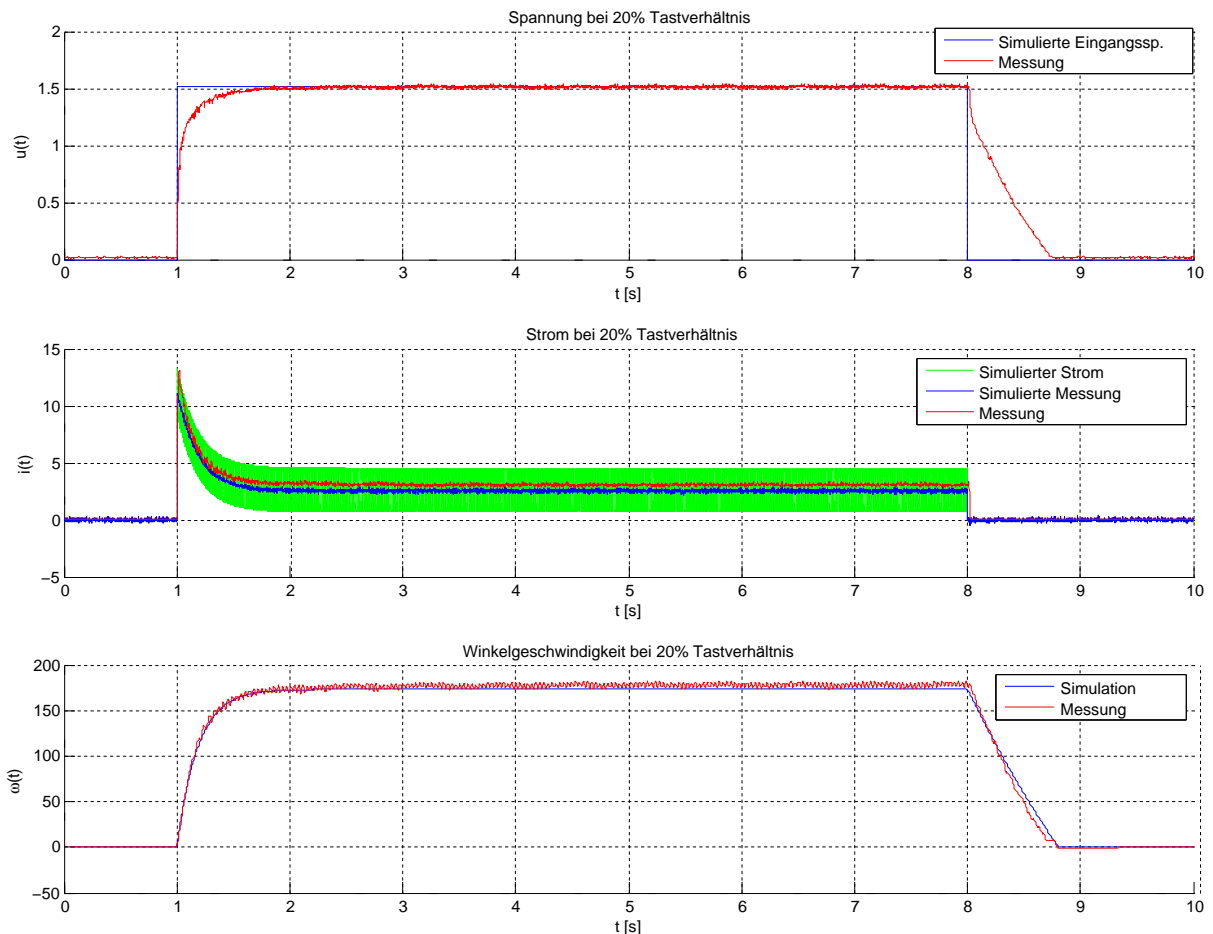
Abbildung 7.8: Sprungantwort des linearen Modells bei 50% Tastverhältnis.

Da die Schaltung zur Messung der Motorspannung ebenfalls einen Tiefpass beinhaltet, weicht der Spannungsverlauf anfangs von der Sprungfunktion ab. Die Form des Stromverlaufes stimmt sehr gut mit dem Messergebnis überein. Aufgrund des stark schwankenden Widerstandes der Gleichstrommaschine sind hier jedoch vor allem im unteren Leistungsbereich (siehe Abbildung 7.7) Abweichungen vom gemessenen Endwert zu verzeichnen. Die simulierte Winkelgeschwindigkeit entspricht in beiden Fällen genau der gemessenen. Der Anteil der trockenen Reibung ist am Prüfstand jedoch so groß, dass er nicht vernachlässigt werden darf. Um obige Simulationsergebnisse zu erzielen, war es daher notwendig, die trockene Reibung als Teil der Eingangsgröße Lastmoment ( $T_{l,lin}$ ) zu modellieren:

$$\frac{d\omega}{dt} = \frac{1}{J} \cdot (k_a i - k_1 \omega - \underbrace{k_0 \text{sign}(\omega)}_{T_{l,lin}} - T_l) \quad (7.8)$$

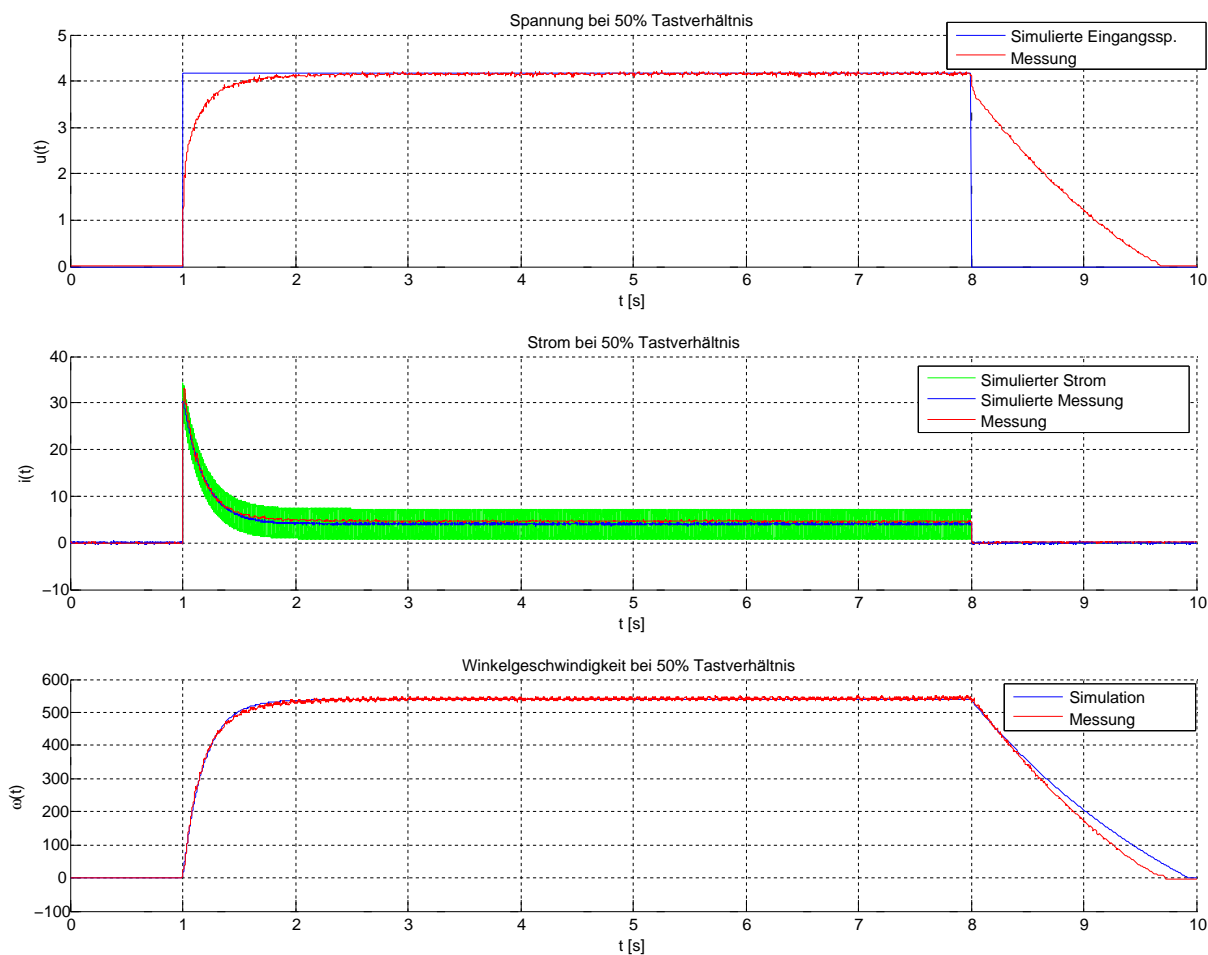
## 7.5.2 Nichtlineares Modell

Um das Streckenverhalten genauer simulieren zu können, wird das nichtlineare Modell laut 5.15a und 5.15b herangezogen. In diesem Fall ist die Eingangsspannung nicht mehr zeitlich gemittelt sondern ein PWM-Signal. Man erkennt in Abbildung 7.9 und 7.10 deutlich den Einfluss der Sperrdioden nach dem Wegfall der Eingangsspannung, da der Strom bei Null Ampere in Begrenzung geht. Danach läuft der Motor durch die Reibung gebremst bis zum Stillstand langsam aus. Auch hier sei anzumerken, dass die Spannungsmessung zu langsam ist, um dem sprungförmigen Anstieg der Eingangsspannung zu folgen. Zum Zeitpunkt  $t = 8$  sinkt das Eingangssignal sprunghaft auf Null Volt ab und die Tatsache, dass nach diesem Zeitpunkt noch eine Spannung gemessen wird, ist nur noch auf die durch die Drehung induzierte Gegenspannung zurückzuführen. Die Geschwindigkeitsverläufe stimmen in beiden Fällen sehr gut mit den Messergebnissen überein.



**Abbildung 7.9:** Sprungantwort des nichtlinearen Modells bei 20% Tastverhältnis: Die grüne Stromkurve ist der tatsächliche durch den Motor fließende Strom, der wegen der PWM-Ansteuerung sehr große Schwankungen aufweist. Dieser Stromverlauf kann beim realen Prüfstand wegen des Tiefpasses am Stromsensorausgang nicht erfasst werden. Bei der Frequenz  $f_{PWM} = 7.812\text{KHz}$  ist der Verlauf jedoch noch nicht lückend<sup>2</sup>. Dieser Fall tritt erst dann ein, wenn die Last in dieselbe Richtung antreibt wie der Prüfstandsmotor und für dessen Eingangsspannung gilt  $|u_{input}| < |k_a\omega|$  (siehe Kapitel 5.2.2.3 für genaue Erläuterung). Die Abweichung beim stationären Strom lässt sich auch hier durch die Ungenauigkeiten beim Widerstand der elektrischen Maschine begründen.

<sup>2</sup>Dies wurde mittels Oszilloskopmessung an einem zwischengeschaltetem Widerstand von  $0.1\Omega$  verifiziert. Da durch den größeren Widerstand die Zeitkonstante verringert wird, kann man daraus schließen, dass auch im Normalbetrieb kein Absinken des Stromes auf Null Ampere stattfindet.



**Abbildung 7.10:** Sprungantwort des nichtlinearen Modells bei 50% Tastverhältnis: Da das Reibmoment durch eine lineare Funktion angenähert wurde, weicht die Auslaufkurve des Motors etwas vom gemessenen Verlauf ab.

# 8 Theoretische Grundlagen

## 8.1 PID-Regler

Industrielle Standardregler sind meist Kombinationen eines Proportional-, eines Integral- und eines Differentialanteils. Kommen alle drei Komponenten zum Einsatz (PID-Regler), wird die Stellgröße  $u(t)$  ausgehend vom Regelfehler  $e(t)$  als Eingangsgröße nach folgender Vorschrift berechnet [Lunze (2010a)]:

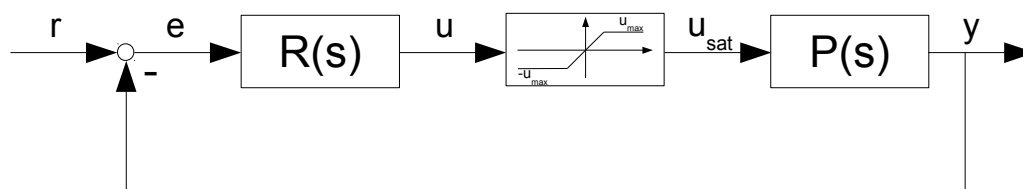
$$u(t) = K_p \cdot \left[ e(t) + \frac{1}{T_i} \cdot \int_0^t e(\tau) d\tau + T_d \cdot \frac{de(t)}{dt} \right] \quad (8.1)$$

Das Verhalten der einzelnen Terme kann wie folgt beschrieben werden:

- **Proportionalanteil:** Ein großer Regelfehler führt hier zu einer großen Stellgröße, um diesen möglichst schnell auszugleichen.
- **Integralanteil:** Der bisherige Verlauf des Regelfehlers wird durch Integration in die Berechnung der aktuellen Stellgröße miteinbezogen. Ein momentaner Wert von  $e(t) = 0$  führt hier zu einer konstanten Stellgröße, sodass für sprunghafte Führungsgrößen stationäre Genauigkeit garantiert werden kann. Ein reiner I-Regler reagiert jedoch eher langsam auf Änderungen der Stellgröße [Lunze (2010a)].
- **Differentialanteil:** Große Änderungen des Regelfehlers führen zu großen Stellgrößen. Dies ermöglicht ein schnelles Gegensteuern des Reglers bevor der Fehler selbst allzu sehr zunimmt, bedeutet jedoch auch hohe Sensibilität in Bezug auf Störeinflüsse wie beispielsweise Messrauschen [Lunze (2010a)].

### 8.1.1 Anti-Windup

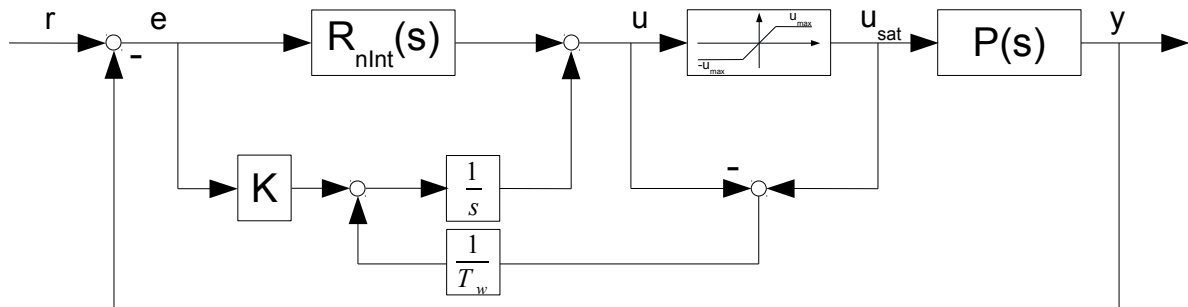
In praktisch jedem Regelkreis unterliegt die Stellgröße  $u(t)$  gewissen Beschränkungen hinsichtlich des Maximal- und Minimalwertes. Häufig sind diese Beschränkungen symmetrischer Natur, sodass die tatsächlich auf die Strecke wirkende Stellgröße  $u_{sat}(t)$  über eine Sättigungsfunktion beschrieben werden kann (siehe Abbildung 8.1) [Adamy (2009)].



**Abbildung 8.1:** Standardregelkreis mit beschränkter Stellgröße: der Regler berechnet  $u(t)$  als Sollwert, kann diesen aber nur innerhalb der Grenzen  $\pm u_{max}$  auf die Strecke schalten.

Bei einer Überschreitung der Stellgrößenbeschränkung arbeitet der Regelkreis nicht mehr linear, da die Stellgröße nun wie eine Konstante auf die Strecke wirkt. Der Regelkreis ist ab diesem

Zeitpunkt unterbrochen. Besonders problematisch ist dieser Umstand bei einem Regler mit integrierendem Anteil. Dieser setzt den Integrationsvorgang während der Beschränkung fort (*Windup*), bis der Regelfehler sein Vorzeichen wechselt (sprich der Sollwert bereits erreicht ist). Erst dann wird der Wert des Integrals wieder abgebaut, die Stellgröße  $u(t)$  ist aber immer noch größer als  $u_{max}$ , d.h. die Verringerung der auf die Strecke wirkenden Stellgröße  $u_{sat}(t)$  erfolgt verzögert. Die Folge ist Überschwingen, das sogar bis zur Instabilität führen kann. Eine Möglichkeit, dies zu beheben, zeigt Abbildung 8.2 [Adamy (2009)].



**Abbildung 8.2:** *Anti-Windup Struktur: Der nicht integrierende Regleranteil ist in  $R_{nInt}(s)$  zusammengefasst. Die Abweichung der gewünschten von der realen Stellgröße wird mit umgekehrtem Vorzeichen in den Integratoreingang zurückgeführt. Dadurch wird dem Aufintegrieren von  $e(t)$  entgegengewirkt.*

Eine andere Variante zur Vermeidung des Regler-Windups, die bei digitalen Regelungen besonders einfach umzusetzen ist, ist die Nullung des Integratoreingangs, sobald die Stellgrößenbeschränkung wirksam wird (d.h.  $(u_{sat} - u) \neq 0$  gilt).

### 8.1.2 Zeitdiskrete Realisierung

Ein großer Teil der heute eingesetzten Regelungen wird mit Hilfe von Rechnern, auf denen der Regler als Programm ausgeführt wird, implementiert. In diesem Fall erfolgt die Berechnung und Aufschaltung der Stellgröße nicht zeitkontinuierlich, sondern mit einer vom Rechner abhängigen Taktfrequenz ( $f_T = \frac{1}{T}$ ). Das Stellsignal wird anschließend mittels Halteglied bis zum nächsten Taktzyklus konstant gehalten. Die einzelnen Bestandteile des PID-Reglers werden dabei durch zeitdiskrete Approximationen realisiert. So ersetzt man den Differentialquotienten durch einen Differenzenquotienten und zur näherungsweisen Bestimmung des Integrals kann unter anderem die rechtsseitige Rechteckregel oder die Trapezregel herangezogen werden. Die nun folgende Tabelle 8.1 fasst die Approximationen zusammen [Lunze (2010b)]:

	Zeitkontinuierlich	Zeitdiskret	Z-Bereich
P-Anteil:	$u(t) = K_p \cdot e(t)$	$u_k = K_p e_k$	$U(z) = K_p E(z)$
I-Anteil:	$u(t) = \frac{1}{T_i} \cdot \int_0^t e(\tau) d\tau$	$u_{k,rect} = \frac{T}{T_i} \cdot \sum_{i=1}^k e_i$ $u_{k,trap} = \frac{T}{T_i} \cdot \left( \frac{e_0}{2} + \sum_{i=1}^{k-1} e_i + \frac{e_k}{2} \right)$	$U_{rect}(z) = \frac{T}{T_i} \cdot \frac{z}{z-1} \cdot E(z)$ $U_{trap}(z) = \frac{T}{2T_i} \cdot \frac{z+1}{z-1}$
D-Anteil:	$u(t) = K_d \dot{e}(t)$	$\Delta u_k = K_d \cdot \frac{e_k - e_{k-1}}{T}$	$U(z) = \frac{K_d}{T} \cdot \frac{z-1}{z} \cdot E(z)$

**Tabelle 8.1:** *Zeitdiskrete Realisierung der PID-Reglerkomponenten. Dabei gilt:  $e_k := e(k \cdot T)$*

Ein gegebener zeitkontinuierlicher PID-Regler kann über die folgende Gleichung zeitdiskret approximiert werden:



$$U(z) = K_p \cdot \left( 1 + \frac{T}{T_i} \cdot \frac{z}{z-1} + \frac{K_d}{T} \cdot \frac{z-1}{z} \right) \cdot E(z) \quad (8.2)$$



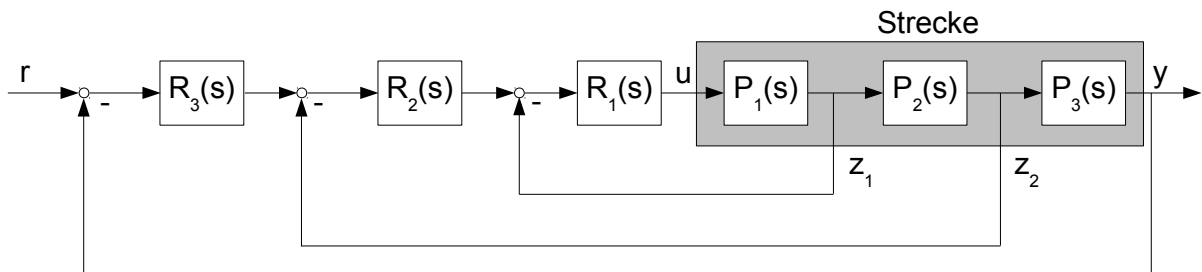
$$u_k = u_{k-1} + K_p \cdot \left( 1 + \frac{T}{T_i} + \frac{K_d}{T} \right) e_k - K_p \cdot \left( 1 + 2 \cdot \frac{K_d}{T} \right) e_{k-1} + K_p \cdot \frac{K_d}{T} \cdot e_{k-2} \quad (8.3)$$

Wenn kein Zusammenhang mit einem zeitkontinuierlichen Entwurf notwendig ist, kann die Differenzgleichung 8.3 auch einfach umgeschrieben werden, indem man die Faktoren vor den Fehlerwerten  $e_k$ ,  $e_{k-1}$  und  $e_{k-2}$  zu Konstanten zusammenfasst [Lunze (2010b)]:

$$u_k = u_{k-1} + c_2 e_k + c_1 e_{k-1} + c_0 e_{k-2} \quad (8.4)$$

## 8.2 Kaskadenregelung

Hohen Anforderungen an die Anregel- und Ausregelzeit, die Überschwingweite oder die Störgrößenunterdrückung kann oftmals mit der weitverbreiteten Standardregelkreisstruktur nicht zufriedenstellend Rechnung getragen werden. Weiters treten immer wieder auch Beschränkungen von anderen Signalen als der Stellgröße auf, was mit dem Standardregelkreis nur bedingt berücksichtigt werden kann. Eine Möglichkeit, diesen Problemen zu begegnen, ist der Einsatz einer Kaskadenregelung. Dabei handelt es sich im Gegensatz zum einschleifigen Standardregelkreis um eine verschachtelte Struktur. Man teilt die Regelstrecke mathematisch in eine Serie von Teilstrecken auf, jede ausgestattet mit einer Rückkopplungsschleife und einem eigenen Regler. Ausgehend von der innersten Strecke entwirft man den ersten Regler. Der innerste Regelkreis bildet dann gemeinsam mit der nachfolgenden Teilstrecke die zweite Strecke, für die ein Entwurf durchgeführt wird. Diese Vorgehensweise wird wiederholt, bis man in der äußersten Schleife angekommen ist [Schröder (2009b)].



