

Fakultät für Ingenieurwissenschaften

Studiengang
Maschinenbau

**Bau und Simulation eines mobilen Hardware-Modells
zur Lageregelung eines Satelliten**

Bachelor Thesis

von

Eduard Ober

Datum der Abgabe: 16.12.2022
Erstprüfer: Prof. Dr.-Ing. Peter Zentgraf
Zweitprüfer: Prof. Dr. Ing. Franz Stubenrauch

Kurzfassung

Es wurde ein Modell eines Satelliten entworfen, der sich wie sein großes Vorbild frei um die eigene Achse drehen kann. Dies wird durch ein mithilfe eines bürstenlosen Gleichstrommotors angetriebenes Drallrad ermöglicht, das sich auf der Oberseite des Körpers befindet. Die passende Beschleunigung wird über einen Raspberry Pi Zero WH als Mikrocontroller errechnet. Dabei wird die Lage des Satelliten so geregelt, dass sie unempfindlich gegenüber Störungen ist. Es werden insbesondere Störungen betrachtet, die durch eine Auslenkung des Satellitenkörpers von Hand entstehen. Die aktuelle Lage des Satelliten wird über einen berührungslosen Lagesensor gemessen. Das Verhalten des Satelliten in der Realität soll dabei gleichzeitig mit der Bewegung eines digitalen Zwillings über Simulink graphisch auf dem Bildschirm verglichen werden können.

Schlagwörter: Satellitentechnik, Lageregelung, Mobiler Demonstrationsversuch, pzMove, MATLAB/ Simulink, Raspberry Pi Zero, Bürstenloser Gleichstrommotor

Abstract

A model of a satellite was designed such that like its larger counterpart it can rotate freely around its own axis. This is made possible by a momentum wheel powered by a brushless DC motor located on top of the body. The appropriate acceleration is calculated using a Raspberry Pi Zero WH as a microcontroller. The position of the satellite is controlled in such a way that it is insensitive to interference. Disturbances caused by manual deflection of the satellite body are considered in particular. The current position of the satellite is measured using a non-contact position sensor. The behaviour of the satellite in reality should be able to be compared with the movement of a digital twin graphically via Simulink.

Keywords: satellite technology, attitude control, mobile demonstration experiment, pzMove, MATLAB/ Simulink, Raspberry Pi Zero, brushless DC motor

Inhaltsverzeichnis

Kurzfassung	II
Abstract	II
Inhaltsverzeichnis	III
Abbildungsverzeichnis	V
Tabellenverzeichnis	VI
Abkürzungsverzeichnis	VII
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung und Randbedingungen	1
2 Entwicklung des Versuchs	2
2.1 Genaue Beschreibung des Versuchsaufbaus	2
2.1.1 Der mechanische Aufbau	2
2.1.2 Lithium-Polymer-Akku-Pack	3
2.1.3 Der Electronic Speed Controller des bürstenlosen Gleichstrommotors	4
2.1.4 Der magnetische induktive Winkelaufnehmer als Lagesensor.....	6
2.2 Die Motoransteuerung des Drallrads zur Veränderung der Lage.....	7
2.2.1 Arbeiten mit dem Mikrocontroller Raspberry Pi Zero WH	7
2.2.2 Erstellte Simulink-Programme	9
2.3 Modellierung der Lagedynamik des Satelliten.....	12
2.3.1 Physikalische Modellierung der Regelstrecke mit dem Drallsatz.....	12
2.3.2 Experimentelle Identifikation der Regelstrecke mit dem Regelungstechnikprogramm pzMove.....	14
3 Das Stellglied	18
3.1 Das Motordrehzahlverhalten durch den Electronic Speed Controller.....	18
3.2 Auswahl des Stellsignals	20
4 Messung des Ein- und Ausgangsverhaltens	22
4.1 Die nichtlineare Natur der Strecke	22
4.2 Bestimmung der Reibung	25
4.3 Überprüfen des normierten geschlossenen Regelkreisverhaltens anhand von Beispielreglern.....	27
4.4 Durchführung der Identifikation des Streckenmodells.....	30
4.5 Vergleich der realen Strecke mit der simulierten Strecke	31

4.6	Vergleich zwischen physikalischen und experimentellen Lagemodell des Satelliten	32
5	Der Reglerentwurf.....	35
5.1	Bestimmen eines Reglers über die Einstellregeln von Chien, Hrones und Reswick	35
5.2	Bestimmen eines Reglers unter Berücksichtigung der Systemgrenzen	37
5.2.1	Verschiedene Regler-Vorschläge	37
5.2.2	Anwendung des entworfenen Reglers an realer Strecke und Vergleich mit der Simulation	39
5.2.3	Bewertung der Regelgüte	41
6	Praktische Demonstration	46
6.1	Digitaler Zwilling des geschlossenen Regelkreises	46
6.2	Der Laserpointer-Aufbau zur Veranschaulichung der aktuellen Lage.....	47
7	Zusammenfassung und Ausblick	48
7.1	Resümee der Ergebnisse.....	48
7.2	Mögliche Perspektiven für Weiterentwicklungen	48
Anhang:	i
MATLAB-Programm – Messdaten_zusammenf.m.....		i
MATLAB-Programm – identSystemvgl.m		ii
MATLAB-Programm – reg_Messdaten_zusammenf.m.....		iii
MATLAB-Programm – mat_reg_plot.m.....		iii
Literaturverzeichnis		v
Eidesstattliche Versicherung		vi
Auszug aus dem Strafgesetzbuch (StGB).....		vi

Abbildungsverzeichnis

Abbildung 1: CAD-Darstellung der ersten Version des Aufbaus.....	2
Abbildung 2: Der verwendeter LiPo-Akku-Pack von ACME-online.de.....	3
Abbildung 3: Schaltplan Flycolor Raptor F390 BLHeli 30A 2-4S OPTO Brushless ESC [4].....	4
Abbildung 4: BLHeli-Software [6].....	5
Abbildung 5: HALJIA AS5600 Magnetic Encoder Module [7] im Satellitenkörper.....	6
Abbildung 6: Simulink-Programm für Winkelaufnehmer.....	6
Abbildung 7: Elektronik im Inneren des Satellitenkörpers	7
Abbildung 8: Hardware-Setup über den Add-On Manager in MATLAB.....	8
Abbildung 9: Programm für Motordrehzahlmessung <i>MotorRun_PWM.slx</i>	9
Abbildung 10: Programm zur Aufnahme der Systemantworten <i>MotorRun_spd.slx</i>	10
Abbildung 11: Regelungsprogramm <i>MotorRun_Reg_Sim.slx</i>	11
Abbildung 12: Auswahl der Zähler- und Nenner-Ordnung in pzMove.....	15
Abbildung 13: Auswahl der Filteroptionen in pzMove.....	15
Abbildung 14: Darstellung des Eingangs- und Ausgangsgraphen in pzMove.....	16
Abbildung 15: Auswahl an Reglerentwurfverfahren in pzMove.....	17
Abbildung 16: Präparierter Motor	18
Abbildung 17: Durch Lasertachometer gemessene Messdaten graphisch dargestellt....	19
Abbildung 18: Vergleich des Geradenmodelles mit der Messung	20
Abbildung 19: Normierte Rampenantworten für zwei verschiedene Rampensteigungen positiv und negativ	22
Abbildung 20: Langsam aufsteigende Winkelposition des Satelliten	23
Abbildung 21: Langsam fallende Winkelposition des Satelliten.....	24
Abbildung 22: Versuch Haftreibung zu bestimmen	25
Abbildung 23: Normierte Rampenantworten, um Linearität des Systems zu bestimmen	26
Abbildung 24: Normierte Sprungantworten, um Linearität des Systems zu bestimmen	27
Abbildung 25: Normierte Graphen des P-geregelten geschlossenen Regelkreises $K_p=200$	28
Abbildung 26: Normierte Graphen des PD-geregelten geschlossenen Regelkreises $K_p=400$; $K_d=200$; $T_1=0.1$ s.....	29
Abbildung 27: Für die Identifikation verwendeter Messdatensatz.....	30
Abbildung 28: Vorgabe der Zähler- und Nenner-Ordnung in pzMove.....	30
Abbildung 29: Rampenantwort einer Eingangsrampe mit Steigung $u =$ 500 rad/s^3	31
Abbildung 30: Sinusantwort mit einer von Amplitude $u=500 \text{ rad/s}^2$ und einer Kreisfrequenz von 2 1/s	32
Abbildung 31: Darstellung der Kennwerte in der Regelstreckensprungantwort [13, Kap. 7 Seite 15].....	35
Abbildung 32: Sprungantwort-Simulation des P-Reglers nach den Einstellregeln von Chien, Hrones und Reswick.....	36

Abbildung 33: Sprungantwort-Simulation des PD-Reglers nach den Einstellregeln von Chien, Hrones und Reswick.....	36
Abbildung 34: Sprungantwort-Simulation des PID-Reglers nach den Einstellregeln von Chien, Hrones und Reswick.....	37
Abbildung 35: Sprungantwort-Simulation des Reglers A	38
Abbildung 36: Sprungantwort-Simulation des Reglers B	38
Abbildung 37: Sprungantwort-Simulation des Reglers C	38
Abbildung 38: Reale Systemreaktion in Vergleich zur Simulation im Regelkreis A	39
Abbildung 39: Reale Systemreaktion in Vergleich zur Simulation im Regelkreis B.....	40
Abbildung 40: Reale Systemreaktion in Vergleich zur Simulation im Regelkreis C.....	41
Abbildung 41: Störkompensation des Regelkreis A durch Auslenken des Satelliten von Hand	43
Abbildung 42: Störkompensation des Regelkreis B durch Auslenken des Satelliten von Hand	44
Abbildung 43: Störkompensation des Regelkreis C durch Auslenken des Satelliten von Hand	44
Abbildung 44: Darstellung des Demonstrationsprogramms.....	46
Abbildung 45: Gesamtaufbau des Demonstrationsversuchs	47

Tabellenverzeichnis

Tabelle 1: Ungefähre Massenträgheitsmomente.....	34
Tabelle 2: Vergleich zwischen den Reglern hinsichtlich des Führungsverhaltens.....	42

Abkürzungsverzeichnis

LiPo-Akku	Lithium-Polymer-Akku
PWM	Pulsweitenmodulation
P-Regler	Proportional-Regler
PI-Regler	Proportional- und Integral-Regler
PD-Regler	Proportional- und Differential-Regler
PID-Regler	Proportional-, Integral- und Differential-Regler
ESC	Electronic Speed Controller
I2C	Inter-Integrated Circuit
rpm	Umdrehungen pro Minute

1 Einleitung

1.1 Motivation

In der Regelungstechnik begegnet dem Studenten erstmals viele zuerst nicht intuitive Konzepte, was es schwierig macht, sich in das Fach hineinzudenken. Um es den interessierten Studenten zu erleichtern und die Regelungstechnik begreifbar zu machen, soll in Rahmen dieser Arbeit ein Demonstrationsversuch angefertigt werden, der einen sich im All befindenden Satelliten darstellt. Satelliten sind in unserer modernen Welt nicht mehr wegzudenken und erfüllen viele Aufgaben. So werden Satelliten verwendet, um allerlei Medien zu verbreiten, GPS bereitzustellen oder um das Weltall zu erforschen. Aus diesem Grund ist die Umsetzung eines Demonstrationsversuchs in Form eines Satelliten ein geeigneter Ansporn, sich mehr mit dem Thema zu beschäftigen. Zusätzlich soll der Versuch aufgrund des einfachen physikalischen Effekts, dem Drallsatz, nachvollziehbar sein. Des Weiteren wird der Aufbau transportierbar sein und auf diese Weise für den Regelungstechnikstudenten im Saal leicht vorführbar sein.

1.2 Zielsetzung und Randbedingungen

Ziel ist es, einen nachvollziehbaren Demonstrationsversuch für Praktikumsstudenten im Fach Regelungstechnik aufzubauen. Dabei sollen die Geschehnisse in der Realität als auch graphisch ersichtlich sein. Es soll ein passendes regelungstechnisches System entworfen werden, welches die Ausrichtung regelt. Diese Arbeit beschäftigt sich zuerst mit den theoretischen Aspekten des Versuchsaufbaus, gefolgt von der empirischen Messung der Daten. Es handelt sich beim Versuchsaufbau um einen Satelliten, der sich frei um die eigene Achse drehen kann. Dabei befindet sich auf der Oberseite des Aufbaus ein bürstenloser Gleichstrommotor, der durch einen Drehzahlregler gesteuert wird, wobei die Drehzahl über Pulsweitenmodulation (PWM) eines Raspberry Pi Zero WH vorgegeben wird. Geregelt wird hierbei die Winkelposition des Satelliten, die mit einem magnetisch-induktiven Winkelaufnehmer gemessen wird. Gestellt wird das System mit einem bürstenlosen Gleichstrommotor, der eine Schwungmasse beschleunigt. Die Regelung wird in Simulink ausgelegt und mit dem Raspberry Pi Zero WH über einen WLAN-Router ausgeführt. Der Praktikumsstudent soll dazu in der Lage sein, das System zu stören und die Reaktion des geregelten Systems zu sehen. Um die Reaktion des Versuchs besser nachzuvollziehen, soll der Aufbau eine Möglichkeit besitzen, seine Orientierung anzuzeigen. Ferner soll hier mithilfe eines Laserpointers eine Fadenkreuzzielscheibe als Ziel angeleuchtet werden können, wobei man das Ziel verschieben sowie den gewünschten Sollwert verändern kann.

2 Entwicklung des Versuchs

Das Kapitel beschäftigt sich mit den Aspekten des Hardware-Aufbaus, der verwendeten Software und der Modellierung des Streckensystems. Diese dienen für die Umsetzung des Versuchs als Grundlage.

2.1 Genaue Beschreibung des Versuchsaufbaus

Der Satellit besteht aus mehreren für den regelungstechnischen Versuch entscheidenden Bauteilen. Zum Verständnis werden im Folgenden der grundlegende mechanische Aufbau, die Energieversorgung, die Ansteuerung der Aktorik und die Sensorik erklärt.

2.1.1 Der mechanische Aufbau

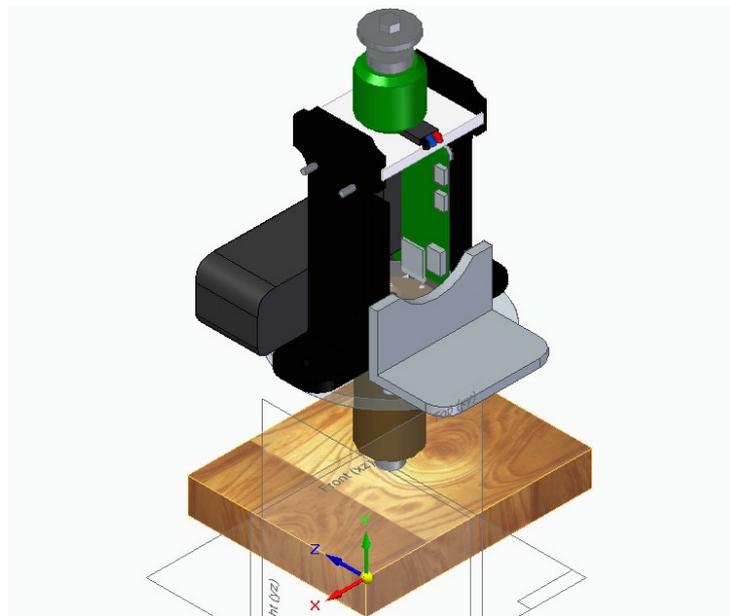


Abbildung 1: CAD-Darstellung der ersten Version des Aufbaus

Die erste Version des Aufbaus, wie graphisch in Abbildung 1 zu sehen ist, besteht aus einer Holzgrundplatte, in dem die Drehachse senkrecht eingeschraubt wurde. Dabei wird die obere Hälfte des Aufbaus frei auf dieser Drehachse wälzgelagert. Verwendet werden zwei Keramikwälzlager [1], die in das 3D-gedruckte Lagergehäuse eingepresst sind. Damit dieses nicht von der Drehachse rutscht, falls der Aufbau auf den Kopf gestellt wird, ist das Lagergehäuse mit einer Schraube auf der Oberseite der Drehachse gesichert. Die in die Drehachse gedrehte Schraube hält zusätzlich den für die Sensorik notwendigen Magneten. Der Magnet wurde in eine Bohrung auf dem Schraubenkopf hinein geklebt. Am Lagergehäuse ist die Plattform aus Acrylglas befestigt, auf dem der Satellitenkörper verschraubt ist. Dieser besteht aus zwei 3D-gedruckten Halterungen, wobei zwischen den beiden Halterungen

oberhalb die Motorplatte mit Gewindestangen festgeklemmt wurde. Zusammen bilden sie ein Gestell. An den Satelliten befestigt, ist ein großer Lithium-Polymer-Akku-Pack (LiPo-Akku-Pack), der zur Stromversorgung der Aktorik dient. Um dem großen Gewicht des Akkumulators entgegenzuwirken, wurde auf der anderen Seite ein Winkelverbinder festgeklebt. Im Inneren des Satellitenkörpers befindet sich die Elektronik des Aufbaus, wie der Raspberry Pi Zero WH, den Motordrehzahlregler und den Winkelaufnehmer. Auf der Oberseite befindet sich der bürstenlose Gleichstrommotor, der dem Drallrad seine Drehung gibt.

2.1.2 Lithium-Polymer-Akku-Pack



Abbildung 2: Der verwendeter LiPo-Akku-Pack von ACME-online.de

Lithium-Polymer-Akkumulatoren sind aufgrund ihrer großen Leistung, der Kapazität und der Fähigkeit wiederaufgeladen werden zu können, gut für den Modellbau geeignet. Das verwendete Akku-Pack (siehe Abbildung 2) hat eine Kapazität von 2200 *mAh* und besteht aus 3 Zellen und liefert somit eine Spannung von 11.1 *Volt*.

Im Demonstrationsversuch dient dieser zur Energieversorgung des bürstenlosen Gleichstrommotors. Dieser zieht unter Last bis zu 10 *A* an Strom [2]. Geladen wird das Akku-Pack mit einem speziellen Ladegerät. Dabei ist es wichtig, dass die einzelnen Zellen gleichmäßig geladen werden. Zusätzlich dürfen die einzelnen Zellen nicht überladen werden. Das Ladegerät schafft dies mit dem Balancer. Außerdem dürfen die einzelnen Zellen nicht zu stark entladen werden [3]. Deswegen wurde nach der Versuchsdurchführung darauf geachtet, das Akku-Pack regelmäßig zu laden und vom Drehzahlregler zu trennen, da dieser im angesteckten Zustand immer noch einen kleinen Strom zieht.

2.1.3 Der Electronic Speed Controller des bürstenlosen Gleichstrommotors

Der Electronic Speed Controller (ESC) dient der Drehzahlregelung des Motors. Dieser ermöglicht die Vorgabe der Motordrehzahl durch den Raspberry Pi Zero WH. Dabei wird der ESC durch PWM angesteuert. Das verwendete PWM-Signal hat eine Frequenz von 50 Hz, wobei der ESC Pulsweiten zwischen 1 ... 2ms verwendet. Dies entspricht einem Tastgrad von 5 ... 10%. Abbildung 3 zeigt den Schaltplan für die Verdrahtung des ESCs in der Mitte. Der LiPo-Akku befindet sich auf der linken Seite, der Raspberry Pi Zero WH ist oberhalb des ESC abgebildet und der Motor befindet sich auf der rechten Seite der Abbildung.

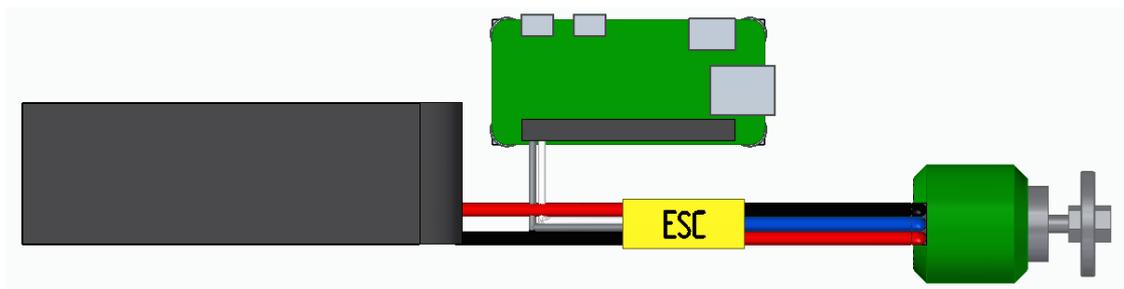


Abbildung 3: Schaltplan Flycolor Raptor F390 BLHeli 30A 2-4S OPTO Brushless ESC [4]

An einem bürstenlosen Gleichstrommotor befinden sich die Permanentmagneten am Rotor, wodurch dieser keine elektrische Versorgung benötigt. Am Stator wiederum befinden sich mehrere Wicklungsstränge für die einzelnen Phasen des Motors [5, Seite 110]. Der verwendete bürstenlose Gleichstrommotor BL-Motor ROXXY BL Outrunner 2827/34 besitzt drei Phasen, die abhängig von der in den Wicklungen induzierten Spannung geschaltet werden. Dadurch wird die Position des Rotors bestimmt. Diese Art der Wicklungsschaltung nennt man „sensorless“, weil keine Sensorik, wie etwa ein Hallsensor, die Position des Rotors erfasst [5, Seite 110].

Mithilfe der in Abbildung 4 gezeigten Software BLHeliSuite können die inneren Parameter des ESCs eingestellt werden. Über einen Arduino Uno lässt sich eine Verbindung zwischen BLHeli-Software und ESC aufbauen. Das Programm bietet viele Möglichkeiten, um das Verhalten des ESC zu bestimmen, in Rahmen dieser Arbeit sind vor allem der Hitzeschutz und die Drehrichtung des Motors zu beachten. Alle anderen Einstellungen wurden belassen. In der Bedienungsanleitung der Software wird unter dem Punkt „Thermal Protection“ geschrieben, dass abhängig davon, ob innerhalb des ESC-Mikrocontrollers eine Temperaturschwelle überschritten wird, die Leistung des Motors reduziert wird [6, Seite 7]. Aus diesem Grund wurde der Hitzeschutz ausgeschaltet, denn es hat sich herausgestellt, dass dieselben Motordrehzahlsprünge eine erheblich andere Systemreaktion am Satelliten hervorbringen. Dabei war die zurückgelegte Winkelstrecke nur halb so weit, wie es vorherige Versuche gezeigt hatten. Der andere Punkt in der Programmierung des ESCs ist die Motordrehrichtung, die unter dem Punkt „Motor Direction“ einstellbar ist. Dabei gibt es 3 Modi die

ausgewählt werden können. Zwei dieser Modi bestimmen die Drehrichtung des Motors. Der dritte mögliche Modus ist der sogenannte „Bidirectional“-Modus, der es ermöglicht den Motor mit demselben PWM-Signal in beide Richtungen drehen zu lassen und so den Drehzahlbereich des Motors zu verdoppeln.

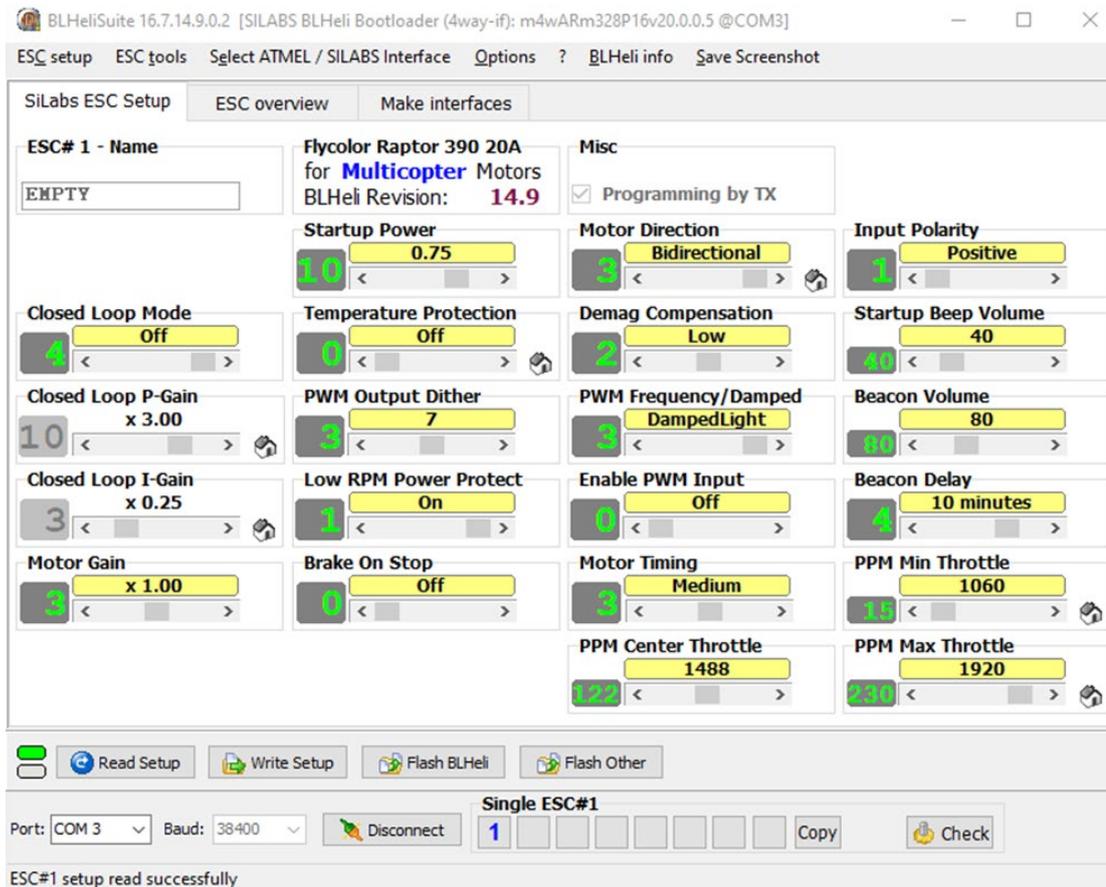


Abbildung 4: BLHeli-Software [6]

Der ESC muss jedes Mal beim Start eines Versuchs armiert werden. Dabei muss zuerst das niedrigste Signal für wenige Sekunden gehalten werden, bevor der Motor anfangen kann, sich zu drehen[6, Seite 8].

Um die maximale Drehzahlaufösung für das PWM-Signal zu erhalten, muss der ESC kalibriert werden. Der ESC lässt sich beim Armieren kalibrieren in dem zuerst, statt des minimalen PWM-Signals das gewünschte maximale Signal im Bereich zwischen 1 ... 2 ms gewählt wird. Für den Versuch wird der maximale Bereich für die beste Auflösung genutzt.

2.1.4 Der magnetische induktive Winkelaufnehmer als Lagesensor

Der Winkelaufnehmer befindet sich direkt über den dazugehörigen Permanentmagneten. Der Sensor ist über einen Inter-Integrated Circuit-Bus (I2C-Bus) mit dem Raspberry Pi Zero WH verbunden. Verwendet wird die in Abbildung 5 gezeigte Sensorik. Aufgrund des berührungslosen Messprinzips über Induktion kann die im System auftretende Reibung klein gehalten werden.

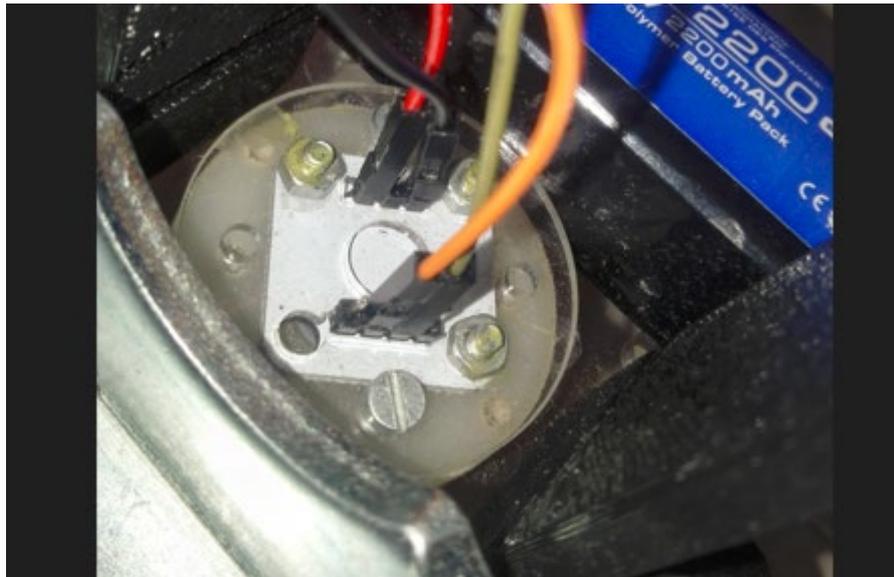


Abbildung 5: HALJIA AS5600 Magnetic Encoder Module [7] im Satellitenkörper

Der magnetische Winkelaufnehmer hat einer Auflösung von 4096 Schritten pro Umdrehung. Die Messdaten werden mit einer Abtastrate von 0,01 s abgetastet. Für diese Abtastrate wurde sich entschieden, um den flüssigen graphischen Verlauf auf Simulink als Scope zu ermöglichen. In Abbildung 6 wird das Simulink-Programm zur Kommunikation mit dem Sensor gezeigt.

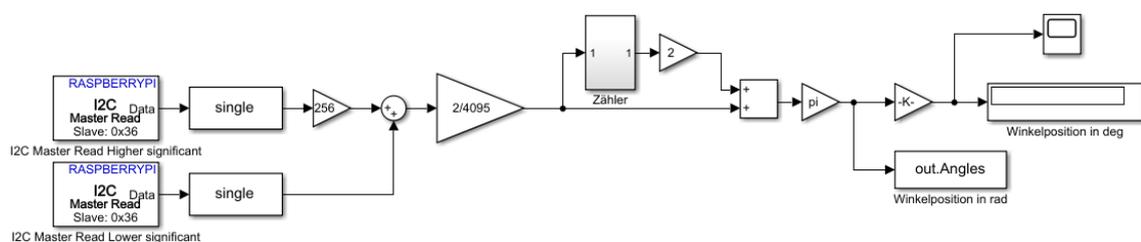


Abbildung 6: Simulink-Programm für Winkelaufnehmer

Um die Winkelposition nach einer ganzen Umdrehung zu messen, ist software-technisch ein Drehzähler implementiert worden. Dieser vergleicht die vorige mit der gerade gemessenen Winkelposition und überprüft, ob sich diese schlagartig um über eine halbe Umdrehung

verändert hat. Abhängig davon, ob der Sprung positiv oder negativ ist, zählt der Zähler-Block eins dazu oder zieht eine Umdrehung ab. Zur schnellen Kontrolle der gemessenen Winkelposition wurden zusätzlich ein Scope- und ein Display-Block eingefügt.

2.2 Die Motoransteuerung des Drallrads zur Veränderung der Lage

Die Motordrehzahl wird durch den Raspberry Pi Zero WH vorgegeben. Hierfür muss das entworfene Simulink-Programm zur Ansteuerung drahtlos auf den Mikrocontroller übertragen werden. Zusätzlich muss Simulink mit dem Raspberry Pi Zero WH kommunizieren um die gewollten Vorgaben sowie die gemessenen Daten zu bestimmen. Beispielsweise, um die Drehzahl vorzugeben.

2.2.1 Arbeiten mit dem Mikrocontroller Raspberry Pi Zero WH

Der Raspberry Zero Pi WH ist aufgrund seiner kompakten Größe, mit den Abmessungen von $65 \times 30 \text{ mm}$ und seiner Fähigkeit über Wireless LAN zu kommunizieren [8], gut für den Versuchsaufbau geeignet (siehe Abbildung 7).

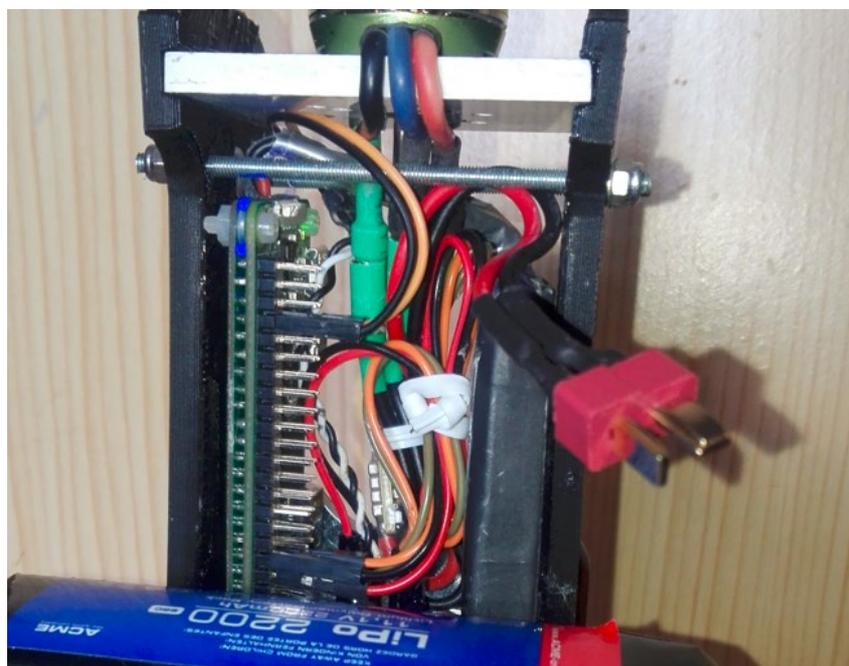


Abbildung 7: Elektronik im Inneren des Satellitenkörpers

Zusätzlich kann der Mikrocontroller mit einem Akkumodul [9] kombiniert werden. Dies ermöglicht die schnelle Umsetzung eines ersten Versuchsaufbaus. Die GPIO-Stiftleiste des Mikrocontrollers lässt eine schnelle Verkabelung mit der Sensorik und der Aktorik zu.

MATLAB und Simulink sind mit verschiedenen Mikrocontrollern kompatibel. Dabei kann der Raspberry über einen WLAN-Router kommunizieren. Simulink bietet dazu die notwendige Kompatibilität und eine Bibliothek an. Dazu müssen in Simulink unter *Hardware Settings* als *Target hardware resources* die Geräte-Adresse, der Username und das Passwort des Raspberry Pi sowie das passende Hardware-Board ausgewählt werden.

Der Raspberry Pi Zero WH dient der Aufnahme und Auswertung der Messsignale des Winkelnehmers. Dabei wird dieser über einen 5V-Pin mit Strom versorgt und kommuniziert mit dem I2C-Bus über den Serial-Data (SDA)- und Serial-Clock (SCL)-Pin des Mikrocontrollers.

Zur Vorgabe der Drehzahl wird der PWM-Block von Simulink verwendet. In diesem kann die gewollte Frequenz und der Ausgangs-Pin gewählt werden. Dieser Block akzeptiert als Eingang einen Wert zwischen 0 und 1, wobei 0 für 0% und 1 für 100% Tastgrad steht.

Auf diese Weise können in Kombination mit dem ESC etwa 200 mögliche Motordrehzahlen vorgegeben werden.

Der Raspberry Pi Zero ist ein kleiner Computer, dessen Betriebssystem idealerweise über MATLAB mit *MathWorks Raspbian image* installiert werden sollte. D.h. es sollte beim Hardware Setup besser auf *Setup hardware with Mathworks Raspbian image* gegangen werden, wie in Abbildung 8 gezeigt wird.

Hardware Setup

Select Linux Operating System

MATLAB and Simulink support package for Raspberry Pi require a customized version of Raspbian Linux Operating System (OS) running on the hardware.

I want to:

- Setup hardware with MathWorks Raspbian image
- Customize the existing Raspbian OS running on my hardware

Abbildung 8: Hardware-Setup über den Add-On Manager in MATLAB

Es hat sich gezeigt, dass es ansonsten zu verschiedenen Verbindungsproblemen kommt.

2.2.2 Erstellte Simulink-Programme

Um die Motordrehzahl beim jeweiligen PWM-Tastgrad zu ermitteln, wurde mit Simulink das Programm *MotorRun_PWM.slx* zur Auswertung erstellt. Die Motordrehzahl kann dann durch einen Lasertachometer gemessen und zugeordnet werden.

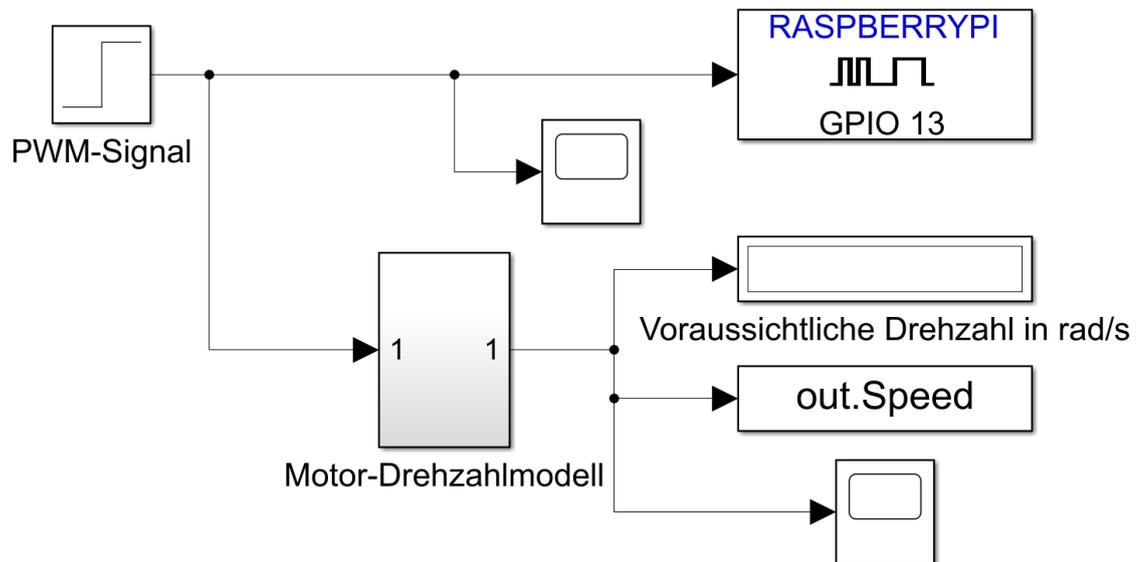


Abbildung 9: Programm für Motordrehzahlmessung *MotorRun_PWM.slx*

Aus den so ermittelten Messdaten kann ein Motor-Drehzahlmodell gebildet werden. Dies wird zum Beispiel durch Bilden einer Geraden über lineare Regression erreicht. Mit den Simulink-Blöcken in Abbildung 9 für das Motor-Drehzahlmodell und dem Display kann die voraussichtliche Motordrehzahl angezeigt werden.

Das Programm *MotorRun_Reg_Sim.slx* in Abbildung 11 dient zur Umsetzung des Reglers und beschreibt zwei parallel laufende Regelkreise. Einen Regelkreis, der die reale Strecke regelt und einen, der die simulierte Strecke über das Regelstreckenmodell regelt. Beide Ausgangssignale der geschlossenen Regelkreise werden über die Scopes angezeigt und ermöglichen so den Vergleich zwischen Simulation und realer Messung.

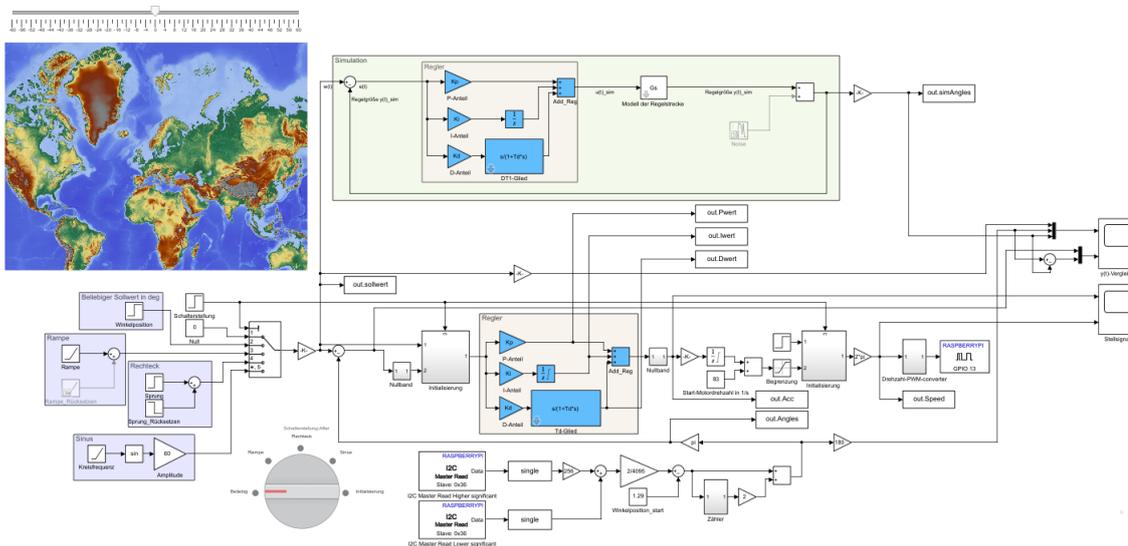


Abbildung 11: Regelungsprogramm *MotorRun_Reg_Sim.slx*

Wie im vorigen Programm gibt dieses die Möglichkeit verschiedene Sollwertsignale zu testen, das Ausgangsverhalten zu messen und wichtige Daten über *ToWorkspace*-Blöcke in den *Workspace* zu übertragen. So werden der vorgegebene Sollwert, die einzelnen Größen des realen Reglers, das Stellsignal des Reglers, die resultierende Motordrehzahl sowie das reale und simulierte Ausgangssignal in den *Workspace* übertragen. Wie beim vorigen Programm, muss dieses zuerst den ESC armieren. Auf dies wird in Kapitel 5.2.2 genauer eingegangen. Durch das MATLAB-Programm *reg_Messdaten_zusammenf.m* (siehe Anhang) können dann alle Daten auf geeignete Weise zusammengefasst werden. Die entstandenen Datensätze *mat_reg* und *simAngles* können dann im MATLAB-Programm *mat_reg_plot.m* graphisch vereint dargestellt werden.

2.3 Modellierung der Lagedynamik des Satelliten

In diesem Kapitel wird sich mit der Streckenmodellierung des Satellitenverhaltens befasst. Dabei werden physikalische als auch experimentelle Modellierungsansätze vorgestellt.