**Abbildung 8.3:** Kaskadenstruktur mit drei Rückkopplungsschleifen: Die eigentliche Stellgröße der Strecke ist  $u(t)$ . Die Zwischengrößen  $z_1$  und  $z_2$  können durch Sollwertbegrenzung (eventuell inklusive Anti-Windup) der jeweiligen Regler jedoch ebenfalls beschränkt werden.

Der Einsatz einer Kaskadenregelung bietet folgende Vorteile im Vergleich zu einem einschleifigen Regelkreis [Schröder (2009b)]:

- Durch die Aufteilung der Strecke ist auch der Reglerentwurf schrittweise für die überschaubareren Subsysteme durchzuführen und gestaltet sich daher einfacher.
- Jede Größe, die über einen Regelkreis manipuliert wird, kann begrenzt werden. Auch der Einsatz von Vorsteuerungen ist leicht möglich.

- Ähnlich wie der Entwurf erfolgt auch die Inbetriebnahme schrittweise von innen nach außen. Das Einstellen der Parameter wird somit erleichtert.
- Störgrößen können vom jeweiligen Regler der Teilstrecke ausgeregelt werden und müssen nicht den gesamten Regelkreis durchlaufen. Das Störverhalten ist daher i.A. besser als bei einem einschleifigen Regelkreis.

Der größte Nachteil dieser Regelkreisstruktur ist die von innen nach außen zunehmende Ersatzzeitkonstante. Deshalb sind Kaskadenregler meist langsamer als einschleifige Regelkreise. Grundvoraussetzung für den Einsatz dieser Regelkreisstruktur ist eine mögliche Unterteilung des Gesamtsystems in eine Serienschaltung kleinerer Systeme [Schröder (2009b)].

### 8.3 Symmetrisches Optimum

Das Symmetrische Optimum bezeichnet ein in der elektrischen Antriebstechnik weit verbreitetes Optimierungskriterium für den Bildbereich. Der Name entstammt dem Verlauf des Phasenganges des offenen optimierten Regelkreises, da dieser symmetrisch zur Durchtrittsfrequenz liegt. In erster Linie findet es bei der Parametereinstellung von Drehzahlregelkreisen mit unterlagerten Strom- bzw. Momentenreglern Verwendung. Es lassen sich jedoch auch allgemeine Kriterien, die den Einsatz dieses Verfahrens rechtfertigen, spezifizieren [Schröder (2009b)]:

1. **Strecken mit integrierendem Anteil:** Zur Erreichung stationärer Genauigkeit ist kein integrierender Regler mehr notwendig, außer wenn mit Störungen zwischen Streckenteilen zu rechnen ist (siehe Abbildung 8.4). Dann benötigt man einen Regelkreis mit insgesamt zwei Integralanteilen.
2. **Ungünstiges Zeitkonstantenverhältnis:** Strecken, die sich als Serienschaltung von PT1-Gliedern darstellen lassen, weisen häufig eine große ( $T_1$ ) und mehrere kleine Zeitkonstanten auf, wobei letztere näherungsweise zu einer Summenzeitkonstante  $T_\sigma$  zusammengefasst werden. Um bei Anwendung des Betragsoptimums<sup>1</sup> ein gutes Störverhalten gewährleisten zu können, muss dabei gelten:

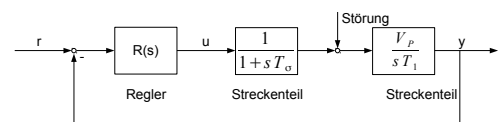
$$T_\sigma < T_1 \leq 4T_\sigma \quad (8.5)$$

Ist diese Bedingung nicht erfüllt, so sind vom Symmetrischen Optimum bessere Ergebnisse zu erwarten.

#### 8.3.1 Einstellung beim offenen Kreis mit zwei Integralanteilen

Die Strecke besteht in diesem Fall aus einem Integralteil und einem PT1-Glied mit kleiner Zeitkonstante  $T_\sigma$  (beispielsweise eine Approximation des unterlagerten Stromregelkreises, Frequenzgang siehe Abbildung 8.5):

$$P(s) = \frac{V_P}{sT_1} \cdot \frac{1}{1 + sT_\sigma} \quad (8.6)$$



**Abbildung 8.4:** Störung, die bei integrierender Strecke einen Reglerintegrator für stationäre Genauigkeit verlangt.

<sup>1</sup>Die Optimierung nach dem Betragsoptimum ist ein weiteres Standardverfahren, das in der Antriebstechnik vor allem zur Einstellung von Stromreglern genutzt wird (siehe [Schröder (2009b)]).

Ein PI-Regler kann als Übertragungsfunktion mit einer reellen Nullstelle und einem Pol bei Null dargestellt werden:

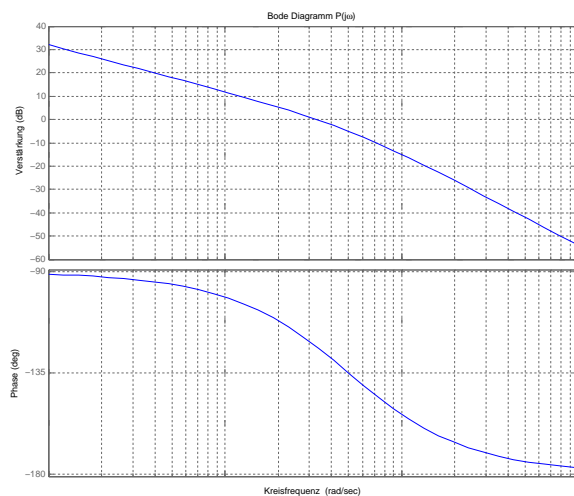
$$R(s) = V_C \cdot \frac{1 + sT_i}{sT_i} \quad (8.7)$$

Ziel des Symmetrischen Optimums ist es, bei der Durchtrittsfrequenz die größte Phasenreserve zu erreichen (siehe Abbildung 8.6). Über den Parameter  $a > 2$  kann auf die Geschwindigkeit und das Überschwingen Einfluss genommen werden. Die Regeln zur Einstellung der Reglerparameter lauten [Schröder (2009b)]:

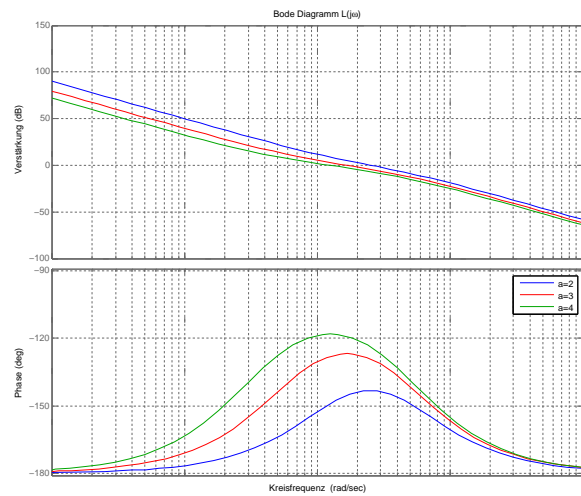
$$T_i = a^2 T_\sigma \quad (8.8a)$$

$$V_C = \frac{T_1}{a V_P T_\sigma} \quad (8.8b)$$

Man erkennt am Frequenzgang des offenen Kreises (Abbildung 8.6) deutlich, dass ein größeres  $a$  zu einer höheren Phasenreserve, jedoch auch zu einer niedrigeren Durchtrittsfrequenz führt.



**Abbildung 8.5:** Bode-Diagramm der Strecke: Aufgrund des Streckenintegrators beginnt der Phasengang bei  $-90^\circ$  und wird durch die Polstelle abermals abgesenkt.

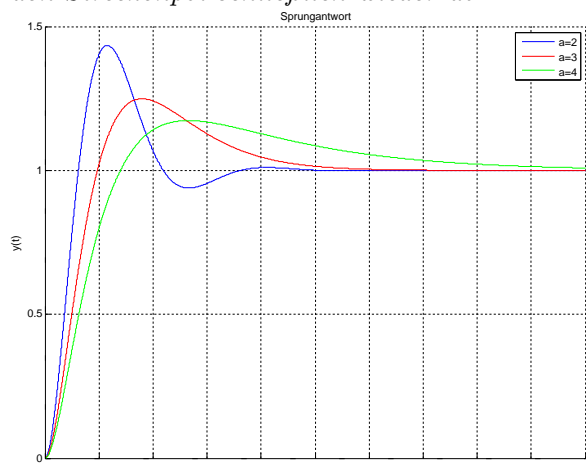


**Abbildung 8.6:** Frequenzgang des offenen Kreises nach der Optimierung: Durch die zwei Integralanteile beginnt der Phasengang des offenen Kreises bei  $-180^\circ$ , wird durch die Nullstelle angehoben und fällt bedingt durch den Streckenpol schließlich wieder ab.

Bei der Standardeinstellung wählt man den Parameter  $a = 2$ , was in einem Überschwingen von etwa 43% resultiert (siehe Abbildung 8.7). Die bei diesem Verfahren entstehende Übertragungsfunktion des geschlossenen Kreises hat die folgende Form:

$$T(s) = \frac{1 + sa^2 T_\sigma}{1 + sa^2 T_\sigma + s^2 a^3 T_\sigma^2 + s^3 a^3 T_\sigma^3} \quad (8.9)$$

Zur Reduktion des Überschwingens wird der Referenzgröße meist ein Verzögerungsglied erster Ordnung als Filter vorgeschaltet, um der Nullstelle des geschlossenen Kreises entgegenzuwirken [Schröder (2009b)].



**Abbildung 8.7:** Sprungantwort des geschlossenen Kreises nach der Optimierung: Je größer  $a$  wird, desto mehr nimmt die Anstiegszeit zu bzw. das Überschwingen ab.

## Filter:

$$F(s) = \frac{1}{1 + sa_{filter}^2 T_\sigma} \quad (8.10)$$

### 8.3.2 Einstellung bei ungünstigem Zeitkonstantenverhältnis

Das Symmetrische Optimum findet auch bei der Serienschaltung von zwei PT1-Gliedern Verwendung, falls die Bedingung 8.5 nicht erfüllt ist und das Betragsoptimum somit entfällt. Dabei ist  $T_1$  wiederum die große Zeitkonstante und  $T_\sigma$  die kleinere (Summen-)Zeitkonstante.

$$P(s) = \frac{V_P}{1 + sT_1} \cdot \frac{1}{1 + sT_\sigma} \quad (8.11)$$

Analog zu Abschnitt 8.3.1 kann ein PI-Regler (siehe Gleichung 8.7) verwendet werden. Damit der geschlossene Kreis auch hier das im Nenner von Gleichung 8.9 auftretende charakteristische Polynom erhält, müssen zwei Korrekturfaktoren  $c_1$  und  $c_2$  eingeführt werden. Die Parameter-einstellung erfolgt dann nach den folgenden Regeln [Schröder (2009b)]:

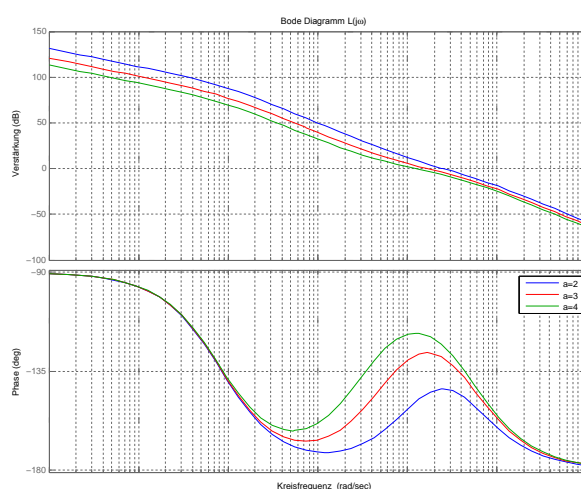
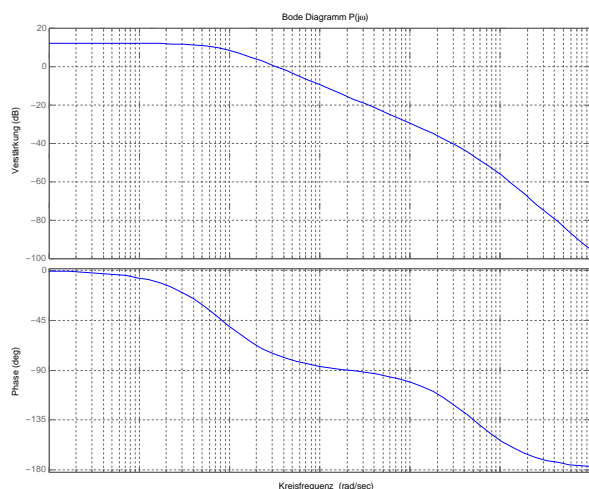
$$T_i = c_1 a^2 T_\sigma \quad (8.12a)$$

$$V_C = \frac{c_2 T_1}{a V_P T_\sigma} \quad (8.12b)$$

$$c_1 = \frac{1 + \left(\frac{T_\sigma}{T_1}\right)^2}{\left(1 + \frac{T_\sigma}{T_1}\right)^3} \quad (8.13a)$$

$$c_2 = 1 + \left(\frac{T_\sigma}{T_1}\right)^2 \quad (8.13b)$$

Die Frequenzgänge der Strecke und des optimierten offenen Kreises sind in den beiden folgenden Abbildungen dargestellt.



**Abbildung 8.8:** Bode-Diagramm der Strecke: Die beiden Verzögerungsglieder sorgen zweimal für eine Phasenabsenkung.

**Abbildung 8.9:** Frequenzgang des offenen Kreises nach Optimierung: Wegen des Reglerintegralanteiles beginnt der Phasengang des offenen Kreises bei  $-90^\circ$ . Durch die Reglernullstelle kommt es zur Phasenhebung bei der Durchtrittsfrequenz.

Auch hier kann das Überschwingen beim Aufschalten einer Referenzsprungfunktion durch die Verwendung eines Vorfilters verringert werden.

## 8.4 Sliding Mode Control

Die standardmäßige Vorgehensweise zur Regelung elektrischer Maschinen basiert auf den in den vorangegangenen Abschnitten erläuterten Konzepten von PI(D)-Regelung und Kaskadenstruktur. Zur systematischen Einstellung des Reglers ist die Kenntnis der Motorparameter unumgänglich. Häufig sind diese Größen jedoch nicht exakt bekannt und unterliegen zudem großen Schwankungen während des Betriebs (siehe Abschnitt 7). In solchen Situationen kann der Einsatz von robusten Regelungsverfahren Abhilfe schaffen. Sliding Mode Regler gehören zu dieser Gruppe und überzeugen vor allem durch die Einfachheit ihrer Implementierung sowie hohe Genauigkeit und Robustheit. Infolgedessen ist es nicht verwunderlich, dass dieses Konzept bereits mehrfach erfolgreich bei der Regelung von elektrischen Maschinen Anwendung fand [Pisano u. a. (2008) und Hung u. a. (1993)]. Aufbauend auf der ursprünglichen Form dieses Verfahrens, den First Order Sliding Mode Reglern (1-SMC), wurden Regelungen basierend auf sogenannten *Higher-Order-Sliding-Modes* entwickelt, zu denen auch die Second Order Sliding Mode Algorithmen (2-SMC) zählen.

### 8.4.1 First Order Sliding Mode Control (1-SMC)

Die Ursprünge dieser Technik liegen bereits in den 1950er Jahren. Damals diente 1-SMC zur Stabilisierung von linearen Systemen zweiter Ordnung. Durch konsequente Weiterentwicklung ist es heute möglich, dieses Verfahren für verschiedenste Regelungsaufgaben bei linearen, nicht-linearen, MIMO-, zeitdiskreten und anderen Systemen einzusetzen. Sliding Mode Control ist eine Form der strukturvariablen Regelung (engl. Variable Structure Control, VSC), d.h. es wird abhängig vom Systemzustand zwischen verschiedenen Regelgesetzen umgeschaltet. Neben der einfachen Implementierung ist vor allem die hohe Robustheit gegenüber Unsicherheiten der Streckenparameter und Störeinflüssen ein Vorteil dieser Verfahren [Hung u. a. (1993)].

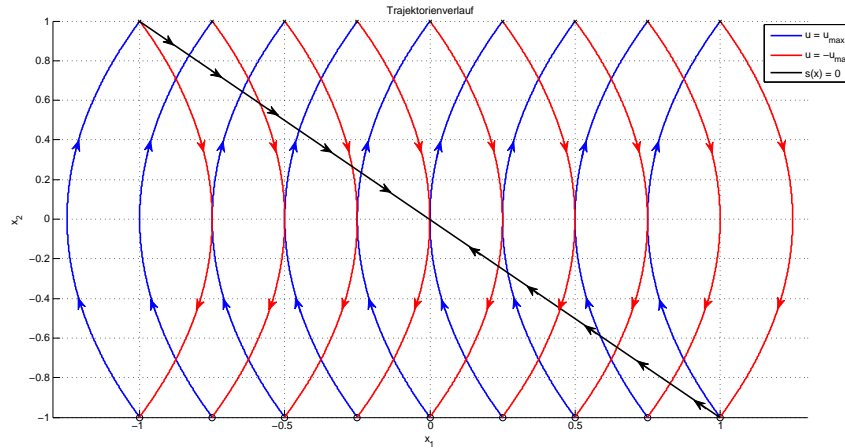
#### 8.4.1.1 Erläuterung anhand eines Beispiels

Die Funktionsweise von Sliding Mode Regelungen lässt sich am einfachsten anhand eines Beispiels (entnommen aus Adamy (2009)) demonstrieren. Die Strecke mit Regler sei wie folgt definiert:

$$\dot{x}_1 = x_2 \quad (8.14a) \quad u = -u_{max} \text{sign}(x_2 + mx_1) \quad (8.15)$$

$$\dot{x}_2 = au \text{ mit } a > 0 \quad (8.14b)$$

Der Zustandsraum wird durch die Schaltgerade  $s(\mathbf{x}) = x_2 + mx_1 = 0$  in zwei Regionen geteilt, wobei unterhalb der Geraden  $u = u_{max}$  gilt und oberhalb  $u = -u_{max}$ . Betrachtet man den Trajektorienverlauf für die beiden Eingangsgrößen und zeichnet zusätzlich die Funktion  $s(\mathbf{x})$  in der Zustandsebene ein, so erkennt man, dass durch diese Wahl der Stellgröße die Trajektorien von beiden Seiten der Gerade auf selbige zulaufen (siehe Abbildung 8.10). Nach dem Auftreffen des Systems auf die Schaltlinie  $s(\mathbf{x}) = 0$  wechselt das System theoretisch unendlich schnell und mit infinitesimal kleiner Auslenkung von einer Seite zur anderen, indem die Stellgröße ständig zwischen  $u_{max}$  und  $-u_{max}$  umgeschaltet wird. Der Zustandsvektor *gleitet* (engl. *slides*) dadurch in die Ruhelage bei  $\mathbf{x} = 0$  [Adamy (2009)]. Die Bewegung entlang der Geraden  $s(\mathbf{x}) = 0$  wird als *Sliding Mode* bezeichnet, die Gerade selbst nennt man *Switching Surface* oder *Switching Manifold*. Die Systembewegung vor dem Erreichen der Schaltlinie heißt *Reaching Mode*. Wichtige Eigenschaften des geregelten Systems nach 8.14a, 8.14b und 8.15 sind [Hung u. a. (1993)]:



**Abbildung 8.10:** Trajektorienverlauf des Systems nach 8.14a, 8.14b und 8.15: Die roten Trajektorien entstehen bei  $u = -u_{max}$ , die blauen bei  $u = u_{max}$  und die schwarze Linie entspricht  $s(\mathbf{x}) = 0$ .

1. Die Dynamik im Sliding Mode, also während  $s(\mathbf{x}) = 0$ , wird von der Switching Surface selbst vorgegeben. Die Funktion beschreibt somit gleichzeitig einen Bereich im Zustandsraum und eine Bewegung:

$$s(\mathbf{x}) = x_2 + mx_1 = \dot{x}_1 + mx_1 = 0 \Rightarrow \dot{x}_1 = -mx_1$$

2. Im Sliding Mode ist die Systemdynamik unabhängig von den Parametern der Strecke. Dadurch lässt sich die Robustheit bezüglich Parameterunsicherheiten erklären.
3. Die Systemordnung im Sliding Mode ist reduziert im Vergleich zu jener des Ausgangssystems.
4. Der Sliding Mode ist keine Trajektorie eines der beiden Systeme, zwischen denen die Umschaltung erfolgt.

#### 8.4.1.2 Problemdefinition

$$\dot{\mathbf{x}} = A(\mathbf{x}, t) + B(\mathbf{x}, t)\mathbf{u}(\mathbf{x}, t) \quad (8.16) \quad \mathbf{u}(\mathbf{x}, t) = \begin{cases} \mathbf{u}^+(\mathbf{x}, t) & \text{wenn } s(\mathbf{x}) > 0 \\ \mathbf{u}^-(\mathbf{x}, t) & \text{wenn } s(\mathbf{x}) < 0 \end{cases} \quad (8.17)$$

Ein mittels 1-SMC geregeltes System weist eine Zustandsraumdarstellung gemäß den Gleichungen 8.16 und 8.17 auf<sup>2</sup>. Das System hat  $n$  Zustandsvariablen und  $m$  Eingangsgrößen. Der Entwurf einer Sliding Mode Regelung besteht im Wesentlichen aus zwei Schritten [Hung u. a. (1993)]:

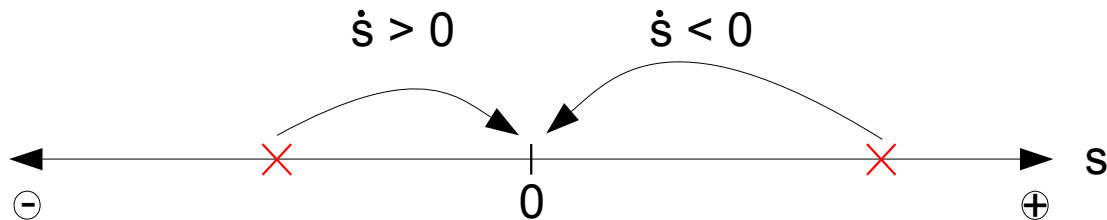
1. Zunächst müssen  $m$  Switching Surfaces  $s(\mathbf{x}) = 0$  ausgewählt werden, die die gewünschte Dynamik im Sliding Mode beschreiben.
2. Anschließend ist sicherzustellen, dass durch geeignete Wahl der Stellgrößen  $\mathbf{u}(\mathbf{x}, t)$  ein beliebiger Anfangszustand *in endlicher Zeit* auf die Switching Surface geführt wird. Daraufhin erfolgt die Bewegung in die Ruhelage bei  $\mathbf{x} = 0$ , das System ist somit global asymptotisch stabil.

<sup>2</sup>Systeme allgemeinerer Form  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$  können in einigen Fällen mittels Zustandstransformation in diese Form gebracht werden.