2.3.1 Physikalische Modellierung der Regelstrecke mit dem Drallsatz

In der physikalischen Modellierung geht es darum, das Streckenmodell über theoretische physikalische Gleichungen zu ermitteln. In diesem Fall wird der Drallsatz zur physikalischen Modellierung herangezogen. Der Drallsatz besagt, dass das äußere Drehmoment gleich der zeitlichen Ableitung des Drehimpulses ist [10, Seite 160 bis 161]. Beim Demonstrationsversuch ist dies zwischen Drallrad und dem sich frei um die eigene Achse drehbaren Satelliten der Fall. Hierbei bewirkt das Drallrad durch die Beschleunigung des Motors ein äußeres Drehmoment (ab hier als Motormoment), dabei ändert sich der Drehimpuls des Satellitenkörpers. D.h. das Motormoment ist gleich die zeitliche Ableitung des Satelliten-Drehimpulses:

$$M_{Mot}(t) = \dot{L}_{Sat}(t)$$

Wobei der Drehimpuls sich durch die Multiplikation von dem Massenträgheitsmoment des Satellitenkörpers und seiner Winkelgeschwindigkeit beschreiben lässt [10, Seite 202]:

$$L_{Sat}(t) = J_{Sat} * \omega_{Sat}(t)$$

Der zeitlich abgeleitete Drehimpuls des Satellitenkörpers wird nun mit dem Motormoment gleichgesetzt und als Gleichung resultiert daraus, dass das Motormoment gleich der Multiplikation aus Massenträgheitsmoment und abgeleiteter Winkelgeschwindigkeit ist:

$$M_{Mot}(t) = J_{Sat} * \dot{\omega}_{Sat}(t)$$

Wobei die Ableitung der Winkelgeschwindigkeit gleich der Winkelbeschleunigung ist und sich die rechte Seite der Gleichung durch das Drehmoment des Satelliten beschreiben lässt [10, Seite 173]:

$$J_{Sat} * \dot{\omega}_{Sat}(t) = J_{Sat} * \alpha_{Sat} = M_{Sat}(t)$$

Des Weiteren ist der Satellit nicht reibungsfrei gelagert, weswegen ein Reibungsterm dazu addiert wird. Insgesamt resultiert daraus, dass das Motormoment gleich der Summe des Satellitenmoments und dem Reibungsmoment ist, zu Gleichung (1):

$$M_{Mot}(t) = M_{Sat}(t) + M_{Reib}(t)$$

Das Reibungsmoment wird als eine Funktion abhängig von der Drehgeschwindigkeit gewählt und wirkt sich dämpfend auf das System aus. Somit kann man das resultierende Reibmoment annähernd als eine Geradengleichung mit dem Reibkoeffizienten k abhängig von der Satellitenwinkelgeschwindigkeit beschreiben [11, Seite 148]:

$$M_{Reib}(t) = k * \omega_{Sat}(t)$$

Das Motormoment ergibt sich aus dem Produkt aus Massenträgheitsmoment von Motor und Drallrad und der Winkelbeschleunigung dieser. Da beim Motor die Drehzahl durch den ESC bestimmt wird, wird die Winkelbeschleunigung als zeitliche Ableitung der Motorwinkelgeschwindigkeit beschrieben, Dargestellt durch Gleichung (2):

$$M_{Mot}(t) = J_{Mot} * \dot{\omega}_{Mot}(t)$$

Möchte man nun alles zusammen als Differentialgleichung ausschreiben, muss man, weil am Ende die Lage des Satelliten gesucht ist, die Winkelgeschwindigkeit als zeitliche Ableitung der Winkelposition des Satelliten schreiben:

$$\omega_{Sat}(t) = \dot{\varphi}_{Sat}(t)$$

Das Reibmoment wird nun über Gleichung (3) beschrieben:

$$M_{Reib}(t) = k * \dot{\varphi}_{Sat}(t)$$

Und das Satellitenmoment beschreibt sich dann als Produkt aus dem Massenträgheitsmoment des Satelliten und der zweifachen Ableitung der Winkellage durch Gleichung (4):

$$M_{Sat}(t) = J_{Sat} * \ddot{\varphi}_{Sat}(t)$$

Setzt man nun in Gleichung (1), die Drehmomente aus den Gleichungen (2-4) so resultiert daraus eine Differentialgleichung mit der Motorwinkelgeschwindigkeit als Eingangssignal und der Winkellage als Ausgangssignal:

$$J_{Mot} * \dot{\omega}_{Mot}(t) = J_{Sat} * \ddot{\varphi}_{Sat}(t) + k * \dot{\varphi}_{Sat}(t)$$

Die Differentialgleichung wird nun Laplace-transformiert, wobei sämtliche Anfangsbedingung gleich null gesetzt werden.

Dabei wird für die Unterscheidung von Winkelgeschwindigkeit und Winkellage die Laplace-Transformationen so gewählt, sodass: $\mathcal{L}\{\omega(t)\} = U'(s)$ und $\mathcal{L}\{\varphi(t)\} = Y(s)$. Entsprechend ist $U'(s)$ der Systemeingang und $Y(s)$ ist der Systemausgang.

$$J_{Mot} * U'(s) * s = J_{Sat} * Y(s) * s^2 + k * Y(s) * s$$

Formt man nun um zu:

$$G(s) = \frac{Y(s)}{U'(s)}$$

Entsteht die Übertragungsfunktion erster Ordnung:

$$G(s) = \frac{J_{Mot} * s}{J_{Sat} * s^2 + k * s} = \frac{J_{Mot}}{J_{Sat} * s + k} = \frac{\frac{J_{Mot}}{k}}{\frac{J_{Sat}}{k} * s + 1}$$

Da der Regler als Kaskadenregler gestaltet ist, muss die Verzögerung des Motordrehzahlstellglieds modelliert werden. Dabei wird vereinfachend davon ausgegangen, dass der

drehzahlgeregelte Motor sich durch ein PT1-System beschreiben lässt. Das Stellglied wird nun mit der durch die Zeitkonstante T_{Mot} verzögerte Übertragungsfunktion beschrieben:

$$G_{Mot}(s) = \frac{1}{T_{Mot} * s + 1}$$

Um jetzt ein vorläufiges Streckenmodell zu erhalten, müssen die beiden Übertragungsfunktionen miteinander multipliziert werden zu Gleichung (5):

$$G_{sp}(s) = G(s) * G_{Mot}(s) = \frac{\frac{J_{Mot}}{k}}{\left(1 + \frac{J_{sat}}{k} * s\right) * (1 + T_{Mot} * s)}$$

Man erhält daraus eine Streckenübertragungsfunktion, die eine Nennerordnung von 2 und eine Zählerordnung von 0 hat. Interessant daran ist, dass man durch Koeffizientenvergleich aus der experimentellen Modellbildung die einzelnen Werte identifizieren kann. Dies ermöglicht eine Zuordnung der Koeffizienten, der durch experimentelle Identifikation bestimmten Übertragungsfunktion.

2.3.2 Experimentelle Identifikation der Regelstrecke mit dem Regelungstechnikprogramm pzMove

Für die experimentelle Streckenmodellierung wird das Tool pzMove angewendet. Es ermöglicht die Identifikation beliebiger linearer Systeme und es basiert auf das Verfahren nach Prof. Zentgraf. Dabei wird aus einem einzigen Datensatz aus fehlerhaft gemessenen Eingangssignal $u(t)$ und Ausgangssignal $y(t)$ das komplette, allgemeine Systemverhalten des Modells bestimmt[12, Seite 2]. Für pzMove ist es wichtig, dass der Datensatz aus gleich großen Zeitschritten besteht.

Abbildung 12 zeigt einen Ausschnitt von pzMove in der Registerkarte *Identification*. Dort sind verschiedene Funktionalitäten abgebildet. Im Bild oben links kann das Zeitintervall für die Schätzung des Modells eingestellt werden. Dabei ist es wichtig nicht den gesamten Messdatensatz für die Identifikation zu verwenden, da man mit den Messdaten außerhalb des Zeitintervalls das Modell verifizieren möchte [12, Seite 3]. Rechts daneben gibt es noch die Möglichkeit den Datensatz auszudünnen, da dieser für die Identifikation zu groß sein kann[12, Seite 5]. Gegebenenfalls kann man den Datensatz von starken Messausschlägen bereinigen und so an ein geeigneteres Modell erlangen. Des Weiteren bietet pzMove die Möglichkeit, Totzeiten im Modell einzuschätzen.

Select time limits (start/end time)

ts = 0 [s] || te = 10 [s] Use every 1 . data Set lower and upper bound of deadline TD On 0 [s] <= TD <= 1 [s]

Set order

Numerator: 1 Denominator: 2

Set coefficients to zero

	s ²	s ¹	s ⁰
GS.num	x	x	
GS.den	x	x	x

Abbildung 12: Auswahl der Zähler- und Nenner-Ordnung in pzMove

In der unteren Hälfte von Abbildung 12, kann man Zähler- und Nenner-Ordnung der Modellübertragungsfunktion bestimmen. Abhängig davon, wie diese gewählt werden, ist das identifizierte Modell schlechter oder besser. In den Feldern, in denen die Buchstaben x stehen, hat man die Möglichkeit, einzelne Koeffizienten der Übertragungsfunktion auf null zu setzen, um beispielsweise integrales oder differentiales Verhalten zu identifizieren.[12, Seite 10]

Abbildung 13 zeigt, dass pzMove eine Filterfunktion für das Aufzeigen explizit stabiler oder instabiler Übertragungsfunktionen bietet. Des Weiteren bietet pzMove, die Möglichkeit die Anfangsbedingungen einzuschätzen.

Nonlinear optimization methods

On additional optimization iJtype2

minimum squares directly corresponding to LTI

Filter solutions iStab

stable or unstable

Assumed initial conditions iy0

zero

Abbildung 13: Auswahl der Filteroptionen in pzMove

Dies bedeutet, pzMove kann die Dynamik des Systems kurz vor dem Messstartzeitpunkt einschätzen und berücksichtigen. Dies ist praktisch, wenn ein Prozess nicht einfach so aus der Ruhelage gestartet werden kann[12, Seite 2].

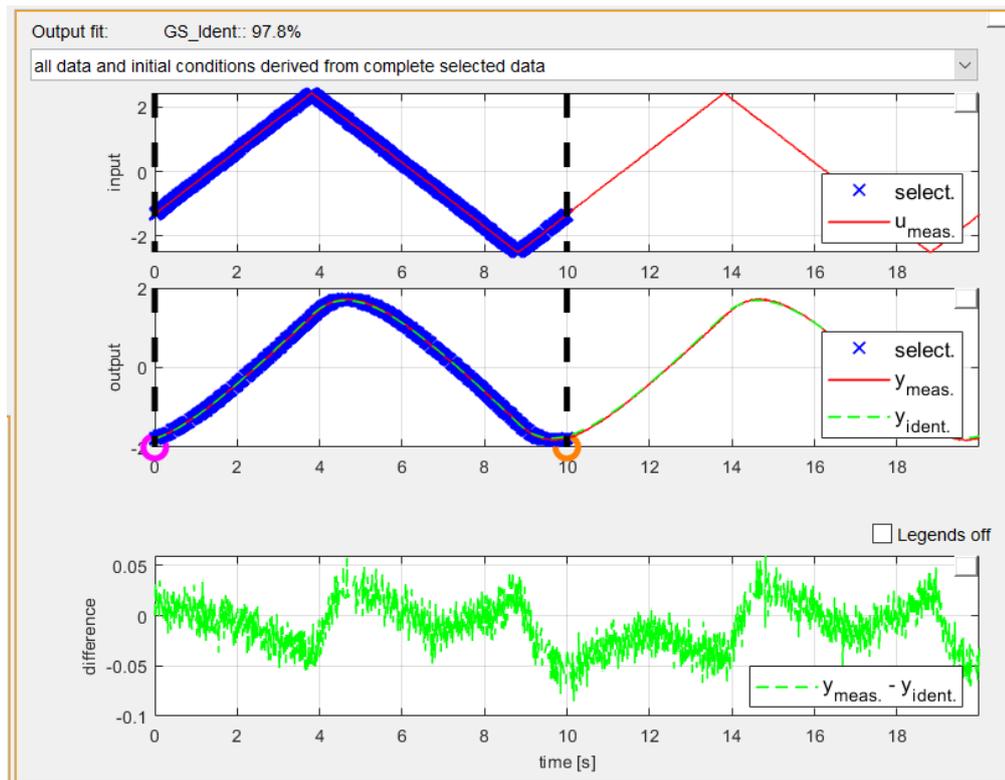


Abbildung 14: Darstellung des Eingangs- und Ausgangsgraphen in pzMove

Die Abbildung 14 zeigt, wie das ausgewählte Zeitintervall in Blau für die Identifikation des Streckenmodells aussieht. Das Verhalten des identifizierten Systems wird im mittleren Plot durch den grün gestrichelten Graphen dargestellt. Die Differenz zwischen gemessenen und simulierten Graphen zeigt sich im dritten Plot. Entscheidend ist, dass die Identifikation des Modells umso besser wird, je unkorrelierter das Verhalten des unteren Graphen zur Messung ist [12, Seite 10]. Zur Bewertung der Identifikation zeigt pzMove einen Fit-Wert an. In der Abbildung wird neben *Output fit* eine Zahl, die bis zu 100% ansteigen kann, gezeigt. Ein guter Fit-Wert bedeutet aber nicht, dass das Modell numerisch exakt bestimmt wurde [12, Seite 8]. Aus diesem Grund sollte das Verhalten der realen Strecke mit einem andersartigen Eingangssignal mit dem identifizierten Modell verglichen werden. Erst dann kann man das Modell für die Reglerauslegung verwenden.

Es soll noch kurz auf die Abbildung 15 eingegangen werden. Die Software pzMove bietet viele Möglichkeiten für das identifizierte Streckenmodell eine Regelung zu entwerfen.

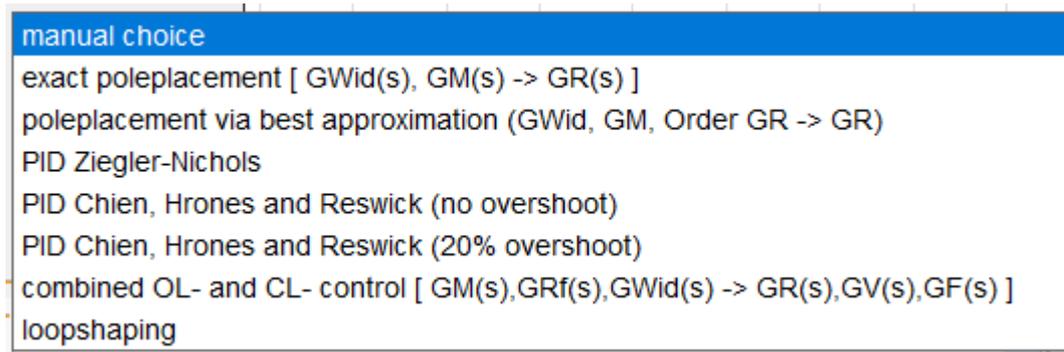


Abbildung 15: Auswahl an Reglerentwurfsverfahren in pzMove

Beispielsweise können in pzMove automatisiert die verschiedenen Einstellregeln ausgewählt werden. Die Software simuliert dann augenblicklich das Verhalten des geschlossenen Regelkreises und zeigt Bode- und Nyquist-Diagramme sowie Sprung- und Impulsantworten des Systems an.

3 Das Stellglied

3.1 Das Motordrehzahlverhalten durch den Electronic Speed Controller

Die Motordrehzahl wird mit Hilfe eines Handdrehzahlmessers *VOLTCRAFT DT-IL* gemessen. Dieses Lasertachometer gibt die Drehzahl des Motors in Umdrehungen pro Minute (rpm) an. Dafür muss der bürstenlose Gleichstrommotor präpariert werden.

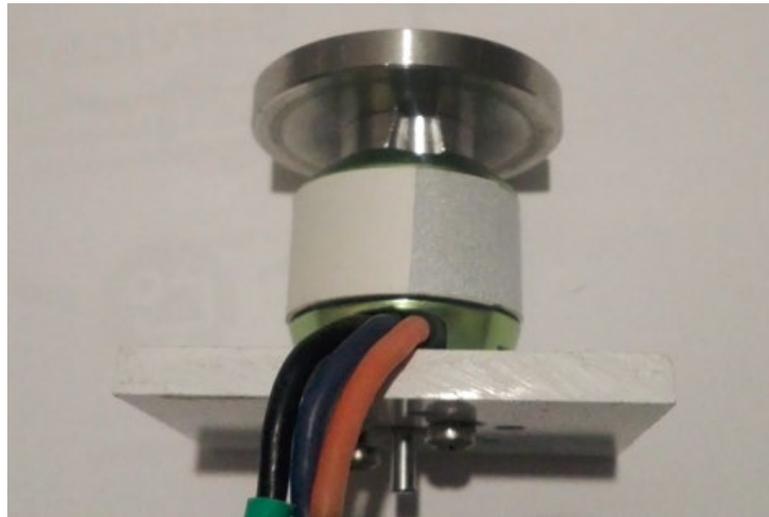


Abbildung 16: Präparierter Motor

Der Motor wurde, wie in Abbildung 16 zu sehen, mit einem Reflexionsstreifen versehen. Damit der Hell-/Dunkelkontrast gut genug ist, wurde der reflektierende Motorkörper mit mattem Papier abgedeckt.

Bei den Messungen wurde schrittweise der Tastgrad erhöht. Zuerst wurden insgesamt 50 Messungen im gleichmäßigen Abstand zueinander ausgeführt. Dabei war der Startwert ein Tastgrad $D = 5.1\%$ und in 0.1% -Schritten wurden Messwerte bis zum Tastgrad $D = 10\%$ bestimmt. Da ein erster Plot schon mal anmuten ließ, dass der Graph sich bis auf seine Enden durch eine Gerade beschreiben lässt, wurden genauere Messungen an den Extremen des Drehzahlbereichs vorgenommen. Gemessen wurde dabei im Bereich zwischen 5% und 5.01% sowie 9.49% und 10% in möglichst kleinen Tastgrad-Schritten. Die Abbildung 17, zeigt wie sich die Drehzahl beim gegebenen Tastgrad verhält.

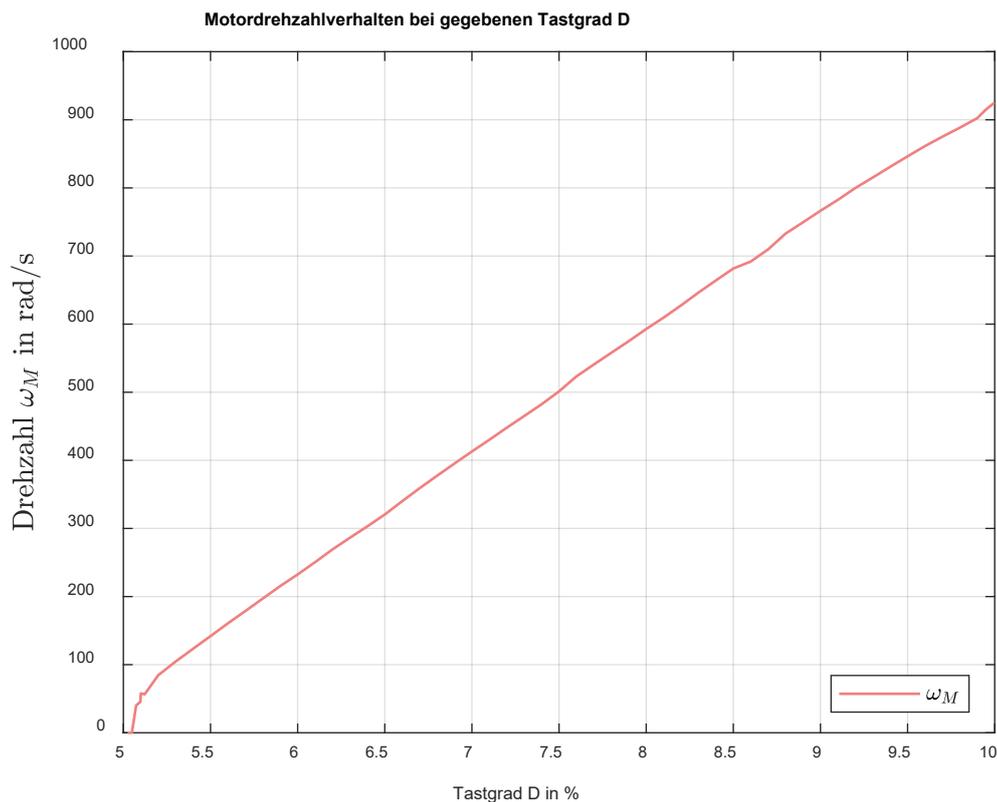


Abbildung 17: Durch Lasertachometer gemessene Messdaten graphisch dargestellt

Für die Zuordnung der gewollten Drehzahl zum benötigten Tastgrad, wurden mit Hilfe der Messwerte für $D = 5.2 \dots 9.4\%$ durch lineare Regression eine Geradengleichung ermittelt und mit diesen verglichen. Um die Einheitlichkeit des Eingangs- und Ausgangssignals zu ermöglichen, werden die jeweiligen Motordrehzahlen statt in rpm in rad/s umgerechnet. Entsprechend wird das Ausgangssignal in rad gemessen.

Der Plot in Abbildung 18 zeigt eine ausreichend lineare Motordrehzahlzuordnung im Bereich $126 \dots 892 rad/s$. Dort weicht das Geradenmodell ω_{LR} maximal um 2% vom gemessenen Drehzahlwert ω_M ab. Um das Geradenmodell für die weitere Bearbeitung nutzbar zu machen, muss man die Umkehrfunktion bilden. Dadurch erhält man eine Geradengleichung, die bei Vorgabe einer Drehzahl den entsprechenden Tastgrad für die PWM ausgibt. Die ermittelte Geradengleichung kann in Form eines Simulink-Blocks für die nachfolgenden Versuche verwendet werden.

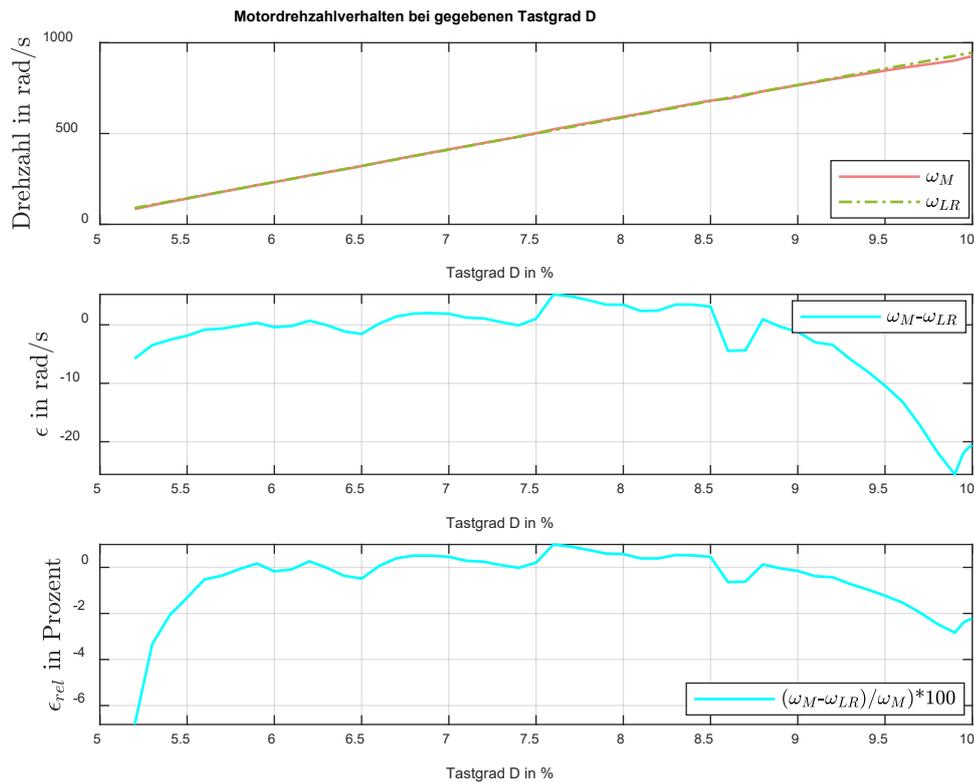


Abbildung 18: Vergleich des Geradenmodelles mit der Messung

3.2 Auswahl des Stellsignals

Der ESC sorgt für die gewollte Motordrehzahl und kann den Motor in zwei Richtungen drehen lassen. Dies könnte man nutzen, um als Eingangssignal des Streckensystems die Motordrehzahl zu verwenden. Problematischerweise kann der ESC die Motordrehzahlen nicht beliebig klein werden lassen. Zusätzlich verhalten sich besonders kleine und große Motordrehzahlen nicht mehr linear zum Tastgrad. Was die Nutzung der Motordrehzahl als Stellsignal erschwert und die Auslegung eines linearen Reglers herausfordernd macht. Ein weiterer Grund sich gegen die Motordrehzahl als Stellsignal zu entscheiden kommt durch den Drallsatz. Dieser besagt, dass das System seinen Drehimpuls erst ändert, wenn ein von außen kommendes Drehmoment auf dieses einwirkt. Das bedeutet in diesem Fall, dass nur eine Beschleunigung des Drallrads oder die im System vorhandene Reibung den Drehimpuls bei dem Satelliten verändert.

Wie unten im Kapitel 4.1 genauer erläutert wird, wirkt auf dem System ein nicht linearer Effekt, der durch die Haftreibung entsteht. Diese lässt sich am besten durch die notwendige Beschleunigung des Drallrads zur Überwindung der Haftreibung einschätzen. Man kann den Einfluss der Haftreibung reduzieren, indem man das Massenträgheitsmoment des Drallrads und somit das äußere Drehmoment erhöht. Zusätzlich ist es möglich durch Reduzierung des Massenträgheitsmoments des Satelliten, das Größenverhältnis zwischen dem Drehmoment

des Drallrads und des Reibmoments vorteilhaft zu verändern. Des Weiteren kann die Gewichtsminderung des Satelliten die auf das System wirkende Reibkraft verringern. Es wurde sich dazu entschieden ein größeres Drallrad zu verwenden, da dies zum einen als ausreichend bewertet wird und zum anderen bietet der größere Radius des Drallrads eine gute Möglichkeit die geregelte Strecke am Systemeingang zu stören.

Ein weiterer Grund, statt der Motordrehzahl die Motorbeschleunigung als Stellsignal zu verwenden, ergibt sich dadurch, den Motor nur in eine Richtung drehen zu müssen. Dies sorgt bei der Drehzahl-Auflösung des ESCs für eine knappe Verdopplung auf ungefähr 200 Drehschritte, da der Motor nicht mehr bidirektional betrieben werden muss.

Eine Änderung des Stellsignals von Motordrehzahl zu Motorbeschleunigung sorgt dafür, dass dieses abgeleitet wird. Dies entspricht im Laplace-Bereich der Multiplikation mit s . Um das Streckenmodell aus Gleichung (5) anzupassen, muss dieses mit s geteilt werden. Daraus resultiert eine neue Übertragungsfunktion als Gleichung (6):

$$G_{spn}(s) = \frac{\frac{J_{Mot}}{k}}{\left(1 + \frac{J_{Sat}}{k} * s\right) * (1 + T_{Mot} * s) * s}$$

Des Weiteren wird die Änderung des Stellsignals, in den Simulink-Programmen, durch einfügen eines *Integrator*-Blocks erreicht. Dieser integriert das kommandierte Beschleunigungssignal und ermittelt auf diese Weise das kommandierte Motordrehzahlsignal

4 Messung des Ein- und Ausgangsverhaltens

Das Kapitel beschäftigt sich zum einen um den nichtlinearen Einfluss, den das Streckensystem durch die Haftreibung erfährt, zum anderen mit der Bestimmung des Streckenmodells.

4.1 Die nichtlineare Natur der Strecke

Die Abbildung 19 zeigt die Auswirkung der Haftreibung. Auf dem Plot sind insgesamt 4 Graphen abgebildet, wobei jeweils zwei von diesen ähnlich zueinander sind. Der Plot zeigt die Systemantworten von Eingangs-Rampen verschiedener Steigung, jeweils in die positive und in die negative Richtung. Die jeweiligen Rampen beginnen jeweils ab dem Zeitpunkt $t = 0$ s zu steigen. Die vorgezeigten Graphen wurden durch Teilung mit ihrer Rampensteigung normiert. Da eines der Ziele der Arbeit die Gestaltung eines linearen Reglers ist, ist eine Abweichung der Graphen zueinander ein Hinweis auf eine Nichtlinearität im System. Während die Steigungsrichtung in die positive als auch in die negative Richtung keine nennenswerte Divergenz erzeugen, scheint die Größe der Steigung einen großen Einfluss zu haben.

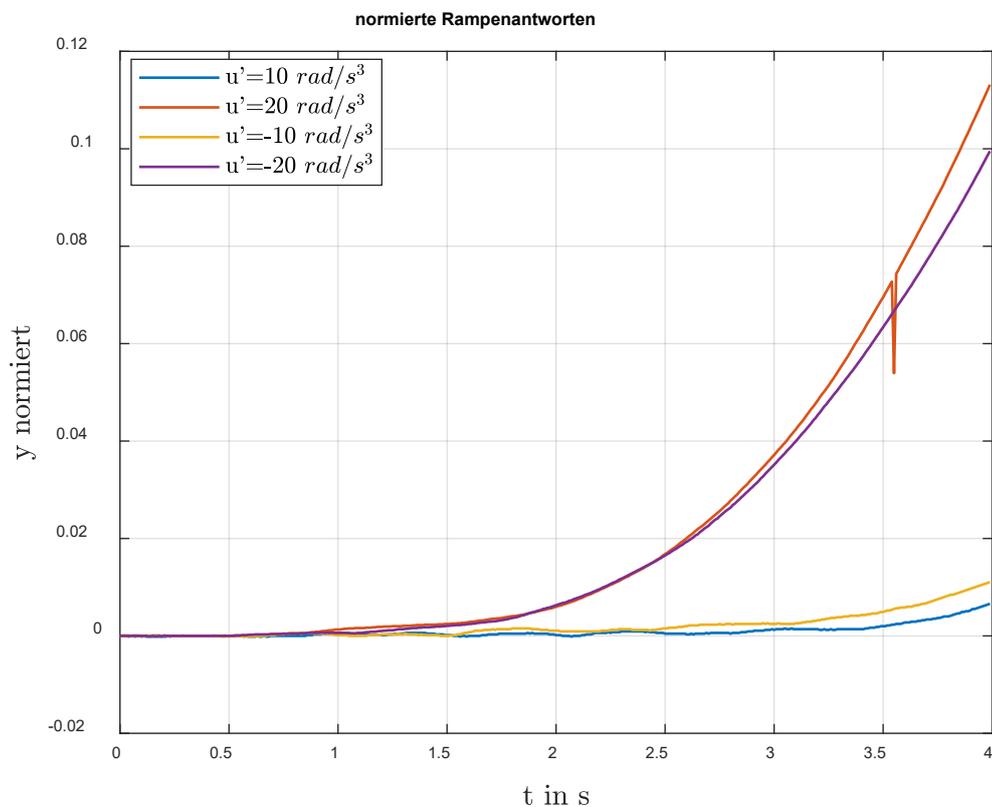


Abbildung 19: Normierte Rampenantworten für zwei verschiedene Rampensteigungen positiv und negativ

Schaut man sich die normierten Graphen, die eine Rampensteigung u' von betragsmäßig 20 rad/s^3 haben, fängt die Strecke erst ab $t = 1$ Sekunde an zu reagieren. Dies entspricht einem Eingangssignal von $u(1 \text{ Sekunde}) = 20 \text{ rad/s}^2$. Wiederum erkennt man, dass das System bei der Eingangs-Rampe mit einer Steigung von 10 rad/s^3 erst einige Zeit später in Bewegung gerät. Man kann also anhand des Plots den Einfluss durch die Haftreibung aufzeigen.

Man kann davon ausgehen, dass die Reibung sich bei einer niedrigen Dynamik der Strecke nichtlinear auswirkt. Dies soll anhand der folgenden Sinus-Antworten gezeigt werden.

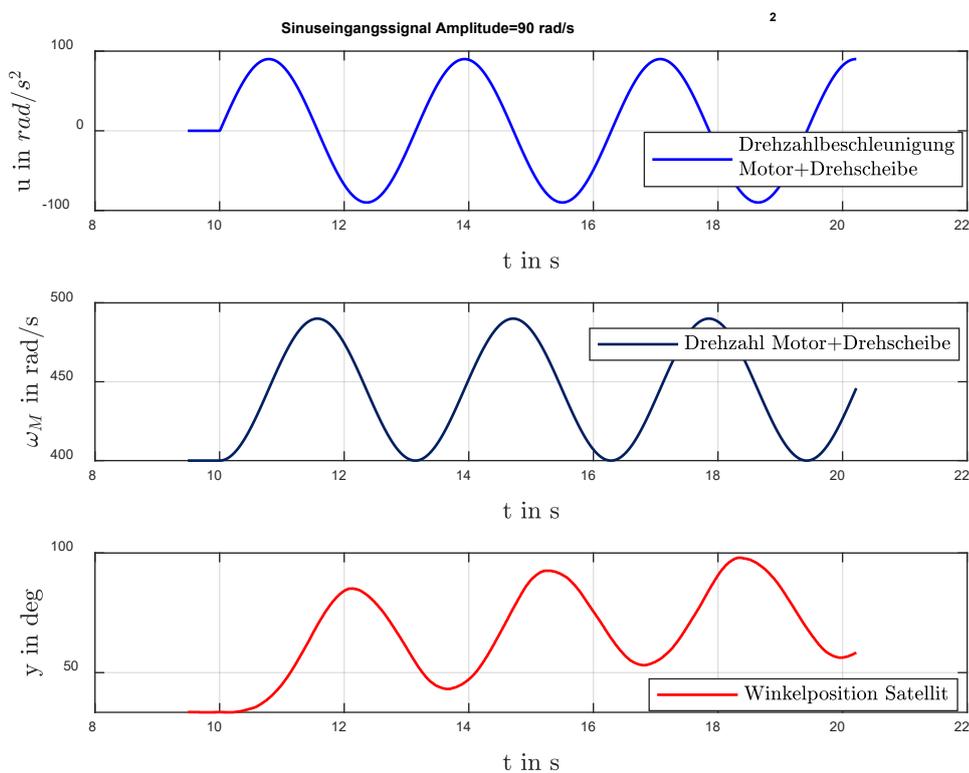


Abbildung 20: Langsam aufsteigende Winkelposition des Satelliten

Das System in Abbildung 20 erhöht langsam seine Winkelposition bei einem Sinuseingang mit einer Kreisfrequenz von 2 1/s und einer Amplitude von 90 rad/s^2 .

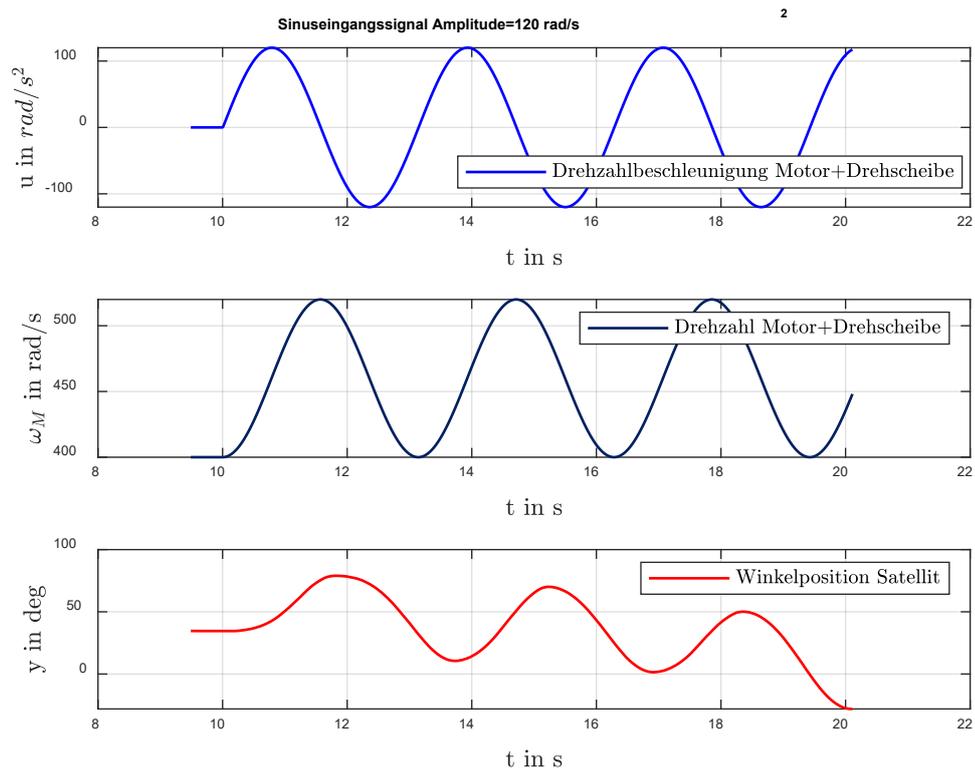


Abbildung 21: Langsam fallende Winkelposition des Satelliten

In Abbildung 21 wiederum sieht man eine langsame Reduzierung der Winkellage und dies bei gleichbleibender Kreisfrequenz, aber dafür auf 120 rad/s^2 erhöhten Amplitude. Es wird vermutet, dass Eingangssignale, die ein Systemverhalten erzeugen, in dem der Satellit seine Drehgeschwindigkeit sehr langsam werden lässt, stark unterschiedliche Systemantworten verursachen können.

4.2 Bestimmung der Reibung

Da die Haftreibung das Streckensystem ungünstig beeinflusst, wird versucht durch rampenförmige Eingangssignale der Haftreibung eine Größe zuzuordnen. Danach wird untersucht, ab welcher Rampensteigung sich das System ausreichend linear verhält.

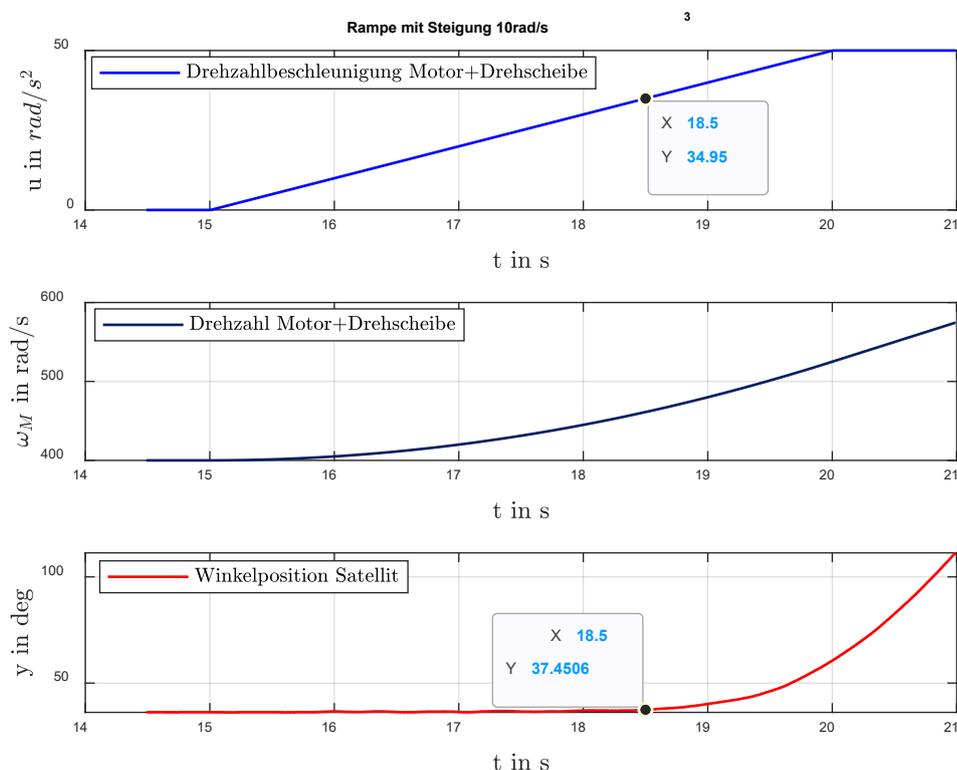


Abbildung 22: Versuch Haftreibung zu bestimmen

Abbildung 22 zeigt, dass das System sich erst bei einem Stellsignal von ca. 35 rad/s^2 bewegt. Dies beschreibt die notwendige Beschleunigung, damit die Haftreibung überwunden wird.

Des Weiteren ist die Strecke aufgrund ihrer leichten Neigung und der unsymmetrischen Massenverteilung zusätzlich nicht linear beeinflusst. Dies wird in der nächsten Abbildung 23 deutlich, dort verhält sich die Rampenantwort mit einer Steigung von 10 rad/s^3 nicht so schön wie in Abbildung 22. Der Grund liegt an der erneuten Messung von einer anderen Startlage des Satelliten. Nun kann man nicht einfach davon ausgehen, dass die Reibung, nachdem sich der Aufbau in Bewegung gesetzt hat, komplett verschwindet und sich durch einen linearen Anteil beschreiben lässt. Aus diesem Grund werden mehrere Versuche für die Aufnahme von Systemantworten mit größer werdenden Eingangsrampen gemessen. Damit können, wie in Abbildung 23 zu sehen ist, die normierten Graphen miteinander verglichen werden.

Die normierten Antworten sollen dabei behilflich sein, einen geeigneten Messdatensatz ausfindig zu machen. Die daraus resultierenden Daten dienen der Identifikation der Strecke.

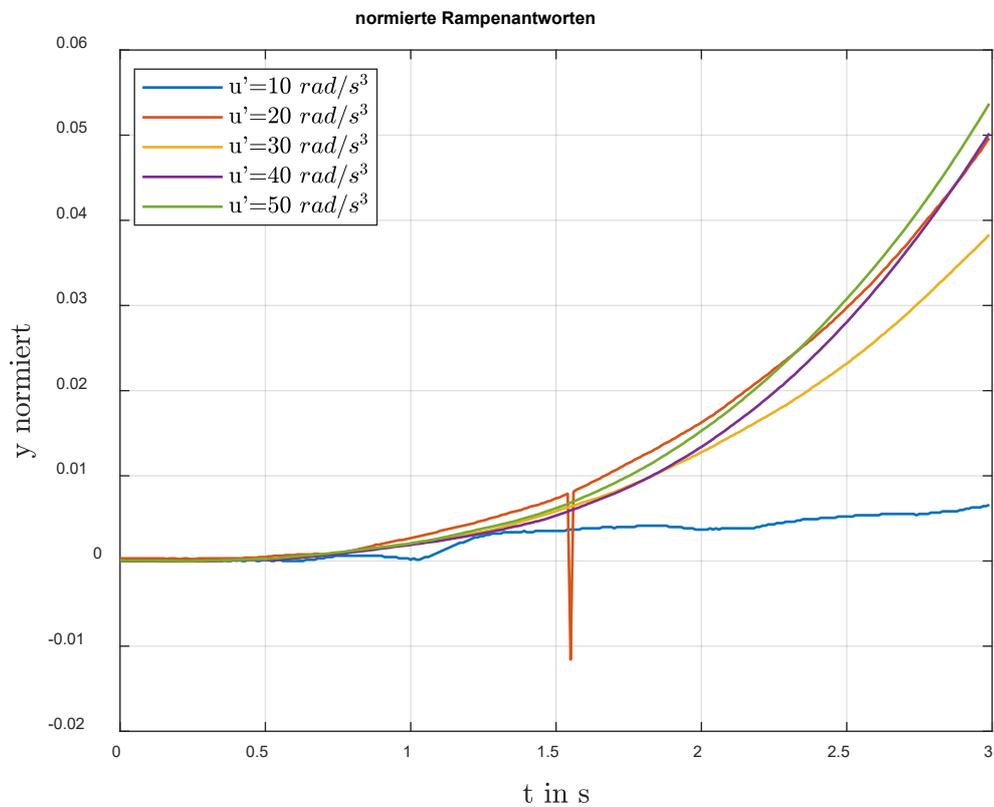


Abbildung 23: Normierte Rampenantworten, um Linearität des Systems zu bestimmen

Schaut man sich die Graphen an, wird sofort deutlich, dass das System schon ab einer Steigung von 20 rad/s^3 sich so verhält wie bei Rampen mit höheren Steigungswerten.