### 8.4.1.3 Reaching Condition

Um zu gewährleisten, dass das System von jedem beliebigen Anfangszustand aus auf die Switching Surface gelangt, muss die sogenannte *Reaching Condition* eingehalten werden. Eine notwendige Bedingung für ein Auftreten eines Sliding Modes ist [Hung u. a. (1993)]:

$$s_i \cdot \dot{s}_i < 0 \text{ mit } i = 1..m \quad (8.18)$$



**Abbildung 8.11:** *Reaching Condition:* Damit sich das System im Sliding Mode befindet, muss  $s(\mathbf{x}) = 0$  gelten, was durch eine Ableitung mit umgekehrtem Vorzeichen erreicht wird.

Diese Ungleichung stellt zwar ein Erreichen der Switching Surface sicher, kann dies in endlicher Zeit jedoch nicht garantieren. Aus diesem Grund kann mittels *Reaching Law Approach* für die Einhaltung der Reaching Condition bei *endlicher* Einstellzeit gesorgt werden, bei gleichzeitiger Festlegung der Dynamik im Reaching Mode. Die zeitliche Ableitung der Sliding Surface wird hier direkt festgelegt. Drei häufig eingesetzte Varianten sind [Hung u. a. (1993)]:

1. **Constant Rate Reaching Law:**

$$\dot{s} = -Q \text{sign}(s) \quad (8.19)$$

2. **Constant Plus Proportional Rate Reaching Law:**

$$\dot{s} = -Q \text{sign}(s) - Ks \quad (8.20)$$

3. **Power Rate Reaching Law:**

$$\dot{s} = -k|s_i|^\alpha \text{sign}(s_i) \text{ mit } i = 1..m \quad (8.21)$$

### 8.4.1.4 Stellgröße

Bei Verwendung des Reaching Law Approach kann die Stellgröße für ein *lineares* System  $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$  sofort aus der gewählten Funktion für  $\dot{s}$  berechnet werden (siehe Gleichung 8.22). Nichtlineare Systeme der Form 8.16 müssen über Zustandstransformationen in spezielle Normalformen gebracht werden [Hung u. a. (1993)].

$$\mathbf{u}(\mathbf{x}) = \left( \frac{\partial \mathbf{s}}{\partial \mathbf{x}} B \right)^{-1} \left[ \dot{s} - \frac{\partial \mathbf{s}}{\partial \mathbf{x}} A \mathbf{x} \right] \quad (8.22)$$

Die im anfänglichen Beispiel verwendete Stellgröße mit idealer Relayfunktion, die ausschließlich zwischen maximalem und minimalem Wert wechselt, ist also nur eine mögliche Realisierungsform von Sliding-Mode Control. In Fällen mit nicht festgelegter Reglerstruktur können auch Kombinationen von kontinuierlichen und nichtkontinuierlichen Funktionen als Stellsignale dienen.

### 8.4.1.5 Rattern

Die grundlegende Voraussetzung für die Herleitung der Sliding Mode Control Theorie ist die Fähigkeit, die Stellgröße unendlich schnell von einem Wert zu einem anderen umschalten zu können. Nur dadurch kann der sogenannte *Ideal Sliding Mode*, also  $s(\mathbf{x}) = 0$ , erreicht werden. Bei realen Strecken kann diese Forderung nicht erfüllt werden, da es sowohl zu Verzögerungen durch die Stellglieder, als auch durch die Berechnung der Stellgröße kommt. Bei den heute weit verbreiteten zeitdiskreten Reglern mittels Mikrocontroller ist dieser zeitliche Verzug inhärent. Durch diese Unzulänglichkeiten in der praktischen Realisierung der Regler entsteht im Sliding Mode und um die Ruhelage eine hochfrequente Schwingung, die als *Rattern* (engl. *chattering*) bezeichnet wird. Dieses kann wiederum nicht modellierte hochfrequente Dynamiken des Systems anregen und dadurch noch verstärkt werden. [Hung u. a. (1993)]. Um diesen unerwünschten Nebeneffekt zu vermeiden, haben sich verschiedene Techniken etabliert:

#### 1. Kontinuierliche Approximation

Da eine ideale Relayfunktion ohne Verzögerungen nicht zu realisieren ist, besteht ein Ansatz zur Reduktion des Ratterns darin, diese durch eine Sättigungsfunktion zu ersetzen. Durch diese Maßnahme entsteht um die Switching Surface ein Bereich, in dem die Stellgröße nur noch rein kontinuierlicher Natur ist. Das Rattern wird dadurch zwar reduziert, jedoch leidet auch die Robustheit des Reglers darunter. Die Sättigungsfunktion ist wie folgt definiert [Hung u. a. (1993)]:

$$\text{sat}(s, \Phi) = \begin{cases} 1 & \text{wenn } s > \Phi \\ \frac{s}{\Phi} & \text{wenn } |s| \leq \Phi \\ -1 & \text{wenn } s < -\Phi \end{cases} \quad (8.23)$$

#### 2. Anpassen des Reaching Law

Ein weitere Möglichkeit, das Rattern zu reduzieren, ist eine geschickte Wahl des Reaching Law. Beispielsweise kann es hilfreich sein, das Constant plus Proportional Reaching Law zu verwenden und dabei die Faktoren vor der Signumfunktion eher klein zu wählen, die Proportionalitätsfaktoren hingegen etwas zu vergrößern. Einen ähnlichen Effekt hat die Verwendung des Power Rate Reaching Law. Beide Vorgehensweisen zielen darauf ab, die Stellaktivität in der Nähe der Switching Surface zu reduzieren, wie es auch bei der kontinuierlichen Approximation der Fall ist [Hung u. a. (1993)].

#### 3. Higher Order Sliding Modes

Hierbei handelt es sich um einen im Vergleich zur klassischen 1-SMC Technik recht jungen Ansatz zur Vermeidung von Rattern. Im Gegensatz zu den in den Punkten 1 bis 2 beschriebenen Vorgehensweisen geht dabei die Robustheit jedoch nicht verloren. Zunächst muss geklärt werden, was unter dem Begriff *Sliding Order* zu verstehen ist: Die Sliding Order ist die Anzahl von stetigen Ableitungen der Funktion  $s(\mathbf{x})$  inklusive der nullten Ableitung (also der Funktion selbst). Bei einer Sliding Order von  $r$  sind somit  $s, \dot{s}, \dots, s^{(r-1)}$  stetige Funktionen und erst die  $r$ -te Ableitung enthält Unstetigkeiten. Unter einem Higher Order Sliding Mode (HOSM) der Ordnung  $r$  versteht man nun eine Systembewegung, die durch die folgenden Gleichungen beschrieben wird [Fridman und Levant (2002)]:

$$s = \dot{s} = \dots = s^{(r-1)} = 0 \quad (8.24)$$

Da beim klassischen Verfahren bereits  $\dot{s}$  Unstetigkeiten enthält, nennt man es in diesem Zusammenhang First Order Sliding Mode Control. Der große Vorteil von HOSM ist die

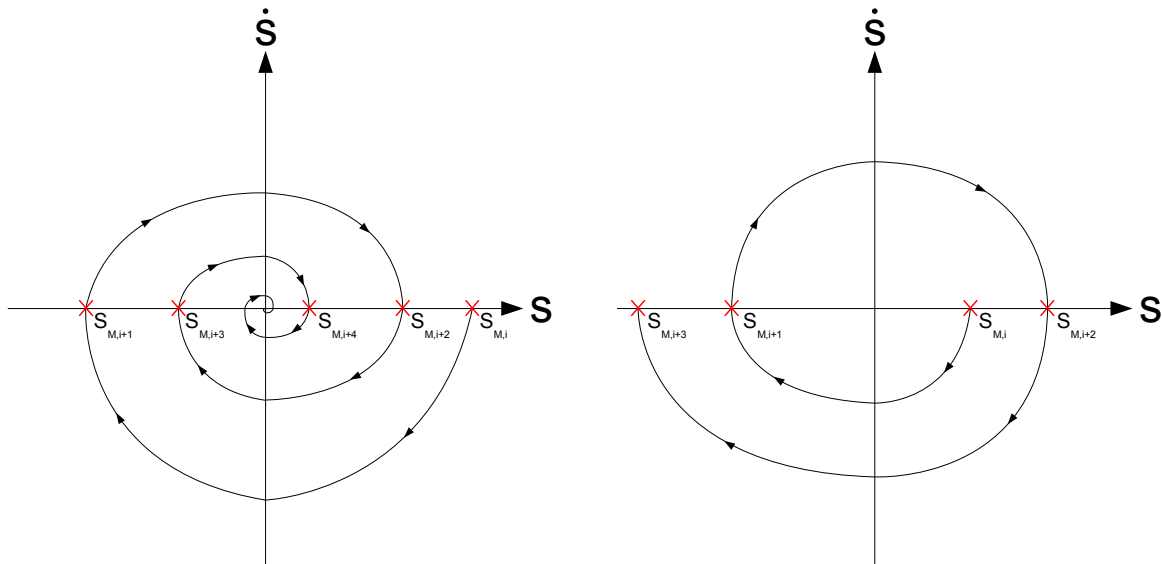


theoretische Eliminierung des Ratterns durch die Verschiebung der nichtkontinuierlichen Stellaktivität auf höhere Ableitungen des Sliding Modes [Fridman und Levant (2002)].

### 8.4.2 Second Order Sliding Mode Control (2-SMC)

Im Gegensatz zu derjenigen des klassischen Sliding Mode Control Ansatzes besteht die Zielsetzung bei Second Order Sliding Mode Control (2-SMC) darin, einen Trajektorienverlauf zu erzielen, der nicht nur durch die Gleichung der Switching Surface  $s(\mathbf{x}) = 0$  bestimmt wird, sondern zusätzlich der Bedingung  $\dot{s}(\mathbf{x}) = 0$  genügt. Anders formuliert ist es die Aufgabe des Reglers, ein *Hilfssystem* mit den beiden Zustandsvariablen  $y_1 = s(\mathbf{x})$  und  $y_2 = \dot{s}(\mathbf{x})$  im Ursprung zu stabilisieren. Um dies sicherzustellen, kann man analog zur Reaching Condition bei 1-SMC Bedingungen in Form von Differentialgleichungen definieren. Durch folgende Forderung (8.25) erreicht man einen sich im Uhrzeigersinn drehenden Trajektorienverlauf in der  $s - \dot{s}$ -Ebene. Dieser allein garantiert jedoch keineswegs Stabilität, sondern nur, dass  $s(\mathbf{x})$  eine Reihe von Extremwerten ( $s_{M,i}$ ) durchläuft, für die naturgemäß stets  $\dot{s}(\mathbf{x}) = 0$  gilt (die Extremwerte liegen also auf der Abszisse, siehe Abbildungen 8.12 und 8.13) [Pisano (2000)].

$$\ddot{s} \leq -\xi(t)|s| \text{ mit } \xi(t) \geq \xi_1 > 0 \quad (8.25)$$



**Abbildung 8.12:** *Stabiler Trajektorienverlauf als Resultat von Ungleichung 8.25.* **Abbildung 8.13:** *Instabiler Trajektorienverlauf trotz Einhaltung von Ungleichung 8.25.*

Simultan zu 1-SMC lautet nun die Forderung, dass sowohl  $s$  als auch  $\dot{s}$  in endlicher Zeit zu Null werden. Eine hinreichende Bedingung dafür (gemeinsam mit Ungleichung 8.25) ist, dass der Betrag und die Zeitspanne zwischen den aufeinanderfolgenden Extremwerten  $s_{M,i}$  abnehmen. Zusammenfassend werden diese Forderungen auch als *Contraction Properties* bezeichnet. Nachdem nun die Bedingungen für ein Erreichen des Sliding Modes  $s = \dot{s} = 0$  erläutert wurden, muss daraus ein Regelgesetz abgeleitet werden. Systemdynamik der Strecke und die erste zeitliche Ableitung der Switching Surface  $s(\mathbf{x})$  seien wie folgt definiert [Pisano (2000)]:

$$\dot{\mathbf{x}} = a(\mathbf{x}, t) + b(\mathbf{x}, t)u(t) \quad (8.26) \quad \dot{s}(\mathbf{x}) = \frac{\partial s}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial s}{\partial \mathbf{x}} a(\mathbf{x}, t) + \frac{\partial s}{\partial \mathbf{x}} b(\mathbf{x}, t)u(t) \quad (8.27)$$

An dieser Stelle muss nun anhand des relativen Grades der Sliding Surface zwischen zwei Fällen unterschieden werden:

1. Relativer Grad ist  $r = 1$ :  $\frac{\partial s}{\partial \mathbf{x}} b(\mathbf{x}, t) \neq 0$
2. Relativer Grad ist  $r = 2$ :  $\frac{\partial s}{\partial \mathbf{x}} b(\mathbf{x}, t) = 0$

Der erste Fall wird auch als *Anti Chattering Case* bezeichnet, da er prinzipiell auch mit einem 1-SMC Ansatz zu lösen wäre. Der Einsatz eines Second Order Sliding Mode Reglers kann jedoch als Maßnahme zur Reduktion des Ratterns sinnvoll sein. Die Stellgröße der Strecke  $u(t)$  ist in diesem Fall eine *kontinuierliche* Funktion. Der 2-SMC Regler erzeugt hier zwar ein *nicht kontinuierliches* Signal, dieses durchläuft jedoch einen Integrator, bevor es auf das System aufgeschaltet wird. Die eigentliche Stellgröße ist somit  $\dot{u}(t)$ . Beim zweiten Fall ist ein 1-SMC Einsatz nicht durchführbar, da die Stellgröße auf  $\dot{s}$  keinen direkten Einfluss hat. Die durch den Regler zu stabilisierenden Hilfssysteme für die beiden relativen Grade lauten [Fridman und Levant (2002), Pisano (2000)]:

Zustandsvariablen in beiden Fällen:  $y_1 = s(\mathbf{x})$ ,  $y_2 = \dot{s}(\mathbf{x})$

**Anti Chattering:**

**Relative Degree Two:**

$$\dot{y}_1 = y_2(\mathbf{x}, u, t) \quad (8.28a) \quad \dot{y}_1 = y_2(\mathbf{x}, t) \quad (8.29a)$$

$$\dot{y}_2 = \varphi(\mathbf{x}, u, t) + \gamma(\mathbf{x}, t)v(t) \quad (8.28b) \quad \dot{y}_2 = \varphi(\mathbf{x}, t) + \gamma(\mathbf{x}, t)v(t) \quad (8.29b)$$

$$v(t) = \dot{u}(t) \quad (8.28c) \quad v(t) = u(t) \quad (8.29c)$$

Da es sich auch bei 2-SMC um ein robustes Regelungsverfahren handelt, dürfen die Parameter der oben genannten Systeme Unsicherheiten innerhalb gewisser Grenzen aufweisen:

$$\Phi > 0, |\varphi| \leq \Phi, 0 < \Gamma_m \leq \gamma \leq \Gamma_M \quad (8.30)$$

Ein Regelgesetz, das die Contraction Properties erfüllt und in endlicher Zeit zum Sliding Mode  $s = \dot{s} = 0$  führt, lautet in seiner *verallgemeinerten* Form [Bartolini u. a. (2003)]:

$$\begin{aligned} v(t) &= -\alpha(t)U \operatorname{sign}(y_1 - \beta s_M) \\ \alpha(t) &= \begin{cases} 1 & \text{wenn } (y_1 - \beta s_M)s_M \geq 0 \\ \alpha^* & \text{wenn } (y_1 - \beta s_M)s_M < 0 \end{cases} \\ \beta &\in [0; 1) \\ U &> 0 \\ \alpha^* &> 1 \end{aligned} \quad (8.31)$$

Die Konstante  $U$  ist so zu wählen, dass die Stellgröße in jedem Fall das Vorzeichen von  $\dot{y}_2 = \ddot{s}$  dominiert. Die Variable  $s_M$  ist der zuletzt aufgetretene Extremwert von  $s$ , sprich der Wert von  $s$  beim Schneiden der Abszisse der  $s - \dot{s}$ -Ebene. Die beiden Parameter  $\alpha^*$  (*modulation factor*) und  $\beta$  (*anticipation factor*) stellen ein Erreichen des Koordinatenursprungs sicher und bestimmen folglich auch die Konvergenzrate. Die empfohlene Vorgehensweise ist, zuerst  $U$  und  $\alpha^*$  einzustellen, anschließend kann über  $\beta$  der Maximalwert von  $|\dot{s}|$  reduziert werden (d.h. der Regler reagiert „langsamer“, der Reaching Mode dauert länger). Bei einer Wahl von  $\beta \rightarrow 1$

gilt  $|\dot{s}| \rightarrow 0$ . Zwei bekannte Spezialfälle dieses verallgemeinerten Algorithmus sind der *Twisting Algorithm* und der *Sub-Optimal Algorithm* [Bartolini u. a. (2003)].

### 8.4.2.1 Twisting Algorithm

Dies ist einer der ersten beschriebenen Second Order Sliding Mode Regler, der seinen Namen dem typischen Trajektorienverlauf in der  $s - \dot{s}$ -Ebene verdankt. Die Trajektorien laufen spiralförmig vom Anfangszustand in den Koordinatenursprung ein (siehe Abbildung 8.14). Dieser Algorithmus ist eine Variante des verallgemeinerten Reglers 8.31 und entsteht durch eine Parameterwahl von  $\beta = 0$  [Bartolini u. a. (2003), Fridman und Levant (2002)]:

$$v(t) = \begin{cases} -U \operatorname{sign}(y_1) & \text{wenn } y_1 s_M \geq 0 \\ -\alpha^* U \operatorname{sign}(y_1) & \text{wenn } y_1 s_M < 0 \end{cases} = \begin{cases} -U \operatorname{sign}(y_1) & \text{wenn } y_1 y_2 \leq 0 \\ -\alpha^* U \operatorname{sign}(y_1) & \text{wenn } y_1 y_2 > 0 \end{cases} \quad (8.32)$$

Die Bedingungen für ein Erreichen des Sliding Modes bei  $y_1 = y_2 = 0$  (bzw.  $s = \dot{s} = 0$ ) in endlicher Zeit lauten [Bartolini u. a. (2003)]:

$$\begin{aligned} U &> \frac{\Phi}{\Gamma_m} \\ \alpha^* &\geq \frac{2\Phi + \Gamma_M U}{\Gamma_m U} \end{aligned} \quad (8.33)$$

Wie in Gleichung 8.32 ersichtlich ist, benötigt der Twisting Algorithm entweder die Vorzeicheninformation der Variable  $\dot{s}$  bzw.  $y_2$  oder jene des letzten Extremwertes von  $s$  bzw.  $y_1$ , wenn man ihn als Spezialfall des verallgemeinerten Algorithmus betrachtet.

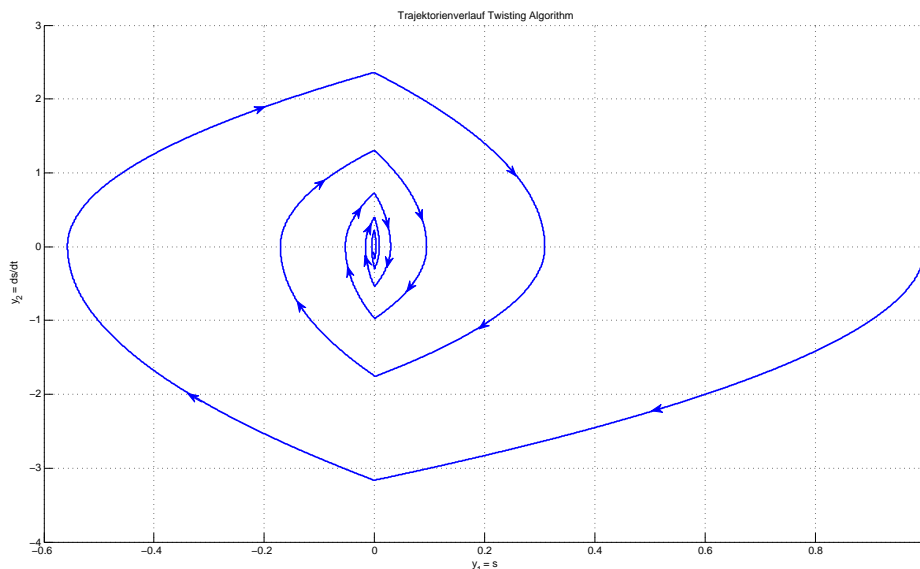


Abbildung 8.14: Trajektorienverlauf des Twisting Algorithm.

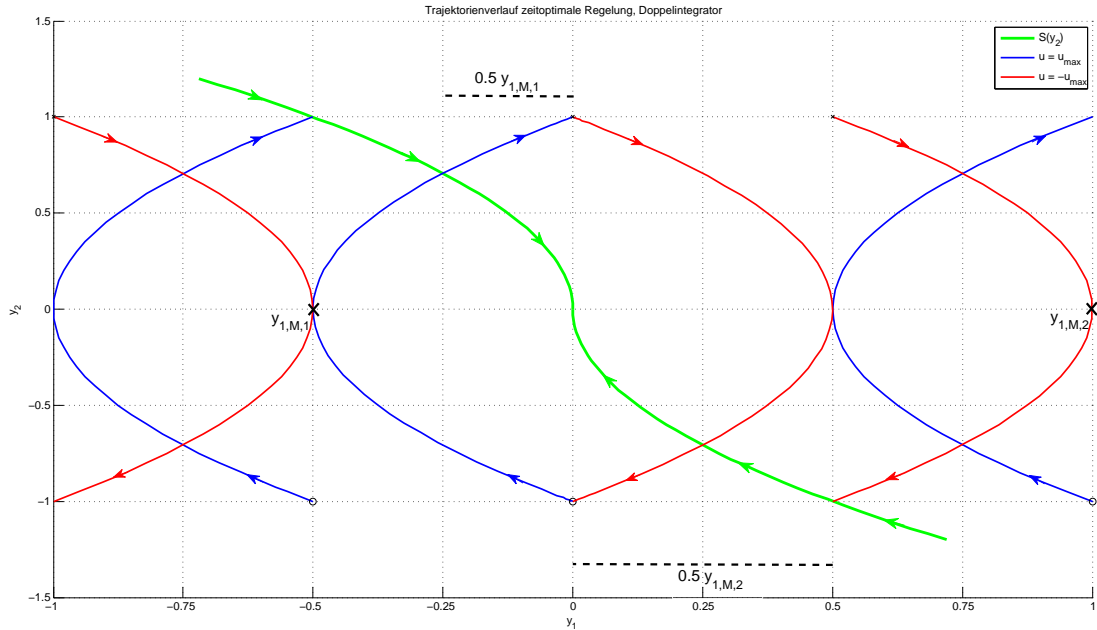
Die Idee hinter dem Twisting Algorithm ist es, abhängig vom aktuellen Quadranten in der  $s - \dot{s}$ -Ebene unterschiedlich hohe zweite Ableitungen  $\ddot{s}$  zu „erzeugen“:

Quadranten I & III: Das Vorzeichen von  $\dot{s}$  führt  $s$  in diesen Bereichen weiter weg vom Ursprung. Durch einen großen Betrag von  $\ddot{s}$  sollen diese Quadranten möglichst schnell verlassen werden.

Quadranten II & IV: Die Variable  $s$  bewegt sich hier in Richtung  $s = 0$ . Der Betrag von  $\dot{s}$  wird hier etwas kleiner gewählt, damit  $\dot{s}$  nicht zu groß wird. Dies wäre im nachfolgenden Quadranten (I oder III) nämlich kontraproduktiv.

### 8.4.2.2 Sub-Optimal Algorithm

Dieser Algorithmus basiert auf der zeitoptimalen Regelung eines Doppelintegrators und kann ebenfalls auf das verallgemeinerte Verfahren zurückgeführt werden, indem der Anticipation Factor zu  $\beta = 0.5$  gewählt wird [Bartolini u. a. (2003)].



**Abbildung 8.15:** Trajektorienverlauf einer zeitoptimalen Doppelintegratorregelung: Die grüne Trajektorie zeigt die Schaltfunktion  $S(y_2)$ , blaue und rote entstehen durch  $u = \pm u_{max}$ . Die beiden Punkte  $y_{1,M,1}$  und  $y_{1,M,2}$  sind Extremwerte der Zustandsvariable  $y_1$  bei unterschiedlichen Anfangszuständen. Der ideale Umschaltzeitpunkt, um in den Koordinatenursprung zu gelangen, liegt genau bei  $0.5 \cdot y_{1,M,i}$ .

Am anschaulichsten lässt sich die Funktionsweise erklären, wenn man davon ausgeht, dass das Hilfssystem 8.29 einem Doppelintegrator entspricht (d.h.  $\varphi = 0$  und  $\gamma = 1$ ). Ein zeitoptimaler Regler, der dieses System in den Ursprung führt, berechnet die Stellgröße anhand folgender Formel [Adamy (2009)]:

$$y_1 = S(y_2) = -\frac{1}{2}y_2|y_2| \quad (8.34a)$$

$$u(t) = \begin{cases} -u_{max} & \text{wenn } y_1 > S(y_2) \wedge (y_1 = S(y_2) \vee y_1 < 0) \\ u_{max} & \text{wenn } y_1 < S(y_2) \wedge (y_1 = S(y_2) \vee y_1 > 0) \end{cases} \quad (8.34b)$$

Für die Umsetzung dieses Regelgesetzes muss also die Zustandsgröße  $y_2$  (entspricht  $\dot{s}(\mathbf{x})$ ) messbar sein. Da sie das aber nicht ist, besteht die grundlegende Idee nun darin, den Umschaltzeitpunkt nur über  $y_1$  zu bestimmen. Unter der Annahme, dass die Extremwerte  $y_{1,M,i}$  bekannt sind,

erreicht man die optimale Trajektorie genau bei  $y_1 = \frac{1}{2}y_{1,M,i}$ . Die Extremwerte  $y_{1,M,i}$  können mithilfe des im Abschnitt 8.4.2.3 beschriebenen *Peak Detectors* erkannt werden, der Startwert ist  $y_1(t = 0)$ . Aus diesem Grund entspricht der Algorithmus bei Anfangszuständen in den Quadranten I und III der zeitoptimalen Lösung, schließlich wird in diesen Fällen die Achse  $y_2 = 0$  durchlaufen und der korrekte Wert  $y_{1,M}$  kann erfasst werden. Bei Startwerten in den anderen beiden Quadranten erreicht man die Schaltfunktion, ohne zuvor  $y_2 = 0$  durchlaufen zu haben, die Schätzung für  $y_{1,M}$  ist hier i.A. falsch. Durch die spätere Umschaltung gelangt man in die Quadranten I oder III, von wo aus der richtige Umschaltzeitpunkt bestimmt werden kann. Da in diesen Quadranten eine Umschaltung mehr als im Zuge der zeitoptimalen Regelung durchgeführt wird, benötigt der Regler hier etwas länger als die zeitoptimale Variante. Zumal beim tatsächlichen System 8.29 jedoch  $\varphi \neq 0$  und  $\gamma \neq 1$  gilt, müssen die Reglerparameter anhand von Ungleichungsbedingungen für ein vergleichbares Verhalten sorgen. In diesem Fall werden eine bzw. zwei Umschaltungen i.A. nicht genügen. Abhängig von den gewählten Parametern  $U$  und  $\alpha^*$  winden sich die Trajektorien um den Ursprung (ähnlich dem Twisting Algorithm) oder  $y_1$  nähert sich diesem monoton an. Die Konvergenzbedingungen des Algorithmus lauten [Bartolini u. a. (2003)]:

$$\begin{aligned} U &> \frac{\Phi}{\Gamma_m} \\ \alpha^* &\geq \frac{2\Phi + \Gamma_M U}{\Gamma_m U} \end{aligned} \quad (8.35)$$

### 8.4.2.3 Peak-Detection

Zentraler Bestandteil der bisher angeführten 2-SMC Algorithmen ist die Erkennung von Extremwerten der Variable  $y_1 = s(\mathbf{x})$ . Eine Möglichkeit zur Detektion bietet der folgende Algorithmus, der auf jenem in Pisano (2000) bzw. Pisano u. a. (2008) basiert:

```

 $x_{-N} = x_{-2N} = x_0;$ 
 $x_{M,0} = x_0;$ 
 $i = 0;$ 

 $\Delta_k = (x_k - x_{k-N})(x_{k-N} - x_{k-2N});$ 

if ( $\Delta_k < 0$ )
{
   $i = i + 1;$ 
   $x_{M,i} = x_{k-N};$ 
}

```

Bei einem lokalen Minimum oder Maximum ( $\dot{x} = 0$ ) haben die beiden Faktoren zur Berechnung von  $\Delta_k$  unterschiedliche Vorzeichen, sodass sich zwischen den Werten, aus denen sich die Faktoren zusammensetzen, ein Extremwert befinden muss. Der Parameter  $N > 0$  soll die Rauschempfindlichkeit verringern. Bei einem rauschfreien Signal  $x$  wäre  $N = 1$  die richtige Wahl.

### 8.4.2.4 Super-Twisting Algorithm

Diese Variante des 2-SMC-Prinzips wurde speziell für Systeme mit einem relativem Grad von  $r = 1$  entwickelt, um Rattern zu verhindern. Der Trajektorienverlauf in der  $s - \dot{s}$ -Ebene weist einen ähnlich Verlauf wie jener des Twisting Algorithms auf. Das Regelgesetz besteht aus zwei zeitkontinuierlichen Anteilen [Fridman und Levant (2002)]:

$$u(t) = -\lambda|y_1|^\rho \text{sign}(y_1) + u_1(t) \quad (8.36a)$$

$$\dot{u}_1 = -U \text{sign}(y_1) \quad (8.36b)$$

Im Gegensatz zu den anderen beiden vorgestellten Algorithmen benötigt der Super Twisting Algorithmus keinerlei Informationen über die zeitliche Ableitung der Sliding Surface. Mit einer Wahl von  $\rho = \frac{1}{2}$  und unter Berücksichtigung der folgenden Bedingungen wird  $s = \dot{s} = 0$  in endlicher Zeit erreicht [Fridman und Levant (2002)]:

$$\begin{aligned} U &> \frac{\Phi}{\Gamma_m} \\ \lambda^2 &\geq \frac{4\Phi}{\Gamma_m^2} \frac{\Gamma_M(U + \Phi)}{\Gamma_m(U - Phi)} \end{aligned} \quad (8.37)$$

# 9 Implementierung Strom- und Momentenregelung

## 9.1 I-Stromregler

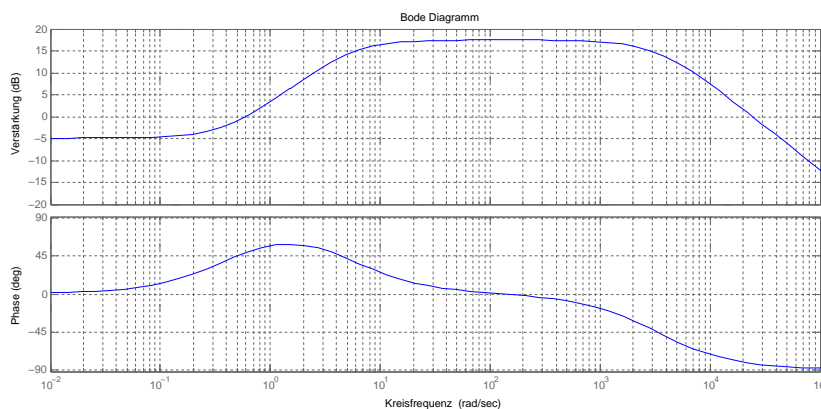
Der in diesem Abschnitt entworfene Stromregler kommt sowohl bei der kaskadierten Geschwindigkeitsregelung im inneren Kreis, als auch bei der Momentenregelung der Gleichstrommaschine zum Einsatz. Für den Entwurf wird die Annahme getroffen, das System befände sich im linearen Bereich, sprich die Auswirkungen der Sperrdioden seien zu vernachlässigen. Außerdem wird die auftretende trockene Reibung als Teil der Eingangsgröße aufgefasst (siehe Abschnitt 7.5.1). Das so entstehende Modell hat die folgende Form:

$$\begin{aligned} \begin{bmatrix} \frac{di}{dt} \\ \frac{d\omega}{dt} \end{bmatrix} &= \begin{bmatrix} -\frac{R}{L} & -\frac{k_a}{L} \\ \frac{k_a}{J} & -\frac{k_1}{J} \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & \frac{1}{J} \end{bmatrix} \cdot \begin{bmatrix} u \\ T_{l,lin} \end{bmatrix} \\ \vec{y} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega \end{bmatrix} \end{aligned} \quad (9.1)$$

Transformiert man das System in den Bildbereich und betrachtet das Verhalten vom Eingang  $u$  zum Ausgang  $i$ , so erhält man die Stromübertragungsfunktion.

$$P_i(s) = \frac{1}{L} \cdot \frac{s + \frac{k_1}{J}}{s^2 + \left(\frac{k_1}{J} + \frac{R}{L}\right) \cdot s + \frac{Rk_1 + k_a^2}{JL}} \quad (9.2)$$

Mit den im Abschnitt 7 ermittelten Parametern ergibt sich daraus folgendes Bode-Diagramm.



**Abbildung 9.1:** Frequenzgang der Stromübertragungsfunktion.

Im unteren Frequenzbereich erkennt man in Abbildung 9.1 deutlich den Einfluss der induzierten Gegenspannung auf den Strom. Diese Spannung wirkt entgegen dem Eingangssignal und führt zu einer negativen Betragskennlinie. Erst bei höheren Frequenzen, wenn das mechanische System zu langsam ist, um der Stromänderung zu folgen, kommt es zu einer positiven Verstärkung. Da

dieser Kennlinienverlauf für einen Entwurf ungünstig ist, behilft man sich mit einer sogenannten *EMK-Kompensation*.

### 9.1.1 EMK-Kompensation

Zur Eliminierung der Rückwirkung der Gegenspannung auf den Verlauf des Motorstromes kann die Spannung auf das Eingangssignal  $u$  aufgeschaltet werden. Dieses Verfahren bezeichnet man als EMK-Kompensation oder EMK-Aufschaltung<sup>1</sup>. Die Stromdifferentialgleichung und die zugehörige Übertragungsfunktion vereinfachen sich dadurch zu [Schröder (2009b)]:

$$\frac{di}{dt} = -\frac{R}{L} \cdot i + \frac{1}{L} \cdot u \quad \bullet \quad \left(s + \frac{R}{L}\right) \cdot I(s) = \frac{1}{L} \cdot U(s) \quad (9.3)$$

Das Verhalten des Stromkreises lässt sich nun über ein PT1-Glied beschreiben.

$$P_{i,EMK}(s) = \frac{\frac{1}{R}}{1 + s \cdot \frac{L}{R}} = \frac{V_i}{1 + sT_i} = \frac{7.475}{1 + s0.0003027} = \frac{7.475}{1 + \frac{s}{3303}} \quad (9.4)$$

### 9.1.2 Entwurf

Üblicherweise stellt man die Parameter des Stromregelkreises nach dem Betragsoptimum ein. Der grundlegende Gedanke hinter diesem Verfahren ist, den Regler so zu konfigurieren, dass der geschlossene Regelkreis der Referenzgröße auch bei hohen Frequenzen noch gut folgen kann. Anders formuliert bedeutet diese Forderung, den Betrag des Amplitudenganges des geschlossenen Kreises für große  $\omega$  bei Eins und den Phasengang bei Null zu halten [Schröder (2009b)].

Voraussetzung für ein Funktionieren dieses Ansatzes ist eine ausreichend schnelle Abtastung. Die mit der vorliegenden Reglerplatine erreichbare Abtastzeit ist jedoch nach unten hin auf eine Millisekunde begrenzt, während die Zeitkonstante des Stromkreises etwa 300 Mikrosekunden beträgt. Außerdem weist der Stromverlauf durch die PWM-Ansteuerung ohnehin schon relativ starke Fluktuationen auf, sodass eine allzu hohe Bandbreite des Regelkreises problematisch wäre. Aus diesen Gründen wurde auf den Einsatz dieses Verfahrens verzichtet. Stattdessen wurde eine Durchtrittsfrequenz von  $\omega_c = 150 \frac{rad}{s}$ , die einen Kompromiss zwischen Anregelzeit und Empfindlichkeit gegenüber Fluktuationen darstellt, gewählt. Die zeitdiskrete Streckenübertragungsfunktion lautet:

$$P_i(z) = \frac{7.2}{z - 0.03672} \quad (9.5)$$

Um Aussagen über den Frequenzgang einer zeitkontinuierlichen Übertragungsfunktion  $G(s)$  machen zu können, wählt man  $s = j\omega$ . Bei einer zeitdiskreten Übertragungsfunktion ist jedoch zu bedenken, dass  $z = e^{sT_s}$  gilt ( $T_s$  ist die Abtastzeit). Um auch in diesem Fall rationale Übertragungsfunktionen der (transformierten) Kreisfrequenz zu erhalten, kann die folgende Transformation durchgeführt werden [Schröder (2009b)]:

$$z = \frac{1 + w \cdot \frac{T_s}{2}}{1 - w \cdot \frac{T_s}{2}} \quad \text{mit } w = \zeta + j\Omega \quad (9.6)$$

<sup>1</sup>Bei sehr großem Zeitkonstantenverhältnis (mechanisch zu elektrisch) ist diese Maßnahme eher theoretischer Natur und muss nicht umgesetzt werden. Es kann dann davon ausgegangen werden, dass der Stromregelvorgang bereits beendet ist, bevor sich die Drehzahl in spürbarem Maße ändert.



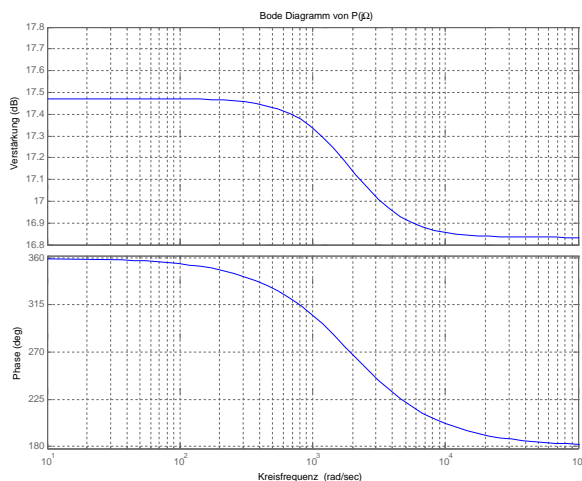
Zwischen der tatsächlichen Kreisfrequenz  $\omega$  und der transformierten Kreisfrequenz  $\Omega$  besteht der folgende Zusammenhang:

$$\Omega = \frac{2}{T_s} \tan\left(\omega \cdot \frac{T_s}{2}\right) \Leftrightarrow \omega = \frac{2}{T_s} \arctan\left(\Omega \cdot \frac{T_s}{2}\right) \quad (9.7)$$

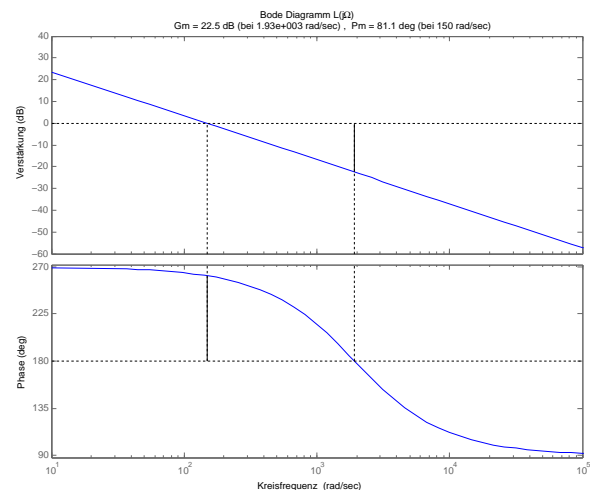
Für kleine Argumente gilt  $\arctan(x) \approx x$ , d.h. reale und transformierte Kreisfrequenz stimmen näherungsweise überein. Je höher die Abtastfrequenz, desto größer ist dieser Bereich. Nach Durchführung dieser Transformation auf Gleichung 9.5 erhält man:

$$P_i(w) = \frac{-6.945w + 13890}{w + 1858} \quad (9.8)$$

Abbildung 9.2 zeigt das Bode-Diagramm der transformierten Streckenübertragungsfunktion. Man erkennt, dass durch Hinzufügen eines einfachen Integrators der Amplitudengang die gewünschte Form erhalten wird. Der Verstärkungsfaktor ist nun nur noch so einzustellen, dass die Durchtrittsfrequenz  $\Omega_c$  bei  $150 \frac{\text{rad}}{\text{s}}$  liegt (siehe Abbildung 9.3).



**Abbildung 9.2:** Frequenzgang der transformierten Strecke.



**Abbildung 9.3:** Bode-Diagramm des offenen Kreises mit Integrierer als Regler.

Die so erhaltene Reglerübertragungsfunktion (Gleichung 9.9) muss anschließend wieder in eine zeitdiskrete Darstellung transformiert werden (Gleichung 9.10) <sup>2</sup>.

$$R_i(w) = \frac{20.076}{w} \quad (9.9)$$

$$R_i(z) = 0.01004 \cdot \frac{z + 1}{z - 1} \quad (9.10)$$

## 9.2 Verbesserung der Stromregelung

Da der Rollenprüfstand naturgemäß meistens mit fahrendem Fahrzeug betrieben wird, muss das Verhalten der Regelung für diesen Fall nochmals untersucht werden, denn schließlich wurde das nichtlineare Verhalten bisher völlig außer Acht gelassen. Das Lastmoment  $T_l$  ist nun ungleich Null, sprich die sich einstellende Drehzahl wird nicht mehr nur vom Ankerstrom der elektrischen Maschine vorgegeben. Das Einsatzgebiet der Stromregelung bei  $T_l \neq 0$  umfasst im Wesentlichen vier verschiedene Betriebspunkte:

<sup>2</sup>Dies entspricht der zeitdiskreten Approximation eines Integrators mit der Trapezregel.

1. **Antreiben mit  $|\bar{u}| > k_a|\omega|$ :** Dies ist der „Standardbetriebspunkt“ bei dem das lineare Motormodell gültig ist und sich die elektrische Maschine im motorischen Quadranten befindet.
2. **Antreiben mit  $|\bar{u}| < k_a|\omega|$ :** Durch den Einfluss der Sperrdioden entsteht auch in diesem Fall ein Strom mit demselben Vorzeichen wie die Winkelgeschwindigkeit. Der Stromfluss ist jedoch lückend, was dank des Tiefpassverhaltens der Motorstrommessvorrichtung auf den Regler wenig Einfluss hat. Die Regelung mittels zeitdiskretem I-Regler funktioniert auch in diesem Bereich wie gewohnt.
3. **Bremsen mit  $|i_{ref}| > |i_0|$ :** Der vorgegebene Bremsstrom ist also größer als der mittels generatorischem Betrieb erzeugbare Kurzschlussstrom  $i_0$ . Der Motor ist über eine der beiden Freilaufdioden und einen der beiden unteren Transistoren der modifizierten H-Brücke permanent kurzgeschlossen, durch die PWM-Eingangsspannung wird der auftretende Stromfluss noch vergrößert. Die Sperrdioden haben hier keinen Einfluss, da Strom und Eingangsspannung dasselbe Vorzeichen haben.
4. **Bremsen mit  $|i_{ref}| < |i_0|$ :** Hier wäre es prinzipiell möglich rein generatorisch zu bremsen. Tut man dies nicht, so schaltet der Regler zur Einstellung des gewünschten Stromes die beiden Diagonalzweige der mod. H-Brücke ständig um (bzw. wechselt das Vorzeichen der Eingangsspannung). Wie im Punkt 3 beschrieben, ist die Maschine in einer der beiden möglichen Polungen aber kurzgeschlossen. Dies bedeutet der Stromstärke ändert sich in eine Richtung viel schneller als in die andere. Daher führt der motorische Betrieb in diesem Fall zu einem äußerst unruhigen Stromverlauf und ist daher zu vermeiden.

Die Stromregelung kann daher verbessert werden, wenn jene Ströme, die sich generatorisch erzeugen lassen, auch auf diese Weise eingestellt werden. Um zwischen generatorischem und motorischem Betrieb umschalten zu können, muss für eine bestimmte Winkelgeschwindigkeit  $\omega_0$  der maximal durch einen Kurzschluss der Maschine erreichbare Bremsstrom  $i_0$  bekannt sein. Die Ankerkreisgleichung vereinfacht sich dann zu:

$$\frac{di}{dt} = -\frac{R}{L} \cdot i - \frac{k_a}{L} \cdot \omega_0 \quad (9.11)$$

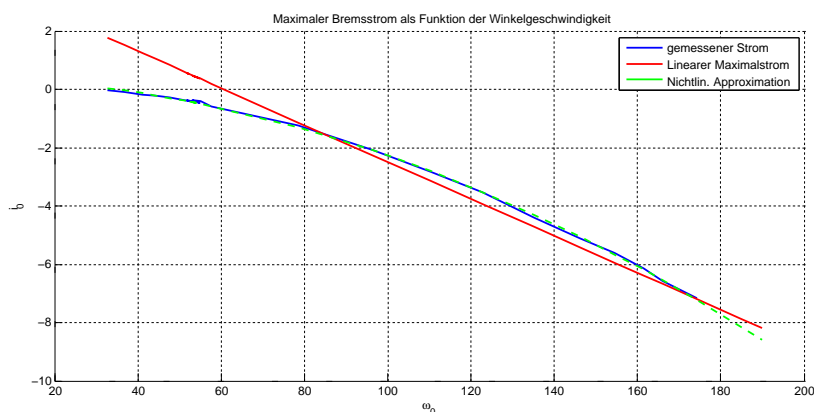
Bei stationärer Drehzahl erhält man für den maximalen Bremsstrom daher:

$$i_0 = -\frac{k_a}{R} \cdot \omega_0 \quad (9.12)$$

Wie in Abbildung 5.7 zu sehen ist, enthält der Stromkreis für den generatorischen Betrieb zusätzliche Dioden. Der daher auftretende Diodenspannungsabfall von etwa  $u_D = 0.4V$  verringert den nutzbaren Strom.

$$i_{0,lin} = -\frac{k_a\omega_0 \pm u_D}{R} \quad (9.13)$$

Weiters kommt es zu einem nichtlinearen Zusammenhang zwischen Winkelgeschwindigkeit und Strom, wenn die induzierte Motorspannung  $u_{ind} = k_a\omega$  kleiner ist als die Diodenspannung. Daher wurde für niedrige Winkelgeschwindigkeiten der Kurzschlussstrom experimentell ermittelt (siehe Abbildung 9.4).