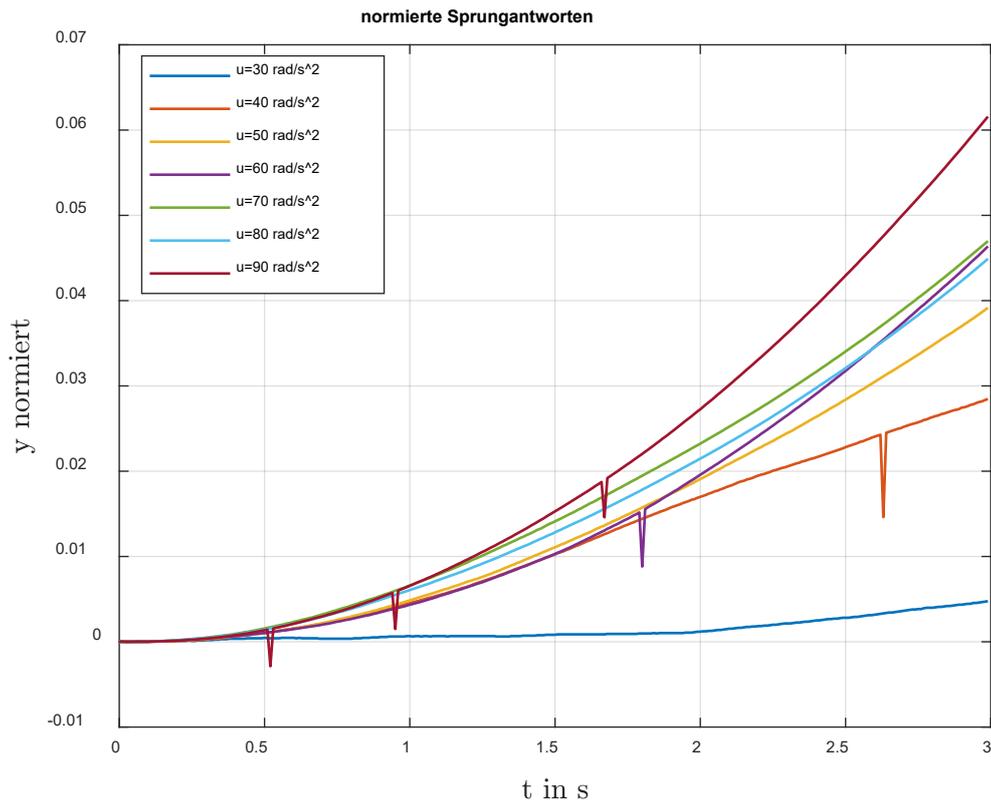


Abbildung 24: Normierte Sprungantworten, um Linearität des Systems zu bestimmen

Die Abbildung 24 zeigt das Verhalten der normierten Sprungantworten. Diese wurden durch Division mit der jeweiligen Sprunghöhe erlangt. Die Sprungantworten scheinen sich ab einer Sprunghöhe von 50 rad/s^2 ausreichend linear zu verhalten. Man kann aber trotzdem eine ausreichend starke Divergenz zwischen den Graphen erkennen. Aus diesem Grund wird für die Streckenidentifikation wie bei der Rampenantwort ein möglichst hoher Steigungs- bzw. Sprungwert verwendet, um ein möglichst lineares Verhalten zu messen.

4.3 Überprüfen des normierten geschlossenen Regelkreisverhaltens anhand von Beispielreglern

Im Entwicklungsstadium des Versuchs wurde eine interessante Eigenschaft im Streckensystem beobachtet. Dieses hatte die Eigenschaft, dass der geschlossene Regelkreis abhängig von dem Sollwertsprung immer schlechtere Dämpfeigenschaften bekommt. Dies war vor allem bei der Proportional-Regelung ein Problem. In dieser Versuchsphase war der Einfluss der Reibung deutlich höher als davor und hat deswegen große Probleme bei der Durchführung verursacht. Die Strecke war sehr schwierig einzuschätzen und eine Streckenidentifikation war nicht wirklich möglich. Das Regelungstechniktool pzMovE konnte keine guten Ergebnisse liefern. Es hat sich herausgestellt, dass der Winkelaufnehmer, aufgrund seiner Nähe zum dazugehörigen Magneten, eine deutlich stärkere Reibung verursacht hat. Dies hatte

unter anderem dafür gesorgt, dass die Reibung stark von der Kompaktheit des Aufbaus abhing, da der Sensor von der Elektronik im Inneren des Satelliten auf den Permanentmagneten gedrückt wurde. Nachdem dieser Fehler behoben wurde, ließen sich mithilfe von pzMove, deutlich einfacher als davor Streckenmodelle herausfinden. Die Hoffnung ist, dass die Strecke sich nun auch ausreichend linear verhält. Aus diesem Grunde heraus hat man sich dafür entschieden, das geregelte Verhalten zu untersuchen. Dabei wurde wie im vorigen fehlerhaften Aufbau das normierte Regelkreisverhalten erarbeitet und analysiert.

Im Gegensatz zum vorigen Aufbau, wird das geregelte Verhalten nicht immer schlechter mit der Sprunghöhe gedämpft. Außerdem ist die Drehrichtung nur noch geringfügig ausschlaggebend.

Die Regelparameter, die in den Abbildungsbeschriftungen von Abbildung 25 und 26 zu lesen sind, setzen sich in den Regler als Übertragungsfunktion in der folgenden Form ein:

$$G_R(s) = K_p + \frac{K_i}{s} + \frac{K_d * s}{T_1 * s + 1}$$

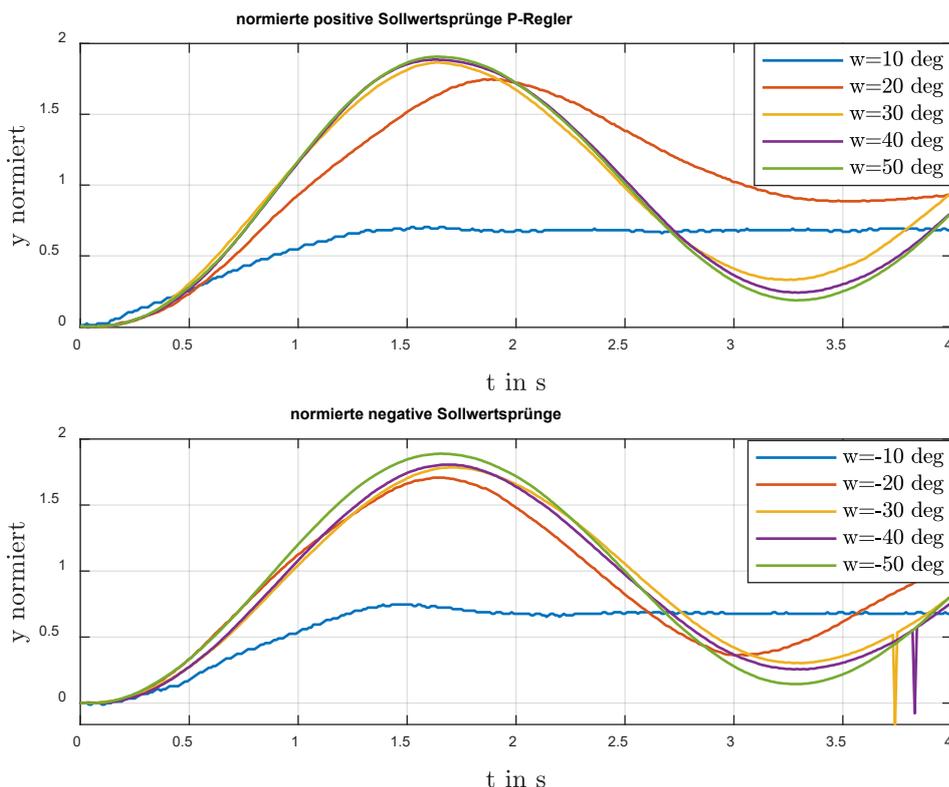


Abbildung 25: Normierte Graphen des P-geregelten geschlossenen Regelkreises $K_p=200$

Man erkennt an den normierten Graphen in Abbildung 25 eine gute Linearität, denn diese ähneln sich ab einer Sollwertsprunghöhe von 30 *deg* zueinander.

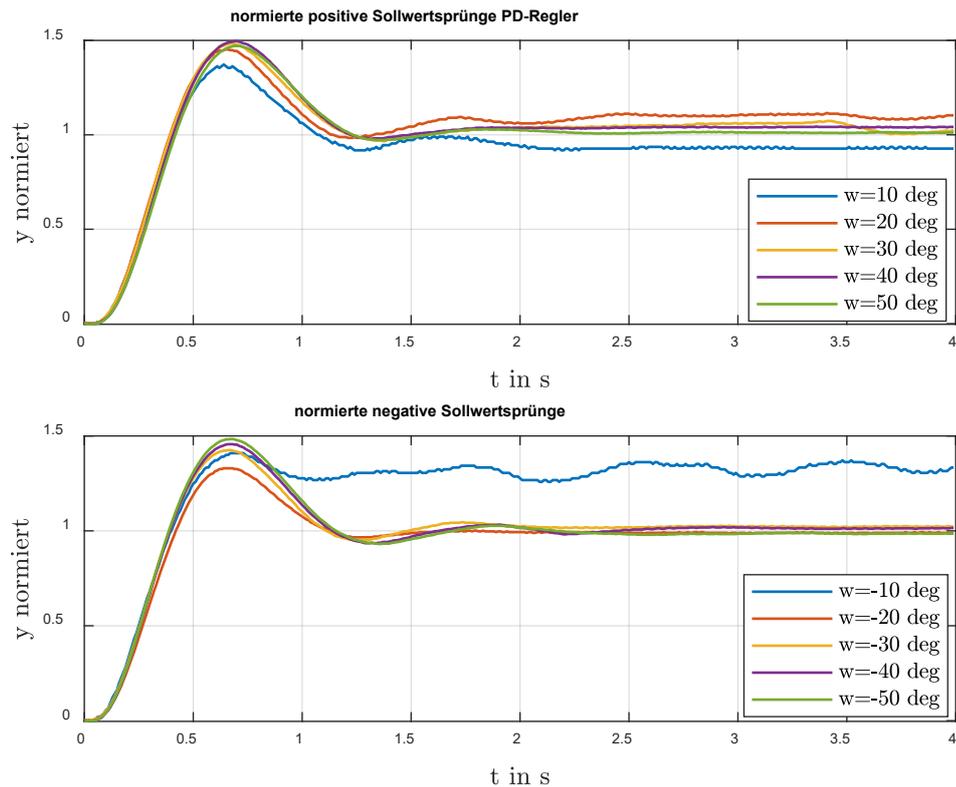


Abbildung 26: Normierte Graphen des PD-geregelten geschlossenen Regelkreises $K_p=400$; $K_d=200$; $T_1=0.1$ s

Das gleiche Linearität kann im PD-geregelten geschlossenen Regelkreislauf beobachtet werden. Die Abbildung 26 zeigt eine noch größere Ähnlichkeit der Graphen zueinander, was wahrscheinlich aus dem doppelt so großen Proportional-Anteil des Reglers resultiert. Aufgrund des linearen geregelten Verhalten wird davon ausgegangen, dass die hohen Stellsignale der beiden Regler relativ zur Mindest-Stellgröße so groß sind, dass die Haftreibung fast keinen Effekt mehr hat. Man erkennt an der Abbildung auch, dass der Regler situationsgemäß nicht mehr richtig funktionieren kann. Dies ist zum Beispiel der Fall, wenn der Abstand zwischen Ist- und Sollwert so klein ist, dass das resultierende Stellsignal kleiner als das Mindest-Stellsignal zur Überwindung der Haftreibung wird.

4.4 Durchführung der Identifikation des Streckenmodells

Um ein möglichst lineares Streckenverhalten zu erhalten, wird eine Sprungantwort mit einer Höhe von 500 rad/s^2 aufgenommen. Die Abbildung 27 zeigt die Graphen des verwendeten Datensatzes. Dabei wird für die Identifikation das Ausgangssignal in rad umgerechnet.

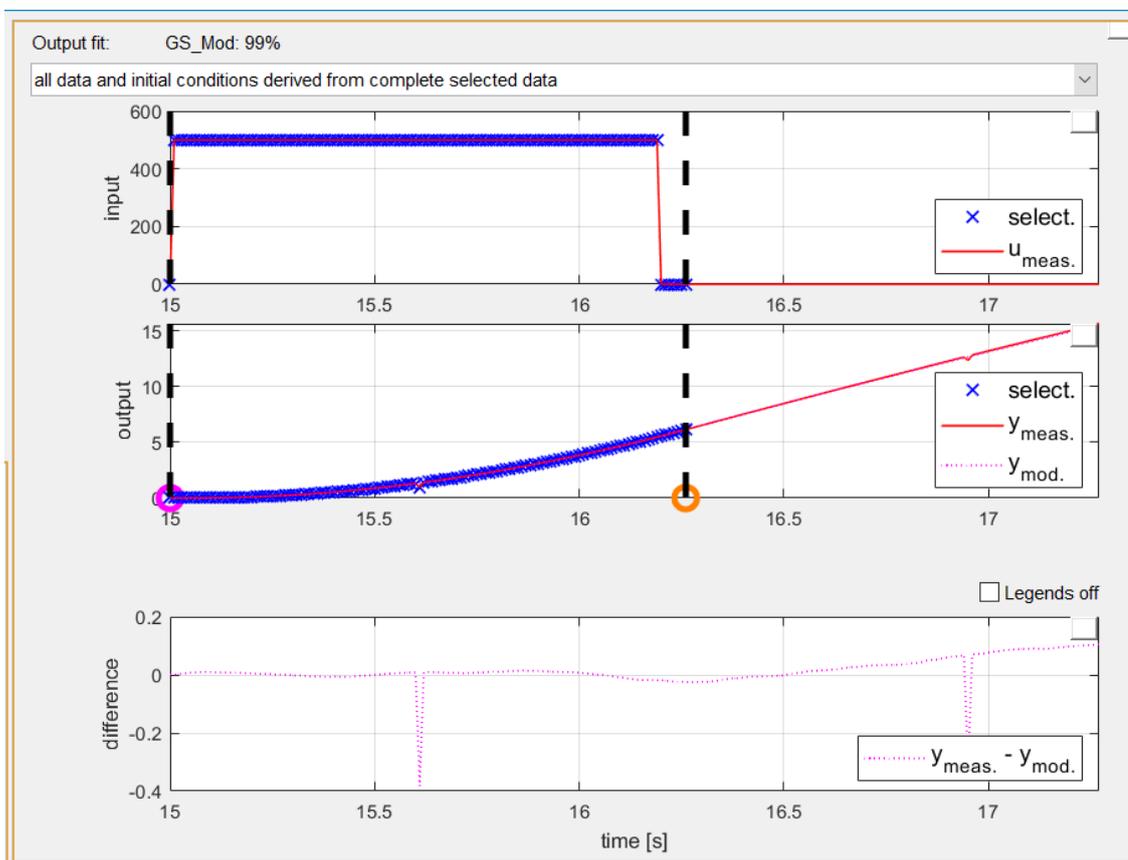


Abbildung 27: Für die Identifikation verwendeter Messdatensatz

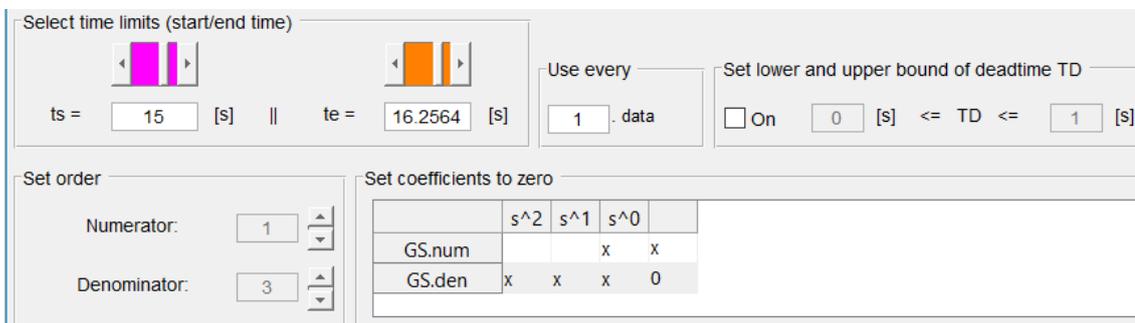


Abbildung 28: Vorgabe der Zähler- und Nenner-Ordnung in pzMove

Als Herangehensweise beim Wählen der Zähler- und Nenner-Ordnung wird zuerst auf die Ordnung des physikalischen Modells aus Gleichung (6) zurückgegriffen und überprüft, inwiefern sich der Fit-Wert verändert. Die Abbildung 28 zeigt, wie die Koeffizienten zur

Identifikation vorgegeben werden. Nun wird versucht durch Veränderung der Ausgangslage den Fit-Wert des Modells zu verbessern. Zuerst wurde versucht für das Modell eine geringere Nennerordnung zu erzwingen, wobei sich der Fit-Wert leicht verschlechtert hat. Eine Erhöhung der Nennerordnung wiederum hat die Modellfitness stark reduziert. Danach wurde die Zählerordnung erhöht, wodurch sich der Fit-Wert von 97.5% auf 99% verbessert hat. Eine weitere Erhöhung der Zählerordnung hat die Modellfitness wiederum verschlechtert.

Die Identifikation mit der Sprungantwort ergibt bei pzMove das folgende Modell als Gleichung (7):

$$G_s(s) = \frac{0.17121 * (1 - 0.03322 * s)}{(1 + 9.794 * s) * (1 + 0.008751 * s) * s}$$

4.5 Vergleich der realen Strecke mit der simulierten Strecke

Für die Vergleichsmessungen wurden ebenfalls möglichst große Eingangssignale verwendet, um die Nichtlinearität der Strecke kleinzuhalten.

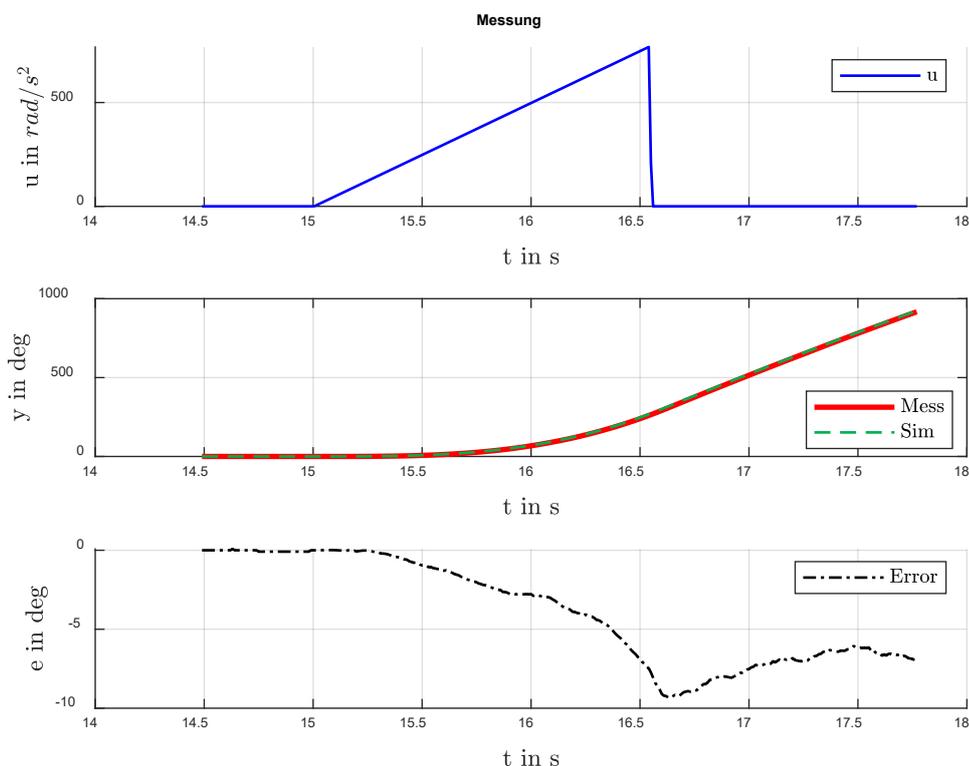


Abbildung 29: Rampenantwort einer Eingangsrampe mit Steigung $\dot{u} = 500 \text{ rad/s}^3$

Der in der Abbildung 29 gezeigte Vergleich von gemessener und simulierter Rampenantwort zeigt bei einem zurückgelegten Winkelweg von 900 deg eine Abweichung e von weniger als 10 deg .