**Abbildung 9.4:** Kurzschlussstrom als Funktion der Winkelgeschwindigkeit: Die aufgezeichneten Daten sind blau dargestellt, während die rote Linie den maximalen Stromverlauf bei linearen Verhältnissen (Gleichung 9.13) zeigt. Da erst ab einer Geschwindigkeit von  $|\omega_0| \approx 60$  die induzierte Spannung größer ist als die Diodenspannung, liefert diese Schätzung erst für größere Geschwindigkeiten brauchbare Ergebnisse. Für  $|\omega| < 60$  kann der Zusammenhang über ein Polynom zweiter Ordnung näherungsweise beschrieben werden (grüner Verlauf). Im Geschwindigkeitsbereich  $|\omega_0| < 33$  muss motorisch gebremst werden.

Die ermittelte Funktion zur Berechnung des nichtlinearen Zusammenhangs zwischen Winkelgeschwindigkeit und Strom lautet:

$$i_{0,nlin}(\omega) = -0.000229 \text{sign}(\omega)\omega^2 - 0.003769\omega + 0.40551 \text{sign}(\omega) \quad (9.14)$$

Auf welche Art und Weise ein gewünschter Bremsstrom erzeugt werden soll, wird anhand der folgenden Kriterien entschieden:

- $|\omega| < 33$ : MOTORISCH
- $33 \leq |\omega| < 185$ : Wenn  $i_{ref} \leq i_{0,nlin}$  gilt, dann soll GENERATORISCH gebremst werden, ansonsten MOTORISCH.
- $|\omega| \geq 185$ : Wenn  $i_{ref} \leq i_{0,lin}$  erfüllt ist, dann soll GENERATORISCH gebremst werden, ansonsten MOTORISCH.

Der im generatorischen Betrieb zum Einatz kommende Regler ist derselbe wie im motorischen Betrieb. Beim Umschalten muss jedoch der Integrator zurückgesetzt werden, ansonsten hätte vor allem das Umschalten vom Generator zum Motor eine unzulässig hohe Stromspitze zur Folge. Um einen kurzzeitigen Einbruch des Stromes auf Null zu verhindern, wird der Integrator zum Umschaltzeitpunkt auf einen Schätzwert der notwendigen Stellgröße anstatt auf Null gesetzt. Für den generatorischen Betrieb wird dieser Schätzwert aus einem durch Versuche ermittelten Zusammenhang zwischen Tastverhältnis und Bremswiderstand berechnet.

### 9.3 2-SMC Algorithmen

Da die Second Order Sliding Mode Regler die Sliding Surface  $s$  und deren erste Ableitung  $\dot{s}$  zu Null machen, bietet es sich an, den Regelfehler als Variable  $s$  zu definieren:

$$s = i - i_{ref} \quad (9.15)$$

Die erste und die zweite Ableitung der Variable  $s$  lauten (unter Vernachlässigung der Gegenspannungsinduktion):

$$\dot{s} = -\frac{R}{L}i + \frac{1}{L}u - \frac{di_{ref}}{dt} \quad (9.16a)$$

$$\ddot{s} = \underbrace{\frac{R^2}{L^2}i - \frac{R}{L^2}u - \frac{d^2i_{ref}}{dt^2}}_{\varphi} + \underbrace{\frac{1}{L}}_{\gamma} \dot{u} \quad (9.16b)$$

Man erkennt also, dass es sich im vorliegenden Fall um den Anti Chattering Case handelt (relativer Grad ist Eins), bei dem der Regler die erste zeitliche Ableitung der Stellgröße manipuliert.

### 9.3.1 Sub-Optimal Algorithm

Beim Einsatz des Sub-Optimal Reglers für System 9.16 muss also noch ein Integrator vor die Stromregelstrecke geschaltet werden. Die zeitdiskrete Umsetzung des Sub-Optimal Stromreglers lautet:

$$u_k = u_{k-1} - T\alpha_k U \operatorname{sign}\left(s_k - \frac{1}{2}s_{M,k}\right)$$

$$\alpha_k = \begin{cases} 1 & \text{wenn } (s_k - \frac{1}{2}s_{M,k})s_{M,k} \geq 0 \\ \alpha^* & \text{sonst} \end{cases} \quad (9.17)$$

$$U > 0 \text{ und } \alpha^* > 1$$

Die Konstante  $T$  in obiger Gleichung ist die Abtastzeit und die Werte  $s_{M,k}$  sind die Schnittpunkte der Trajektorie mit der Abszisse in der  $s - \dot{s}$ -Ebene, die mit Hilfe des im Abschnitt 8.4.2.3 beschriebenen *Peak Detectors* erfasst werden. Trotz der starken Fluktuationen des Stromverlaufs hat sich die Wahl  $N = 1$  für den Peak Detector bewährt. Analog zum linearen I-Regler wird auch in diesem Fall zwischen dem generatorischen und dem motorischen Betrieb anhand des experimentell ermittelten maximalen Bremsstroms bei gegebener Winkelgeschwindigkeit umgeschaltet (siehe Kapitel 9.2). Um den Einfluss einer niedrigeren Abtastzeit zu zeigen, werden im Ergebniskapitel (11.1.2) auch Simulationsergebnisse dargestellt. Daher gibt es zwei Parametersätze, einen für die Simulation und einen für den realen Regler.

Parameter	Simulation mit $T = 100\mu s$	Prüfstand mit $T = 1ms$
U	110	15
$\alpha^*$	1.82	1.33

**Tabelle 9.1:** Parameterwahl des Sub-Optimal Sliding Mode Reglers für Simulation und realen Prüfstand mit unterschiedlichen Abtastzeiten.

Die Einstellung der Parameter erfolgte zuerst in der Simulation mittels „Trial & Error“ und wurde anschließend am realen Prüfstand angepasst.

### 9.3.2 Super Twisting Algorithm

Analog zum Sub-Optimal Algorithm bildet der Stromregelfehler lt. 9.15 auch hier die Sliding Surface  $s$ . Die zeitdiskrete Realisierung des Super Twisting Reglers lautet:

$$\begin{aligned}
u_k &= -\lambda(s_k)\sqrt{|s_k|}\text{sign}(s_k) + u_{1,k} \\
u_{1,k} &= u_{1,k-1} - T \cdot W \text{sign}(s_k) \\
W &> 0 \text{ und } \lambda(s_k) > 0
\end{aligned} \tag{9.18}$$

Eine Umschaltung zwischen generatorischem und motorischem Betriebsmodus erfolgt nach denselben Kriterien wie bei den zuvor beschriebenen Reglern. Wegen der hohen Abtastperiode  $T = 0.001$  und der starken Fluktuationen war es bei der Implementierung am realen Prüfstand notwendig, den Parameter  $\lambda$  in Abhängigkeit vom Regelfehler (also  $s$ ) zu wählen. Bei betragsmäßig großem  $s$  führt ein hoher Wert von  $\lambda$  zu schnellerer Konvergenz, in der Nähe des Referenzwertes führt ein kleineres  $\lambda$  hingegen zu einem ruhigeren Stromverlauf:

$$\lambda(s_k) = \begin{cases} 0.6 & \text{wenn } |s_k| > 0.7 \\ 0.24 & \text{sonst} \end{cases} \tag{9.19}$$

Für die Simulation bei höheren Abtastzeiten kam ein konstanter Wert für  $\lambda$  zum Einsatz. Die Parameter des Algorithmus sind der folgenden Tabelle zu entnehmen:

Parameter	Simulation mit $T = 100\mu s$	Prüfstand mit $T = 1ms$
W	60	25
$\lambda$	0.45	siehe Gl 9.19

**Tabelle 9.2:** *Parameterwahl des Super Twisting Sliding Mode Reglers für Simulation und realen Prüfstand mit unterschiedlichen Abtastzeiten.*

Auch hier wurden die Parameter anfangs in der Simulation grob eingestellt, am Prüfstand erfolgte die Feinabstimmung.

## 9.4 Momentenregelung

Die Hauptaufgabe des Rollenprüfstandes ist es, bei fahrendem Auto ein vorgegebenes Moment auf die Rollen aufzuschalten, um verschiedene Belastungssituationen nachzustellen. Wie in Kapitel 5.1.2 erläutert, ist das Drehmoment einer Gleichstrommaschine proportional zum fließenden Ankerstrom. Daher ist es naheliegend, die Momentenregelung mit Hilfe der im vorangegangenen Abschnitt entwickelten Stromregelung zu realisieren. Das an den Rollen anliegende Moment setzt sich aus dem strombezogenen Anteil und den Reibungsverlusten gewichtet mit dem Übersetzungsverhältnis zusammen:

$$T_{roller} = g_{m2r}(k_a i - [k_1 \omega + k_0 \text{sign}(\omega)]) \tag{9.20}$$

Mit dem geforderte Moment  $T_{ref}$  kann nun ein Regelfehler  $e_T = T_{ref} - T_{roller}$  definiert werden. Daraus ergibt sich der notwendige Ankerstrom  $i_{ref}$ :

$$i_{ref} = \frac{e_T}{g_{m2r}k_a} + i \tag{9.21}$$



# 10 Implementierung Geschwindigkeitsregelung

## 10.1 Kaskadenstruktur mit PI-Regler

Zur Regelung der Geschwindigkeit wurde dem im Abschnitt 9.1 beschriebenen Stromregler ein zusätzlicher PI-Regelkreis überlagert. Die Reglerparametrierung erfolgte mit Hilfe des *Symmetrischen Optimums* (Kapitel 8.3). Zusätzlich beinhaltet der Regler Anti-Windup Vorkehrungen, um die Beschränkung des Motorstromes ordnungsgemäß umzusetzen.

### 10.1.1 Entwurf

Die Übertragungsfunktion des mechanischen Streckenteiles ist ein Verzögerungsglied erster Ordnung mit dem Motorstrom als Stellgröße:

$$P_\omega(s) = \frac{\Omega(s)}{I(s)} = \frac{\frac{k_a}{k_1}}{1 + s \cdot \frac{J}{k_1}} = \frac{240.7}{1 + s2.45} \quad (10.1)$$

Die Zeitkonstante dieses Streckenteiles beträgt  $\tau_\omega = 2.45$ , sodass ein quasikontinuierlicher Entwurf vorgenommen werden kann. Die Führungsübertragungsfunktion des Stromregelkreises setzt sich aus dem im vorangegangenen Kapitel entworfenen Stromregler  $R_i(z)$  und der Stromregelstrecke  $P_{i,EMK}(z)$  zusammen:

$$\tilde{P}_i(z) = \frac{R_i(z)P_{i,EMK}(z)}{1 + R_i(z)P_{i,EMK}(z)} = \frac{0.0723 \cdot (z + 1)}{(z - 0.834) \cdot (z - 0.131)} \quad (10.2)$$

Für einen quasikontinuierlichen Entwurf des Geschwindigkeitsreglers muss diese zeitdiskrete Übertragungsfunktion in eine zeitkontinuierliche transformiert werden. Die so entstehende Funktion wird fortan als  $\tilde{P}_i(s)$  bezeichnet und weist die Ordnung zwei auf.

Zur Vereinfachung des weiteren Entwurfsvorgangs ist es zweckmäßig,  $\tilde{P}_i(s)$  durch ein Verzögerungsglied erster Ordnung  $\hat{P}_i(s)$  zu approximieren (siehe Abbildung 10.1).

$$\tilde{P}_i(s) = \frac{-65.89 \cdot (s - 5615)}{(s + 2034) \cdot (s + 181.9)} \quad (10.3a)$$

$$\hat{P}_i(s) = \frac{1}{1 + s0.0062} \quad (10.3b)$$

Die Parameter dieser Ersatzübertragungsfunktion wurden durch numerische Optimierung des quadratischen Fehlers der Sprungantwort ermittelt. Durch serielle Zusammenschaltung der beiden PT1-Glieder  $\hat{P}_i(s)$  und  $P_\omega(s)$  entsteht die Gesamtstreckenübertragungsfunktion  $P(s)$ . Deren Struktur entspricht der in Abschnitt

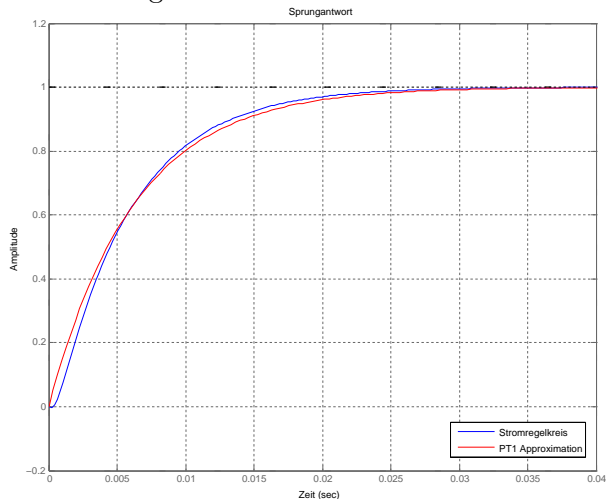


Abbildung 10.1: Näherung des Stromregelkreises durch ein PT1-Glied.

8.3.2 vorgestellten Anneinanderreihung von PT1-Gliedern mit ungünstigem Zeitkonstantenverhältnis für einen Entwurf nach dem Betragsoptimum.

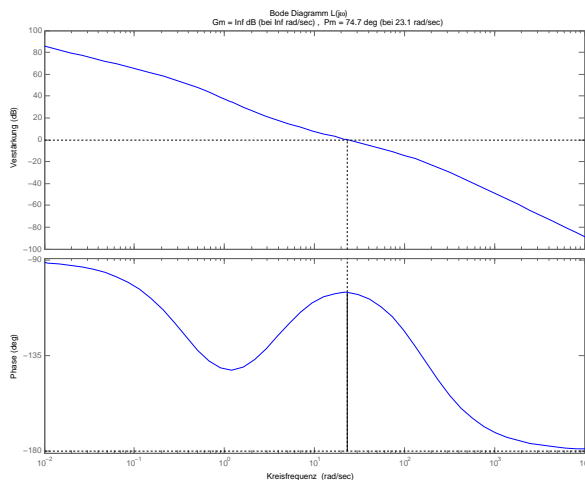
$$P(s) = P_\omega(s) \cdot \hat{P}_i(s) = \frac{V_P}{1 + sT_1} \cdot \frac{1}{1 + sT_\sigma} = \frac{240.7}{1 + s2.45} \cdot \frac{1}{1 + s0.0062} \quad (10.4)$$

Die über die Einstellregeln nach dem Symmetrischen Optimum unter Verwendung des Parameters<sup>1</sup>  $a = 7$  erhaltene PI-Reglerübertragungsfunktion lautet:

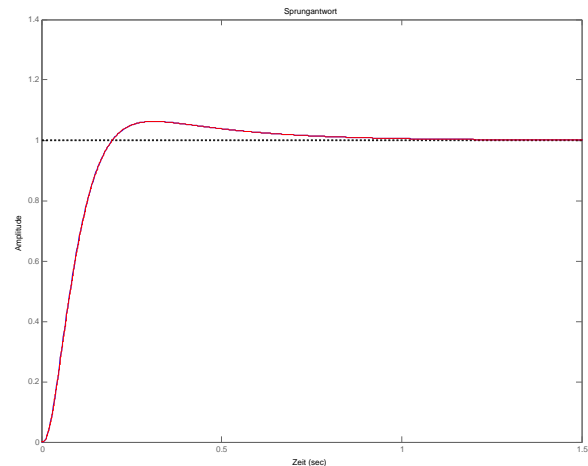
$$R_\omega(s) = \frac{0.234 \cdot (s + 3.32)}{s} \quad (10.5)$$

Der Frequenzgang weist die bei der Symmetrischen Optimierung typische Phasenhebung bei der Durchtrittsfrequenz auf (Abbildung 10.2). Zur Dämpfung des Überschwingens bei einem Führungssprung wird die Referenzgröße gefiltert, wobei die Filterzeitkonstante etwas kleiner gewählt wurde als die der Nullstelle der Führungsübertragungsfunktion ( $a_{filter} = 3$ ). Dadurch konnte die Anregelzeit etwas verringert werden.

$$F(s) = \frac{1}{1 + s0.0553} \quad (10.6)$$



**Abbildung 10.2:** Bode-Diagramm des offenen Kreises nach Optimierung: Die Grenzfrequenz beträgt  $\omega_c = 23.1 \frac{\text{rad}}{\text{s}}$  bei einer Phasenreserve von etwa  $\Phi_r = 74.7^\circ$ .



**Abbildung 10.3:** Sprungantwort des geschlossenen Kreises inklusive Vorfilter  $F(s)$ : Der zeitkontinuierliche und der zeitdiskrete Verlauf sind nahezu identisch.

Die zeitdiskreten Versionen des Reglers und des Filters lauten:

$$R_\omega(z) = \frac{0.2344z - 0.2336}{z - 1} \quad (10.7)$$

$$F(z) = \frac{0.01791}{z - 0.9821} \quad (10.8)$$

<sup>1</sup>Die Wahl eines so großen Parameters führt zu einem langsameren Regler. Zwar war das Führungsverhalten bereits für  $a \geq 5$  sehr gut, doch beim praktischen Einsatz zeigten sich Probleme, sobald Störungen, verursacht durch das Aktivieren des Fahrzeugmotors, auftraten.



## 10.2 2-SMC Algorithmen

Zur Einhaltung der Strombegrenzung ist es auch im Falle der Second Order Sliding Mode Geschwindigkeitsregelung zweckmäßig, auf eine Kaskadenstruktur mit unterlagertem Stromregelkreis zurückzugreifen. Da die 2-SMC Stromregler aufgrund der hohen Abtastperiode eine unbefriedigende Leistung an den Tag legen, wird bei den nun folgenden Regelungen in der inneren Schleife ebenfalls der lineare Stromregler aus Abschnitt 9.1 verwendet. Die Approximation des geschlossenen Stromregelkreises (siehe 10.3b) im Frequenz- und Zeitbereich lautet:

$$\hat{P}_i(s) = \frac{I(s)}{I_{ref}(s)} = \frac{1}{1 + s0.0062} \bullet \text{---} \circ \frac{di}{dt} = - \underbrace{\frac{1}{0.0062}}_{=:a_i} i(t) + \underbrace{\frac{1}{0.0062}}_{=:b_i} i_{ref}(t) \quad (10.9)$$

Analog zur Stromregelung ist auch hier die Wahl des Regelfehlers  $e(t)$  als Sliding Surface sinnvoll:

$$s = \omega - \omega_{ref} \quad (10.10)$$

Die zeitlichen Ableitungen erster und zweiter Ordnung der Sliding Surface lauten:

$$\dot{s} = \frac{k_a}{J} i - \frac{k_1}{J} \omega - \frac{1}{J} T_{l,lin} - \dot{\omega}_{ref} \quad (10.11a)$$

$$\ddot{s} = \underbrace{\frac{k_1^2}{J^2} \omega - \left( \frac{k_1 k_a}{J^2} + \frac{k_a a_i}{J} \right) i - \left( \frac{1}{J} \dot{T}_{l,lin} + \ddot{\omega}_{ref} \right)}_{\varphi} + \underbrace{\frac{k_a b_i}{J}}_{\gamma} i_{ref} \quad (10.11b)$$

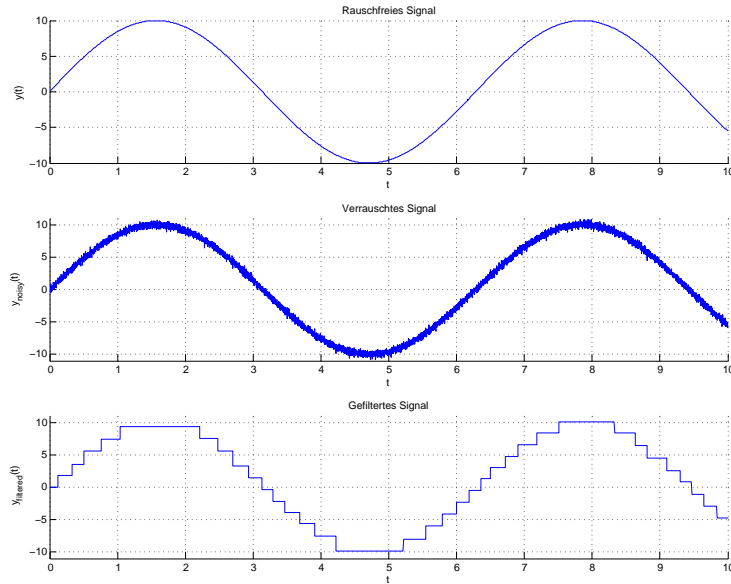
Daran ist erkenntlich, dass das System einen relativen Grad von zwei besitzt.

### 10.2.1 Sub-Optimal 2-SMC

Abgesehen von der unterschiedlichen Definition der Sliding Surface entspricht der Sub-Optimal Geschwindigkeitsregler hinsichtlich seiner Struktur exakt des Sub-Optimal Stromreglers (Gleichung 9.17). Die Parameter lauten  $U = 100$ ,  $\alpha^* = 1.4$  und  $N = 1$ . Grundlage für ein Funktionieren dieses Algorithmus ist die korrekte Detektion der Extremwerte  $s_M$  der Sliding Variable  $s$ . Die Winkelgeschwindigkeit der Motorwelle weist jedoch selbst im Betrieb mit konstantem Pulsweitenverhältnis beträchtliche Schwankungen auf und ist zusätzlich noch von Rauschen überlagert. Daher enthält das Geschwindigkeitsmesssignal eine Vielzahl an lokalen Extrema, die vom Peak Detector aus Kapitel 8.4.2.3 erfasst werden. Laut Pisano (2000) und Pisano u. a. (2008) soll eine Erhöhung des Parameters  $N$  in solchen Fällen Abhilfe schaffen. Diese Behauptung hat sich in den Versuchen am Prüfstand jedoch nicht bestätigt: Um die Anzahl der „falsch“ detektierten Werte  $s_M$  zu reduzieren, musste  $N$  so weit erhöht werden, dass die dadurch entstehende verzögerte Detektion ebenfalls zu Problemen in Form von Überschwingen und unruhiger Drehzahl führte. Damit der Algorithmus dennoch eingesetzt werden kann, ist eine Vorfilterung der Winkelgeschwindigkeit unerlässlich. Erst das gefilterte Signal wird anschließend vom Peak Detector weiterverarbeitet.

#### 10.2.1.1 Vorfilterung von $\omega$

Das Ausgangssignal des Filters ist eine stückweise konstante Approximation des Originalsignals (siehe Abbildung 10.4). Auf diese Weise gehen die unerwünschten „falschen“ Peaks der Messgröße nicht in die Detektion ein.



**Abbildung 10.4:** Beispiel zur Demonstration des Vorfilters: Das rauschfreie Signal weist drei Extremwerte auf, die vom Peak Detektor erfasst werden sollen. Führt man das verrauschte Signal direkt dem Peak Detektor zu, so würde eine Vielzahl von lokalen Extrema erkannt werden. Beim gefilterten Signal ist dies jedoch nicht mehr der Fall.

Die Berechnung des Ausgangssignals erfolgt nach folgender Vorschrift, wobei sich für die Konstante ein Wert von  $\sigma = 1$  bewährt hat:

$$y_{filtered,k} = \begin{cases} y_{input,k} & \text{wenn } |y_{input,k} - y_{filtered,k-1}| > \sigma \\ y_{filtered,k-1} & \text{sonst} \end{cases} \quad (10.12)$$

### 10.2.2 Super Twisting 2-SMC

Der Super Twisting Algorithmus wurde eigentlich für Systeme mit einem relativen Grad von eins entwickelt, für das vorliegende System gilt hingegen  $r = 2$ . Andererseits ist die Zeitkonstante des geschlossenen Stromregelkreises im Vergleich zum mechanischen Teil (siehe Kapitel 10.1.1) relativ klein und im Gegensatz zum Sub-Optimal Algorithm ist keine Peak Detection erforderlich, welche sich ja als besonders kritisch herausgestellt hat. Die Berechnungsvorschrift der Stellgröße ist der Gleichung 9.18 zu entnehmen, der Parameter des Integrators wurde zu  $W = 55$  gewählt, während  $\lambda$  ähnlich wie bei der Stromregelung in Abhängigkeit von der momentanen Regelabweichung bestimmt wird.

$$\lambda(s_k) = \begin{cases} 1.5 & \text{wenn } |s_k| > 10 \\ 0.6 & \text{sonst} \end{cases} \quad (10.13)$$

# 11 Ergebnisse

Die im Folgenden vorgestellten Ergebnisse wurden durch Simulationen sowie durch Messungen am Prüfstand gesammelt. Die Datenübertragung fand bei letzteren über die UART-Schnittstelle statt, um eine feinere zeitliche Auflösung zu erhalten. Die Qualität der entworfenen Regelalgorithmen wird anhand des Verhaltens beim Aufschalten von Sprungfunktionen als Referenz- und als Störgrößen beurteilt. Das Zusammenspiel der Momentenregelung des Prüfstandes und der Geschwindigkeitsregelung des Fahrzeuges kann außerdem noch durch das Abfahren eines längeren Fahrzyklus überprüft werden.

## 11.1 Strom- / Momentenregelung

### 11.1.1 I-Regler

Da, wie in Kapitel 9.4 erläutert, die Momentenregelung auf die Stromregelung aufbaut, werden hier nur die Testergebnisse der Stromregelung präsentiert. Als Referenzgrößen kommen dabei Sprungfunktionen auf ein Ampere und auf 6.5 Ampere zum Einsatz. Das erzeugte Moment bei letzterer reicht aus, um die Rollen mit daraufstehendem Fahrzeug in Bewegung zu setzen. Zuletzt wird der Einfluss von Widerstandsmomentenänderungen durch das Fahrzeug untersucht.

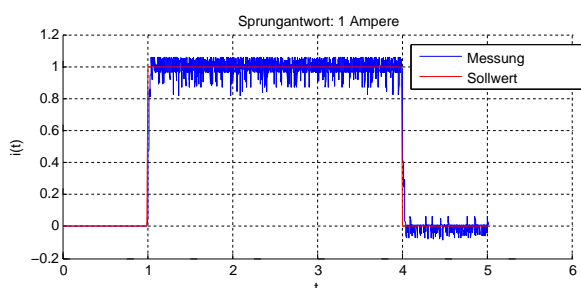


Abbildung 11.1: Sprungantwort des Stromregelkreises bei ein Ampere Sollwert.

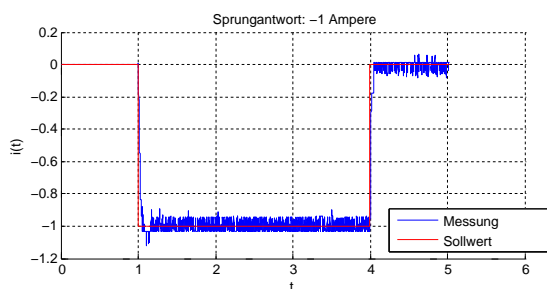


Abbildung 11.2: Sprungantwort des Stromregelkreises bei minus ein Ampere Sollwert.

Da sich bei einem Ampere der Motor noch nicht zu drehen beginnt, sind die hier auftretenden Störungen wesentlich geringer (keine Kommutierung, keine Gegenspannungsinduktion, keine Fluktuationen des Widerstandsmomentes, ...).

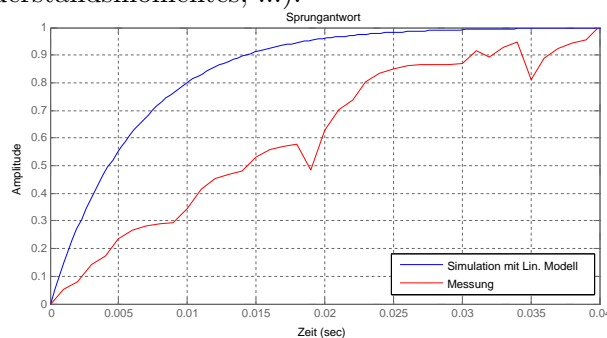
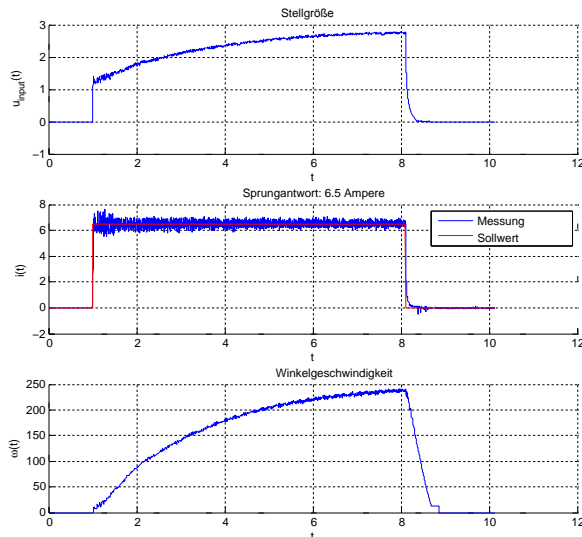
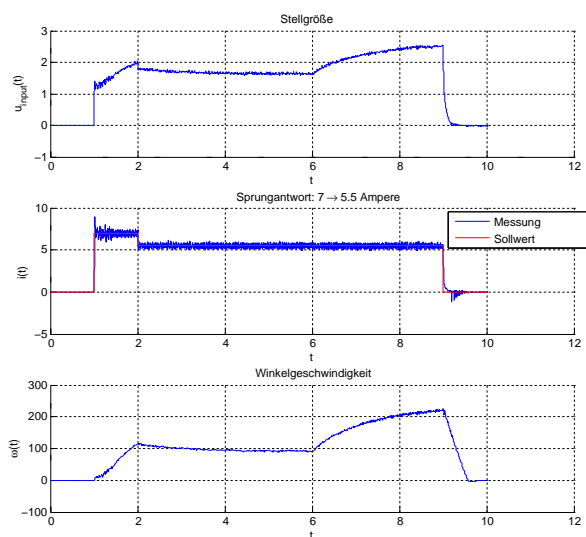


Abbildung 11.3: Vergleich der Sprungantworten von Simulation und Messung: Die Anregelzeit stimmt relativ gut mit der Simulation überein.



**Abbildung 11.4:** Sprungantwort des Stromregelkreises auf 6.5 Ampere Sollwert: Das entstehende Moment reicht aus, um den Motor in Drehung zu versetzen. Die hier auftretenden Störungen sind erheblich größer als im Falle des stillstehenden Betriebs.



**Abbildung 11.5:** Stromregelkreis unter Störeinfluss: Zum Zeitpunkt  $t = 6$  wird die Drehzahl durch Aktivieren des Fahrzeugmotors erhöht. Dies hat jedoch keinerlei Einfluss auf die Stromregelung.

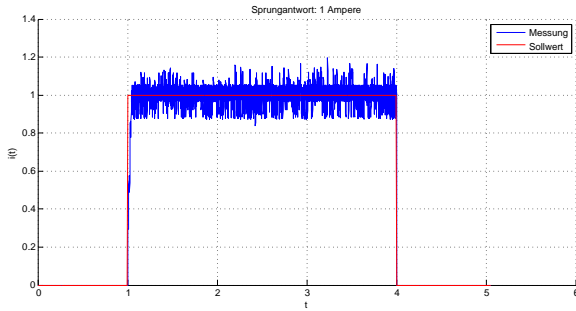
## 11.1.2 Second Order Sliding Mode Regler

Wie alle Sliding Mode Verfahren profitieren auch die Higher Order Sliding Mode Regler von möglichst kleinen Abtastzeiten, da in diesem Fall die Nulldurchgänge der Sliding Surface besser erkannt werden können. Die am Prüfstand implementierten 2-SMC Stromregler können aufgrund der langen Abtastzeit von einer Millisekunde nicht mit dem im vorigen Abschnitt vorgestellten I-Stromregler konkurrieren. Daher werden in diesem Abschnitt zusätzlich zu den realen Messergebnissen auch Simulationsergebnisse bei geringerer Abtastzeit ( $T = 100\mu s$ ) präsentiert. Zur Simulation wird das nichtlineare Streckenmodell gemäß 5.15a und 5.15b mit PWM-Stellgröße benutzt, um die Fluktuationen des Stroms im stationären Betrieb in den Vergleich miteinbeziehen zu können. Um einen fairen Vergleich mit den Standardreglern anstellen zu können, wird auch deren Bandbreite erhöht. Dabei ist jedoch Vorsicht geboten, da damit auch die Tendenz zum Überschwingen merkbar ansteigt. Als Kompromiss aus Schnelligkeit und Dämpfung wurde die Grenzfrequenz auf  $w_c = 500$  vergrößert, die Reglerübertragungsfunktion lautet nun:

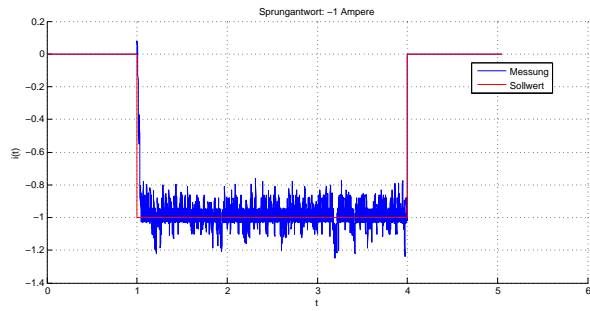
$$R(z) = \frac{0.003657z + 0.003107}{z - 1} \quad (11.1)$$

### 11.1.2.1 Sub-Optimal Stromregler (real)

Wie im Abschnitt 9.3.1 beschrieben, handelt es sich hierbei um einen Integrator mit nicht kontinuierlichem Eingangssignal. Grundsätzlich könnte man daher eine ähnliche Performance wie beim linearen I-Stromregler erwarten. Aufgrund der langen Abtastperiode musste die Konstante  $U$  jedoch eher klein gewählt werden, um im stationären Zustand nicht zu anfällig gegenüber den durch PWM-Betrieb und Kommutierung entstehenden Fluktuationen zu sein.

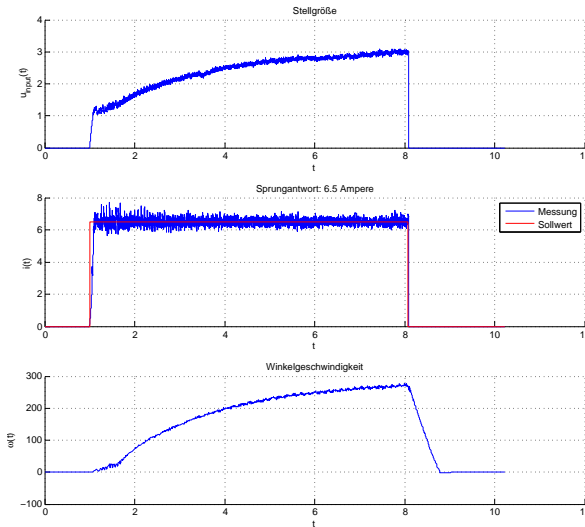


**Abbildung 11.6:** Sprungantwort des 2-SMC Sub. Opt. Alg. bei ein Ampere Sollwert.

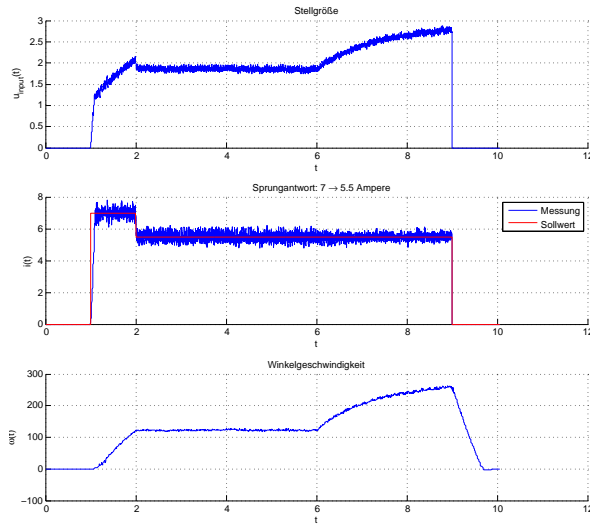


**Abbildung 11.7:** Sprungantwort des 2-SMC Sub. Opt. Alg. bei minus ein Ampere Sollwert.

Durch den betragsmäßig konstanten Integratoreingang und die großen Abtastperiode sind die auftretenden Stromschwankungen höher als beim I-Regler. Auch die Anregelzeit ist wegen der kleinen Konstante  $U$  etwas länger.



**Abbildung 11.8:** Sprungantwort des 2-SMC Sub. Opt. Alg. auf 6.5 Ampere Sollwert.

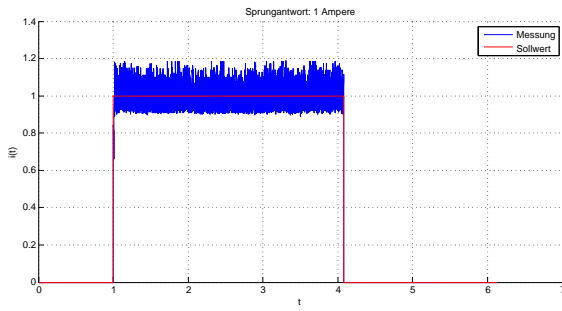


**Abbildung 11.9:** 2-SMC Sub. Opt. Alg. unter Störungseinfluss: Der aktivierte Fahrzeugmotor bei  $t = 6$  hat auch hier keinen Einfluss auf die Stromregelung.

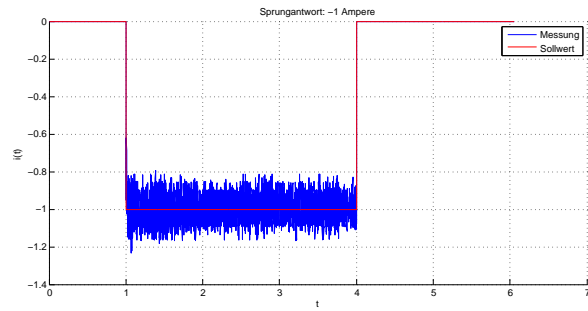
In den beiden Abbildungen 11.8 und 11.9 ist deutlich der lineare Anstieg der Stellgröße durch den konstanten Integratoreingang zu erkennen. Die klein gewählte Konstante  $U$  wirkt sich hier besonders nachteilig aus.

### 11.1.2.2 Super Twisting Stromregler (real)

Das Ergebnis des Super Twisting Stromreglers mit variablem  $\lambda$  ist in den beiden Abbildungen 11.10 und 11.11 dargestellt. Die Anregelzeit ist wegen des zusätzlich zum nichtlinearen Integrator vorhandenen, der Wurzel des Regelfehlers proportionalen Anteils zwar kürzer als beim linearen und beim Sub-Optimal Regler, dafür ist das Rattern um den Sollwert jedoch größer. Weiters zeigt sich in den Abbildungen 11.12 und 11.13, dass die Schwankungen bei höheren Strömen ebenfalls zunehmen, sodass es hier im stationären Bereich zu einer Umschaltung zum größeren Wert für  $\lambda$  kommt. Dies könnte durch einen mit dem Strom wachsenden Schwellenwert für die Umschaltung verhindert werden.

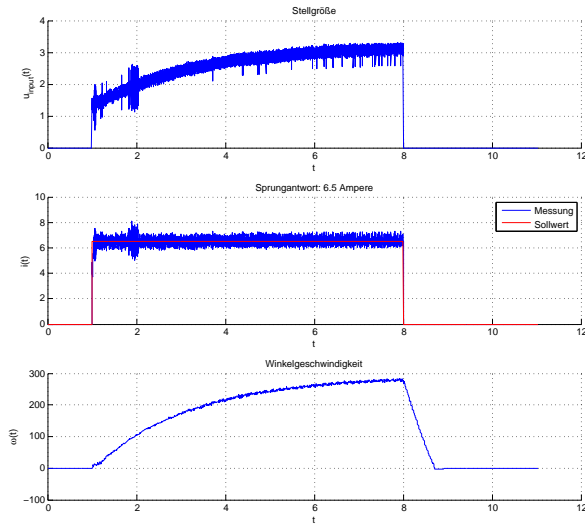


**Abbildung 11.10:** Sprungantwort des 2-SMC Super Twisting Reglers bei ein Ampere Sollwert.

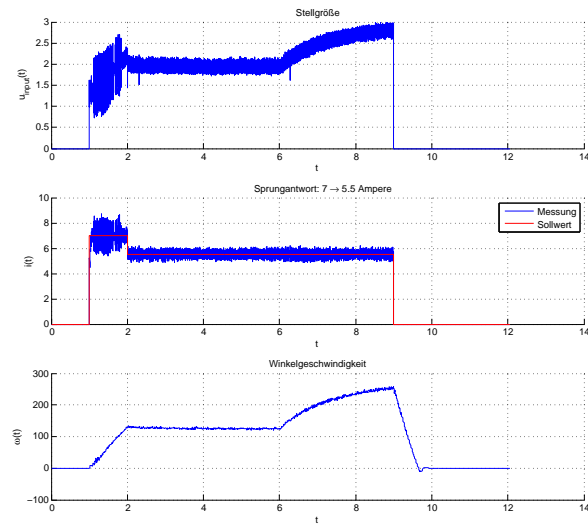


**Abbildung 11.11:** Sprungantwort des 2-SMC Super Twisting Reglers bei minus ein Ampere Sollwert.

Insgesamt zeigt sich jedoch auch an dieser Stelle, dass die lange Abtastzeit den sinnvollen Einsatz dieses Algorithmus verhindert. Die zusätzlichen Maßnahmen würden den Regler einerseits weiter verkomplizieren und andererseits nur zu verhältnismäßig geringen Verbesserungen führen, sodass darauf verzichtet wurde.



**Abbildung 11.12:** Sprungantwort des 2-SMC Super Twisting Reglers auf 6.5 Ampere Sollwert: In der Nähe des Zeitpunktes  $t = 2$  kommt es fälschlicherweise zu einem kurzzeitigen Wechsel von  $\lambda = 0.24$  auf  $\lambda = 0.6$ .



**Abbildung 11.13:** 2-SMC Super Twisting Regler unter Störungseinfluss: Der Fahrzeugmotor wurde bei  $t = 6$  aktiviert, was keine Auswirkung auf die Regelung hatte. Bei sieben Ampere Sollwert tritt aufgrund des unruhigen Stromverlaufs eine Umschaltung auf den größeren Wert des Parameters  $\lambda$  auf.

### 11.1.2.3 Simulationsergebnisse

Die Simulation wurde mit einer Abtastzeit von  $T = 100\mu s$  für den motorischen Betrieb durchgeführt. Bei den Führungsgrößen handelt es sich um Sprungfunktionen von einem und vier Ampere. Da sich bereits bei den Versuchen am realen Prüfstand gezeigt hatte, dass Störungen durch den mechanischen Teil des Systems vernachlässigbar sind, wurde auf entsprechende Simulationen verzichtet. Für einen objektiven Vergleich der Regler wird der mittlere quadratische Fehler (engl. *mean squared error*, *MSE*) berechnet. Zunächst zeigen die Abbildungen 11.14 bis 11.17 die Ergebnisse des linearen Reglers. Man erkennt ein Überschwingen von knapp 25%.

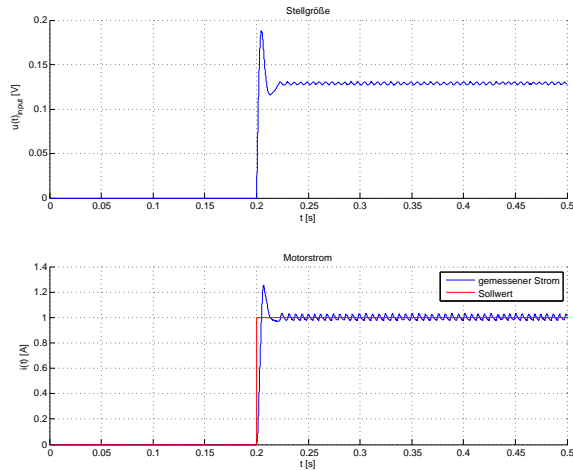


Abbildung 11.14: Sprungantwort des PI-Reglers bei  $i_{ref} = 1$ .

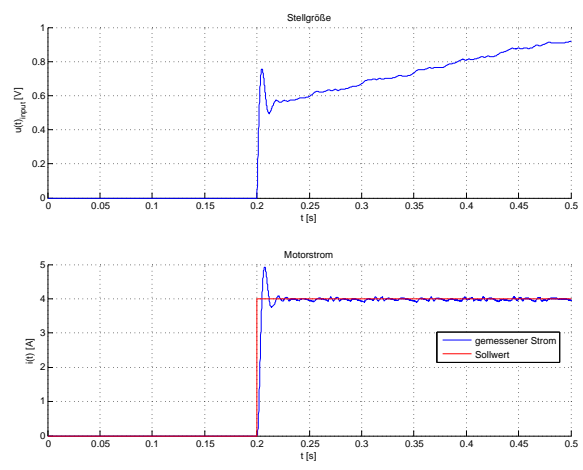


Abbildung 11.15: Sprungantwort des PI-Reglers bei  $i_{ref} = 4$ .