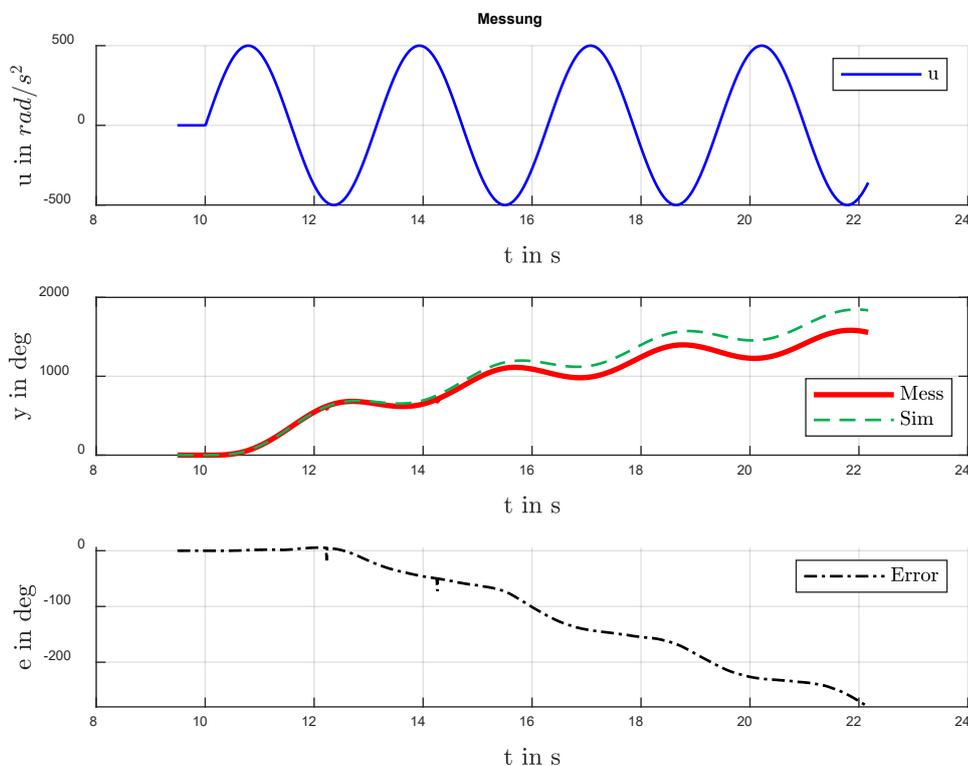


Abbildung 30: Sinusantwort mit einer von Amplitude $u=500 \text{ rad/s}^2$ und einer Kreisfrequenz von 2 1/s

In Abbildung 30 sieht man den Vergleich des simulierten und gemessenen Sinusverhalten, sieht man für die ersten zwei Sekunden eine sehr geringe Abweichung nahe null. Mit vergangener Zeit erhöht sich der Abstand zwischen Simulation und Messung. Dies lässt sich unter anderem durch die Haftreibung erklären, denn wie es scheint, erhöht sich der Abstand stärker zwischen den beiden Kurven ungefähr immer dann, wenn der Satellit kurzzeitig keine Drehgeschwindigkeit mehr hat.

4.6 Vergleich zwischen physikalischen und experimentellen Lagemodell des Satelliten

Aus den Werten der identifizierten Übertragungsfunktion, sollen die physikalischen Größen zurückgerechnet und hinsichtlich ihrer Plausibilität bestimmt werden. Das entstandene Modell, sieht dem physikalischen Modell sehr ähnlich. Einzig im Zähler befindet sich eine zusätzliche negative Nullstelle. Eine negative Nullstelle bedeutet, dass das Ausgangssignal

sich zuerst in die andere Richtung bewegt. Da diese Reaktion in keinen der gemessenen Graphen vorkommt, kann davon ausgegangen werden, dass die negative Nullstelle einen geringen Einfluss auf das System zeigt. Des Weiteren hat die Polstelle $(1 + 9.794 * s)$ einen deutlich größeren Einfluss in der Streckendynamik und ist ungefähr um das Tausendfache langsamer als die andere Polstelle und um das Hundertfache langsamer als die Nullstelle.

Vergleicht man beide Gleichungen (6) und (7) zeigen sich plausible Gemeinsamkeiten.

$$G_{spn}(s) = \frac{\frac{J_{Mot}}{k}}{\left(1 + \frac{J_{Sat}}{k} * s\right) * (1 + T_{Mot} * s) * s}$$

$$G_s(s) = \frac{0.17121 * (1 - 0.03322 * s)}{(1 + 9.794 * s) * (1 + 0.008751 * s) * s}$$

Vergleicht man die Koeffizienten in der Übertragungsfunktion $G_{spn}(s)$ in denen sich die Massenträgheitsmomente befinden, mit den dazugehörigen Zahlenwerten, erkennt man den Einfluss, den diese haben. Bildet man mit den Koeffizienten aus Gleichung (6) ein Verhältnis, kürzt sich der Reibkoeffizient k heraus. Übrig bleibt das Massenträgheitsmomentverhältnis von Satellit und Drallrad. Gleichung (8):

$$\frac{\frac{J_{Sat}}{k}}{\frac{J_{Mot}}{k}} = \frac{J_{Sat}}{J_{Mot}} = \frac{9.794}{0.17121} = 57.2046$$

Dieses Ergebnis erscheint realistisch in der Hinsicht auf die ungefähren Massenträgheitsmomente. Bei der Bestimmung der Massenträgheitsmomente wurden die Bestandteile des Aufbaus bezüglich Längenmaße, Gewicht und Abstand zur Drehachse vermessen.

Zur Bestimmung des Massenträgheitsmoments des Drallrads und der Plattform wurde die Formel für den Zylinder mit homogener Dichte verwendet [10, Seite 182]. Wobei m die Masse und r der Radius des Zylinders sind:

$$J_{Zyl} = m * r^2 / 2$$

Der Hauptkörper wird als Quader mit homogener Dichte gesehen. Dabei werden die Breiten- und Längenmaße b und l betrachtet [10, Seite 182]:

$$J_{Quad} = m * (b^2 + l^2) / 12$$

Der LiPo-Akkupack und der Winkelverbinder werden ebenfalls als mehrere Quader betrachtet. Diese befinden sich aber nicht auf der Drehachse des Satelliten und müssen mit Hilfe des Steinerschen Satz verrechnet werden [10, Seite 178]:

$$J_{res} = J_{Quad} + m_{Quad} * r_s^2$$

Dabei beschreibt J_{Quad} das in der vorigen Formel beschriebene Massenträgheitsmoment eines Quaders, dessen Drehachse durch seine Schwerpunktmittel geht. Damit sich das resultierende Massenträgheitsmoment J_{res} auf die Satelliten-Drehachse bezieht, muss der Abstand der dazu parallel liegenden Quaderdrehachse r_S und die Masse des Quaders m_{Quad} betrachtet werden. Die berechneten Werte werden in Tabelle 1 angezeigt.

Tabelle 1: Ungefähre Massenträgheitsmomente

Bauteil	Massenträgheitsmoment um die Drehachse
Hauptkörper	$71 * 10^{-6} \text{kg} * \text{m}^2$
Winkelverbinder	$121 * 10^{-6} \text{kg} * \text{m}^2$
LiPo-Akkupack	$197 * 10^{-6} \text{kg} * \text{m}^2$
Plattform	$58 * 10^{-6} \text{kg} * \text{m}^2$
Satellitenkörper insgesamt: J_{Sat}	$447 * 10^{-6} \text{kg} * \text{m}^2$
Kleine Drallscheibe	$1 * 10^{-6} \text{kg} * \text{m}^2$
Große Drallscheibe	$10 * 10^{-6} \text{kg} * \text{m}^2$
Drallrad insgesamt: J_{Mot}	$11 * 10^{-6} \text{kg} * \text{m}^2$

Die jeweiligen Massenträgheitsmomente werden zusammengefasst, sodass die ungefähren Werte für J_{Sat} und J_{Mot} ermittelt werden.

Dabei ergibt das Massenträgheitsverhältnis der errechneten Werte:

$$\frac{J_{Sat}}{J_{Mot}} = \frac{389 * 10^{-6} \text{kg} * \text{m}^2}{11 * 10^{-6} \text{kg} * \text{m}^2} = 40.64$$

Damit befindet sich dieses Verhältnis in derselben Größenordnung wie das Ergebnis aus Gleichung (8) von 57.2046.

Die Zeitkonstante der anderen Polstelle, mit dem Wert 0.008751 s wiederum, scheint eine realistische Annahme für die Motorzeitkonstante T_{Mot} zu sein.

5 Der Reglerentwurf

Beim Reglerentwurf wird zuerst versucht unter Verwendung der Einstellregeln von Chien, Hrones und Reswick erste mögliche Regelungen zu erhalten. Daraufhin wird durch Anpassen der Regelungsparameter versucht, einen Regler zu entwerfen, der der Regelgüte entspricht.

5.1 Bestimmen eines Reglers über die Einstellregeln von Chien, Hrones und Reswick

Die Einstellregeln nach Chien, Hrones und Reswick ermöglichen, das schnelle Auslegen eines möglicherweise geeigneten Reglers. Der Vorteil bei diesen ist, dass sie im Gegensatz zu den Einstellregeln von Ziegler und Nichols auch auf grenzstabile System angewendet werden können, d.h. die Strecke darf einen Integrator haben. Des Weiteren schmiegt sich die resultierende Sprungantwort aperiodisch an den Sollwert bzw. schwingt um 20% über [13, Kap.7 Seite 14-15]. Normalerweise werden die Einstellregeln für die empirische Bestimmung einer Regelung angewendet. Dabei werden experimentell Kennwerte aus der Sprungantwort der Strecke ermittelt. Da die identifizierte Strecke einen Integrator aufweist, werden, wie in Abbildung 31 zu sehen ist, die Verzugszeit und die stationäre Verstärkung als Kennwerte benötigt. Die stationäre Verstärkung errechnet sich beim Einheitssprung aus dem Steigungswert der stationären Geradenasymptote.

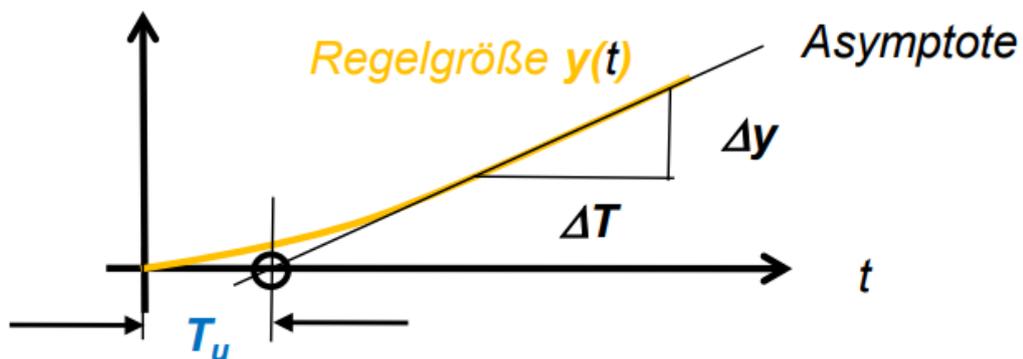


Abbildung 31: Darstellung der Kennwerte in der Regelstreckensprungantwort [13, Kap. 7 Seite 15]

Dabei werden die Regelungsparameter abhängig vom gewählten Reglertypen automatisch über pzMove umgesetzt und simuliert.

Da der bürstenlose Gleichstrommotor keine beliebig hohen Drehzahlen erreichen kann, darf die Anstiegszeit des Satelliten bei einem Sollwertsprung nicht zu lange sein. Außerdem bedeutet eine langsame Anstiegszeit auch eine geringe Motorbeschleunigung. Dies ist zum einen nicht sinnvoll, da die Motordrehzahlaufösung irgendwann nicht mehr ausreicht und

zum anderen führt eine geringe Drehzahlbeschleunigung dazu, dass der Satellit sich aufgrund der Haftreibung nicht in Bewegung setzt.

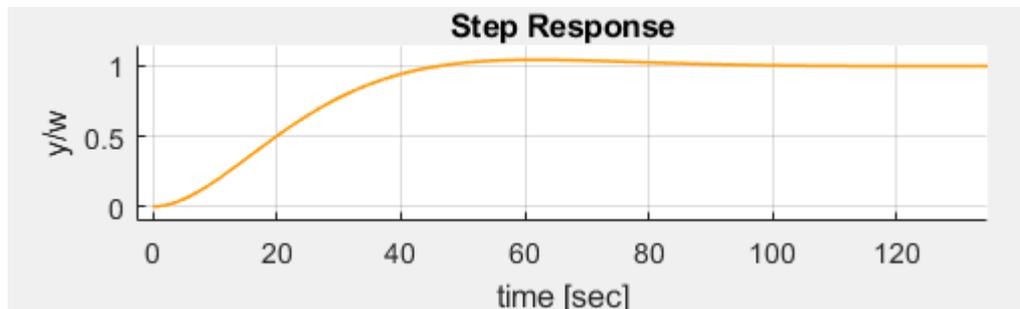


Abbildung 32: Sprungantwort-Simulation des P-Reglers nach den Einstellregeln von Chien, Hrones und Reswick

$$G_R(s) = 0.29691$$

Wie in Abbildung 32 zu sehen ist, hat der durch die Einstellregeln bestimmte P-Regler ein geringes Überschwingen und die Form des Anstiegs ist ebenfalls passend. Das Problem bei der Sprungantwort von diesem Regler liegt in der Zeit, in dem der geschlossene Regelkreis sein Ziel erreicht. Der Sollwert wird erst nach ca. 50 Sekunden erreicht, was im realen System nicht umsetzbar ist.

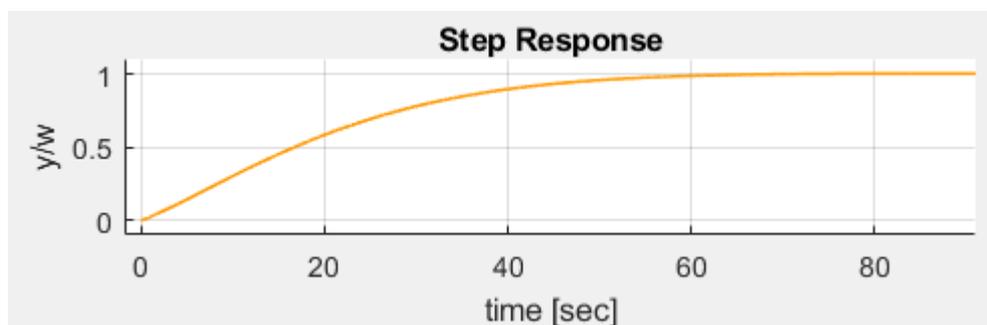


Abbildung 33: Sprungantwort-Simulation des PD-Reglers nach den Einstellregeln von Chien, Hrones und Reswick

$$G_R(s) = 0.29691 + \frac{1.4602 * s}{0.1 * s + 1}$$

Ein ähnliches Problem ist in Abbildung 33 zu sehen, dort ist die Ausregelzeit des Sollwert-sprungs des PD-geregelten System ebenfalls zu langwierig.

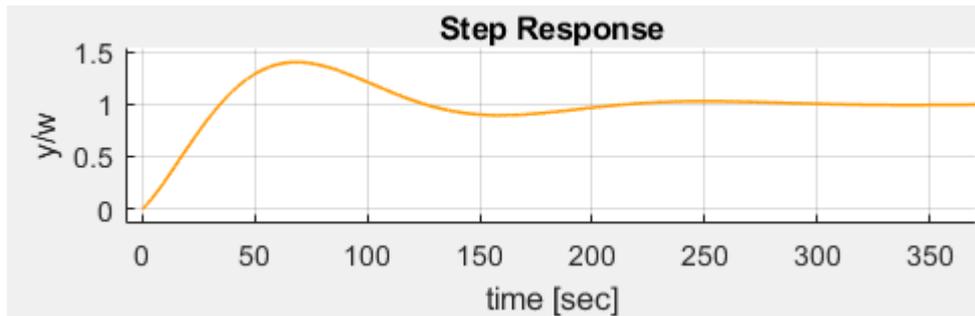


Abbildung 34: Sprungantwort-Simulation des PID-Reglers nach den Einstellregeln von Chien, Hrones und Reswick

$$G_R(s) = 0.23753 + \frac{0.0075465}{s} + \frac{1.1682 * s}{0.1 * s + 1}$$

Als Letztes wurde versucht mit Hilfe eines Integrators die Ausregelzeit zu verbessern. Dies war leider nicht erfolgreich, die Abbildung 34 zeigt, dass sich nach den Einstellregeln ein Integrator negativ auswirkt. Die Anstiegszeit hat sich nicht verbessert und die Regelgröße schwingt sogar über.

5.2 Bestimmen eines Reglers unter Berücksichtigung der Systemgrenzen

Damit der Regler seine Funktion erfüllen kann, muss das Stellsignal die Reibung überwinden können. Das bedeutet, dass die Auslegung des Reglers, hinsichtlich der Anstiegs- und Beruhigungszeit begrenzt werden muss.

Es wird nun durch Probieren ein Regler gesucht, der sehr schnell den Sollwert erreicht. Es wird sich an den Beispiel-PD-Regler aus Kapitel 4.3 orientiert. Dieser arbeitet mit K_p - und K_d -Werten, hat aber eine stark überschwingende Dynamik. Im Nachfolgenden werden drei Regler vorgestellt, die verschiedene Vor- und Nachteile hinsichtlich Führungsverhalten und Störkompensation aufweisen.

5.2.1 Verschiedene Regler-Vorschläge

Die in den Abbildungen 35 bis 37 gezeigten Sollwertsprungantworten, zeigen das Verhalten der verschiedenen Regler A bis C. Bei der Bestimmung der gewählten Regelungsparameter wurde darauf geachtet, den Proportional-Anteil schrittweise zu erhöhen, um das dynamische Verhalten der geschlossenen Regelkreise zu verändern. Unter den jeweiligen Abbildungen auf der nächsten Seite befinden sich die korrespondierenden Regler-Übertragungsfunktionen.

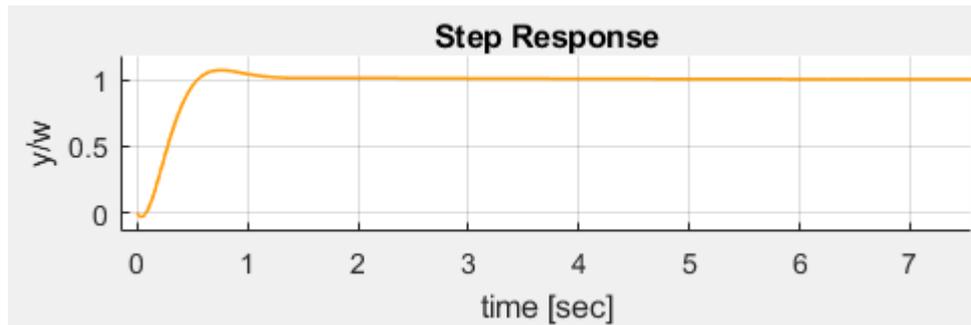


Abbildung 35: Sprungantwort-Simulation des Reglers A

$$G_{RA}(s) = 50 + \frac{200 * s}{0.1 * s + 1}$$

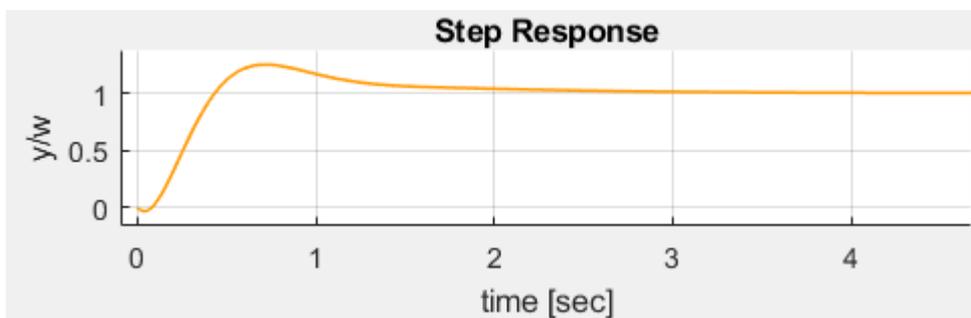


Abbildung 36: Sprungantwort-Simulation des Reglers B

$$G_{RB}(s) = 200 + \frac{200 * s}{0.1 * s + 1}$$

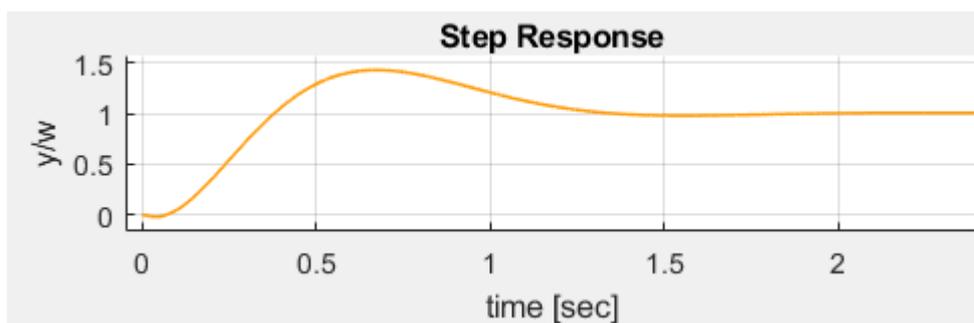


Abbildung 37: Sprungantwort-Simulation des Reglers C

$$G_{RC}(s) = 400 + \frac{200 * s}{0.1 * s + 1}$$

5.2.2 Anwendung des entworfenen Reglers an realer Strecke und Vergleich mit der Simulation

Die theoretischen Regler werden an der realen Strecke ausprobiert. Dabei ist es wichtig, das Verhalten des Systems messtechnisch zu erfassen. Wobei dann das theoretische Verhalten mit dem realen verglichen wird. Durch Verwendung des in Kapitel 2.2.2 beschriebenen Simulink-Programm *MotorRun_Reg_Sim.slx* kann die Regelung an der realen Strecke und der simulierten Strecke parallel angewendet werden. Die dazu notwendigen Regelparameter, für die Regelungsübertragungsfunktion $G_R(s)$ und das Regelstreckenmodell $G_S(s)$ werden dafür in den *Workspace* übertragen und von dem Simulink-Programm übernommen. Unter der Registerkarte *Hardware* wird der Regler durch Drücken auf *Monitor & Tune* gestartet. In der Initialisierungsphase gibt der an dem Akku angeschlossene ESC seine akustischen Signale und lässt den Motor in seiner Ausgangsdrehzahl hochfahren. Dabei sollte der Satellit festgehalten werden, damit dieser sich nicht zu weit von seiner Nullposition wegdreht. Nach zehn Sekunden schaltet sich der Regler an und führt den Satelliten in seine Nullposition und nach weiteren fünf Sekunden wird die Sollwertgröße auf den gewünschten Wert gestellt.