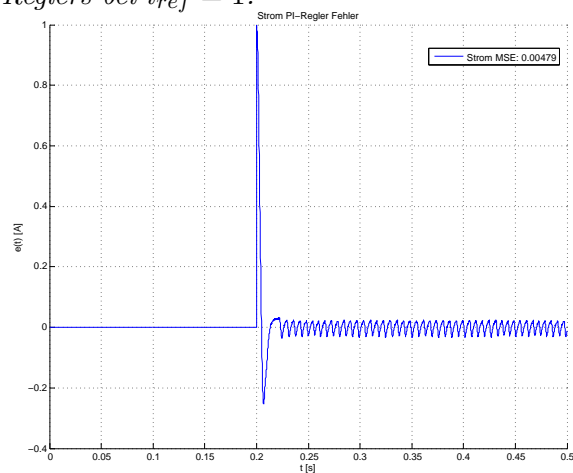


Abbildung 11.16: Regelabweichung des PI-Reglers bei  $i_{ref} = 1$ .

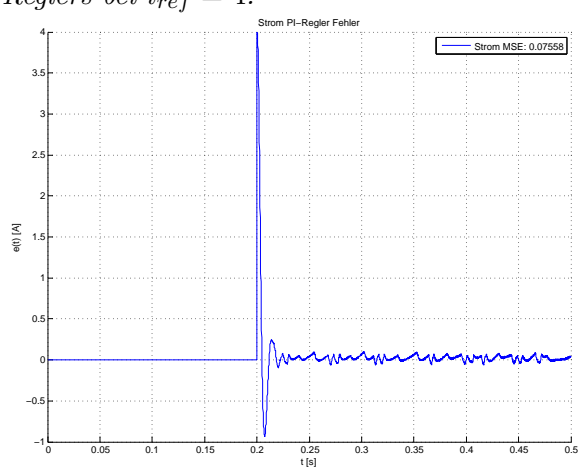


Abbildung 11.17: Regelabweichung des PI-Reglers  $i_{ref} = 4$ .

Die nun folgenden Bilder zeigen die Ergebnisse des 2-SMC Sub-Optimal Stromreglers.

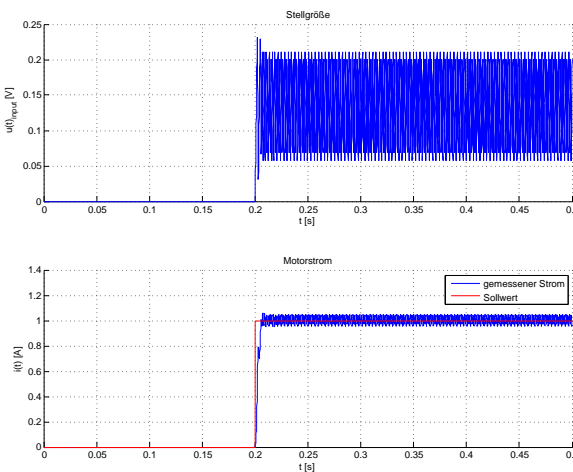


Abbildung 11.18: Sprungantwort des Sub-Optimal Algorithms bei  $i_{ref} = 1$ .

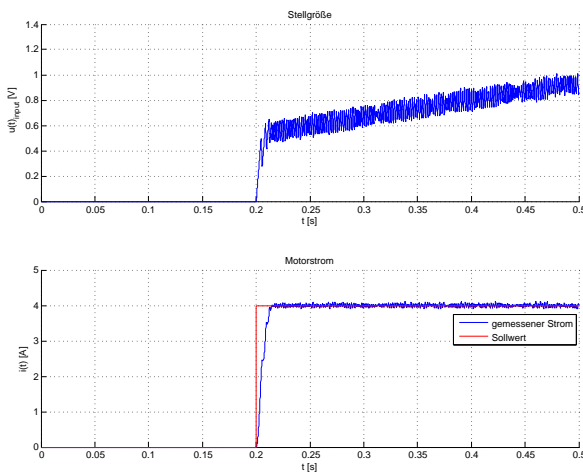
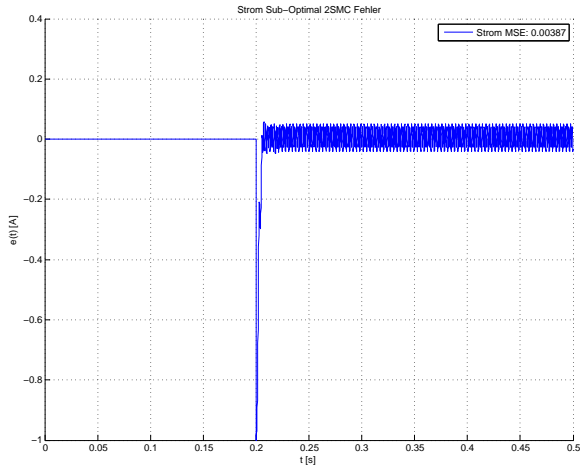
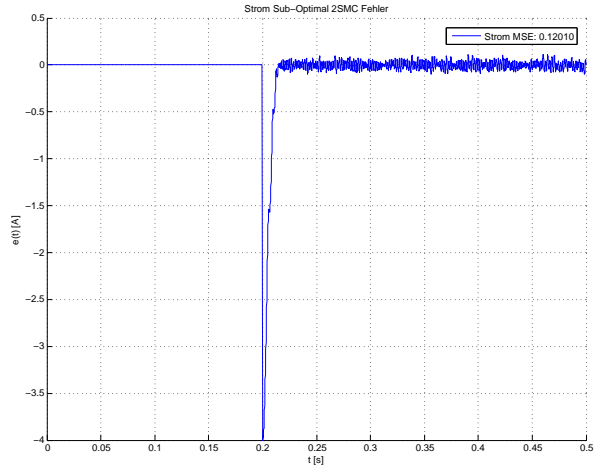


Abbildung 11.19: Sprungantwort des Sub-Optimal Algorithms bei  $i_{ref} = 4$ .

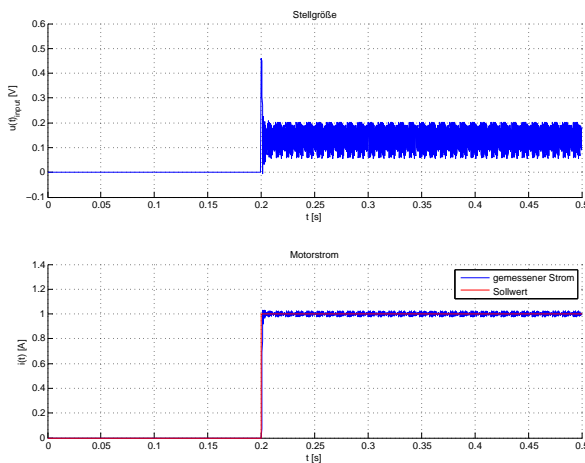


**Abbildung 11.20:** Regelabweichung des Sub-Optimal Algorithms bei  $i_{ref} = 1$ .

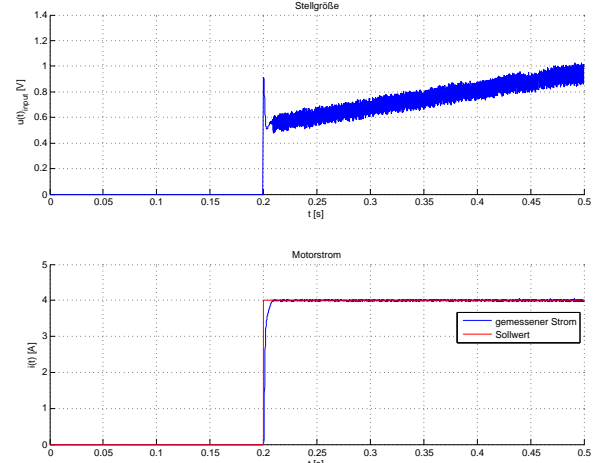
Während bei einem Führungssprung von ein Ampere der Sub-Optimal Regler noch einen etwas kleineren MSE erzielt, schneidet er bei höheren Sprungfunktionen schlechter ab als der lineare Regler. Die Integration eines betragsmäßig konstanten Wertes wirkt sich hier negativ aus, da die Anregelzeit dadurch verlängert wird. Dafür kommt es jedoch zu keinem Überschwingen. Die Resultate des Super Twisting Stromreglers sind in den nun folgenden Abbildungen zu sehen.



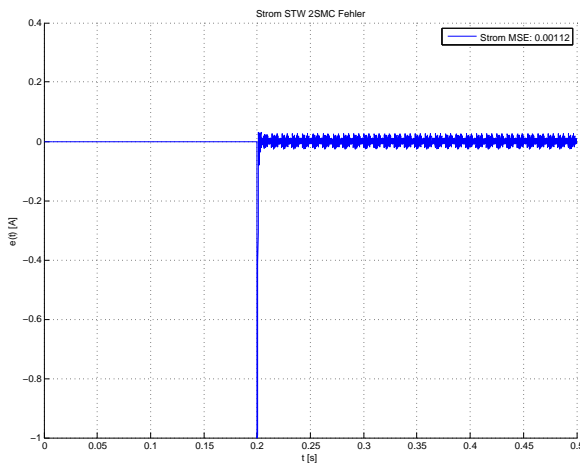
**Abbildung 11.21:** Regelabweichung des Sub-Optimal Algorithms  $i_{ref} = 4$ .



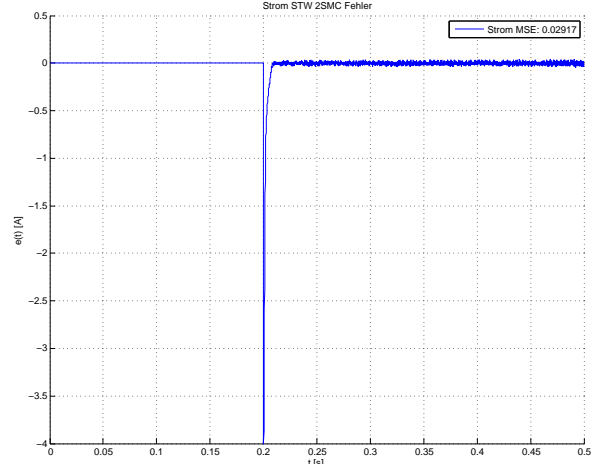
**Abbildung 11.22:** Sprungantwort des Super Twisting Algorithms bei  $i_{ref} = 1$ .



**Abbildung 11.23:** Sprungantwort des Super Twisting Algorithms bei  $i_{ref} = 4$ .



**Abbildung 11.24:** Regelabweichung des Super Twisting Algorithms bei  $i_{ref} = 1$ .



**Abbildung 11.25:** Regelabweichung des Super Twisting Algorithms  $i_{ref} = 4$ .

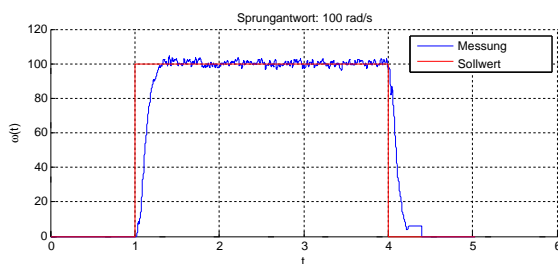


Die Diagramme 11.22 bis 11.25 zeigen die Überlegenheit des Super Twisting Reglers gegenüber den anderen beiden Verfahren. Neben dem kleinsten MSE bietet er den Vorteil, kein Überschwingen zu verursachen.

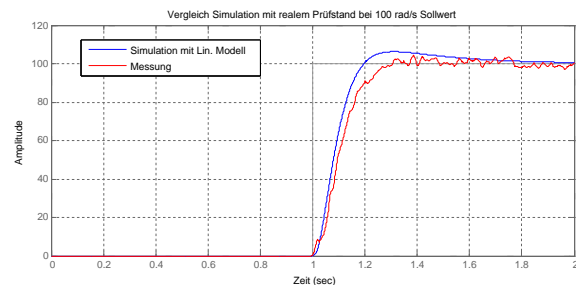
## 11.2 Geschwindigkeitsregelung

### 11.2.1 PI-Kaskadenregelung

Zum Test der Geschwindigkeitsregelung werden zwei unterschiedliche Sprungfunktionen als Referenzgrößen verwendet, wobei einmal die Strombegrenzung der Regelung zum Zuge kommt. Außerdem wird das Verhalten der Regelung bei Störeinflüssen durch das Fahrzeug (Bremsen und Beschleunigen) erfasst.

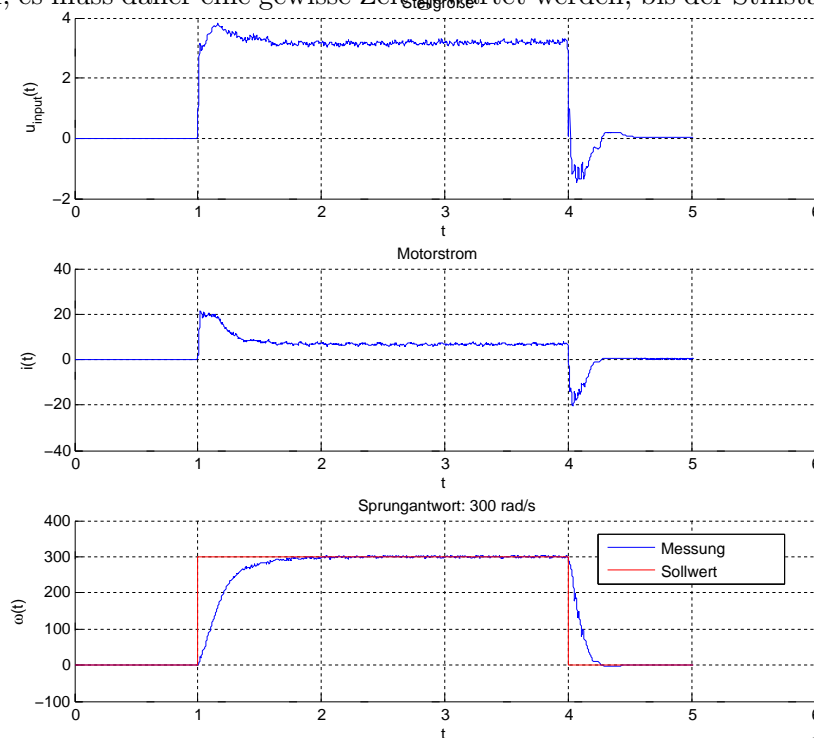


**Abbildung 11.26:** Sprungantwort des Geschwindigkeitsregelkreises auf  $100 \frac{\text{rad}}{\text{s}}$  Sollwert: Es kommt noch zu keiner Strombegrenzung.

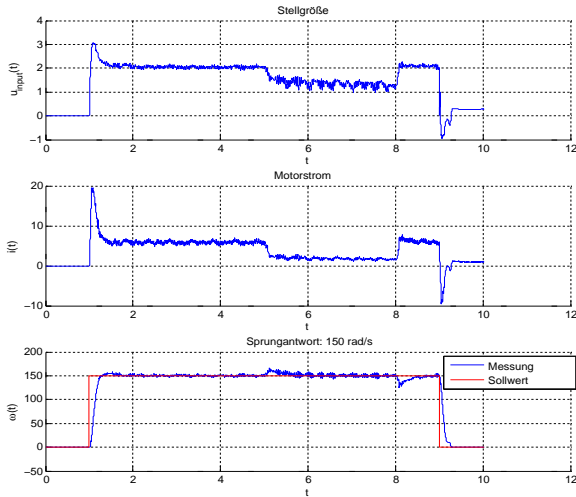


**Abbildung 11.27:** Vergleich der Sprungantworten von Simulation und Messung.

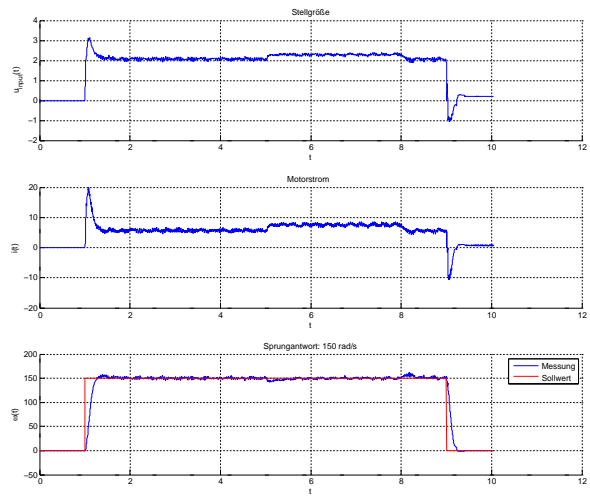
Die Unstetigkeit des Geschwindigkeitsverlaufes in Abbildung 11.26 entsteht durch die Geschwindigkeitsmessung. Diese basiert auf der Zeitmessung zwischen Interrupts. Bei  $0 \frac{\text{rad}}{\text{s}}$  treten keine Interrupts auf, es muss daher eine gewisse Zeit abgewartet werden, bis der Stillstand erkannt wird.



**Abbildung 11.28:** Sprungantwort des Geschwindigkeitsregelkreises auf  $300 \frac{\text{rad}}{\text{s}}$  Sollwert: Die Strombegrenzung lag bei 20 Ampere und wurde hier kurzzeitig erreicht.



**Abbildung 11.29:** Einfluss der Fahrzeugbeschleunigung: Im Zeitraum  $5 \leq t \leq 8$  beschleunigt das Fahrzeug mit 20 Prozent Duty Cycle.



**Abbildung 11.30:** Einfluss eines Fahrzeugbremsvorganges: Im Bereich  $5 \leq t \leq 8$  bremst das Fahrzeug generatorisch mit 80 Prozent Duty Cycle.

## 11.2.2 Second Order Sliding Mode Regler

### 11.2.2.1 Sub-Optimal Algorithm

Dieser Algorithmus benötigt zusätzlich zum aktuellen Wert der Sliding Variable  $s$  die Schnittpunkte der Trajektorie mit der Abszisse der  $s - \dot{s}$ -Ebene, was die Fehleranfälligkeit erhöht. Durch den Einsatz des nichtlinearen Vorfilters (siehe 10.2.1.1) konnte das Ergebnis verbessert werden. Die Schwingungen um den Sollwert sind im Vergleich zum PI-Regler jedoch etwas größer. Die Anregelzeit ist bei  $100 \frac{\text{rad}}{\text{s}}$  (siehe Abb. 11.31) in etwa gleich lang wie beim linearen Regler, bei  $300 \frac{\text{rad}}{\text{s}}$  (Abb. 11.32) ist der Sub-Optimal Regler sogar etwas schneller.



**Abbildung 11.31:** Sprungantwort des Sub-Optimal Algorithm bei  $100 \frac{\text{rad}}{\text{s}}$  Sollwert.

In den beiden Abbildungen 11.33 und 11.34 sind die Ergebnisse unter Störeinfluss des Fahrzeuges zu sehen. Beschleunigt das Auto mit 20 Prozent, kommt es zu einer leichten Verstärkung des Ratterns. Dies liegt daran, dass bei dieser Wahl der Führungs- und Störgröße ein ständiges Umschalten zwischen motorischem und generatorischem Betrieb notwendig ist. Bremst das Auto hingegen, so sind auch hier kaum Auswirkungen auf die Geschwindigkeit zu beobachten.

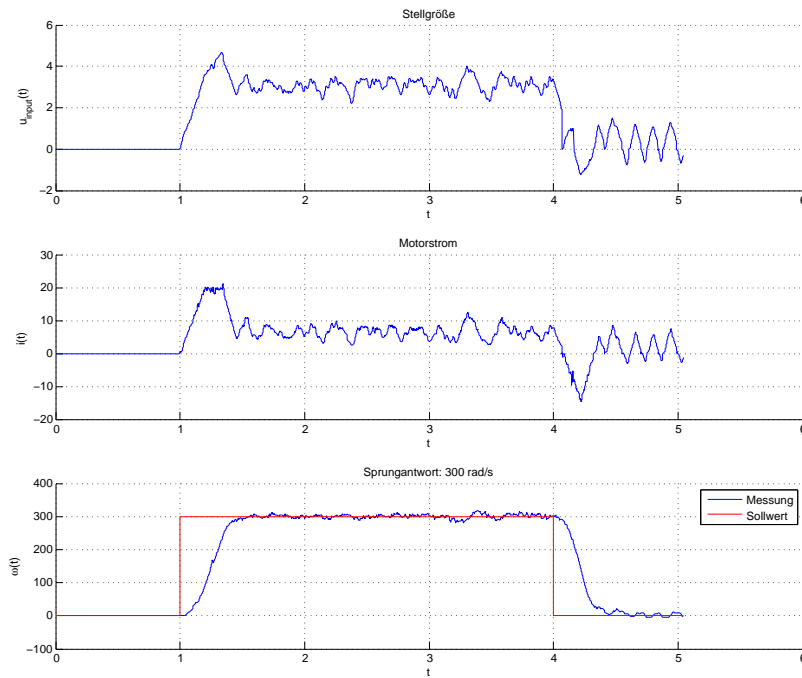


Abbildung 11.32: Sprungantwort des Sub-Optimal Alg. auf  $300 \frac{\text{rad}}{\text{s}}$  Sollwert: Einhaltung der Strombegrenzung von 20 Ampere.

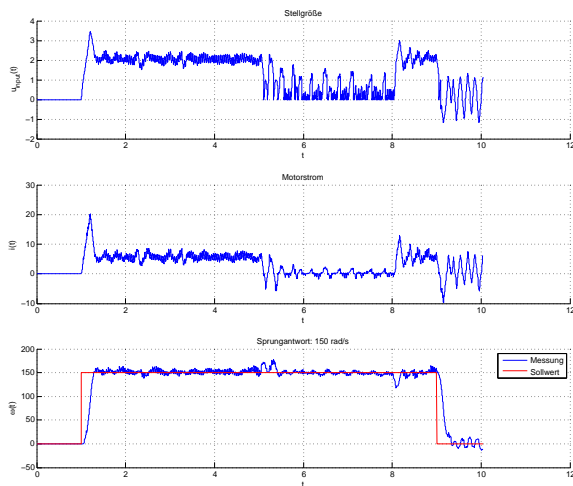


Abbildung 11.33: Einfluss der Fahrzeugbeschleunigung: Im Zeitraum  $5 \leq t \leq 8$  beschleunigt das Fahrzeug mit 20 Prozent Duty Cycle.

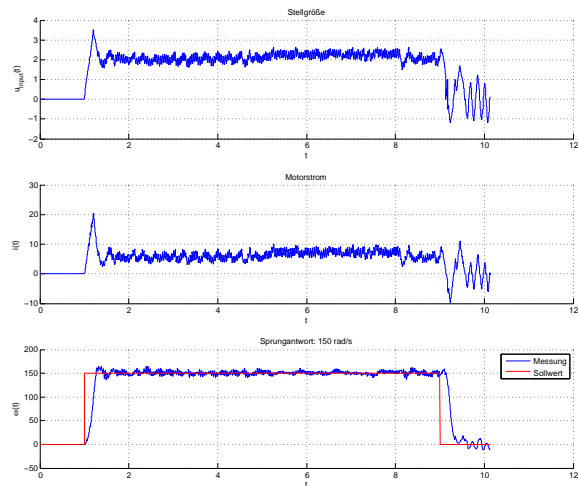


Abbildung 11.34: Einfluss eines Fahrzeugbremsvorganges: Im Bereich  $5 \leq t \leq 8$  bremst das Fahrzeug generatorisch mit 80 Prozent Duty Cycle.

### 11.2.2.2 Super Twisting Algorithm

Der Super Twisting Geschwindigkeitsregler weist die schnellste Anregelzeit von allen drei getesteten Verfahren auf. Das Rattern um den Sollwert ist geringer als beim Sub-Optimal Regler, insbesondere bei höheren Geschwindigkeiten. Die Laufruhe des PI-Reglers ist jedoch bei niedrigen Geschwindigkeiten noch etwas besser.

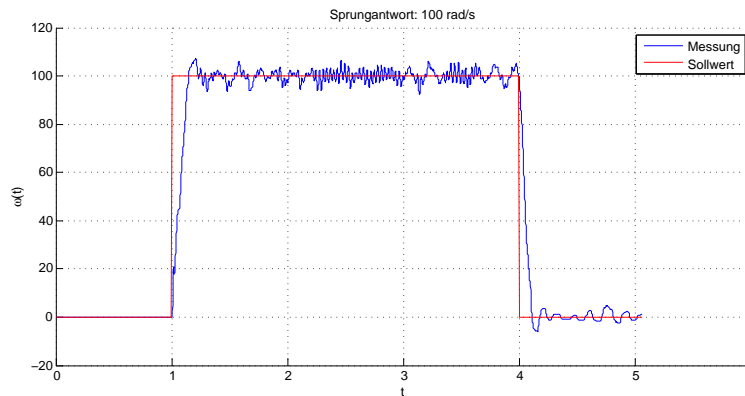


Abbildung 11.35: Sprungantwort des Super Twisting Algorithm bei  $100 \frac{\text{rad}}{\text{s}}$  Sollwert.

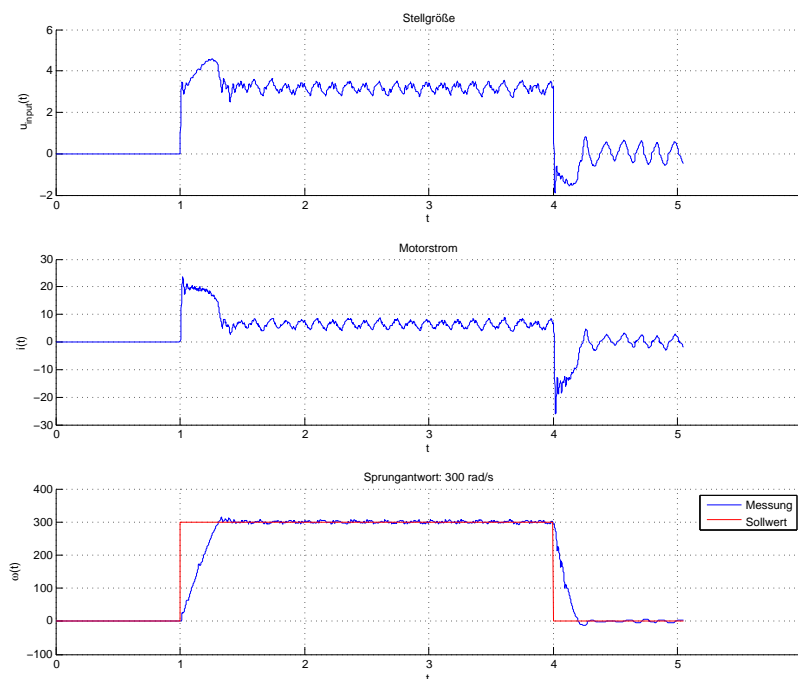
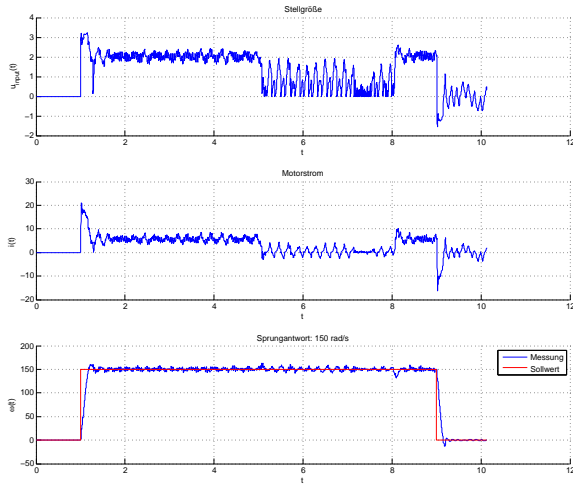
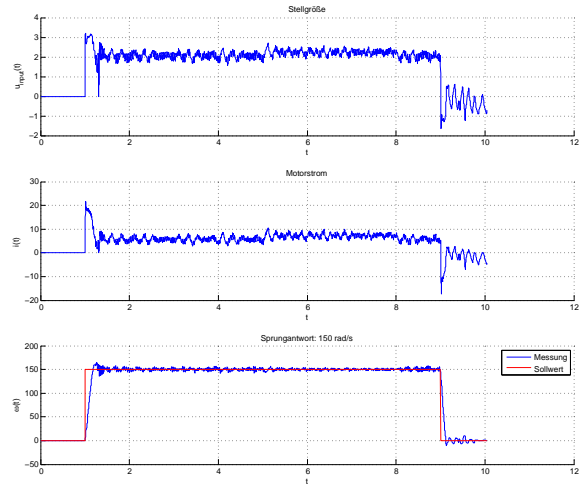


Abbildung 11.36: Sprungantwort des Super Twisting Alg. auf  $300 \frac{\text{rad}}{\text{s}}$  Sollwert: Einhaltung der Strombegrenzung von 20 Ampere.

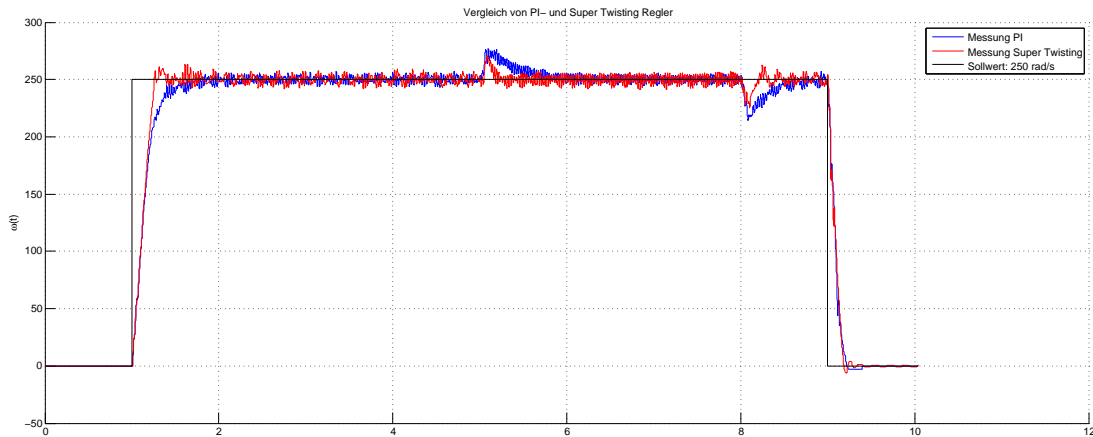
Besonders vorteilhaft ist das schnellere Ausregeln von Störungen durch das Fahrzeug, egal ob es sich um ein positives oder ein negatives Lastmoment handelt (siehe Abbildungen 11.37 und 11.38). Zuletzt zeigt Diagramm 11.39 noch eine Gegenüberstellung der Reaktionen von PI- und Super Twisting Regler auf einen Fahrzeugbeschleunigungsvorgang mit 30 Prozent Tastverhältnis.



**Abbildung 11.37:** Einfluss der Fahrzeugbeschleunigung: Im Zeitraum  $5 \leq t \leq 8$  beschleunigt das Fahrzeug mit 20 Prozent Duty Cycle.



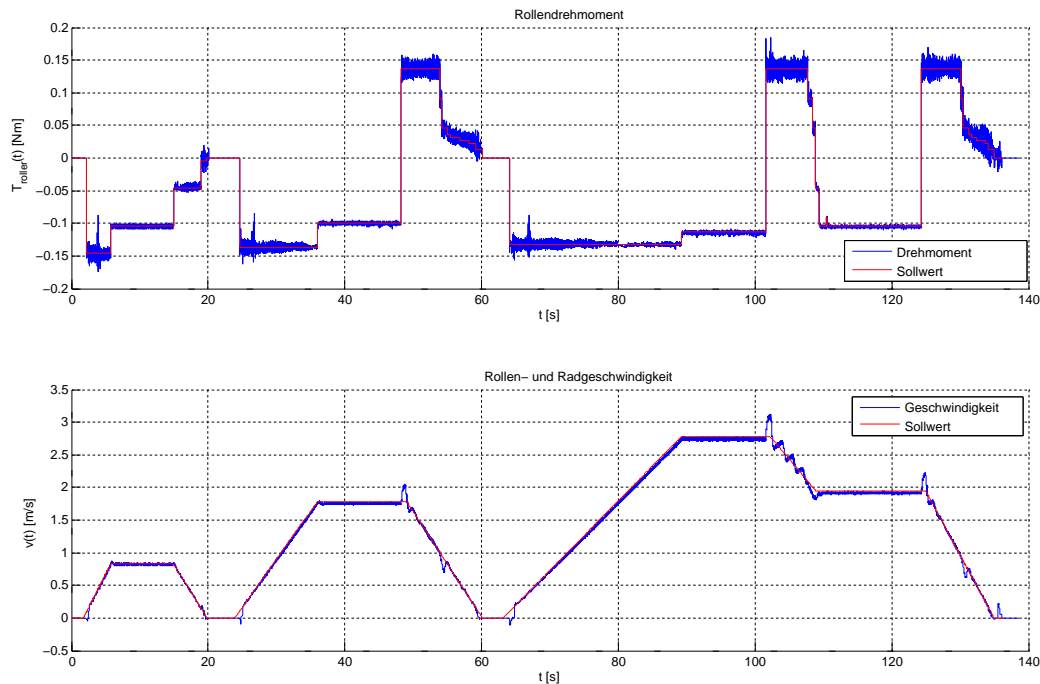
**Abbildung 11.38:** Einfluss eines Fahrzeugbremsvorganges: Im Bereich  $5 \leq t \leq 8$  bremst das Fahrzeug generatorisch mit 80 Prozent Duty Cycle.



**Abbildung 11.39:** Direkter Vergleich von PI- und Super Twisting Regler: Die Störungen des Fahrzeuges werden schneller ausgeglet. Bei höheren Geschwindigkeiten ist auch die Laufruhe beider Regler in etwa gleich.

### 11.3 Fahrzyklus

Dieser Test besteht aus dem Abfahren eines etwa zweiminütigen Fahrzyklus. Der Rollenprüfstand erzeugt dabei an den Rollen ein positives oder negatives Drehmoment, während das Fahrzeug einem vorgegebenen Geschwindigkeitsverlauf zu folgen hat. Da beim Fahrzeug noch die ursprünglichen Sensoren eingesetzt werden (Probleme: siehe Abschnitt 1.3.2), kann auch die Regelung nicht verbessert werden, was sich in schlechtem Führungsverhalten beim Bremsen äußert. Da bereits bei der Betrachtung der Ergebnisse aus Kapitel 11.1 deutlich wurde, dass unter den gegebenen Bedingungen der lineare Stromregler am besten funktioniert, werden hier nur dessen Ergebnisse vorgestellt.



**Abbildung 11.40:** Test der Momentenregelung des Prüfstandes bei einem Fahrzyklus: Da Rollen- und Radgeschwindigkeiten übereinstimmen, kann auch das Regelverhalten des Fahrzeuges vom Prüfstand aus gemessen werden. Die größten Abweichungen der Momentenregelung treten in den Bereichen mit den vom Fahrzeug verursachten Geschwindigkeitsschwankungen auf. Kurze Einbrüche beim negativen Drehmoment treten in etwa zu den Zeitpunkten  $t = 4$ ,  $t = 26$  und  $t = 66$  auf und sind im Umschalten vom Gegenstrombremsen zum generatorischen Bremsen begründet.

## 11.4 Zusammenfassung

Das Ziel dieser Masterarbeit war der Entwurf einer Momenten- und einer Geschwindigkeitsregelung für einen elektrischen Rollenprüfstand im Maßstab 1:5, um die Simulation diverser Belastungssituationen eines zugehörigen Elektrofahrzeuges zu ermöglichen. Dazu zählen beispielsweise Steigungen und Gefälle der Straße, der Luftwiderstand oder die Fahrzeugträgheit. Beide Komponenten werden von permanentmagneterregten Gleichstrommaschinen angetrieben und besitzen Sensoren zu Spannungs-, Strom- und Drehzahlmessung. Die Messdatenerfassung und die Steuerung der Motoren erfolgen jeweils über einen 8-Bit Mikrocontroller für die entwickelten Regelalgorithmen in C implementiert wurden. Ein Flexray-Netzwerk ermöglicht die Kommunikation eines PCs mit den beiden Mikrocontrollern, um die Vorgabewerte für die Regelungen zu senden sowie die aktuellen Messdaten zu erhalten. Die gewählte Softwarestruktur gewährleistet die Einhaltung der Abtastperiode von einer Millisekunde, die für den zeitdiskreten Entwurf der Regelungen berücksichtigt wurde. Die Softwareschnittstelle zur Kommunikation mit Prüfstand und Fahrzeug erlaubt neben der Messdatenübertragung und Vorgabe von Sollwert, Regelgröße (Drehmoment, Strom oder Geschwindigkeit) und Reglerart auch die Parameterkonfiguration der Algorithmen. Mithilfe dieser Kommunikationsbibliothek wurde auch eine Anbindung der beiden Komponenten an Matlab/Simulink realisiert.

Im nächsten Schritt wurde ein mathematisches Modell des Rollenprüfstandes erstellt. Grundlage dafür war die Zustandsraumdarstellung der permanentmagneterregten Gleichstrommaschine, die jedoch aufgrund der modifizierten H-Brückensteuerung um nichtlineare Terme erweitert werden musste. Durch gezielte Experimente und numerische Optimierung konnten die Parame-

ter des Modells bestimmt werden. Neben einem neuen Strommesssensor wurde auch ein neuer zweikanaliger Drehgeber eingebaut. Im Zuge der Überlegungen hinsichtlich der zu erwartenden Messgenauigkeit stellte sich heraus, dass unter den vorliegenden Bedingungen eine Berechnung der Winkelgeschwindigkeit über die Periodendauermessung vorteilhaft ist.

Das Hauptaugenmerk der Arbeit liegt auf dem Entwurf und der Implementierung verschiedener Regelungsalgorithmen für Strom, Drehmoment und Geschwindigkeit. Zur Einstellung eines vorgegebenen Motorstromes kommt ein einschleifiger Regelkreis zum Einsatz, der in Abhängigkeit vom Sollwert und der aktuellen Drehzahl zwischen motorischem und generatorischem Betrieb der elektrischen Maschine umschaltet. Für die Geschwindigkeitsregelung wird auf eine Kaskadenstruktur zurückgegriffen, da diese die Berücksichtigung von Strombegrenzungen, die eine Beschädigung des Motors und der Elektronik verhindern, erleichtert. Bei bekannten Streckenparametern kann aus einem gegebenen Drehmoment der für dessen Erzeugung benötigte Motorstrom berechnet werden, sodass die Momentenregelung äquivalent zur Stromregelung ist. Da sich bei der Bestimmung der Modellparameter teilweise große Unsicherheiten gezeigt hatten, fiel die Entscheidung, zusätzlich zu den anfangs entworfenen PI-Reglern robuste Algorithmen in Form von Second Order Sliding Mode Control zu verwenden. Aus dieser Klasse wurden zwei Verfahren, der *Sub-Optimal Algorithm* und der *Super Twisting Algorithm*, ausgewählt und als Strom- und Geschwindigkeitsregler verwendet. Aufgrund der hohen Abtastperiode konnten die Sliding Mode Stromregler nicht mit dem entworfenen linearen Regler konkurrieren. In der Simulation zeigte sich jedoch, dass bei niedrigeren Abtastzeiten der Vergleich zugunsten des Super Twisting Reglers ausgeht. Im Falle der Geschwindigkeitsregelung kann der Super Twisting Algorithmus bereits unter den derzeit vorliegenden Bedingungen eine kürzere Anregelzeit und eine bessere Störgrößenunterdrückung am realen Prüfstand erreichen. Der Sub-Optimal-Regler leidet zusätzlich zur hohen Abtastzeit unter einem erhöhten Informationsbedarf bezüglich des Regelfehlers (Stichwort *Peak Detection*), dem man im Falle eines unruhigen und verrauschten Geschwindigkeitsverlaufes nur schwer gerecht werden kann. Die zugehörigen Berechnungen führen außerdem zu einem höheren Rechen- und Speicheraufwand sowie zu zusätzlich einzustellenden Parametern. Der Super Twisting Regler hingegen hat wie ein PI-Regler nur zwei Parameter, die sich relativ einfach nach dem Prinzip von „Trial & Error“ finden lassen.

## 11.5 Mögliche zukünftige Erweiterungen

1. **Fahrzeugumbau:** Der am Rollenprüfstand vorgenommene Umbau der Sensorik wurde am Fahrzeug noch nicht durchgeführt. Daher ist dort die Strommessung stark fehlerbehaftet und in einigen Betriebspunkten vorzeichenfalsch sowie die erfasste Drehzahl vorzeichenlos. Ein systematischer Regelungsentwurf war dort folglich nicht möglich. Vor allem in Hinblick auf das Abfahren von Fahrzyklen mit aktivierter Geschwindigkeitsregelung des Fahrzeuges dürfte sich ein Umbau jedoch bezahlt machen, da mit einer Reduktion des Überschwingens beim Bremsen zu rechnen ist. Außerdem führt ein Fehler in der Platine zu einem Kurzschluss beim Rückwärtsfahren des Autos.
2. **Leistungsfähigerer Mikrocontroller:** Die beschränkte Rechenleistung des ATmega128 Mikrocontrollers verhinderte die Erhöhung der Abtastrate, die ein deutlich besseres Ergebnis der Sliding Mode Regler erwarten lässt. Insbesondere die Verarbeitung der flexray-spezifischen Interrupts, die jeweils 150  $\mu\text{s}$  dauert, stellt ein Problem dar.





# Literaturverzeichnis

- [Adamy 2009] ADAMY, J.: *Nichtlineare Regelungen*. Springer-Verlag Berlin Heidelberg, 2009. – ISBN 9783642007934
- [Atmel Corporation 2011] Atmel Corporation (Veranst.): *ATmega128A: 8-bit AVR Microcontroller with 128KBytes In-System Programmable Flash*. 2011
- [Bartolini u. a. 2003] BARTOLINI, G. ; PISANO, A. ; PUNTA, E. ; USAI, E.: A survey of applications of second-order sliding mode control to mechanical systems. In: *International Journal of Control* 76 (2003), Nr. 9-10, S. 875–892
- [Fridman und Levant 2002] FRIDMAN, L. ; LEVANT, A.: Higher Order Sliding Modes. In: BARBOT, J.-P. (Hrsg.) ; PERRUGUETTI, W. (Hrsg.): *Sliding Mode Control in Engineering*. CRC Press, 2002 (Automation and Control Engineering)
- [Geimer u. a. 2006] GEIMER, M. ; KRÜGER, T. ; LINSEL, P.: Co-Simulation, gekoppelte Simulation oder Simulatorkopplung? Ein Versuch der Begriffsvereinheitlichung. In: *O+P Zeitschrift für Fluidtechnik* (2006), S. 572–576. – URL <http://digbib.ubka.uni-karlsruhe.de/volltexte/1000011318>
- [Hung u. a. 1993] HUNG, J. Y. ; GAO, W. ; HUNG, J. C.: Variable Structure Control: A Survey. In: *IEEE Transactions On Industrial Electronics* 40 (1993), Nr. 1, S. 2–22
- [León und Kiencke 2011] LEÓN, F.P. ; KIENCKE, U.: *Messtechnik: Systemtheorie für Ingenieure und Informatiker*. 8. Auflage. Springer-Verlag Berlin Heidelberg, 2011. – ISBN 9783642202384
- [Lunze 2010a] LUNZE, J.: *Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen*. 8. Auflage. Springer-Verlag Berlin Heidelberg, 2010 (Springer-Lehrbuch). – ISBN 9783642138072
- [Lunze 2010b] LUNZE, J.: *Regelungstechnik 2: Mehrgrößensysteme, Digitale Regelung*. 6. Auflage. Springer-Verlag Berlin Heidelberg, 2010 (Springer-Lehrbuch). – ISBN 9783642101977
- [Pisano 2000] PISANO, A.: *Second Order Sliding Modes: Theory and Applications*, Università di Cagliari, Dissertation, 2000
- [Pisano u. a. 2008] PISANO, A. ; DAVILA, A. ; FRIDMAN, L. ; USAI, E.: Cascade Control of PM DC Drives Via Second-Order Sliding-Mode Technique. In: *IEEE Transactions On Industrial Electronics* 55 (2008), Nr. 11, S. 3846–3854
- [Schlecht 2009] SCHLECHT, B.: *Maschinenelemente 2: Getriebe, Verzahnungen, Lagerungen*. Pearson Studium, 2009. – ISBN 9783827371461
- [Schröder 2009a] SCHRÖDER, D.: *Elektrische Antriebe: Grundlagen*. 4. Auflage. Springer-Verlag Berlin Heidelberg, 2009 (Springer-Lehrbuch). – ISBN 9783642029899

- [Schröder 2009b] SCHRÖDER, D.: *Elektrische Antriebe: Regelung von Antriebssystemen*. 3. Auflage. Springer-Verlag Berlin Heidelberg, 2009 (Springer-Lehrbuch). – ISBN 9783540896128
- [Zimmermann und Schmidgall 2007] ZIMMERMANN, W. ; SCHMIDGALL, R.: *Bussysteme in der Fahrzeugtechnik: Protokolle und Standards*. 2. Auflage. Vieweg, 2007 (ATZ-MTZ Fachbuch). – ISBN 9783834802354