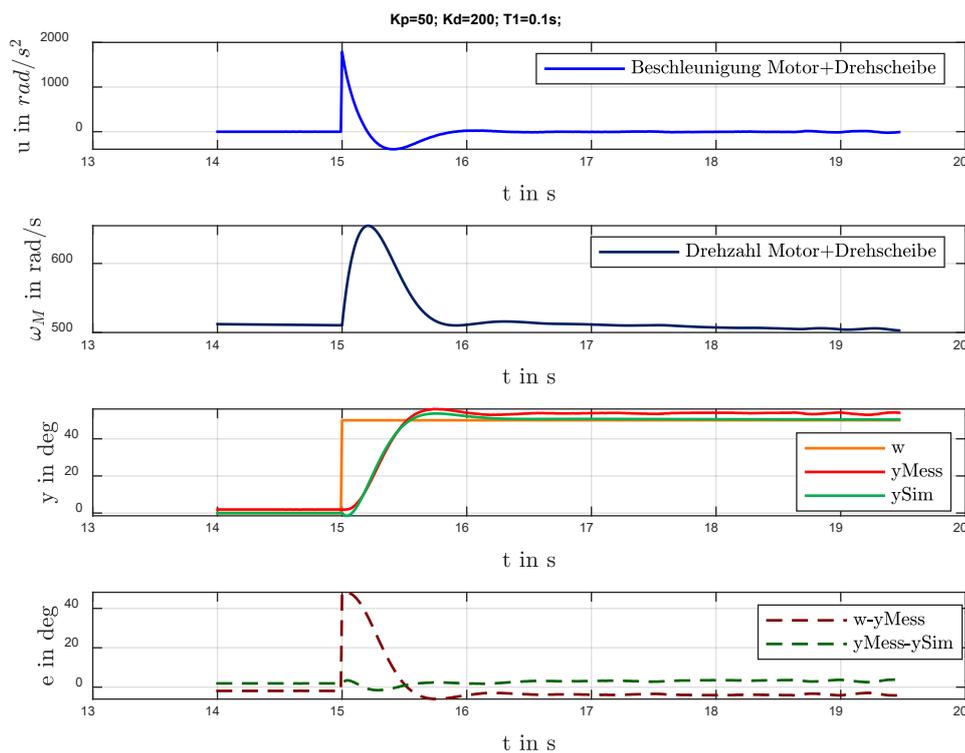


Abbildung 38: Reale Systemreaktion in Vergleich zur Simulation im Regelkreis A

Nun können reale und simulierte Strecke zusammen getestet werden. Das Regelkreisverhalten wurde durch die Messung mehrerer verschieden großer Sollwertsprünge in die positive und negative Richtung getestet.

Die Abbildungen 38 bis 40 zeigen das Verhalten der geschlossenen Regelkreise A, B und C bei einem Sollwertsprung. Dabei zeigt der erste Plot die Stellgröße u . Der zweite Plot zeigt das Motordrehzahlverhalten an. Der dritte Plot zeigt den Sollwertsprung in Orange, das gemessene Ausgangssignal y_{Mess} in Rot und das simulierte Ausgangssignal y_{Sim} in Grün. Der vierte Plot zeigt sowohl die Regeldifferenz $w-y_{Mess}$ in Dunkelrot als auch den Modellfehler $y_{Mess}-y_{Sim}$ in Dunkelgrün.

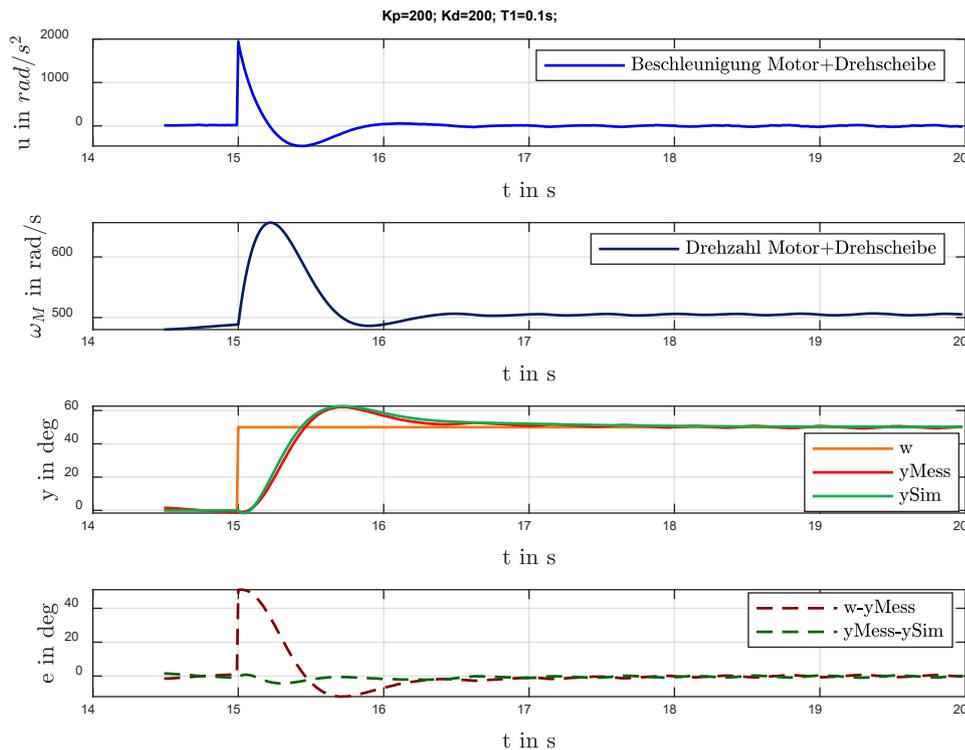


Abbildung 39: Reale Systemreaktion in Vergleich zur Simulation im Regelkreis B

Man kann anhand des Modellfehlers sehen, dass sich reale und simulierte Regelstrecke stark ähneln. Das Modell beschreibt also das reale Verhalten gut genug.

Mit der graphisch dargestellten Simulation können nun auch die Unterschiede zwischen dem realen und dem simulierten geschlossenen Regelkreis gezeigt werden, die bei der aktiven Störung der Regelstrecke, bei kleinen Sollwertsprüngen oder durch das Erreichen der Stellgliedergrenzen entstehen können.

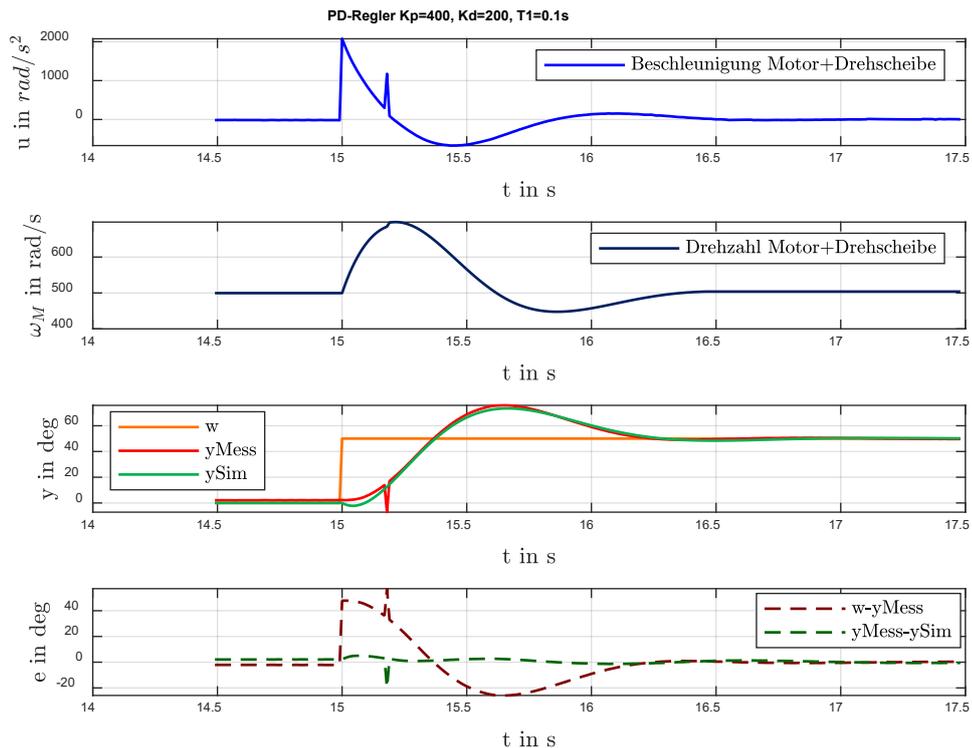


Abbildung 40: Reale Systemreaktion in Vergleich zur Simulation im Regelkreis C

5.2.3 Bewertung der Regelgüte

Bei der Bewertung der Regelgüte werden mehrere Aspekte betrachtet. Im Folgenden werden die Regler A, B und C nach Führungsverhalten und Störunterdrückung bewertet. Dabei wird betrachtet, wie sich die Regelkreise dynamisch und nach der Beruhigung stationär verhalten. Des Weiteren wird auf das Stellglied, ferner auf die resultierende Motordrehzahl geachtet.

5.2.3.1 Stationäres und dynamisches Führungsverhalten

Alle betrachteten Regelkreise weisen ein Ausgangssignal mit einem Überschwinger auf, das sich dann dem Sollwert anschmiegt. Für die bessere Vergleichbarkeit wurden in Tabelle 2 die Regler hinsichtlich der verschiedenen Kriterien nebeneinander aufgeführt. Dabei wird sich an den Messdaten aus den Abbildungen 38-40 orientiert, in denen jeweils an dem geschlossenen Regelkreis ein Sollwertsprung von 50 deg angelegt wurde.

Die Kriterien orientieren sich daran, wo sich die jeweiligen Werte kurz vor dem Sollwertsprung befanden. Die *Überschwingzeit* beschreibt den Zeitpunkt nach der Sollwertveränderung in dem das Ausgangssignal am weitesten über den Sollwert geschwungen ist. Dabei wird der vorige Zeitwert von 15 Sekunden von der Überschwingzeit abgezogen. Die *Überschwingweite* beschreibt, wie viel weiter der höchste Punkt des Überschwingers zum Sollwert liegt. Sie wird in der Tabelle prozentual zum Sollwert angegeben. Die *Beruhigungszeit* beschreibt den Zeitpunkt nach dem Überschwinger, ab dem der Istwert nicht mehr höher als

5% zum Sollwert abweicht. Hier wird sich an der Simulation orientiert, da nicht jeder der vorgeschlagenen Regler unterhalb der gewählten Toleranz $\epsilon = 5\%$ kommt. Die Regelabweichung ist auch von der gegebenen Sollwertsprunghöhe abhängig. Dies liegt an der schon in Kapitel 4.1 beschriebenen Reibung, bei der die Regelung bei kleinen Regeldifferenzen nicht mehr dazu imstande ist, ein ausreichend großes Stellsignal zur Überwindung zu liefern.

Tabelle 2: Vergleich zwischen den Reglern hinsichtlich des Führungsverhaltens

Regler	A	B	C
Überschwingzeit	0.77 s	0.70 s	0.67 s
Überschwingweite	11.8 %	25.2 %	46.9 %
Beruhigungszeit (Simulation $\epsilon = 5\%$)	0.96 s	1.76 s	1.18 s
Bleibende Regelabweichung	3.9 deg	0.7 deg	0.6 deg
Max. Motordrehzahldifferenz in <i>rad/s</i>	144.2	168.1	187.8
Motordrehzahldifferenz. nachher in <i>rad/s</i>	5.4	17.7	4.3

Die *bleibende Regelabweichung* beschreibt, wie weit, nach langer Zeit, der Istwert vom Sollwert entfernt ist. Bei der Bestimmung wurde so vorgegangen, dass im Zeitintervall 3 ... 5 s nach dem Sollwertsprung der größte Abstandswert verwendet wird. Dieser Wert zeigt an, wie gut der jeweilige Regler der Reibung entgegenwirkt. Die *maximale Motordrehzahldifferenz* wird durch Vergleich der Ausgangsmotordrehzahl bestimmt. Sie soll als Maß dafür dienen, wie groß der für die Regelung notwendige Arbeitsbereich des Motors sein muss, da der Satellit sich normalerweise nicht mehr als um 50 deg drehen muss. Die *Motordrehzahldifferenz nachher* dient der Einschätzung, wie oft der Satellit seine Sollwertposition in eine Richtung verändern kann, bis der Motor an seine Grenzen kommt. Sie beschreibt den bleibenden Abstand der neuen Motordrehzahl zum vorigen Wert nach dem Regelungsvorgang.

Schaut man sich die ermittelten Werte an, sieht man, dass die Regelkreise in den Punkten Überschwingzeit, Beruhigungszeit und bei den Motordrehzahldifferenzen sich ungefähr gleich verhalten. Während die Überschwingweite beim Regelkreis A die niedrigste ist, ist dafür die bleibende Regelabweichung am größten. Das bedeutet, dass der Regler A Schwierigkeiten beim Regeln kleinerer Sollwerte haben wird. Die Regelkreise B und C haben dieselbe Regelabweichung, aber C hat dafür eine fast doppelt so große Überschwingweite. Deswegen wird sich dafür entschieden, Regelkreis B, das bessere Führungsverhalten zuzuweisen.

5.2.3.2 Stationäre und dynamische Störkompensation

Bei der Demonstration des Versuchsaufbaus ist, die Störunterdrückung der Satellitenlageregelung wichtiger als das Führungsverhalten des Ausgangssignals, da der Praktikumsstudent die Reaktion der Regelung auch erfühlen sollte.

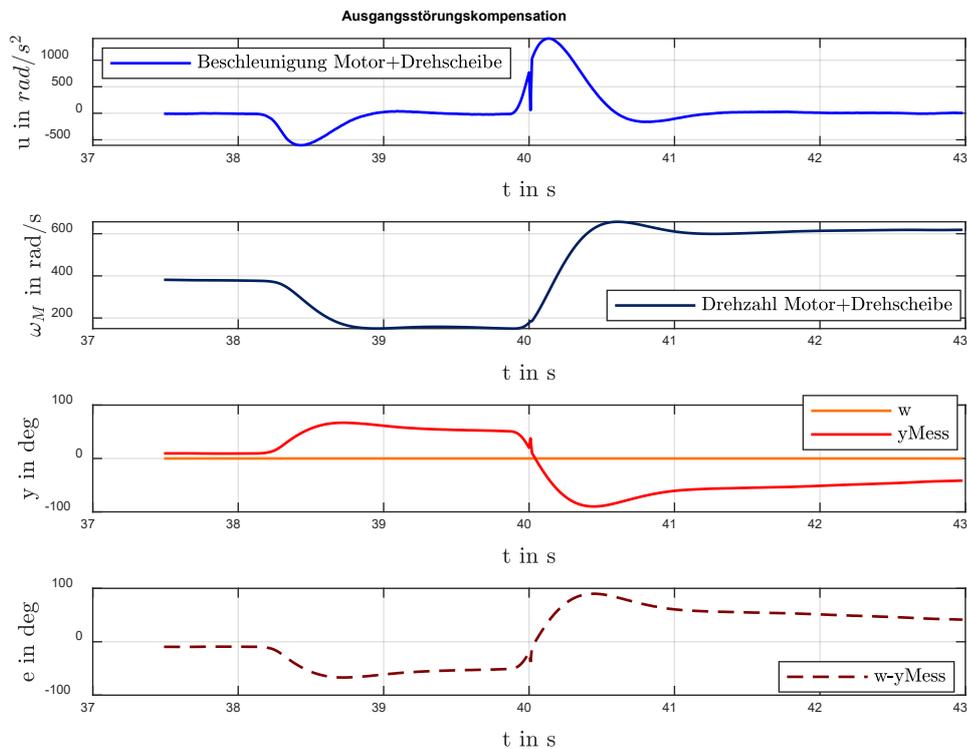


Abbildung 41: Störkompensation des Regelkreis A durch Auslenken des Satelliten von Hand

D.h. der Regler sollte möglichst schnell den Fehler beheben können und durch hohe Motorbeschleunigungen eine spürbare Kraft erzeugen. In den Abbildungen 41 bis 43 werden die Reaktionen der Regelkreise A, B und C aufgezeigt.

Der Regelkreis A zeigt eine schlechte Störkompensation. Dies liegt an dem niedrigen P-Anteil der Regelung. Diese sorgt bei einer Regeldifferenz von 50 deg lediglich für ein Stellsignal von 43.635 rad/s^2 . Dieses Stellsignal sorgt zum einen für ein sehr langsames Zurückkommen und zum anderen befindet sich das Stellsignal nahe der Signalgrenze, um die Haftreibung zu überwinden. Der hohe D-Anteil des Reglers wiederum reagiert nicht so gut auf die Störung, da die Auslenkung des Satelliten von Hand vergleichsweise langsam, gegenüber einem Sollwertsprung ist. Des Weiteren wird bei der Auslenkung von Hand der Satellit einige Zeit lang festgehalten, sodass die Wirkung des D-Anteils beim Auslenkvorang schon verschwindet. D.h. Regler A ist nicht dafür geeignet Ausgangsstörungen ausreichend zu kompensieren.

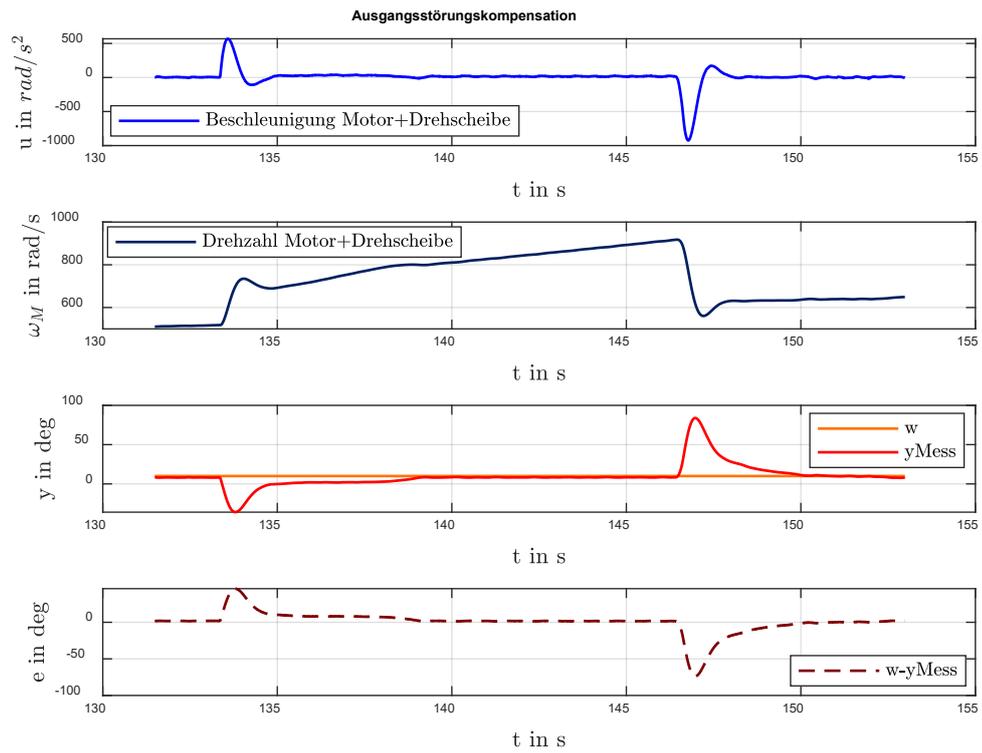


Abbildung 42: Störkompensation des Regelkreis B durch Auslenken des Satelliten von Hand

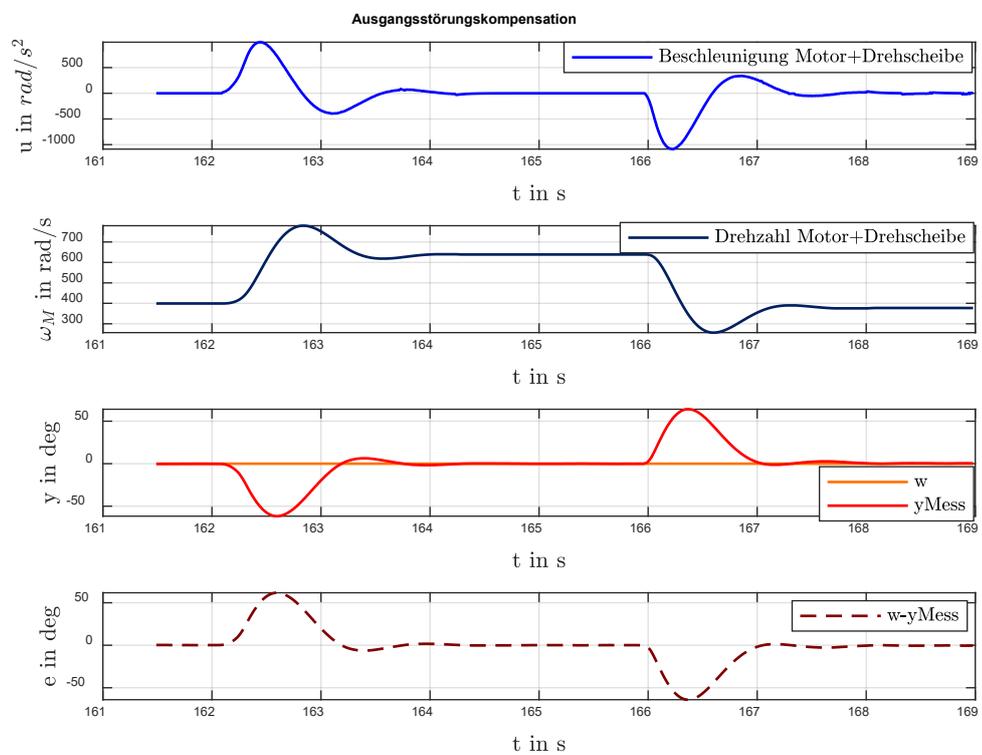


Abbildung 43: Störkompensation des Regelkreis C durch Auslenken des Satelliten von Hand

Der Regler B wiederum soll eine Kompromisslösung zwischen Regler A und C darstellen. Zum einen soll der P-Anteil groß genug sein um das langsame Auslenken von Hand zu umgehen und zum anderen soll der P-Anteil nicht so groß sein, sodass es zu einem starken Überschwinger kommt. Der Unterschied zwischen B und C ist die Geschwindigkeit, mit der die Sollwertgröße erreicht wird. Regler B benötigt für die Ausregelung eine Zeit im Bereich von zwei bis drei Sekunden und Regler C benötigt dafür ungefähr eine Sekunde. Für den Demonstrationsversuch kann es sinnvoll sein, im Sinne der Anschaulichkeit eine langsamere Regler-Reaktion zu bevorzugen. Bewertet man nach diesem Kriterium, dann ist die Anwendung von Regler B gegenüber Regler C sinnvoller.

6 Praktische Demonstration

Dieses Kapitel beschäftigt sich mit den Aspekten der Demonstration und des Hardware-Aufbaus.

6.1 Digitaler Zwilling des geschlossenen Regelkreises

Zur Anschaulichkeit werden beim Versuch zwei *Scope*-Blöcke verwendet. Dabei sollen die Scopes das Regelkreisverhalten in Echtzeit dokumentieren.

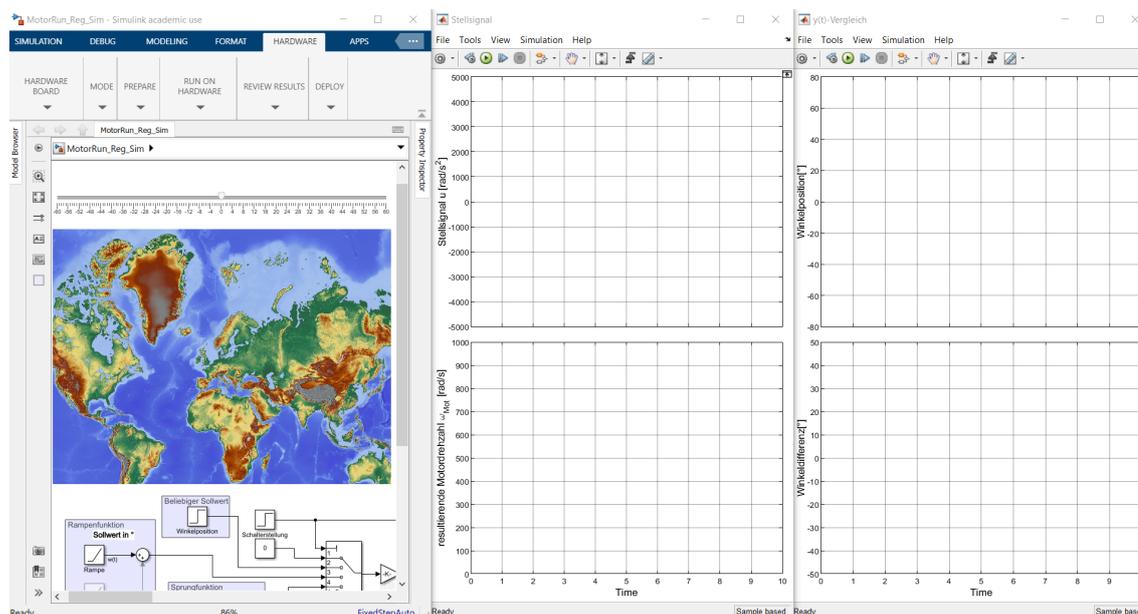


Abbildung 44: Darstellung des Demonstrationsprogramms

Das Programm *MotorRun_Reg_Sim.slx*, welches auch für die Demonstration verwendet werden kann, wird in Abbildung 44 gezeigt. Beim Starten des Programms öffnen sich automatisch die Scopes. Wobei das linke Scope im oberen Fenster das kommandierte Stellsignal u aufzeigt und das untere Fenster zeigt die kommandierte Motordrehzahl ω_{Mot} . Im rechten Scope werden im oberen Fenster Sollwert w , gemessenes Ausgangssignal y_{Mess} und simuliertes Ausgangssignal y_{Sim} in deg angezeigt. Im unteren Fenster werden die Regeldifferenz $w - y_{Mess}$ und der Modellfehler $y_{Mess} - y_{Sim}$ gezeigt.

Auf der linken Seite im Programm sieht man die Weltkarte, die auch als Hintergrund für die Demonstration verwendet wird. Über dieser befindet sich ein *Slider*-Block, mit dem die Sollwertwinkelposition in Bereich von $-60 \dots 60 \text{ deg}$ eingestellt werden kann. Alternativ lässt sich der Sollwert über den Block im *Beliebiger Sollwert in deg* – Fenster einstellen. Gestartet wird das Programm, wie in Kapitel 5.2.2 beschrieben.

6.2 Der Laserpointer-Aufbau zur Veranschaulichung der aktuellen Lage

Im Rahmen dieser Bachelorarbeit wurde wie in Abbildung 45 gezeigt ein CAD-Modell für einen verbesserten Hardware-Aufbau erstellt.

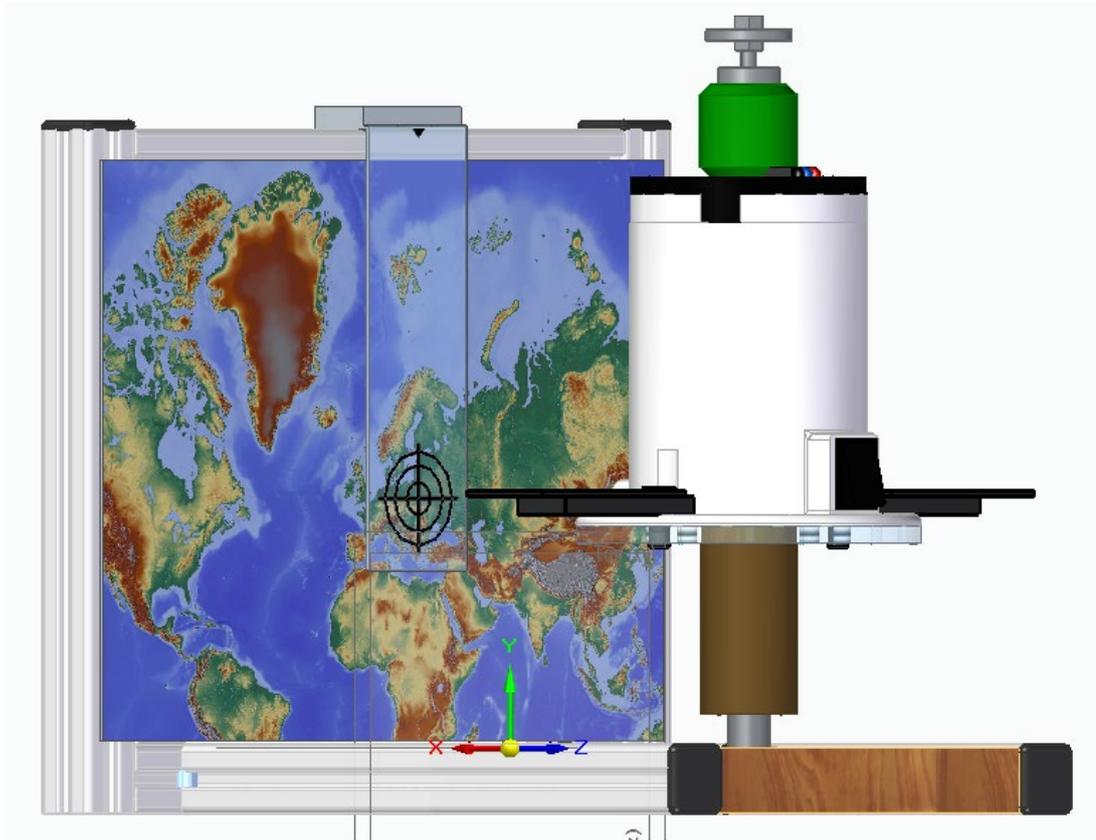


Abbildung 45: Gesamtaufbau des Demonstrationsversuchs

Im Inneren des Aufbaus gibt es die Möglichkeit, Zwischenwände einzusetzen. Dabei sollen die einzelnen Komponenten durch diese befestigt werden können. Auf der Rückseite befindet sich ein Schalter, um den ganzen Versuch ein- bzw. auszuschalten.

Das Gestell wird aus Aluprofilen der Firma Item zusammengebaut. Oberhalb der Weltkarte befindet sich eine Skala zum Vorgeben der Winkelposition. Das Fadenkreuz soll mithilfe einer Lasche auf der Oberseite des Gestells verschiebbar sein. Erstmal wird sich das Fadenkreuz auf einer bedruckbaren Folie befinden. Dieses dient als Ziel für den an den Satellitenkörper befestigten Laserpointer. Der Laserpointer wird hierbei über die Spannungsversorgung des Raspberry Pi Zeros betrieben. Die Weltkarte wird auf einer Hartschaumstoffplatte gedruckt. Wobei diese entweder auf die Vorderseite des Gestells befestigt wird oder in die Aluprofile geschoben wird als Einrahmung.

7 Zusammenfassung und Ausblick

7.1 Resümee der Ergebnisse

In dieser Arbeit wurde erfolgreich ein Modell eines Satelliten gebaut, welches leicht zu transportieren ist und sich wie ein sich frei im All schwebender Satellit mit Hilfe einer Lageregelung um die eigene Achse drehen kann. Die entworfene Regelung kann die Satellitenorientierung bei Auslenkstörungen von Hand gut entgegenwirken. Dabei wird die Lage des Satelliten durch ein Drallrad auf der Oberseite des Aufbaus gestellt. Das Drallrad wird dabei durch einen bürstenlosen Gleichstrommotor angetrieben und erzeugt so ein äußeres Drehmoment, wo durch der Satellit sich in Bewegung setzt. Zur Bestimmung des Dralls wurde das Drehzahlverhalten des über Electronic Speed Controller drehzahlgeregelten Motors bestimmt und für die weitere Bearbeitung nutzbar gemacht. Es ist gelungen, die Lage des Satelliten über einen magnetisch induktiven Winkelaufnehmer zu messen. Über den Mikrocontroller Raspberry Pi Zero WH, wurden alle wichtigen Komponenten des Aufbaus zielführend verbunden. Dabei kann mit erstellten Simulink-Programmen über einen WLAN-Router kommuniziert werden. Das Modellieren des Verhaltens der realen Strecke ist trotz vorhandener Nichtlinearität mit Hilfe von MATLAB, Simulink und pzMove gelungen. Mit diesem Streckenmodell wurde eine Satellitenlageregelung entworfen. Das dazu erstellte Programm ermöglicht die Regelung den Satelliten sowohl in der Realität als auch in einer in Echtzeit laufenden Simulation. Zusätzlich bietet das Programm die Möglichkeit, die Sollwertposition zu wählen.

7.2 Mögliche Perspektiven für Weiterentwicklungen

Das in dieser Arbeit vorgestellte Ergebnis lässt sich verbessern. In diesem Aufbau wurde darauf vertraut, dass der ESC tatsächlich die gewünschten Geschwindigkeiten und Beschleunigungen erzeugt. Hierbei könnten Motordrehzahlmessungen über die in der Arbeit von M. Krettek verwendete Sensorik erfolgen[14]. Zusätzlich könnte ein Beschleunigungsregler mittels eines Pseudo-Differentials in der Form $s/(1 + Ts)$ umgesetzt werden.

Des Weiteren können die dynamischen Eigenschaften des Satellitenkörpers verbessert werden. Verwendet man einen kleineren und damit leichteren LiPo-Akku, kann durch Weglassen des Akkus des Raspberry Pi Zero WH, im Inneren des Aufbaus Platz geschaffen werden (siehe Kapitel 6.2). So kann überschüssiges Gewicht wie der schwere Winkelverbinder entfernt werden. Der so entstehende neue Satellitenkörper hätte ein geringeres Massenträgheitsmoment und ließe sich leichter durch den Motor stellen.

Anhang:

MATLAB-Programm – Messdaten_zusammenf.m

```
%Zusammenfassen der Mess- und kommandierten Eingangssignale
td=0.01; %Abtaste
gg=0:td:(length(Angles)-1)*td; %Zeitvektor
mat=[gg',Speed,Angles]; %Zusammenfassen
t0=14.5;%Startzeit
mat=mat(t0/td:end,:); %Bestimmen der Datensatz-Matrize für pzMove

%mat=mat(1:250,:); %Vorgabe der Matrizenlänge

t = mat(:,1); %Zeitvektor
ui=mat(:,2); %Motordrehzahlvektor
u=ableit(ui,td); %kommandiertes Eingangssignal-Vektor

mat(:,2)=u; %Bestimmen der Datensatz-Matrize für pzMove
mat(:,3)=mat(:,3)-mat(1,3); %Anfangsbedingung y0 auf 0 setzen
y=mat(:,3)*180/pi; % Ausgangssignal-Vektor

%Plotten
%% subplot 1
figure('NumberTitle','off','Name','mat')
% display u(t) kommandiertes Signal
subplot(3,1,1)
plot(t,u,'b-','Linewidth',1)
legend('Drehzahlbeschleunigung Motor+Drehscheibe','location','ne',...
       'interpreter','latex')
ylabel('u in $rad/s^2$','interpreter','latex')
xlabel('t in s','interpreter','latex')
title('Messung')
grid on
%% subplot 2
% display w_M(t) kommandiertes Motordrehzahl-Signal
subplot(3,1,2)
p=plot(t,ui,'-','Linewidth',1);
p.Color=[0 32 96]/255;
legend('Drehzahl Motor+Drehscheibe','location','ne','interpreter','latex')
ylabel('$\omega_M$ in rad/s','interpreter','latex')
xlabel('t in s','interpreter','latex')
grid on
%% subplot 3
% display y(t) gemessenes Ausgangssignal
subplot(3,1,3)
plot(t,y,'r-','Linewidth',1)
legend('Winkelposition Satellit','location','ne','interpreter','latex')
ylabel('y in deg','interpreter','latex')
xlabel('t in s','interpreter','latex')
grid on
```

MATLAB-Programm – identSystemvgl.m

```
%Mit diesem Skript werden identifizierte Systeme mit Messdaten verglichen
s=tf('s');

Gs=(0.17121*(1-0.03322*s))/(s*(1+0.008751*s)*(1+9.794*s)); %Strecken-Modell

t=mat(:,1); %Zeitvektor in s
u=mat(:,2); %kommandiertes Eingangssignal in rad/s^2
y=mat(:,3)*180/pi; %gemessenes Ausgangssignal in deg
ySimGs=lsim(Gs,u,t)*180/pi; %simuliertes Ausgangssignal in deg

figure('NumberTitle','off','Name','mat')
%% subplot 1
% plot u(t) kommandiertes Eingangssignal
subplot(3,1,1)
hold on
plot(t,u,'b-','Linewidth',1)
hold off
ylabel('u','interpreter','latex')
legend('u','location','ne','interpreter','latex')
ylabel('u in $rad/s^2$','interpreter','latex')
xlabel('t in s','interpreter','latex')
title('Messung')
grid on
%% subplot 2
% plot y(t) gemessenes und simuliertes Ausgangssignal
subplot(3,1,2)
plot(t,y,'r-','Linewidth',2)
hold on
p=plot(t,ySimGs,'--','Linewidth',1);
p.Color=[0 176 80]/255;
hold off
legend('Mess','Sim','location','ne','interpreter','latex')
ylabel('y in deg','interpreter','latex')
xlabel('t in s','interpreter','latex')
grid on
%% subplot 3
% plot e(t) Fehler zwischen Messung und Simulation
subplot(3,1,3)
hold on
plot(t,(y-ySimGs),'black-.','Linewidth',1)
hold off
legend('Error','location','ne','interpreter','latex')
ylabel('e in deg','interpreter','latex')
xlabel('t in s','interpreter','latex')
grid on
```

MATLAB-Programm – reg_Messdaten_zusammenf.m

```
td=0.01; %Abtastrate

%Zusammenfassen von Stell-, Motordrehzahl-, Ausgangs-, Sollwertsignal
mat_reg=[(0:td:(length(Angles)-1)*td)',Acc,Speed,Angles,sollwert];
%Zusammenfassen der PID-Stellsignale
mat_pid=[(0:td:(length(Angles)-1)*td)',Pwert,Iwert,Dwert];
```

MATLAB-Programm – mat_reg_plot.m

```
%Plot für Regelung der Messdaten und Eingangsignalen
%u,w_M u yMess, ySim u w-yMess, yMess-ySim

zzG=1; %Soll gesamtes Messdatensatz gezeigt werden? 0=nein; 1=ja;
ssim=1;%Soll simulierter Verlauf gezeigt werden? 0=nein; 1=ja;
%%
if (zzG)

mat_regg=mat_reg(1000:end,:);
mat_Sim=simAngles(1000:end,:);

else
%%
gg=1000;
mat_regg=mat_reg(gg-50:end,:);
mat_Sim=simAngles(gg-50:end,:);
zz=550;
mat_regg=mat_regg(1:zz,:);
mat_Sim=mat_Sim(1:zz,:);
end
%%
t=mat_regg(:,1);%Zeitvektor in s
u=mat_regg(:,2);%kommandiertes Eingangssignal in rad/s^2
w_M=mat_regg(:,3);%kommandiertes Motordrehzahlsignal in rad/s
yMess=mat_regg(:,4)*180/pi;%gemessenes Ausgangssignal in deg
ySim=mat_Sim;%simuliertes Ausgangssignal in deg

sllwert=mat_regg(:,5)*180/pi;%Sollwertsignal in deg

figure('NumberTitle','off','Name','mat')
%% subplot 1
% plot u(t) kommandiertes Signal
subplot(4,1,1)
plot(t,u,'b-','Linewidth',1)
legend('Beschleunigung Motor+Drehscheibe','location','ne','interpreter','latex')
ylabel('u in $rad/s^2$','interpreter','latex')
xlabel('t in s','interpreter','latex')
title('Messung')
grid on

%% subplot 2
% plot w_M(t) kommandiertes Signal
subplot(4,1,2)
p=plot(t,w_M,'-','Linewidth',1);
```

```
p.Color=[0 32 96]/255;
legend('Drehzahl Motor+Drehscheibe','location','ne','interpreter','latex')
ylabel('$\omega_M$ in rad/s','interpreter','latex')
xlabel('t in s','interpreter','latex')
grid on
%% subplot 3
% plot sollwert u yMess u ySim
subplot(4,1,3)
p=plot(t,sllwert,'-','Linewidth',1);
p.Color=[255 116 0]/255;
hold on
plot(t,yMess,'r-','Linewidth',1)
hold off
if (ssim)
hold on
p=plot(t,ySim,'-','Linewidth',1);
p.Color=[0 176 80]/255;
hold off
end
legend('w','yMess','ySim','location','ne','interpreter','latex')
ylabel('y in deg','interpreter','latex')
xlabel('t in s','interpreter','latex')
grid on

%% subplot 4
% plot w-yMess und yMess-ySim
subplot(4,1,4)
p=plot(t,sllwert-yMess,'--','Linewidth',1);
p.Color=[128 0 0]/255;
if (ssim)
hold on
p=plot(t,yMess-ySim,'--','Linewidth',1);
p.Color=[0 100 0]/255;
hold off
end
legend('w-yMess','yMess-ySim','location','ne','interpreter','latex')
ylabel('e in deg','interpreter','latex')
xlabel('t in s','interpreter','latex')
grid on
```

Literaturverzeichnis

- [1] <https://products.bnl-bearings.com/products/>. Zugriff am 09.12.2022
- [2] https://www.multiplex-rc.de/userdata/filegallery/original/851_315082-bediensanleitung-roxxy-bl-outrunner-c35-48-1150kv--de-en-fr-it-es.pdf. Zugriff am 09.12.2022
- [3] http://www.hacker-motor.com/daten/sonstiges/Manual_TopFuel-LiPo_2014.pdf. Zugriff am 09.12.2022
- [4] <http://en.flycolor.net/uploadfile/201710/1bb2160b3e.pdf>. Zugriff am 09.12.2022
- [5] R. Hagl, Elektrische Antriebstechnik, 3., überarbeitete und erweiterte Auflage. München, Hanser, 2021.
- [6] https://bluerobotics.com/wp-content/uploads/2018/10/BLHeli_S-manual-SiLabs-Rev16.x.pdf. Zugriff am 09.12.2022
- [7] <https://bigamart.com/product/haljia-as5600-magnetic-encoder-magnetic-induction-angle-measurement-sensor-module-magnetized-encoder-12-bit-high-precision-with-magnet/>. Zugriff am 09.12.2022
- [8] <https://www.berrybase.de/raspberry-pi-zero-wh>. Zugriff am 09.12.2022
- [9] <https://www.pisugar.com/>. Zugriff am 09.12.2022
- [10] C. Eller, Holzmann/Meyer/Schumpich Technische Mechanik Kinematik und Kinetik, 12., überarbeitende und erweiterte Auflage. Wiesbaden, Springer Vieweg, 2016.
- [11] R. Isermann, Mechatronische Systeme: Grundlagen, 2. Vollständig neu bearbeitete Auflage. Springer, 2008.
- [12] P. Zentgraf. Ein neues Verfahren zur Modellierung linearer Systeme - Experimentelle Modellbildung mit Ein- und Ausgangsdaten. atp magazin, 2019.
- [13] F.A. King und P. Zentgraf, Skript zur Vorlesung Regelungstechnik I. Rosenheim, Sommersemester 2020.
- [14] M. Krettek, Entwicklung mobiler und einfacher regelungstechnischer Demonstrationsversuche auf einem Arduino-Board mit MATLAB/Simulink-Anbindung. Rosenheim, 2019

